The Journal of Machine Learning Research
Print-Archive Edition

Volume 20
Issues 93–184

Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Print-Archive Edition
Volume 20
Issues 93–184

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this edition are articles published electronically in JMLR in 2019.

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

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

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

Collection copyright © 2019 The Journal of Machine Learning Research, Inc. and Microtome Publishing.
Individual articles copyright © 2019 by their respective authors and distributed under a Creative Commons Attribution 4.0 International License.

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

JMLR Editorial Board

Editor-in-Chief
Francis Bach, Centre de Recherche INRIA de Paris

Editor-in-Chief
David Blei, Columbia University

Editor-in-Chief
Bernhard Schölkopf, MPI for Intelligent Systems, Germany

Managing Editor
Fabian Pedregosa, Google Reseach

Editorial Assistant
Karin Bierig, MPI for Intelligent Systems

Production Editor
Alp Kucukelbir, Fero Labs and Columbia University

Production Staff
Trevor Campbell, Massachusetts Institute of Technology **Behzad Tabibian**, MPI for Intelligent Systems **Hoon Cho**, Massachusetts Institute of Technology

Web Master
Alp Kucukelbir, Fero Labs and Columbia University **Fabian Pedregosa**, Google Reseach

JMLR Action Editors
Ryan Adams, Princeton University, USA **Shivani Agarwal**, University of Pennsylvania, USA **Edoardo M. Airoldi**, Harvard University, USA **Genevera Allen**, Rice University, USA **Anima Anandkumar**, California Institute of Technology, USA **Peter Auer**, University of Leoben, Austria **Samy Bengio**, Google Research, USA **Yoshua Bengio**, Université de Montréal, Canada **Jeff Bilmes**, University of Washington, USA **Karsten Borgwardt**, ETH Zurich, Switzerland **Miguel A. Carreira-Perpinan**, University of California, Merced, USA **Alexander Clark**, King's College London, UK **Corinna Cortes**, Google Research, USA **Koby Crammer**, Technion, Israel **John P. Cunningham**, Columbia University, USA **Arnak Dalalyan**, ENSAE / CREST, France **Inderjit S. Dhillon**, University of Texas, Austin, USA **Tina Eliassi-Rad**, Northeastern University **Gal Elidan**, Hebrew University, Israel **Charles Elkan**, University of California at San Diego, USA **Barbara Engelhardt**, Princeton University, USA **Rob Fergus**, New York University, USA **Kenji Fukumizu**, The Institute of Statistical Mathematics, Japan **Amir Globerson**, Tel Aviv University, Israel **Moises Goldszmidt**, Microsoft Research, USA **Russ Greiner**, University of Alberta, Canada **Arthur Gretton**, University College London, UK **Maya Gupta**, Google Research, USA **Isabelle Guyon**, ClopiNet, USA **Zaid Harchaoui**, University of Washington **Moritz Hardt**, Google Research, USA **Philipp Hennig**, University of Tübingen **Bert Huang**, Virginia Tech, USA **Aapo Hyvärinen**, University of Helsinki, Finland **Tommi Jaakkola**, Massachusetts Institute of Technology, USA **Prateek Jain**, Microsoft Research, India **Stefanie Jegelka**, Massachusetts Institute of Technology, USA **Samuel Kaski**, Aalto University, Finland **Sathiya Keerthi**, Microsoft Research, USA **Emtiyaz**

Khan, RIKEN Center for Advanced Intelligence, Japan **George Konidaris**, Duke University, USA **Andreas Krause**, ETH Zurich, Switzerland **Sanjiv Kumar**, Google Research, USA **Simon Lacoste-Julien**, Quebec Institute of Artificial Intelligence, Canada **Christoph Lampert**, Institute of Science and Technology, Austria **Honglak Lee**, Google and University of Michigan, Ann Arbor **Daniel Lee**, University of Pennsylvania, USA **Qiang Liu**, Dartmouth College, USA **Gábor Lugosi**, Pompeu Fabra University, Spain **Michael Mahoney**, University of California at Berkeley, USA **Vikash K. Mansinghka**, Massachusetts Institute of Technology, USA **Julien Mairal**, INRIA **Shie Mannor**, Technion, Israel **Jon McAuliffe**, University of California at Berkeley, USA **Robert E. McCulloch**, University of Chicago, USA **Chris Meek**, Microsoft Research, USA **Qiaozhu Mei**, University of Michigan, USA **Vahab Mirrokni**, Google Research, USA **Mehryar Mohri**, New York University, USA **Joris Mooij**, University of Amsterdam, Netherlands **Sayan Mukherjee**, Duke University, USA **Boaz Nadler**, Weizmann Institute of Science, Israel **Long Nguyen**, University of Michigan, USA **Sebastian Nowozin**, Google Research **Una-May O'Reilly**, Massachusetts Institute of Technology, USA **Laurent Orseau**, Google Deepmind, USA **Jie Peng**, University of California, Davis, USA **Jan Peters**, Technische Universitaet Darmstadt, Germany **Avi Pfeffer**, Charles River Analytics, USA **Joelle Pineau**, McGill University, Canada **Massimiliano Pontil**, Istituto Italiano di Tecnologia (Italy), University College London (UK) **Pushmeet Kohli**, DeepMind **Luc de Raedt**, Katholieke Universiteit Leuven, Belgium **Garvesh Raskutti**, University of Wisconsin-Madison, USA **Ben Recht**, University of California, Berkeley, USA **Lorenzo Rosasco**, Massachusetts Institute of Technology, USA **Ruslan Salakhutdinov**, University of Toronto, Canada **Sujay Sanghavi**, University of Texas, Austin, USA **Marc Schoenauer**, INRIA Saclay, France **Ohad Shamir**, Weizmann Institute of Science, Israel **John Shawe-Taylor**, University College London, UK **Xiaotong Shen**, University of Minnesota, USA **David Sontag**, Massachusetts Institute of Technology **Peter Spirtes**, Carnegie Mellon University, USA **Suvrit Sra**, Massachusetts Institute of Technology, USA **Ingo Steinwart**, University of Stuttgart, Germany **Amos Storkey**, University of Edinburgh, UK **Erik Sudderth**, University of California, Irvine, USA **Csaba Szepesvari**, University of Alberta, Canada **Olivier Teytaud**, INRIA Saclay, France **Ambuj Tewari**, University of Michigan **Ivan Titov**, University of Amsterdam, Netherlands **Ryan Tibshirani**, Carnegie Mellon University **Ryota Tomioka**, Microsoft Research Cambridge, UK **Koji Tsuda**, National Institute of Advanced Industrial Science and Technology, Japan **Nicolas Vayatis**, ENS Cachan, France **Jean-Philippe Vert**, Google Research, France **S V N Vishwanathan**, Purdue University, USA **Manfred Warmuth**, University of California at Santa Cruz, USA **Kilian Weinberger**, Cornell University, USA **David Wipf**, Microsoft Research Asia, China **Daniela Witten**, University of Washington **Frank Wood**, University of British Columbia **Tong Zhang**, Baidu Inc, China **Zhihua Zhang**, Peking University, China

JMLR MLOSS Editors

Alexandre Gramfort, INRIA, Université Paris-Saclay, France **Antti Honkela**, University of Helsinki, Finland **Balázs Kégl**, CNRS / Université Paris-Saclay, France **Andreas C. Müller**, Columbia University, USA **Cheng Soon Ong**, Australian National University, Australia

JMLR Editorial Board

Naoki Abe, IBM TJ Watson Research Center, USA **Yasemin Altun**, Google Inc, Switzerland **Jean-Yves Audibert**, CERTIS, France **Jonathan Baxter**, Australia National University, Australia **Richard K. Belew**, University of California at San Diego, USA **Kristin Bennett**, Rensselaer Polytechnic Institute, USA **Christopher M. Bishop**, Microsoft Research, Cambridge, UK **Lashon Booker**, The Mitre Corporation, USA **Henrik Boström**, Stockholm University/KTH, Sweden **Craig Boutilier**,

Google Research, USA **John Patrick Cunningham**, Columbia University, USA **Nello Cristianini**, University of Bristol, UK **Peter Dayan**, University College, London, UK **Dennis DeCoste**, eBay Research, USA **Thomas Dietterich**, Oregon State University, USA **Saso Dzeroski**, Jozef Stefan Institute, Slovenia **Ran El-Yaniv**, Technion, Israel **Peter Flach**, Bristol University, UK **Dan Geiger**, Technion, Israel **Claudio Gentile**, Università degli Studi dell'Insubria, Italy **Sally Goldman**, Google Research, USA **Thore Graepel**, Google DeepMind and University College London, UK **Tom Griffiths**, University of California at Berkeley, USA **Carlos Guestrin**, University of Washington, USA **Stefan Harmeling**, University of Düsseldorf, Germany **David Heckerman**, Microsoft Research, USA **Katherine Heller**, Duke University, USA **Philipp Hennig**, MPI for Intelligent Systems, Germany **Larry Hunter**, University of Colorado, USA **Jens Kober**, Delft University of Technology, Netherlands **Risi Kondor**, University of Chicago, USA **Aryeh Kontorovich**, Ben-Gurion University of the Negev, Israel **Samory Kpotufe**, Princeton University, USA **John Lafferty**, University of Chicago, USA **Erik Learned-Miller**, University of Massachusetts, Amherst, USA **Fei Fei Li**, Stanford University, USA **Yi Lin**, University of Wisconsin, USA **Wei-Yin Loh**, University of Wisconsin, USA **Richard Maclin**, University of Minnesota, USA **Sridhar Mahadevan**, University of Massachusetts, Amherst, USA **Vikash Mansingkha**, Massachusetts Institute of Technology, USA **Yishay Mansour**, Tel-Aviv University, Israel **Jon McAuliffe**, University of California, Berkeley, USA **Andrew McCallum**, University of Massachusetts, Amherst, USA **Raymond J. Mooney**, University of Texas, Austin, USA **Klaus-Robert Muller**, Technical University of Berlin, Germany **Kevin Murphy**, Google, USA **Guillaume Obozinski**, Ecole des Ponts - ParisTech, France **Pascal Poupart**, University of Waterloo, Canada **Konrad Rieck**, University of Göttingen, Germany **Cynthia Rudin**, Massachusetts Institute of Technology, USA **Suchi Saria**, Johns Hopkins University, USA **Robert Schapire**, Princeton University, USA **Fei Sha**, University of Southern California, USA **Shai Shalev-Shwartz**, Hebrew University of Jerusalem, Israel **Padhraic Smyth**, University of California, Irvine, USA **Bharath Sriperumbudur**, Pennsylvania State University, USA **Alexander Statnikov**, New York University, USA **Jean-Philippe Vert**, Mines ParisTech, France **Martin J. Wainwright**, University of California at Berkeley, USA **Chris Watkins**, Royal Holloway, University of London, UK **Max Welling**, University of Amsterdam, Netherlands **Stefan Wrobel**, Fraunhofer IAIS and University of Bonn, Germany **Chris Williams**, University of Edinburgh, UK **Alice Zheng**, GraphLab, USA

JMLR Advisory Board

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

Journal of Machine Learning Research

Volume 20

Part A

20(1):1–30 **Adaptation Based on Generalized Discrepancy**
Corinna Cortes, Mehryar Mohri, Andrés Muñoz Medina

20(2):1–52 **Transport Analysis of Infinitely Deep Neural Network**
Sho Sonoda, Noboru Murata

20(3):1–44 **Parsimonious Online Learning with Kernels via Sparse Projections in Function Space**
Alec Koppel, Garrett Warnell, Ethan Stump, Alejandro Ribeiro

20(4):1–45 **Convergence Rate of a Simulated Annealing Algorithm with Noisy Observations**
Clément Bouttier, Ioana Gavra

20(5):1–37 **Non-Convex Projected Gradient Descent for Generalized Low-Rank Tensor Regression**
Han Chen, Garvesh Raskutti, Ming Yuan

20(6):1–22 **scikit-multilearn: A Python library for Multi-Label Classification**
Piotr Szymański, Tomasz Kajdanowicz

20(7):1–45 **Scalable Approximations for Generalized Linear Problems**
Murat Erdogdu, Mohsen Bayati, Lee H. Dicker

20(8):1–39 **Forward-Backward Selection with Early Dropping**
Giorgos Borboudakis, Ioannis Tsamardinos

20(9):1–49 **Dynamic Pricing in High-dimensions**
Adel Javanmard, Hamid Nazerzadeh

20(10):1–44 **Graphical Lasso and Thresholding: Equivalence and Closed-form Solutions**
Salar Fattahi, Somayeh Sojoudi

20(11):1–23 **An Approach to One-Bit Compressed Sensing Based on Probably Approximately Correct Learning Theory**
Mehmet Eren Ahsen, Mathukumalli Vidyasagar

20(12):1–49 **Scalable Kernel K-Means Clustering with Nystrom Approximation: Relative-Error Bounds**
Shusen Wang, Alex Gittens, Michael W. Mahoney

20(13):1–34 **Train and Test Tightness of LP Relaxations in Structured Prediction**
Ofer Meshi, Ben London, Adrian Weller, David Sontag

20(14):1–37 **Approximations of the Restless Bandit Problem**
Steffen Grünewälder, Azadeh Khaleghi

20(15):1–38 **Automated Scalable Bayesian Inference via Hilbert Coresets**
Trevor Campbell, Tamara Broderick

20(16):1–24 **Smooth neighborhood recommender systems**
Ben Dai, Junhui Wang, Xiaotong Shen, Annie Qu

20(17):1–38 **Delay and Cooperation in Nonstochastic Bandits**
Nicolò Cesa-Bianchi, Claudio Gentile, Yishay Mansour

20(18):1–29 **Multiplicative local linear hazard estimation and best one-sided cross-validation**
Maria Luz Gámiz, María Dolores Martínez-Miranda, Jens Perch Nielsen

20(19):1–5 **spark-crowd: A Spark Package for Learning from Crowdsourced Big Data**
Enrique G. Rodrigo, Juan A. Aledo, José A. Gámez

20(20):1–33 **Accelerated Alternating Projections for Robust Principal Component Analysis**
HanQin Cai, Jian-Feng Cai, Ke Wei

20(21):1–55 **Spectrum Estimation from a Few Entries**
Ashish Khetan, Sewoong Oh

20(22):1–36 **Random Feature-based Online Multi-kernel Learning in Environments with Unknown Dynamics**
Yanning Shen, Tianyi Chen, Georgios B. Giannakis

20(23):1–38 **Determining the Number of Latent Factors in Statistical Multi-Relational Learning**
Chengchun Shi, Wenbin Lu, Rui Song

20(24):1–29 **Joint PLDA for Simultaneous Modeling of Two Factors**
Luciana Ferrer, Mitchell McLaren

20(25):1–49 **Group Invariance, Stability to Deformations, and Complexity of Deep Convolutional Representations**
Alberto Bietti, Julien Mairal

20(26):1–6 **TensorLy: Tensor Learning in Python**
Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, Maja Pantic

20(27):1–42 **Monotone Learning with Rectified Wire Networks**
Veit Elser, Dan Schmidt, Jonathan Yedidia

20(28):1–6 **Pyro: Deep Universal Probabilistic Programming**
Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, Noah D. Goodman

20(29):1–28 **Iterated Learning in Dynamic Social Networks**
Bernard Chazelle, Chu Wang

20(30):1–34 **Exact Clustering of Weighted Graphs via Semidefinite Programming**
Aleksis Pirinen, Brendan Ames

20(31):1–45 **Kernels for Sequentially Ordered Data**
Franz J. Kiraly, Harald Oberhauser

20(32):1–50 **NetSDM: Semantic Data Mining with Network Analysis**
Jan Kralj, Marko Robnik-Sikonja, Nada Lavrac

20(33):1–38 **The Relationship Between Agnostic Selective Classification, Active Learning and the Disagreement Coefficient**
Roei Gelbhart, Ran El-Yaniv

20(34):1–44 **Matched Bipartite Block Model with Covariates**
Zahra S. Razaee, Arash A. Amini, Jingyi Jessica Li

20(35):1–93 **Optimal Policies for Observing Time Series and Related Restless Bandit Problems**
Christopher R. Dance, Tomi Silander

20(36):1–37 **A New Approach to Laplacian Solvers and Flow Problems**
Patrick Rebeschini, Sekhar Tatikonda

20(37):1–59 **A Well-Tempered Landscape for Non-convex Robust Subspace Recovery**
Tyler Maunu, Teng Zhang, Gilad Lerman

20(38):1–27 **Approximation Hardness for A Class of Sparse Optimization Problems**
Yichen Chen, Yinyu Ye, Mengdi Wang

20(39):1–40 **A Bootstrap Method for Error Estimation in Randomized Matrix Multiplication**
Miles E. Lopes, Shusen Wang, Michael W. Mahoney

20(40):1–47 **Stochastic Modified Equations and Dynamics of Stochastic Gradient Algorithms I: Mathematical Foundations**
Qianxiao Li, Cheng Tai, Weinan E

20(41):1–57 **Decontamination of Mutual Contamination Models**
Julian Katz-Samuels, Gilles Blanchard, Clayton Scott

20(42):1–56 **Utilizing Second Order Information in Minibatch Stochastic Variance Reduced Proximal Iterations**
Jialei Wang, Tong Zhang

20(43):1–58 **DSCOVR: Randomized Primal-Dual Block Coordinate Algorithms for Asynchronous Distributed Optimization**
Lin Xiao, Adams Wei Yu, Qihang Lin, Weizhu Chen

20(44):1–5 **Picasso: A Sparse Learning Library for High Dimensional Data Analysis in R and Python**
Jason Ge, Xingguo Li, Haoming Jiang, Han Liu, Tong Zhang, Mengdi Wang, Tuo Zhao

20(45):1–41 **Robust Frequent Directions with Application in Online Learning**
Luo Luo, Cheng Chen, Zhihua Zhang, Wu-Jun Li, Tong Zhang

20(46):1–36 **Boosted Kernel Ridge Regression: Optimal Learning Rates and Early Stopping**
Shao-Bo Lin, Yunwen Lei, Ding-Xuan Zhou

20(47):1–47 **Analysis of spectral clustering algorithms for community detection: the general bipartite setting**
Zhixin Zhou, Arash A.Amini

20(48):1–23 **Efficient augmentation and relaxation learning for individualized treatment rules using observational data**
Ying-Qi Zhao, Eric B. Laber, Yang Ning, Sumona Saha, Bruce E. Sands

20(49):1–24 **Using Simulation to Improve Sample-Efficiency of Bayesian Optimization for Bipedal Robots**
Akshara Rai, Rika Antonova, Franziska Meier, Christopher G. Atkeson

20(50):1–24 **No-Regret Bayesian Optimization with Unknown Hyperparameters**
Felix Berkenkamp, Angela P. Schoellig, Andreas Krause

20(51):1–18 **Bayesian Combination of Probabilistic Classifiers using Multivariate Normal Mixtures**
Gregor Pirš, Erik Štrumbelj

20(52):1–24 **Thompson Sampling Guided Stochastic Searching on the Line for Deceptive Environments with Applications to Root-Finding Problems**
Sondre Glimsdal, Ole-Christoffer Granmo

20(53):1–32 **Tunability: Importance of Hyperparameters of Machine Learning Algorithms**
Philipp Probst, Anne-Laure Boulesteix, Bernd Bischl

20(54):1–31 **Deep Reinforcement Learning for Swarm Systems**
Maximilian Hüttenrauch, Adrian Šošić, Gerhard Neumann

20(55):1–21 **Neural Architecture Search: A Survey**
Thomas Elsken, Jan Hendrik Metzen, Frank Hutter

20(56):1–23 **Near Optimal Frequent Directions for Sketching Dense and Sparse Matrices**
Zengfeng Huang

20(57):1–31 **Multi-class Heterogeneous Domain Adaptation**
Joey Tianyi Zhou, Ivor W. Tsang, Sinno Jialin Pan, Mingkui Tan

20(58):1–49 **The Common-directions Method for Regularized Empirical Risk Minimization**
Po-Wei Wang, Ching-pei Lee, Chih-Jen Lin

20(59):1–36 **Kernel Approximation Methods for Speech Recognition**
Avner May, Alireza Bagheri Garakani, Zhiyun Lu, Dong Guo, Kuan Liu, Aurélien Bellet, Linxi Fan, Michael Collins, Daniel Hsu, Brian Kingsbury, Michael Picheny, Fei Sha

20(60):1–49 **Robust Estimation of Derivatives Using Locally Weighted Least Absolute Deviation Regression**
WenWu Wang, Ping Yu, Lu Lin, Tiejun Tong

20(61):1–42 **The Sup-norm Perturbation of HOSVD and Low Rank Tensor Denoising**
Dong Xia, Fan Zhou

20(62):1–37 **Multi-scale Online Learning: Theory and Applications to Online Auctions and Pricing**
Sébastien Bubeck, Nikhil R. Devanur, Zhiyi Huang, Rad Niazadeh

20(63):1–17 **Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks**
Peter L. Bartlett, Nick Harvey, Christopher Liaw, Abbas Mehrabian

20(64):1–32 **A Representer Theorem for Deep Kernel Learning**
Bastian Bohn, Michael Griebel, Christian Rieger

20(65):1–50 **Active Learning for Cost-Sensitive Classification**
Akshay Krishnamurthy, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, John Langford

20(66):1–38 **Proximal Distance Algorithms: Theory and Practice**
Kevin L. Keys, Hua Zhou, Kenneth Lange

20(67):1–28 **Learnability of Solutions to Conjunctive Queries**
Hubie Chen, Matthew Valeriote

20(68):1–55 **Variance-based Regularization with Convex Objectives**
John Duchi, Hongseok Namkoong

20(69):1–39 **On Consistent Vertex Nomination Schemes**
Vince Lyzinski, Keith Levin, Carey E. Priebe

20(70):1–33 **Semi-Analytic Resampling in Lasso**
Tomoyuki Obuchi, Yoshiyuki Kabashima

20(71):1–42 **Lazifying Conditional Gradient Algorithms**
Gábor Braun, Sebastian Pokutta, Daniel Zink

20(72):1–47 **Redundancy Techniques for Straggler Mitigation in Distributed Optimization and Learning**
Can Karakus, Yifan Sun, Suhas Diggavi, Wotao Yin

20(73):1–46 **Analysis of Langevin Monte Carlo via Convex Optimization**
Alain Durmus, Szymon Majewski, Błażej Miasojedow

20(74):1–25 **Deep Optimal Stopping**
Sebastian Becker, Patrick Cheridito, Arnulf Jentzen

20(75):1–42 **Fairness Constraints: A Flexible Approach for Fair Classification**
Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, Krishna P. Gummadi

20(76):1–70 **Generalized Score Matching for Non-Negative Data**
Shiqing Yu, Mathias Drton, Ali Shojaie

20(77):1–33 **Nonuniformity of P-values Can Occur Early in Diverging Dimensions**
Yingying Fan, Emre Demirkaya, Jinchi Lv

20(78):1–39 **Prediction Risk for the Horseshoe Regression**
Anindya Bhadra, Jyotishka Datta, Yunfan Li, Nicholas G. Polson, Brandon Willard

20(79):1–33 **Quantification Under Prior Probability Shift: the Ratio Estimator and its Extensions**
Afonso Fernandes Vaz, Rafael Izbicki, Rafael Bassi Stern

20(80):1–37 **Learning to Match via Inverse Optimal Transport**
Ruilin Li, Xiaojing Ye, Haomin Zhou, Hongyuan Zha

20(81):1–8 **Tight Lower Bounds on the VC-dimension of Geometric Set Systems**
Mónika Csikós, Nabil H. Mustafa, Andrey Kupavskii

20(82):1–5 **SMART: An Open Source Data Labeling Platform for Supervised Learning**
Rob Chew, Michael Wenger, Caroline Kery, Jason Nance, Keith Richards, Emily Hadley, Peter Baumgartner

20(83):1–28 **On the optimality of the Hedge algorithm in the stochastic regime**
Jaouad Mourtada, Stéphane Gaïffas

20(84):1–40 **Differentiable Game Mechanics**
Alistair Letcher, David Balduzzi, Sébastien Racanière, James Martens, Jakob Foerster, Karl Tuyls, Thore Graepel

20(85):1–35 **Bayesian Space-Time Partitioning by Sampling and Pruning Spanning Trees**
Leonardo V. Teixeira, Renato M. Assunção, Rosangela H. Loschi

20(86):1–62 **Streaming Principal Component Analysis From Incomplete Data**
Armin Eftekhari, Gregory Ongie, Laura Balzano, Michael B. Wakin

20(87):1–30 **An asymptotic analysis of distributed nonparametric methods**
Botond Szabó, Harry van Zanten

20(88):1–26 **Model Selection via the VC Dimension**
Merlin Mpoudeu, Bertrand Clarke

20(89):1–43 **Dependent relevance determination for smooth and structured sparse regression**
Anqi Wu, Oluwasanmi Koyejo, Jonathan Pillow

20(90):1–56 **A Particle-Based Variational Approach to Bayesian Non-negative Matrix Factorization**
Muhammad A Masood, Finale Doshi-Velez

20(91):1–39 **Best Arm Identification for Contaminated Bandits**
Jason Altschuler, Victor-Emmanuel Brunel, Alan Malek

20(92):1–6 **AffectiveTweets: a Weka Package for Analyzing Affect in Tweets**
Felipe Bravo-Marquez, Eibe Frank, Bernhard Pfahringer, Saif M. Mohammad

Part B

20(93):1–8 **iNNvestigate Neural Networks!**
Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, Pieter-Jan Kindermans

20(94):1–34 **Simultaneous Private Learning of Multiple Concepts**
Mark Bun, Kobbi Nissim, Uri Stemmer

20(95):1–41 **High-Dimensional Poisson Structural Equation Model Learning via ℓ_1-Regularized Regression**
Gunwoong Park, Sion Park

20(96):1–7 **PyOD: A Python Toolbox for Scalable Outlier Detection**
Yue Zhao, Zain Nasrullah, Zheng Li

20(97):1–22 **Relative Error Bound Analysis for Nuclear Norm Regularized Matrix Completion**
Lijun Zhang, Tianbao Yang, Rong Jin, Zhi-Hua Zhou

20(98):1–63 **Adaptive Geometric Multiscale Approximations for Intrinsically Low-dimensional Data**
Wenjing Liao, Mauro Maggioni

20(99):1–51 **Change Surfaces for Expressive Multidimensional Changepoints and Counterfactual Prediction**
William Herlands, Daniel B. Neill, Hannes Nickisch, Andrew Gordon Wilson

20(100):1–31 **Hamiltonian Monte Carlo with Energy Conserving Subsampling**
Khue-Dung Dang, Matias Quiroz, Robert Kohn, Minh-Ngoc Tran, Mattias Villani

20(101):1–43 **Low Permutation-rank Matrices: Structural Properties and Noisy Completion**
Nihar B. Shah, Sivaraman Balakrishnan, Martin J. Wainwright

20(102):1–56 **Non-Convex Matrix Completion and Related Problems via Strong Duality**
Maria-Florina Balcan, Yingyu Liang, Zhao Song, David P. Woodruff, Hongyang Zhang

20(103):1–68 **Regularization via Mass Transportation**
Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, Peyman Mohajerin Esfahani

20(104):1–34 **Complete Search for Feature Selection in Decision Trees**
Salvatore Ruggieri

20(105):1–23 **Optimal Transport: Fast Probabilistic Approximation with Exact Solvers**
Max Sommerfeld, Jörn Schrieber, Yoav Zemel, Axel Munk

20(106):1–25 **Solving the OSCAR and SLOPE Models Using a Semismooth Newton-Based Augmented Lagrangian Method**
Ziyan Luo, Defeng Sun, Kim-Chuan Toh, Naihua Xiu

20(107):1–34 **Scalable Interpretable Multi-Response Regression via SEED**
Zemin Zheng, M. Taha Bahadori, Yan Liu, Jinchi Lv

20(108):1–30 **Maximum Likelihood for Gaussian Process Classification and Generalized Linear Mixed Models under Case-Control Sampling**
Omer Weissbrod, Shachar Kaufman, David Golan, Saharon Rosset

20(109):1–30 **Learning Unfaithful K-separable Gaussian Graphical Models**
De Wen Soh, Sekhar Tatikonda

20(110):1–30 **A Representer Theorem for Deep Neural Networks**
Michael Unser

20(111):1–40 **An Efficient Two Step Algorithm for High Dimensional Change Point Regression Models Without Grid Search**
Abhishek Kaul, Venkata K. Jandhyala, Stergios B. Fotopoulos

20(112):1–49 **Measuring the Effects of Data Parallelism on Neural Network Training**
Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, George E. Dahl

20(113):1–41 **Distributed Inference for Linear Support Vector Machine**
Xiaozhou Wang, Zhuoyi Yang, Xi Chen, Weidong Liu

20(114):1–34 **Sharp Restricted Isometry Bounds for the Inexistence of Spurious Local Minima in Nonconvex Matrix Recovery**
Richard Y. Zhang, Somayeh Sojoudi, Javad Lavaei

20(115):1–58 **Learning Attribute Patterns in High-Dimensional Structured Latent Attribute Models**
Yuqi Gu, Gongjun Xu

20(116):1–42 **Graph Reduction with Spectral and Cut Guarantees**
Andreas Loukas

20(117):1–63 **Generic Inference in Latent Gaussian Process Models**
Edwin V. Bonilla, Karl Krauth, Amir Dezfouli

20(118):1–34 **Binarsity: a penalization for one-hot encoded features in linear supervised learning**
Mokhtar Z. Alaya, Simon Bussy, Stéphane Gaïffas, Agathe Guilloux

20(119):1–38 **Layer-Wise Learning Strategy for Nonparametric Tensor Product Smoothing Spline Regression and Graphical Models**
Kean Ming Tan, Junwei Lu, Tong Zhang, Han Liu

20(120):1–49 **Ivanov-Regularised Least-Squares Estimators over Large RKHSs and Their Interpolation Spaces**
Stephen Page, Steffen Grünewälder

20(121):1–39 **Scaling Up Sparse Support Vector Machines by Simultaneous Feature and Sample Reduction**
Bin Hong, Weizhong Zhang, Wei Liu, Jieping Ye, Deng Cai, Xiaofei He, Jie Wang

20(122):1–55 **Approximate Profile Maximum Likelihood**
Dmitri S. Pavlichin, Jiantao Jiao, Tsachy Weissman

20(123):1–26 **ADMMBO: Bayesian Optimization with Unknown Constraints using ADMM**
Setareh Ariafar, Jaume Coll-Font, Dana Brooks, Jennifer Dy

20(124):1–62 **Deep Exploration via Randomized Value Functions**
Ian Osband, Benjamin Van Roy, Daniel J. Russo, Zheng Wen

20(125):1–5 **ORCA: A Matlab/Octave Toolbox for Ordinal Regression**
Javier Sánchez-Monedero, Pedro A. Gutiérrez, María Pérez-Ortiz

20(126):1–45 **Learning Representations of Persistence Barcodes**
Christoph D. Hofer, Roland Kwitt, Marc Niethammer

20(127):1–32 **Causal Learning via Manifold Regularization**
Steven M. Hill, Chris J. Oates, Duncan A. Blythe, Sach Mukherjee

20(128):1–73 **Unsupervised Basis Function Adaptation for Reinforcement Learning**
Edward Barker, Charl Ras

20(129):1–30 **Time-to-Event Prediction with Neural Networks and Cox Regression**
Håvard Kvamme, Ørnulf Borgan, Ida Scheel

20(130):1–47 **Logical Explanations for Deep Relational Machines Using Relevance Information**
Ashwin Srinivasan, Lovekesh Vig, Michael Bain

20(131):1–27 **Decoupling Sparsity and Smoothness in the Dirichlet Variational Autoencoder Topic Model**
Sophie Burkhardt, Stefan Kramer

20(132):1–59 **More Efficient Estimation for Logistic Regression with Optimal Subsamples**
HaiYing Wang

20(133):1–34 **Spurious Valleys in One-hidden-layer Neural Network Optimization Landscapes**
Luca Venturi, Afonso S. Bandeira, Joan Bruna

20(134):1–47 **Stochastic Variance-Reduced Cubic Regularization Methods**
Dongruo Zhou, Pan Xu, Quanquan Gu

20(135):1–36 **Gaussian Processes with Linear Operator Inequality Constraints**
Christian Agrell

20(136):1–37 **Local Regularization of Noisy Point Clouds: Improved Global Geometric Estimates and Data Analysis**
Nicolás García Trillos, Daniel Sanz-Alonso, Ruiyi Yang

20(137):1–68 **Multiclass Boosting: Margins, Codewords, Losses, and Algorithms**
Mohammad Saberian, Nuno Vasconcelos

20(138):1–29 **Generalized Maximum Entropy Estimation**
Tobias Sutter, David Sutter, Peyman Mohajerin Esfahani, John Lygeros

20(139):1–62 **Decentralized Dictionary Learning Over Time-Varying Digraphs**
Amir Daneshmand, Ying Sun, Gesualdo Scutari, Francisco Facchinei, Brian M. Sadler

20(140):1–81 **Nonparametric Bayesian Aggregation for Massive Data**
Zuofeng Shang, Botao Hao, Guang Cheng

20(141):1–43 **Provably Accurate Double-Sparse Coding**
Thanh V. Nguyen, Raymond K. W. Wong, Chinmay Hegde

20(142):1–39 **Model-free Nonconvex Matrix Completion: Local Minima Analysis and Applications in Memory-efficient Kernel PCA**
Ji Chen, Xiaodong Li

20(143):1–57 **Minimal Sample Subspace Learning: Theory and Algorithms**
Zhenyue Zhang, Yuqing Xia

20(144):1–30 **Convergence of Gaussian Belief Propagation Under General Pairwise Factorization: Connecting Gaussian MRF with Pairwise Linear Gaussian Model**
Bin Li, Yik-Chung Wu

20(145):1–30 **Bayesian Optimization for Policy Search via Online-Offline Experimentation**
Benjamin Letham, Eytan Bakshy

20(146):1–33 **Characterizing the Sample Complexity of Pure Private Learners**
Amos Beimel, Kobbi Nissim, Uri Stemmer

20(147):1–50 **Robustifying Independent Component Analysis by Adjusting for Group-Wise Stationary Noise**
Niklas Pfister, Sebastian Weichwald, Peter Bühlmann, Bernhard Schölkopf

20(148):1–43 **Collective Matrix Completion**
Mokhtar Z. Alaya, Olga Klopp

20(149):1–24 **On Asymptotic and Finite-Time Optimality of Bayesian Predictors**
Daniil Ryabko

20(150):1–75 **Learning Optimized Risk Scores**
Berk Ustun, Cynthia Rudin

20(151):1–49 **Nonparametric Estimation of Probability Density Functions of Random Persistence Diagrams**
Vasileios Maroulas, Joshua L Mike, Christopher Oballe

20(152):1–44 **High-dimensional Varying Index Coefficient Models via Stein's Identity**
Sen Na, Zhuoran Yang, Zhaoran Wang, Mladen Kolar

20(153):1–33 **Approximation Algorithms for Stochastic Clustering**
David G. Harris, Shi Li, Thomas Pensyl, Aravind Srinivasan, Khoa Trinh

20(154):1–52 **Convergence Guarantees for a Class of Non-convex and Non-smooth Optimization Problems**
Koulik Khamaru, Martin J. Wainwright

20(155):1–35 **Quantifying Uncertainty in Online Regression Forests**
Theodore Vasiloudis, Gianmarco De Francisci Morales, Henrik Boström

20(156):1–8 **SimpleDet: A Simple and Versatile Distributed Framework for Object Detection and Instance Recognition**
Yuntao Chen, Chenxia Han, Yanghao Li, Zehao Huang, Yi Jiang, Naiyan Wang, Zhaoxiang Zhang

20(157):1–28 **Simultaneous Phase Retrieval and Blind Deconvolution via Convex Programming**
Ali Ahmed, Alireza Aghasi, Paul Hand

20(158):1–7 **GraSPy: Graph Statistics in Python**
Jaewon Chung, Benjamin D. Pedigo, Eric W. Bridgeford, Bijan K. Varjavand, Hayden S. Helm, Joshua T. Vogelstein

20(159):1–31 **Optimal Convergence Rates for Convex Distributed Optimization in Networks**
Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, Laurent Massoulié

20(160):1–56 **Learning by Unsupervised Nonlinear Diffusion**
Mauro Maggioni, James M. Murphy

20(161):1–44 **Sparse Kernel Regression with Coefficient-based ℓ_q-regularization**
Lei Shi, Xiaolin Huang, Yunlong Feng, Johan A.K. Suykens

20(162):1–56 **A Kernel Multiple Change-point Algorithm via Model Selection**
Sylvain Arlot, Alain Celisse, Zaid Harchaoui

20(163):1–42 **Two-Layer Feature Reduction for Sparse-Group Lasso via Decomposition of Convex Sets**
Jie Wang, Zhanqiu Zhang, Jieping Ye

20(164):1–31 **The Reduced PC-Algorithm: Improved Causal Structure Learning in Large Random Networks**
Arjun Sondhi, Ali Shojaie

20(165):1–37 **On the Convergence of Gaussian Belief Propagation with Nodes of Arbitrary Size**
Francois Kamper, Sarel J. Steel, Johan A. du Preez

20(166):1–40 **Unsupervised Evaluation and Weighted Aggregation of Ranked Classification Predictions**
Mehmet Eren Ahsen, Robert M Vogel, Gustavo A Stolovitzky

20(167):1–46 **Stochastic Canonical Correlation Analysis**
Chao Gao, Dan Garber, Nathan Srebro, Jialei Wang, Weiran Wang

20(168):1–70 **Determinantal Point Processes for Coresets**
Nicolas Tremblay, Simon Barthelmé, Pierre-Olivier Amblard

20(169):1–26 **Embarrassingly Parallel Inference for Gaussian Processes**
Michael Minyi Zhang, Sinead A. Williamson

20(170):1–50 **DBSCAN: Optimal Rates For Density-Based Cluster Estimation**
Daren Wang, Xinyang Lu, Alessandro Rinaldo

20(171):1–37 **Shared Subspace Models for Multi-Group Covariance Estimation**
Alexander M. Franks, Peter Hoff

20(172):1–59 **Optimization with Non-Differentiable Constraints with Applications to Fairness, Recall, Churn, and Other Goals**
Andrew Cotter, Heinrich Jiang, Maya Gupta, Serena Wang, Taman Narayan, Seungil You, Karthik Sridharan

20(173):1–27 **Fast Automatic Smoothing for Generalized Additive Models**
Yousra El-Bachir, Anthony C. Davison

20(174):1–42 **Learning Overcomplete, Low Coherence Dictionaries with Linear Inference**
Jesse A. Livezey, Alejandro F. Bujan, Friedrich T. Sommer

20(175):1–6 **DataWig: Missing Value Imputation for Tables**
Felix Biessmann, Tammo Rukat, Phillipp Schmidt, Prathik Naidu, Sebastian Schelter, Andrey Taptunov, Dustin Lange, David Salinas

20(176):1–49 **New Convergence Aspects of Stochastic Gradient Algorithms**
Lam M. Nguyen, Phuong Ha Nguyen, Peter Richtárik, Katya Scheinberg, Martin Takáč, Marten van Dijk

20(177):1–81 **All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously**
Aaron Fisher, Cynthia Rudin, Francesca Dominici

20(178):1–29 **Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning**
Daniel C. Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, Ben Glocker

20(179):1–62 **Differentiable reservoir computing**
Lyudmila Grigoryeva, Juan-Pablo Ortega

20(180):1–7 **DPPy: DPP Sampling with Python**
Guillaume Gautier, Guillermo Polito, Rémi Bardenet, Michal Valko

20(181):1–23 **Neural Empirical Bayes**
Saeed Saremi, Aapo Hyvärinen

20(182):1–46 **Model Selection in Bayesian Neural Networks via Horseshoe Priors**
Soumya Ghosh, Jiayu Yao, Finale Doshi-Velez

20(183):1–42 **Log-concave sampling: Metropolis-Hastings algorithms are fast**
Raaz Dwivedi, Yuansi Chen, Martin J. Wainwright, Bin Yu

20(184):1–25 **Why do deep convolutional networks generalize so poorly to small image transformations?**
Aharon Azulay, Yair Weiss

Journal of Machine Learning Research 20 (2019) 1-8 Submitted 8/18; Revised 5/19; Published 5/19

iNNvestigate Neural Networks!

Maximilian Alber MAXIMILIAN.ALBER@TU-BERLIN.DE
Technische Universität Berlin, Machine Learning Group
10623 Berlin, Germany

Sebastian Lapuschkin SEBASTIAN.LAPUSCHKIN@HHI.FRAUNHOFER.DE
Fraunhofer Heinrich Hertz Institute, Video Coding and Analytics
10587 Berlin, Germany

Philipp Seegerer PHILIPP.SEEGERER@TU-BERLIN.DE
Miriam Hägele HAEGELE@TU-BERLIN.DE
Kristof T. Schütt KRISTOF.SCHUETT@TU-BERLIN.DE
Grégoire Montavon GREGOIRE.MONTAVON@TU-BERLIN.DE
Technische Universität Berlin, Machine Learning Group
10623 Berlin, Germany

Wojciech Samek WOJCIECH.SAMEK@HHI.FRAUNHOFER.DE
Fraunhofer Heinrich Hertz Institute, Video Coding and Analytics
10587 Berlin, Germany

Klaus-Robert Müller KLAUS-ROBERT.MUELLER@TU-BERLIN.DE
Technische Universität Berlin, Machine Learning Group
10623 Berlin, Germany
Korea University, Department of Brain and Cognitive Engineering
Seoul 02841, Korea
Max Planck Institute for Informatics
66123 Saarbrücken, Germany

Sven Dähne SVEN.DAEHNE@TU-BERLIN.DE
Pieter-Jan Kindermans P.KINDERMANS@TU-BERLIN.DE
Technische Universität Berlin, Machine Learning Group
10623 Berlin, Germany

Editor: Alexandre Gramfort

Abstract

In recent years, deep neural networks have revolutionized many application domains of machine learning and are key components of many critical decision or predictive processes. Therefore, it is crucial that domain specialists can understand and analyze actions and predictions, even of the most complex neural network architectures. Despite these arguments neural networks are often treated as black boxes. In the attempt to alleviate this shortcoming many analysis methods were proposed, yet the lack of reference implementations often makes a systematic comparison between the methods a major effort. The presented library ***iNNvestigate*** addresses this by providing a common interface and out-of-the-box implementation for many analysis methods, including the reference implementation for PatternNet and PatternAttribution as well as for LRP-methods. To demonstrate the versatility of *iNNvestigate*, we provide an analysis of image classifications for variety of state-of-the-art neural network architectures.

©2019 Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, Pieter-Jan Kindermans.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-540.html.

Keywords: Artificial neural networks, deep learning, analyzing classifiers, explaining classifiers, computer vision

1. Introduction

In recent years deep neural networks have revolutionized many domains, e.g., image recognition, speech recognition, speech synthesis, and knowledge discovery (Krizhevsky et al., 2012; LeCun et al., 2012; Schmidhuber, 2015; LeCun et al., 2015; Van Den Oord et al., 2016). Due to their ability to naturally learn from structured data and exhibit superior performance, they are increasingly used in practical applications and critical decision processes, such as novel knowledge discovery techniques, autonomous driving or medical image analysis. To fully leverage their potential it is essential that users can *comprehend and analyze* these processes. E.g., in neural architecture (Zoph et al., 2018) or chemical compound searches (Montavon et al., 2013; Schütt et al., 2017) it would be extremely useful to know which properties help a neural network to choose appropriate candidates. Furthermore for some applications understanding the decision process might be a legal requirement.

Despite these arguments neural networks are often treated as black boxes, because their complex internal workings and the basis for their predictions are not fully understood. In the attempt to alleviate this shortcoming several methods were proposed, e.g., Saliency Map (Baehrens et al., 2010; Simonyan et al., 2013), SmoothGrad (Smilkov et al., 2017), IntegratedGradients (Sundararajan et al., 2017), Deconvnet (Zeiler and Fergus, 2014), GuidedBackprop (Springenberg et al., 2015), PatternNet and PatternAttribution (Kindermans et al., 2018), LRP (Bach et al., 2015; Lapuschkin et al., 2016a,b, 2019; Montavon et al., 2018), and DeepTaylor (Montavon et al., 2017). Theoretically it is not clear which method solves the stated problems best, therefore an empirical comparison is required (Samek et al., 2017; Kindermans et al., 2017). In order to evaluate these methods, we present ***iNNvestigate*** which provides a common interface to a variety of analysis methods.

In particular, *iNNvestigate* contributes:

- A common interface for a growing number of analysis methods that is applicable to a broad class of neural networks. With this instantiating a method is as uncomplicated as passing a trained neural network to it and allows for easy qualitative comparisons of methods. For quantitative evaluations of (image) classification task we further provide an implementation of the method "perturbation analysis" (Samek et al., 2017).
- Support of all methods listed above—this includes the first reference implementation for PatternNet and PatternAttribution and an extended implementation for LRP—and an open source repository for further contributions.
- A clean and modular implementation, casting each analysis in terms of layer-wise forward and backward computations. This limits code redundancy, takes advantage of automatic differentiation, and eases future integration of new methods.

iNNvestigate is available at repository: `https://github.com/albermax/innvestigate`. It can be simply installed as Python package and contains documentation for code and applications. To demonstrate the versatility of *iNNvestigate* we provide examples for the analysis of image classifications for a variety of state-of-the-art neural networks.

Terminology: The different methods pose different assumption to tasks and are designed for different objectives, yet they are related to "explaining" or "interpreting" neural networks, see Montavon et al. (2018). We actively refrain from using this terminology in order to prevent misunderstandings between the design choices of the algorithms and the implicit assumption these terms bring along. Therefore we will solely use the neutral term *analyzing* and leave any interpretation to the user.

2. Library

Interface: The main feature is a common interface to several analysis methods. The workflow is as simple as passing a Keras neural network model to instantiate an analyzer object for a desired algorithm. Then, if needed, the analyzer will be fitted to the data and eventually be used to analyze the model's predictions. The corresponding Python code is:

```
import innvestigate
model = create_a_keras_model()
analyzer = innvestigate.create_analyzer(''analyzer_name'', model)
analyzer.fit(X_train) # if needed
analysis = analyzer.analyze(X_test)
```

Implemented methods: At publication time the following algorithms are supported: Gradient Saliency Map, SmoothGrad, IntegratedGradients, Deconvnet, GuidedBackprop, PatternNet and PatternAttribution, DeepTaylor, and LRP including LRP-Z, -Epsilon, -AlphaBeta. In contrast, current related work is limited to gradient- and perturbation-based methods (Kotikalapudi and contributors, 2017; Ancona et al., 2018) or focuses on a single algorithm (E.g., Lundberg and Lee, 2017; Ribeiro et al., 2016). For an overview see Alber (2019). We intend to extend this selection and invite the community to contribute implementations as new methods emerge.

Documentation: The library's documentation contains several introductory scripts and example applications. We demonstrate how the analyses can be applied to the following state-of-the-art models: VGG16 (Simonyan and Zisserman, 2014), InceptionV3 (Szegedy et al., 2016), ResNet50 (He et al., 2016), InceptionResNetV2 (Szegedy et al., 2017), DenseNet (Huang et al., 2017), NASNet (Zoph et al., 2018). Figure 1 shows the result for a subset.

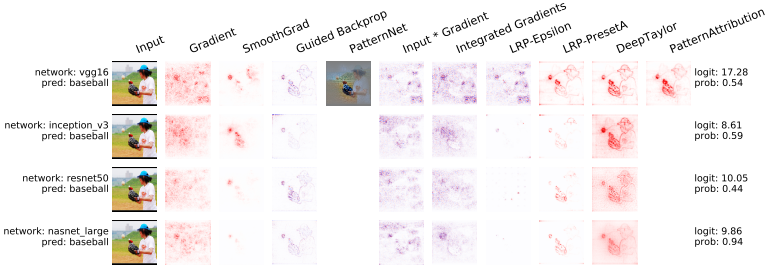


Figure 1: Result of methods applied to various neural networks (blank, if a method does not support a network's architecture).

2.1. Details

Modular implementation: All of the methods have in common that they perform a back-propagation from the model outputs to the inputs. The core of *iNNvestigate* is a set of base classes and functions that is designed to allow for rapid and easy development of such algorithms. The developer only needs to implement specific changes to the base algorithm and the library will take care of the complex and error-prone handling of the propagation along the graph structure. Further details can be found in the repositories documentation.

Another advantage of the modular design is that one can extend any analyzer with a given set of wrappers. One application of this is the smoothing of the analysis results by adding Gaussian noise to the copies of the input and averaging the outcome. E.g., SmoothGrad is realized in this way by combining a smoothing wrapper with a gradient analyzer.

Training: PatternNet and PatternAttribution (Kindermans et al., 2018) are two novel approaches that condition their analysis on the data distribution. This is done by identifying the signal and noise direction for each neuron of a neural network. Our software scales favorably, e.g., one can train required patterns for the methods on large data sets like Imagenet (Deng et al., 2009) in less than an hour using one GPU. We present the first reference implementation of these methods.

Quantitative evaluation: Often analysis methods for neural networks are compared by qualitative (visual) inspection of the result. This is can lead to subjective evaluations and one approach to create a more objective and quantitative comparison of analysis algorithms is the method "perturbation analysis" (Samek et al., 2017, also known as "PixelFlipping"). The intuition behind this method is that perturbing regions which are recognized as important for the classification task by the analyzing method, will impact the classification most. This allows to assess which analysis method best identifies regions that matter for a specific task and neural network. *iNNvestigate* contains an implementation of this method.

Installation & license: *iNNvestigate* is published as open-source software under the BSD-2-license and can be downloaded from: `https://github.com/albermax/innvestigate`. It is built as a Python 2 or 3 application on top of the popular and established Keras (Chollet et al., 2015) framework. This allows to use the library on various platforms and devices like CPUs and GPUs. At the time of publication only the TensorFlow (Abadi et al., 2016) Keras-backend is supported. The library can be simply installed as Python package.

3. Conclusion

We have presented *iNNvestigate*, a library that makes it easier to analyze neural networks' predictions and to compare different analysis methods. This is done by providing a common interface and implementations for many analysis methods as well as making tools for training and comparing methods available. In particular it contains reference implementations for many methods (PatternNet, PatternAttribution, LRP) and example application for a large number of state-of-the-art applications. We expect that this library will support the field of analyzing machine learning and facilitate research using neural networks in domains such as drug design or medical image analysis.

Acknowledgments

Correspondence to MA, SL, KRM, WS and PJK. This work was supported by the Federal Ministry of Education and Research (BMBF) for the Berlin Big Data Center BBDC (01IS14013A). Additional support was provided by the BK21 program funded by Korean National Research Foundation grant (No. 2012-005741) and the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (no. 2017-0-00451, No. 2017-0-01779).

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation*, 16:265–283, 2016.

Maximilian Alber. Software and application patterns for explanation methods. In *Interpretable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 2015.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831, 2010.

François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, 2017.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un)reliability of saliency methods. *Neural Information Processing Systems 2017 - Interpreting, Explaining and Visualizing Deep Learning - Now what? workshop*, 2017.

Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: PatternNet and PatternAttribution. In *International Conference on Learning Representations*, 2018.

Raghavendra Kotikalapudi and contributors. keras-vis. `https://github.com/raghakot/keras-vis`, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012.

Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Analyzing classifiers: Fisher vectors and deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2912–2920, 2016a.

Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. The layer-wise relevance propagation toolbox for artificial neural networks. *Journal of Machine Learning Research*, 17:3938–3942, 2016b.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communcations*, 10:ID: 1096, 2019.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. 2012.

Yann A LeCun, Yoshua Bengio, and Geoffrey E Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774, 2017.

Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and Anatole Von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):ID: 095003, 2013.

Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2017.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61: 85–117, 2015.

Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8:ID: 13890, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: Removing noise by adding noise. *International Conference on Machine Learning 2017 - Workshop on Visualization for Deep Learning*, 2017.

Jost T Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations - Workshop track*, 2015.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 4278–4284, 2017.

Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the 2014 European Conference on Computer Vision*, pages 818–833, 2014.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018.

Journal of Machine Learning Research 20 (2019) 1-34 Submitted 8/18; Published 6/19

Simultaneous Private Learning of Multiple Concepts*

Mark Bun MBUN@BU.EDU
Department of Computer Science
Boston University

Kobbi Nissim KOBBI.NISSIM@GEORGETOWN.EDU
Department of Computer Science
Georgetown University

Uri Stemmer U@URI.CO.IL
Department of Computer Science
Ben-Gurion University

Editor: Mehryar Mohri

Abstract

We investigate the *direct-sum* problem in the context of differentially private PAC learning: What is the sample complexity of solving k learning tasks *simultaneously* under differential privacy, and how does this cost compare to that of solving k learning tasks without privacy? In our setting, an individual example consists of a domain element x labeled by k unknown concepts $(c_1, \ldots, c_k)$. The goal of a *multi*-learner is to output k hypotheses $(h_1, \ldots, h_k)$ that generalize the input examples.

Without concern for privacy, the sample complexity needed to simultaneously learn k concepts is essentially the same as needed for learning a single concept. Under differential privacy, the basic strategy of learning each hypothesis independently yields sample complexity that grows polynomially with k. For some concept classes, we give multi-learners that require fewer samples than the basic strategy. Unfortunately, however, we also give lower bounds showing that even for very simple concept classes, the sample cost of private multi-learning must grow polynomially in k.

Keywords: Differential privacy, PAC learning, Agnostic learning, Direct-sum

1. Introduction

The work on *differential privacy* (Dwork et al., 2006b) is aimed at providing useful analyses on privacy-sensitive data while providing strong individual-level privacy protection. One family of such analyses that has received a lot of attention is PAC learning (Valiant, 1984). These tasks abstract many of the computations performed over sensitive information (Kasiviswanathan et al., 2011).

We address the *direct-sum* problem – what is the cost of solving multiple instances of a computational task simultaneously as compared to solving each of them separately? – in the context of differentially private PAC learning. In our setting, individual examples are drawn from domain X and labeled by k unknown concepts $(c_1, \ldots, c_k)$ taken from a concept class $C = \{c : X \to \{0, 1\}\}$, i.e., each example is of the form $(x, y_1, \ldots, y_k)$, where $x \in X$ and $y_i = c_i(x)$. The goal of a multi-learner is to output k hypotheses $(h_1, \ldots, h_k)$ that generalize the input examples while preserving the privacy of individuals.

The direct-sum problem has its roots in complexity theory, and is a basic problem for many algorithmic tasks. It also has implications for the practical use of differential privacy. Consider, for instance, a hospital that collects information about its patients and wishes to use this information for medical research. The hospital records for each patient a collection of attributes such as age, sex, and the results of various diagnostic tests (for each patient, these attributes make up a point x in some domain X) and, for each of k diseases, whether the patient suffers from the disease (the k labels $(y_1, \ldots, y_k)$). Based on this collection of data, the hospital researchers wish to learn good predictors for the k diseases. One option for the researchers is to perform each of the learning tasks on a fresh sample of patients, hence enlarging the number of patient examples needed (i.e. the *sample complexity*) by a factor of k, which can be very costly.

Without concern for privacy, the sample complexity that is necessary and sufficient for performing the k learning tasks is actually fully characterized by the VC dimension of the concept class C – it is independent of the number of learning tasks k. In this work, we set out to examine if the situation is similar when the learning is performed with differential privacy. Interestingly, we see that with differential privacy the picture is quite different, and in particular, the required number of examples can grow polynomially in k.

Private learning. A *private learner* is an algorithm that is given a sample of labeled examples $(x, c(x))$ (each representing the information and label pertaining to an individual) and outputs a generalizing hypothesis h that guarantees differential privacy with respect to its examples. The first differentially private learning algorithms were given by Blum et al. (2005) and the notion of *private learning* was put forward and formally researched by Kasiviswanathan et al. (2011). Among other results, the latter work presented a generic construction of differentially private learners with sample complexity $O(\log |C|)$.

In contrast, the sample complexity of (non-private) PAC learning is $\Theta(\mathrm{VC}(C))$, which can be much lower than $\log |C|$ for specific concept classes. This gap led to a line of work examining the sample complexity of private learning, which has revealed a significantly more complex picture than there is for non-private learning. In particular, for *pure* differentially private learners, it is known that the sample complexity of proper learning (where the learner returns a hypothesis h taken from C) is sometimes higher than the sample complexity of improper learners (where h comes from an arbitrary hypothesis class H). The latter is characterized by the *representation dimension* of the concept class C, which is generally higher than the VC dimension (Beimel et al., 2014; Chaudhuri and Hsu, 2011; Beimel et al., 2013a; Feldman and Xiao, 2014). By contrast, a sample complexity gap between proper and improper learners does not exist for non-private learning. In the case of *approximate* differential privacy no such combinatorial characterization is currently known. It is however known that the sample complexity of such learners can be significantly lower than that of pure-differentially private learners and yet higher than the VC dimension of C (Beimel et al., 2013a,b; Feldman and Xiao, 2014; Bun et al., 2015; Alon et al., 2018). Furthermore, there exist (infinite) PAC-learnable concept classes for which no differentially private learner (pure or approximate) exists.

*. A preliminary version of this paper appeared in ITCS 2016.

©2019 Mark Bun and Kobbi Nissim and Uri Stemmer.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-549.html.

Private multi-learning. In this work we examine the sample complexity of private multi-learning. Our work is motivated by the recurring research theme of the direct-sum, as well as by the need to understand whether multi-learning remains feasible under differential privacy, as it is without privacy constraints.

At first glance, private multi-learning appears to be similar to the query release problem, the goal of which is to approximate the average values of a large collection of predicates on a dataset. One surprising result in differential privacy is that it is possible to answer an exponential number of such queries on a dataset (Blum et al., 2013; Roth and Roughgarden, 2010; Hardt and Rothblum, 2010). For example, Blum et al. (2013) showed that given a dataset D and a concept class C, it is possible to generate with differential privacy a dataset $\hat{D}$ such that the average value of c on D approximates the average of c on $\hat{D}$ for every $c \in C$ *simultaneously* (this returned database $\hat{D}$ is called a *sanitized* database, and the algorithm computing $\hat{D}$ is called a *sanitizer*). The sample complexity required, i.e., the size of the database D, to perform this sanitization is only logarithmic in $|C|$. Results of this flavor suggest that we can also learn exponentially many concepts simultaneously. However, we give negative results showing that this is not the case, and that multi-learning can have significantly higher sample complexity than query release.

1.1. Our results

Prior work on privately learning the simple concept classes $\texttt{POINT}_X$ (of functions that evaluate to 1 on exactly one point of their domain X and to 0 otherwise) and $\texttt{THRESH}_X$ (of functions that evaluate to 1 on a prefix of the domain X and to 0 otherwise) has demonstrated a rather complex picture, depending on whether learners are proper or improper, and whether learning is performed with pure or approximate differential privacy (Beimel et al., 2014, 2013a,b; Bun et al., 2015). We analyze the sample complexity of multi-learning of these simple concept classes, as well as general concept classes. We also consider the class $\texttt{PAR}_d$ of parity functions, but in this case we restrict our attention to uniformly selected examples. We examine both proper and improper PAC and agnostic learning under pure and approximate differential privacy. For ease of reference, we include tables with our results in Section 1.3, where we omit the dependency on the privacy and accuracy parameters.

Techniques for private k-learning. Composition theorems for differential privacy show that the sample complexity of learning k concepts simultaneously is at most a factor of k larger than the sample complexity of learning one concept (and may be reduced to $\sqrt{k}$ for approximate differential privacy). Unfortunately, privately learning one concept from a concept class C can sometimes be quite costly, requiring much higher sample complexity than $\mathrm{VC}(C)$ which is needed to learn non-privately. Building on techniques of Beimel et al. (2015), we show that the multiplicative dependence on k can always be reduced to the VC-dimension of C, at the expense of producing a one-time sanitization of the dataset.

Theorem 1 (Informal) *Let C be a concept class for which there is pure differentially private sanitizer for $C^{\oplus} = \{f \oplus g : f, g \in C\}$ with sample complexity m. Then there is a pure differentially private agnostic k-learner for C with sample complexity $O(m + k \cdot \mathrm{VC}(C))$.*

Similarly, if $C^{\oplus}$ has an approximate differentially private sanitizer with sample complexity m, then there is an approximate differentially private agnostic k-learner for C with sample complexity $O(m + \sqrt{k} \cdot \mathrm{VC}(C))$.

The best known general-purpose sanitizers require sample complexity $m = O(\mathrm{VC}(C) \log |X|)$ for pure differential privacy (Blum et al., 2013) and $m = O(\log |C| \sqrt{\log |X|})$ for approximate differential privacy (Hardt and Rothblum, 2010). However, for specific concept classes (such as $\texttt{POINT}_X$ and $\texttt{THRESH}_X$), the sample complexity of sanitization can be much lower.

In the case of approximate differential privacy, the sample complexity of k-learning can be even lower than what is achievable with our generic learner. Using stability-based arguments, we show that point functions and parities under the uniform distribution can be PAC k-learned with sample complexity $O(\mathrm{VC}(C))$ – independent of the number of concepts k (see Theorems 40 and 39).

Lower bounds. In light of the above results, one might hope to be able to reduce the dependence on k further, or to eliminate it entirely (as is possible in the case of non-private learning). We show that this is not possible, even for the simplest of concept classes. In the case of pure differential privacy, a packing argument (Feldman et al., 2009; Hardt and Talwar, 2010; Beimel et al., 2014) shows that any non-trivial concept class requires sample complexity $\Omega(k)$ to privately k-learn (Theorem 52). For approximate differential privacy, we use fingerprinting codes (Boneh and Shaw, 1998; Bun et al., 2014) to show that unlike points and parities, threshold functions require sample complexity $\tilde{\Omega}(k^{1/3})$ to PAC learn privately (Corollary 46). Moreover, any non-trivial concept class requires sample complexity $\tilde{\Omega}(\sqrt{k})$ to privately learn in the agnostic model (Theorem 47). In the case of point functions, this matches the upper bound achievable by our generic learner.

We highlight a few of the main takeaways from our results:

A complex answer to the direct sum question. Our upper bounds show that solving k learning problems simultaneously can require substantially lower sample complexity than solving the problems individually. On the other hand, our lower bounds show that a significant dependence on k is generally necessary.

Separation between private PAC and private agnostic learning. Non-privately, the sample complexities of PAC and agnostic learning are of the same order (differing only in the dependency in the accuracy parameters). Beimel et al. (2015) showed that this is also the case with differentially private learning (of one concept). Our results on learning point functions show that private PAC and agnostic *multi-learning* can be substantially different (even for learning up to constant error). In the case of approximate differential privacy, $O(1)$ samples suffice to PAC-learn multiple point functions. However, $\tilde{\Omega}(\sqrt{k})$ samples are needed to learn k points agnostically.

Separation between improper learning with approximate differential privacy and non-private learning. Bun et al. (2015) and Alon et al. (2018) showed that the sample complexity of approximate-private learning can be asymptotically larger than that of non-private learning, but this separation is very mild. Specifically, they showed that learning one threshold function over a domain X requires sample complexity at least $\Omega(\log^* |X|)$ with approximate differential privacy, as opposed to $O(1)$ without privacy. An interesting open

question is whether there are cases in which a stronger separation exists, or is it the case that the extra costs in the sample complexity of approximate private learning can always be reduced to $O(\log^* |X|)$. While we do not address this question directly, we exhibit a strong separation for multi-learning. In particular, learning k thresholds with approximate differential privacy requires $\tilde{\Omega}(k^{1/3})$ samples, while $O(1)$ samples suffices non-privately.

1.2. Related work

Differential privacy was defined in (Dwork et al., 2006b) and the relaxation to approximate differential privacy is from (Dwork et al., 2006a). Most related to our work is the work on private learning and its sample complexity (Blum et al., 2005; Kasiviswanathan et al., 2011; Chaudhuri and Hsu, 2011; Dwork et al., 2010; Beimel et al., 2013a,b, 2014, 2015, 2019; Kaplan et al., 2019; Feldman and Xiao, 2014; Bun et al., 2015; Alon et al., 2018) and the early work on sanitization (Blum et al., 2013). That many "natural" learning tasks can be performed privately was shown in the early work of Blum et al. (2005) and Kasiviswanathan et al. (2011). A characterization for the sample complexity of *pure-private* learners was given by Beimel et al. (2013a), in terms of a new combinatorial measure – the *Representation Dimension*, that is, given a class C, the number of samples needed and sufficient for privately learning C is $\Theta(\mathrm{RepDim}(C))$. Building on (Beimel et al., 2013a), Feldman and Xiao (2014) showed an equivalence between the representation dimension of a concept C and the randomized one-way communication complexity of the evaluation problem for concepts from C.

The problem of learning multiple concepts simultaneously (without privacy) has been considered before. Motivated by the problem of bridging computational learning and reasoning, Valiant (2006) also observed that (without privacy) multiple concepts can be learned from a common dataset in a data efficient manner.

1.3. Tables of results

The following tables summarize the results of this work. In the tables below C is a class of concepts (i.e., predicates) defined over domain X. Sample complexity upper and lower bounds is given in terms of $|C|$ and $|X|$. Note that for $\mathtt{POINT}_X$, $\mathtt{THRESH}_X$, and $\mathtt{PAR}_d$ we have $|C| = \Theta(|X|)$.

Where not explicitly noted, upper bounds hold for the setting of agnostic learning and lower bounds are for the (potentially easier) setting of PAC learning. Similarly, where not explicitly noted, upper bounds are for proper learning and lower bounds are for the (less restrictive) setting of improper learning. For simplicity, these tables hide constant and logarithmic factors, as well as dependencies on the learning and privacy parameters.

Multi-learning with pure differential privacy.

Upper bounds:

C	PAC learning and agnostic learning proper	improper	References
$\mathtt{POINT}_X$	$k + \log \lvert C\rvert$	k	Thm. 31, Cor. 33
$\mathtt{THRESH}_X$	$k + \log \lvert C\rvert$		Thm. 31
General	$\min\{k \log \lvert C\rvert, k\,\mathrm{VC}(C) + \log \lvert X\rvert\,\mathrm{VC}(C)\}$		Thm. 31
$\mathtt{PAR}_d$ (uniform)	$k \log \lvert C\rvert$		Thm. 31

Lower bounds:

C	PAC learning and agnostic learning proper	improper	References
$\mathtt{POINT}_X$	$k + \log \lvert C\rvert$	k	Thm. 52, (Beimel et al., 2014)
$\mathtt{THRESH}_X$	$k + \log \lvert C\rvert$		Thm. 52, (Beimel et al., 2014) (Feldman and Xiao, 2014)
$\mathtt{PAR}_d$ (uniform)	$k \log \lvert C\rvert$		Thm. 55

Multi-learning with approximate differential privacy.

Upper bounds:

C	PAC learning (proper and improper)	Agnostic learning (proper and improper)
$\mathtt{POINT}_X$	1 (Thm. 40)	$\sqrt{k}$ (Cor. 37)
$\mathtt{THRESH}_X$	$2^{\log^* \lvert X\rvert} + \sqrt{k}$ (Cor. 38)	
General C	$\min\{\sqrt{k} \log \lvert C\rvert, \sqrt{k}\,\mathrm{VC}(C) + \log \lvert X\rvert\,\mathrm{VC}(C), \sqrt{k}\,\mathrm{VC}(C) + \sqrt{\log \lvert X\rvert} \log \lvert C\rvert\}$ (Thm. 31)	
$\mathtt{PAR}_d$ (uniform)	$\log \lvert C\rvert$ (Thm. 39)	$\sqrt{k} \log \lvert C\rvert$ (Thm. 31)

Lower bounds:

C	PAC learning (proper and improper)	Agnostic learning (proper and improper)	References
$\mathtt{POINT}_X$	1	$\sqrt{k}$	Cor. 50
$\mathtt{THRESH}_X$	$\log^* \lvert X\rvert + k^{1/3}$	$\log^* \lvert X\rvert + \sqrt{k}$	Cor. 46, Cor. 50, (Bun et al., 2015), (Alon et al., 2018)
$\mathtt{PAR}_d$ (uniform)	$\log \lvert C\rvert$	$\sqrt{k} + \log \lvert C\rvert$	Cor. 50

2. Preliminaries

We recall and extend standard definitions from learning theory and differential privacy.

2.1. Multi-learners

In the following X is some arbitrary domain. A concept (similarly, hypothesis) over domain X is a predicate defined over X. A concept class (similarly, hypothesis class) is a set of concepts.

Definition 2 (Population Error) *Let $\mathcal{P} \in \Delta(X \times \{0,1\})$ be a probability distribution over $X \times \{0,1\}$. The* population error *of a hypothesis $h : X \to \{0,1\}$ w.r.t. $\mathcal{P}$ is defined as* $\mathrm{error}_{\mathcal{P}}(h) = \Pr_{(x,y)\sim\mathcal{P}}[h(x) \neq y]$.

Let $\mathcal{D} \in \Delta(X)$ be a probability distribution over X and let $c : x \to \{0,1\}$ be a concept. The population error *of hypothesis $h : X \to \{0,1\}$ w.r.t. c and $\mathcal{D}$ is defined as* $\mathrm{error}_{\mathcal{D}}(c,h) = \Pr_{x\sim\mathcal{D}}[h(x) \neq c(x)]$. *If $\mathrm{error}_{\mathcal{D}}(c,h) \leq \alpha$ we say that h is α-good for c and $\mathcal{D}$.*

Definition 3 (Multi-labeled database) *A k-*labeled *database over a domain X is a database $S \in (X \times \{0,1\}^k)^*$. That is, S contains $|S|$ elements from X, each concatenated with k binary labels.*

Let $\mathcal{A} : \left(X \times \{0,1\}^k\right)^n \to \left(2^X\right)^k$ be an algorithm that operates on a k-labeled database and returns k hypotheses. Let C be a concept class over a domain X and let H be a hypothesis class over X. We now give a generalization of the notion of PAC learning (Valiant, 1984) to multi-labeled databases (the standard PAC definition is obtained by setting $k = 1$):

Definition 4 (PAC Multi-Learner) *Algorithm $\mathcal{A}$ is an (α,β)-PAC k-learner for concept class C using hypothesis class H with sample complexity n if for every distribution $\mathcal{D}$ over X and for every tuple $(c_1,\ldots,c_k)$ from C, given a k-labeled database as an input $S = ((x_i, c_1(x_i),\ldots,c_k(x_i)))_{i=1}^n$ where each x_i is drawn i.i.d. from $\mathcal{D}$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1,\ldots,h_k)$ from H satisfying*

$$\Pr\left[\max_{1\leq j\leq k}\left(\mathrm{error}_{\mathcal{D}}(c_j,h_j)\right) > \alpha\right] \leq \beta.$$

The probability is taken over the random choice of the examples in S according to $\mathcal{D}$ and the coin tosses of the learner $\mathcal{A}$. If $H \subseteq C$ then A is called a proper *learner; otherwise, it is called an* improper *learner.*

Definition 5 (Agnostic PAC Multi-Learner) *Algorithm $\mathcal{A}$ is an (α,β)-PAC agnostic k-learner for C using hypothesis class H and sample complexity n if for every distribution $\mathcal{P}$ over $X \times \{0,1\}^k$, given a k-labeled database $S = ((x_i, y_{1,i},\ldots,y_{k,i}))_{i=1}^n$ where each k-labeled sample $(x_i, y_{1,i}\ldots,y_{k,i})$ is drawn i.i.d. from $\mathcal{P}$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1,\ldots,h_k)$ from H satisfying*

$$\Pr\left[\max_{1\leq j\leq k}\left(\mathrm{error}_{\mathcal{P}_j}(h_j) - \min_{c\in C}\left(\mathrm{error}_{\mathcal{P}_j}(c)\right)\right) > \alpha\right] \leq \beta,$$

where $\mathcal{P}_j$ is the marginal distribution of $\mathcal{P}$ on the examples and the jth label. The probability is taken over the random choice of the examples in S according to $\mathcal{P}$ and the coin tosses of the learner $\mathcal{A}$. If $H \subseteq C$ then A is called a proper *learner; otherwise, it is called an* improper *learner.*

2.2. The Sample Complexity of Multi-Learning

Without privacy considerations, the sample complexities of PAC and agnostic learning are essentially characterized by a combinatorial quantity called the *Vapnik-Chervonenkis (VC) dimension*. We state these characterizations in the context of multi-learning.

2.2.1. The Vapnik-Chervonenkis Dimension

Definition 6 *Fix a concept class C over domain X. A set $\{x_1,\ldots,x_d\} \in X$ is* shattered *by C if for every labeling $b \in \{0,1\}^d$, there exists $c \in C$ such that $b_1 = c(x_1),\ldots,b_d = c(x_d)$. The* Vapnik-Chervonenkis (VC) dimension *of C, denoted $\mathrm{VC}(C)$, is the size of the largest set which is shattered by C.*

The Vapnik-Chervonenkis (VC) dimension is an important combinatorial measure of a concept class. Classical results in statistical learning theory show that the population error of a hypothesis h and its empirical error (observed on a large enough sample) are similar.

Definition 7 (Empirical Error) *Let $S = ((x_i, y_i))_{i=1}^n \in (X \times \{0,1\})^n$ be a labeled sample from X. The* empirical error *of a hypothesis $h : X \to \{0,1\}$ w.r.t. S is defined as* $\mathrm{error}_S(h) = \frac{1}{n}|\{i : h(x_i) \neq y_i\}|$.

Let $D \in X^n$ be a (unlabeled) sample from X and let $c : x \to \{0,1\}$ be a concept. The empirical error *of hypothesis $h : X \to \{0,1\}$ w.r.t. c and D is defined as* $\mathrm{error}_D(c,h) = \frac{1}{n}|\{i : h(x_i) \neq c(x_i)]$.

Theorem 8 (VC-Dimension Generalization Bound, e.g. (Blumer et al., 1989)) *Let $\mathcal{D}$ and C be, respectively, a distribution and a concept class over a domain X, and let $c \in C$. For a sample $S = ((x_i, c(x_i)))_{i=1}^n$ where $n \geq \frac{64}{\alpha}(\mathrm{VC}(C)\ln(\frac{64}{\alpha}) + \ln(\frac{8}{\beta}))$ and the x_i are drawn i.i.d. from $\mathcal{D}$, it holds that*

$$\Pr\left[\exists h \in C \text{ s.t. } \mathrm{error}_{\mathcal{D}}(h,c) > \alpha \ \wedge\ \mathrm{error}_S(h) \leq \frac{\alpha}{2}\right] \leq \beta.$$

This generalization argument extends to the setting of *agnostic learning*, where a hypothesis with small empirical error might not exist.

Theorem 9 (VC-Dimension Agnostic Generalization Bound, e.g. (Anthony and Bartlett, 2009; Anthony and Shawe-Taylor, 1993)) *Let H be a concept class over a domain X, and let $\mathcal{P}$ be a distribution over $X \times \{0,1\}$. For a sample $S = ((x_i, y_i))_{i=1}^n$ containing $n \geq \frac{64}{\alpha^2}(\mathrm{VC}(H)\ln(\frac{6}{\alpha}) + \ln(\frac{8}{\beta}))$ i.i.d. elements from $\mathcal{P}$, it holds that*

$$\Pr\left[\exists h \in H \text{ s.t. } \left|\mathrm{error}_{\mathcal{P}}(h) - \mathrm{error}_S(h)\right| > \alpha\right] \leq \beta.$$

Using theorems 8 and 9, an upper bound of $O(\mathrm{VC}(C))$ on the sample complexity of learning a concept class C follows by reduction to the *empirical learning* problem. The goal of empirical learning is similar to that of PAC learning, except accuracy is measured only with respect to a fixed input database. Theorems 8 and 9 state that when an empirical learner is run on sufficiently many samples, it is also accurate with respect to a distribution on inputs.

Definition 10 (Empirical Learner) *Algorithm $\mathcal{A}$ is an (α,β)-*accurate empirical *k-*learner *for a concept class C using hypothesis class H with sample complexity n if for every collection of concepts $(c_1,\ldots,c_k)$ from C and database $S = ((x_i, c_1(x_i),\ldots,c_k(x_i)))_{i=1}^n \in (X \times \{0,1\}^k)^n$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1,\ldots,h_k)$ from H satisfying*

$$\Pr\left[\max_{1\leq j\leq k}\left(\mathrm{error}_{S|_j}(h_j)\right) > \alpha\right] \leq \beta,$$

where $S|_j = ((x_i, c_j(x_i)))_{i=1}^n$. *The probability is taken over the coin tosses of* $\mathcal{A}$.

Definition 11 (Agnostic Empirical Learner) *Algorithm* $\mathcal{A}$ *is an* agnostic (α, β)-accurate empirical k-learner *for a concept class* C *using hypothesis class* H *with sample complexity* n *if for every database* $S = ((x_i, y_{1,i}, \ldots, y_{k,i}))_{i=1}^n \in (X \times \{0,1\}^k)^n$, *algorithm* $\mathcal{A}$ *outputs* k *hypotheses* $(h_1, \ldots, h_k)$ *from* H *satisfying*

$$\Pr\left[\max_{1\le j\le k}\left(\mathrm{error}_{S|_j}(h_j) - \min_{c\in C}\left(\mathrm{error}_{S|_j}(c)\right)\right) > \alpha\right] \le \beta,$$

where $S|_j = ((x_i, y_{j,i}))_{i=1}^n$. *The probability is taken over the coin tosses of* $\mathcal{A}$.

Theorem 12 *Let* $\mathcal{A}$ *be an* (α, β)*-accurate empirical* k*-learner for a concept class* C *(resp. agnostic empirical* k*-learner) using hypothesis class* H. *Then* $\mathcal{A}$ *is also a* $(2\alpha, \beta + \beta')$*-accurate PAC learner for* C *when given at least* $\max\{n, \frac{32}{\alpha}(\mathrm{VC}(H \oplus C)\log(32/\alpha) + \log(8/\beta'))\}$ *samples (resp.* $\max\{n, \frac{64}{\alpha^2}(\mathrm{VC}(H)\log(6/\alpha) + \log(8k/\beta'))$ *samples). Here,* $H \oplus C = \{h \oplus c : h \in H, c \in C\}$.

Proof We begin with the non-agnostic case. Let $\mathcal{A}$ be an (α, β)-accurate empirical k-learner for C. Let $\mathcal{D}$ be a distribution over the example space X. Let S be a random i.i.d. sample of size m from $\mathcal{D}$. The generalization bound for PAC learning (Theorem 8) states that if $m \ge \frac{32}{\alpha}(d\log(32/\alpha) + \log(8/\beta'))\}$, then

$$\Pr[\exists c \in C, h \in H : \mathrm{error}_S(c, h) \le \alpha \wedge \mathrm{error}_{\mathcal{D}}(c, h) > 2\alpha] \le \beta',$$

where $d = \mathrm{VC}(H \oplus C)$. The result follows by a union bound over the failure probability of $\mathcal{A}$ and the failure of generalization.

Now we turn to the agnostic case. Let $\mathcal{A}$ be an agnostic (α, β)-accurate empirical k-learner for C. Fix an index $j \in [k]$, and let $\mathcal{P}_j$ be a distribution over $X \times \{0,1\}$. Let S be a random i.i.d. sample of size m from $\mathcal{P}_j$. Then generalization for agnostic learning (Theorem 9) yields

$$\Pr[\exists h \in H : |\mathrm{error}_S(h) - \mathrm{error}_{\mathcal{P}_j}(h)| > \alpha] \le \frac{\beta'}{k}$$

for $m \ge \frac{64}{\alpha^2}(\mathrm{VC}(H)\log(6/\alpha) + \log(8k/\beta'))\}$. The result follows by a union bound over the failure probability of $\mathcal{A}$ and the failure of generalization for each of the indices $j = 1, \ldots, k$. ■

Applying the above theorem in the special case where $\mathcal{A}$ finds the concept $c \in C$ that minimizes the empirical error on its given sample, we obtain the following sample complexity upper bound for proper multi-learning.

Corollary 13 *Let* C *be a concept class with VC dimension* d. *There exists an* (α, β)*-accurate proper PAC* k*-learner for* C *using* $O(\frac{1}{\alpha}(d\log(1/\alpha) + \log(1/\beta))$ *samples. Moreover, there exists an* (α, β)*-accurate proper agnostic PAC* k*-learner for* C *using* $O(\frac{1}{\alpha^2}(d\log(1/\alpha) + \log(k/\beta))$ *samples.*

Proof For the non-agnostic case, we simply let $\mathcal{A}$ be the $(0, 0)$-accurate empirical learner that outputs any vector of hypotheses that is consistent with its given examples (one is guaranteed to exist, since the target concept satisfies this condition). The claim follows from Theorem 12 noting that $\mathrm{VC}(C \oplus C) = O(\mathrm{VC}(C))$.

For the agnostic case, consider the algorithm $\mathcal{A}$ that on input S outputs hypotheses $(h_1, \ldots, h_k)$ that minimize the quantities $\mathrm{error}_{S_j}(h_j)$. Applying the agnostic generalization bound (Anthony and Bartlett, 2009), this is an $(\alpha/2, \beta/2)$-accurate agnostic empirical learner given $O(\frac{1}{\alpha^2}(d\log(1/\alpha) + \log(k/\beta))$ samples. The claim then follows from Theorem 12. ■

It is known that even for $k = 1$, the sample complexities of PAC and agnostic learning are at least $\Omega(\mathrm{VC}(C)/\alpha)$ and $\Omega(\mathrm{VC}(C)/\alpha^2)$, respectively. Therefore, the above sample complexity upper bound is tight up to logarithmic factors.

We define a few specific concept classes which will play an important role in this work.

$\mathtt{POINT}_X$**:** Let X be any domain. The class of *point functions* is the set of all concepts that evaluate to 1 on exactly one element of X, i.e. $\mathtt{POINT}_X = \{c_x : x \in X\}$ where $c_x(y) = 1$ iff $y = x$. The VC-dimension of $\mathtt{POINT}_X$ is 1 for any X.

$\mathtt{THRESH}_X$**:** Let X be any totally ordered domain. The class of *threshold functions* takes the form $\mathtt{THRESH}_X = \{c_x : x \in X\}$ where $c_x(y) = 1$ iff $y \le x$. The VC-dimension of $\mathtt{THRESH}_X$ is 1 for any X.

$\mathtt{PAR}_d$**:** Let $X = \{0,1\}^d$. The class of *parity functions* on X is given by $\mathtt{PAR}_d = \{c_x : x \in X\}$ where $c_x(y) = \langle x, y\rangle \pmod 2$. The VC-dimension of $\mathtt{PAR}_d$ is d.

In this work, we focus our study of the concept class $\mathtt{PAR}_d$ on the problem of learning parities under the uniform distribution. The PAC and agnostic learning problems are defined as before, except we only require a learner to be accurate when the marginal distribution on examples is the uniform distribution U_d over $\{0,1\}^d$.

Definition 14 (PAC Learning $\mathtt{PAR}_d$ under Uniform) *Algorithm* $\mathcal{A}$ *is an* (α, β)*-PAC* k*-learner for* $\mathtt{PAR}_d$ *using hypothesis class* H *and sample complexity* n *if for every fixed* $(c_1, \ldots, c_k)$ *from* C, *given a* k*-labeled database as an input* $S = ((x_i, c_1(x_i), \ldots, c_k(x_i)))_{i=1}^n$ *where each* x_i *is drawn i.i.d. from* U_d, *algorithm* $\mathcal{A}$ *outputs* k *hypotheses* $(h_1, \ldots, h_k)$ *from* H *satisfying*

$$\Pr\left[\max_{1\le j\le k}(\mathrm{error}_{U_d}(c_j, h_j)) > \alpha\right] \le \beta.$$

Definition 15 (Agnostically Learning $\mathtt{PAR}_d$ under Uniform) *Algorithm* $\mathcal{A}$ *is an* (α, β)*-PAC agnostic* k*-learner for* $\mathtt{PAR}_d$ *using hypothesis class* H *and sample complexity* n *if for every distribution* $\mathcal{P}$ *over* $\{0,1\}^d \times \{0,1\}^k$, *with marginal distribution* U_d *over the data universe* $\{0,1\}^d$, *given a* k*-labeled database* $S = ((x_i, y_{1,i}, \ldots, y_{k,i}))_{i=1}^n$ *where each* k*-labeled sample* $(x_i, y_{1,i} \ldots, y_{k,i})$ *is drawn i.i.d. from* $\mathcal{P}$, *algorithm* $\mathcal{A}$ *outputs* k *hypotheses* $(h_1, \ldots, h_k)$ *from* H *satisfying*

$$\Pr\left[\max_{1\le j\le k}\left(\mathrm{error}_{\mathcal{P}_j}(h_j) - \min_{c\in C}\left(\mathrm{error}_{\mathcal{P}_j}(c)\right)\right) > \alpha\right] \le \beta,$$

where $\mathcal{P}_j$ *is the marginal distribution of* $\mathcal{P}$ *on the examples and the* j^{th} *label.*

2.3. Differential privacy

Two k-*labeled* databases $S, S' \in (X \times \{0,1\}^k)^n$ are called *neighboring* if they differ on a single (multi-labeled) entry, i.e., $|\{i : (x_i, y_{1,i}, \ldots, y_{k,i}) \neq (x'_i, y'_{1,i}, \ldots, y'_{k,i})\}| = 1$.

Definition 16 (Differential Privacy (Dwork et al., 2006b)) *Let $\mathcal{A} : \left(X \times \{0,1\}^k\right)^n \to \left(2^X\right)^k$ be an algorithm that operates on a k-labeled database and returns k hypotheses. Let $\epsilon, \delta \geq 0$. Algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all neighboring S, S' and for all $T \subseteq \left(2^X\right)^k$,*

$$\Pr[\mathcal{A}(S) \in T] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(S') \in T] + \delta,$$

where the probability is taken over the coin tosses of the algorithm $\mathcal{A}$. When $\delta = 0$ we say that $\mathcal{A}$ satisfies pure *differential privacy, otherwise (i.e., if $\delta > 0$) we say that $\mathcal{A}$ satisfies* approximate *differential privacy.*

Our learning algorithms are designed via repeated applications of differentially private algorithms on a database. Composition theorems for differential privacy show that the price of privacy for multiple (adaptively chosen) interactions degrades gracefully.

Theorem 17 (Composition of Differential Privacy (Dwork et al., 2006a; Dwork and Lei, 2009; Dwork et al., 2010)) *Let $0 < \epsilon, \delta' < 1$ and $\delta \in [0,1]$. Suppose an algorithm $\mathcal{A}$ accesses its input database S only through m adaptively chosen executions of (ϵ, δ)-differentially private algorithms. Then $\mathcal{A}$ is*

1. *$(m\epsilon, m\delta)$-differentially private, and*
2. *$(\epsilon', m\delta + \delta')$-differentially private for $\epsilon = \sqrt{2m \ln(1/\delta')} \cdot \epsilon + 2m\epsilon^2$.*

2.4. Differentially Private Tools

The most basic constructions of differentially private algorithms are via the Laplace Mechanism as follows.

Definition 18 (The Laplace Distribution) *A random variable has probability distribution* $\mathrm{Lap}(b)$ *if its probability density function is $f(x) = \frac{1}{2b}\exp(-\frac{|x|}{b})$, where $x \in \mathbb{R}$.*

Definition 19 (Sensitivity) *The* sensitivity *of a function $f : X^n \to \mathbb{R}^d$ is the smallest s such that for every neighboring $D, D' \in X^n$, we have $\|f(D) - f(D')\|_1 \leq s$. We use the term* sensitivity-s function *to mean a function of sensitivity $\leq s$.*

Theorem 20 (The Laplace Mechanism (Dwork et al., 2006b)) *Let $f : X^n \to \mathbb{R}^d$ be a function of sensitivity s. The mechanism A that on input $D \in X^n$ adds independently generated noise with distribution $\mathrm{Lap}(\frac{s}{\epsilon})$ to each of the d output terms of $f(D)$ preserves ϵ-differential privacy.*

We next describe the exponential mechanism of McSherry and Talwar (2007), which is an important building block in many differentially private constructions. A quality function $q : X^* \times \mathcal{F} \to \mathbb{N}$ defines an *optimization problem* over the domain X and a finite solution set

$\mathcal{F}$: Given a database $S \in X^*$, choose $f \in \mathcal{F}$ that (approximately) maximizes $q(S, f)$. The exponential mechanism solves such an optimization problem by sampling a random $f \in \mathcal{F}$ with probability $\propto \exp(\epsilon \cdot q(S, f)/2\Delta q)$, where Δq is the sensitivity of the quality function q, defined as the maximum over all $f \in \mathcal{F}$ of the sensitivity of the function $q(\cdot, f)$.

Proposition 21 (Properties of the Exponential Mechanism (McSherry and Talwar, 2007))

1. *The exponential mechanism is $(\epsilon, 0)$-differentially private.*
2. *Let q be a quality function with sensitivity at most 1. Fix a database $S \in X^n$ and let $\mathrm{OPT} = \max_{f \in \mathcal{F}}\{q(S, f)\}$. Let $t > 0$. Then exponential mechanism outputs a solution f with $q(S, f) \leq \mathrm{OPT} - tn$ with probability at most $|\mathcal{F}| \cdot \exp(-\epsilon t n/2)$.*

We next describe algorithm $\mathcal{A}_{\mathrm{dist}}$ of Smith and Thakurta (2013). Our discussion follows the treatment of (Beimel et al., 2013b). As before, a quality function $q : X^* \times \mathcal{F} \to \mathbb{N}$ defines an optimization problem. Algorithm $\mathcal{A}_{\mathrm{dist}}$ privately identifies the *exact* maximizer as long as it is sufficiently stable.

Algorithm 1 $\mathcal{A}_{\mathrm{dist}}$

Input: Privacy parameters ϵ, δ, database $S \in X^*$, sensitivity-1 quality function q

1. Let $f_1, f_2 \in \mathcal{F}$ be the highest scoring and second-highest scoring solutions to $q(S, \cdot)$, respectively.
2. Let $\mathrm{gap} = q(S, f_1) - q(S, f_2)$, and $\widehat{\mathrm{gap}} = \mathrm{gap} + \mathrm{Lap}(1/\epsilon)$.
3. If $\widehat{\mathrm{gap}} < \frac{1}{\epsilon}\log\frac{1}{\delta}$, output $\perp$. Otherwise, output f_1.

Proposition 22 (Properties of $\mathcal{A}_{\mathrm{dist}}$ (Smith and Thakurta, 2013))

1. *Algorithm $\mathcal{A}_{\mathrm{dist}}$ is (ϵ, δ)-differentially private.*
2. *When run on a database S with $\mathrm{gap} > \frac{1}{\epsilon}\log\frac{1}{\delta\beta}$, Algorithm $\mathcal{A}_{\mathrm{dist}}$ outputs the highest scoring solution f_1 with probability at least $1 - \beta$.*

2.5. Differentially Private Sanitization

Data sanitization is a fundamental task in differential privacy. Given a database $D = (x_1, \ldots, x_n) \in X^n$, the goal of a sanitizer is to privately produce a synthetic database $\hat{D} \in X^m$ that captures the statistical properties of D. We are primarily interested in sanitization for boolean-valued functions (equivalently referred to as *counting queries*). Given a function $c : X \to \{0,1\}$ and a database $D = (x_1, \ldots, x_n)$, we write $c(D) = \frac{1}{n}\sum_{i=1}^n c(x_i)$.

Definition 23 (Sanitization) *An algorithm $\mathcal{A} : X^n \to X^m$ is an (α, β)-accurate sanitizer for a concept class C if for every $D \in X^n$, the algorithm $\mathcal{A}$ produces a database $\hat{D} \in X^m$ such that*

$$\Pr[\exists c \in C : |c(D) - c(\hat{D})| > \alpha] \leq \beta.$$

Here, the probability is taken over the coins of $\mathcal{A}$.

In an influential result, Blum et al. (2013) showed that any concept class C admits a differentially private sanitizer with sample complexity $O(\mathrm{VC}(C)\log|X|)$:

Theorem 24 (Blum et al. (2013)) *For any concept class C over a domain X, there exists an (α,β)-accurate and $(\epsilon,0)$-differentially private sanitizer $\mathcal{A}: X^n \to X^m$ for C when*

$$n = O\left(\frac{\mathrm{VC}(C)\cdot\log|X|\cdot\log(1/\alpha)}{\alpha^3\epsilon} + \frac{\log(1/\beta)}{\alpha\epsilon}\right),$$

and $m = O(\mathrm{VC}(C)\log(1/\alpha)/\alpha^2)$.

When relaxing to (ϵ,δ)-differential privacy, the private multiplicative weights algorithm of Hardt and Rothblum (2010) can sometimes achieve lower sample complexity (roughly $O(\log|C|\sqrt{\log|X|})$).

Theorem 25 (Hardt and Rothblum (2010)) *For any concept class C over a domain X, there exists an (α,β)-accurate and (ϵ,δ)-differentially private sanitizer $\mathcal{A}: X^n \to X^m$ for C when*

$$n = O\left(\frac{(\log|C| + \log(1/\beta))\cdot\sqrt{\log|X|\cdot\log(1/\delta)}}{\alpha^2\epsilon}\right),$$

and $m = O(\mathrm{VC}(C)\log(1/\alpha)/\alpha^2)$.

However, for specific concept classes, sanitizers are known to exist with much lower sample complexity. For example, Bun et al. (2015) gave a sanitizer for threshold functions with sample complexity roughly $2^{\log^*|X|}$ (improving on work of Beimel et al. (2013b)).

Proposition 26 (Bun et al. (2015)) *There exists an (α,β)-accurate and (ϵ,δ)-differentially private sanitizer for $\mathtt{THRESH}_X$ with sample complexity*

$$n = O\left(\frac{1}{\alpha\epsilon}\cdot 2^{\log^*|X|}\cdot\log^*|X|\cdot\log\left(\frac{\log^*|X|}{\epsilon\delta}\right)\cdot\log(1/\beta)\cdot\log^{2.5}(1/\alpha)\right).$$

2.6. Private learners and multi-learners

Generalizing on the concept of private learners (Kasiviswanathan et al., 2011), we say that an algorithm $\mathcal{A}$ is $(\alpha,\beta,\epsilon,\delta)$-private PAC k-learner for C using H if $\mathcal{A}$ is (α,β)-PAC k-learner for C using H, and $\mathcal{A}$ is (ϵ,δ)-differentially private (similarly with agnostic private PAC k-learners). We omit the parameter k when $k=1$ and the parameter δ when $\delta=0$.

For the case $k=1$, we have a generic construction with sample complexity proportional to $\log|C|$:

Theorem 27 (Kasiviswanathan et al. (2011)) *Let C be a concept class, and $\alpha,\beta,\epsilon > 0$. There exists an (α,β,ϵ)-private agnostic proper learner for C with sample complexity*

$$O\left((\log|C| + \log 1/\beta)(1/(\epsilon\alpha) + 1/\alpha^2)\right).$$

A number of works (Beimel et al., 2014, 2013a,b; Feldman and Xiao, 2014; Bun et al., 2015) have established upper and lower bounds for learning the specific concept classes $\mathtt{POINT}_X$ and $\mathtt{THRESH}_X$. In the case of pure differential privacy, $\mathtt{POINT}_X$ requires $\Theta(\log|X|)$ samples to learn properly (Beimel et al., 2014), but can be learned improperly with $O(1)$ samples. On the other hand, the class of threshold functions $\mathtt{THRESH}_X$ require $\Omega(\log|X|)$ samples to learn, even improperly (Feldman and Xiao, 2014). In the case of approximate differential privacy, $\mathtt{POINT}_x$ and $\mathtt{THRESH}_x$ can be learned properly with sample complexities $O(1)$ (Beimel et al., 2013b) and $\tilde{O}(2^{\log^*|X|})$ (Bun et al., 2015), respectively. Moreover, learning threshold functions requires sample complexity $\Omega(\log^*|X|)$ (Bun et al., 2015; Alon et al., 2018).

2.7. Private PAC learning vs. Empirical Learning

We saw by Theorem 12 that when an empirical k-learner $\mathcal{A}$ for a concept class C is run on a random sample of size $\Omega(\mathrm{VC}(C))$, it is also a (agnostic) PAC k-learner. In particular, if an empirical k-learner $\mathcal{A}$ is differentially private, then it also serves as a differentially private (agnostic) PAC k-learner.

Generalizing a result of (Bun et al., 2015), the next theorem shows that the converse is true as well: a differentially private (agnostic) PAC k-learner yields a private empirical k-learner with only a constant factor increase in the sample complexity.

Theorem 28 *Let $\epsilon \le 1$. Suppose $\mathcal{A}$ is an (ϵ,δ)-differentially private (α,β)-accurate (agnostic) PAC k-learner for a concept class $\mathcal{C}$ with sample complexity n. Then there is an (ϵ,δ)-differentially private (α,β)-accurate (agnostic) empirical k-learner $\tilde{\mathcal{A}}$ for $\mathcal{C}$ with sample complexity $m = 9n$. Moreover, if $\mathcal{A}$ is proper, then so is the resulting empirical learner $\tilde{\mathcal{A}}$.*

Proof We give the proof for the agnostic case; the non-agnostic case is argued identically, and is immediate from (Bun et al., 2015). To construct the empirical learner $\tilde{\mathcal{A}}$, we use the fact that the given learner $\mathcal{A}$ performs well on any distribution over labeled examples – in particular, it performs well on the uniform distribution over rows of the input database to $\tilde{\mathcal{A}}$. Consider a database $S = ((x_i, y_{1,i},\ldots,y_{k,i}))_{i=1}^m \in (X\times\{0,1\}^k)^m$. On input S, define $\tilde{\mathcal{A}}$ by sampling n rows from S (with replacement), and outputting the result of running $\mathcal{A}$ on the sample. Let $\mathcal{S}$ denote the uniform distribution over the rows of S, and let $\mathcal{S}_j$ be its marginal distribution which is uniform over $S|_j = ((x_i, y_{j,i}))_{i=1}^m$. Then sampling n rows from S is equivalent to sampling n rows i.i.d. from $\mathcal{S}$. Hence, if $(h_1,\ldots,h_k)$ is the output of $\mathcal{A}$ on the subsample, we have

$$\Pr\left[\max_{1\le j\le k}\left(\mathrm{error}_{S|_j}(h_j) - \min_{c\in C}\left(\mathrm{error}_{S_j}(c)\right)\right) > \alpha\right] = \Pr\left[\max_{1\le j\le k}\left(\mathrm{error}_{\mathcal{S}_j}(h_j) - \min_{c\in C}\left(\mathrm{error}_{\mathcal{S}_j}(c)\right)\right) > \alpha\right] \le \beta.$$

To show that $\tilde{\mathcal{A}}$ remains (ϵ,δ)-differentially private, we apply the following "secrecy-of-the-sample" lemma (Kasiviswanathan et al., 2011; Bun et al., 2015), which shows that the sampling procedure does not hurt privacy.

Lemma 29 *Fix $\epsilon \leq 1$ and let $\mathcal{A}$ be an (ϵ, δ)-differentially private algorithm with sample complexity n. For $m \geq 2n$, the algorithm $\tilde{\mathcal{A}}$ described above is $(\tilde{\epsilon}, \tilde{\delta})$ for*

$$\tilde{\epsilon} = \frac{6\epsilon m}{n} \quad \text{and} \quad \tilde{\delta} = 4\exp\left(\frac{6\epsilon m}{n}\right) \cdot \frac{m}{n} \cdot \delta.$$

■

3. Differentially Private Sanitization for Point Functions

We begin by presenting a new sanitizer for the class of point functions, which will be useful for our constructions in the following sections. Our point sanitizer improves and simplifies a result of Beimel et al. (2013b), and exhibits an essentially optimal sample complexity.

Proposition 30 *There exists an (α, β)-accurate and (ϵ, δ)-differentially private sanitizer for $\texttt{POINT}_X$ with sample complexity*

$$n = O\left(\frac{\log(1/\alpha\beta\delta)}{\alpha\epsilon}\right).$$

Proof

To give a $(2\alpha, \beta)$-accurate sanitizer, it suffices to produce, for each point function c_x, an approximate answer $a_x \in [0, 1]$ with $|a_x - c_x| \leq \alpha$. This is because given these approximate answers, one can reconstruct a database $\hat{D}$ of size $O(1/\alpha)$ with $|c_x(\hat{D}) - a_x| \leq \alpha$ for every $x \in X$.

The algorithm for producing the answers a_x is as follows.

Algorithm 2 Query release for $\texttt{POINT}_X$

Input: Privacy parameters (ϵ, δ), database $D \in X^n$

For each $x \in X$, do the following:

1. If $c_x(D) \leq \frac{\alpha}{4}$, release $a_x = 0$
2. Let $\hat{a}_x = c_x(D) + \text{Lap}(2/\epsilon n)$
3. If $\hat{a}_x \leq \frac{\alpha}{2}$, release $a_x = 0$
4. Otherwise, release $a_x = \hat{a}_x$

First, we argue that Algorithm 2 is (ϵ, δ)-differentially private. Below, we write $X \approx_{(\epsilon,\delta)} Y$ to denote the fact that for every measurable set S in the union of the supports of X and Y, we have $\Pr[X \in S] \leq e^{\epsilon} \Pr[Y \in S] + \delta$.

Let $D \sim D'$ be adjacent databases of size n, with $x \in D$ replaced by $x' \in D'$. Then the output distribution of the mechanism differs only on its answers to the queries c_x and $c_{x'}$. Let us focus on c_x. If both $c_x(D) \leq \alpha/4$ and $c_x(D') \leq \alpha/4$, then the mechanism always releases 0 for both queries. If both $c_x(D) > \alpha/4$ and $c_x(D') > \alpha/4$, then $a_x(D) \approx_{(\epsilon/2,0)} a_x(D')$

by properties of the Laplace mechanism. Finally, if $c_x(D) > \alpha/4$ but $c_x(D') \leq \alpha/4$, then $c_x(D') = 0$ with probability 1. Moreover, we must have $\texttt{POINT}_x(D) \leq \alpha/4 + 1/n$, so

$$\Pr[a_x(D) = 0] \geq \Pr[Lap(2/\epsilon n) < \alpha/4 - 1/n] = 1 - \frac{1}{2}\exp(-\epsilon n\alpha/8 + \epsilon/2) \geq 1 - \delta/2.$$

So in this case, $a_x(D) \approx_{(0,\delta/2)} a_x(D')$. Therefore, we conclude that overall $a_x(D) \approx_{(\epsilon/2,\delta/2)} a_x(D')$. An identical argument holds for $a_{x'}$, so the mechanism is (ϵ, δ)-differentially private.

Now we argue that the answers a_x are accurate. First, the answers are trivially α-accurate for all queries c_x on which $c_x(D) \leq \alpha/4$. For each of the remaining queries, it is α-accurate with probability at least

$$\Pr[|Lap(2/\epsilon n)| < \alpha/2] = 1 - \exp(-\epsilon n\alpha/4) \geq 1 - \frac{\alpha\beta}{4}.$$

Taking a union bound over the at most $4/\alpha$ queries with $\texttt{POINT}_x(D) > \alpha/4$, we conclude that the mechanism is α-accurate for all queries with probability at least $1 - \beta$.

■

4. Upper Bounds on the Sample Complexity of Private Multi-Learners

4.1. Generic Construction

In this section we present the following general upper bounds on the sample complexity of private k-learners.

Theorem 31 *Let C be a finite concept class, and let $k \geq 1$. There exists a proper agnostic $(\alpha, \beta, \epsilon)$-private PAC k-learner for C with sample complexity*

$$O_{\alpha,\beta,\epsilon}\Big(k \cdot \log k + \min\left\{k \cdot \log|C|\ ,\ (k + \log|X|) \cdot \text{VC}(C)\right\}\Big),$$

and there exists a proper agnostic $(\alpha, \beta, \epsilon, \delta)$-private PAC k-learner for C with sample complexity

$$O_{\alpha,\beta,\epsilon,\delta}\left(\sqrt{k} \cdot \log k + \min\left\{\sqrt{k} \cdot \log|C|\ ,\ (\sqrt{k} + \log|X|) \cdot \text{VC}(C)\ ,\ \sqrt{k} \cdot \text{VC}(C) + \sqrt{\log|X|} \cdot \log|C|\right\}\right).$$

The straightforward approach for constructing a private k-learner for a class C is to separately apply a (standard) private learner for C for each of the k target concepts. Using composition theorem 17 to argue the overall privacy guarantee of the resulting learner, we get the following observation.

Observation 32 *Let C be a concept class and let $k \geq 1$. If there is an $(\alpha, \beta, \epsilon, \delta)$-PAC learner for C with sample complexity n, then*

- *There is an $(\alpha, k\beta, k\epsilon, k\delta)$-PAC k-learner for C with sample complexity n.*
- *There is an $(\alpha, k\beta, O(\sqrt{k\log(\frac{1}{\delta})}\epsilon + k\epsilon^2), O(k\delta))$-PAC k-learner for C with sample complexity n.*

Moreover, if the initial learner is proper and/or agnostic, then so is the resulting learner.

In cases where sample efficient private PAC learners exist, it might be useful to apply Observation 32 in order to obtain a private k-learner. For example, Beimel et al. (2014, 2013a) gave an improper agnostic (α,β,ϵ)-PAC learner for $\texttt{POINT}_X$ with sample complexity $O_\alpha(\frac{1}{\epsilon}\log\frac{1}{\beta})$. Using Observation 32 yields the following corollary.

Corollary 33 *There exists an improper agnostic (α,β,ϵ)-PAC k-learner for $\texttt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon}(k\log k)$.*

For a general concept class C, we can use Observation 32 with the generic construction of Theorem 27, stating that for every concept class C there exists a private agnostic proper learner $\mathcal{A}$ that uses $O(\log|C|)$ labeled examples.

Corollary 34 *Let C be a concept class, and $\alpha,\beta,\epsilon>0$. There exists an (α,β,ϵ)-private agnostic proper k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon}(k\cdot\log|C|+k\cdot\log k)$. Moreover, there exists an $(\alpha,\beta,\epsilon,\delta)$-private agnostic proper k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(\sqrt{k}\cdot\log|C|+\sqrt{k}\cdot\log k)$.*

Example 1 *There exists a proper agnostic (α,β,ϵ)-PAC k-learner for $\texttt{PAR}_d$ with sample complexity $O_{\alpha,\beta,\epsilon}(kd+k\log k)$.*

As we will see in Section 6, the bounds of Corollary 33 and Example 1 on the sample complexity of k-learning $\texttt{POINT}_X$ and $\texttt{PAR}_d$ are tight (up to logarithmic factors). That is, with pure-differential privacy, the direct sum gives (roughly) optimal bounds for improperly learning $\texttt{POINT}_X$, and for (properly or improperly) learning $\texttt{PAR}_d$. This is not the case for learning $\texttt{THRESH}_X$ or for *properly* learning learning $\texttt{POINT}_X$.

In order to avoid the factor $k\log|C|$ (or $\sqrt{k}\log|C|$) in Corollary 34, we now show how an idea used in (Beimel et al., 2015) (in the context of semi-supervised learning) can be used to construct sample efficient private k-learners. In particular, this construction will achieve tight bounds for learning $\texttt{THRESH}_X$ and for properly learning learning $\texttt{POINT}_X$ under pure-differential privacy.

Fix a concept class C, target concepts $c_1,\ldots,c_k\in C$, and a k-labeled database S (we use D to denote the unlabeled portion of S). For every $1\le j\le k$, the goal is to identify a hypothesis $h_j\in C$ with low $\text{error}_D(c_j,h_j)$ (such a hypothesis also has good generalization). Beimel et al. (2015) observed that given a sanitization $\hat{D}$ of D w.r.t. $C^\oplus=\{f\oplus g: f,g\in C\}$, for every $f,g\in C$ it holds that

$$\text{error}_D(f,g)=\frac{1}{|D|}|\{x\in D:(f\oplus g)(x)=1\}|\approx\frac{1}{|\hat{D}|}|\{x\in\hat{D}:(f\oplus g)(x)=1\}|=\text{error}_{\hat{D}}(f,g).$$

Hence, a hypothesis h with low $\text{error}_{\hat{D}}(h,c_j)$ also has low $\text{error}_D(h,c_j)$ and vice versa. Let H be a minimal subset of C such that for every $c\in C$, there exists $f^*\in H$ such that $f^*(x)=c(x)$ for every $x\in\hat{D}$. Then in particular, for every j, there exists $f_j^*\in H$ that agrees with c_j on $\hat{D}$, i.e., there exists $f_j^*\in H$ s.t. $\text{error}_{\hat{D}}(f_j^*,c_j)=0$, and hence $\text{error}_D(f_j^*,c_j)$ is also low. The thing that works in our favor here is that H is small – at most $2^{|\hat{D}|}\le 2^{\text{VC}(C)}$ – and hence choosing a hypothesis out of H is easy. Therefore, for every j we can use the exponential mechanism to identify a hypothesis $h_j\in H$ with low $\text{error}_D(h_j,c_j)$.

Lemma 35 *Let C be a concept class, and $\alpha,\beta,\epsilon,\delta>0$. There exists an (α,β,ϵ)-private agnostic k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon}(\text{VC}(C)\cdot\log|X|+k\cdot\text{VC}(C)+k\cdot\log k)$. Moreover, there exists an $(\alpha,\beta,\epsilon,\delta)$-private agnostic k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(\min\{\text{VC}(C)\cdot\log|X|,\log|C|\cdot\sqrt{\log|X|}\}+\sqrt{k}\cdot\text{VC}(C)+\sqrt{k}\cdot\log k)$.*

Lemma 35 follows from the following lemma.

Lemma 36 *Let $\epsilon'>0$ and let $\mathcal{A}$ be an $(\frac{\alpha}{5},\frac{\beta}{5})$-accurate (ϵ,δ)-private sanitizer for $C^\oplus$ with sample complexity m. Then there is an (α,β)-PAC agnostic k-learner for C with sample complexity*

$$O\left(m+\frac{\text{VC}(C)}{\alpha^3\epsilon'}\log(\frac{1}{\alpha})+\frac{1}{\alpha\epsilon'}\log(\frac{k}{\beta})+\frac{1}{\alpha^2}\text{VC}(C)\log(\frac{k}{\alpha\beta})\right).$$

Moreover, it is both $(\epsilon+k\epsilon',\delta)$ and $(\epsilon+\sqrt{2k\ln(1/\delta)}\epsilon'+2k\epsilon'^2,2\delta)$-differentially private.

Using Lemma 36 with the generic sanitizer of Theorem 24 or Theorem 25 results in Lemma 35.

Algorithm 3 *GenericLearner*

Input: Concept class C, privacy parameters $\epsilon',\epsilon,\delta$, and a k-labeled database $S=(x_i,y_{i,1},\ldots,y_{i,k})_{i=1}^n$. We use $D=(x_i)_{i=1}^n$ to denote the unlabeled portion of S.
Used Algorithm: An $(\frac{\alpha}{5},\frac{\beta}{5})$-accurate (ϵ,δ)-private sanitizer for $C^\oplus$ with sample complexity m.

1. Initialize $H=\emptyset$.
2. Construct an (ϵ,δ)-private sanitization $\widetilde{D}$ of D w.r.t. $C^\oplus$, where $|\widetilde{D}|=O\left(\frac{\text{VC}(C^\oplus)}{\alpha^2}\log(\frac{1}{\alpha})\right)=O\left(\frac{\text{VC}(C)}{\alpha^2}\log(\frac{1}{\alpha})\right)$.
3. Let $B=\{b_1,\ldots,b_{|B|}\}$ be the set of all points appearing at least once in $\widetilde{D}$.
4. For every $(z_1,\ldots,z_{|B|})\in\Pi_C(B)=\{(c(b_1),\ldots,c(b_{|B|})):c\in C\}$, add to H an arbitrary concept $c\in C$ s.t. $c(b_\ell)=z_\ell$ for every $1\le\ell\le|B|$.
5. For every $1\le j\le k$, use the exponential mechanism with privacy parameter ϵ' to choose and return a hypothesis $h_j\in H$ with (approximately) minimal error on the examples in S w.r.t. their j^{th} label.

Proof [Proof of Lemma 36] The proof is via the construction of *GenericLearner* (algorithm 3). Note that *GenericLearner* only accesses S via a sanitizer (on Step 2) and using the exponential mechanism (on Step 5). Composition theorem 17 state that *GenericLearner* is both $(\epsilon+k\epsilon',\delta)$-differentially private and $(\epsilon+\sqrt{2k\ln(1/\delta)}\epsilon'+2k\epsilon'^2,2\delta)$-differentially private. We, thus, only need to prove that with high probability the learner returns α-good hypotheses.

Fix a distribution $\mathcal{P}$ over $X\times\{0,1\}^k$, and let $\mathcal{P}_j$ denote the marginal distribution of $\mathcal{P}$ on the examples and the jth label. Let S consist of examples $(x_i,y_{i,1},\ldots,y_{i,k})\sim\mathcal{P}$. We use $D=(x_i)_{i=1}^n$ to denote the unlabeled portion of S, and use $S|_j=((x_i,y_{j,i}))_{i=1}^n$ to denote a

database containing the examples in S together with their j^{th} label. Define the following three events:

E_1 : For every $f, h \in C$ it holds that $|\text{error}_D(f,h) - \text{error}_{\widetilde{D}}(f,h)| \leq \frac{2\alpha}{5}$.

E_2 : For every $f \in C$ and for every $1 \leq j \leq k$ it holds that $|\text{error}_{S|_j}(f) - \text{error}_{\mathcal{P}_j}(f)| \leq \frac{\alpha}{5}$.

E_3 : For every $1 \leq j \leq k$, the hypothesis h_j chosen by the exponential mechanism is such that $\text{error}_{S|_j}(h_j) \leq \frac{\alpha}{5} + \min_{f \in H}\left\{\text{error}_{S|_j}(f)\right\}$.

We first argue that when these three events happen algorithm *GenericLearner* returns good hypotheses. Fix $1 \leq j \leq k$, and let $c_j^* = \text{argmin}_{f \in C}\{\text{error}_{\mathcal{P}_j}(f)\}$. We denote $\Delta = \text{error}_{\mathcal{P}_j}(c_j^*)$. We need to show that if $E_1 \cap E_2 \cap E_3$ occurs, then the hypothesis h_j returned by *GenericLearner* is s.t. $\text{error}_{\mathcal{P}_j}(h_j) \leq \alpha + \Delta$.

For every $(y_1, \ldots, y_{|B|}) \in \Pi_C(B)$, algorithm *GenericLearner* adds to H a hypothesis f s.t. $\forall 1 \leq \ell \leq |B|,\ f(b_\ell) = y_\ell$. In particular, H contains a hypothesis h_j^* s.t. $h_j^*(x) = c_j^*(x)$ for every $x \in B$, that is, a hypothesis h_j^* s.t. $\text{error}_{\widetilde{D}}(h_j^*, c_j^*) = 0$. As event E_1 has occurred we have that this h_j^* satisfies $\text{error}_D(h_j^*, c_j^*) \leq \frac{2\alpha}{5}$. Using the triangle inequality (and event E_2) we get that this h_j^* satisfies $\text{error}_{S|_j}(h_j^*) \leq \text{error}_D(h_j^*, c_j^*) + \text{error}_{S|_j}(c_j^*) \leq \frac{3\alpha}{5} + \Delta$. Thus, event E_3 ensures that algorithm *GenericLearner* chooses (using the exponential mechanism) a hypothesis $h_j \in H$ s.t. $\text{error}_{S|_j}(h_j) \leq \frac{4\alpha}{5} + \Delta$. Event E_2 ensures, therefore, that this h_j satisfies $\text{error}_{\mathcal{P}_j}(h_j) \leq \alpha + \Delta$. We will now show $E_1 \cap E_2 \cap E_3$ happens with high probability.

Standard arguments in learning theory state that (w.h.p.) the empirical error on a (large enough) random sample is close to the population error. Specifically, by setting $n \geq O(\frac{1}{\alpha^2}\,\text{VC}(C)\log(\frac{k}{\alpha\beta}))$, Theorem 9 ensures that Event E_2 occurs with probability at least $(1 - \frac{2}{5}\beta)$.

Assuming that $n \geq m$ (the sample complexity of the sanitizer used in Step 5), with probability at least $(1 - \frac{\beta}{5})$ for every $(h \oplus f) \in C^{\oplus}$ (i.e., for every $h, f \in C$) it holds that

$$\begin{aligned}
\frac{\alpha}{5} &\geq |Q_{(h\oplus f)}(D) - Q_{(h\oplus f)}(\widetilde{D})| \\
&= \left| \frac{|\{x \in D : (h\oplus f)(x)=1\}|}{|D|} - \frac{|\{x \in \widetilde{D} : (h\oplus f)(x)=1\}|}{|\widetilde{D}|} \right| \\
&= \left| \frac{|\{x \in D : h(x)\neq f(x)\}|}{|D|} - \frac{|\{x \in \widetilde{D} : h(x)\neq f(x)\}|}{|\widetilde{D}|} \right| \\
&= \left|\text{error}_D(h,f) - \text{error}_{\widetilde{D}}(h,f)\right|.
\end{aligned}$$

Event E_1 occurs therefore with probability at least $(1 - \frac{\beta}{5})$.

The exponential mechanism ensures that the probability of event E_3 is at least $1 - k|H| \cdot \exp(-\epsilon'\alpha m/10)$ (see Proposition 21). Note that $\log|H| \leq |B| \leq |\widetilde{D}| = O\left(\frac{\text{VC}(C)}{\alpha^2}\log(\frac{1}{\alpha})\right)$. Therefore, for $n \geq O\left(\frac{\text{VC}(C)}{\alpha^3\epsilon'}\log(\frac{1}{\alpha}) + \frac{1}{\alpha\epsilon'}\log(\frac{k}{\beta})\right)$, Event E_3 occurs with probability at least $(1 - \frac{\beta}{5})$.

All in all, setting $n \geq O\left(m + \frac{\text{VC}(C)}{\alpha^3\epsilon'}\log(\frac{1}{\alpha}) + \frac{1}{\alpha\epsilon'}\log(\frac{k}{\beta}) + \frac{1}{\alpha^2}\,\text{VC}(C)\log(\frac{k}{\alpha\beta})\right)$, ensures that the probability of *GenericLearner* failing is at most β. ■

Theorem 31 now follows by combining Lemma 35 and Corollary 34.

For certain concept classes, there are sanitizers with substantially lower sample complexity than the generic sanitizers. Combining Lemma 35 with Proposition 30, we obtain:

Corollary 37 *There is an (α, β)-PAC agnostic k-learner for* $\texttt{POINT}_X$ *with sample complexity*

$$O\left(\frac{\log(1/\alpha\beta\delta)}{\alpha\epsilon} + \frac{\log(1/\alpha)}{\alpha^3\epsilon'} + \frac{\log(k/\beta)}{\alpha\epsilon'} + \frac{\log(k/\alpha\beta)}{\alpha^2}\right).$$

Moreover, it is both $(\epsilon + k\epsilon', \delta)$ and $(\epsilon + \sqrt{2k\ln(1/\delta)}\epsilon' + 2k\epsilon'^2, 2\delta)$-differentially private.

Similarly, combining Lemma 35 with Proposition 26, we obtain:

Corollary 38 *There is an (α, β)-PAC agnostic k-learner for* $\texttt{THRESH}_X$ *with sample complexity*

$$O\left(\frac{2^{\log^*|X|} \cdot \log^*|X| \cdot \log\left(\frac{\log^*|X|}{\epsilon\delta}\right) \cdot \log(1/\beta) \cdot \log^{2.5}(1/\alpha)}{\alpha\epsilon} + \frac{\log(1/\alpha)}{\alpha^3\epsilon'} + \frac{\log(k/\beta)}{\alpha\epsilon'} + \frac{\log(k/\alpha\beta)}{\alpha^2}\right).$$

Moreover, it is both $(\epsilon + k\epsilon', \delta)$ and $(\epsilon + \sqrt{2k\ln(1/\delta)}\epsilon' + 2k\epsilon'^2, 2\delta)$-differentially private.

4.2. Upper Bounds for Approximate Private Multi-Learners

In this section we give two examples of cases where the sample complexity of private k-learning is of the same order as that of non-private k-learning (the sample complexity does not depend on k). Our algorithms are (ϵ, δ)-differentially private, and rely on stability arguments: the identity of the best k concepts, as an entire vector, is unlikely to change on nearby k-labeled databases. Hence, it can be released privately.

4.2.1. Learning Parities under the Uniform Distribution

Theorem 39 *For every k, d there exists an $(\alpha{=}0, \beta, \epsilon, \delta)$-PAC (non-agnostic) k-learner for* $\texttt{PAR}_d$ *under the uniform distribution with sample complexity $O(d\log(\frac{1}{\beta}) + \frac{1}{\epsilon}\log(\frac{1}{\beta\delta}))$.*

Recall that (even without privacy constraints) the sample complexity of PAC learning $\texttt{PAR}_d$ under the uniform distribution is $\Omega(d)$. Hence the sample complexity of privately k-learning $\texttt{PAR}_d$ (non-agnostically) under the uniform distribution is of the same order as that of non-private k-learning.

For the intuition behind Theorem 39, let $c_1, \ldots, c_k$ denote the k target concepts, and consider the quality function $q(D, (h_1, \ldots, h_k)) = \max_{1 \leq j \leq k}\{\text{error}_D(h_j, c_j)\}$. On a large enough sample D we expect that $q(D, (h_1, \ldots, h_k)) \approx \frac{1}{2}$ for every $(h_1, \ldots, h_k) \neq (c_1, \ldots, c_k)$, while $q(D, (c_1, \ldots, c_k)) = 0$. The k target concepts can hence be privately identified (exactly) using stability techniques.

In order to make our algorithm computationally efficient, we apply the "subsample and aggregate" idea of Nissim et al. (2007). We divide the input sample into a small number

Algorithm 4 *ParityLearner*

Input: Parameters ϵ, δ, and a k-labeled database S of size $n = O(\frac{d}{\epsilon}\log(\frac{1}{\beta\delta}))$.
Output: Hypotheses $h_1, \ldots, h_k$.

1. Split S into $m = O(\frac{1}{\epsilon}\log(\frac{1}{\beta\delta}))$ disjoint samples $S_1, \ldots, S_m$ of size $O(d)$ each. Initiate Y as the empty multiset.
2. For every $1 \leq t \leq m$:
   (a) For every $1 \leq j \leq k$ try to use Gaussian elimination to identify a parity function y_j that agrees with the labels of the jth column of S_t.
   (b) If a parity is identified for every j, then set $Y = Y \cup \{(y_1, ..., y_k)\}$. Otherwise set $Y = Y \cup \{\perp\}$.
3. Use algorithm $\mathcal{A}_{\text{dist}}$ with privacy parameters ϵ, δ to choose and return a vector of k parity functions $(h_1, \ldots, h_k) \in (\texttt{PAR}_d)^k$ with a large number of appearances in Y.

of subsamples, use Gaussian elimination to (non-privately) identify a candidate hypothesis vector on each subsample, and then select from these candidates privately.

Proof [Proof of Theorem 39] The proof is via the construction of *ParityLearner* (algorithm 4). First note that changing a single input element in S can change (at most) one element of Y. Hence, applying (the (ϵ, δ)-private) algorithm $\mathcal{A}_{\text{dist}}$ on Y preserves privacy (applying *ParityLearner* on neighboring inputs amounts to executing $\mathcal{A}_{\text{dist}}$ on neighboring inputs).

Now fix k target concepts $c_1, \ldots, c_k \in \texttt{PAR}_d$ and let S be a random k-labeled database containing n i.i.d. elements from the uniform distribution U_d over $X = \{0,1\}^d$, each labeled by $c_1, \ldots, c_k$. Observe that (for every $1 \leq t \leq m$) we have that S_t contains i.i.d. elements from U_d labeled by $c_1, \ldots, c_k$. We use D_t to denote the unlabeled portion of S_t. Standard arguments in learning theory (cf. Theorem 8) state that for $|S_t| \geq \Omega(d)$,

$$\Pr\left[\exists h, f \in \texttt{PAR}_d \text{ s.t. } \text{error}_{U_d}(h,f) \geq \frac{1}{4} \quad \wedge \quad \text{error}_{D_t}(h,f) \leq \frac{1}{40}\right] \leq \frac{1}{8}.$$

The above inequality holds, in particular, for every hypothesis $h \in \texttt{PAR}_d$ and every target concept c_j, and hence,

$$\Pr\left[\exists h \in \texttt{PAR}_d \text{ and } j \text{ s.t. } \text{error}_{U_d}(h,c_j) \geq \frac{1}{4} \quad \wedge \quad \text{error}_{D_t}(h,c_j) \leq \frac{1}{40}\right] \leq \frac{1}{8}.$$

Recall that under the uniform distribution, the only $h \in \texttt{PAR}_d$ s.t. $\text{error}_{U_d}(h,c_j) \neq \frac{1}{2}$ is c_j itself, and hence

$$\Pr\left[\exists h \in \texttt{PAR}_d \text{ and } j \text{ s.t. } h \neq c_j \quad \wedge \quad \text{error}_{D_t}(h,c_j) \leq \frac{1}{40}\right] \leq \frac{1}{8}.$$

So, for every $1 \leq t \leq m$, with probability $7/8$ we have that for every label column j the only hypothesis with empirical error less than $\frac{1}{40}$ on S_t is the jth target concept itself (with

empirical error 0). In such a case, step 2a (Gaussian elimination) identifies exactly the vector of k target concepts $(c_1, \ldots, c_k)$. Since $m \geq O(\log(\frac{1}{\beta}))$, the Chernoff bound ensures that except with probability $\beta/2$, the vector $(c_1, \ldots, c_k)$ is identified in at least $3/4$ of the iterations of step 2. Assuming that this is the case, the vector $(c_1, \ldots, c_k)$ appears in Y at least $3m/4$ times, while every other vector can appear at most $m/4$ times. Provided that $m \geq O(\frac{1}{\epsilon}\log(\frac{1}{\beta\delta}))$, algorithm $\mathcal{A}_{\text{dist}}$ ensures that the k target concepts are chosen with probability $1 - \beta/2$.

All in all, algorithm *ParityLearner* identifies the k target concepts (exactly) with probability $1 - \beta$, provided that $n \geq O(\frac{d}{\epsilon}\log(\frac{1}{\beta\delta}))$. ■

4.2.2. Learning Points

We next show that the class of $\texttt{POINT}_X$ can be (non-agnostically) k-learned using constant sample complexity, matching the non-private sample complexity.

Theorem 40 *For every domain X and every $k \in \mathbb{N}$ there exists an $(\alpha, \beta, \epsilon, \delta)$-PAC (non-agnostic) k-learner for $\texttt{POINT}_X$ with sample complexity $O(\frac{1}{\alpha\epsilon}\log(\frac{1}{\alpha\beta\delta}))$.*

The proof is via the construction of Algorithm 5. The algorithm begins by privately identifying (using sanitization) a set of $O(1/\alpha)$ "heavy" elements in the input database, appearing $\Omega(\alpha)$ times. The k labels of such a heavy element can be privately identified using stability arguments (since their duplicity in the database is large). The labels of a "non-heavy" element can be set to 0 since a target concept can evaluate to 1 on at most one such non-heavy element, in which case the error is small.

Notation. We use $\#_S(x)$ to denote the duplicity of a domain element x in a database S. For a distribution μ we denote $\mu(x) = \Pr_{\hat{x}\sim\mu}[\hat{x} = x]$.

Proof The proof is via the construction of *PointLearner* (algorithm 5). First note the algorithm only access the input database using sanitization on step 1, and using algorithm $\mathcal{A}_{\text{dist}}$ on step 4. By composition theorem 17, algorithm *PointLearner* is (ϵ, δ)-differentially private.

Let μ be a distribution over X, and let $c_1, \ldots, c_k \in \texttt{POINT}_X$ be the fixed target concepts. Consider the execution of *PointLearner* on a database $S = (x_i, y_{i,1}, \ldots, y_{i,k})_{i=1}^n$ sampled from μ and labeled by $c_1, \ldots, c_k$. We use D to denote the unlabeled portion of S, $\hat{D}$ for the sanitization of D constructed on step 1, and write $m = |\hat{D}|$. Define the following good events.

E_1 : For every $x \in X$ s.t. $\mu(x) \geq \alpha$ it holds that $\frac{1}{n}\#_S(x) \geq \alpha/10$.

E_2 : For every $x \in X$ we have that $|\frac{1}{m}\#_{\hat{D}}(x) - \frac{1}{n}\#_S(x)| \leq \alpha/30$.

E_3 : Algorithm $\mathcal{A}_{\text{dist}}$ returns a vector set V s.t. $q(S, x, \vec{v}_x) \geq 1$ for every $x \in G$.

We now argue that when these three events happen algorithm *PointLearner* returns good hypotheses. First, observe that the set G contains every element x s.t. $\mu(x) \geq \alpha$: Let x be s.t. $\mu(x) \geq \alpha$. As event E_1 has occurred, we have that $\frac{1}{n}\#_S(x) \geq \alpha/10$. As event E_2 has occurred, we have that $\frac{1}{m}\#_{\hat{D}}(x) \geq \alpha/15$, and therefore $x \in G$.

Algorithm 5 *PointLearner*

Input: Privacy parameters ϵ, δ, and a k-labeled database $S = (x_i, y_{i,1}, \ldots, y_{i,k})_{i=1}^n$. We use $D = (x_i)_{i=1}^n$ to denote the unlabeled portion of S.
Output: Hypotheses $h_1, \ldots, h_k$.

1. Let $\hat{D} \in X^m$ be an $(\frac{\epsilon}{2}, \frac{\delta}{2})$-private $(\frac{\alpha}{30}, \frac{\beta}{4})$-accurate sanitization of D w.r.t. $\texttt{POINT}_X$ (e.g., using Proposition 30).
2. Let $G = \{x \in X : \frac{1}{m}\#_{\hat{D}}(x) \geq \alpha/15\}$ be the set of all "$\frac{\alpha}{15}$-heavy" domain elements w.r.t. the sanitization $\hat{D}$. Note that $|G| \leq 15/\alpha$.
3. Let q be the quality function that on input a k-labeled database S, a domain element x, and a binary vector $\vec{v} \in \{0,1\}^k$, returns the number of appearances of $(x, \vec{v})$ in S. That is, $q(S, x, (v_1, \ldots, v_k)) = |\{i : x_i = x \wedge y_{i,1} = v_1 \wedge \cdots \wedge y_{i,k} = v_k\}|$.
4. Use algorithm $\mathcal{A}_{\text{dist}}$ with privacy parameters $\frac{\epsilon}{2}, \frac{\delta}{2}$ to choose a set of vectors $V = \{\vec{v}_x \in \{0,1\}^k : x \in G\}$ maximizing $Q(S, V) = \min_{\vec{v}_x \in V}\{q(S, x, \vec{v}_x)\}$. That is, we use algorithm $\mathcal{A}_{\text{dist}}$ to choose a set of $|G|$ vectors – a vector $\vec{v}_x$ for every $x \in G$ – such that the minimal number of appearances of an entry $(x, \vec{v}_x)$ in the database S is maximized.
5. For $1 \leq j \leq k$: If the jth entry of every $\vec{v}_x \in V$ is 0, then set $h_j \equiv 0$. Otherwise, let x be s.t. $\vec{v}_x \in V$ has 1 as its jth entry, and define $h_j : X \to \{0,1\}$ as $h_j(y) = 1$ iff $y = x$.
6. Return $h_1, \ldots, h_k$.

Note that if $q(S, x, \vec{v}) \geq 1$ then the example x is labeled as $\vec{v}$ by the target concepts. Thus, as event E_3 has occurred, for every $\vec{v}_x \in V$ it holds that $\vec{v}_x = (c_1(x), \ldots, c_k(x))$. Now let h_j be the jth returned hypothesis. We next show that h_j is α-good. If $h \not\equiv 0$, then let x be the unique element s.t. $h_j(x) = 1$, and note that (according to step 5) the jth entry of $\vec{v}_x$ is 1, and hence, $c_j(x) = 1$. So $h_j = c_j$ (since c_j is a concept in $\texttt{POINT}_X$).

If $h_j \equiv 0$ then the jth entry of every $\vec{v}_x \in V$ is 0. Note that in such a case h_j only errs on the unique element x s.t. $c_j(x) = 1$, and it suffices to show that $\mu(x) < \alpha$. Assume towards contradiction that $\mu(x) \geq \alpha$. As before, event $E_1 \cap E_2$ implies that $x \in G$. As event E_3 has occurred, we also have that $\vec{v}_x \in V$ is s.t. $q(S, x, \vec{v}_x) \geq 1$, and the example x is labeled as $\vec{v}_x$ by the target concepts. This contradicts the assumption that the jth entry of $\vec{v}_x \in V$ is 0.

Thus, whenever $E_1 \cap E_2 \cap E_3$ happens, algorithm *PointLearner* returns α-good hypotheses. We will now show $E_1 \cap E_2 \cap E_3$ happens with high probability. Provided $n \geq O(\frac{1}{\alpha\epsilon}\log(\frac{1}{\alpha\delta}))$, event E_2 is guaranteed to hold with all but probability $\beta/4$ by the utility properties of the sanitizer used on step 1. See Proposition 30.

Theorem 8 (VC bound) ensures that event E_1 holds with probability $1 - \beta/4$, provided that $n \geq O(\frac{1}{\alpha}\log(\frac{1}{\alpha\beta}))$. To see this, let $z \equiv 0$ denote the constant 0 hypothesis, and consider the class $C = \texttt{POINT}_X \cup \{z\}$. Note that $\text{VC}(C) = 1$. Hence, Theorem 12 states that, with all but probability $1 - \beta/4$, for every $c \in \texttt{POINT}_x$ s.t. $\text{error}_\mu(c, z) \geq \alpha$ it holds that $\text{error}_D(c, z) \geq \alpha/10$. That is, with all but probability $1 - \beta/4$, for every $x \in X$ s.t. $\mu(x) \geq \alpha$ it holds that $\frac{1}{n}\#_D(x) = \frac{1}{n}\#_S(x) \geq \alpha/10$.

Before analyzing event E_3, we show that if E_2 occurs, then every $x \in G$ is s.t. $\#_S(x) \geq \alpha/30$. Let $x \in G$, that is, x s.t. $\frac{1}{m}\#_{\hat{D}}(x) \geq \alpha/15$. Assuming event E_2 has occurred, we therefore have that $\frac{1}{n}\#_S(x) \geq \alpha/30$. So every $x \in G$ appears in S at least $\alpha n/30$ times with the labels $(c_1(x), \ldots, c_k(x)) \triangleq \vec{c}(x)$. Thus, $q(S, x, \vec{c}(x)) \geq \alpha n/30$. In addition, for every $\vec{v} \neq \vec{c}(x)$ it holds that $q(S, x, \vec{v}) = 0$, since *every* appearance of the example x is labeled by the target concepts. Hence, provided that $n \geq O(\frac{1}{\alpha\epsilon}\log(\frac{1}{\beta\delta}))$, algorithm $\mathcal{A}_{\text{dist}}$ ensures that event E_3 happens with probability at least $1 - \beta/2$.

Overall, $E_1 \cap E_2 \cap E_3$ happens with probability at least $1 - \beta$. ■

5. Approximate Privacy Lower Bounds from Fingerprinting Codes

In this section, we show how fingerprinting codes can be used to obtain $\text{poly}(k)$ lower bounds against privately learning k concepts, even for very simple concept classes. Fingerprinting codes were introduced by Boneh and Shaw (1998) to address the problem of watermarking digital content. The connection between fingerprinting codes and differential privacy lower bounds was established by Bun et al. (2014) in the context of private query release, and has since been extended to a number of other differentially private analyses (Bassily et al., 2014; Dwork et al., 2014; Steinke and Ullman, 2015; Bun et al., 2015).

A (fully-collusion-resistant) fingerprinting code is a scheme for distributing codewords $w_1, \ldots, w_n$ to n users that can be uniquely traced back to each user. Moreover, if any group of users combines its codewords into a pirate codeword w', then the pirate codeword can still be traced back to one of the users who contributed to it. Of course, without any assumption on how the pirates can produce their combined codeword, no secure tracing is possible. To this end, the pirates are constrained according to a *marking assumption*, which asserts that the combined codeword must agree with at least one of the pirates' codeword in each position. Namely, at an index j where $w_{ij} = b$ for every $i \in b$, the pirates are constrained to output w' with $w'_j = b$ as well.

To illustrate our technique, we start with an informal discussion of how the original Boneh-Shaw fingerprinting code yields an $\tilde{\Omega}(k^{1/3})$ sample complexity lower bound for multi-learning threshold functions. For parameters n and k, the (n, k)-Boneh-Shaw codebook is a matrix $W \in \{0,1\}^{n \times k}$, whose rows w_i are the codewords given to users $i = 1, \ldots, n$. The codebook is built from a number of highly structured columns, where a "column of type i" consists of n bits where the first i bits are set to 1 and the last $n - i$ bits are set to 0. For $i = 1, \ldots, n-1$, each column of type i is repeated a total of $k/(n-1)$ times, and the codebook W is obtained as a random permutation of these k columns. The security of the Boneh-Shaw code is a consequence of the secrecy of this random permutation. If a coalition of pirates is missing the codeword of user i, then it is unable to distinguish columns of type $i - 1$ from columns of type i. Hence, if a pirate codeword is too consistent with a user i's codeword in both the columns of type $i - 1$ and the columns of type i, a tracing algorithm can reasonably conclude that user i contributed to it. Boneh and Shaw showed that such a code is indeed secure for $k = \tilde{O}(n^3)$.

To see how this fingerprinting code gives a lower bound for multi-learning thresholds, consider thresholds over the data universe $X = \{1, \ldots, |X|\}$ for $|X| \geq n$. The key observation is that each column of the Boneh-Shaw codebook can be obtained as a labeling of

the examples $1, \ldots, n$ by a threshold concept. Namely, a column of type i is the labeling of $1, \ldots, n$ by the concept c_i. Now suppose a coalition of users $T \subseteq [n]$ constructs a database S where each row is an example $i \in T$ together with the labels $w_{i1}, \ldots, w_{ik}$ coming from the codeword given to user i. Let $(h_1, \ldots, h_k)$ be the hypotheses produced by running a threshold multi-learner on the database. If every user has a bit b at index j of her codeword, then the hypothesis produced by the learner must also evaluate to b on most of the examples. Thus, the empirical averages of the hypotheses $(h_1, \ldots, h_k)$ on the examples can be used to obtain a pirate codeword satisfying the marking assumption. The security of the fingerprinting code, i.e. the fact that this codeword can be traced back to a user $i \in T$, implies that the learner cannot be differentially private. Hence, n samples is insufficient for privately learning $k = \tilde{O}(n^3)$ threshold concepts, giving a sample complexity lower bound of $\tilde{\Omega}(k^{1/3})$.

The lower bounds in this section are stated for empirical learning, but extend to PAC learning by Theorem 28. We also remark that they hold against the relaxed privacy notion of *label privacy*, where differential privacy only needs to hold with respect to changing the labels of one example.

5.1. Fingerprinting Codes

An (n, k)-*fingerprinting code* consists of a pair of randomized algorithms (Gen, Trace). The parameter n is the number of users supported by the fingerprinting code, and k is the length of the code. The codebook generator Gen produces a *codebook* $W \in \{0,1\}^{n \times k}$. Each row $w_i \in \{0,1\}^k$ of W is the *codeword* of user i. For a subset $T \subseteq [n]$, we let W_T denote the set $\{w_i : i \in T\}$ of codewords belonging to users in T. The accusation algorithm Trace takes as input a pirate codeword w' and accuses some $i \in [n]$ (or $\perp$ if it fails to accuse any user).

We define the feasible set of pirate codewords for a coalition T and codebook W by

$$F(W_T) = \{w' \in \{0,1\}^k : \forall j = 1, \ldots, k \; \exists i \in S \text{ s.t. } w_{ij} = w'_j\}.$$

The basic marking assumption is that the pirate codeword $w' \in F(W_T)$. We say column j is b-*marked* if $w_{ij} = b$ for every $i \in [n]$.

Definition 41 (Fingerprinting Codes) *For $n, k \in \mathbb{N}$ and $\xi \in (0, 1]$, a pair of algorithms* (Gen, Trace) *is an* (n, k)-fingerprinting code with security ξ *if* Gen *outputs a codebook $W \in \{0,1\}^{n \times k}$ and for every (possibly randomized) adversary $\mathcal{A}_{FP}$, and every coalition $T \subseteq [n]$, if we take $w' \leftarrow_R \mathcal{A}_{FP}(W_T)$, then the following properties hold.*

Completeness: $\Pr[w' \in F(W_T) \wedge \text{Trace}(w') = \perp] \leq \xi$,

Soundness: $\Pr[\text{Trace}(w') \in [n] \setminus T] \leq \xi$,

Each probability is taken over the coins of Gen, Trace, *and $\mathcal{A}_{FP}$. The algorithms* Gen *and* Trace *may share a common state, which is hidden to ease notation.*

5.2. Lower Bound for Improper PAC Learning

Our lower bounds for multi-learning follow from constructions of fingerprinting codes with additional structural properties.

Definition 42 *Let C be a concept class over a domain X. An (n, k)-fingerprinting code* (Gen, Trace) *is* compatible with concept class C *if there exist $x_1, \ldots, x_n \in X$ such that for every codebook W in the support of* Gen, *there exist concepts $c_1, \ldots, c_k$ such that $w_{ij} = c_j(x_i)$ for every $i = 1, \ldots, n$ and $j = 1, \ldots, k$.*

Theorem 43 *Suppose there exists an (n, k)-fingerprinting code compatible with a concept class C with security ξ. Let $\alpha \leq 1/3$, $\beta, \epsilon > 0$, and $\delta < \frac{1-\xi-\beta}{n} - e^{\epsilon}\xi$. Then every (improper) (α, β)-accurate and (ϵ, δ)-differentially private empirical k-learner for C requires sample complexity greater than n.*

The proof of Theorem 43 follows the ideas sketched above.

Proof Let (Gen, Trace) be an (n, k)-fingerprinting code compatible with the concept class C, and let $x_1, \ldots, x_n \in X$ be its associated universe elements. Let $D = (x_1, \ldots, x_n)$ and let $\mathcal{A}$ be an (α, β)-accurate empirical k-learner for C with sample complexity n. We will use $\mathcal{A}$ to design an adversary $\mathcal{A}_{FP}$ against the fingerprinting code.

Let $T \subseteq [n]$ be a coalition of users, and consider a codebook $W \leftarrow_{\mathrm{R}}$ Gen. The adversary strategy $\mathcal{A}_{FP}(W_T)$ begins by constructing a labeled database $S = (S_i)_{i=1}^n$ by setting $S_i = (x_i, w_{i1}, \ldots, w_{ik})$ for each $i \in T$ and to a nonce row for $i \notin T$. It then runs $\mathcal{A}(S)$ obtaining hypotheses $(h_1, \ldots, h_k)$. Finally, it computes for each $j = 1, \ldots, k$ the averages

$$h_j(D) = \frac{1}{n} \sum_{i=1}^{n} h_j(x_i)$$

and produces a pirate word w' by setting each w'_j to the value of a_j rounded to 0 or 1.

Now consider the coalition $T = [n]$. Since the fingerprinting code is compatible with C, each column $(w_{1j}, \ldots, w_{nj}) = (c_j(x_1), \ldots, c_j(x_n))$ for some concept $c_j \in C$. Thus, if the hypotheses $(h_1, \ldots, h_k)$ are α-accurate for $(c_1, \ldots, c_k)$ on S, then $w' \in F(W_T) = F(W)$. Therefore, by the completeness property of the code and the (α, β)-accuracy of $\mathcal{A}$, we have

$$\Pr[\text{Trace}(\mathcal{A}_{FP}(W)) \neq \perp] \geq 1 - \xi - \beta.$$

In particular, there exists an i^* for which

$$\Pr[\text{Trace}(\mathcal{A}_{FP}(W)) = i^*] \geq \frac{1 - \xi - \beta}{n}.$$

On the other hand, by the soundness property of the code,

$$\Pr[\text{Trace}(\mathcal{A}_{FP}(W_{-i^*})) = i^*] \leq \xi.$$

Thus, $\mathcal{A}$ cannot be (ϵ, δ)-differentially private whenever

$$\frac{1 - \xi - \beta}{n} > e^{\epsilon} \cdot \xi + \delta.$$

■

Remark 44 *If we additionally assume that there exists an element $x_0 \in X$ with $c_1(x_0) = c_2(x_0) = \cdots = c_k(x_0)$, then we can use a "padding" argument to obtain a stronger lower bound of $n/3\alpha$. More specifically, suppose $c_1(x_0) = \cdots = c_k(x_0) = 0$. We pad the database S constructed above with $(1/3\alpha - 1)n$ copies of the junk row $(x_0, 0, \ldots, 0)$. Now if a hypothesis h is α-accurate for a 0-marked column, it's empirical average will be at most α. On the other hand, an α-accurate hypothesis for a 1-marked column will have empirical average at least 2α. Since there is a gap between these two quantities, a pirate algorithm can still turn an accurate vector of k hypotheses into a feasible codeword.*

As observed earlier, the (n, k)-Boneh-Shaw code is compatible with the concept class $\texttt{THRESH}_X$ for any $|X| \geq n$. Thus, instantiating Theorem 43 (and Remark 44) with the Boneh-Shaw code yields a lower bound for k-learning thresholds.

Lemma 45 (Boneh and Shaw (1998)) *Let X be a totally ordered domain with $|X| \geq n$ for some $n \in \mathbb{N}$. Then there exists an (n, k)-fingerprinting code compatible with the concept class $\texttt{THRESH}_X$ with security ξ as long as $k \geq 2n^3 \log(2n/\xi)$.*

Corollary 46 *Every improper (α, β)-accurate and $(\epsilon = O(1), \delta = o(1/n))$-differentially private empirical k-learner for $\texttt{THRESH}_X$ requires sample complexity $\min\{|X|, \tilde{\Omega}(k^{1/3}/\alpha)\}$.*

Discussion. Compatibility with a concept class is an interesting measure of the complexity of a fingerprinting code which warrants further attention. Peikert et al. (2003) showed that structural constraints (related to compatibility) on a fingerprinting code give a lower bound on its length beyond the general lower bound of $k = \tilde{\Omega}(n^2)$ for arbitrary fingerprinting codes. In particular, they showed that the length $k = \tilde{O}(n^3)$ of the Boneh-Shaw code is essentially tight for the "multiplicity paradigm", where a codebook is a random permutation of a fixed set of columns, each repeated the same number of times. We take this as evidence that our $\tilde{\Omega}(k^{1/3})$ lower bound for $\texttt{THRESH}_X$ cannot be improved via compatible fingerprinting codes. However, closing the gap between our lower bound and the upper bound of roughly $\sqrt{k}$ remains an intriguing open question.

A natural avenue for obtaining stronger poly(k) lower bounds for private k-learning is to identify compatible fingerprinting codes with shorter length. Tardos (2008) showed the existence of an (n, k)-fingerprinting code of optimal length $k = \tilde{O}(n^2)$ (see Proposition 49). The construction of his code differs significantly from multiplicity paradigm: for each column j of the Tardos code, a bias $p_j \in (0, 1)$ is sampled from a fixed distribution, and then each bit of the column is sampled i.i.d. with bias p_j. Hence, the columns of the Tardos code are supported on all bit vectors in $\{0, 1\}^n$. This means that for a concept class C to be compatible with the (n, k)-Tardos code, it must be the case that $\mathrm{VC}(C) \geq n$. Thus, the lower bound one obtains against k-learning C only matches the lower bound for PAC learning C (without privacy). It would be very interesting to construct a fingerprinting code of optimal length $k = \tilde{O}(n^2)$ with substantially fewer than 2^n column types (and hence compatible with a concept class of VC-dimension smaller than n).

5.3. Lower Bound for Agnostic Learning

In the agnostic learning model, a learner has to perform well even when the columns of a multi-labeled database do not correspond to any concept. This allows us to apply the

argument of Theorem 43 without the constraint of compatibility. The result is that *any* fingerprinting code, in particular one with optimal length, gives an agnostic learning lower bound for any non-trivial concept class.

Theorem 47 *Suppose there exists an (n, k)-fingerprinting code with security ξ. Let C be a concept class with at least two distinct concepts. Let $\alpha \leq 1/3$, $\beta, \epsilon > 0$, and $\delta < \frac{1-\xi-\beta}{n} - e^{\epsilon}\xi$. Then every (improper) agnostic (α, β)-accurate and (ϵ, δ)-differentially private empirical k-learner for C requires sample complexity greater than n.*

Proof The proof follows in much the same way as that of Theorem 43. Let (Gen, Trace) be an (n, k)-fingerprinting code, and let $x \in X$ be such that there exist $c_0, c_1 \in C$ with $c_0(x) = 0$ and $c_1(x) = 1$. Let $\mathcal{A}$ be an agnostic (α, β)-accurate empirical k-learner for C with sample complexity n. Define a the fingerprinting code adversary $\mathcal{A}_{FP}$ just as in Theorem 43. Namely, $\mathcal{A}_{FP}$ constructs examples of the form $(x, w_{i1}, \ldots, w_{ij})$ with the available rows of the fingerprinting code, runs $\mathcal{A}$ on the result, and returns the rounded empirical averages of the k resulting hypotheses.

To show that $\mathcal{A}$ cannot be (ϵ, δ)-differentially private, it suffices to show that if $\mathcal{A}$ produces accurate hypotheses $h_1, \ldots, h_k$, then the pirate codeword produced by $\mathcal{A}_{FP}$ is feasible. To see this, suppose $h_1, \ldots, h_k$ are accurate, i.e.

$$\max_{1 \leq j \leq k} \left(\mathrm{error}_{S|_j}(h_j) - \min_{c \in C} \left(\mathrm{error}_{S|_j}(c) \right) \right) \leq \alpha.$$

Let column j of the codebook W be 0-marked, i.e. $w_{ij} = 0$ for all $i \in [n]$. Recall that $c_0(x) = 0$, and hence $\mathrm{error}_{S|_j}(c_0) = 0$. Therefore, since hypothesis h_j is α-accurate, we have $\mathrm{error}_{S|_j}(h_j) \leq \alpha$. This implies that bit w'_j of the pirate codeword is 0. An identical argument shows that the bits of the pirate codeword in the 1-marked columns are also 1. Thus, if $\mathcal{A}$ produces accurate hypotheses, the pirate codeword produced by $\mathcal{A}_{FP}$ is feasible. The rest of the argument in the proof of Theorem 43 completes the proof. ■

Remark 48 *Just as in Remark 44, a padding argument shows how to obtain a lower bound of $n/3\alpha$ under some additional assumptions on C, e.g. if the distinct concepts also share a common point x' with $c_0(x') = c_1(x')$.*

Proposition 49 (Tardos (2008)) *For $n \in \mathbb{N}$ and $\xi \in (0, 1)$, there exists an (n, k)-fingerprinting code with security ξ as long as $k = O(n^2 \log(n/\xi))$.*

Corollary 50 *Every improper agnostic (α, β)-accurate and $(\epsilon = O(1), \delta = o(1/n))$-differentially private empirical k-learner for $\texttt{POINT}_X, \texttt{THRESH}_X, \texttt{PAR}_d$ requires sample complexity $\min\{|X|, \tilde{\Omega}(k^{1/2})\}$.*

The same proof yields a lower bound for agnostically learning parities under the uniform distribution.

Proposition 51 *Suppose there exists an (n, k)-fingerprinting code with security ξ. Let $\alpha \leq 1/6, \beta > 0$ and $d = \log n$. Then every (improper) agnostic $(\alpha, \beta, \epsilon = O(1), \delta = o(1/n))$-PAC k-learner for $\texttt{PAR}_d$ requires sample complexity $\Omega(n)$.*

Proof [Proof sketch] By Lemma 28, it is enough to rule out a private empirical learner for a database whose n examples are the distinct binary strings in $\{0,1\}^d$. To do so, we follow the proof of Theorem 47, highlighting the changes that need to be made. First, we let c_0 be the all-zeroes concept, and let c_1 be an arbitrary other parity function. Second, $\mathcal{A}_{FP}$ instead constructs examples of the form $(x_i, w_{i1}, \ldots, w_{ik})$ where x_i is the ith binary string. Finally, when converting the hypotheses $(h_1, \ldots, h_k)$ into a feasible codeword, we instead set w'_j to 0 if $h_j(D) \leq \alpha$, and set w'_j to 1 if $h_j(D) \geq \frac{1}{2} - \alpha$. This works because, while $\text{error}_{S|_j}(c_0) = 0$ with respect to 0-marked columns, any concept (and in particular, c_1) has error $\frac{1}{2}$ with respect to 1-marked columns. ■

6. Examples where Direct Sum is Optimal

In this section we show several examples for cases where the direct sum is (roughly) optimal. As we saw in Section 5, with (ϵ, δ)-differential privacy, every non-trivial *agnostic* k-learner requires sample complexity $\Omega(\sqrt{k})$. We can prove a similar result for ϵ-private learners, that holds even for non-agnostic learners:

Theorem 52 *Let C be any non-trivial concept class over a domain X (i.e., $|C| \geq 2$). Every proper or improper $(\alpha, \beta{=}\frac{1}{2}, \epsilon)$-private PAC k-learner for C requires sample complexity $\Omega(k/\epsilon)$.*

Beimel et al. (2014, 2013a,b) presented an agnostic proper learner for $\texttt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(1)$ under (ϵ, δ)-privacy, and an agnostic improper learner for $\texttt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(1)$ under ϵ-privacy. Hence, using Observation 32 (direct sum) with their results yields an $(\alpha, \beta, \epsilon, \delta)$-PAC agnostic proper k-learner for $\texttt{POINT}_X$ with sample complexity $\tilde{O}_{\alpha,\beta,\epsilon,\delta}(\sqrt{k})$, and an $(\alpha, \beta, \epsilon)$-PAC agnostic improper k-learner for $\texttt{POINT}_X$ with sample complexity $\tilde{O}_{\alpha,\beta,\epsilon}(k)$. As supported by our lower bounds (Corollary 50 and Theorem 52), those learners have roughly optimal sample complexity (ignoring the dependency in $\alpha, \beta, \epsilon, \delta$ and logarithmic factors in k).

Proof [Proof of Theorem 52] The proof is based on a packing argument (Hardt and Talwar, 2010; Beimel et al., 2014). Let $x \in X$ and $f, g \in C$ be s.t. $f(x) \neq g(x)$. Let μ denote the constant distribution over X giving probability 1 to the point x. Note that $\text{error}_\mu(f, g) = 1$. Moreover, observe that for every concept h, if $\text{error}_\mu(h, f) < 1$ then $h(x) = f(x)$, and similarly with h, g.

Let $\mathcal{A}$ be an $(\alpha, \beta, \epsilon)$-private PAC k-learner for C with sample complexity n. For every choice of k target functions $(c_1, \ldots, c_k) = \vec{c} \in \{f, g\}^k$, let $S_{\vec{c}}$ denote the k-labeled database containing n copies of the point x, each of which is labeled by $c_1, \ldots, c_k$. Without loss of generality, we can assume that on such databases $\mathcal{A}$ returns hypotheses in $\{f, g\}$ (since under μ we can replace an arbitrarily chosen hypothesis h with f if $f(x) = h(x)$ or with g if $g(x) = h(x)$). Therefore, by the utility properties of $\mathcal{A}$, for every $\vec{c} = (c_1, \ldots, c_k) \in \{f, g\}^k$ we have that $\Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = (c_1, \ldots, c_k)] \geq \frac{1}{2}$. By changing the database $S_{\vec{c}}$ to $S_{\vec{c}'}$ one row at a time while applying the differential privacy constraint, we see that

$$\Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = (c'_1, \ldots, c'_k)] \geq \frac{1}{2} e^{-\epsilon n}.$$

Since the above inequality holds for every two databases $S_{\vec{c}}$ and $S_{\vec{c}'}$, we get

$$\frac{1}{2} \geq \Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) \neq (c_1, \ldots, c_k)] \geq (2^k - 1)\frac{1}{2} e^{-\epsilon n}.$$

Solving for n, this yields $n = \Omega(k/\epsilon)$. ■

Remark 53 *The above proof could easily be strengthened to show that $n = \Omega(\frac{k}{\alpha\epsilon})$, provided that C contains two concepts f, g s.t. $\exists x, y \in X$ for which $f(x) \neq g(x)$ and $f(y) = g(y)$.*

The following lemma shows that the sample complexities of properly and improperly learning parities under the uniform distribution are the same. Thus, for showing lower bounds, it is without loss of generality to consider proper learners.

Lemma 54 *Let $\alpha < 1/4$. Let $\mathcal{A}$ be a (possibly improper) $(\alpha, \beta, \epsilon, \delta)$-PAC k-learner for $\texttt{PAR}_d$ under the uniform distribution with sample complexity n. Then there exists a proper $(\alpha' = 0, \beta, \epsilon, \delta)$-PAC k-learner $\mathcal{A}'$ for $\texttt{PAR}_d$ (under the uniform distribution) with sample complexity n.*

Proof The algorithm $\mathcal{A}'$ runs $\mathcal{A}$ and "rounds" each hypothesis h_j produced to the nearest parity function. That is, it outputs $(h'_1, \ldots, h'_k)$ where h'_j is a parity function that minimizes $\Pr_{x \sim U_d}[h'_j(x) \neq h_j(x)]$. Since this is just post-processing of the differentially private algorithm $\mathcal{A}$, the proper learner $\mathcal{A}$ remains (ϵ, δ)-differentially private.

Now suppose $(h_1, \ldots, h_k)$ is α-accurate for parity functions $(c_1, \ldots, c_k)$ on the uniform distribution. Then for each j,

$$\begin{aligned}\Pr_{x \sim U_d}[h'_j(x) \neq c_j(x)] &\leq \Pr_{x \sim U_d}[h'_j(x) \neq h_j(x)] + \Pr_{x \sim U_d}[h_j(x) \neq c_j(x)] \\ &\leq 2 \Pr_{x \sim U_d}[h_j(x) \neq c_j(x)] \\ &\leq 2\alpha.\end{aligned}$$

Hence, $\text{error}_{U_d}(h'_j, c_j) < 1/2$. Since the error of any parity function from c_j (other than c_j itself) is exactly $1/2$ under the uniform distribution, we conclude that $(h'_1, \ldots, h'_k)$ is in fact 0-accurate for $(c_1, \ldots, c_k)$. ■

Theorem 55 *Let $\alpha < \frac{1}{4}$. Every $(\alpha, \beta{=}\frac{1}{2}, \epsilon)$-PAC k-learner for $\texttt{PAR}_d$ (under the uniform distribution) requires sample complexity $\Omega(kd/\epsilon)$.*

As we saw in Example 1, applying direct sum for k-learning parities results in a proper agnostic $(\alpha, \beta, \epsilon)$-PAC k-learner for $\texttt{PAR}_d$ with sample complexity $O_{\alpha,\beta,\epsilon}(kd + k \log k)$. As stated by Theorem 55, this is the best possible (ignoring logarithmic factors and the dependency in α, β, ϵ).

Proof [Proof of Theorem 55] The proof is based on a packing argument (Hardt and Talwar, 2010; Beimel et al., 2014). Let $\mathcal{A}$ be an $(\alpha, \beta, \epsilon)$-PAC k-learner for $\texttt{PAR}_d$ with sample

complexity n. By Lemma 54, we may assume $\mathcal{A}$ is proper and learns the hidden concepts exactly.

For every choice of k parity functions $(c_1, \ldots, c_k) = \vec{c} \in (\mathtt{PAR}_d)^k$, let $S_{\vec{c}}$ denote a random k-labeled database containing n i.i.d. elements from U_d, each labeled by $(c_1, \ldots, c_k)$. By the utility properties of $\mathcal{A}$ we have that $\Pr_{U_d,\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = \vec{c}] \geq \frac{1}{2}$. In particular, for every $\vec{c} \in (\mathtt{PAR}_d)^k$ there exists a database $D_{\vec{c}}$ labeled by $\vec{c}$ s.t. $\Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = \vec{c}] \geq \frac{1}{2}$. By changing the database $D_{\vec{c}}$ to $D_{\vec{c'}}$ one row at a time while applying the differential privacy constraint, we see that

$$\Pr_{\mathcal{A}}[\mathcal{A}(D_{\vec{c}}) = \vec{c'}] \geq \frac{1}{2}e^{-\epsilon n}.$$

Since the above inequality holds for every two databases $D_{\vec{c}}$ and $D_{\vec{c'}}$, we get

$$\frac{1}{2} \geq \Pr_{\mathcal{A}}[\mathcal{A}(D_{\vec{c}}) \neq \vec{c}] \geq (|\,\mathtt{PAR}_d\,|^k - 1)\frac{1}{2}e^{-\epsilon n}.$$

Solving for n, this yields $n = \Omega(kd/\epsilon)$. ■

Acknowledgments

This work was done while Mark Bun was at the John A. Paulson School of Engineering & Applied Sciences, Harvard University and visiting Yale University, supported by an NDSEG fellowship and NSF grant CNS-1237235. Kobbi Nissim and Uri Stemmer were supported by NSF grant no. 1565387. TWC: Large: Collaborative: Computing Over Distributed Sensitive Data.

References

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. *CoRR*, abs/1806.00949, 2018. URL `http://arxiv.org/abs/1806.00949`.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, New York, NY, USA, 1st edition, 2009. ISBN 052111862X, 9780521118620.

Martin Anthony and John Shawe-Taylor. A result of Vapnik with applications. *Discrete Applied Mathematics*, 47(3):207–217, 1993.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014. doi: 10.1109/FOCS.2014.56. URL `http://dx.doi.org/10.1109/FOCS.2014.56`.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *ITCS*, pages 97–110, 2013a.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, pages 363–378, 2013b.

Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014. doi: 10.1007/s10994-013-5404-1. URL `http://dx.doi.org/10.1007/s10994-013-5404-1`.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. In *SODA*, pages 461–477, 2015. doi: 10.1137/1.9781611973730.32. URL `http://dx.doi.org/10.1137/1.9781611973730.32`.

Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. *CoRR*, abs/1902.10731, 2019. URL `http://arxiv.org/abs/1902.10731`.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *PODS*, pages 128–138, 2005.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *J. ACM*, 60(2):12, 2013. doi: 10.1145/2450142.2450148. URL `http://doi.acm.org/10.1145/2450142.2450148`.

Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, October 1989. ISSN 0004-5411. doi: 10.1145/76359.76371. URL `http://doi.acm.org/10.1145/76359.76371`.

Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, 1998.

Mark Bun, Jonathan Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, pages 1–10, 2014. doi: 10.1145/2591796.2591877. URL `http://doi.acm.org/10.1145/2591796.2591877`.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 634–649, Berkeley, CA, USA, October 18-20, 2015.

Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In Sham M. Kakade and Ulrike von Luxburg, editors, *COLT*, volume 19 of *JMLR Proceedings*, pages 155–186. JMLR.org, 2011.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 371–380, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536466. URL `http://doi.acm.org/10.1145/1536414.1536466`.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006b.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: Optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 11–20, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2710-7. doi: 10.1145/2591796.2591883. URL `http://doi.acm.org/10.1145/2591796.2591883`.

Dan Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim. Private coresets. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 361–370. ACM, 2009. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536465. URL `http://doi.acm.org/10.1145/1536414.1536465`.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *COLT*, pages 1000–1019, 2014. URL `http://jmlr.org/proceedings/papers/v35/feldman14b.html`.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, 2010.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714, 2010.

Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Differentially private learning of geometric concepts. *CoRR*, abs/1902.05017, 2019. URL `http://arxiv.org/abs/1902.05017`.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011. doi: 10.1137/090756090. URL `http://dx.doi.org/10.1137/090756090`.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-3010-9. doi: 10.1109/FOCS.2007.41. URL `http://dx.doi.org/10.1109/FOCS.2007.41`.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 75–84, 2007.

Chris Peikert, abhi shelat, and Adam Smith. Lower bounds for collusion-secure fingerprinting. In *SODA*, pages 472–479, 2003. URL `http://dl.acm.org/citation.cfm?id=644108.644187`.

Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *STOC*, pages 765–774, 2010.

Adam Smith and Abhradeep Thakurta. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *COLT 2013*, pages 819–850, 2013. URL `http://jmlr.org/proceedings/papers/v30/Guha13.html`.

Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. In *TPDP 2015*, 2015.

Gábor Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL `http://doi.acm.org/10.1145/1968.1972`.

Leslie G. Valiant. Knowledge infusion. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1546–1551. AAAI Press, 2006. URL `http://www.aaai.org/Library/AAAI/2006/aaai06-247.php`.

Journal of Machine Learning Research 20 (2019) 1-41 Submitted 12/18; Revised 3/19; Published 5/19

High-Dimensional Poisson Structural Equation Model Learning via ℓ_1-Regularized Regression

Gunwoong Park GW.PARK23@GMAIL.COM
Department of Statistics
University of Seoul
Seoul, 02504, South Korea

Sion Park ROCKGOAT95@GMAIL.COM
Department of Statistics
University of Seoul
Seoul, 02504, South Korea

Editor: Qiang Liu

Abstract

In this paper, we develop a new approach to learning high-dimensional Poisson structural equation models from only observational data without strong assumptions such as faithfulness and a sparse moralized graph. A key component of our method is to decouple the ordering estimation or parent search where the problems can be efficiently addressed using ℓ_1-regularized regression and the moments relation. We show that sample size $n = \Omega(d^2 \log^9 p)$ is sufficient for our polynomial time Moments Ratio Scoring (MRS) algorithm to recover the true directed graph, where p is the number of nodes and d is the maximum indegree. We verify through simulations that our algorithm is statistically consistent in the high-dimensional $p > n$ setting, and performs well compared to state-of-the-art ODS, GES, and MMHC algorithms. We also demonstrate through multivariate real count data that our MRS algorithm is well-suited to estimating DAG models for multivariate count data in comparison to other methods used for discrete data.

Keywords: Bayesian Networks, Directed Acyclic Graph, Identifiability, Structure Learning, ℓ_1-Regularization, Multivariate Count Distribution

1. Introduction

Directed acyclic graphical (DAG) models, also referred to as Bayesian networks, are popular probabilistic statistical models to analyze and visualize (functional) causal or directional dependence relationships among random variables.(see e.g., Kephart and White, 1991; Friedman et al., 2000; Doya, 2007; Peters and Bühlmann, 2014). However, learning DAG models from only observational data is a notoriously difficult problem due to non-identifiability and exponentially growing computational complexity. Prior works have addressed the question of identifiability for different classes of joint distribution $P(G)$. Frydenberg (1990) and Heckerman et al. (1995) show the Markov equivalence class (MEC) where graphs that belong to the same MEC have the same conditional independence relations. Spirtes et al. (2000), Chickering (2003), Tsamardinos and Aliferis (2003) and Zhang and Spirtes (2016) show that the underlying graph of a DAG model is recoverable up to the MEC under faithfulness or related assumptions that can be very restrictive (Uhler et al., 2013).

©2019 Gunwoong Park and Sion Park.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-819.html.

Also well studied is how learning a DAG model is computationally non-trivial due to the super-exponentially growing size of the space of DAGs in the number of nodes. Hence, it is NP-hard to search DAG space (Chickering et al., 1994; Chickering, 1996), and many existing algorithms such as PC (Spirtes et al., 2000), Greedy Equivalence Search (GES) (Chickering, 2003), Min-Max Hill Climbing (MMHC) (Tsamardinos et al., 2006) and Greedy DAG Search (GDS) (Peters and Bühlmann, 2014), take greedy search methods that may not guarantee to recover the true MEC.

Recently, a number of fully identifiable classes of DAG models have been introduced (Shimizu et al., 2006; Hoyer et al., 2009; Peters et al., 2011; Peters and Bühlmann, 2014; Park and Raskutti, 2015, 2018; Ghoshal and Honorio, 2018; Park and Park, 2019). In addition, some of these models can be successfully learned from high-dimensional data by decomposing the DAG learning problem into ordering estimation and skeleton estimation (Shimizu et al., 2011; Bühlmann et al., 2014; Ghoshal and Honorio, 2017b; Drton et al., 2018). The main reasoning is that if ordering is known or recoverable, learning a directed graphical model is as hard as learning an undirected graphical model or Markov random field (MRF). Meinshausen and Bühlmann (2006), Wainwright et al. (2006), Ravikumar et al. (2011) and Yang et al. (2015) show that sparse undirected graphs can be estimated via ℓ_1-regularized regression in high-dimensional settings under suitable conditions.

In this paper, we focus on learning Poisson DAG models (Park and Raskutti, 2015, 2018) for multivariate count data in high-dimensional settings since large-scale multivariate *count data* frequently arises in many fields, such as high-throughput genomic sequencing data, spatial incidence data, sports science data, and disease incidence data. Like learning the Poisson undirected graphical model or MRF introduced in Yang et al. (2015), where the sample bound is $\Omega(d_m^2 \log^3 p)$, it is not surprising that Poisson DAG models can be learned in high dimensional settings when the indegree of the graph d is bounded. Park and Raskutti (2018) establishes the consistency of learning Poisson DAG models with the sample bound $n = \Omega(\max\{d_m^4 \log^{12} p, \log^{5+d} p\})$ where d_m is the maximum degree of the moralized graph and d is the maximum indegree of a graph. This huge sample complexity difference between directed and undirected graphical models is induced mainly for three reasons: (i) nonexistence ordering, (ii) the known parametric functional form (the standard log link) for the dependencies, and (iii) the restrictive non-positive parameter space in Poisson MRFs (see details in Yang et al., 2015).

The main objective of this work is to propose a new milder identifiability assumption for Poisson DAG models, and to develop a new polynomial time approach, called Moments Ratio Scoring (MRS), for learning a high-dimensional Poisson structural equation models (SEM), that is a Poisson DAG model where the parametric functional form for the dependencies is known while the parameters are unbounded and unknown. We address the question of learning high-dimensional Poisson SEMs under the causal sufficiency assumption that all relevant variables have been observed. However, we do not require the sparse moralized graph and faithfulness assumption that might be restrictive (Uhler et al., 2013).

The MRS algorithm combines the idea of the mean-variance (moments) relation for recovering an ordering, and the sparsity-encouraging ℓ_1-regularized regression in finding the parents of each node. We provide its sufficient conditions and sample complexity $n = \Omega(d^2 \log^9 p)$ under which the MRS algorithm recovers the Poisson SEM with a high probability in the high-dimensional $p > n$ setting. The sample complexity of $n = \Omega(d^2 \log^9 p)$ is

close to the information-theoretic limit of $\Omega(d \log p)$ for learning sparse DAG models with any exponential family distributions (Ghoshal and Honorio, 2017a). We point out that the sample complexity does not depend on the maximum degree of the moralized graph, d_m, but on the indegree of a DAG, d. Since a sparse directed graph does not necessarily lead to the sparse moralized graph (e.g., a star graph in Fig. 2), to the best of our knowledge, the proposed algorithm is the most efficient and probable for learning sparse Poisson SEMs.

We demonstrate through simulations and a real baseball data application involving multivariate count data that our MRS algorithm performs better than state-of-the-art OverDispersion Scoring (ODS) (Park and Raskutti, 2015), GES (Chickering, 2003), MMHC (Tsamardinos et al., 2006), and Poisson MRF learning (PMRF) algorithms (Yang et al., 2015), on average, in terms of the both run-time and accuracy of recovering a graph structure and its MEC. In our simulation study, we consider both the extremely sparse ($d = 1$) and sparse ($d = 10$) high-dimensional settings. Our real data example involving MLB player statistics for 2003 season shows that our MRS algorithm is applicable to multivariate count data while the PMRF algorithm finds too many edges, and the MMHC algorithm tends to select very few edges when variables represent counts. We also investigate the accuracy of our MRS algorithm when samples are generated from general Poisson DAG models and (truncated) Poisson MRFs. The simulation results empirically verify that the MRS algorithm can consistently recover the true edges.

1.1. Our Contributions

We summarize the major contributions of the paper as follows:

- We introduce a milder identifiability condition for Poisson DAG models for multivariate count data.
- We develop the reliable and scalable lasso-based MRS algorithm which learns sparse high-dimensional Poisson SEMs.
- We provide the more realistic conditions for learning Poisson SEMs in Section 3.2.
- We also provide the sample complexity $n = \Omega(d^2 \log^9 p)$ under which the MRS algorithm recovers the Poisson SEM. We emphasize that our theoretical result does not depend on the degree of the moralized graph d_m, and hence, the MRS algorithm can recover a graph with hub nodes in the high dimensional setting.

To the best of our knowledge, our MRS algorithm is the only provable and realistic method that applies for the high-dimensional multivariate count data when samples are from Poisson SEMs with hub nodes. We must point out that such improved assumptions and sample complexity are not only from our new identifiability condition, but from the additional constraints on the standard log link function for the dependencies.

The remainder of this paper is structured as follows. Section 2.1 summarizes the necessary notations and problem settings, Section 2.2 discusses the Poisson DAG model and its new identifiability condition, and Section 2.3 provides a detailed comparison between Poisson DAG models and MRFs. In Section 3, we introduce our polynomial-time DAG learning algorithm, which we refer to as the Moments Ratio Scoring (MRS). Section 3.1 discusses

computational complexity of our algorithm, and Section 3.2 provides statistical guarantees for learning Poisson SEMs via the MRS algorithm. Section 4 empirically evaluates our methods, compared to state-of-the-art ODS, GES, and MMHC algorithms using synthetic data, and confirms that our algorithm is one of the few DAG-learning algorithms that performs well in terms of statistical and computational complexity in low and high-dimensional settings. In addition, we investigate how well the MRS algorithm learns general Poisson DAG models and (truncated) Poisson MRFs using synthetic data. Section 5 compares our MRS algorithm to the Poisson MRF and MMHC algorithm by analyzing a real 2003 season MLB multivariate count data. Lastly, Section 6 discusses some future works.

2. Poisson DAG Models

We first introduce some necessary notations and definitions for DAG models. Then, we give a detailed description of previous work on learning Poisson DAG models (Park and Raskutti, 2015), and we propose a strictly milder identifiability condition. Lastly, we discuss how Poisson DAG models and MRFs (Yang et al., 2015) are related.

2.1. Problem Set-up and Notations

A DAG $G = (V, E)$ consists of a set of nodes $V = \{1, 2, \cdots, p\}$ and a set of directed edges $E \subset V \times V$ with no directed cycles. A directed edge from node j to k is denoted by (j, k) or $j \to k$. The set of *parents* of node k, denoted by $\mathrm{Pa}(k)$, consists of all nodes j such that $(j, k) \in E$. If there is a directed path $j \to \cdots \to k$, then k is called a *descendant* of j, and j is an *ancestor* of k. The set $\mathrm{De}(k)$ denotes the set of all descendants of node k. The *non-descendants* of node k are $\mathrm{Nd}(k) := V \setminus (\{k\} \cup \mathrm{De}(k))$. An important property of DAGs is that there exists a (possibly non-unique) *ordering* $\pi = (\pi_1, ..., \pi_p)$ of a directed graph that represents directions of edges such that for every directed edge $(j, k) \in E$, j comes before k in the ordering. Hence, learning a graph is equivalent to learning the ordering and the skeleton that is the set of directed edges without their directions.

We consider a set of random variables $X := (X_j)_{j \in V}$ with a probability distribution taking values in a sample space $\mathcal{X}_V$ over the nodes in G. Suppose that a random vector X has a joint probability density function $P(G) = P(X_1, X_2, ..., X_p)$. For any subset S of V, let $X_S := \{X_j : j \in S \subset V\}$ and $\mathcal{X}_S := \times_{j \in S} \mathcal{X}_j$ where $\mathcal{X}_j$ is a sample space of X_j. For any node $j \in V$, $P(X_j \mid X_S)$ denotes the conditional distribution of a variable X_j given a random vector X_S. Then, a DAG model has the following factorization (Lauritzen, 1996):

$$P(G) = P(X_1, X_2, ..., X_p) = \prod_{j=1}^{p} P(X_j \mid X_{\mathrm{Pa}(j)}), \tag{1}$$

where $P(X_j \mid X_{\mathrm{Pa}(j)})$ is the conditional distribution of X_j given its parents variables $X_{\mathrm{Pa}(j)} := \{X_k : k \in \mathrm{Pa}(j) \subset V\}$.

We suppose that there are n independent and identically distributed samples $X^{1:n} := (X^{(i)})_{i=1}^n$ from a given graphical model where $X^{(i)} := X_{1:p}^{(i)} = (X_1^{(i)}, X_2^{(i)}, \cdots, X_p^{(i)})$ is a p-variate random vector. The notation $\widehat{\cdot}$ denotes an estimate based on samples $X^{1:n}$. We also accept the causal sufficiency assumption that all important variables have been observed.

2.2. Poisson DAG Model and its Identifiability

The definition of Poisson DAG models in Park and Raskutti (2015) is that each conditional distribution given its parents $X_j \mid X_{\mathrm{Pa}(j)}$ is Poisson such that

$$X_j \mid X_{\mathrm{Pa}(j)} \sim \mathrm{Poisson}(g_j(X_{\mathrm{Pa}(j)})), \tag{2}$$

where for any arbitrary positive link function $g_j : \mathcal{X}_{\mathrm{Pa}(j)} \to \mathbb{R}^+$. Hence using the factorization in Equation (1), the joint distribution is as follows:

$$f_G(X) = \prod_{j \in V} f_j(X_j \mid X_{\mathrm{Pa}(j)}). \tag{3}$$

where f_j is the probability density function of Poisson.

A Poisson structural equation model (SEM) is a special case of a Poisson DAG model where the link functions g_j's in Equation (2) are the standard log link function for Poisson generalized linear models (GLMs), i.e., $g_j(X_{\mathrm{Pa}(j)}) = \exp(\theta_j + \sum_{k \in \mathrm{Pa}(j)} \theta_{jk} X_k)$ where $(\theta_{jk})_{k \in \mathrm{Pa}(j)}$ represents the linear weights. Using factorization (1), the joint distribution of a Poisson SEM can be written as:

$$f(X_1, X_2, ..., X_p) = \exp\Big(\sum_{j \in V} \theta_j X_j + \sum_{(k,j) \in E} \theta_{jk} X_j X_k - \sum_{j \in V} \log X_j! - \sum_{j \in V} e^{\theta_j + \sum_{k \in \mathrm{Pa}(j)} \theta_{jk} X_k} \Big). \tag{4}$$

Poisson DAG models have a useful moments relation for the identifiability:

Proposition 1 *Consider a Poisson DAG model* (3) *with non-degenerated rate parameter functions* $(g_j(X_{Pa(j)}))_{j \in V}$. *Then, for any node* $j \in V$, *and any set* $S_j \subset Nd(j)$, *the following moments relation holds:*

$$\frac{\mathbb{E}(X_j^2)}{\mathbb{E}\left[\mathbb{E}(X_j \mid X_{S_j}) + \mathbb{E}(X_j \mid X_{S_j})^2\right]} \geq 1 \tag{5}$$

Equivalently,

$$\mathbb{E}(\mathit{Var}(\mathbb{E}(X_j \mid X_{Pa(j)}) \mid X_{S_j})) \geq 0.$$

The equality only holds when S_j *contains all parents of* j, *that is,* $Pa(j) \subset S_j$.

We include the proof in Section A. Proposition 1 claims that when all parents of j, $\mathrm{Pa}(j)$, contribute to its rate parameter, the moments ratio in Equation (5) is equal to 1 if a condition set S_j contains all parents of j, $\mathrm{Pa}(j) \subset S_j$, otherwise greater than 1. In Poisson SEMs, it is clear that the non-degenerated rate parameter function assumptions are equivalent to the *non-zero coefficients* conditions, $|\theta_{jk}| > 0$ for all $k \in \mathrm{Pa}(j)$ since $g_j(X_{\mathrm{Pa}(j)}) = \exp(\theta_j + \sum_{k \in \mathrm{Pa}(j)} \theta_{jk} X_k)$.

Now, we briefly explain how Poisson DAG models are identifiable from the moments ratio in Proposition 1 using the bivariate Poisson DAG models illustrated in Fig. 1: $G_1 : X_1 \sim \mathrm{Poisson}(\lambda_1), X_2 \sim \mathrm{Poisson}(\lambda_2)$, where X_1 and X_2 are independent; $G_2 : X_1 \sim \mathrm{Poisson}(\lambda_1)$ and $X_2 \mid X_1 \sim \mathrm{Poisson}(g_2(X_1))$; and $G_3 : X_2 \sim \mathrm{Poisson}(\lambda_2)$ and $X_1 \mid X_2 \sim \mathrm{Poisson}(g_1(X_2))$ for arbitrary non-degenerated positive functions $g_1, g_2 : \mathbb{N} \cup \{0\} \to \mathbb{R}^+$.

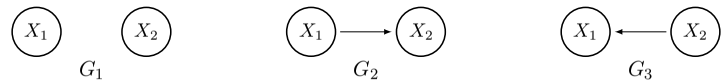


Figure 1: Bivariate directed acyclic graphs of G_1, G_2, and G_3.

By Proposition 1, we can see that $\mathbb{E}(X_j^2) = \mathbb{E}(X_j) + \mathbb{E}(X_j)^2$ for all $j \in \{1, 2\}$ in G_1. In G_2, we can also see that

$$\mathbb{E}(X_1^2) = \mathbb{E}(X_1) + \mathbb{E}(X_1)^2, \quad \text{and} \quad \mathbb{E}(X_2^2) > \mathbb{E}(X_2) + \mathbb{E}(X_2)^2.$$

Similarly, in G_3, we have $\mathbb{E}(X_1^2) > \mathbb{E}(X_1) + \mathbb{E}(X_1)^2$, while $\mathbb{E}(X_2^2) = \mathbb{E}(X_2) + \mathbb{E}(X_2)^2$. Hence, we can determine the true graph based on the moments ratio $\mathbb{E}(X_j^2)/(\mathbb{E}(X_j) + \mathbb{E}(X_j)^2)$.

This idea of a moments relation in Proposition 1 can easily apply to general p-variate Poisson DAG models, and hence, the models are identifiable by testing whether the moments ratio in Equation (5) is equal to 1 or greater than 1.

Theorem 2 *Consider a Poisson DAG model* (3) *with rate parameters* $(g_j(X_{Pa(j)}))_{j \in V}$. *If for any* $j \in V$, *rate parameter* $g_j(\cdot)$ *is non-degenerated, the Poisson DAG model is identifiable.*

We include the proof in Section 3.2. Theorem 2 claims that any Poisson DAG model is identifiable if all parents of node j contribute to its rate parameter. Hence, Theorem 2 shows that any Poisson SEM is identifiable under the non-zero coefficients condition, $|\theta_{jk}| > 0$ for all $k \in \mathrm{Pa}(j)$. This condition is also commonly assumed in (Gaussian) linear structural equation models for the model identifiability (Spirtes, 1995; Ghoshal and Honorio, 2017b, 2018; Loh and Bühlmann, 2014; Peters and Bühlmann, 2014; Park and Park, 2019). We believe that it is a natural condition that is in accordance with the intuitive understanding of relationships among variables.

Our identifiability condition is strictly milder than the previous identifiability result in Park and Raskutti (2015) that is equivalent to $\mathrm{Var}(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)}) \mid X_{S_j} = x) > 0$ for all $x \in \mathcal{X}_{S_j}$ when $\mathrm{Pa}(j) \not\subset S_j$. For a better comparison, we consider a fully connected graph where $X_1 \sim \mathrm{Poisson}(\lambda)$, $X_2 \mid X_1 \sim \mathrm{Poisson}(\lambda + X_1)$, and $X_3 \mid X_1, X_2 \sim \mathrm{Poisson}(\lambda + X_2 \mathbf{1}(X_1 \neq 0))$ where λ is a positive constant and $\mathbf{1}(\cdot)$ is an indicator function. In this case, we can see $\mathrm{Var}(\mathbb{E}(X_3 \mid X_1, X_2) \mid X_1 = 0) = 0$, and hence, the identifiability condition in Park and Raskutti (2015) is not satisfied, while our condition is satisfied.

In a Poisson SEM, the identifiability assumption in Park and Raskutti (2015) is also satisfied under the non-zero coefficients condition. However, in the finite sample setting, the difference of both assumptions gets more crucial. For a positive constant c, Park and Raskutti (2015) requires $\min_{x \in \mathcal{X}_{S_j}} \mathrm{Var}(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)}) \mid X_{S_j} = x)) > c$, while we need $\mathbb{E}(\mathrm{Var}(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)}) \mid X_{S_j})) > c$. Hence, our new identifiability assumption makes learning Poisson SEMs easier. We discuss this more in Section 3.2.

2.3. Comparison to Poisson MRF

In this section, we compare Poisson DAG models and MRFs where the conditional distributions of each node given its parents and neighbors are Poisson, respectively. To simplify

the comparison, we consider the joint distribution of a Poisson SEM in Equation (4). This is a form similar to the joint distribution of Poisson MRFs in Yang et al. (2015), where the joint distribution has the following form:

$$f(X_1, X_2, ..., X_p) = \exp\Big(\sum_{j\in V}\theta_j X_j + \sum_{(k,j)\in E}\theta_{jk}X_jX_k - \sum_{j\in V}\log X_j! - A(\theta)\Big), \tag{6}$$

where $A(\theta)$ is the log of the normalization constant. The key difference between a Poisson SEM and a Poisson MRF is the normalization constant $A(\theta)$ in Equation (6), as opposed to the term $\sum_{j\in V} e^{\theta_j + \sum_{k\in \mathrm{Pa}(j)}\theta_{jk}X_k}$ in Equation (4), which depends on variables.

Yang et al. (2015) proves that a Poisson MRF (6) is normalizable if and only if all (θ_{jk}) values are less than or equal to 0. This means Poisson MRFs only capture negative dependency relations. In addition, Yang et al. (2015) addresses the learning Poisson MRFs when the functional form of dependencies is $X_j \mid X_{V\setminus j} \sim \mathrm{Poisson}(\exp(\theta_j + \sum_{k\in\mathcal{N}(j)}\theta_{jk}X_k))$ where $\mathcal{N}(j)$ denotes the neighbors of a node j in the graph.

While Poisson MRFs have strong restrictions on the functional form for dependencies and the parameter space, they can be successfully learned in the high-dimensional settings with less restrictive constraints of sparsity. Yang et al. (2015) shows that Poisson MRFs can be recovered via ℓ_1-regularized regression if $n = \Omega\left(d_m^2 \log^3 p\right)$, where d_m is the degree of the undirected graph. In contrast, Park and Raskutti (2018) shows that Poisson DAG models can be learned via the ODS algorithm if $n = \Omega(\max\{d_m^4 \log^{12} p, \log^{5+d} p\})$ where d_m is obtained by the moralized graph and d is the maximum indegree of the graph. This big difference in the sample complexity primarily comes from the unknown functional form for the dependencies in Poisson DAG models. In the next section, we will show that a significant advantage can be achieved by assuming the parametric function for the dependencies in terms of recovering the graphs.

3. Algorithm

Here, we present our Moments Ratio Scoring (MRS) algorithm for learning the identifiable Poisson SEM (4). Our algorithm alternates between an element-wise ordering search using the (conditional) moments ratio, and a parent search using ℓ_1-regularized GLM. Hence, the algorithm chooses a node for the first element of the ordering, and then determines its parents. The algorithm iterates this procedure until the last element of the ordering and its parents are determined.

Without loss of generality, assume that $\pi = (1, 2, \cdots, p)$ is the true ordering. Then Poisson SEMs (4) have the conditional distribution of X_j given that all variables before j in the ordering are reduced to the following Poisson GLM:

$$P(X_j \mid X_{1:(j-1)}) = \exp\Big\{\theta_j X_j + \sum_{k\in 1:(j-1)}\theta_{jk}X_kX_j + \log X_j! - \exp\Big(\theta_j + \sum_{k\in 1:(j-1)}\theta_{jk}X_k\Big)\Big\}, \tag{7}$$

where $\theta_{jk} \in \mathbb{R}$ represents the influence of node k on node j. For ease of notation, let $\theta(j)$ be a set of parameters related to Poisson GLM (7). Then $\theta(j) = (\theta_j, \theta_{\setminus j}) \in \mathbb{R}\times\mathbb{R}^{j-1}$ where $\theta_{\setminus j} = (\theta_{jk})_{k\in\{1,2,...,j-1\}}$ is a zero-padded vector with non-zero entries if $k \in \mathrm{Pa}(j)$.

Our MRS (Algorithm 1) involves learning the ordering by comparing moments ratio scores of nodes using the following equations:

$$\widehat{\mathcal{S}}(1,j) := \frac{\widehat{\mathbb{E}}(X_j^2)}{\widehat{\mathbb{E}}(X_j) + \widehat{\mathbb{E}}(X_j)^2} \quad \text{and} \quad \widehat{\mathcal{S}}(m,j) := \frac{\widehat{\mathbb{E}}(X_j^2)}{\widehat{\mathbb{E}}\big(\widehat{\mathbb{E}}(X_j \mid X_{\widehat{\pi}_{1:(m-1)}}) + \widehat{\mathbb{E}}(X_j \mid X_{\widehat{\pi}_{1:(m-1)}})^2\big)}, \tag{8}$$

where $\widehat{\pi}_{1:m} = \{\widehat{\pi}_1, ..., \widehat{\pi}_m\}$, $\widehat{\mathbb{E}}(X_j) = \frac{1}{n}\sum_{i=1}^n X_j^{(i)}$, and $\widehat{\mathbb{E}}(\widehat{\mathbb{E}}(X_j \mid X_S)) = \frac{1}{n}\sum_{i=1}^n \exp\big(\widehat{\theta}_j^S + \sum_{k\in S}\widehat{\theta}_{jk}^S X_k^{(i)}\big)$, and $\widehat{\mathbb{E}}(\widehat{\mathbb{E}}(X_j \mid X_S)^2) = \frac{1}{n}\sum_{i=1}^n \exp\big(2\widehat{\theta}_j^S + 2\sum_{k\in S}\widehat{\theta}_{jk}^S X_k^{(i)}\big)$ where $\widehat{\theta}_S(j) = (\widehat{\theta}_j^S, \widehat{\theta}_{\setminus j}^S)$ is the solution of the following ℓ_1-regularized GLM:

$$\widehat{\theta}_S(j) := \arg\min \frac{1}{n}\sum_{i=1}^n \Big[-X_j^{(i)}\Big(\theta_j + \sum_{k\in S}\theta_{jk}X_k^{(i)}\Big) + \exp\Big(\theta_j + \sum_{k\in S}\theta_{jk}X_k^{(i)}\Big)\Big] + \lambda_j\sum_{k\in S}|\theta_{jk}|. \tag{9}$$

This score is an estimator of the moments ratio relation in Equation (5). Hence, the correct element of the ordering has a score of 1, otherwise strictly greater than 1 in population. The ordering is determined one node at a time by selecting the node with the smallest score. Similar strategies of element-wise ordering learning can be found in many existing algorithms (e.g., Shimizu et al., 2011; Ghoshal and Honorio, 2017b, 2018; Drton et al., 2018).

The novelty of our algorithm is learning an ordering by testing which nodes have the smallest moments ratio in Equation (5) using the ℓ_1-regularized GLM. By substituting the estimation of parameters $\theta(j)$ for an estimation of the conditional mean, we gain significant computational and statistical improvements compared to the previous works in Park and Raskutti (2015, 2018) where the method of moments is used for estimating the conditional mean and variance.

In principle, the number of conditional variances exponentially grows in the number of conditioning variables. Hence, if a conditioning set contains d-variables with 10 possible outcomes, then the number of possible computations is 10^d. In other words, the minimum sample size for the ODS algorithm to be implemented is possibly 10^d, otherwise, none of conditional variances can be estimated.

As we discussed, the problem of a learning directed graph structure is the same as the problem of an learning undirected graph structure if the ordering is known. Hence, given the estimated ordering, the parents of each node j can be learned via ℓ_1-regularized GLM (see details in Meinshausen and Bühlmann, 2006; Wainwright et al., 2006; Ravikumar et al., 2011; Yang et al., 2015). Therefore, we determine the estimated parents of a node j as $\widehat{\mathrm{Pa}}(j) := \{k \in S : \widehat{\theta}_{jk}^S \neq 0\}$ where $S = \widehat{\pi}_{1:(j-1)}$ and $\widehat{\theta}_S(j)$ is the solution to Equation (9).

3.1. Computational Complexity

The computational complexity for the MRS algorithm involves the ℓ_1-regularized GLM algorithm (Friedman et al., 2009) where the worse-case complexity is $O(np)$ for a single ℓ_1-regularized regression run. More precisely, the coordinate descent method updates each

Algorithm 1: Moments Ratio Scoring (MRS)

Input : n i.i.d. samples, $X^{1:n}$
Output: Estimated ordering $\widehat{\pi} = (\widehat{\pi}_1, ..., \widehat{\pi}_p)$ and an edge structure, $\widehat{E} \subset V \times V$

Set $\widehat{\pi}_0 = \emptyset$;
for $m = \{1, 2, \cdots, p\}$ **do**
  Set $S = \{\widehat{\pi}_1, \cdots, \widehat{\pi}_{m-1}\}$;
  for $j \in \{1, 2, \cdots, p\} \setminus S$ **do**
    Estimate $\widehat{\theta}_S(j)$ for ℓ_1-regularized generalized linear model (9);
    Calculate scores $\widehat{\mathcal{S}}(m, j)$ using Equation (8);
  end
  The mth element of the ordering, $\widehat{\pi}_m = \arg\min_j \widehat{\mathcal{S}}(m, j)$;
  The parents of the mth element of the ordering, $\widehat{\mathrm{Pa}}(\widehat{\pi}_m) = \{k \in S \mid \widehat{\theta}^S_{\widehat{\pi}_m k} \neq 0\}$;
end
Return: Estimate the edge set, $\widehat{E} = \cup_{m \in V} \{(k, \widehat{\pi}_m) \mid k \in \widehat{\mathrm{Pa}}(\widehat{\pi}_m)\}$

gradient in $O(p)$ operations. Hence, with d non-zero terms in the GLM, a complete cycle costs $O(pd)$ operations if no new variables become non-zero, and costs $O(np)$ for each new variable entered (see details in Friedman et al., 2010). Since our algorithm has p iterations and there are $p-j+1$ regressions with $j-1$ features for the jth iteration, the total worst-case complexity is $O(np^3)$.

The estimation of a Poisson MRF also involves a node-wise ℓ_1-regularized GLM over all other variables, and hence the worse-case complexity is $O(np^2)$ if the coordinate descent method is exploited. The addition of estimation of ordering makes p times more computationally inefficient than the standard method for learning Poisson MRFs.

Learning a DAG model is NP-hard in general (Chickering et al., 1994). Hence, many state-of-the-art MEC and DAG learning algorithms, such as PC (Spirtes et al., 2000), GES (Chickering, 2003), and MMHC (Tsamardinos et al., 2006), are inherently greedy search algorithms. In the numerical experiments in Section 4, we compare MRS to greedy hill-climbing search-based GES and MMHC algorithms in terms of run time, and show that MRS has a significantly better computational complexity.

3.2. Theoretical Guarantees

In this section, we provide theoretical guarantees on the MRS algorithm for learning Poisson SEMs (4). The main result is expressed in terms of the triple (n, p, d), where n is a sample size, p is a graph node size, and d is the indegree of a graph.

3.2.1. Assumptions

We begin by discussing the assumptions we impose on Poisson SEMs. Since we apply ℓ_1-regularized regression for the parent selection, most assumptions are similar to those imposed in Wainwright et al. (2006), Ravikumar et al. (2011), Yang et al. (2015) and Park and Raskutti (2018) where ℓ_1-regularized regression was used for graphical model learning.

Important quantities are the Hessian matrices of the negative conditional log-likelihood of a node j given some subsets of the nodes in the ordering, $S_j \in \{\{\pi_1\}, \{\pi_1, \pi_2\}, ..., \{\pi_1, ..., \pi_{j-1}\}\}$. Let $Q^{j,S_j} := \nabla^2 \ell_j^{S_j}(\theta^*_S(j); X^{1:n})$ where

$$\ell_j^{S_j}(\theta_{S_j}(j), X^{1:n}) := \frac{1}{n}\sum_{i=1}^{n}\Big[- X_j^{(i)}\Big(\theta_j^{S_j} + \sum_{k \in S_j} \theta_{jk}^{S_j} X_k^{(i)}\Big) + \exp\Big(\theta_j^{S_j} + \sum_{k \in S_j} \theta_{jk}^{S_j} X_k^{(i)}\Big)\Big], \tag{10}$$

$$\theta^*_{S_j}(j) := \arg\min \mathbb{E}\Big[- X_j\Big(\theta_j^{S_j} + \sum_{k \in S_j} \theta_{jk}^{S_j} X_k\Big) + \exp\Big(\theta_j^{S_j} + \sum_{k \in S_j} \theta_{jk}^{S_j} X_k\Big)\Big]. \tag{11}$$

For ease of notation, we define a set for the non-zero elements of $\theta^*_{S_j}(j)$,

$$T_j := \{k \in S_j \mid \theta^*_{jk} \neq 0 \quad \text{where } \theta^*_{S_j}(j) = (\theta^*_j, \theta^*_{jk})\}. \tag{12}$$

We note that if S_j contains all parents of j, $\mathrm{Pa}(j) \subset S_j$, then $T_j = \mathrm{Pa}(j)$. Lastly, for simplicity, we let A_{SS} denote the $|S| \times |S|$ sub-matrix of the matrix A corresponding to variables X_S.

Assumption 1 (Dependence Assumption) *For any $j \in V$ and any $S_j \in \{\{\pi_1\}, \{\pi_1, \pi_2\}, ..., \{\pi_1, ..., \pi_{j-1}\}\}$, there exist positive constants $\rho_{\min}$ and $\rho_{\max}$ such that*

$$\min_{j \in V} \lambda_{\min}\left(Q^{j,S_j}_{T_jT_j}\right) \geq \rho_{\min}, \quad \text{and} \quad \max_{j \in V} \lambda_{\max}\left(\frac{1}{n}\sum_{i=1}^{n} X^{(i)}_{Pa(j)} (X^{(i)}_{Pa(j)})^T\right) \leq \rho_{\max},$$

where T_j is in Equation (12), $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are the smallest and largest eigenvalues of the matrix A, respectively.

Assumption 2 (Incoherence Assumption) *For any $j \in V$ and any $S_j \in \{\{\pi_1\}, \{\pi_1, \pi_2\}, ..., \{\pi_1, ..., \pi_{j-1}\}\}$, there exists a constant $\alpha \in (0, 1]$ such that*

$$\max_{j, S_j} \max_{t \in T_j^c} \|Q^{j,S_j}_{tT_j} (Q^{j,S_j}_{T_jT_j})^{-1}\|_1 \leq 1 - \alpha,$$

where T_j is in Equation (12).

Assumption 1 ensures that the parent variables are not too dependent. In addition, Assumption 2 ensures that parent and non-parent variables are not highly correlated. These two assumptions are standard in all neighborhood regression approaches to variable selection involving ℓ_1-regularized based methods, and these conditions have imposed in proper works for both high-dimensional regression and graphical model learning.

To control the tail behavior of likelihood functions, we require a bounded sample assumption which is also imposed in the standard ℓ_1-regularized Poisson regression (e.g., Jia et al., 2017).

Assumption 3 (Bounded Sample Assumption) *For any $i \in \{1, 2, ..., n\}$, $j \in V$, and for all $S_j \in \{\{\pi_1\}, \{\pi_1, \pi_2\}, ..., \{\pi_1, ..., \pi_{j-1}\}\}$, the samples are bounded:*

$$\max_{i,j}\{X_j^{(i)}\} < C_x \log(\max\{n, p\}) \quad \textit{and} \quad \max_{i,j}\{exp(\theta_j^* + \sum_{k \in S_j} \theta_{jk}^* X_k^{(i)})\} < C_x \log(\max\{n, p\}).$$

where $C_x > 2$ is a positive constant.

Assumption 3 is closely related to the rate parameters. For instance, the rate parameter of $X_j^{(i)}$ is $\exp(\theta_j^* + \sum_{k \in \mathrm{Pa}(j)} \theta_{jk}^* X_k^{(i)})$ by the definition of Poisson SEMs. Hence, Assumption 3 can be understood that too large rate parameters, that leads to a large value of a sample, are not allowed for all conditional distributions.

In fact, Assumption 3 is satisfied with a high probability when (θ_{jk}^*) are negative. Since the second condition in Assumption 3 is directly satisfied with negative (θ_{jk}^*), we discuss the first condition: Using the union bound,

$$P\left(\max_{i,j} X_j^{(i)} \geq C_x \log(\max\{n, p\})\right) \leq n.p \max_{i,j} \frac{\mathbb{E}(\exp(X_j^{(i)}))}{(\max\{n, p\})^{C_x}} \leq \max_{i,j} \frac{\mathbb{E}(\exp(X_j^{(i)}))}{(\max\{n, p\})^{C_x - 2}}.$$

In addition, the moment generating function is bounded when (θ_{jk}^*) are negative.

$$\mathbb{E}(\exp(X_j)) \leq \mathbb{E}(\mathbb{E}(\exp(X_j) \mid X_{\mathrm{Pa}(j)})) \leq \mathbb{E}(\exp(\theta_j^* + \sum \theta_{jk}^* X_k)) \leq \exp(\theta_j^*).$$

Hence, given the negative (θ_{jk}^*) assumption, Assumption 3 is satisfied with probability at least $1 - \max_j \exp(\theta_j)/(\max\{n, p\})^{C_x - 2}$.

Lastly, we require a stronger version of the moments ratio relation in Equation (5), because we move from the population to the finite samples. This assumption only involves learning the ordering of a graph.

Assumption 4 *For all $j \in V$ and $S_j \in \{\{\pi_1\}, \{\pi_1, \pi_2\}, ..., \{\pi_1, ..., \pi_{j-1}\}\}$, there exists an $M_{\min} > 0$ such that*

$$\mathbb{E}(X_j^2) > (1 + M_{\min})\mathbb{E}[\mathbb{E}(X_j \mid X_{S_j}) + \mathbb{E}(X_j \mid X_{S_j})^2].$$

Now, we compare Assumptions 1, 2, 3, and 4 to the assumptions for learning Poisson MRFs and DAG models. As discussed, our assumptions are similar to the assumptions in Yang et al. (2015) and Park and Raskutti (2018) since all methods exploit the ℓ_1-regularized GLM. However, the assumptions in Yang et al. (2015) only involve neighbors of node j, that is, $S_j = V \setminus j$. While our assumptions involve some subsets of parents, that is, $S_j \in \{\{\pi_1\}, \{\pi_1, \pi_2\}, ..., \{\pi_1, ..., \pi_{j-1}\}\}$ due to the unknown ordering. In addition, they do not assume the bounded sample assumption. However, they assume the restricted negative parameter space $\theta_{jk} < 0$ due to the normalizability issue. As we explained, if all parameters are negative in a Poisson SEM, the moment generating function is bounded, and hence, the bounded sample assumption is satisfied with a high probability. Lastly, Yang et al. (2015) does not have the moments ratio assumption, since it is only used for recovering the ordering.

We compare the required assumptions for the MRS and ODS algorithms in Park and Raskutti (2018). A major difference is that the MRS algorithm directly estimates the

graph, while the ODS algorithm estimates the moralized graph to reduce the search space of DAGs, and then, estimates the graph. Hence, our assumptions involve some parents of node j, while their assumptions involve not only parents, but neighbors of node j, that is, $S_j = \{\{\pi_1, ..., \pi_{j-1}\}, V \setminus j\}$. In addition, they require a sparse moralized graph and adjacent faithfulness that are also known to be restrictive. We note that the sparse moralized graph assumption can be very strong since a sparse moralized graph is not implied by a sparse graph. For instance, consider a star graph where $X_1 \to X_j$ for all $j \in \{2, 3, ..., p\}$ in Fig. 2. This star graph has the maximum degree of the moralized graph is $p-1$, while the maximum indegree is 1.

Another major difference is in the moments ratio assumption. More precisely, Park and Raskutti (2015, 2018) assume $\mathrm{Var}(\mathbb{E}(X_j \mid X_S = x)) > c$ for all $x \in \mathcal{X}_S$ when $\mathrm{Pa}(j) \not\subset S$, while we require $\mathbb{E}(\mathrm{Var}(\mathbb{E}(X_j \mid X_S = x))) > c$. To emphasize the difference, we consider a 3-node graph $X_1 \to X_2 \to X_3$ where $X_1 \sim Poisson(\lambda), X_2 \mid X_1 \sim Poisson(\exp(\theta_1 X_1))$, and $X_3 \mid X_2 \sim Poisson(\exp(\theta_2 X_2))$. Then, for $j = 3$ and $S = 1$, we have

$$\mathrm{Var}(\mathbb{E}(X_3 \mid X_2) \mid X_1) = \mathrm{Var}(\exp(\theta_2 X_2) \mid X_1) < \mathbb{E}(\exp(2\theta_2 X_2) \mid X_1) = \exp(e^{\theta_1 X_1}(e^{2\theta_2} - 1)).$$

Hence, for some constants θ_1, θ_2 and c, if $X_1 < \frac{1}{\theta_1}(\log\log c - \log(e^{2\theta_2} - 1))$, their assumption is not satisfied, while Assumption 4 holds.

Lastly, the ODS algorithm requires at least two distinct element of $X_{\mathrm{Pa}(j)}^{(i)}$ for a conditional variance estimation, $\mathrm{Var}(X_j \mid X_{\mathrm{Pa}(j)})$. In principle, it can be 2^d by assuming all variables are binary. Hence when d is not so sparse, the ODS algorithm often fails to be implemented. In Section 4, we empirically verify that it can be a critical issue for the ODS algorithm when a graph is not so sparse ($d = 5$). Therefore, we believe that the assumptions for the MRS algorithm are more realistic.

Although our assumptions are standard in the previous works of ℓ_1-regularized Poisson regressions, we have to note that the assumptions cannot be confirmed from data and they could be restrictive. However, they are not strong for ℓ_1-regularized regression when samples are from Gaussian SEMs (see e.g., Ravikumar et al., 2011). Hence, we conjecture that our assumptions can be satisfied with a high probability under mild conditions, and leave this to future study.

3.2.2. Main Result

Putting together Assumptions 1, 2, 3, and 4, we have the following main result that a Poisson SEM can be recovered via our MRS algorithm in high-dimensional settings. The theorem provides not only sufficient conditions, but also the probability that our method recovers the true graph structure.

Theorem 3 *Consider a Poisson SEM* (4) *with parameter vector $(\theta(j))_{j \in V}$ and the maximum indegree of the graph d. Suppose that the regularization parameter* (9) *is chosen, such that*

$$\frac{4C_x^2\sqrt{2}(2-\alpha)}{\alpha}\frac{\log^2(\max\{n, p\})}{\kappa_1(n, p)} \leq \lambda_j \leq \frac{\alpha\rho_{\min}^2}{10^2 C_x^2(2-\alpha)\rho_{\max} d \log^2(\max\{n, p\})},$$

for any $\alpha = (0, 1]$, and $\kappa_1(n, p) \geq \frac{4\sqrt{2}\cdot 10^2 C_x^4 \cdot (2-\alpha)^2}{\alpha^2}\frac{\rho_{\max}}{\rho_{\min}^2} d \log^4(\max\{n, p\})$. Suppose also that Assumptions 1, 2, 3 and 4 are satisfied and the values of the parameters in Equation (4) *are*

sufficiently large such that $\min_{(j,k)\in E}|\theta_{jk}| \geq \frac{10}{\rho_{\min}}\sqrt{d}\lambda_j$. *Then, for any* $\epsilon > 0$, *there exists a positive constant* C_ϵ *such that if the sample size is sufficiently large* $n > C_\epsilon(\kappa_1(n,p))^2 \log p$, *then the MRS algorithm uniquely recovers the graph with a high probability:*

$$P(\widehat{G} = G) \geq 1 - \epsilon.$$

Detailed proof is provided in Appendices C and D. Appendix C provides the error probability that ℓ_1-regularized regression recovers the true parents of each node given the true ordering, and Appendix D provides the error probability that ℓ_1-regularized regression recovers the ordering. The key technique for the proof is that the *primal-dual witness* method used in sparse regularized regressions and related techniques (Meinshausen and Bühlmann, 2006; Wainwright et al., 2006; Ravikumar et al., 2011; Yang et al., 2015). Theorem 3 intuitively makes sense because neighborhood selection via the ℓ_1-regularized regression is a well-studied problem, and its bias can be controlled by choosing the appropriate regularization parameter λ_j. Hence, our moments ratio scores can be sufficiently close to the true scores to recover the true ordering.

Theorem 3 claims that if $n = \Omega(d^2 \log^9 p)$, our MRS algorithm recovers an underlying graph with a high probability. Hence, our MRS algorithm works in a high-dimensional setting, provided that the indegree of a graph d is bounded. This sample bound result shows that our method has much more relaxed constraints on the sparsity of the graph than the previous work in Park and Raskutti (2018), where the sample bound is $n = \Omega(\max\{d_m^4 \log^{12} p, \log^{5+d} p\})$. Moreover, it also shows that learning Poisson DAG models may require more samples than the learning Poisson MRFs in Yang et al. (2015), where the sample bound is $n = \Omega(d_m^2 \log^3 p))$ due to the existence of the ordering and the unrestricted parameter space.

3.2.3. Poisson SEM with a Star Graph Example

In this section, we discuss the validity of our assumptions using a special Poisson SEM with the star graph in Fig. 2 where $X_1 \sim Poisson(\lambda)$, $X_j \mid X_1 \sim Poisson(\exp(\theta X_1))$, for $j \in \{2, 3, ...p\}$. This consists of a single hub node connected to the rest of nodes. With this star graph, we show that our assumptions can be satisfied with positive (θ_{jk}).

In order to discuss the validity of Assumptions 1, 2, 3, and 4 in this particular example, we first calculate the expectation of the Hessian matrix of Equation (10): For any $j \in \{2, 3, ...p\}$,

$$\begin{aligned}
\mathbb{E}(X_1^2 \exp(\theta X_1)) &= \frac{\partial^2}{\partial\theta^2}\mathbb{E}(\exp(\theta X_1)) = \lambda\exp(\lambda(\exp(\theta)-1)+\theta)(\lambda\exp(\theta)+1), \\
\mathbb{E}(X_1 X_j \exp(\theta X_1)) &= \frac{\partial}{\partial\theta}\mathbb{E}(\exp(\theta_1 X_1)X_j) = \frac{\partial}{\partial\theta}\mathbb{E}(\exp(\theta X_1)E(X_j \mid X_1)) \\
&= \frac{\partial}{\partial\theta}\mathbb{E}(\exp(2\theta X_1)) = 2\lambda\exp(\lambda(\exp(2\theta)-1)+2\theta).
\end{aligned}$$

Hence, the population version of Assumption 1 is reduced to

$$\rho_{\min} < \lambda\exp(\lambda(\exp(\theta)-1)+\theta)(\lambda\exp(\theta)+1) \quad \text{and} \quad \lambda + \lambda^2 < \rho_{\max}.$$

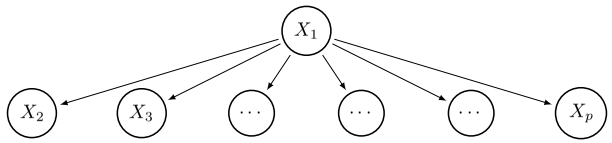


Figure 2: Star graph example

It can be satisfied with some positive values of θ. For $\lambda = 2$, $\rho_{\min} = 0.01$ and $\rho_{\max} = 10$, Assumption 1 is satisfied if $\theta > -3.426$. In addition, for $\lambda = 5$, $\rho_{\min} = 0.01$ and $\rho_{\max} = 50$, it is also satisfied if $\theta > -2.205$.

In addition, the population version of Assumption 2 can be written as

$$\max_{j\in V\setminus\{1\}} \max_{t\in V\setminus\{1,j\}} |\mathbb{E}(Q_{t1}^{j,1})\mathbb{E}((Q_{11}^{j,1})^{-1})| = \frac{2\cdot\exp(\lambda\exp(\theta)(\exp(\theta)-1)+\theta)}{\lambda\exp(\theta)+1} \leq 1-\alpha.$$

This condition is also satisfied with positive values of θ. For $\lambda = 2$ and $\alpha = 0.01$, a simple algebra yields that Assumption 2 is satisfied if $\theta < 0.141$. In addition, for $\lambda = 5$ and $\alpha = 0.01$, the assumption is satisfied if $\theta < 0.165$.

In terms of Assumption 3, we also claim that it can be satisfied with some positive θ. Since the moment generating function of X_1 is $\exp(\lambda(e-1))$, we have,

$$P(X_1^{(i)} > C_x \log(\max\{n,p\})) < \frac{\mathbb{E}(\exp(X_1^{(i)}))}{\max\{n,p\}^{C_x}} = \frac{\exp(\lambda(e-1))}{\max\{n,p\}^{C_x}}.$$

where $C_x > 2$ is a positive constant in Assumption 3.

For other nodes $j \in \{2, 3, ..., p\}$, we have,

$$P(X_j^{(i)} > C_x \log(\max\{n,p\})) \leq \frac{\mathbb{E}(\mathbb{E}(\exp(X_j^{(i)}) \mid X_1^{(i)}))}{\max\{n,p\}^{C_x}} = \frac{\mathbb{E}(\exp(\exp(\theta X_1^{(i)})(e-1)))}{\max\{n,p\}^{C_x}}.$$

For $\theta < \log X_1^{(i)}/X_1^{(i)}$, we have,

$$P(X_j^{(i)} > C_x \log(\max\{n,p\})) \leq \frac{\mathbb{E}(\exp(X_1^{(i)}(e-1)))}{\max\{n,p\}^{C_x}} = \frac{\exp(\lambda(e^{e-1}-1))}{\max\{n,p\}^{C_x}}.$$

Hence, for $\theta < \log(C_x \log(\max\{n,p\}))/C_x \log(\max\{n,p\})$ that is the lower bound of $\log X_1^{(i)}/X_1^{(i)}$ given $X_1^{(i)} < C_x \log(\max\{n,p\})$, Assumption 3 is satisfied with a high probability:

$$P\left(\max_{i,j} X_j^{(i)} > C_x \log(\max\{n,p\})\right) \leq \frac{\exp(\lambda(e^{e-1}-1))}{\max\{n,p\}^{C_x-2}}.$$

Now, we discuss Assumption 4. A simple calculation shows that, for any $j \in \{2, 3, ..., p\}$,

$$\mathbb{E}(X_j) = \exp(\lambda(\exp(\theta-1))), \text{ and } \mathbb{E}(X_j^2) = \exp(\lambda(\exp(2\theta)-1)) + \exp(\lambda(\exp(\theta)-1)).$$

Hence, Assumption 4 is equivalent to the constraint,

$$\exp(\lambda(\exp(2\theta)-1)) > M_{min}\exp(\lambda(\exp(\theta)-1)) + (1+M_{min})\exp(2\lambda(\exp(\theta)-1)).$$

This condition is also satisfied with some positive θ. For $\lambda = 1$ and $M_{min} = 0$, as we discussed in Proposition 1, Assumption 2 is always satisfied with any value of $\theta \neq 0$. For $\lambda = 2$ and $M_{min} = 0.001$, Assumption 2 is satisfied if $|\theta| > 0.033$. Lastly, for $\lambda = 5$ and $M_{min} = 0.001$, Assumption 2 is satisfied if $|\theta| > 0.021$. Therefore, we show that for this particular star graph, Assumption 1, 2, 3, and 4 can be satisfied with a high probability by allowing positive θ.

Finally, we emphasize that the sample complexity of the MRS algorithm, $n = \Omega(d^2 \log^9 p)$, does not rely on the maximum degree of the moralized graph, d_m, while many DAG learning algorithms using the sparsity of the moralized graph or Markov blanket inevitably depend on d_m. For the star graph with $d = 1$ and $d_m = p - 1$, the MRS algorithm requires $n = \Omega(\log^9 p)$ to recover the graph in high dimensional settings, while the ODS algorithm may fail since its sample complexity is $\Omega(d_m^4 \log^{12} p)$. This fact implies that, unlike the ODS algorithm, the MRS algorithm can recover a sparse graph containing hub nodes in high dimensional settings.

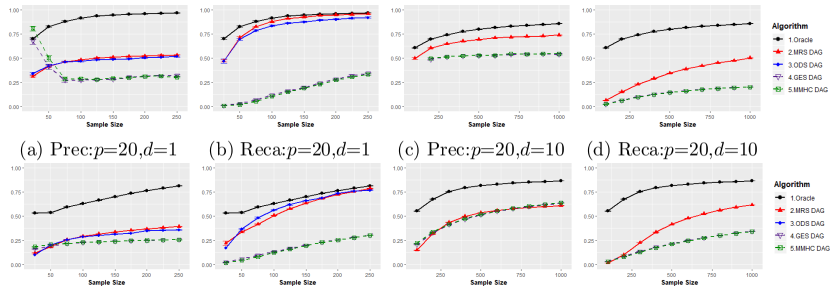
4. Numerical Experiments

In this section, we provide simulation results to support our main theoretical results of Theorem 3 and the computational complexity in Section 3.1: (i) the MRS algorithm recovers the ordering and edges more accurately as sample size increases; (ii) the required sample size $n = \Omega(d^2 \log^9 p)$ depends on the number of nodes p and the complexity of the graph d; (iii) the MRS algorithm accurately learns the graphs in high-dimensional settings ($p > n$); and (iv) the computational complexity is $O(np^3)$ at worst. We also show that the MRS algorithm performs favorably compared to the ODS (Park and Raskutti, 2015), GES (Chickering, 2003), and MMHC (Tsamardinos et al., 2006) algorithms. In addition, we investigate how sensitive our MRS algorithm is to deviations from the assumption about the link functions by using the identity link function in Equation (3). Lastly, we also investigate how well the MRS algorithm recovers undirected edges when samples are generated by Poisson and truncated Poisson MRFs (Yang et al., 2013, 2015; Inouye et al., 2017).

4.1. Random Poisson SEMs

We conducted simulations using 200 realizations of p-node Poisson SEMs (4) with the randomly generated underlying DAG structures while respecting the indegree constraints $d \in \{1, 5, 10\}$. A graph with $d = 1$ is a special case where there is no v-structure, and therefore, the corresponding MEC is completely undirected. The set of non-zero parameters $\theta_j, \theta_{jk} \in \mathbb{R}$ in Equation (4) was generated uniformly at random in the range $\theta_j \in [1, 3]$, $\theta_{jk} \in [-1.5, -0.5] \cup [0.5, 1.5]$ for $d = 1$, and $\theta_{jk} \in [-1, -0.1] \cup [0.1, 1]$ for $d = 5, 10$, which helps the generated values of samples to avoid either all zeros or from going beyond the maximum possible value of the R program ($> 10^{309}$). Nevertheless, if some samples were beyond the maximum possible value, we regenerated the parameters and samples.

The MRS and ODS algorithms were implemented using ℓ_1-regularized likelihood where we used five-fold cross validation to choose the regularization parameters. Where mean squared error was within two standard error of the minimum mean squared error, we chose the minimum value for the moments ratio scores and the largest value for parent selection. That was because a less biased estimator is preferred for the score calculation, and we



(a) Prec:p=20,d=1 (b) Reca:p=20,d=1 (c) Prec:p=20,d=10 (d) Reca:p=20,d=10

(e) Prec:p=200,d=1 (f) Reca:p=200,d=1 (g) Prec:p=200,d=10 (h) Reca:p=200,d=10

Figure 3: Comparison of the MRS algorithm to the oracle, ODS, GES and MMHC algorithms in terms of precision and recall for Poisson SEMs with $p \in \{20, 200\}$ and $d \in \{1, 10\}$.

preferred a sparse graph containing only legitimate edges. We acknowledge that the level of sparsity can be adjusted according to the importance of precision or recall.

In Fig. 3, we compare the MRS algorithm to state-of-the-art ODS, GES and MMHC algorithms for graph node size $p = \{20, 200\}$, varying sample size $n \in \{25, 50, ..., 250\}$ for $d = 1$ and $n = \{100, 200, ..., 1000\}$ for $d = 10$, and provide two results: (i) the average precision ($\frac{\text{\# of correctly estimated edges}}{\text{\# of estimated edges}}$); (ii) the average recall ($\frac{\text{\# of correctly estimated edges}}{\text{\# of true edges}}$). As discussed, the both GES and MMHC algorithms only recover the partial graph by leaving some arrows undirected. Therefore, we also provide average precision and recall for the estimated MECs in Fig. 4. Lastly, we provide an oracle, where the true parents of each node are used, while the ordering is estimated via ℓ_1-regularized GLM. Hence, we can see where the errors come from between the ordering estimation or parent selection. We considered more parameters (θ_{jk}, n, p, d), but for brevity, we focus on these settings.

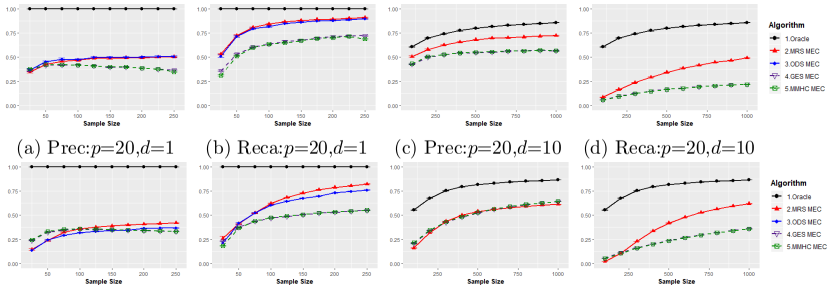
As we can see in Fig. 3, the MRS algorithm more accurately recovers the true directed edges as sample size increases. In addition, the MRS algorithm is more precise for small sparse graphs than for large-scale or dense graphs, given the same sample size. Hence it confirms that the MRS algorithm is consistent, and the sample bound $n = \Omega(d^2 \log^9 p)$ depends on p and d.

The MRS algorithm significantly outperforms state-of-the-art GES and MMHC algorithms in terms of both precision and recall, on average, except for cases $p = 20, d = 1, n \leq 50$. It is worth noting that the GES and MMHC algorithms are not consistent, because the recall for any tree graph must be zero in population, whereas the recall from GES and MMHC increases as sample size increases. Hence, we can conclude that the GES and MMHC algorithms find correct directed edges by finding incorrect v-structures. It is an expected result because the comparison methods only work with a non-faithful distribution, which rarely arises in finite sample settings (Uhler et al., 2013).

Fig. 3 shows that the MRS and ODS algorithms have similar performance in identifying directed edges when the indegree is a small $d = 1$. It makes sense because the ODS

n	100	200	300	400	500	600	700	800	900	1000
p = 20	199	175	107	64	1	0	0	0	0	0
p = 50	200	200	200	199	192	179	151	140	99	86

Table 1: Number of failures in ODS algorithm implementations from among 200 sets of samples for different node sizes $p \in \{20, 50\}$, and sample sizes $n \in \{100, 200, ..., 1000\}$, when the indegree is $d = 5$.



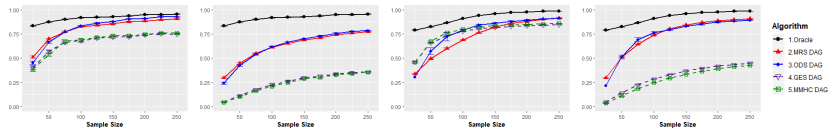
(a) Prec:p=20,d=1 (b) Reca:p=20,d=1 (c) Prec:p=20,d=10 (d) Reca:p=20,d=10

(e) Prec:p=200,d=1 (f) Reca:p=200,d=1 (g) Prec:p=200,d=10 (h) Reca:p=200,d=10

Figure 4: Comparison of the MRS algorithm to the oracle, ODS, GES, and MMHC algorithms in terms of the precision and recall for the MECs of Poisson SEMs with $p \in \{20, 100\}$ and $d \in \{1, 10\}$.

algorithm recovers any Poisson DAG models if the moralized graph is sparse. In other words, the accuracy of the ODS algorithm may be poor for the non-sparse graph. Moreover, the ODS algorithm often fails to be implemented due to a lack of samples for the estimation of conditional variance, that is, $\sum_{i=1}^{n} \mathbf{1}(X_S^{(i)} = x) < 2$ for all $x \in \mathcal{X}_S$. Table 1 shows the number of failures in the ODS algorithm implementations for node size $p \in \{20, 50\}$ and sample size $n \in \{100, 200, ..., 1000\}$ when the indegree is $d = 5$, and the degree of the moralized graph is at most $d_m = p - 1$. It empirically confirms that the ODS algorithm requires a huge number of samples to be implemented when a true graph is not sparse. Hence, we do not apply the ODS algorithm for the graphs with $d = 10$. It is consistent with our main result that our method can learn the Poisson SEMs with some hub nodes while the ODS algorithm might not.

Fig. 4 shows the analogous results for the recovery of MECs, in which the MRS and all comparison algorithms consistently learn the true MECs. The performance of the MRS algorithm gets better as sample size increases or node size decreases. In addition, we can see that the MRS algorithm still recovers the MEC of the Poisson SEM better on average than the comparison methods. However, it must be pointed out that our MRS algorithm applies to Poisson SEMs (4), while the ODS algorithm accurately learns sparse Poisson



(a) Prec:p=20,d=2 (b) Reca:p=20,d=2 (c) Prec:p=100,d=2 (d) Reca:p=100,d=2

Figure 5: Comparison of the MRS algorithm to the oracle, ODS, GES and MMHC algorithms in terms of the precision and recall for Poisson DAG models with $p \in \{20, 100\}$, $d = 2$, and the identity link function.

DAG models where arbitrary link functions are allowed. In addition, the GES and MMHC algorithms apply to more general classes of DAG models.

4.2. Random Poisson DAG Models

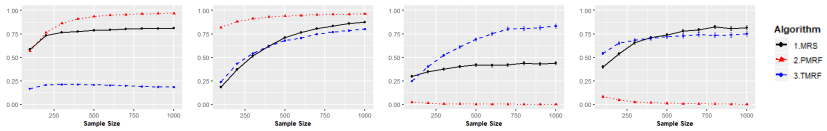
When the data are generated by a random Poisson DAG model (2) where g_j is not the standard log link function, our MRS algorithm is not guaranteed to estimate the true directed acyclic graph and its ordering. Hence, an important question is how sensitive our method is to deviations from the link assumption. In this section, we empirically investigate this question.

We generated the 200 samples with the same procedure specified in Section 4.1, but with the indegree constraint $d = 2$, and except that identity link function $g_j(\eta) = \eta$ and the range of parameters was $\theta_{jk} \in [-1.5, -0.5] \cup [0.5, 1.5]$. We note that the link function must be positive, but we allow the negative value of θ_{jk} by randomly choosing $\theta_j \in [1, 10]$. If any Poisson rate parameter is negative, we regenerated the parameters.

In Fig. 5, we compare the MRS to state-of-the-art ODS, GES and MMHC algorithms for varying sample size $n \in \{25, 50, ..., 250\}$, and node size $p \in \{20, 100\}$. Fig. 5 shows that the MRS algorithm consistently recovers the true graph, and hence, we can see that the MRS algorithm is not so sensitive to deviations from the link assumption. Comparing it to the ODS algorithm, the MRS algorithm shows slightly worse performance because the ODS algorithm is designed to learn general Poisson DAG models with any type of link functions. However, we can see that the MRS algorithm still performs better than the greedy search-based methods in both average precision and recall.

4.3. Random Poisson and Truncated Poisson Markov Random Fields

When samples are generated by a Poisson or truncated Poisson MRF, our MRS algorithm is not guaranteed to find the true dependence relationships of variables. Hence, it is also important to investigate how well our algorithm recovers undirected edges when multivariate count data is from an MRF. In this section, we compare our MRS algorithm to state-of-the-art Poisson MRF (PMRF) and truncated Poisson MRF learning (TMRF) algorithms (Yang et al., 2013, 2015; Inouye et al., 2017) when multivariate count data is from Poisson MRFs and truncated Poisson MRFs, respectively. We used the R package XMRF (Wan et al., 2016) for truncated Poisson MRFs.



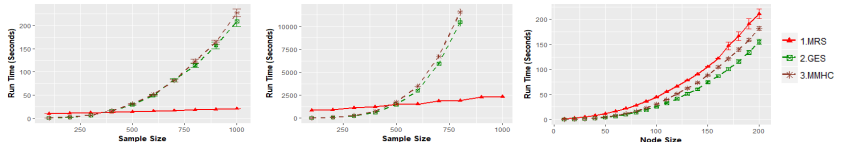
(a) Poisson: Prec (b) Poisson: Reca (c) Truncated: Prec (d) Truncated: Reca

Figure 6: Comparison of the MRS algorithm to the Poisson MRF learning (PMRF) and truncated Poisson MRF learning (TMRF) algorithms in terms of the precision and recall for undirected edges of random 20-nodes Poisson MRFs and truncated Poisson MRFs with $d_m = 5$, and $R = 100$.

We generated 100 samples of 20-nodes random Poisson MRF and truncated Poisson MRF with the randomly generated underlying undirected graphs, respectively. For Poisson MRFs, we set the maximum Markov blanket $d_m = 5$ and the non-zero parameters in Equation (6) was generated uniformly at random in the range $\theta_j \in [1, 2]$, but we fixed $\theta_{jk} = -0.1$ for all $j \in V$. This is a similar setting used in Yang et al. (2015). For truncated Poisson MRFs, we set $d_m = 5$, $\theta_j = 0$, $\theta_{jk} = 0.1$, and the truncation level is $R = 100$, meaning that all samples are less than 100 (see details in Equation 3 of Yang et al., 2013). In terms of the choice of regularization parameters for the MRS and PMRF algorithms, we used five-fold cross validation as we used in Section 4.1. For the TMRF algorithm, we set the regularization parameters to 0.1 since this value seems to work well.

Fig. 6 compares the MRS algorithm to state-of-the-art PMRF and TMRF algorithms in terms of recovering undirected edges by varying sample size $n \in \{100, 200, ..., 1000\}$. For a fair comparison, we used the skeleton of the estimated MEC via the MRS algorithm, because our algorithm returns a DAG. As we can see in Fig. 6, the MRS algorithm consistently finds the true edges from both Poisson MRF and truncated Poisson MRF samples. Hence, we empirically verify that the MRS algorithm can recover some dependence relationships of variables even if samples are from Poisson or truncated Poisson MRFs.

Fig. 6 also shows that the MRS algorithm performs significantly worse than the comparison PMRF and TMRF algorithm, on average, when samples are from Poisson MRFs and truncated Poisson MRFs, respectively. It is an expected result because the PMRF and TMRF algorithms are for learning Poisson MRFs and truncated MRFs, while our algorithm is for Poisson SEMs. However, it is worth noting that the TMRF algorithm seems not to work on average when samples are from a Poisson MRF in our setting. It is mainly because the TMRF algorithm is for learning truncated Poisson MRFs, not Poisson MRFs. We emphasize that, in another setting where θ_j is fixed to 1, the TMRF algorithm works much better. It is also worth noting that the PMRF algorithm seems not to recover any undirected edges when samples are from a truncated Poisson MRF. It can be clearly explained by the fact that the PMRF algorithm cannot capture the positive dependencies, however all parameters are positive in our setting.



(a) $p = 100, d = 5$ (b) $p = 500, d = 5$ (c) $n = 500, d = 5$

Figure 7: Comparison of the MRS algorithm to the GES and MMHC algorithms in terms of the running time with respect to node size p and sample size n

4.4. Computational Complexity

Fig. 7 compares the run-time of the MRS, GES, and MMHC algorithms for learning Poisson SEMs with indegree $d = 5$ by varying sample size $n \in \{100, 200, ..., 1000\}$ with fixed node size $p \in \{100, 500\}$, and varying node size $p \in \{10, 20, ..., 200\}$ with fixed sample size $n = 500$. Fig. 7 supports the worst case computational complexity $O(np^3)$ discussed in Section 3.1. In addition, it shows that the MRS algorithm is significantly faster than the greedy search-based GES and MMHC algorithms when a sample size is large ($n > 500$).

5. Real Multivariate Count Data: MLB Statistics

We now apply the MRS algorithm and state-of-the-art ODS and MMHC algorithms to a simple data set that involves multivariate count data that models baseball statistics for Major League Baseball (MLB) players during the 2003 season. To the best of our knowledge, our MRS algorithm is the only algorithm that provides a reliable and scalable approach to non-sparse DAG learning with multivariate count data although it is under strong assumptions. In particular, other approaches, such as PC, MMHC, and approaches based on conditional independence testing, suffer severely from the fact that we are dealing with count variables where the number of discrete states is potentially infinite. In addition, ODS algorithm cannot deal with a non-sparse graph such as a graph containing hub nodes. Lastly, both Poisson MRF and truncated Poisson MRF may provide an extremely complicated graph because it connects all pairs of nodes having a common child like a moralized graph.

Our original data set consists of 800 MLB player salary and batting statistics from the 2003 season (see R package Lahman in Friendly, 2017 for detailed information). The data set contains 23 covariates: Salary, Number of: Games Played (G), At Bats (AB), Runs (R), Hits (H), Doubles (X2B), Triples (X3B), Home Runs (HR), Runs Batted In (RBI), Stolen Bases (SB), times Caught Stealing (CS), Bases on Balls (BB), Strikeouts (SO), Intentional Walks (IBB), times Hit by Pitch (HBP), Sacrifice Hits (SH), Sacrifice Flies (SF), and times Grounded into Double Plays (GIDP), plus Player ID, Year ID, Stint, Team ID, and League ID. However, we eliminated Player ID, Year ID, Stint, Team ID, and League ID because our focus is to find the directional or causal relationships between salary and batting statistics. In addition, we only considered players in the top 25% in terms of the number of games played, because the baseball statistics relationships from players who played only a few

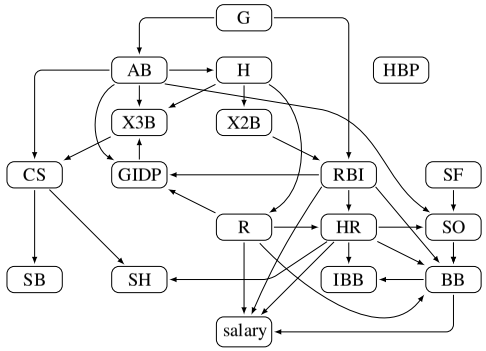


Figure 8: MLB player statistics directed graph estimated by the MRS algorithm for Poisson DAG models.

games could be uncertain. Therefore, the data set we considered contained 18 variables and 200 observations.

We assumed each node to a conditional distribution given its parents is Poisson because most MLB statistics, except for salary, reflect the number of successes or attempts that were counted during the season. Hence, we applied the MRS algorithm for Poisson DAG models with leave-one-out cross validation to choose the tuning parameters, and we chose the largest value where mean squared error is within 2.5 standard error of the minimum mean squared error, because we prefer a sparse graph containing only legitimate edges.

Fig. 8 shows the directed graph estimated by our MRS algorithm. The estimated graph reveals clear causal/directional relationships between batting statistics. This makes sense, because players with larger numbers of HR, BB, RBI, and/or R have a better salary. The more games played, or the more batting chances, the higher H, BB, SO, RBI, and other statistics. Moreover, the higher the total number of hits, the more X2Bs, X3Bs, Rs and the fewer SOs. Players with more home runs and base on balls get intentional walks more frequently. Lastly, the more stolen bases are attempted, the more they are caught stealing, because there is no success without failure.

We acknowledge that our proposed DAG model returns many errors due to restrictive assumptions that are not completely satisfied by the real data. However, the benefit is best seen by comparing MRS to other DAG learning approaches and an undirected graphical model for multivariate count data. In particular, we applied Poisson undirected graphical models (Yang et al., 2015) in which ℓ_1-regularized Poisson regressions are applied. We provide the estimated undirected graph with the largest tuning parameter where mean squared of error is within 2.5 standard error of the minimum mean squared error. The estimated undirected graph in Fig. 9 (left side) shows that a lot of nodes are connected by edges, that many edges are unexplainable, and that some legitimate edges are missing (e.g., [H, X3B], [SB, CS] are not connected), because the Poisson undirected graphical model only

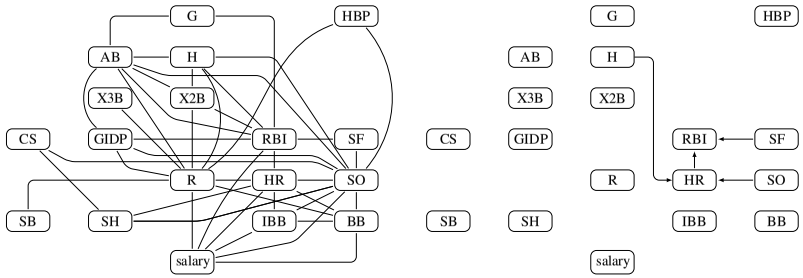


Figure 9: MLB player statistics undirected graph estimated by ℓ_1-penalized likelihood regression (left) and a directed acyclic graph estimated by the MMHC algorithm (right).

permits negative conditional relationships, whereas most variables are positively correlated. Hence, it may not be useful to understand the relationships between MLB statistics.

We also compared the MMHC algorithm. As discussed, the MMHC algorithm does not guarantee to find a complete directed graph, and prefers a sparser graph when the faithfulness assumption is violated, which often arises in finite sample settings (Uhler et al., 2013). Hence, the estimated directed graph in Fig. 9 (right side) is extremely sparse, with only four directed edges: [H, HR], [SO, HR], [HR, RBI], and [SF, RBI]. Lastly, ODS algorithm failed to be implemented as expected because of some hub nodes such as the number of games, at bats, and runs batted in.

Since our method is the first identifiability result for the strongly correlated count data when variables are directional/causal relationships and there exist hub variables, to the best of our knowledge, our method better identifies the directional/causal relationships between MLB statistics. However, we acknowledge that, like most other DAG-learning approaches, very strong assumptions, such as dependency, incoherence, are required for reliable recovery.

6. Future Works

Several topics remain for future works. Although our assumptions are similar to the assumptions in the previous works of ℓ_1-regularized Poisson regression, our assumptions could be very restrictive. In addition, they cannot be confirmed from data. However, we conjecture that the assumptions are satisfied with a high probability under mild conditions, and one may be able to prove this. In addition, it is an important problem of finding the minimax rate of the Poisson DAG models, and it should be investigated in the future. Lastly, it would be also interesting to explore if our idea can be applied to other structural equation models with Binomial, Negative Binomial, Exponential, and Gamma distributions. We believe that our node-wise ℓ_1-regularized based approach can be extended to the identifiable linear SEMs under some suitable conditions.

Acknowledgments

This work was supported under the framework of international cooperation program managed by National Research Foundation of Korea(NRF-2018R1C1B5085420)

Appendix A. Proof for Proposition 1

Proof For a notational simplicity, we define a moments related function for Poisson, $f(\mu) = \mu + \mu^2$ for $\mu > 0$. Then, for any node $j \in V$, any non-empty set $S_j \subset \mathrm{Nd}(j)$,

$$\mathbb{E}(X_j^2 \mid S_j) \;=\; \mathbb{E}(\mathbb{E}(X_j^2 \mid X_{\mathrm{Pa}(j)}) \mid S_j) = \mathbb{E}(f(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)})) \mid S_j).$$

Using the Jensens inequality and $f(\cdot)$ is convex, we have,

$$\mathbb{E}(f(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)})) \mid S_j) \geq f(\mathbb{E}(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)}) \mid S_j)) = f(\mathbb{E}(X_j \mid S_j)).$$

Using the fact that $\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)}) = g_j(X_{\mathrm{Pa}(j)})$ and it is non-degenerated by definition, the equality only holds when S_j contains all parents of j, $\mathrm{Pa}(j) \subset S_j \subset \mathrm{Nd}(j)$.

By restating the above inequality, we have,

$$\mathbb{E}(X_j^2 \mid S_j) - \mathbb{E}(X_j \mid S_j) - \mathbb{E}(X_j \mid S_j)^2 \geq 0.$$

In addition, by taking the expectations, we have,

$$\mathbb{E}(X_j^2) - \mathbb{E}\left(\mathbb{E}(X_j \mid X_{S_j}) + \mathbb{E}(X_j \mid X_{S_j})^2\right) \geq 0.$$

Since j and S_j are arbitrary, we complete the first part of the proof.

Now, we prove that $\mathbb{E}(X_j^2) \geq \mathbb{E}\left(\mathbb{E}(X_j \mid X_{S_j}) + \mathbb{E}(X_j \mid X_{S_j})^2\right)$ is equivalent to $\mathbb{E}(\mathrm{Var}(\mathbb{E}(X_j \mid \mathrm{Pa}(j)) \mid X_{S_j})) \geq 0$. Using the total variance decomposition, we have,

$$\mathbb{E}(\mathrm{Var}(X_j \mid X_{S_j})) = \mathbb{E}(\mathbb{E}(\mathrm{Var}(X_j \mid X_{\mathrm{Pa}(j)}) \mid X_{S_j})) + \mathbb{E}(\mathrm{Var}(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)}) \mid X_{S_j})).$$

Using the fact that the conditional distribution, $X_j \mid X_{\mathrm{Pa}(j)}$, is Poisson where its mean and variance are equal, we have,

$$\mathbb{E}(\mathrm{Var}(\mathbb{E}(X_j \mid X_{\mathrm{Pa}(j)}) \mid X_{S_j})) = \mathbb{E}(\mathrm{Var}(X_j \mid X_{S_j})) - \mathbb{E}(X_j).$$

Using the definition of the conditional variance, we have,

$$\mathbb{E}(\mathrm{Var}(X_j \mid X_{S_j})) - \mathbb{E}(X_j) = \mathbb{E}(X_j^2) - \mathbb{E}\left(\mathbb{E}(X_j \mid X_{S_j}) + \mathbb{E}(X_j \mid X_{S_j})^2\right).$$

Therefore, we complete the proof. ■

Appendix B. Proof for Theorem 2

Proof Without loss of generality, we assume the true ordering is unique, and $\pi = (\pi_1, ..., \pi_p)$. For simplicity, we define $X_{1:j} = (X_{\pi_1}, X_{\pi_2}, \cdots, X_{\pi_j})$ and $X_{1:0} = \emptyset$. In addition, we define a moments related function, $f(\mu) = \mu + \mu^2$.

We now prove identifiability of Poisson DAG models using mathematical induction:

Step (1) For the first step π_1, using Proposition 1, we have $\mathbb{E}(X_{\pi_1}^2) = \mathbb{E}(f(\mathbb{E}(X_{\pi_1})))$, while for any node $j \in V \setminus \{\pi_1\}$: $\mathbb{E}(X_j^2) > \mathbb{E}(f(\mathbb{E}(X_j)))$.

Hence, we can determine π_1 as the first element of the causal ordering.

Step (m-1) For the $(m-1)^{th}$ element of the ordering, assume that the first $m-1$ elements of the ordering and their parents are correctly estimated.

Step (m) Now, we consider the mth element of the causal ordering and its parents. It is clear that π_m achieves $\mathbb{E}(X_{\pi_m}^2) = \mathbb{E}(f(\mathbb{E}(X_{\pi_m} \mid X_{1:(m-1)})))$. However, for $j \in \{\pi_{m+1}, \cdots, \pi_p\}$, $\mathbb{E}(X_j^2) > \mathbb{E}(f(\mathbb{E}(X_j \mid X_{1:(m-1)})))$ by Proposition 1. Hence, we can estimate a true m^{th} component of the ordering π_m.

In terms of the parent search, it is clear that by conditional independence relations naturally encoded by factorization (1) $\mathbb{E}(X_{\pi_m}^2) = \mathbb{E}(f(\mathbb{E}(X_{\pi_m} \mid X_{1:(m-1)}))) = \mathbb{E}(f(\mathbb{E}(X_{\pi_m} \mid X_{\mathrm{Pa}(\pi_m)})))$. Hence, we can also choose the minimum conditioning set from among $X_{1:(m-1)}$ as the parents of π_m such that the above moments relation holds. By mathematical induction, this completes the proof.

■

Appendix C. Proof for Theorem 3: Parents Recovery

Proof We provide the proof for Theorem 3 using the *primal-dual witness method* that is also used many other works (Meinshausen and Bühlmann, 2006; Wainwright et al., 2006; Ravikumar et al., 2011; Yang et al., 2015). In this proof, we show in Appendix C, the error probability for the recovery of the parents of a node π_j from among all the nodes given the partial ordering $(\pi_1, \pi_2, ..., \pi_{j-1})$ via ℓ_1-regularized regression. In Appendix D, the error bounds for the recovery of the ordering both via ℓ_1-regularized regression.

Without loss of generality, let the true ordering be $\pi = (1, 2, ..., p)$, and hence, $\pi_{1:j} = (\pi_1, \pi_2, ..., \pi_j) = (1, 2, ..., j)$. For ease of notation, $[\cdot]_k$ and $[\cdot]_S$ denote parameters corresponding to variable X_k and random vector X_S, respectively. In order to make the arguments easier to understand, we restate the negative log likelihood (10) and related arguments.

First, we define a new parameter vector $\theta_{S_j} \in \mathbb{R}^{|S_j|}$ without parameter θ_j corresponding to the node j since the node j is not penalized in regression problem (9). Then, the conditional negative log-likelihood of the GLM for X_j given X_{S_j} can be written as:

$$\ell_j^{S_j}(\theta_{S_j}; X^{1:n}) := \frac{1}{n}\sum_{i=1}^{n}\left(-X_j^{(i)}\langle\theta_{S_j}, X_{S_j}^{(i)}\rangle + \exp\big(\langle\theta_{S_j}, X_{S_j}^{(i)}\rangle\big)\right), \tag{13}$$

where $\langle\cdot,\cdot\rangle$ is an inner product.

We also define $\theta_{S_j}^* \in \mathbb{R}^{|S_j|}$ for Equation (11):

$$\theta_{S_j}^* := \arg\min_{\theta\in\mathbb{R}^{|S_j|}} \mathbb{E}\left(-X_j(\langle\theta, X_{S_j}\rangle) + \exp(\langle\theta, X_{S_j}\rangle)\right). \tag{14}$$

We define a set non-zero elements index of $\theta_{S_j}^*$ as in Equation (12), $T_j := \{k \in S_j \mid [\theta_{S_j}^*]_k \neq 0\}$ where $\theta_{S_j}^*$ is in Equation (14).

The main goal of the proof is to find the unique minimizer of the following convex problem:

$$\widehat{\theta}_{S_j} := \arg\min_{\theta\in\mathbb{R}^{|S_j|}} \mathcal{L}_j(\theta, \lambda_j) = \arg\min_{\theta\in\mathbb{R}^{|S_j|}} \{\ell_j^{S_j}(\theta; X^{1:n}) + \lambda_j\|\theta\|_1\}. \tag{15}$$

By setting the *sub-differential* to 0, $\widehat{\theta}_{S_j}$ satisfies the following condition:

$$\nabla_\theta \mathcal{L}_j^{S_j}(\widehat{\theta}_{S_j}, \lambda_j) = \nabla_\theta \ell_j^{S_j}(\widehat{\theta}_{S_j}; X^{1:n}) + \lambda_j \widehat{Z}_j^{S_j} = 0 \tag{16}$$

where $\widehat{Z}_j^{S_j} \in \mathbb{R}^{|S_j|}$ and $[\widehat{Z}_j^{S_j}]_t = \mathrm{sign}([\widehat{\theta}_{S_j}]_t)$ if $t \in T_j$, otherwise $[\widehat{Z}_j^{S_j}]_t < 1$.

Lemma 4 directly follows from the prior work (Yang et al., 2015), where each node's conditional distribution is in the form of a generalized linear model.

Lemma 4 (Uniqueness of Solution, Lemma 8 in Yang et al., 2015) *Suppose that* $|[\widehat{Z}_j^{S_j}]_t| < 1$ *for* $t \notin T_j$ *in Equation* (16). *Then, the solution* $\widehat{\theta}_{S_j}$ *of Equation* (15) *satisfies* $[\widehat{\theta}_{S_j}]_t = 0$ *for all* $t \notin T_j$. *Furthermore, if the sub-matrix of Hessian matrix* $Q_{T_jT_j}^{S_j}$ *is invertible, then* $\widehat{\theta}_{S_j}$ *is unique.*

The remainder of the proof is to show $|[\widehat{Z}_j^{S_j}]_t| < 1$ for all $t \notin T_j$. Note that the restricted solution in Equation (22) is $(\widetilde{\theta}_{S_j}, \widetilde{Z}_j^{S_j})$ and the unrestricted solution in Equation (15) is $(\widehat{\theta}_{S_j}, \widehat{Z}_j^{S_j})$. Equation (16) with the dual solution can be represented by

$$\nabla^2 \ell_j^{S_j}(\theta_{S_j}^*; X^{1:n})(\widetilde{\theta}_{S_j} - \theta_{S_j}^*) = -\lambda_j \widetilde{Z}_j^{S_j} - W_j^{S_j} + R_j^{S_j} \tag{17}$$

where

(a) $W_j^{S_j}$ is the sample score function:

$$W_j^{S_j} := -\nabla \ell_j(\theta_{S_j}^*; X^{1:n}). \tag{18}$$

(b) $R_j^{S_j} = (R_{jk}^{S_j})_{k\in S_j}$ and $R_{jk}^{S_j}$ is the remainder term by applying the coordinate-wise mean value theorem:

$$R_{jk}^{S_j} := [\nabla^2 \ell_j^{S_j}(\theta_{S_j}^*; X^{1:n}) - \nabla^2 \ell_j^{S_j}(\bar{\theta}_{S_j}; X^{1:n})]_k^T (\widetilde{\theta}_{S_j} - \theta_{S_j}^*). \tag{19}$$

Here $\bar{\theta}_{S_j}$ is a vector on the line between $\widetilde{\theta}_{S_j}$ and $\theta_{S_j}^*$, and $[\cdot]_k^T$ is the row of a matrix corresponding to variable X_k.

Then, the following proposition provides a sufficient condition to control $\widetilde{Z}_j^{S_j}$.

Proposition 5 *If* $\max(\|W_j^{S_j}\|_\infty, \|R_j^{S_j}\|_\infty) \leq \frac{\lambda_j\alpha}{4(2-\alpha)}$, *then* $|[\widetilde{Z}_j^{S_j}]_t| < 1$ *for all* $t \notin T_j$.

Next, we introduce the following three lemmas under Assumptions 1, 2, and 3 to show that conditions in Proposition 5 hold. For ease of notation, let $\eta = \max\{n, p\}$, $\widetilde{\theta}_S = [\widetilde{\theta}_{S_j}]_{T_j}$, $\widetilde{Z}_S = [\widetilde{Z}_j^{S_j}]_{T_j}$, $\widetilde{\theta}_{S^c} = [\widetilde{\theta}_{S_j}]_{S_j\setminus T_j}$, and $\widetilde{Z}_{S^c} = [\widetilde{Z}_j^{S_j}]_{S_j\setminus T_j}$.

Lemma 6 *For any* $S_j \in \{\pi_1, \pi_{1:2}, ..., \pi_{1:j-1}\}$ *and* $\lambda_j \geq \frac{4C_x^2\sqrt{2}(2-\alpha)}{\alpha}\frac{\log^2\eta}{\kappa_1(n,p)}$ *for some* $\alpha \in (0, 1]$,

$$P\left(\frac{\|W_j^{S_j}\|_\infty}{\lambda_j} \leq \frac{\alpha}{4(2-\alpha)}\right) \geq 1 - 2d \cdot exp\left(-\frac{n}{\kappa_1(n,p)^2}\right).$$

where $\kappa_1(n, p)$ *is an arbitrary function of* n *and* p.

Lemma 7 *Suppose that for all $S_j \in \{\pi_1, \pi_{1:2}, ..., \pi_{1:j-1}\}$, $\|W_j^{S_j}\|_\infty \le \frac{\lambda_j}{4}$. Then, for $\lambda_j \le \frac{\rho_{\min}^2}{10C_x^2\rho_{\max} d\log^2\eta}$,*

$$\|\widetilde{\theta}_S - \theta_S^*\|_2 \le \frac{5}{\rho_{\min}}\sqrt{d}\lambda_j$$

Lemma 8 *Suppose that for all $S_j \in \{\pi_1, \pi_{1:2}, ..., \pi_{1:j-1}\}$, $\|W_j^{S_j}\|_\infty \le \frac{\lambda_j}{4}$. Then, for $\lambda_j \le \frac{\alpha\rho_{\min}^2}{100C_x^2(2-\alpha)\rho_{\max} d\log^2\eta}$ and $\alpha \in (0,1]$,*

$$\frac{\|R_j^{S_j}\|_\infty}{\lambda_j} \le \frac{\alpha}{4(2-\alpha)}$$

The rest of the proof is straightforward using Lemmas 6, 7, and 8. Consider the choice of regularization parameter $\lambda_{j0} = \frac{4\sqrt{2}C_x^2(2-\alpha)}{\alpha}\frac{\log^2\eta}{\kappa_1(n,p)}$, where $\kappa_1(n,p) \ge \frac{4\sqrt{2}C_x^4\cdot 10^2(2-\alpha)^2}{\alpha^2}\frac{\rho_{\max}}{\rho_{\min}^2} d\log^4\eta$ ensuring that $\frac{4C_x^2\sqrt{2}(2-\alpha)}{\alpha}\frac{\log^2\eta}{\kappa_1(n,p)} \le \lambda_{j0} \le \frac{\alpha\rho_{\min}^2}{10^2C_x^2(2-\alpha)\rho_{\max} d\log^2\eta}$ for any $\alpha = (0,1]$. Hence, if we set $\kappa_1(n,p) = C_{\max} d\log^4\eta$ where $C_{\max} = \frac{4\sqrt{2}\cdot 10^2C_x^4\cdot(2-\alpha)^2}{\alpha^2}\frac{\rho_{\max}}{\rho_{\min}^2}$, then all conditions for Lemma 6, 7, and 8 are satisfied. Therefore,

$$\|\widetilde{Z}_{S^c}\|_\infty \le (1-\alpha) + (2-\alpha)\left[\frac{\|W_j^{S_j}\|_\infty}{\lambda_j} + \frac{\|R_j^{S_j}\|_\infty}{\lambda_j}\right] \le (1-\alpha) + \frac{\alpha}{4} + \frac{\alpha}{4} < 1, \tag{20}$$

with a probability of at least $1 - 2d\cdot\exp\left(-\frac{n}{\kappa_1(n,p)^2}\right) = 1 - 2d\cdot\exp\left(-\frac{n}{C_{\max}^2 d^2\log^8\eta}\right)$.

Proposition 9 *Suppose that, for any $j \in V$, partial ordering $(\pi_1, ..., \pi_j)$ is correctly estimated. If $\min_{t\in S}[\theta_S^*]_t \ge \frac{10}{\rho_{\min}}\sqrt{d}\ \lambda_j$ for all $j \in V$,*

$$supp(\widehat{\theta}_{S_j}) = Pa(j).$$

Proposition 9 guarantees that ℓ_1-regularized likelihood regression recovers the parents for each node with a high probability. Since there are p regression problems, for any $\epsilon > 0$, there exists a positive constant $C_\epsilon > 0$ such that if $n \ge C_\epsilon(\kappa_1(n,p)^2\log p)$ for $\kappa_1(n,p) \ge C_{\max} d\log^4\eta$,

$$P(\widehat{G} = G) \ge 2dp\cdot\exp\left(-\frac{n}{\kappa_1(n,p)^2}\right) \ge 1 - 2dp\cdot\exp\left(-C_\epsilon\log p\right) \ge 1-\epsilon.$$

■

Appendix D. Proof for Theorem 3: Ordering Recovery

Proof We begin by reintroducing some necessary notations and definitions to make the proof concise. Without loss of generality, assume that the true ordering is unique and $\pi = (\pi_1, ..., \pi_p) = (1, 2, ..., p)$. For notational convenience, we define $X_{1:j} = (X_{\pi_1}, X_{\pi_2}, \cdots, X_{\pi_j})$

$= (X_1, X_2, ..., X_j)$ and $X_{1:0} = \emptyset$. We restate the moments ratio scores for a node k and the jth element of the ordering:

$$\mathcal{S}(j,k) := \frac{\mathbb{E}(X_k^2)}{\mathbb{E}(f(\mathbb{E}(X_k \mid X_{1:(j-1)})))} \quad \text{and} \quad \widehat{\mathcal{S}}(j,k) := \frac{\widehat{\mathbb{E}}(X_k^2)}{\widehat{\mathbb{E}}(f(\widehat{\mathbb{E}}(X_k \mid X_{\widehat{\pi}_{1:(j-1)}})))},$$

where $f(\mu) := \mu + \mu^2$, $\mathbb{E}(X_k \mid X_{S_k}) = \exp(\theta_k^* + \sum_{t\in S_k}\theta_{kt}^* X_t)$, and $\widehat{\mathbb{E}}(X_k \mid X_{S_k}) = \exp(\hat{\theta}_k + \sum_{t\in S_k}\hat{\theta}_{kt}X_t)$ where $\theta_{S_k}^* = (\theta_k^*, \theta_{kt}^*)$ and $\hat{\theta}_{S_k} = (\hat{\theta}_k, \hat{\theta}_{kt})$ are the solutions of the problem (11) and of the ℓ_1-regularized GLM (9), respectively. In addition, we use the unbiased method-of-moment estimator for a marginal expectation, $\widehat{\mathbb{E}}(X_k^2) = \frac{1}{n}\sum_{i=1}^n (X_k^{(i)})^2$ and $\widehat{\mathbb{E}}(f(\widehat{\mathbb{E}}(X_k \mid X_S))) = \frac{1}{n}\sum_{i=1}^n f(\exp(\hat{\theta}_k + \sum_{t\in S_k}\hat{\theta}_{kt}X_t^{(i)}))$.

We define the following necessary events: For each node $j \in V$, $S_j \in \{\pi_1, \pi_{1:2}, ..., \pi_{1:(j-1)}\}$ and any $\epsilon_1 > 0$;

$$\begin{aligned}
\zeta_1 &:= \left\{\max_{j=1,...,p-1}\max_{k=j,...,p}\left|\mathcal{S}(j,\pi_k) - \widehat{\mathcal{S}}(j,\pi_k)\right| > \frac{M_{\min}}{2}\right\},\\
\zeta_2 &:= \left\{\max_{j\in V}\left|\widehat{\mathbb{E}}(X_j^2) - \mathbb{E}(X_j^2)\right| < \epsilon_1\right\},\\
\zeta_3 &:= \left\{\max_{j\in V}\left|\widehat{\mathbb{E}}\left(f\left(\widehat{\mathbb{E}}(X_j \mid X_{S_j})\right)\right) - \mathbb{E}\left(f\left(\widehat{\mathbb{E}}(X_j \mid X_{S_j})\right)\right)\right| < \epsilon_1\right\},\\
\zeta_4 &:= \left\{\max_{j\in V}\left|\mathbb{E}\left(f\left(\widehat{\mathbb{E}}(X_j \mid X_{S_j})\right)\right) - \mathbb{E}\left(f\left(\mathbb{E}(X_j \mid X_{S_j})\right)\right)\right| < \epsilon_1\right\}.
\end{aligned}$$

We begin by proving that our algorithm recovers the ordering of a Poisson SEM in the high-dimensional setting. The probability that ordering is correctly estimated from our method can be written as

$$\begin{aligned}
&P\left(\widehat{\pi} = \pi\right)\\
=&P\left(\widehat{\mathcal{S}}(1,\pi_1) < \min_{j=2,...,p}\widehat{\mathcal{S}}(1,\pi_j), \widehat{\mathcal{S}}(2,\pi_2) < \min_{j=3,...,p}\widehat{\mathcal{S}}(2,\pi_j), ..., \widehat{\mathcal{S}}(p-1,\pi_{p-1}) < \widehat{\mathcal{S}}(p-1,\pi_p)\right)\\
=&P\left(\min_{j=1,...,p-1}\min_{k=j+1,...,p}\widehat{\mathcal{S}}(j,\pi_k) - \widehat{\mathcal{S}}(j,\pi_j) > 0\right)\\
=&P\left(\min_{\substack{j=1,...,p-1\\k=j+1,...,p}}\left\{\left(\mathcal{S}(j,\pi_k) - \mathcal{S}(j,\pi_j)\right) - \left(\mathcal{S}(j,\pi_k) - \widehat{\mathcal{S}}(j,\pi_k)\right) + \left(\mathcal{S}(j,\pi_j) - \widehat{\mathcal{S}}(j,\pi_j)\right)\right\} > 0\right)\\
\ge&P\left(\min_{\substack{j=1,...,p-1\\k=j+1,...,p}}\left\{(\mathcal{S}(j,\pi_k) - \mathcal{S}(j,\pi_j))\right\} > M_{\min}\ , and \max_{\substack{j=1,...,p-1\\k=j,...,p}}\left|\mathcal{S}(j,\pi_k) - \widehat{\mathcal{S}}(j,\pi_k)\right| < \frac{M_{\min}}{2}\right).
\end{aligned}$$

The first term in the above probability is always satisfied because $\mathcal{S}(j,\pi_k) - \mathcal{S}(j,\pi_j) > (1 + M_{\min}) - 1 = M_{\min}$ from Assumption 4. Hence, the lower bound of the probability that

ordering is correctly estimated using our method is reduced to

$$\begin{aligned} P(\widehat{\pi}=\pi) &\geq P\left(\max_{j=1,\dots,p-1}\max_{k=j,\dots,p}\left|\mathcal{S}(j,\pi_k)-\widehat{\mathcal{S}}(j,\pi_k)\right|<\frac{M_{\min}}{2}\right) \\ &= 1-P(\zeta_1) \\ &= 1-P(\zeta_1\mid\zeta_2,\zeta_3,\zeta_4)P(\zeta_2,\zeta_3,\zeta_4)-P(\zeta_1\mid(\zeta_2,\zeta_3,\zeta_4)^c)P((\zeta_2,\zeta_3,\zeta_4)^c) \\ &\geq 1-P(\zeta_1\mid\zeta_2,\zeta_3,\zeta_4)-P((\zeta_2,\zeta_3,\zeta_4)^c) \\ &\geq 1-\underbrace{P(\zeta_1\mid\zeta_2,\zeta_3,\zeta_4)}_{Lem\ 10}-\underbrace{P(\zeta_2^c)-P(\zeta_3^c)-P(\zeta_4^c)}_{Lem\ 11}. \end{aligned} \tag{21}$$

Next, we introduce the following two lemmas to show the lower bound of the probability in (21) as a function of the triple (n,p,d):

Lemma 10 *Given the sets ζ_2,ζ_3,ζ_4 and under Assumption 4, $P(\zeta_1\mid\zeta_2,\zeta_3,\zeta_4)=0$ if for some small ϵ_1 such that for any $S_j\in\{\pi_1,\pi_{1:2},\dots,\pi_{1:(j-1)}\}$,*

$$\epsilon_1<\min\left\{\frac{\mathbb{E}(X_j^2)M_{\min}}{2(M_{\min}+3)(M_{\min}+1)},\frac{M_{\min}}{6}\frac{\mathbb{E}(f(\mathbb{E}(X_j\mid X_{S_j})))^2}{\mathbb{E}(X_j^2)}\right\},$$

where $f(\mu)=\mu+\mu^2$.

The condition in Lemma 10 implies that if ϵ_1 is sufficiently small, the estimated score is close to the true score value.

The second lemma shows the error bound for the consistency of the estimators.

Lemma 11 *For any $\epsilon_1>0$ and*

(i) For ζ_2, $P(\zeta_2^c)\leq 1-2\cdot p\cdot exp\left\{-\frac{n\epsilon_1^2}{2C_x^2\log^2\eta}\right\}$.

(ii) For ζ_3, there exist some positive constants $C_{\max}$ and $D_{\max}$ such that

$$P(\zeta_3^c)\leq 1-2\cdot p\cdot d\cdot\ exp\left(-\frac{n}{\kappa_1(n,p)^2}\right)-2\cdot p\cdot exp\left\{-\frac{n\epsilon_1^2}{D_{\max}\log^4\eta}\right\}.$$

where $\kappa_1(n,p)\geq C_{\max}d\log^4\eta$.

(iii) For ζ_4, $P(\zeta_4^c)=0$.

Therefore, we complete the proof: our method recovers the true ordering at least of

$$P(\widehat{\pi}=\pi)\ \geq\ 1-C_1p\cdot d\cdot\exp\left(-C_2\frac{n}{\kappa_1(n,p)^2}\right).$$

for $\kappa_1(n,p)\geq C_{\max}d\log^4\eta$, and some positive constants C_1 and C_2

■

Appendix E. Proposition 12

We begin by introducing an important proposition to control the tail behavior for the distribution of each node, which are required to prove the lemmas.

Proposition 12 *For given $j\in V$ and $S_j\in\{\pi_1,\pi_{1:2},\dots,\pi_{1:(j-1)}\}$, the solution $\widehat{\theta}_{S_j}$ in Equation (15) satisfies*

$$\frac{1}{n}\sum_{i=1}^n exp\left(\langle\widehat{\theta}_{S_j},X_{S_j}^{(i)}\rangle\right)<C_x\log\eta.$$

where $C_x>2$ is a constant in Assumption 3.

Proof By the first-order optimality condition of $\mathcal{L}_j^{S_j}(\theta_{S_j},X^{1:n})$ in Equation (15), we have

$$\begin{aligned} \sum_{i=1}^n X_j^{(i)} &= \sum_{i=1}^n\exp(\langle\hat{\theta}_{S_j},X_{S_j}^{(i)}\rangle) \\ \sum_{i=1}^n X_j^{(i)}X_k^{(i)} &= \sum_{i=1}^n\exp\left(\langle\hat{\theta}_{S_j},X_{S_j}^{(i)}\rangle X_k^{(i)}\right)+\lambda_j\text{sign}([\hat{\theta}_{S_j}]_k). \end{aligned}$$

By Assumption 3, we have

$$\frac{1}{n}\sum_{i=1}^n\exp\left(\langle\hat{\theta}_{S_j},X_{S_j}^{(i)}\rangle\right)\leq C_x\log\eta\iff\frac{1}{n}\sum_{i=1}^n X_j^{(i)}\leq C_x\log\eta.$$

■

Appendix F. Proof for Propositions 5 and 9

F.1. Proof for Proposition 5

Proof We note that $\widetilde{\theta}_{S^c}=(0,0,\dots,0)^T\in\mathbb{R}^{|S^c|}$ in our primal-dual construction. To improve readability, we let $\theta_S=[\theta_{S_j}]_{T_j},\theta_{S^c}=[\theta_{S_j}]_{S_j\setminus T_j}$, and $A_S=[A_j^{S_j}]_{T_j}$ and $A_{S^c}=[A_j^{S_j}]_{S_j\setminus T_j}$. With these notations, W_S and R_S are sub-vectors of $W_j^{S_j}$ and $R_j^{S_j}$ corresponding to variables X_S, respectively.

We can restate condition (17) in block form as follows:

$$\begin{aligned} Q_{S^cS}[\widetilde{\theta}_S-\theta_S^*] &= W_{S^c}-\lambda_j\widetilde{Z}_{S^c}+R_{S^c}, \\ Q_{SS}[\widetilde{\theta}_S-\theta_S^*] &= W_S-\lambda_j\widetilde{Z}_S+R_S. \end{aligned}$$

Since Q_{SS} is invertible, the above equations can be rewritten as

$$Q_{S^cS}Q_{SS}^{-1}[W_S-\lambda_j\widetilde{Z}_S-R_S]=W_{S^c}-\lambda_j\widetilde{Z}_{S^c}-R_{S^c}.$$

Therefore,

$$[W_{S^c}-R_{S^c}]-Q_{S^cS}Q_{SS}^{-1}[W_S-R_S]+\lambda_jQ_{S^cS}Q_{SS}^{-1}\widetilde{Z}_S=\lambda_j\widetilde{Z}_{S^c}.$$

Taking the ℓ_∞ norm of both sides yields

$$\|\widetilde{Z}_{S^c}\|_\infty \;\le\; |||Q_{S^cS}Q_{SS}^{-1}|||_\infty\left[\frac{\|W_S\|_\infty}{\lambda_j}+\frac{\|R_S\|_\infty}{\lambda_j}+1\right]+\frac{\|W_{S^c}\|_\infty}{\lambda_j}+\frac{\|R_{S^c}\|_\infty}{\lambda_j}.$$

Recalling Assumption (2), we obtain $|||Q_{S^cS}Q_{SS}^{-1}|||_\infty \le (1-\alpha)$, and hence, we have

$$\begin{aligned}\|\widetilde{Z}_{S^c}\|_\infty &\le (1-\alpha)\left[\frac{\|W_S\|_\infty}{\lambda_j}+\frac{\|R_S\|_\infty}{\lambda_j}+1\right]+\frac{\|W_{S^c}\|_\infty}{\lambda_j}+\frac{\|R_{S^c}\|_\infty}{\lambda_j}\\ &\le (1-\alpha)+(2-\alpha)\left[\frac{\|W_j^{S_j}\|_\infty}{\lambda_j}+\frac{\|R_j^{S_j}\|_\infty}{\lambda_j}\right].\end{aligned}$$

If both $\|W_j^{S_j}\|_\infty$ and $\|R_j^{S_j}\|_\infty$ are less than $\frac{\lambda_j\alpha}{4(2-\alpha)}$, as assumed, then

$$\|\widetilde{Z}_{S^c}\|_\infty \le (1-\alpha)+\frac{\alpha}{2}<1.$$

■

F.2. Proof for Proposition 9

Proof To prove the support of $\hat{\theta}_S$ is not strictly subset the true support X_S, it is sufficient to show that the maximum bias is bounded:

$$\|\widehat{\theta}_S-\theta_S^*\|_\infty \le \frac{\min_{t\in S}[\theta_S^*]_t}{2}.$$

From Lemma 7, we have, with a high probability,

$$\|\widehat{\theta}_S-\theta_S^*\|_\infty \le \|\widehat{\theta}_S-\theta_S^*\|_2 \le \frac{5}{\rho_{\min}}\sqrt{d}\;\lambda_j.$$

Therefore, if $\min_{t\in S}[\theta_S^*]_t \ge \frac{10}{\rho_{\min}}\sqrt{d}\;\lambda_j$,

$$\|\widehat{\theta}_S-\theta_S^*\|_\infty \le \frac{\min_{t\in S}[\theta_S^*]_t}{2}.$$

■

Appendix G. Proof for Lemmas

G.1. Proof for Lemma 4

Proof This lemma can be proved by the same manner developed for the special cases (Wainwright et al., 2006; Ravikumar et al., 2011). In addition, this proof is directly from Lemma 8 in Yang et al. (2015). And, we restate the proof in our framework. The main idea of the proof is the *primal-dual-witness* method which asserts that there is a solution to the dual problem $\widetilde{\theta}_{S_j}=\widehat{\theta}_{S_j}$ if the following Karush-Kuhn-Tucker (KKT) conditions are satisfied.

(a) We define $\widetilde{\theta}_{S_j}\in\Theta_{S_j}$, where $\Theta_{S_j}=\{\theta\in\mathbb{R}^{|S_j|}:\theta_{S^c}=0\}$ is the solution to the following optimization problem:

$$\widetilde{\theta}_{S_j} := \arg\min_{\theta\in\Theta_{S_j}}\mathcal{L}_j^{S_j}(\theta,\lambda_j)=\arg\min_{\theta\in\Theta_{S_j}}\{\ell_j^{S_j}(\theta;X^{1:n})+\lambda_j\|\theta\|_1\}. \tag{22}$$

(b) Define $\widetilde{Z}_j^{S_j}$ to be a sub-differential for the regularizer $\|\cdot\|_1$ evaluated at $\widetilde{\theta}_{S_j}$. For any $t\in T_j$ in Equation (12), $[\widetilde{Z}_j^{S_j}]_t=\text{sign}([\widetilde{\theta}_{S_j}]_t)$.

(c) For any $t\notin T_j$, $|[\widetilde{Z}_j^{S_j}]_t|<1$.

If conditions (a) to (c) are satisfied, $\widetilde{\theta}_{S_j}=\widehat{\theta}_{S_j}$ meaning that the solution to unrestricted problem (15) is the same as the solution to restricted problem (22) (See Ravikumar et al., 2011 for details).

In addition, if the sub-matrix of the Hessian $Q_{SS}^{S_j}$ is invertible, restricted problem (22) is strictly convex, and hence, $\widetilde{\theta}_{S_j}$ is unique. ■

G.2. Proof for Lemma 6

Proof In order to improve readability, we omit the superscript S_j if it is understood (i.e., $W_j=W_j^{S_j}$). Each entry of the sample score function W_j in Equation (18) has the form $W_{jt}=\frac{1}{n}\sum_{i=1}^n W_{jt}^{(i)}$ for any $t\in S:=\{k\in S_j\mid[\theta_{S_j}^*]_k\neq 0\}$. In addition, $W_{jt}=0$ for all $t\notin S$, since $[\theta_{S_j}^*]_t=0$ by the definition of S.

Hence simple calculation yields that, for any $t\in S$ and $i\in\{1,2,\cdots,n\}$,

$$W_{jt}^{(i)}=X_t^{(i)}X_j^{(i)}-\exp(\langle\theta_S^*,X_S^{(i)}\rangle)X_t^{(i)},$$

and $(|W_{jt}^{(i)}|)_{i=1}^n$ has mean 0 by the first-order optimality condition, $\mathbb{E}(X_j)=\mathbb{E}(\exp(\langle\theta_S^*,X_S\rangle))$.

Now, we show that $W_{jt}^{(i)}$ is bounded with a high probability given Assumption 3 by using Hoeffding's inequality. The both terms are bounded above $C_x^2\log^2\eta$ by Assumption 3. Therefore, $|W_{jt}^{(i)}|$ is bounded by $2C_x^2\log^2\eta$.

Applying the union bound and Hoeffding's inequality, we have

$$P(\|W_j\|_\infty>\delta)\le d\cdot\max_{t\in S}P(|W_{jt}|>\delta)\le 2d\cdot\exp\left(-\frac{2n\delta^2}{4C_x^4\log^4\eta}\right).$$

Suppose that $\delta=\frac{\lambda_j\alpha}{4(2-\alpha)}$ and $\lambda_j\ge\frac{4(2-\alpha)}{\alpha}\frac{2C_x^2\log^2\eta}{\sqrt{2}\kappa_1(n,p)}$. Then, we complete the proof:

$$P\left(\frac{\|W_j\|_\infty}{\lambda_j}>\frac{\alpha}{4(2-\alpha)}\right)\le 2d\cdot\exp\Big(-\frac{\alpha^2}{16(2-\alpha)^2}\frac{2n\lambda_j^2}{4C_x^4\log^4\eta}\Big)\le 2d\cdot\exp\left(-\frac{n}{\kappa_1(n,p)^2}\right). \tag{23}$$

■

G.3. Proof for Lemma 7

Proof In order to establish error bound $\|\widetilde{\theta}_S - \theta^*_S\| \le B$ for some radius B, several works (Meinshausen and Bühlmann, 2006; Wainwright et al., 2006; Ravikumar et al., 2011; Yang et al., 2015; Park and Raskutti, 2018) already proved that it suffices to show $F(u_S) > 0$ for all $u_S := \widetilde{\theta}_S - \theta^*_S$ such that $\|u_S\|_2 = B$ where

$$F(a) := \ell_j(\theta^*_S + a; X^{1:n}) - \ell_j(\theta^*_S; X^{1:n}) + \lambda_j(\|\theta^*_S + a\|_1 - \|\theta^*_S\|_1). \tag{24}$$

More specifically, since u_S is the minimizer of F and $F(0) = 0$ by the construction of Equation (24), $F(u_S) \le 0$. Note that F is convex, and therefore we have $F(u_S) < 0$.

Next we claim that $\|u_S\|_2 \le B$. In fact, if u_S lies outside the ball of radius B, then there exists $v \in (0,1)$ such that the convex combination $v \cdot u_S + (1-v)\cdot 0$ would lie on the boundary of the ball. However it contradicts the assumed strict positivity of F on the boundary because, by convexity,

$$F(v \cdot u_S + (1-v)\cdot 0) \le v \cdot F(u_S) + (1-v)\cdot 0 \le 0. \tag{25}$$

Thus it suffices to establish strict positivity of F on the boundary of the ball with radius $B := M_B \lambda_j \sqrt{d}$ where $M_B > 0$ is a parameter to be chosen later in the proof. Let $u_S \in \mathbb{R}^{|S|}$ be an arbitrary vector with $\|u_S\|_2 = B$. By the Taylor series expansion of F in (24),

$$F(u_S) = (W_S)^T u_S + u_S^T[\nabla^2 \ell_j(\theta^*_S + v u_S; X^{1:n})]u_S + \lambda_j(\|\theta^*_S + u_S\|_1 - \|\theta^*_S\|_1), \tag{26}$$

for some $v \in [0,1]$.

The first term in Equation (26) has the following bound: applying $\|W_S\|_\infty \le \frac{\lambda_j}{4}$ by assumption and $\|u_S\|_1 \le \sqrt{d}\|u_S\|_2 \le \sqrt{d}\cdot B$,

$$|(W_S)^T u_S| \le \|W_S\|_\infty \|u_S\|_1 \le \|W_S\|_\infty \sqrt{d}\|u_S\|_2 \le (\lambda_j\sqrt{d})^2 \frac{M_B}{4}.$$

The third term in Equation (26) has the following bound: Applying the triangle inequality,

$$\lambda_j(\|\theta^*_S + u_S\|_1 - \|\theta^*_S\|_1) \ge -\lambda_j\|u_S\|_1 \ge -\lambda_j\sqrt{d}\|u_S\|_2 = -M_B(\lambda_j\sqrt{d})^2.$$

Now we show the bound for the second term using the minimum eigenvalue of a matrix $\nabla^2 \ell_j(\theta^*_S + v u_S)$:

$$\begin{aligned}
q^* &:= \lambda_{\min}\left(\nabla^2 \ell_j(\theta^*_S + v u_S)\right) \\
&\ge \min_{v\in[0,1]} \lambda_{\min}\left(\nabla^2 \ell_j(\theta^*_S + v u_S)\right) \\
&\ge \lambda_{\min}\left(\nabla^2 \ell_j(\theta^*_S)\right) - \max_{v\in[0,1]} \left\| \frac{1}{n}\sum_{i=1}^n \exp\left(\langle \theta^*_S + v u_S, X_S^{(i)}\rangle\right) u_S^T X_S^{(i)} X_S^{(i)} (X_S^{(i)})^T \right\|_2 \\
&\ge \rho_{\min} - \max_{v\in[0,1]} \max_{y:\|y\|_2=1} \frac{1}{n}\sum_{i=1}^n \exp\left(\langle \theta^*_S + v u_S, X_S^{(i)}\rangle\right)\cdot\left(y^T X_S^{(i)}\right)^2 \cdot \left|u_S^T X_S^{(i)}\right|.
\end{aligned} \tag{27}$$

We first show the bound of the first term in Equation (27): Note that $\theta^*_S + v u_S$ is a linear (convex) combination of θ^*_S and $\widetilde{\theta}_S$. Hence, by Assumption 3 and Proposition 12, we obtain

$$\frac{1}{n}\sum_{i=1}^n \exp\left(\langle \theta^*_S + v u_S, X_S^{(i)}\rangle\right) \le C_x \log\eta.$$

Now, we bound the second term in Equation (27): Recall that $\|X_S^{(i)}\|_\infty \le C_x \log\eta$ for all i by Assumption 3. Recall $[u_S]_t = 0$ for $t \notin S$ by the primal-dual construction of (17). Applying $\|u_S\|_1 \le \sqrt{d}\|u_S\|_2 \le \sqrt{d}\cdot B$,

$$\left|u_S^T X_S^{(i)}\right| \le C_x \log(\eta)\sqrt{d}\|u_S\|_2 \le C_x \log(\eta)\cdot M_B \lambda_j d.$$

Lastly, it is clear that $\max_{y:\|y\|_2=1}(y^T X_S^{(i)})^2 \le \rho_{\max}$ by the definition of the maximum eigenvalue and Assumption 1. Together with the above bounds, we obtain

$$P\left(q^* \le \rho_{\min} - C_x^2 M_B \rho_{\max} d\lambda_j \log^2\eta\right) \le M\eta^{-2}.$$

For $\lambda_j \le \frac{\rho_{\min}}{2C_x^2 M_B \rho_{\max} d\log^2\eta}$, we have $q^* \ge \frac{\rho_{\min}}{2}$ with a high probability. Therefore,

$$F(u) \ge (\lambda_j\sqrt{n})^2\Big\{-\frac{1}{4}M_B + \frac{\rho_{\min}}{2}M_B^2 - M_B\Big\},$$

which is strictly positive for $M_B = \frac{5}{\rho_{\min}}$. Therefore, for $\lambda_j \le \frac{\rho_{\min}^2}{10C_x^2\rho_{\max} d\log^2\eta}$,

$$\|\widetilde{\theta}_S - \theta^*_S\|_2 \le \frac{5}{\rho_{\min}}\sqrt{d}\lambda_j.$$

■

G.4. Proof for Lemma 8

Proof To improve readability, we use $R_S = [R_j^{S_j}]_S$ where $S := \{k \in S_j \mid [\theta^*_{S_j}]_k \ne 0\}$. Then, each entry of $R_j^{S_j}$ in Equation (19) has the form $R_{jk} = \frac{1}{n}\sum_{i=1}^n R_{jk}^{(i)}$ for any $k \in S_j$, and it can be expressed as

$$\begin{aligned}
R_{jk} &= \frac{1}{n}\sum_{i=1}^n [\nabla^2 \ell_j(\theta^*_{S_j}; X^{1:n}) - \nabla^2 \ell_j(\bar{\theta}_{S_j}; X^{1:n})]_k^T (\widetilde{\theta}_{S_j} - \theta^*_{S_j}) \\
&= \frac{1}{n}\sum_{i=1}^n \left[\exp\left(\left\langle \theta^*_S, X_S^{(i)}\right\rangle\right) - \exp\left(\left\langle \bar{\theta}_S, X_S^{(i)}\right\rangle\right)\right]\left[X_S^{(i)}(X_S^{(i)})^T\right]_k^T \left(\widetilde{\theta}_S - \theta^*_S\right)
\end{aligned}$$

for $\bar{\theta}_S$, which is a point on the line between $\widetilde{\theta}_S$ and θ^*_S (i.e., $\bar{\theta}_S^{(t)} = v\cdot\widetilde{\theta}_S + (1-v)\cdot\theta^*_S$ for some $v \in [0,1]$). The second equality holds because $\theta^*_{S^c} = \widetilde{\theta}_{S^c} = (0,0,...,0) \in \mathbb{R}^{|S^c|}$.

Applying the mean value theorem again, we have,

$$R_{jk} = \frac{1}{n}\sum_{i=1}^n \left\{\exp\left(\left\langle \bar{\bar{\theta}}_S, X_S^{(i)}\right\rangle\right) X_k^{(i)}\right\}\left\{v(\widetilde{\theta}_S - \theta^*_S)^T X_S^{(i)}(X_S^{(i)})^T(\widetilde{\theta}_S - \theta^*_S)\right\}$$

for $\bar{\bar{\theta}}_{S_j}$ which is a point on the line between $\bar{\theta}_{S_j}$ and $\theta^*_{S_j}$ (i.e., $\bar{\bar{\theta}}_{S_j} = v \cdot \bar{\theta}_{S_j} + (1-v) \cdot \theta^*_{S_j}$ for $v \in [0,1]$).

Note that $\bar{\bar{\theta}}_{S_j}$ is a linear (convex) combination of θ^*_S and $\widetilde{\theta}_S$. Hence, from Assumption 3 and Proposition 12, we obtain,

$$\frac{1}{n}\sum_{i=1}^{n} \exp\left(\left\langle \bar{\bar{\theta}}_{S_j}, X^{(i)}_{S_j}\right\rangle\right) \le C_x \log \eta, \quad \text{and} \quad \max_{i,j} X^{(i)}_j < C_x \log \eta.$$

Therefore, we have $|R_{jk}| \le \rho_{\max} C_x^2 \log^2 \eta \|\widetilde{\theta}_S - \theta^*_S\|_2^2$ for all $j, k \in V$.

In Section G.3, we showed that $\|\widetilde{\theta}_S - \theta^*_S\|_2 \le \frac{5}{\rho_{\min}}\sqrt{d}\lambda_j$ for $\lambda_j \le \frac{\rho^2_{\min}}{10 C_x^2 \rho_{\max} d \log^2 \eta}$. Therefore, if $\lambda_j \le \frac{\rho^2_{\min}}{25 \cdot C_x^2 \rho_{\max} d \log^2 \eta} \frac{\alpha}{4(2-\alpha)}$, we obtain,

$$P\left(\|R_j\|_\infty > \frac{\alpha}{4(2-\alpha)}\lambda_j\right) \le P\left(\|R_j\|_\infty > 25 C_x^2 \lambda_j^2 \frac{\rho_{\max}}{\rho^2_{\min}} d \log^2 \eta\right) = 0.$$

Therefore, we have,

$$\|R_j\|_\infty \le \frac{\alpha}{4(2-\alpha)}\lambda_j$$

■

G.5. Proof for Lemma 10

Proof Conditioning on the sets ζ_2, ζ_3, and ζ_4, we provide the following results for different two cases:

(i) For any $j \in \{1, 2, ..., p-1\}$, and $X_S = X_{1:(j-1)}$, we have $\frac{\mathbb{E}(X_j^2)}{\mathbb{E}(f(\mathbb{E}(X_j|X_S)))} = 1$. Therefore, for $k = \pi_j$, we have the following probability bound:

$$\begin{aligned}
& P\left(|\widehat{\mathcal{S}}(j,k) - \mathcal{S}(j,k)| < \frac{M_{min}}{2} \middle| \zeta_2, \zeta_3, \zeta_4\right) \\
= \; & P\left(\left|\frac{\widehat{\mathbb{E}}(X_k^2)}{\widehat{\mathbb{E}}(f(\widehat{\mathbb{E}}(X_k \mid X_S)))} - \frac{\mathbb{E}(X_k^2)}{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))}\right| < \frac{M_{min}}{2} \middle| \zeta_2, \zeta_3, \zeta_4\right) \\
\ge \; & P\Bigg(\frac{\mathbb{E}(X_k^2) + \epsilon_1}{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S))) - 2\epsilon_1} - \frac{\mathbb{E}(X_k^2))}{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))} < \frac{M_{min}}{2} \; and \\
& \qquad \frac{\mathbb{E}(X_k^2))}{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))} - \frac{\mathbb{E}(X_k^2)) - \epsilon_1}{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S))) + 2\epsilon_1} < \frac{M_{min}}{2}\Bigg) \\
\ge \; & P\left(\epsilon_1 < \frac{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))M_{\min}}{2(M_{\min}+3)}\right) \\
\ge \; & P\left(\epsilon_1 < \frac{\mathbb{E}(X_k^2)M_{\min}}{2(M_{\min}+3)(M_{\min}+1)}\right).
\end{aligned}$$

(ii) For $j \in \{1, 2, ..., p-1\}$, $k \in \{\pi_{j+1}, ..., \pi_p\}$ having parent π_j, and $X_S = X_{1:(j-1)}$, we have $\mathbb{E}(X_k^2) > (1 + M_{\min})\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))$ by Assumption 4. In addition, some elementary

but complicated computations yield

$$\begin{aligned}
& P\left(\left|\widehat{\mathcal{S}}(j,k) - \mathcal{S}(j,k)\right| < \frac{M_{min}}{2} \middle| \zeta_2, \zeta_3, \zeta_4\right) \\
\ge \; & P\left(\epsilon_1 < \frac{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))^2 M_{\min}}{4\mathbb{E}(X_k^2) + 2\mathbb{E}(f(\mathbb{E}(X_k \mid X_S))) + 2\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))M_{\min}}\right) \\
\ge \; & P\left(\epsilon_1 < \frac{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))^2 M_{\min}(1+M_{\min})}{4\mathbb{E}(X_k^2)(1+M_{\min}) + 2\mathbb{E}(X_k^2) + 2M_{\min}\mathbb{E}(X_k^2)}\right) \\
\ge \; & P\left(\epsilon_1 < \frac{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))^2 M_{\min}(1+M_{\min})}{6(1+M_{\min})\mathbb{E}(X_k^2)}\right) \\
\ge \; & P\left(\epsilon_1 < \frac{M_{\min}}{6}\frac{\mathbb{E}(f(\mathbb{E}(X_k \mid X_S)))^2}{\mathbb{E}(X_k^2)}\right).
\end{aligned}$$

Therefore $P(\zeta_1 \mid \zeta_2, \zeta_3, \zeta_4) = 0$ if ϵ_1 is sufficiently small enough. For any node j, any set $S_j \in \{\pi_1, \pi_{1:2}, ..., \pi_{1:(j-1)}\}$, and $k \in \{\pi_j, \pi_{j+1}, ..., \pi_p\}$,

$$\epsilon_1 < \min\left\{\frac{\mathbb{E}(X_k^2)M_{\min}}{2(M_{\min}+3)(M_{\min}+1)}, \frac{M_{\min}}{6}\frac{\mathbb{E}(f(\mathbb{E}(X_k \mid X_{S_j})))^2}{\mathbb{E}(X_k^2)}\right\}.$$

■

G.6. Proof for Lemma 11

The proof for Lemma 11 is closely related to the proof in Appendix C. Hence, for brevity, we do not present the details of the proof already shown in Appendix C.

(i) $P(\zeta_2^c) \le 2p \cdot \exp\left\{-\frac{n\epsilon_1^2}{2C_x^2 \log^2 \eta}\right\}$.

Proof Using Hoeffding's inequality given Assumption 3, for any $\epsilon > 0$ and $j \in V$,

$$P\left(\left|\widehat{\mathbb{E}}(X_j^2) - \mathbb{E}(X_j^2)\right| > \epsilon_1\right) \le 2 \cdot \exp\left\{-\frac{n\epsilon_1^2}{2C_x^2 \log^2 \eta}\right\}. \tag{28}$$

Hence, using the union bound, we have

$$P\left(\max_{j \in V}\left|\widehat{\mathbb{E}}(X_j^2) - \mathbb{E}(X_j^2)\right| > \epsilon_1\right) \le 2p \cdot \exp\left\{-\frac{n\epsilon_1^2}{2C_x^2 \log^2 \eta}\right\}.$$

■

(ii) $P(\zeta_3^c) \le 2p.d \cdot \exp\left(-\frac{n}{\kappa_1(n,p)^2}\right) + 2p \cdot \exp\left\{-\frac{n\epsilon_1^2}{D_{\max}\log^4 \eta}\right\}$ for some constants $D_{\max} > 0$.

Proof We restate the condition in the set ζ_3 as

$$\left|\frac{1}{n}\sum_{i=1}^{n} f\big(\widehat{\mathbb{E}}(X_j \mid X^{(i)}_{S_j})\big) - \mathbb{E}\big(f\big(\mathbb{E}(X_j \mid X_{S_j})\big)\big)\right| < \epsilon_1.$$

In order to apply Hoeffding's inequality, we first show the bound for $\widetilde{E}(X_j \mid X_{S_j})$. Recall that $[\theta^*]_{S^c}$ and $[\widetilde{\theta}]_{S^c} = (0, 0, ..., 0) \in \mathbb{R}^{|S^c|}$ by the definition of S, and $|S| \leq d$. In Appendix G.3, we showed that $\|\widetilde{\theta}_S - \theta^*_S\|_2 \leq \frac{5}{\rho_{\min}}\sqrt{d}\lambda_j$ for $\lambda_j \leq \frac{\rho^2_{\min}}{10C^2_x \rho_{\max} d \log^2 \eta}$ with a high probability. Therefore, given Assumption 3, for all $i \in \{1, 2, ..., n\}$,

$$\begin{aligned}
\exp(\langle \widehat{\theta}_{S_j}, X^{(i)}_{S_j} \rangle) &= \exp(\langle \widehat{\theta}_{S_j} - \theta^*_{S_j}, X^{(i)}_{S_j} \rangle) \cdot \exp(\langle \theta^*_{S_j}, X^{(i)}_{S_j} \rangle) \\
&\leq \exp(\|\widehat{\theta}_S - \theta^*_S\|_2 \|X^{(i)}_S\|_2 \rangle) \cdot \exp(\langle \theta^*_S, X^{(i)}_S \rangle) \\
&\leq \exp\left\{\frac{5C_x d\ \lambda_j}{\rho_{\min}} \|X^{(i)}_S\|_\infty\right\} \cdot \exp(\langle \theta^*_S, X^{(i)}_S \rangle) \\
&\leq \exp\left\{\frac{5C_x d\ \lambda_j}{\rho_{\min}} \log(\eta)\right\} \cdot C_x \log \eta \\
&\leq \exp\left\{\frac{\rho_{\min}}{2C_x \rho_{\max} \log \eta}\right\} \cdot C_x \log \eta.
\end{aligned}$$

Therefore,

$$f\big(\widehat{\mathbb{E}}(X^{(i)}_j \mid X^{(i)}_{S_j})\big) \leq C^2_x \cdot \exp\left\{\frac{\rho_{\min}}{C_x \rho_{\max}}\right\} \log^2 \eta + C_x \cdot \exp\left\{\frac{\rho_{\min}}{2C_x \rho_{\max}}\right\} \log \eta.$$

Hence there exists a positive constant $D_1 > 0$ such that for all $i \in \{1, 2, ..., n\}$,

$$f\big(\widehat{\mathbb{E}}(X_j \mid X^{(i)}_{S_j})\big) \leq D_1 \log^2 \eta.$$

Applying Hoeffding's inequality,, for any $\epsilon_1 > 0$ and any $j \in V$,

$$P\left(\left|\widehat{\mathbb{E}}(f(\widetilde{E}(X_j \mid X_{S_j}))) - \mathbb{E}(f(\widetilde{E}(X_j \mid X_{S_j})))\right| > \epsilon_1\right) \leq 2 \cdot \exp\left\{-\frac{2n\epsilon_1^2}{D_1^2 \log^4 \eta}\right\}. \quad (29)$$

Hence, there exist some constants $D_{\max} > 0$ such that

$$P\left(\max_{j \in V} \zeta^c_3\right) \leq 1 - 2p.d \cdot \exp\left(-\frac{n}{\kappa_1(n,p)^2}\right) - 2p \cdot \exp\left\{-\frac{n\epsilon_1^2}{D_{\max} \log^4 \eta}\right\}.$$

■

(iii) $P(\zeta^c_4) = 0$.

Proof

We restate the condition in the set ζ_4 as

$$\left|\mathbb{E}\left(f(\mathbb{E}(X_j \mid X_{S_j})) - f(\widehat{\mathbb{E}}(X_j \mid X_{S_j}))\right)\right| < \epsilon_1.$$

By the mean-value theorem, for some $v \in [0, 1]$,

$$\begin{aligned}
&f\big(\widehat{\mathbb{E}}(X_j \mid X_{S_j})\big) - f\big(\mathbb{E}(X_j \mid X_{S_j})\big) \\
&\quad = f'\big(v\widehat{\mathbb{E}}(X_j \mid X_{S_j}) + (1-v)\mathbb{E}(X_j \mid X_{S_j})\big)\big(\widehat{\mathbb{E}}(X_j \mid X_{S_j}) - \mathbb{E}(X_j \mid X_{S_j})\big) \\
&\quad = 2\big(v\widehat{\mathbb{E}}(X_j \mid X_{S_j}) + (1-v)\mathbb{E}(X_j \mid X_{S_j}) + 1/2\big)\big(\widehat{\mathbb{E}}(X_j \mid X_{S_j}) - \mathbb{E}(X_j \mid X_{S_j})\big).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}(f\big(\widehat{\mathbb{E}}(X_j \mid X_{S_j})\big) - f\big(\mathbb{E}(X_j \mid X_{S_j})\big)) \\
&\quad = f'\big(v\widehat{\mathbb{E}}(X_j \mid X_{S_j}) + (1-v)\mathbb{E}(X_j \mid X_{S_j})\big)\big(\widehat{\mathbb{E}}(X_j \mid X_{S_j}) - \mathbb{E}(X_j \mid X_{S_j})\big) \\
&\quad = 2\big(v\widehat{\mathbb{E}}(X_j \mid X_{S_j}) + (1-v)\mathbb{E}(X_j \mid X_{S_j}) + 1/2\big)\big(\widehat{\mathbb{E}}(X_j \mid X_{S_j}) - \mathbb{E}(X_j \mid X_{S_j})\big) \\
&\quad \leq \max\left|2\big(v\widehat{\mathbb{E}}(X_j \mid X_{S_j}) + (1-v)\mathbb{E}(X_j \mid X_{S_j}) + 1/2\big)\right| \cdot \mathbb{E}\left(\widehat{\mathbb{E}}(X_j \mid X_{S_j}) - \mathbb{E}(X_j \mid X_{S_j})\right) \\
&\quad = 0
\end{aligned}$$

In the same manner, $\mathbb{E}(f\big(\mathbb{E}(X_j \mid X_{S_j})\big) - f\big(\widehat{\mathbb{E}}(X_j \mid X_{S_j})\big)) \leq 0$. This completes the proof.

■

References

Peter Bühlmann, Jonas Peters, Jan Ernest, et al. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

David M Chickering, Dan Geiger, David Heckerman, et al. Learning bayesian networks is np-hard. Technical report, Citeseer, 1994.

David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.

David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003.

Kenji Doya. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

Mathias Drton, Wenyu Chen, and Y Samuel Wang. On causal discovery with equal variance assumption. *arXiv preprint arXiv:1807.03419*, 2018.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4), 2009.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

Michael Friendly. *Lahman: Sean 'Lahman' Baseball Database*, 2017. URL `https://CRAN.R-project.org/package=Lahman`. R package version 6.0-0.

Morten Frydenberg. The chain graph markov property. *Scandinavian Journal of Statistics*, pages 333–353, 1990.

Asish Ghoshal and Jean Honorio. Information-theoretic limits of bayesian network structure learning. In *Artificial Intelligence and Statistics*, pages 767–775, 2017a.

Asish Ghoshal and Jean Honorio. Learning identifiable gaussian bayesian networks in polynomial time and sample complexity. In *Advances in Neural Information Processing Systems*, pages 6457–6466, 2017b.

Asish Ghoshal and Jean Honorio. Learning linear structural equation models in polynomial time and sample complexity. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 1466–1475, 2018.

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems*, pages 689–696, 2009.

David I Inouye, Eunho Yang, Genevera I Allen, and Pradeep Ravikumar. A review of multivariate distributions for count data derived from the poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3), 2017.

Jinzhu Jia, Fang Xie, and Lihu Xu. Sparse poisson regression with penalized weighted score function. *arXiv preprint arXiv:1703.03965*, 2017.

Jeffrey O Kephart and Steve R White. Directed-graph epidemiological models of computer viruses. In *Research in Security and Privacy, 1991. Proceedings., 1991 IEEE Computer Society Symposium on*, pages 343–359. IEEE, 1991.

Steffen L Lauritzen. *Graphical models*. Oxford University Press, 1996.

Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1): 3065–3105, 2014.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.

Gunwoong Park and Hyewon Park. Identifiability of generalized hypergeometric distribution (ghd) directed acyclic graphical models. In *Proceedings of Machine Learning Research*, pages 158–166, 2019.

Gunwoong Park and Garvesh Raskutti. Learning large-scale poisson dag models based on overdispersion scoring. In *Advances in Neural Information Processing Systems*, pages 631–639, 2015.

Gunwoong Park and Garvesh Raskutti. Learning quadratic variance function (qvf) dag models via overdispersion scoring (ods). *Journal of Machine Learning Research*, 18(224): 1–44, 2018.

Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

Jonas Peters, Joris Mooij, Dominik Janzing, and Bernhard Schölkopf. Identifiability of causal graphs using functional models. pages 589–598, Corvallis, OR, USA, July 2011. AUAI Press.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, Bin Yu, et al. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-Gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, 7:2003–2030, 2006.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, and Kenneth Bollen. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research*, 12(Apr):1225–1248, 2011.

Peter Spirtes. Directed cyclic graphical representations of feedback models. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 491–498. Morgan Kaufmann Publishers Inc., 1995.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Ioannis Tsamardinos and Constantin F Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the ninth international workshop on Artificial Intelligence and Statistics*. Morgan Kaufmann Publishers: Key West, FL, USA, 2003.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pages 436–463, 2013.

Martin J Wainwright, John D Lafferty, and Pradeep K Ravikumar. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *Advances in Neural Information Processing Systems*, pages 1465–1472, 2006.

Ying-Wooi Wan, Genevera I Allen, Yulia Baker, Eunho Yang, Pradeep Ravikumar, Matthew Anderson, and Zhandong Liu. Xmrf: an r package to fit markov networks to high-throughput genetics data. *BMC systems biology*, 10(3):69, 2016.

Eunho Yang, Pradeep K Ravikumar, Genevera I Allen, and Zhandong Liu. On poisson graphical models. In *Advances in Neural Information Processing Systems*, pages 1718–1726, 2013.

Eunho Yang, Pradeep Ravikumar, Genevera I Allen, and Zhandong Liu. Graphical models via univariate exponential family distributions. *Journal of Machine Learning Research*, 16(1):3813–3847, 2015.

Jiji Zhang and Peter Spirtes. The three faces of faithfulness. *Synthese*, 193(4):1011–1027, 2016.

Journal of Machine Learning Research 20 (2019) 1-7 Submitted 1/19; Revised 4/19; Published 5/19

PyOD: A Python Toolbox for Scalable Outlier Detection

Yue Zhao ZHAOY@CMU.EDU
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Zain Nasrullah ZNASRULLAH@CS.TORONTO.EDU
University of Toronto
Toronto, ON M5S 2E4, Canada

Zheng Li JK_ZHENGLI@HOTMAIL.COM
Northeastern University Toronto
Toronto, ON M5X 1E2, Canada

Editor: Alexandre Gramfort

Abstract

`PyOD` is an open-source Python toolbox for performing scalable outlier detection on multivariate data. Uniquely, it provides access to a wide range of outlier detection algorithms, including established outlier ensembles and more recent neural network-based approaches, under a single, well-documented API designed for use by both practitioners and researchers. With robustness and scalability in mind, best practices such as unit testing, continuous integration, code coverage, maintainability checks, interactive examples and parallelization are emphasized as core components in the toolbox's development. `PyOD` is compatible with both Python 2 and 3 and can be installed through Python Package Index (PyPI) or `https://github.com/yzhao062/pyod`.

Keywords: anomaly detection, outlier detection, outlier ensembles, neural networks, machine learning, data mining, Python

1. Introduction

Outlier detection, also known as anomaly detection, refers to the identification of rare items, events or observations which differ from the general distribution of a population. Since the ground truth is often absent in such tasks, dedicated outlier detection algorithms are extremely valuable in fields which process large amounts of unlabelled data and require a means to reliably perform pattern recognition (Akoglu et al., 2012). Industry applications include fraud detection in finance (Ahmed et al., 2016), fault diagnosis in mechanics (Shin et al., 2005), intrusion detection in network security (Garcia-Teodoro et al., 2009) and pathology detection in medical imaging (Baur et al., 2018).

To help approach these problems, established outlier detection packages exist in various programming languages such as `ELKI Data Mining` (Achtert et al., 2010) and `RapidMiner` (Hofmann and Klinkenberg, 2013) in Java and `outliers` (Komsta and Komsta, 2011) in R. However, Python, one of the most important languages in machine learning, still lacks a dedicated toolkit for outlier detection. Existing implementations either stand as single algorithm tools like `PyNomaly` (Constantinou, 2018) or exist as part of a general-purpose framework like `scikit-learn` (Pedregosa et al., 2011) which does not cater specifically to anomaly detection. To fill this gap, we propose and implement `PyOD`—a comprehensive Python toolbox for scalable outlier detection.

∗. Work initialized while at University of Toronto.

©2019 Yue Zhao, Zain Nasrullah and Zheng Li.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/19-011.html`.

Method	Category	JIT Enabled	Multi-core
LOF (Breunig et al., 2000)	Proximity	No	Yes
kNN (Ramaswamy et al., 2000)	Proximity	No	Yes
AvgkNN (Angiulli and Pizzuti, 2002)	Proximity	No	Yes
CBLOF (He et al., 2003)	Proximity	Yes	No
OCSVM (Schölkopf et al., 2001)	Linear Model	No	No
LOCI (Papadimitriou et al., 2003)	Proximity	Yes	No
PCA (Shyu et al., 2003)	Linear Model	No	No
MCD (Hardin and Rocke, 2004)	Linear Model	No	No
Feature Bagging (Lazarevic and Kumar, 2005)	Ensembling	No	Yes
ABOD (Kriegel et al., 2008)	Proximity	Yes	No
Isolation Forest (Liu et al., 2008)	Ensembling	No	Yes
HBOS (Goldstein and Dengel, 2012)	Proximity	Yes	No
SOS (Janssens et al., 2012)	Proximity	Yes	No
AutoEncoder (Sakurada and Yairi, 2014)	Neural Net	Yes	No
AOM (Aggarwal and Sathe, 2015)	Ensembling	No	No
MOA (Aggarwal and Sathe, 2015)	Ensembling	No	No
SO-GAAL (Liu et al., 2019)	Neural Net	No	No
MO-GAAL (Liu et al., 2019)	Neural Net	No	No
XGBOD (Zhao and Hryniewicki, 2018b)	Ensembling	No	Yes
LSCP (Zhao et al., 2019)	Ensembling	No	No

Table 1: Select outlier detection models in `PyOD`

Compared to existing libraries, `PyOD` has six distinct advantages. Firstly, it contains more than 20 algorithms which cover both classical techniques such as local outlier factor and recent neural network architectures such as autoencoders or adversarial models. Secondly, `PyOD` implements combination methods for merging the results of multiple detectors and outlier ensembles which are an emerging set of models. Thirdly, `PyOD` includes a unified API, detailed documentation and interactive examples across all algorithms for clarity and ease of use. Fourthly, all models are covered by unit testing with cross platform continuous integration, code coverage and code maintainability checks. Fifthly, optimization instruments are employed when possible: just-in-time (JIT) compilation and parallelization are enabled in select models for scalable outlier detection. Lastly, `PyOD` is compatible with both Python 2 and 3 across major operating systems (`Windows`, `Linux` and `MacOS`). Popular detection algorithms implemented in `PyOD` (version 0.7.0) are summarized in Table 1.

2. Project Focus

Build robustness. Continuous integration tools (*Travis CI*, *AppVeyor* and *CircleCI*) are leveraged to conduct automated testing under various versions of Python and operating systems. Tests are executed daily, when commits are made to the development and master branches, or when a pull request is opened.
Quality assurance. The project follows the `PEP8` standard; maintainability is ensured by *CodeClimate*, an automated code review and quality assurance tool. Additionally, code blocks with high cognitive complexity are actively refactored and a standard set of unit tests exist to ensure 95% overall code coverage. These design choices enhance collaboration and this standard is enforced on all pull requests for consistency.
Community-based development. `PyOD`'s code repository is hosted on GitHub[1] to facilitate collaboration. At the time of this writing, eight contributors have helped develop the code base and others have contributed in the form of bug reports and feature requests.
Documentation and examples. Comprehensive documentation is developed using `sphinx` and `numpydoc` and rendered using *Read the Docs*[2]. It includes detailed API references, an installation guide, code examples and algorithm benchmarks. An interactive Jupyter notebook is also hosted on *Binder* allowing others to experiment prior to installation.
Project relevance. `PyOD` has been used in various academic and commercial projects (Zhao and Hryniewicki, 2018a; Ramakrishnan et al., 2019; Krishnan and Wu, 2019). The GitHub repository receives more than 10,000 monthly views and its PyPI downloads exceed 6,000 per month.

3. Library Design and Implementation

`PyOD` is compatible with both Python 2 and 3 using `six`; it relies on `numpy`, `scipy` and `scikit-learn` as well. Neural networks such as autoencoders and SO_GAAL additionally require `Keras`. To enhance model scalability, select algorithms (Table 1) are optimized with JIT using `numba`. Parallelization for multi-core execution is also available for a set of algorithms using `joblib`. Inspired by `scikit-learn`'s API design (Buitinck et al., 2013), all implemented outlier detection algorithms inherit from a base class with the same interface: (i) `fit` processes the train data and computes the necessary statistics; (ii) `decision_function` generates raw outlier scores for unseen data after the model is fitted; (iii) `predict` returns a binary class label corresponding to each input sample instead of the raw outlier score and (iv) `predict_proba` offers the result as a probability using either normalization or Unification (Kriegel et al., 2011). Within this framework, new models are easy to implement by taking advantage of inheritance and polymorphism. Base methods can then be overridden as necessary.

Once a detector has been fitted on a training set, the corresponding train outlier scores and binary labels are accessible using its `decision_scores_` and `labels_` attributes. Once fitted, the model's `predict`, `decision_function` and `predict_proba` methods may be called for use on new data. An example showcasing the ease of use of this API is shown in Code Snippet 1 with an angle-based outlier detector (ABOD).

1. https://github.com/yzhao062/pyod
2. https://pyod.readthedocs.io


```
>>> from pyod.models.abod import ABOD
>>> from pyod.utils.data import generate_data
>>> from pyod.utils.data import evaluate_print
>>> from pyod.utils.example import visualize
>>>
>>> X_train, y_train, X_test, y_test = generate_data(\
...    n_train=200, n_test=100, n_features=2)
>>>
>>> clf = ABOD(method="fast")                        # initialize detector
>>> clf.fit(X_train)
>>>
>>> y_test_pred = clf.predict(X_test)                # binary labels
>>> y_test_scores = clf.decision_function(X_test)    # raw outlier scores
>>> y_test_proba = clf.predict_proba(X_test)         # outlier probability
>>>
>>> evaluate_print("ABOD", y_test, y_test_scores)    # performance evaluation
ABOD Performance; ROC: 0.934; Precision at n: 0.902
>>>
>>> visualize(y_test, y_test_scores)                 # prediction visualization
```

Code Snippet 1: Demo of PyOD API with the ABOD detector

In addition to the outlier detection algorithms, a set of helper and utility functions (`generate_data`, `evaluate_print` and `visualize`) are included in the library for quick model exploration and evaluation. The two-dimensional artificial data used in the example is created by `generate_data` which generates inliers from a Gaussian distribution and outliers from a uniform distribution. An example of using `visualize` is shown in Figure 1.

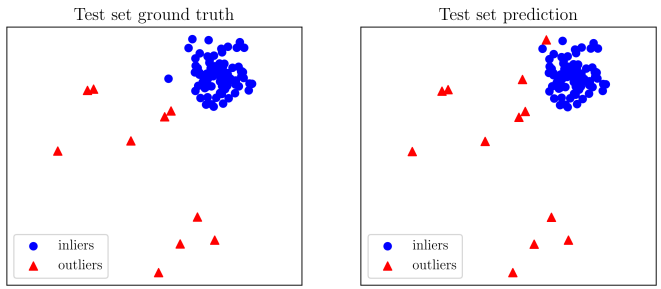


Figure 1: Demonstration of using `PyOD` in visualizing prediction result

4. Conclusion and Future Plans

This paper presents PyOD, a comprehensive toolbox built in Python for scalable outlier detection. It includes more than 20 classical and emerging detection algorithms and is being used in both academic and commercial projects. As avenues for future work, we plan to enhance the toolbox by implementing models that work well with time series and geospatial data, improving computational efficiency through distributed computing and addressing engineering challenges such as handling sparse matrices or memory limitations.

Acknowledgments

We thank the editor and anonymous reviewers for their constructive comments. This work was partly supported by Mitacs through the Mitacs Accelerate program.

References

Elke Achtert, Hans-Peter Kriegel, Lisa Reichert, Erich Schubert, Remigius Wojdanowski, and Arthur Zimek. Visual evaluation of outlier detection models. In *International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 396–399. Springer, 2010.

Charu C Aggarwal and Saket Sathe. Theoretical foundations and algorithms for outlier ensembles. *ACM SIGKDD Explorations Newsletter*, 17(1):24–47, 2015.

Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

Leman Akoglu, Hanghang Tong, Jilles Vreeken, and Christos Faloutsos. Fast and reliable anomaly detection in categorical data. In *International Conference on Information and Knowledge Management (CIKM)*, pages 415–424. ACM, 2012.

Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 15–27. Springer, 2002.

Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. *arXiv preprint arXiv:1804.04488*, 2018.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM SIGMOD Record*, volume 29, pages 93–104, 2000.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

Valentino Constantinou. Pynomaly: Anomaly detection using local outlier probabilities (LoOP). *The Journal of Open Source Software (JOSS)*, 3:845, 2018.

Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2):18–28, 2009.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. *Annual German Conference on Artificial Intelligence (KI-2012)*, pages 59–63, 2012.

Johanna Hardin and David M Rocke. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics & Data Analysis*, 44(4):625–638, 2004.

Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.

Markus Hofmann and Ralf Klinkenberg. *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013.

JHM Janssens, Ferenc Huszár, EO Postma, and HJ van den Herik. Stochastic outlier selection. Technical report, Technical report TiCC TR 2012-001, Tilburg University, Tilburg Center for Cognition and Communication, Tilburg, The Netherlands, 2012.

Lukasz Komsta and Maintainer Lukasz Komsta. *Package outliers*, 2011. URL `https://cran.r-project.org/web/packages/outliers/outliers.pdf`. R package version 0.14.

Hans-Peter Kriegel, Arthur Zimek, et al. Angle-based outlier detection in high-dimensional data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 444–452. ACM, 2008.

Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek. Interpreting and unifying outlier scores. In *SIAM International Conference on Data Mining (SDM)*, pages 13–24. SIAM, 2011.

Sanjay Krishnan and Eugene Wu. Alphaclean: Automatic generation of data cleaning pipelines. *arXiv preprint arXiv:1904.11827*, 2019.

Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 157–166. ACM, 2005.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *International Conference on Data Mining (ICDM)*, pages 413–422. IEEE, 2008.

Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2019.

Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *International Conference on Data Engineering (ICDE)*, pages 315–326. IEEE, 2003.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.

Jagdish Ramakrishnan, Elham Shaabani, Chao Li, and Mátyás A Sustik. Anomaly detection for an e-commerce pricing system. *arXiv preprint arXiv:1902.09566*, 2019.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, volume 29, pages 427–438, 2000.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Pacific Rim International Conference on Artificial Intelligence (PRICAI), Workshop on Machine Learning for Sensory Data Analysis (MLSDA)*, 2014.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

Hyun Joon Shin, Dong-Hwan Eom, and Sung-Shick Kim. One-class support vector machinesan application in machine fault detection and classification. *Computers & Industrial Engineering*, 48(2):395–408, 2005.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, Department of Electrical and Computer Engineering, University of Miami, 2003.

Yue Zhao and Maciej K Hryniewicki. DCSO: dynamic combination of detector scores for outlier ensembles. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Outlier Detection De-constructed Workshop*, London, UK, August 2018a.

Yue Zhao and Maciej K Hryniewicki. XGBOD: improving supervised outlier detection with unsupervised representation learning. In *International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, July 2018b. IEEE.

Yue Zhao, Zain Nasrullah, Maciej K Hryniewicki, and Zheng Li. LSCP: locally selective combination in parallel outlier ensembles. In *SIAM International Conference on Data Mining (SDM)*, pages 585–593, Calgary, Canada, May 2019. SIAM.

Journal of Machine Learning Research 20 (2019) 1-22 Submitted 9/15; Revised 3/19; Published 6/19

Relative Error Bound Analysis for Nuclear Norm Regularized Matrix Completion

Lijun Zhang ZHANGLJ@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China

Tianbao Yang TIANBAO-YANG@UIOWA.EDU
Department of Computer Science
The University of Iowa, Iowa City, IA 52242, USA

Rong Jin JINRONG.JR@ALIBABA-INC.COM
Machine Intelligence Technology
Alibaba Group, Bellevue, WA 98004, USA

Zhi-Hua Zhou ZHOUZH@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China

Editor: Nathan Srebro

Abstract

In this paper, we develop a *relative* error bound for nuclear norm regularized matrix completion, with the focus on the completion of full-rank matrices. Under the assumption that the top eigenspaces of the target matrix are incoherent, we derive a relative upper bound for recovering the best low-rank approximation of the unknown matrix. Although multiple works have been devoted to analyzing the recovery error of full-rank matrix completion, their error bounds are usually additive, making it impossible to obtain the perfect recovery case and more generally difficult to leverage the skewed distribution of eigenvalues. Our analysis is built upon the optimality condition of the regularized formulation and existing guarantees for low-rank matrix completion. To the best of our knowledge, this is the first relative bound that has been proved for the regularized formulation of matrix completion.

Keywords: matrix completion, nuclear norm regularization, least squares, low-rank, full-rank, relative error bound

1. Introduction

Matrix completion is concerned with the problem of recovering an unknown matrix from a small fraction of its entries (Candès and Tao, 2010). Recently, the problem of low-rank matrix completion has received a great deal of interests due to the theoretical advances (Candès and Recht, 2009; Keshavan et al., 2010a), as well as its application to a wide range of real-world problems, including collaborative filtering (Goldberg et al., 1992), sensor networks (Biswas et al., 2006), computer vision (Cabral et al., 2011), and machine learning (Jalali et al., 2011).

Let A be an unknown matrix of size $m \times n$, and without loss of generality, we assume $m \leq n$. The information available about A is a sampled set of entries A_{ij}, $(i,j) \in \Omega$, where

©2019 Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/15-504.html.

Ω is a subset of the complete set of entries $[m] \times [n]$. Our goal is to recover A as precisely as possible. In a seminal work, Candès and Recht (2009) assume that A is low-rank, and propose to recover A from the observed entries in Ω by solving the following nuclear norm minimization problem

$$\begin{aligned} \min \quad & \|B\|_* \\ \text{s.t.} \quad & B_{ij} = A_{ij} \ \forall (i,j) \in \Omega. \end{aligned} \tag{1}$$

Under the incoherence condition, they prove that with a high probability the solution to (1) yields a perfect reconstruction of A, provided that a sufficiently large number of entries are observed randomly. When A is of full rank, a similar nuclear norm minimization problem has been proposed. Suppose $A = Z + N$, where Z is a low-rank matrix to recover, and N is the residual matrix. Candès and Plan (2010) introduce the following problem for recovering A

$$\begin{aligned} \min \quad & \|B\|_* \\ \text{s.t.} \quad & \sqrt{\textstyle\sum_{(i,j)\in\Omega}(B_{ij} - A_{ij})^2} \leq \delta \end{aligned} \tag{2}$$

where δ is an upper bound for $\sqrt{\sum_{(i,j)\in\Omega} N_{ij}^2}$. Although a relative error bound has been established for (2) when δ is large enough (Candès and Plan, 2010), the high computational cost with solving the optimization problem in (2), mostly due to the constraint and non-smooth objective function, makes it practically less attractive.

An alternative approach to (2) for matrix completion is to solve a nuclear norm regularized least squares problem

$$\min_{B \in \mathbb{R}^{m\times n}} \quad \frac{1}{2} \sum_{(i,j)\in\Omega} (B_{ij} - A_{ij})^2 + \lambda \|B\|_*. \tag{3}$$

This is the approach that is favored by practitioners because it can be solved significantly more efficiently than (2). In fact, a number of efficient optimization methods have been designed (Ji and Ye, 2009; Toh and Yun, 2010; Pong et al., 2010; Zhang et al., 2012; Hsieh and Olsen, 2014). Using the accelerated gradient method (Nesterov, 2013), the convergence rate for solving (3) is $O(1/T^2)$, where T is number of iterations, and can be even boosted to a linear convergence under mild conditions (Hou et al., 2013). In contrast, the convergence rate for (2) could be as low as $O(1/\sqrt{T})$.

Although (3) is computation-friendly, its recovery guarantee remains unclear. One may argue that (2) and (3) are equivalent by setting δ and λ appropriately, but the exact correspondence between them is unknown in general. To bridge the gap between practice and theory, in this paper we provide a *relative* error bound for the regularized formulation in (3). More specifically, assume A is a matrix of full rank to be recovered. Let A_r be the best rank-r approximation of A, and $\widehat{A}$ be the matrix recovered from the observed entries in Ω. A relative upper bound takes the following form

$$\|\widehat{A} - A_r\|_F \leq U(r, m, n, |\Omega|)\|A - A_r\|_F \tag{4}$$

where $U(\cdot)$ is a function of r, m, n and $|\Omega|$. [1] Note that this kind of bounds is very popular in compressive sensing (Cohen et al., 2009) and low-rank matrix approximation (Boutsidis

1. By the triangle inequality $\|\widehat{A} - A\|_F \leq \|\widehat{A} - A_r\|_F + \|A - A_r\|_F$, a relative upper bound for recovering A_r directly implies a relative upper bound for recovering A.

et al., 2009). Compared to the additive error bound, the key advantage of the relative error bound is that it bounds the error based on $\|A - A_r\|_F$, the approximation error between the original matrix A and its low-rank approximation A_r. As a result, when A is low-rank and $A - A_r = \mathbf{0}$, relative error bounds imply a perfect recovery of A, which will never be accomplished by additive bounds.

In this work, we are interested in bounding $\|B_* - A_r\|_F$ in the form of (4), where B_* is the optimal solution to (3). Similar to previous studies, we assume that the top eigenspaces of A satisfy the classical incoherence condition (Candès and Recht, 2009). Based on the celebrated result of low-rank matrix completion (Recht, 2011), we derive an upper bound for $\|B_* - A_r\|_F$, which induces a relative upper bound under favored conditions. We summarize the key features of our results as follows:

- We present a general theorem that allows us to bound the recovery error of (3) for any $\lambda > 0$. In contrast, Candès and Plan (2010) only analyze the performance of (2) when $\delta \geq \sqrt{\sum_{(i,j)\in\Omega} N_{ij}^2}$.
- By choosing λ appropriately, we obtain a relative upper bound of $O(\frac{mn\sqrt{r}}{|\Omega|}\|A - A_r\|_F)$ in general, and a tighter bound of $O(\sqrt{\frac{mnr}{|\Omega|}}\|A - A_r\|_F)$ when $A - A_r$ is flat, i.e., $\|A - A_r\|_\infty / \|A - A_r\|_F$ is not too large. Although Koltchinskii et al. (2011) and Negahban and Wainwright (2012) have analyzed some variants of (3), their bounds are additive in the sense that they are not proportional to $\|A - A_r\|_F$. To the best of our knowledge, this is the *first* relative error bound for the nuclear norm regularized matrix completion.
- Our relative upper bound for (3) is tighter than that for (2) developed by Candès and Plan (2010), and more general than those proved by Keshavan et al. (2010b) and Eriksson et al. (2012) under different conditions.
- Compared to the additive upper bounds of other methods (Keshavan et al., 2010b; Koltchinskii et al., 2011; Foygel and Srebro, 2011), our relative upper bound is tighter when $\|A - A_r\|_F$ is small. In addition, our relative error bound implies the perfect recovery case when the target matrix A is low-rank while the additive error never vanishes.

Notations For a matrix X, we use $\|X\|_*$, $\|X\|_F$, $\|X\|$, and $\|X\|_\infty$ to denote its nuclear norm, Frobenius norm, spectral norm, and the absolute value of the largest element in magnitude, respectively,

2. Related Work

In this section, we provide a brief review of existing work.

2.1. Low-rank Matrix Completion

The mathematical study of matrix completion began with Candès and Recht (2009). Specifically, they have proved that if A obeys the incoherence condition, $|\Omega| \geq Cn^{6/5} r \log(n)$ is sufficient to ensure that with a high probability, A is the unique solution to (1), where C is a constant independent from r, m, and n (Candès and Recht, 2009). The lower bound for the size of Ω is subsequently improved to $nr\log^6(n)$ under a stronger assumption (Candès and

Tao, 2010). These theoretical guarantees are without question great breakthroughs, but the proof techniques are highly involved. In two subsequent studies (Recht, 2011; Gross, 2011), the authors present a very elegant approach for analyzing (1), and give slightly better bounds. For example, Recht (2011) improves the bound for $|\Omega|$ to $rn\log^2(n)$ and requires the weakest assumptions on A. The simplification of the analysis also leads to better understanding of matrix completion, and lays the foundations of the study in this paper.

In an alternative line of work, Keshavan et al. (2010a) study matrix completion using a combination of spectral techniques and manifold optimization. The proposed algorithm named OPTSPACE, also achieves exact recovery if $|\Omega| \geq Cnr\max(\log(n), r)$. However, the constant C in their bound depends on many factors of A such as the aspect ratio and the condition number. After the pioneering work mentioned above, various algorithms and theories of matrix completion have been developed, including distributed matrix completion (Mackey et al., 2011), matrix completion with side information (Xu et al., 2013), 1-bit matrix completion (Cai and Zhou, 2013), noisy matrix completion (Klopp, 2014), coherent matrix completion (Chen et al., 2014), universal matrix completion (Bhojanapalli and Jain, 2014), and non-convex matrix completion (Sun and Luo, 2015), to name a few amongst many.

2.2. Full-rank Matrix Completion

Since existing studies for full-rank matrix completion differ significantly in their assumptions, their theoretical guarantees may not be directly comparable. In the following, we will state previous results in the most general form, and (if possible) characterize their behaviors with respect to m, n, r, and $|\Omega|$.

Denote the optimal solution of (2) by $\widehat{B}$. Under the assumption $\delta \geq \sqrt{\sum_{(i,j)\in\Omega} N_{ij}^2}$, Theorem 7 of Candès and Plan (2010) shows

$$\|\widehat{B} - Z\|_F \leq \left(1 + m\sqrt{\frac{n}{|\Omega|}}\right)\delta.$$

Let $Z = A_r$, $N = A - A_r$, and consider the optimal choice that $\delta = O\left(\sqrt{\sum_{(i,j)\in\Omega} N_{ij}^2}\right)$. The above bound becomes

$$\|\widehat{B} - A_r\|_F \leq \left(1 + m\sqrt{\frac{n}{|\Omega|}}\right)\sqrt{\sum_{(i,j)\in\Omega}(A - A_r)_{ij}^2}. \tag{5}$$

One limitation of this work is that the theoretical guarantee is only valid when δ is sufficiently large. On the other hand, if we use a very large δ, the upper bound becomes loose. Our result overcomes this limitation as our error bound holds for any positive regularization parameter $\lambda > 0$.

An investigation of OPTSPACE (Keshavan et al., 2010a) for full-rank matrix completion is discussed in Keshavan et al. (2010b). In particular, Theorem 1.1 of Keshavan et al. (2010b) implies the following additive upper bound

$$O\left(\|A_r\|_\infty m^{1/4} n^{5/4}\sqrt{\frac{r}{|\Omega|}} + \frac{mn\sqrt{r}}{|\Omega|}\|U\|\right) \tag{6}$$

where U is some matrix that depends on $A - A_r$ and Ω. Although it is possible to derive a relative upper bound from Theorem 1.2 of Keshavan et al. (2010b), it requires very strong assumptions about the coherence, the aspect ratio (n/m), the condition number of A_r and the r-th singular value of A. Thus, the bound derived from Theorem 1.2 of Keshavan et al. (2010b) is significantly more restricted than the bound proved here.

Foygel and Srebro (2011) study the problem of matrix completion from the view point of supervised learning. The optimization problem is formulated as least squares minimization subject to nuclear norm or max norm constraints. Their theoretical results follow from generic generalization guarantees based on the Rademacher complexity. Specifically, Theorem 6 of Foygel and Srebro (2011) implies the following additive upper bound

$$O\left(\|A - A_r\|_F + n\sqrt{\frac{rm}{|\Omega|}} + \sqrt{n\|A - A_r\|_F}\left(\frac{rm}{|\Omega|}\right)^{\frac{1}{4}}\right) \tag{7}$$

where logarithmic factors are ignored. The derivation of (7) is given in Appendix A.

Koltchinskii et al. (2011) have investigated a general trace regression model, which includes matrix completion as a special case. For matrix completion, they propose the following optimization problem

$$\min_{B\in\mathbb{R}^{m\times n}} \quad \frac{1}{2}\left\|B - \frac{mn}{|\Omega|}\sum_{(i,j)\in\Omega} A_{ij}\mathbf{e}_i\mathbf{e}_j^\top\right\|_F^2 + \lambda\|B\|_*.$$

Let $\widehat{B}$ be the optimal solution to the above problem. Under appropriate conditions, it has been proved that with a high probability (Koltchinskii et al., 2011, Corollary 2)

$$\|\widehat{B} - A\|_F^2 + \|\widehat{B} - X\|_F^2 \leq \|X - A\|_F^2 + \frac{Cmn^2\log(n)\mathrm{rank}(X)}{|\Omega|} \tag{8}$$

for all $X \in \mathbb{R}^{m\times n}$. However, due to the presence of the second term in the upper bound, it is impossible to obtain a relative error bound.

Negahban and Wainwright (2012) have analyzed a variant of (3), which contains an additional ℓ_∞-norm constraint. Based on assumptions about the spikiness and rank of the target matrix, they derive the restricted strong convexity condition, and establish the following additive bound (Negahban and Wainwright, 2012, Theorem 2)

$$\begin{aligned}&\max\left(\left(\frac{mn^2\log n}{|\Omega|}\right)^{1/4}\sqrt{\|A - A_r\|_*}, n\sqrt{\frac{rm\log n}{|\Omega|}}\right)\\ \leq &\max\left(\left(\frac{m^2n^2\log n}{|\Omega|}\right)^{1/4}\sqrt{\|A - A_r\|_F}, n\sqrt{\frac{rm\log n}{|\Omega|}}\right).\end{aligned} \tag{9}$$

Thus, their optimization problem, assumptions and theoretical guarantees are all different from ours.

In a recent work, Eriksson et al. (2012) consider a high-rank matrix completion problem in which the columns of A belong to a union of multiple low-rank subspaces. Under certain

assumptions about the coherence as well as the geometrical arrangement of subspaces and the distribution of the columns in the subspaces, they develop a multi-step algorithm that is able to recover *each* column of A with a high probability, as long as $O(rn\log^2(m))$ entries of A are observed uniformly at random. However, the recovery guarantee of their algorithm for general full-rank matrices is unclear.

3. Our Results

We first describe theoretical guarantees and then provide some discussions.

3.1. Theoretical Guarantees

Let $U = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$ and $V = [\mathbf{v}_1, \ldots, \mathbf{v}_r]$ be two matrices that contain the first r left and right singular vectors of matrix A, respectively. Let $\mathbf{e}_i$ and $\mathbf{e}_j$ be the i-th and j-th standard basis in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively. Following the previous studies in matrix completion (Candès and Recht, 2009; Recht, 2011), we define the coherence measure μ_0 as

$$\mu_0 = \max\left(\frac{m}{r}\max_{1\leq i\leq m}\|P_U\mathbf{e}_i\|^2, \frac{n}{r}\max_{1\leq j\leq n}\|P_V\mathbf{e}_j\|^2\right)$$

where $P_U = UU^\top$ and $P_V = VV^\top$ are two projection operators. We also define μ_1 as

$$\mu_1 = \max_{i\in[m],j\in[n]}\sqrt{\frac{mn}{r}}\left|[UV^\top]_{ij}\right|.$$

Define two projection operators $\mathcal{P}_T$ and $\mathcal{P}_{T^\perp}$ for matrices as

$$\mathcal{P}_T(Z) = P_UZ + ZP_V - P_UZP_V, \text{ and } \mathcal{P}_{T^\perp}(Z) = (I - P_U)Z(I - P_V).$$

We assume the indices are sampled uniformly with replacement, and thus Ω is a *collection* that may contain duplicate indices. The linear operator $\mathcal{R}_\Omega : \mathbb{R}^{m\times n} \mapsto \mathbb{R}^{m\times n}$ is defined as

$$\mathcal{R}_\Omega(Z) = \sum_{(i,j)\in\Omega}\langle\mathbf{e}_i\mathbf{e}_j^\top, Z\rangle\mathbf{e}_i\mathbf{e}_j^\top.$$

To simplify the notation, we define

$$\varepsilon = \|A - A_r\|_F.$$

3.1.1. A General Result

Let B_* be the optimal solution to (3). Based on the optimality condition of B_* and the guarantee for low-rank matrix completion (Recht, 2011), we obtain the following theorem.

Theorem 1 *Assume*

$$|\Omega| \geq 114\max(\mu_0, \mu_1^2)r(m+n)\beta\log^2(2n) \tag{10}$$

for some $\beta > 1$, and $n \geq 5$. With a probability at least $1 - 6\log(n)(m+n)^{2-2\beta} - n^{2-2\beta^{1/2}}$, we have

$$\|\mathcal{P}_{T^\perp}(B_*)\|_F \leq \|\mathcal{P}_{T^\perp}(B_*)\|_* \leq \frac{256\beta\log^2(n)\varepsilon^2}{9\lambda} + \frac{3mnr\log(2n)\lambda}{|\Omega|},$$

$$\|\mathcal{P}_T(A_r - B_*)\|_F \leq \frac{16\log(n)\varepsilon}{3}\sqrt{\frac{2\beta mn}{|\Omega|}} + \frac{2mn\lambda}{|\Omega|}\sqrt{3r\log(2n)} + 64\log(n)\sqrt{\frac{mn\beta}{6|\Omega|}}\|\mathcal{P}_{T^\perp}(B_*)\|_F.$$

As can be seen, our upper bound is valid for any $\lambda > 0$. In contrast, the upper bound for (2) in (Candès and Plan, 2010) is limited to the case $\delta \geq \sqrt{\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle}$.

By choosing λ to minimize the upper bounds in the above theorem, we obtain the following corollary.

Corollary 2 *Under the condition in Theorem 1. Set*

$$\lambda = \frac{16\varepsilon}{3}\sqrt{\frac{\beta\log(2n)|\Omega|}{3mnr}}. \tag{11}$$

With a probability at least $1 - 6\log(n)(m+n)^{2-2\beta} - n^{2-2\beta^{1/2}}$, we have

$$\|\mathcal{P}_{T^\perp}(B_*)\|_F \leq \frac{32\log(2n)}{3}\sqrt{\frac{3\beta mnr\log(2n)}{|\Omega|}}\varepsilon,$$

$$\|\mathcal{P}_T(A_r - B_*)\|_F \leq \left(19\log(2n)\sqrt{\frac{\beta mn}{|\Omega|}} + \frac{2048\beta\log^2(2n)mn}{3|\Omega|}\sqrt{\frac{r\log(2n)}{2}}\right)\varepsilon,$$

and thus

$$\|A_r - B_*\|_F \leq O\left(\log(n)\sqrt{\frac{mnr\log(n)}{|\Omega|}} + \frac{mn\log^2(n)}{|\Omega|}\sqrt{r\log(n)}\right)\varepsilon.$$

Corollary 2 shows that, with an appropriate choice of the parameter λ, we can obtain a relative upper bound. One way to estimate λ is to use the cross validation technique, an approach that is widely used in learning. More specifically, we can divide the observed entries into two separate sets: the training set and the validation set. We will use the training set to find the optimal solution to the recovered matrix, and use the validation set to determine the appropriate parameter λ. When (11) holds, we can also express the upper bounds in terms of λ. It is easy to verify that with a high probability, we have

$$\|A_r - B_*\|_F \leq O\left(\frac{mnr\log^2(n)}{|\Omega|}\sqrt{\frac{mn}{|\Omega|}}\right)\lambda. \tag{12}$$

In the special case when A is a rank-r matrix, i.e., $A = A_r$, (12) implies the smaller the λ, the better the bound. In other words, we have $\|A - B_*\|_F \to 0$ as $\lambda \to 0$.

Finally, we note that whether the upper bound in Corollary 2 is tight remains open. Although both Koltchinskii et al. (2011, Theorem 6) and Negahban and Wainwright (2012, Theorem 3) have established lower bounds for noisy full-rank matrix completion, their bounds become 0 in the noisy-free setting. Thus, existing lower bounds cannot be used to examine the optimality of our result, and we will investigate the lower bound for noisy-free setting in the future.

3.1.2. A Special Result with Tighter Bounds

In the case that the residual matrix $A - A_r$ is not too spiky, in other words, $\|A - A_r\|_\infty / \|A - A_r\|_F$ is not too large, we obtain a tighter theorem as stated below.

Theorem 3 *Assume*

$$|\Omega| \geq \max\left(114\max(\mu_0, \mu_1^2)r(m+n)\beta\log^2(2n), \frac{8mn\|A - A_r\|_\infty^2}{3\|A - A_r\|_F^2}\beta\log(n)\right) \tag{13}$$

for some $\beta > 1$, and $n \geq 5$. With a probability at least $1 - 6\log(n)(m+n)^{2-2\beta} - n^{2-2\beta^{1/2}} - n^{-\beta}$, we have

$$\|\mathcal{P}_{T^\perp}(B_*)\|_F \leq \|\mathcal{P}_{T^\perp}(B_*)\|_* \leq \frac{8|\Omega|\varepsilon^2}{mn\lambda} + \frac{3mnr\log(2n)\lambda}{|\Omega|},$$

$$\|\mathcal{P}_T(A_r - B_*)\|_F \leq 4\varepsilon + \frac{2mn\lambda}{|\Omega|}\sqrt{3r\log(2n)} + 64\log(n)\sqrt{\frac{mn\beta}{6|\Omega|}}\|\mathcal{P}_{T^\perp}(B_*)\|_F.$$

In this theorem, we have two lower bounds for $|\Omega|$ in (13). If A is low-rank, the second lower bound will vanish. Furthermore, it can be dropped when $(\|A - A_r\|_\infty^2 / \|A - A_r\|_F^2) \leq O(r\log n/m)$, i.e., when the residual matrix does not concentrate on a small number of entries. Note that our flatness assumption is a condition over the residual matrix $A - A_r$. It is different from the ℓ_∞-norm constraint of Negahban and Wainwright (2012), which is a requirement over the target matrix A.

By choosing λ to minimize the upper bounds in Theorem 3, we obtain the following relative upper bounds.

Corollary 4 *Under the condition in Theorem 1. Set*

$$\lambda = \frac{2|\Omega|\varepsilon}{mn}\sqrt{\frac{2}{3r\log(2n)}}.$$

With a probability at least $1 - 6\log(n)(m+n)^{2-2\beta} - n^{2-2\beta^{1/2}} - n^{-\beta}$, we have

$$\|\mathcal{P}_{T^\perp}(B_*)\|_F \leq 4\sqrt{6r\log(2n)}\varepsilon,$$

$$\|\mathcal{P}_T(A_r - B_*)\|_F \leq \left(10 + 256\sqrt{\frac{mnr\log^3(2n)\beta}{|\Omega|}}\right)\varepsilon,$$

and thus

$$\|A_r - B_*\|_F \leq O\left(\sqrt{r\log(n)} + \sqrt{\frac{mnr\log^3(n)}{|\Omega|}}\right)\varepsilon.$$

As can be seen, the upper bound for $\|A_r - B_*\|_F$ in the above corollary is tighter than that in Corollary 2 by a factor of $\log(n)\sqrt{\frac{mn}{|\Omega|}}$.

3.2. Comparisons

We compare our theoretical guarantees with previous results for matrix completion in this section. We focus on the practical scenario $|\Omega| \leq mn$, and for simplicity *ignore* logarithmic factors.

The most comparable study is the relative upper bound derived by Candès and Plan (2010) for the constrained problem in (2), since their analysis also relies on the incoherence condition. In the general case, Corollary 2 gives the following relative error bound

$$\|A_r - B_*\|_F \leq O\left(\frac{mn\sqrt{r}}{|\Omega|}\right)\varepsilon.$$

From (5) in Section 2.2, we observe that Candès and Plan (2010) give the following bound

$$\|A_r - \widehat{B}\|_F \leq O\left(m\sqrt{\frac{n}{|\Omega|}}\right)\varepsilon.$$

Because $|\Omega| \geq Cnr$ for some constant C, we have

$$m\sqrt{\frac{n}{|\Omega|}} \geq \sqrt{C}\frac{mn\sqrt{r}}{|\Omega|},$$

which implies our bound is always tighter than that of Candès and Plan (2010). In the case that (13) holds, Corollary 4 indicates our relative error bound can be improved to

$$\|A_r - B_*\|_F \leq O\left(\sqrt{\frac{mnr}{|\Omega|}}\right)\varepsilon.$$

Using Lemma 2 in Section 4.7, the error bound of Candès and Plan (2010) can also be improved and becomes

$$\|A_r - \widehat{B}\|_F \leq O\left(m\sqrt{\frac{n}{|\Omega|}}\right)\sqrt{\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle} \overset{(43)}{\leq} O\left(\sqrt{m}\varepsilon\right),$$

which is again worse than our bound since $|\Omega| \geq Cnr$. Compared to the relative upper bounds of Keshavan et al. (2010b) and Eriksson et al. (2012), our result is applicable to a more general case as their bounds only hold for a very restricted class of matrices.

Next, we compare our relative error bound with the additive bounds in previous studies (Keshavan et al., 2010b; Foygel and Srebro, 2011; Koltchinskii et al., 2011; Negahban and Wainwright, 2012). Since those results are derived under different assumptions, the comparison should be treated conservatively. We remark that those assumptions are incomparable in general, since we can construct matrices to satisfy one assumption but violate others (Negahban and Wainwright, 2012, Setion 3.4.2). Our goal is to show that relative bounds could be tighter than additive bounds under certain conditions.

For brevity, we only provide the comparison using the tighter bound in Corollary 4. Our relative bound $O(\sqrt{\frac{mnr}{|\Omega|}}\varepsilon)$ is tighter than the additive bound in (6) derived by Keshavan et al. (2010b), if $\varepsilon \leq O(n^{3/4}/m^{1/4})$ and also tighter than the additive bound in (7) derived by Foygel and Srebro (2011), if $\varepsilon \leq O(\sqrt{n})$. To compare with the additive bound in (8) derived by Koltchinskii et al. (2011), we set $X = A_r$ and have, with a high probability,

$$\left\|\widehat{B} - A_r\right\|_F \leq \varepsilon + O\left(\sqrt{\frac{mn^2r}{|\Omega|}}\right)$$

which is worse than our bound if $\varepsilon \leq O(\sqrt{n})$. Our bound is better than the additive bound in (9) of Negahban and Wainwright (2012), when

$$\epsilon \leq \max\left(\frac{\sqrt{\Omega}}{r}, \sqrt{n}\right).$$

Finally, we note that although our analysis is devoted to full-rank matrix completion, it can also be applied to noisy low-rank matrix completion. In this case, we have $A = Z + N$, where Z is a low-rank matrix and N is the matrix of noise. As long as the eigenspaces of Z satisfy the incoherence condition, our theoretical guarantees are valid by setting $A_r = Z$ and $\varepsilon = \|N\|_F$, even when Z may not be the best rank-r approximation of A. [2] To compare with previous studies, let's assume $\|Z\|_F = 1$ and entries of N are independent sampled from $\mathcal{N}(0, \sigma^2)$ where $\sigma^2 = 1/(mn)$. Based on the concentration inequality for χ^2-distributions (Laurent and Massart, 2000, Lemma 1), with a high probability, $\varepsilon^2 = O(\sigma^2 mn) = O(1)$. Then, our Corollary 2 and Corollary 4 imply that with a high probability

$$\|Z - B_*\|_F \leq O\left(\frac{mn\sqrt{r}}{|\Omega|}\right) \text{ and } \|Z - B_*\|_F \leq O\left(\sqrt{\frac{mnr}{|\Omega|}}\right),$$

respectively. In contrast, existing results for noisy low-rank matrix completion, e.g., Corollary 1 of Negahban and Wainwright (2012) and Theorem 7 of Klopp (2014), have established an $O(\sqrt{rn/|\Omega|})$ bound. Thus, our bounds are loose for noisy low-rank matrix completion, which is probably because our analysis did not exploit the fact that entries of N are i.i.d. sampled.

4. Analysis

Although the current analysis is built upon the result from Recht (2011) that requires the incoherence assumption, it can be extended to support other assumptions for matrix completion. The key is to replace Theorem 5 below with the corresponding theorem derived under other assumptions. With appropriate replacement of Theorem 5, we should still be able to obtain a relative error bound, of course with different dependence on m, n, r, and $|\Omega|$. For example, using the assumptions and theorems in Bhojanapalli and Jain (2014), we can generalize our result to a *universal* guarantee for full-matrix completion. We leave the extension of our analysis to other assumptions as a future work.

2. This can be easily verified because our analysis only requires the eigenspaces of A_r satisfy the incoherence condition.

4.1. Sketch of the Proof

As we mentioned before, our analysis is built upon the existing theoretical guarantee for low-rank matrix completion, which is summarized below (Recht, 2011).

Theorem 5 *Suppose*

$$|\Omega| \geq 32 \max(\mu_0, \mu_1^2) r(m+n)\beta \log^2(2n) \tag{14}$$

for some $\beta > 1$. Then, with a probability at least $1 - 6\log(n)(m+n)^{2-2\beta} - n^{2-2\beta^{1/2}}$, the following statements are true:

- $$\left\| \frac{mn}{|\Omega|} \mathcal{P}_T \mathcal{R}_\Omega \mathcal{P}_T - \mathcal{P}_T \right\| \leq \frac{1}{2}. \tag{15}$$
- $$\|\mathcal{R}_\Omega\| \leq \frac{8}{3}\sqrt{\beta}\log(n). \tag{16}$$
- *There exists a $Y \in \mathbb{R}^{m\times n}$ in the range of $\mathcal{R}_\Omega$ such that*

$$\left\| \mathcal{P}_T(Y) - UV^\top \right\|_F \leq \sqrt{\frac{r}{2n}}, \tag{17}$$

$$\left\| \mathcal{P}_{T^\perp(Y)} \right\| \leq \frac{1}{2}, \tag{18}$$

$$|\langle Y, A\rangle| \leq \sqrt{\frac{3mnr\log(2n)}{8|\Omega|}} \sqrt{\langle \mathcal{R}_\Omega(A), A\rangle}, \tag{19}$$

for all $A \in \mathbb{R}^{m\times n}$.

The first part of above theorem contains concentration inequalities for the random linear operator $\mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T$ and $\mathcal{R}_\Omega$, and the second part describes some important properties of a special matrix Y, which is used as an (approximate) dual certificate of (1).

Next, we will examine the optimality of B_* based on techniques from convex analysis, leading to the following theorem.

Theorem 6 *Let B_* be the optimal solution to (3), we have*

$$\lambda\langle B_* - A_r, UV^\top\rangle + \lambda\|\mathcal{P}_{T^\perp}(B_*)\|_* \leq \langle \mathcal{R}_\Omega(B_* - A), A_r - B_*\rangle. \tag{20}$$

Based on Theorems 5 and 6, we are ready to prove the main results. However, the analysis is a bit lengthy, so we split it into two parts, and will first show the following intermediate theorem.

Theorem 7 *Suppose (14) holds. With a probability at least $1 - 6\log(n)(m+n)^{2-2\beta} - n^{2-2\beta^{1/2}}$, we have*

$$\begin{aligned} &\frac{1}{2}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle + \frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \\ \leq &\lambda\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F + \langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle + \frac{3mnr\log(2n)\lambda^2}{8|\Omega|}. \end{aligned} \tag{21}$$

Then, we can prove Theorem 1 by further lower bounding and upper bounding the L.H.S. and R.H.S. of (21), respectively. If the residual matrix $A - A_r$ is not too spiky, we can apply Bernstein's inequality to derive a tighter bound for $\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle$ and obtain Theorem 3 in a similar way.

4.2. Property of the Linear Operator $\mathcal{R}_\Omega$

Before going to the detail, we first introduce a lemma that will be used throughout the analysis. Since Ω may contain duplicate indices, $\langle \mathcal{R}_\Omega(A), A\rangle \neq \|\mathcal{R}_\Omega(A)\|_F^2$ in general. We use the following lemma to take care of this issue.

Lemma 1

$$\langle \mathcal{R}_\Omega(A), A\rangle \leq \|\mathcal{R}_\Omega(A)\|_F^2, \tag{22}$$

$$|\langle \mathcal{R}_\Omega(A), B\rangle| \leq \sqrt{\langle \mathcal{R}_\Omega(A), A\rangle}\sqrt{\langle \mathcal{R}_\Omega(B), B\rangle} \tag{23}$$

for all $A, B \in \mathbb{R}^{m\times n}$.

Proof Denote the number of unique indices in Ω by u, and let $\Theta = \{(a_k, b_k)\}_{k=1}^u$ be a *set* that contains all the unique indices in Ω. Let t_k denote the times that (a_k, b_k) appears in Ω. Then, we have

$$\langle \mathcal{R}_\Omega(A), A\rangle = \sum_{k=1}^u t_k A_{a_k b_k}^2 \leq \sum_{k=1}^u t_k^2 A_{a_k b_k}^2 = \|\mathcal{R}_\Omega(A)\|_F^2.$$

To show (23), we have

$$|\langle \mathcal{R}_\Omega(A), B\rangle| = \left| \sum_{(i,j)\in\Omega} A_{ij}B_{ij} \right| \leq \sqrt{\sum_{(i,j)\in\Omega} A_{ij}^2}\sqrt{\sum_{(i,j)\in\Omega} B_{ij}^2} = \sqrt{\langle \mathcal{R}_\Omega(A), A\rangle}\sqrt{\langle \mathcal{R}_\Omega(B), B\rangle}$$

where the inequality is due to Cauchy–Schwarz inequality. ■

4.3. Proof of Theorem 5

Except for the last inequality in (19), all the others can be found directly from Section 4 of Recht (2011). Thus, we only provide the derivation of (19), which is based on some intermediate results of Recht (2011).

We first state those intermediate results. Following the construction of Recht (2011, Section 4), we partition Ω into p partitions of size q. By assumption, we can choose

$$q \geq \frac{128}{3}\max(\mu_0, \mu_1^2) r(m+n)\beta\log(m+n)$$

such that

$$p = \frac{\Omega}{q} = \frac{3}{4}\log 2n.$$

Let Ω_j denote the set of indices corresponding to the j-th partition. We define $W_0 = UV^\top$,

$$Y_k = \frac{mn}{q}\sum_{j=1}^{k}\mathcal{R}_{\Omega_j}(W_{j-1}), \text{ and } W_k = UV^\top - \mathcal{P}_T(Y_k)$$

for $k = 1, \ldots, p$. Then, we set $Y = Y_p$. Recht (2011) has proved that with a probability at least $1 - 6\log(n)(m+n)^{2-2\beta} - n^{2-2\beta^{1/2}}$,

$$\|W_k\|_F \le 2^{-k}\sqrt{r}, \tag{24}$$

$$\left\|\frac{mn}{q}\mathcal{P}_T\mathcal{R}_{\Omega_k}\mathcal{P}_T - \mathcal{P}_T\right\| \le \frac{1}{2}, \tag{25}$$

for $k = 1, \ldots, p$.

We proceed to prove (19). Since $W_j = \mathcal{P}_T(W_j)$, we have

$$\begin{aligned}\frac{mn}{q}\langle\mathcal{R}_{\Omega_j}(W_j), W_j\rangle &= \left\langle W_j, \frac{mn}{q}\mathcal{P}_T\mathcal{R}_{\Omega_j}\mathcal{P}_T(W_j)\right\rangle \\ &\overset{(25)}{\le} \frac{3}{2}\|\mathcal{P}_T(W_j)\|_F^2 = \frac{3}{2}\|W_j\|_F^2 \overset{(24)}{\le} \frac{3r}{2}4^{-j}.\end{aligned} \tag{26}$$

Then,

$$\begin{aligned}|\langle Y, A\rangle| \le& \frac{mn}{q}\sum_{j=1}^{p}\left|\langle\mathcal{R}_{\Omega_j}(W_{j-1}), A\rangle\right| \\ \overset{(23)}{\le}& \frac{mn}{q}\sum_{j=1}^{p}\sqrt{\langle\mathcal{R}_{\Omega_j}(W_{j-1}), W_{j-1}\rangle}\sqrt{\langle\mathcal{R}_{\Omega_j}(A), A\rangle} \\ \le& \frac{mn}{q}\sqrt{\sum_{j=1}^{p}\langle\mathcal{R}_{\Omega_j}(W_{j-1}), W_{j-1}\rangle}\sqrt{\sum_{j=1}^{p}\langle\mathcal{R}_{\Omega_j}(A), A\rangle} \\ =& \sqrt{\frac{mn}{q}}\sqrt{\sum_{j=1}^{p}\left\langle\frac{mn}{q}\mathcal{R}_{\Omega_j}(W_{j-1}), W_{j-1}\right\rangle}\sqrt{\langle\mathcal{R}_\Omega(A), A\rangle} \\ \overset{(26)}{\le}& \sqrt{\frac{mn}{q}}\sqrt{\langle\mathcal{R}_\Omega(A), A\rangle}\sqrt{\frac{3r}{2}\sum_{j=1}^{p}4^{-j}} \\ \le& \sqrt{\frac{mnr}{2q}}\sqrt{\langle\mathcal{R}_\Omega(A), A\rangle} = \sqrt{\frac{mnpr}{2|\Omega|}}\sqrt{\langle\mathcal{R}_\Omega(A), A\rangle}.\end{aligned}$$

4.4. Proof of Theorem 6

Since B_* is the optimal solution to (3), we have

$$\langle\mathcal{R}_\Omega(B_* - A) + \lambda E, A_r - B_*\rangle \ge 0 \tag{27}$$

where $E \in \partial\|B_*\|_*$ is *certain* subgradient of $\|\cdot\|_*$ evaluated at B_*. Let $F \in \partial\|A_r\|_*$ be *any* subgradient of $\|\cdot\|_*$ evaluated at A_r. From the property of convexity, we have

$$\langle B_* - A_r, E - F\rangle \ge 0. \tag{28}$$

From (27) and (28), we get

$$\langle\mathcal{R}_\Omega(B_* - A) + \lambda F, A_r - B_*\rangle \ge 0. \tag{29}$$

Next, we consider bounding $\lambda\langle F, A_r - B_*\rangle$. From previous studies (Candès and Recht, 2009), we know that the set of subgradients of $\|A_r\|_*$ takes the following form:

$$\partial\|A_r\|_* = \left\{UV^\top + W : W \in \mathbb{R}^{m\times n}, U^\top W = 0, WV = 0, \|W\| \le 1\right\}.$$

Thus, we can choose

$$F = UV^\top + \mathcal{P}_{T^\perp}(N),$$

where $N = \text{argmax}_{\|X\|\le 1}\langle\mathcal{P}_{T^\perp}(B_*), X\rangle$. Then, it is easy to verify that

$$\begin{aligned}\langle B_* - A_r, F\rangle =& \langle B_* - A_r, UV^\top\rangle + \langle B_* - A_r, \mathcal{P}_{T^\perp}(N)\rangle \\ =& \langle B_* - A_r, UV^\top\rangle + \|\mathcal{P}_{T^\perp}(B_*)\|_*.\end{aligned} \tag{30}$$

We complete the proof by combining (29) and (30).

4.5. Proof of Theorem 7

We continue the proof by lower bounding $\langle B_* - A_r, UV^\top\rangle$ in (20) of Theorem 6. To this end, we need the matrix Y given in Theorem 5.

$$\begin{aligned}\langle B_* - A_r, UV^\top\rangle =& \langle B_* - A_r, UV^\top - Y\rangle + \langle B_* - A_r, Y\rangle \\ =& \langle B_* - A_r, UV^\top - \mathcal{P}_T(Y)\rangle + \langle A_r - B_*, \mathcal{P}_{T^\perp}(Y)\rangle + \langle B_* - A_r, Y\rangle.\end{aligned}$$

Next, we bound the last three terms by utilizing the conclusions in Theorem 5.

$$\begin{aligned}&\langle B_* - A_r, UV^\top - \mathcal{P}_T(Y)\rangle = \langle\mathcal{P}_T(B_* - A_r), UV^\top - \mathcal{P}_T(Y)\rangle \\ \ge& -\|\mathcal{P}_T(B_* - A_r)\|_F\|UV^\top - \mathcal{P}_T(Y)\|_F \overset{(17)}{\ge} -\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F. \\ &\langle A_r - B_*, \mathcal{P}_{T^\perp}(Y)\rangle = \langle\mathcal{P}_{T^\perp}(A_r - B_*), \mathcal{P}_{T^\perp}(Y)\rangle \\ =& \langle\mathcal{P}_{T^\perp}(-B_*), \mathcal{P}_{T^\perp}(Y)\rangle \ge -\|\mathcal{P}_{T^\perp}(B_*)\|_*\|\mathcal{P}_{T^\perp}(Y)\| \overset{(18)}{\ge} -\frac{1}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_*. \\ &\langle B_* - A_r, Y\rangle \overset{(19)}{\ge} -\sqrt{\frac{3mnr\log(2n)}{8|\Omega|}}\sqrt{\langle\mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle}.\end{aligned}$$

Putting the above inequalities together, we have

$$\begin{aligned}\langle B_* - A_r, UV^\top\rangle \ge& -\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F - \frac{1}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \\ &- \sqrt{\frac{3mnr\log(2n)}{8|\Omega|}}\sqrt{\langle\mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle}.\end{aligned} \tag{31}$$

Substituting (31) into (20) and rearranging, we get

$$\begin{aligned}
&\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle + \frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \\
\le &\langle \mathcal{R}_\Omega(A_r - A), A_r - B_*\rangle + \lambda\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F \\
&+ \lambda\sqrt{\frac{3mnr\log(2n)}{8|\Omega|}}\sqrt{\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle} \\
\overset{(23)}{\le} &\sqrt{\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle}\sqrt{\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle} + \lambda\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F \\
&+ \lambda\sqrt{\frac{3mnr\log(2n)}{8|\Omega|}}\sqrt{\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle}.
\end{aligned} \tag{32}$$

From the the basic inequality $\frac{1}{4}\alpha^2 - \alpha\beta + \beta^2 \ge 0$, we have

$$\begin{aligned}
&\sqrt{\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle}\sqrt{\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle} \\
\le &\frac{1}{4}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle + \langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle,
\end{aligned} \tag{33}$$

$$\begin{aligned}
&\lambda\sqrt{\frac{3mnr\log(2n)}{8|\Omega|}}\sqrt{\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle} \\
\le &\frac{1}{4}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle + \lambda^2\frac{3mnr\log(2n)}{8|\Omega|}.
\end{aligned} \tag{34}$$

We complete the proof by summing (32), (33), and (34) together.

4.6. Proof of Theorem 1

The lower bound of $|\Omega|$ in (10) is due to Theorem 5, but we use a larger constant (114 instead of 32) to ensure

$$\frac{8\log(n)}{3}\sqrt{\frac{rm\beta}{|\Omega|}} \le \frac{1}{4} \tag{35}$$

which is used later.

Based on Lemma 1 and Theorem 5, we have

$$\sqrt{\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle} \overset{(22)}{\le} \|\mathcal{R}_\Omega(A - A_r)\|_F \overset{(16)}{\le} \frac{8}{3}\sqrt{\beta}\log(n)\varepsilon.$$

Substituting the above inequality into (21), we have

$$\frac{1}{2}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle + \frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \le \lambda\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F + \Gamma, \tag{36}$$

where

$$\Gamma = \frac{64\beta\log^2(n)\varepsilon^2}{9} + \frac{3mnr\log(2n)\lambda^2}{8|\Omega|}. \tag{37}$$

4.6.1. Upper Bound for $\|\mathcal{P}_{T^\perp}(B_*)\|_F$

We upper bound $\|\mathcal{P}_T(A_r - B_*)\|_F^2$ in (36) by

$$\|\mathcal{P}_T(A_r - B_*)\|_F^2 = \langle \mathcal{P}_T(A_r - B_*), A_r - B_*\rangle \overset{(15)}{\le} 2\frac{mn}{|\Omega|}\langle \mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T(A_r - B_*), A_r - B_*\rangle.$$

Plugging the above inequality in (36), we have

$$\begin{aligned}
&\frac{1}{2}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle + \frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \\
\le &\lambda\sqrt{\frac{rm}{|\Omega|}}\sqrt{\langle \mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T(A_r - B_*), A_r - B_*\rangle} + \Gamma.
\end{aligned} \tag{38}$$

Since $\mathcal{P}_T + \mathcal{P}_{T^\perp} = I$, we have

$$\begin{aligned}
&\frac{1}{2}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle \\
= &\frac{1}{2}\underbrace{\langle \mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T(A_r - B_*), A_r - B_*\rangle}_{:=\Theta^2} + \frac{1}{2}\underbrace{\langle \mathcal{P}_{T^\perp}\mathcal{R}_\Omega\mathcal{P}_{T^\perp}(B_*), B_*\rangle}_{:=\Lambda^2} \\
&- \langle \mathcal{R}_\Omega(\mathcal{P}_T(A_r - B_*)), \mathcal{P}_{T^\perp}(B_*)\rangle \\
\overset{(23)}{\ge} &\frac{1}{2}\Theta^2 + \frac{1}{2}\Lambda^2 - \Theta\Lambda = \frac{1}{2}(\Theta - \Lambda)^2.
\end{aligned} \tag{39}$$

Substituting (39) into (38), we have

$$\frac{1}{2}(\Theta - \Lambda)^2 + \frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \le \lambda\sqrt{\frac{rm}{|\Omega|}}\Theta + \Gamma.$$

Combining with the fact

$$\frac{1}{2}(\Theta - \Lambda)^2 - \lambda\sqrt{\frac{rm}{|\Omega|}}\Theta + \lambda\sqrt{\frac{rm}{|\Omega|}}\Lambda + \frac{rm\lambda^2}{2|\Omega|} = \frac{1}{2}\left(\Theta - \Lambda - \lambda\sqrt{\frac{rm}{|\Omega|}}\right)^2 \ge 0$$

we have

$$\begin{aligned}
\frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \le &\lambda\sqrt{\frac{rm}{|\Omega|}}\sqrt{\langle \mathcal{P}_{T^\perp}\mathcal{R}_\Omega\mathcal{P}_{T^\perp}(B_*), B_*\rangle} + \frac{rm\lambda^2}{2|\Omega|} + \Gamma \\
\overset{(22)}{\le} &\lambda\sqrt{\frac{rm}{|\Omega|}}\|\mathcal{R}_\Omega\mathcal{P}_{T^\perp}(B_*)\|_F + \frac{rm\lambda^2}{2|\Omega|} + \Gamma \\
\overset{(16)}{\le} &\frac{8\lambda\log(n)}{3}\sqrt{\frac{rm\beta}{|\Omega|}}\|\mathcal{P}_{T^\perp}(B_*)\|_F + \frac{rm\lambda^2}{2|\Omega|} + \Gamma \\
\overset{(35)}{\le} &\frac{\lambda}{4}\|\mathcal{P}_{T^\perp}(B_*)\|_F + \frac{rm\lambda^2}{2|\Omega|} + \Gamma \\
\overset{(37)}{\le} &\frac{\lambda}{4}\|\mathcal{P}_{T^\perp}(B_*)\|_* + \frac{64\beta\log^2(n)\varepsilon^2}{9} + \frac{3mnr\log(2n)\lambda^2}{4|\Omega|}
\end{aligned} \tag{40}$$

where in the last line we use the fact

$$\frac{1}{2} \leq \frac{3n\log(2n)}{8}, \ \forall n \geq 2.$$

From (40), we immediately have

$$\|\mathcal{P}_{T^\perp}(B_*)\|_* \leq \frac{256\beta\log^2(n)\varepsilon^2}{9\lambda} + \frac{3mnr\log(2n)\lambda}{|\Omega|}.$$

4.6.2. Upper Bound for $\|\mathcal{P}_T(A_r - B_*)\|_F$

Similar to (39), we have

$$\begin{aligned}
&\frac{1}{2}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle \\
\overset{(23)}{\geq}& \frac{1}{2}\langle \mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T(A_r - B_*), A_r - B_*\rangle + \frac{1}{2}\Lambda^2 - \sqrt{\langle \mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T(A_r - B_*), A_r - B_*\rangle}\Lambda \\
\overset{(15)}{\geq}& \frac{|\Omega|}{4mn}\|\mathcal{P}_T(A_r - B_*)\|_F^2 + \frac{1}{2}\Lambda^2 - \sqrt{\frac{3|\Omega|}{2mn}}\|\mathcal{P}_T(A_r - B_*)\|_F\Lambda
\end{aligned}$$

where

$$\Lambda = \sqrt{\langle \mathcal{P}_{T^\perp}\mathcal{R}_\Omega\mathcal{P}_{T^\perp}(B_*), B_*\rangle} \overset{(22)}{\leq} \|\mathcal{R}_\Omega\mathcal{P}_{T^\perp}(B_*)\|_F \overset{(16)}{\leq} \frac{8}{3}\sqrt{\beta}\log(n)\|\mathcal{P}_{T^\perp}(B_*)\|_F.$$

By plugging the above inequalities into (36), we have

$$\begin{aligned}
&\frac{|\Omega|}{4mn}\|\mathcal{P}_T(A_r - B_*)\|_F^2 + \frac{1}{2}\Lambda^2 + \frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \\
\leq& \lambda\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F + \Gamma + 8\log(n)\sqrt{\frac{|\Omega|\beta}{6mn}}\|\mathcal{P}_{T^\perp}(B_*)\|_F\|\mathcal{P}_T(A_r - B_*)\|_F
\end{aligned}$$

and thus

$$\begin{aligned}
\|\mathcal{P}_T(A_r - B_*)\|_F^2 \overset{(37)}{\leq}& \frac{2m\lambda\sqrt{2rn}}{|\Omega|}\|\mathcal{P}_T(A_r - B_*)\|_F + \frac{256\beta mn\log^2(n)\varepsilon^2}{9|\Omega|} + \frac{3rm^2n^2\lambda^2\log(2n)}{2|\Omega|^2} \\
&+ 32\log(n)\sqrt{\frac{mn\beta}{6|\Omega|}}\|\mathcal{P}_{T^\perp}(B_*)\|_F\|\mathcal{P}_T(A_r - B_*)\|_F.
\end{aligned} \tag{41}$$

Recall that

$$x^2 \leq bx + c \Rightarrow x \leq 2b + \sqrt{2c}.$$

From (41), we have

$$\begin{aligned}
\|\mathcal{P}_T(A_r - B_*)\|_F \leq& \frac{4m\lambda\sqrt{2rn}}{|\Omega|} + 64\log(n)\sqrt{\frac{mn\beta}{6|\Omega|}}\|\mathcal{P}_{T^\perp}(B_*)\|_F \\
&+ \frac{16\log(n)\varepsilon}{3}\sqrt{\frac{2\beta mn}{|\Omega|}} + \frac{mn\lambda}{|\Omega|}\sqrt{3r\log(2n)}.
\end{aligned}$$

We complete the proof by noticing

$$4\sqrt{2n} \leq n\sqrt{3\log(2n)}, \ \forall n \geq 5.$$

4.7. Proof of Theorem 3

With the second lower bound of $|\Omega|$ in (13), we can prove the following upper bound for $\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle$.

Lemma 2 *Suppose*

$$|\Omega| \geq \frac{8mn\|A - A_r\|_\infty^2}{3\|A - A_r\|_F^2}\beta\log(n) \tag{42}$$

for some $\beta > 1$. Then, with a probability at least $1 - n^{-\beta}$, we have

$$\sqrt{\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle} \leq \varepsilon\sqrt{\frac{2|\Omega|}{mn}}. \tag{43}$$

Proof For each index $(a_k, b_k) \in \Omega$, we define a random variable

$$\xi_k = \langle \mathbf{e}_{a_k}\mathbf{e}_{b_k}^\top, A - A_r\rangle^2 - \frac{1}{mn}\|A - A_r\|_F^2.$$

Then, it is easy to verify that

$$\begin{aligned}
\mathrm{E}[\xi_k] =& 0, \\
|\xi_k| =& \left|\langle \mathbf{e}_{a_k}\mathbf{e}_{b_k}^\top, A - A_r\rangle^2 - \frac{1}{mn}\|A - A_r\|_F^2\right| \\
\leq& \max\left(\langle \mathbf{e}_{a_k}\mathbf{e}_{b_k}^\top, A - A_r\rangle^2, \frac{1}{mn}\|A - A_r\|_F^2\right) \leq \|A - A_r\|_\infty^2, \\
\mathrm{E}[\xi_k^2] =& \mathrm{E}\left[\langle \mathbf{e}_{a_k}\mathbf{e}_{b_k}^\top, A - A_r\rangle^4\right] - \frac{1}{m^2n^2}\|A - A_r\|_F^4 \\
\leq& \frac{1}{mn}\sum_{i,j}[A - A_r]_{ij}^4 \leq \frac{1}{mn}\|A - A_r\|_\infty^2\|A - A_r\|_F^2.
\end{aligned}$$

From Bernstein's inequality, we have

$$\begin{aligned}
&\mathrm{P}\left[\langle \mathcal{R}_\Omega(A - A_r), A - A_r\rangle \geq 2\frac{|\Omega|}{mn}\|A - A_r\|_F^2\right] \\
=& \mathrm{P}\left[\sum_{k=1}^{|\Omega|}\xi_k \geq \frac{|\Omega|}{mn}\|A - A_r\|_F^2\right] \leq \exp\left(-\frac{3|\Omega|}{8\|A - A_r\|_\infty^2}\frac{1}{mn}\|A - A_r\|_F^2\right) \overset{(42)}{\leq} n^{-\beta}.
\end{aligned}$$

■

Following the derivation of (36), we have

$$\frac{1}{2}\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle + \frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \leq \lambda\sqrt{\frac{r}{2n}}\|\mathcal{P}_T(A_r - B_*)\|_F + \Gamma',$$

where

$$\Gamma' = \frac{2|\Omega|\varepsilon^2}{mn} + \frac{3mnr\log(2n)\lambda^2}{8|\Omega|}.$$

The rest of the analysis is almost identical to that of Theorem 1. In particular, (40) becomes

$$\frac{\lambda}{2}\|\mathcal{P}_{T^\perp}(B_*)\|_* \le \frac{\lambda}{4}\|\mathcal{P}_{T^\perp}(B_*)\|_* + \frac{2|\Omega|\varepsilon^2}{mn} + \frac{3mnr\log(2n)\lambda^2}{4|\Omega|},$$

and (41) becomes

$$\begin{aligned}\|\mathcal{P}_T(A_r - B_*)\|_F^2 \le & \frac{2m\lambda\sqrt{2rn}}{|\Omega|}\|\mathcal{P}_T(A_r - B_*)\|_F + 8\varepsilon^2 + \frac{3rm^2n^2\lambda^2\log(2n)}{2|\Omega|^2} \\ & + 32\log(n)\sqrt{\frac{mn\beta}{6|\Omega|}}\|\mathcal{P}_{T^\perp}(B_*)\|_F\|\mathcal{P}_T(A_r - B_*)\|_F.\end{aligned}$$

A complete proof can be found in an early version of this paper (Zhang et al., 2015).

5. Conclusion and Future Work

In this paper, we develop a relative error bound for the nuclear norm regularized matrix completion, under the assumption that the top eigenspaces of the target matrix are incoherent. To the best of our knowledge, this is the first work toward relative error bound for nuclear norm regularized matrix completion, and an extensive comparison shows that our bound is tighter than previous results under favored conditions.

In many real-world applications, it is appropriate to assume the observed entries are corrupted by noise. As we discussed in the end of Section 3.2, it is possible to extend our analysis to the noisy case. More specifically, let N be the matrix of noise. We just need to add $\langle \mathcal{R}_\Omega(N), A_r - B_*\rangle$ to the R.H.S. of (20), which leads to an additional term $\sqrt{\langle \mathcal{R}_\Omega(N), N\rangle}\sqrt{\langle \mathcal{R}_\Omega(A_r - B_*), A_r - B_*\rangle}$ in the R.H.S. of (32). The rest of the proof is almost the same, and finally we will obtain an upper bound that depends on both $A - A_r$ and N.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (2018YFB-1004300), NSFC-NRF Joint Research Project (61861146001), JiangsuSF (BK20160658), YESS (2017QNRC001), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

Appendix A. Derivation of (7)

Let $\widehat{B}$ be the solution found by the algorithm in Foygel and Srebro (2011). Let $\epsilon > 0$ be the mean-squared reconstruction error. Using the notations in this paper, (5) of Foygel and Srebro (2011) becomes

$$\frac{1}{mn}\|\widehat{B} - A\|_F^2 \le \frac{1}{mn}\|A - A_r\|_F^2 + \epsilon \tag{44}$$

under the condition

$$|\Omega| \ge O\left(\frac{r(n+m)}{\epsilon^2}\left(\epsilon + \frac{1}{mn}\|A - A_r\|_F^2\right)\right) \tag{45}$$

where logarithmic factors are ignored. (45) can be rewritten as

$$|\Omega|\epsilon^2 \ge O\left(r(n+m)\epsilon + \frac{r(n+m)}{mn}\|A - A_r\|_F^2\right).$$

Since we assume $m \le n$, it can be further simplified to

$$|\Omega|\epsilon^2 \ge O\left(rn\epsilon + \frac{r}{m}\|A - A_r\|_F^2\right).$$

Let consider the optimal case, i.e.,

$$|\Omega|\epsilon^2 = Crn\epsilon + C\frac{r}{m}\|A - A_r\|_F^2$$

for some constant $C > 0$. Then, we have

$$\begin{aligned}\epsilon &= \frac{Crn + \sqrt{C^2r^2n^2 + 4|\Omega|C\frac{r}{m}\|A - A_r\|_F^2}}{2|\Omega|} \\ &\le \frac{Crn + \sqrt{|\Omega|C\frac{r}{m}\|A - A_r\|_F^2}}{|\Omega|} = O\left(\frac{rn}{|\Omega|} + \sqrt{\frac{r}{m|\Omega|}\|A - A_r\|_F^2}\right).\end{aligned}$$

As a result, (44) becomes

$$\frac{1}{mn}\|\widehat{B} - A\|_F^2 \le O\left(\frac{1}{mn}\|A - A_r\|_F^2 + \frac{rn}{|\Omega|} + \sqrt{\frac{r}{m|\Omega|}\|A - A_r\|_F^2}\right)$$

which implies

$$\|\widehat{B} - A\|_F^2 \le O\left(\|A - A_r\|_F^2 + \frac{rmn^2}{|\Omega|} + \sqrt{\frac{rmn^2}{|\Omega|}\|A - A_r\|_F^2}\right)$$

from which we obtain (7).

References

Srinadh Bhojanapalli and Prateek Jain. Universal matrix completion. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks*, 2(2):188–220, 2006.

Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977, 2009.

Ricardo S. Cabral, Fernando Torre, Joao P. Costeira, and Alexandre Bernardino. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems 24*, pages 190–198, 2011.

Tony Cai and Wen-Xin Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *Journal of Machine Learning Research*, 14:3619–3647, 2013.

Emmanuel J. Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Coherent matrix completion. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American Mathematical Society*, 22(1):211C–231, 2009.

Brian Eriksson, Laura Balzano, and Robert Nowak. High-rank matrix completion. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 373–381, 2012.

Rina Foygel and Nathan Srebro. Concentration-based guarantees for low-rank matrix reconstruction. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 315–340, 2011.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory,*, 57(3):1548–1566, 2011.

Ke Hou, Zirui Zhou, Anthony Man-Cho So, and Zhi-Quan Luo. On the linear convergence of the proximal gradient method for trace norm regularization. In *Advances in Neural Information Processing Systems 26*, pages 710–718, 2013.

Cho-Jui Hsieh and Peder Olsen. Nuclear norm minimization via active subspace selection. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Ali Jalali, Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1001–1008, 2011.

Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 457–464, 2009.

Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010a.

Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010b.

Olga Klopp. Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*, 20:282–303, 2014.

Vladimir Koltchinskii, Karim Lounici, and Alexandre B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Lester W. Mackey, Michael I. Jordan, and Ameet Talwalkar. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems 24*, pages 1134–1142, 2011.

Sahand Negahban and Martin J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697, 2012.

Yu. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20 (6):3465–3489, 2010.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6(3):615–640, 2010.

Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in Neural Information Processing Systems 26*, pages 2301–2309, 2013.

Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Analysis of nuclear norm regularization for full-rank matrix completion. *ArXiv e-prints*, arXiv:1504.06817, 2015.

Xinhua Zhang, Yaoliang Yu, and Dale Schuurmans. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems 25*, pages 2906–2914, 2012.

Journal of Machine Learning Research 20 (2019) 1-63 Submitted 5/17; Revised 1/19; Published 4/19

Adaptive Geometric Multiscale Approximations for Intrinsically Low-dimensional Data

Wenjing Liao WLIAO60@GATECH.EDU
School of Mathematics
Georgia Institute of Technology, Atlanta, GA, 30313, USA

Mauro Maggioni MAUROMAGGIONIJHU@ICLOUD.COM
Department of Mathematics, Department of Applied Mathematics and Statistics,
Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, USA

Editor: Sujay Sanghavi

Abstract

We consider the problem of efficiently approximating and encoding high-dimensional data sampled from a probability distribution ρ in $\mathbb{R}^D$, that is nearly supported on a d-dimensional set $\mathcal{M}$ - for example supported on a d-dimensional manifold. Geometric Multi-Resolution Analysis (GMRA) provides a robust and computationally efficient procedure to construct low-dimensional geometric approximations of $\mathcal{M}$ at varying resolutions. We introduce GMRA approximations that adapt to the unknown regularity of $\mathcal{M}$, by introducing a thresholding algorithm on the geometric wavelet coefficients. We show that these data-driven, empirical geometric approximations perform well, when the threshold is chosen as a suitable universal function of the number of samples n, on a large class of measures ρ, that are allowed to exhibit different regularity at different scales and locations, thereby efficiently encoding data from more complex measures than those supported on manifolds. These GMRA approximations are associated to a dictionary, together with a fast transform mapping data to d-dimensional coefficients, and an inverse of such a map, all of which are data-driven. The algorithms for both the dictionary construction and the transforms have complexity $CDn \log n$ with the constant C exponential in d. Our work therefore establishes Adaptive GMRA as a fast dictionary learning algorithm, with approximation guarantees, for intrinsically low-dimensional data. We include several numerical experiments on both synthetic and real data, confirming our theoretical results and demonstrating the effectiveness of Adaptive GMRA.

Keywords: Dictionary Learning, Multi-Resolution Analysis, Adaptive Approximation, Manifold Learning, Compression

1. Introduction

We model a data set as n i.i.d. samples $\mathcal{X}_n := \{x_i\}_{i=1}^n$ from a probability measure ρ in $\mathbb{R}^D$. We make the assumption that ρ is supported on or near a set $\mathcal{M}$ of dimension $d \ll D$, and consider the problem, given $\mathcal{X}_n$, of learning a data-dependent dictionary that enables

©2019 Liao and Maggioni.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-252.html.

efficient encoding of (future) data sampled from ρ, together with fast forward and inverse transforms between $\mathbb{R}^D$ and the space of encodings.

In order to circumvent the curse of dimensionality, a popular model for data is sparsity: we say that the data is k-sparse on a suitable dictionary (i.e. a collection of vectors) $\Phi = \{\varphi_i\}_{i=1}^m \subset \mathbb{R}^D$ if each data point $x \in \mathbb{R}^d$ may be expressed as a linear combination of at most k elements of Φ. Clearly the case of interest is $k \ll D$. These *sparse representations* have been used in a variety of statistical signal processing tasks, compressed sensing, machine learning (see e.g. Protter and Elad, 2007; Peyré, 2009; Lewicki et al., 1998; Kreutz-Delgado et al., 2003; Maurer and Pontil, 2010; Chen et al., 1998; Donoho, 2006; Aharon et al., 2005; Candes and Tao, 2007, among many others), and spurred much research about how to learn data-driven dictionaries (see Gribonval et al., 2015; Vainsencher et al., 2011; Maurer and Pontil, 2010, and references therein). The algorithms used in dictionary learning are often computationally demanding, and based on high-dimensional non-convex optimization (Mairal et al., 2010). These approaches have the strength of being very general, with minimal assumptions on the geometry of the dictionary or on the distribution from which the samples are generated. This "worst-case" approach incurs bounds depending upon the ambient dimension D in general (even in the standard case of data lying on one hyperplane).

It is possible to tackle the dictionary learning problem under geometric assumptions on data sets (Maggioni et al., 2016), namely that data lie on or near a low-dimensional set $\mathcal{M}$. There are of course various possible geometric assumptions, the simplest one being that $\mathcal{M}$ is a single d-dimensional subspace, in which case Principal Component Analysis (PCA) (see Pearson, 1901; Hotelling, 1933, 1936) suffices for estimating the subspace. More generally, one may assume that data lie on a union of several low-dimensional planes instead of a single one. The problem of estimating multiple planes, called subspace clustering, is more challenging (see Fischler and Bolles, 1981; Ho et al., 2003; Vidal et al., 2005; Yan and Pollefeys, 2006; Ma et al., 2007, 2008; Chen and Lerman, 2009; Elhamifar and Vidal, 2009; Zhang et al., 2010; Liu et al., 2010; Chen and Maggioni, 2011). This model was shown effective in various applications, including image processing (Fischler and Bolles, 1981), computer vision (Ho et al., 2003) and motion segmentation (Yan and Pollefeys, 2006). Yet another type of geometric model gives rise to manifold learning, where $\mathcal{M}$ is assumed to be a d-dimensional manifold isometrically embedded in $\mathbb{R}^D$, see (Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003; Donoho and Grimes, 2003; Coifman et al., 2005a,b; Zhang and Zha, 2004) and many others. It is of interest to move beyond this model to even more general geometric models, for example where the regularity of the manifold is reduced, and data are not forced to lie exactly on a manifold, but only close to it.

Geometric Multi-Resolution Analysis (GMRA) was proposed in Chen and Maggioni (2010), refined in Allard et al. (2012). In GMRA, geometric approximations of $\mathcal{M}$ are constructed with multiscale techniques that have their roots in geometric measure theory, harmonic analysis and approximation theory. GMRA performs a multiscale tree decomposition of data and builds multiscale low-dimensional geometric approximations to $\mathcal{M}$. Given data, the cover tree algorithm (Beygelzimer et al., 2006) is run to obtain a multiscale tree in which every node is a subset of $\mathcal{M}$, called a dyadic cell, and all dyadic cells at a fixed scale form a partition of $\mathcal{M}$. After the tree is constructed, PCA is performed on the data in each cell to locally approximate $\mathcal{M}$ by the d-dimensional principal subspace, so that every point in that cell may be encoded by the d coefficients for the corresponding principal

directions. At a fixed scale $\mathcal{M}$ is thereby approximated by a piecewise linear set. In Allard et al. (2012) the performance of GMRA for volume measures on a $\mathcal{C}^s$, $s \in (1,2]$ manifold was analyzed in the continuous case (i.e. with no sampling), albeit the effectiveness of GMRA was demonstrated empirically on simulated and real-world data, but for a fixed data set, and without out-of-sample extension. In Maggioni et al. (2016), the approximation error of $\mathcal{M}$ was estimated in the non-asymptotic regime with n i.i.d. samples from a measure ρ, satisfying certain technical assumptions, supported on a thin tube of a $\mathcal{C}^2$ manifold of dimension d isometrically embedded in $\mathbb{R}^D$. The concentration bounds in Maggioni et al. (2016) depend on n and d, but not on D, successfully avoiding the curse of dimensionality caused by the ambient dimension. The assumption that ρ is supported in a tube around a manifold can account for noise and does not force the data to lie exactly on a smooth low-dimensional manifold.

In both Allard et al. (2012) and Maggioni et al. (2016), GMRA approximations are constructed on uniform partitions, at a fixed scale, in which all the cells have similar diameters. However, when the regularity of $\mathcal{M}$, such as smoothness or curvature, weighted by the ρ measure, varies at different scales and locations, uniform partitions do not yield optimal approximations. Inspired by the adaptive methods in classical multi-resolution analysis of functions (see Donoho and Johnstone, 1994, 1995; Cohen et al., 2002; Binev et al., 2005, 2007, among many others, and references therein), we propose an adaptive version of GMRA to construct low-dimensional geometric approximations of $\mathcal{M}$ on an adaptive partition, and provide finite sample performance guarantees for a larger classes of geometric structures $\mathcal{M}$ than those considered in Maggioni et al. (2016). This truly takes advantage of the multiscale structure of GMRA, and leads to simple yet provably powerful approximations for a large class of geometric objects that are not necessarily equally regular at all scales and locations.

Our main result (Theorem 8) in this paper may be paraphrased as follows: Let ρ be a probability measure supported on or near a compact d-dimensional manifold $\mathcal{M} \hookrightarrow \mathbb{R}^D$, with $d \geq 3$. Assume that ρ admits a(n unknown) multiscale decomposition satisfying the technical assumptions A1-A5 in section 2.1. Given n i.i.d. samples of ρ, the intrinsic dimension d, and a parameter κ large enough, Adaptive GMRA outputs a dictionary $\widehat{\Phi}_n = \{\widehat{\phi}_i\}_{i \in \mathcal{J}_n}$, an encoding operator $\widehat{\mathcal{D}}_n : \mathbb{R}^D \to \mathbb{R}^{(d+1)\mathcal{J}_n}$ and a decoding operator $\widehat{\mathcal{D}}_n^{-1} : \mathbb{R}^{(d+1)\mathcal{J}_n} \to \mathbb{R}^D$ that, with high probability, satisfy the following properties. For every $x \in \mathbb{R}^D$, $\|\widehat{\mathcal{D}}_n x\|_0 \leq d+1$ (i.e. only $d+1$ entries of the encoded data are non-zero), and the Mean Squared Error (MSE), over data sampled from ρ, satisfies

$$\text{MSE} := \mathbb{E}_{x \sim \rho}[\|x - \widehat{\mathcal{D}}_n^{-1}\widehat{\mathcal{D}}_n x\|^2] \lesssim \left(\frac{\log n}{n}\right)^{\frac{2s}{2s+d-2}}.$$

Here s is a regularity parameter of ρ (as in definition 5), which allows us to consider $\mathcal{M}$'s and ρ's with nonuniform regularity, varying at different locations and scales. The parameter κ is used in choosing the threshold on the geometric wavelet coefficients, and selecting from the GMRA a multiscale partition and set of local approximate tangent planes to use for encoding the data. Note that the algorithm does not need to know s (indeed, κ is independent of s), but it automatically adapts to obtain a rate that depends on s. We believe, but do not prove, that this rate is indeed optimal. As for computational complexity, constructing $\widehat{\Phi}_n$ takes $\mathcal{O}((C^d + d^2)Dn \log n)$ and computing $\widehat{\mathcal{D}}_n x$ only takes $\mathcal{O}(d(D+d^2)\log n)$, which means we have a fast transform mapping data to their sparse encoding on the dictionary.

In Adaptive GMRA, the dictionary is composed of the low-dimensional planes on adaptive partitions and the encoding operator transforms a point to the local affine $d+1$ principal coefficients of the data in a piece of the partition (the first affine principal component here means the local mean). We state this results in terms of encoding and decoding to stress that learning the geometry in fact yields efficient representations of data, which may be used for performing signal processing tasks in a domain where the data admit a sparse representation, e.g. in compressive sensing or estimation problems (see Iwen and Maggioni, 2013; Chen et al., 2012; Eftekhari and Wakin, 2015). Adaptive GMRA is designed towards robustness, both in the sense of tolerance to noise and to model error (i.e. data not lying on a manifold). We assume d is given throughout this paper. If not, we refer to Little et al. (2017, 2009a,b) for the estimation of intrinsic dimensionality.

The paper is organized as follows. Our main results, including the construction of GMRA, Adaptive GMRA and their finite sample analysis, are presented in Section 2. We show numerical experiments in Section 3. The detailed analysis of GMRA and Adaptive GMRA is presented in Section 4. In Section 5, we discuss the computational complexity of Adaptive GMRA and extend our work to adaptive orthogonal GMRA. Proofs are collected in the appendix.

Notation. We will introduce some basic notation here. $f \lesssim g$ means that there exists a constant C independent on any variable upon which f and g depend, such that $f \leq Cg$; similarly for $\gtrsim$. $f \asymp g$ means that $f \lesssim g$ and $f \gtrsim g$. The cardinality of a set A is denoted by $\#A$. For $x \in \mathbb{R}^D$, $\|x\|$ denotes the Euclidean norm and $B_r(x)$ denotes the Euclidean ball of radius r centered at x. Given a subspace $V \in \mathbb{R}^D$, we denote its dimension by $\dim(V)$ and the orthogonal projection onto V by Proj_V. If A is a linear operator on $\mathbb{R}^D$, $||A||$ is its operator norm. The identity operator is denoted by $\mathbb{I}$.

2. Main results

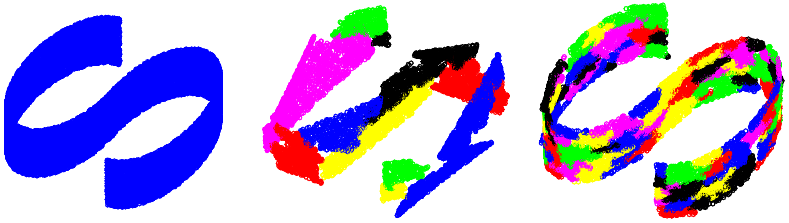
GMRA was proposed in Allard et al. (2012) to efficiently represent points on or near a low-dimensional manifold in high dimensions. We refer the reader to that paper for details of the construction, and we summarize here the main ideas in order to keep the presentation self-contained. The construction of GMRA involves the following steps:

(i) construct a *multiscale tree* $\mathcal{T}$ and the associated decomposition of $\mathcal{M}$ into nested cells $\{C_{j,k}\}_{k \in \mathcal{K}_j, j \in \mathbb{Z}}$ where j represents scale and k location;

(ii) perform *local PCA* on each $C_{j,k}$: let the mean ("center") be $c_{j,k}$ and the d-dim principal subspace $V_{j,k}$. Define $\mathcal{P}_{j,k}(x) := c_{j,k} + \text{Proj}_{V_{j,k}}(x - c_{j,k})$.

(iii) construct a *"difference" subspace* $W_{j+1,k'}$ capturing $\mathcal{P}_{j,k}(C_{j,k}) - \mathcal{P}_{j+1,k'}(C_{j+1,k'})$, for each $C_{j+1,k'} \subseteq C_{j,k}$ (these quantities are associated with the refinement criterion in Adaptive GMRA).

$\mathcal{M}$ may be approximated, at each scale j, by its projection $\mathcal{P}_{\Lambda_j}$ onto the family of linear sets $\Lambda_j := \{\mathcal{P}_{j,k}(C_{j,k})\}_{k \in \mathcal{K}_j}$. For example, linear approximations of the S-manifold at scale 6 and 10 are displayed in Figure 1. In a variety of distances, $\mathcal{P}_{\Lambda_j}(\mathcal{M}) \to \mathcal{M}$. In practice $\mathcal{M}$ is



(a) S-manifold (b) scale 6 (c) scale 10

Figure 1: (a) S-manifold; (b,c) Linear approximations at scale $6, 10$.

unknown, and the construction above is carried over on training data, and its result is random with respect to the training samples. Naturally we are interested in the performance of the construction on new samples. This is analyzed in a setting of "smooth manifold+noise" in Maggioni et al. (2016). When the regularity (such as smoothness or curvature) of $\mathcal{M}$ varies at different locations and scales, linear approximations on fixed uniform partitions are not optimal. Inspired by adaptive methods in classical multi-resolution analysis (see Cohen et al., 2002; Binev et al., 2005, 2007), we propose an adaptive version of GMRA which learns adaptive and near-optimal approximations.

We will start with the multiscale tree decomposition in Section 2.1 and present GMRA and Adaptive GMRA in Section 2.3 and 2.4 respectively.

2.1. Multiscale partitions and trees

A multiscale set of partitions of $\mathcal{M}$ with respect to the probability measure ρ is a family of sets $\{C_{j,k}\}_{k\in\mathcal{K}_j, j\in\mathbb{Z}}$, called dyadic cells, satisfying Assumptions (A1-A5) below, for all integers $j \geq j_{\min}$:

(A1) for any $k \in \mathcal{K}_j$ and $k' \in \mathcal{K}_{j+1}$, either $C_{j+1,k'} \subseteq C_{j,k}$ or $\rho(C_{j+1,k'} \cap C_{j,k}) = 0$. We denote the children of $C_{j,k}$ by $\mathscr{C}(C_{j,k}) = \{C_{j+1,k'} : C_{j+1,k'} \subseteq C_{j,k}\}$. We assume that $a_{\min} \leq \#\mathscr{C}(C_{j,k}) \leq a_{\max}$. Also for every $C_{j,k}$, there exists a unique $k' \in \mathcal{K}_{j-1}$ such that $C_{j,k} \subseteq C_{j-1,k'}$. We call $C_{j-1,k'}$ the parent of $C_{j,k}$.

(A2) $\rho(\mathcal{M} \setminus \cup_{k\in\mathcal{K}_j} C_{j,k}) = 0$, i.e. $\Lambda_j := \{C_{j,k}\}_{k\in\mathcal{K}_j}$ is a cover for $\mathcal{M}$.

(A3) $\exists \theta_1 > 0 : \#\Lambda_j \leq 2^{jd}/\theta_1$.

(A4) $\exists \theta_2 > 0$ such that, if x is drawn from $\rho_{|C_{j,k}}$, then a.s. $\|x - c_{j,k}\| \leq \theta_2 2^{-j}$.

(A5) Let $\lambda_1^{j,k} \geq \lambda_2^{j,k} \geq \ldots \geq \lambda_D^{j,k}$ be the eigenvalues of the covariance matrix $\Sigma_{j,k}$ of $\rho_{|C_{j,k}}$, defined in Table 1. Then:

(i) $\exists \theta_3 > 0$ such that $\forall j \geq j_{\min}$ and $k \in \mathcal{K}_j$, $\lambda_d^{j,k} \geq \theta_3 2^{-2j}/d$,

(ii) $\exists \theta_4 \in (0,1)$ such that $\lambda_{d+1}^{j,k} \leq \theta_4 \lambda_d^{j,k}$.

(A1) implies that the $\{C_{j,k}\}_{k\in\mathcal{K}_j, j\geq j_{\min}}$ are associated with a tree structure, and with some abuse of notation we call the above *tree decompositions*. (A1)-(A5) are natural assumptions, easily satisfied by natural multiscale decompositions when $\mathcal{M}$ is a d-dimensional manifold isometrically embedded in $\mathbb{R}^D$: see the work (Maggioni et al., 2016) for a detailed discussion, where the connections between the constants θ_i's and geometric properties of $\mathcal{M}$ (curvatures, reach, etc...) are also discussed. (A2) guarantees that the cells at scale j form a partition of $\mathcal{M}$; (A3) says that there are at most $2^{jd}/\theta_1$ dyadic cells at scale j. (A4) ensures $\mathrm{diam}(C_{j,k}) \lesssim 2^{-j}$. When $\mathcal{M}$ is a d-dimensional manifold, (A5)(i) is the condition that the best rank d approximation to $\Sigma_{j,k}$ is close to the covariance matrix of a d-dimensional Euclidean ball, while (A5)(ii) imposes that the $(d+1)$-th eigenvalue is smaller that the d-th eigenvalue, i.e. the set has significantly larger variances in d directions than in all the remaining ones. The conditions generalize those in (Allard et al., 2012) (which corresponded to the case when $\mathcal{M}$ is a manifold) and in (Maggioni et al., 2016), for example by not assuming that all sets $\{C_{j,k}\}_k$ (for any fixed j) have roughly the same volume, and also by weakening (A5). These changes enlarge the class of measures ρ and sets $\mathcal{M}$ that we consider here, for exampling allowing for a highly nonuniform measure ρ, and an $\mathcal{M}$ substantially "thickened" in many dimensions.

We will construct such $\{C_{j,k}\}_{k\in\mathcal{K}_j, j\geq j_{\min}}$ in Section 2.6. In our construction (A1-A4) is satisfied when ρ is a regular doubling probability measure[1] (see Christ, 1990; Deng and Han, 2008). If we further assume that $\mathcal{M}$ is a d-dimensional $\mathcal{C}^s$, $s \in (1,2]$ closed manifold isometrically embedded in $\mathbb{R}^D$, then (A5) is satisfied as well (See Proposition 14).

It may happen that at the coarsest scales conditions (A3)-(A5) are satisfied but with very poor constants θ: it will be clear that in all that follows we may discard a few coarse scales (i.e. by enlarging $j_{\min}$), and only work at scales that are fine enough and for which (A3)-(A5) truly capture the local geometry of $\mathcal{M}$.

Some notation: a master tree $\mathcal{T}$ is associated with $\{C_{j,k}\}_{k\in\mathcal{K}_j, j\geq j_{\min}}$ (using property (A1)), constructed on $\mathcal{M}$; since $C_{j,k}$'s at scale j have similar diameters, $\Lambda_j := \{C_{j,k}\}_{k\in\mathcal{K}_j}$ is called a *uniform partition at scale* j. A proper subtree $\tilde{\mathcal{T}}$ of $\mathcal{T}$ is a collection of nodes of $\mathcal{T}$ with the properties: (i) the root node is in $\tilde{\mathcal{T}}$, (ii) if $C_{j,k}$ is in $\tilde{\mathcal{T}}$ then the parent of $C_{j,k}$ is also in $\tilde{\mathcal{T}}$. Any finite proper subtree $\tilde{\mathcal{T}}$ is associated with a unique partition $\Lambda = \Lambda(\tilde{\mathcal{T}})$ which consists of its outer leaves, by which we mean those $C_{j,k} \in \mathcal{T}$ such that $C_{j,k} \notin \tilde{\mathcal{T}}$ but its parent is in $\tilde{\mathcal{T}}$.

2.2. Empirical GMRA

In practice the master tree $\mathcal{T}$ is not given, nor can be constructed since $\mathcal{M}$ is not known: we will construct one on samples by running a variation of the cover tree algorithm (see Beygelzimer et al., 2006), which only creates candidate "centers" for the $C_{j,k}$, by adding a multiscale partitioning step. From now on we denote the training data by $\mathcal{X}_{2n}$. We randomly split the data into two disjoint groups such that $\mathcal{X}_{2n} = \mathcal{X}'_n \cup \mathcal{X}_n$ where $\mathcal{X}'_n = \{x'_1, \ldots, x'_n\}$ and $\mathcal{X}_n = \{x_1, \ldots, x_n\}$, apply our variation on cover trees on $\mathcal{X}'_n$ to construct a tree satisfying (A1-A5) (see section 2.6). After the tree is constructed, we assign points in the second

1. ρ is regular doubling if there exists $C_1 > 0$ such that $C_1^{-1} r^d \leq \rho(\mathcal{M} \cap B_r(x)) \leq C_1 r^d$ for any $x \in \mathcal{M}$ and $r > 0$. C_1 is called the doubling constant of ρ.

	GMRA	Empirical GMRA
Linear projection on $C_{j,k}$	$\mathcal{P}_{j,k}(x) := c_{j,k} + \mathrm{Proj}_{V_{j,k}}(x - c_{j,k})$	$\widehat{\mathcal{P}}_{j,k}(x) := \widehat{c}_{j,k} + \mathrm{Proj}_{\widehat{V}_{j,k}}(x - \widehat{c}_{j,k})$
Linear projection at scale j	$\mathcal{P}_j := \sum_{k\in\mathcal{K}_j} \mathcal{P}_{j,k}\mathbf{1}_{j,k}$	$\widehat{\mathcal{P}}_j := \sum_{k\in\mathcal{K}_j} \widehat{\mathcal{P}}_{j,k}\mathbf{1}_{j,k}$
Measure	$\rho(C_{j,k})$	$\widehat{\rho}(C_{j,k}) = \widehat{n}_{j,k}/n$
Center	$c_{j,k} := \mathbb{E}_{j,k}x$	$\widehat{c}_{j,k} := \frac{1}{\widehat{n}_{j,k}} \sum_{x_i\in C_{j,k}} x_i$
Principal subspaces	$V_{j,k}$ minimizes $\mathbb{E}_{j,k}\lVert x - c_{j,k} - \mathrm{Proj}_V(x - c_{j,k})\rVert^2$ among d-dim subspaces	$\widehat{V}_{j,k}$ minimizes $\frac{1}{\widehat{n}_{j,k}} \sum_{x_i\in C_{j,k}} \lVert x - \widehat{c}_{j,k} - \mathrm{Proj}_V(x - \widehat{c}_{j,k})\rVert^2$ among d-dim subspaces
Covariance matrix	$\Sigma_{j,k} := \mathbb{E}_{j,k}(x - c_{j,k})(x - c_{j,k})^T$	$\widehat{\Sigma}_{j,k} := \frac{1}{\widehat{n}_{j,k}} \sum_{x_i\in C_{j,k}} (x_i - \widehat{c}_{j,k})(x_i - \widehat{c}_{j,k})^T$
Inner product with respect to ρ	$\langle \mathcal{P}X, \mathcal{Q}X\rangle := \int_{\mathcal{M}} \langle \mathcal{P}x, \mathcal{Q}x\rangle d\rho$	$1/n \sum_{x_i\in\mathcal{X}_n} \langle \mathcal{P}x_i, \mathcal{Q}x_i\rangle$
Norm with respect to ρ	$\lVert\mathcal{P}X\rVert := \left(\int_{\mathcal{M}} \lVert\mathcal{P}x\rVert^2 d\rho\right)^{\frac{1}{2}}$	$\left(1/n \sum_{x_i\in\mathcal{X}_n} \lVert\mathcal{P}x_i\rVert^2\right)^{\frac{1}{2}}$

Table 1: This table summarizes GMRA-related quantities and their empirical counterparts (Allard et al., 2012; Maggioni et al., 2016). $\mathbf{1}_{j,k}$ is the indicator function on $C_{j,k}$ (i.e.,$\mathbf{1}_{j,k}(x) = 1$ if $x \in C_{j,k}$ and 0 otherwise). Here $\mathbb{E}_{j,k}$ stands for expectation with respect to the conditional distribution $d\rho_{|C_{j,k}}$. The measure of $C_{j,k}$ is $\rho(C_{j,k})$ and the empirical measure is $\widehat{\rho}(C_{j,k}) = \widehat{n}_{j,k}/n$ where $\widehat{n}_{j,k}$ is the number of points in $C_{j,k}$. $V_{j,k}$ and $\widehat{V}_{j,k}$ are the eigen-spaces associated with the largest d eigenvalues of $\Sigma_{j,k}$ and $\widehat{\Sigma}_{j,k}$ respectively. Here $\mathcal{P}, \mathcal{Q}$: $\mathcal{M} \to \mathbb{R}^D$ are two operators.

half of data $\mathcal{X}_n$, to the appropriate cells. In this way we obtain a family of multiscale partitions for the points in $\mathcal{X}_n$, which we truncate to the largest subtree whose leaves contain at least d points in $\mathcal{X}_n$. This subtree is called the *data master tree*, denoted by $\mathcal{T}^n$. We then use $\mathcal{X}_n$ to perform local PCA to obtain the empirical mean $\widehat{c}_{j,k}$ and the empirical d-dimensional principal subspace $\widehat{V}_{j,k}$ on each $C_{j,k}$. Define the empirical projection $\widehat{\mathcal{P}}_{j,k}(x) := \widehat{c}_{j,k} + \mathrm{Proj}_{\widehat{V}_{j,k}}(x - \widehat{c}_{j,k})$ for $x \in C_{j,k}$. Table 1 summarizes the GMRA-related quantities and their empirical counterparts.

2.3. Geometric Multi-Resolution Analysis: uniform partitions

GMRA with respect to the distribution ρ associated with the multiscale tree $\mathcal{T}$ consists a collection of piecewise affine projectors $\{\mathcal{P}_j : \mathbb{R}^D \to \mathbb{R}^D\}_{j\geq j_{\min}}$ on the multiscale partitions $\{\Lambda_j := \{C_{j,k}\}_{k\in\mathcal{K}_j}\}_{j\geq j_{\min}}$. At scale j, $\mathcal{M}$ is approximated by the piecewise linear sets $\{\mathcal{P}_{j,k}(C_{j,k})\}_{k\in\mathcal{K}_j}$. The approximation error of $\mathcal{M}$ by the empirical linear sets $\{\widehat{\mathcal{P}}_{j,k}(C_{j,k})\}_{k\in\mathcal{K}_j}$ is defined as:

$$\mathbb{E}\lVert X - \widehat{\mathcal{P}}_j X\rVert^2 = \mathbb{E}\int_{\mathcal{M}} \lVert x - \widehat{\mathcal{P}}_j x\rVert^2 d\rho = \mathbb{E}\sum_{k\in\mathcal{K}_j}\int_{C_{j,k}} \lVert x - \widehat{\mathcal{P}}_{j,k} x\rVert^2 d\rho$$

where $\widehat{\mathcal{P}}_j$ and $\widehat{\mathcal{P}}_{j,k}$ are built from random samples $x_i \sim \rho$ (according to the GMRA algorithm), X is a random vector distributed according to ρ, and the expectation is taken over X. The squared approximation error above is also called the Mean Square Error (MSE) of GMRA. In order to understand the error, we split it into a bias term and a variance term:

$$\mathbb{E}\lVert X - \widehat{\mathcal{P}}_j X\rVert \leq \underbrace{\lVert X - \mathcal{P}_j X\rVert}_{\text{bias}} + \mathbb{E}\underbrace{\lVert \mathcal{P}_j X - \widehat{\mathcal{P}}_j X\rVert}_{\sqrt{\text{variance}}}. \tag{1}$$

To bound the bias term, we need regularity assumptions on ρ, while for the variance term we prove concentration bounds of the relevant quantities around their expected values.

For a fixed distribution ρ, the approximation error of $\mathcal{M}$ at scale j, measured by $\lVert X - \mathcal{P}_j X\rVert$, decays at a rate dependent on the regularity of $\mathcal{M}$ in the ρ-measure (see Allard et al., 2012). We quantify the regularity of ρ as follows:

Definition 1 (Model class $\mathcal{A}_s$) *A probability measure ρ supported on $\mathcal{M}$ is in $\mathcal{A}_s$ if*

$$|\rho|_{\mathcal{A}_s} = \sup_{\mathcal{T}} \inf\{A_0 : \lVert X - \mathcal{P}_j X\rVert \leq A_0 2^{-js}, \forall\, j \geq j_{\min}\} < \infty\,, \tag{2}$$

where $\mathcal{T}$ varies over the set, assumed non-empty, of multiscale tree decompositions satisfying Assumptions (A1-A5).

We capture the case where the L^2 approximation error is roughly the same on every cell with the following definition:

Definition 2 (Model class $\mathcal{A}_s^\infty$) *A probability measure ρ supported on $\mathcal{M}$ is in $\mathcal{A}_s^\infty$ if*

$$|\rho|_{\mathcal{A}_s^\infty} = \sup_{\mathcal{T}} \inf\{A_0 : \lVert (X - \mathcal{P}_{j,k}X)\mathbf{1}_{j,k}\rVert \leq A_0 2^{-js}\sqrt{\rho(C_{j,k})},\ \forall\, k \in \mathcal{K}_j, j \geq j_{\min}\} < \infty \tag{3}$$

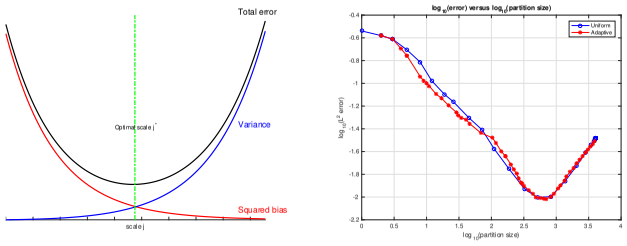
where $\mathcal{T}$ varies over the set, assumed non-empty, of multiscale tree decompositions satisfying Assumptions (A1-A5).

Clearly $\mathcal{A}_s^\infty \subset \mathcal{A}_s$. Also, since $\mathrm{diam}(C_{j,k}) \leq 2\theta_2 2^{-j}$, necessarily $\lVert(\mathbb{I} - \mathcal{P}_{j,k})\mathbf{1}_{j,k}X\rVert \leq \theta_2 2^{-j}\sqrt{\rho(C_{j,k})}$, $\forall\, k \in \mathcal{K}_j, j \geq j_{\min}$, and therefore $\rho \in \mathcal{A}_1^\infty$ in any case. Moreover, these classes contain suitable measures supported on manifolds:

Proposition 3 *Let $\mathcal{M}$ be a closed manifold of class $\mathcal{C}^s$, $s \in (1, 2]$ isometrically embedded in $\mathbb{R}^D$, and ρ be a doubing probability measure on $\mathcal{M}$ with the doubling constant C_1. Then our construction of $\{C_{j,k}\}_{k\in\mathcal{K}_j, j\geq j_{\min}}$ in Section 2.6 satisfies (A1-A5), and $\rho \in \mathcal{A}_s^\infty$.*

The proof is postponed to Appendix A.2.

Example 1 *We consider the d-dim S-manifold whose x_1 and x_2 coordinates are on an S-shaped curve and x_i ranges in $[0, 1]$ for $i = 3, \ldots, d + 1$. By the Proposition just stated, the volume measure on this S-manifold is in $\mathcal{A}_2^\infty$. Numerically one can identify s from data sampled from $\rho \in \mathcal{A}_s$ as the slope of the line approximating $\log_{10}\lVert X - \mathcal{P}_j X\rVert$ as a function of $\log_{10} r_j$ where r_j is the average diameter of $C_{j,k}$'s at scale j. Our numerical experiments in Figure 5 (b) give rise to $s \approx 2.0, 2.1, 2.1$ when $d = 3, 4, 5$ respectively.*



(a) Bias and variance tradeoff (b) Error versus the partition size

Figure 2: (a) Plot of the bias and variance estimates in Eq. (1), with $s = 2, d = 5, n = 100$. (b) shows the approximation error on test data versus the partition size in GMRA and Adaptive GMRA for the 3-dim S-manifold. When the partition size is between 1 and $10^{2.8}$, the bias dominates the error so the error decreases; after that, the variance dominates the error, which becomes increasing.

Example 2 *As a comparison we consider the d-dimensional Z-manifold whose x_1 and x_2 coordinates are on a Z-shaped curve and x_i ranges in $[0, 1]$, for $i = 3, \ldots, d + 1$. Volume measure on the Z manifold is in $\mathcal{A}_{1.5}$ (see appendix B.2). Our numerical experiments in Figure 5 (c) give rise to $s \approx 1.5, 1.7, 1.6$ when $d = 3, 4, 5$ respectively.*

The squared bias in (1) satisfies $\|X - \mathcal{P}_j X\|^2 \leq |\rho|^2_{\mathcal{A}_s} 2^{-2js}$ whenever $\rho \in \mathcal{A}_s$ (by definition of $\mathcal{A}_s$). In Proposition 16 we will show that the variance term is estimated in terms of the sample size n and the scale j as follows:

$$\mathbb{E}\|\mathcal{P}_j X - \widehat{\mathcal{P}}_j X\|^2 \leq \frac{d^2 \#\Lambda_j \log[\alpha d \#\Lambda_j]}{\beta 2^{2j} n} = \mathcal{O}\left(\frac{j 2^{j(d-2)}}{n}\right),$$

where α, β are constants depending on θ_2, θ_3. In the case $d = 1$ both the squared bias and the variance decrease as j increases, so choosing the finest scale of the data tree $\mathcal{T}^n$ yields the best rate of convergence. When $d \geq 2$, the squared bias decreases but the variance increases as j gets large as shown Figure 2, as a manifestation of the classical bias-variance tradeoff, except that it arises here in a geometric setting (a related instance of this phenomenon appears in Canas et al. (2012)). By choosing a proper scale j^* to balance these two terms, we obtain the following rate of convergence for empirical GMRA truncated at scale j^*:

Theorem 4 *Suppose $\rho \in \mathcal{A}_s$ for $s \geq 1$. Let $\nu > 0$ be arbitrary and fix $\mu > 0$. Let j^* be chosen such that*

$$2^{-j^*} = \begin{cases} \mu \frac{\log n}{n} & \text{for } d = 1 \\ \mu \left(\frac{\log n}{n}\right)^{\frac{1}{2s+d-2}}, & \text{for } d \geq 2 \end{cases} \tag{4}$$

then there exists $C_1 := C_1(\theta_1, \theta_2, \theta_3, \theta_4, d, \nu, \mu)$ and $C_2 := C_2(\theta_1, \theta_2, \theta_3, \theta_4, d, \mu)$ such that:

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{j^*} X\| \geq (|\rho|_{\mathcal{A}_s} \mu^s + C_1) \frac{\log n}{n}\right\} \leq C_2 n^{-\nu}, \quad \text{for } d = 1, \tag{5}$$

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{j^*} X\| \geq (|\rho|_{\mathcal{A}_s} \mu^s + C_1) \left(\frac{\log n}{n}\right)^{\frac{s}{2s+d-2}}\right\} \leq C_2 n^{-\nu}, \quad \text{for } d \geq 2\,. \tag{6}$$

Theorem 4 is proved in Section 4.2. From the perspective of dictionary learning, it says that GMRA provides a dictionary Φ_{j^*} of cardinality $\asymp dn/\log n$ for $d = 1$ and $\asymp d(n/\log n)^{\frac{d}{2s+d-2}}$ for $d \geq 2$, consisting of the principal directions in each of the $C_{j^*,k}$'s (forming the columns of $\widehat{V}_{j^*,x}$) and the means of the $C_{j^*,k}$'s, so that every x sampled from ρ (and not just samples in the training data) may be encoded with a vector with $d + 1$ nonzero entries: one entry encodes the location k of x on the tree, e.g. $(j^*, x) = (j^*, k)$ such that $x \in C_{j^*,k}$, and the other d entries are the coefficients $\widehat{V}^T_{j^*,x}(x - \widehat{c}_{j^*,x})$. We also remind the reader that GMRA automatically constructs a fast transform mapping points x to the vector representing Φ_{j^*} (See Allard et al. (2012); Maggioni et al. (2016) for a discussion). Note that by choosing ν large enough in the Theorem,

$$(6) \implies \text{MSE} = \mathbb{E}\|X - \widehat{\mathcal{P}}_{j^*} X\|^2 \lesssim \left(\frac{\log n}{n}\right)^{\frac{2s}{2s+d-2}},$$

and (5) implies MSE $\lesssim (\frac{\log n}{n})^2$ for $d = 1$. Clearly, one could fix a desired MSE of size ε^2, and obtain a dictionary of size dependent only on ε and independent of n, for n sufficiently large, thereby obtaining a way of compressing data (for further discussion on this point see Maggioni et al. (2016), where also a special case of Theorem 4 with $s = 2$ was proved).

2.4. Geometric Multi-Resolution Analysis: Adaptive Partitions

The performance guarantee in Theorem 4 is not fully satisfactory for two reasons: (i) the regularity parameter s is required to be known to choose the optimal scale j^*, and this parameter is typically unknown in any practical setting, and (ii) none of the uniform partitions $\{C_{j,k}\}_{k \in \mathcal{K}_j}$ will be optimal if the regularity of ρ (and/or $\mathcal{M}$) varies at different locations and scales. This lack of uniformity in regularity can appear in a wide variety of data sets for various reasons: when clusters exist that have cores denser than the remaining regions of space, when sampled trajectories of a dynamical system linger in certain regions of space for much longer time intervals than others (e.g. metastable states in molecular

	Definition (infinite sample)	Empirical version
Difference operator	$\mathcal{Q}_{j,k} := (\mathcal{P}_j - \mathcal{P}_{j+1})\mathbf{1}_{j,k}$	$\widehat{\mathcal{Q}}_{j,k} := (\widehat{\mathcal{P}}_j - \widehat{\mathcal{P}}_{j+1})\mathbf{1}_{j,k}$
Norm of difference	$\Delta^2_{j,k} := \int_{C_{j,k}} \|\mathcal{Q}_{j,k} x\|^2 d\rho$	$\widehat{\Delta}^2_{j,k} := \frac{1}{n} \sum_{x_i \in C_{j,k}} \|\widehat{\mathcal{Q}}_{j,k} x_i\|^2$

Table 2: Refinement criterion and the empirical counterparts

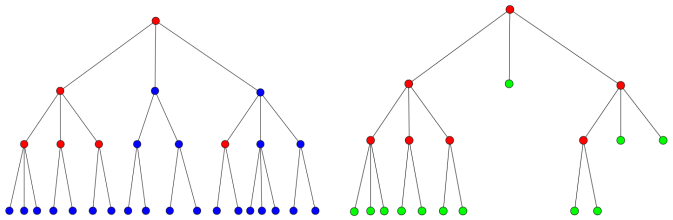


Figure 3: Left: a master tree in which red nodes satisfy $\Delta_{j,k} \geq 2^{-j}\tau_n$ but blue nodes do not. Right: the subtree of the red nodes is the smallest proper subtree that contains all the nodes satisfying $\Delta_{j,k} \geq 2^{-j}\tau_n$, i.e. were red in the figure on the left. Green nodes form the adaptive partition.

dynamics (Rohrdanz et al., 2011; Zheng et al., 2011)), in data sets of images where details exist at different level of resolutions, affecting regularity at different scales in the ambient space, and so on. To fix the ideas we consider again one simplest manifestations of this phenomenon in the examples considered above: uniform partitions work well for the volume measure on the S-manifold but are not optimal for the volume measure on the Z-manifold, for which the ideal partition is coarse on flat regions but finer at and near the corners (see Figure 4). In applications, for example to mesh approximation, it is often the case that the point clouds to be approximated are not uniformly smooth and include different levels of details at different locations and scales (see Figure 9). We therefore propose an adaptive version of GMRA that automatically adapts to the regularity of the data and choose a near-optimal partition.

We expect $\Delta_{j,k}$ defined in Table 2 to be small on approximately flat regions, and large $\Delta_{j,k}$ at many scales at irregular locations. We also expect $\widehat{\Delta}_{j,k}$ to have the same behavior, at least when $\widehat{\Delta}_{j,k}$ is with high confidence close to $\Delta_{j,k}$. We see this phenomenon represented in Figure 4 (a,b): as j increases, for the S-manifold $\|\widehat{\mathcal{P}}_{j+1}x_i - \widehat{\mathcal{P}}_j x_i\|$ decays uniformly at all points, while for the Z-manifold, the same quantity decays rapidly on flat regions but remains large even at fine scales near the corners (where "near" is scale-dependent, decreasing with scale). We wish to include in our approximation the nodes where this quantity is large, since we may expect a large improvement in approximation by including such nodes. However if too few samples exist in a node, then this quantity is not to be trusted, because its variance is large. It turns out that it is enough to consider the following criterion: let $\widehat{\mathcal{T}}_{\tau_n}$ be the smallest proper subtree of $\mathcal{T}^n$ that contains all $C_{j,k} \in \mathcal{T}^n$ for which $\widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n$ where $\tau_n = \kappa\sqrt{(\log n)/n}$. Crucially, κ may be chosen independently of the regularity index (see Theorem 8). Empirical Adaptive GMRA returns piecewise affine projectors on $\widehat{\Lambda}_{\tau_n}$, the partition associated with the outer leaves of $\widehat{\mathcal{T}}_{\tau_n}$. Our algorithm is summarized in Algorithm 1.

Algorithm 1 - Adaptive GMRA

Input: data $\mathcal{X}_{2n} = \mathcal{X}'_n \cup \mathcal{X}_n$, intrinsic dimension d, threshold κ

Output: $\mathcal{T}^n$, $\{C_{j,k}\}$, $\widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}}$: multiscale tree, corresponding cells and adaptive piecewise linear projectors on an adaptive partition.

1: Construct $\mathcal{T}^n$ and $\{C_{j,k}\}$ from $\mathcal{X}'_n$

2: Now use $\mathcal{X}_n$. Compute $\widehat{\mathcal{P}}_{j,k}$ and $\widehat{\Delta}_{j,k}$ on every node $C_{j,k} \in \mathcal{T}^n$.

3: $\widehat{\mathcal{T}}_{\tau_n} \leftarrow$ smallest proper subtree of $\mathcal{T}^n$ containing all $C_{j,k} \in \mathcal{T}^n$: $\widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n$ where $\tau_n = \kappa\sqrt{(\log n)/n}$.

4: $\widehat{\Lambda}_{\tau_n} \leftarrow$ the partition associated with outer leaves of $\widehat{\mathcal{T}}_{\tau_n}$

5: $\widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}} \leftarrow \sum_{C_{j,k}\in\widehat{\Lambda}_{\tau_n}} \widehat{\mathcal{P}}_{j,k}\mathbf{1}_{j,k}$.

Adaptive partitions may be effectively selected with a criterion that determines whether or not a cell should participate in the adaptive partition. The quantities involved in the selection and their empirical version are summarized in Table 2.

We will provide a finite sample performance guarantee of the empirical Adaptive GMRA for a model class that is more general than $\mathcal{A}_s^\infty$. Given any fixed threshold $\eta > 0$, we let $\mathcal{T}_{(\rho,\eta)}$ be the smallest proper subtree of $\mathcal{T}$ that contains all $C_{j,k} \in \mathcal{T}$ for which $\Delta_{j,k} \geq 2^{-j}\eta$. The corresponding adaptive partition $\Lambda_{(\rho,\eta)}$ consists of the outer leaves of $\mathcal{T}_{(\rho,\eta)}$. We let $\#_j\mathcal{T}_{(\rho,\eta)}$ be the number of cells in $\mathcal{T}_{(\rho,\eta)}$ at scale j.

Definition 5 (Model class $\mathcal{B}_s$) *In the case $d \geq 3$, given $s > 0$, a probability measure ρ supported on $\mathcal{M}$ is in $\mathcal{B}_s$ if ρ satisfies the following regularity condition:*

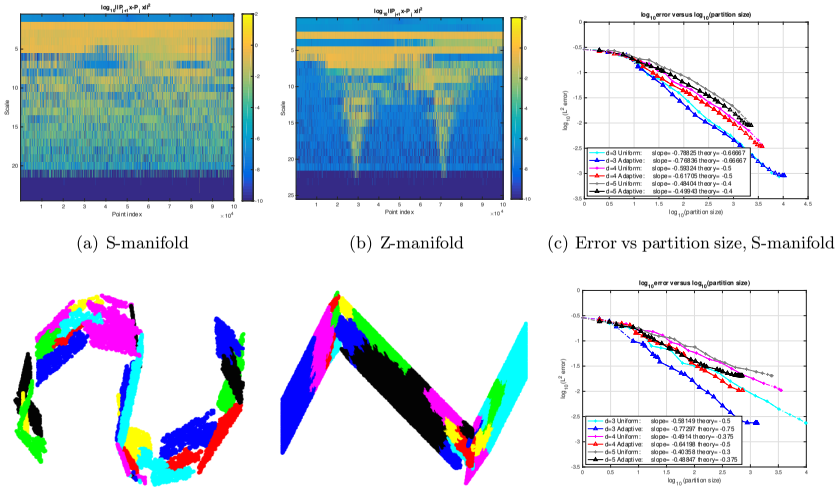
$$|\rho|_{\mathcal{B}_s} := \left(\sup_{\mathcal{T}} \sup_{\eta>0} \eta^p \sum_{j\geq j_{\min}} 2^{-2j}\#_j\mathcal{T}_{(\rho,\eta)}\right)^{\frac{1}{p}} < \infty, \quad \text{with } p = \frac{2(d-2)}{2s+d-2} \tag{7}$$

where $\mathcal{T}$ varies over the set, assumed nonempty, of multiscale tree decompositions satisfying Assumptions (A1-A5).

For elements in the model class $\mathcal{B}_s$ we have control on the growth rate of the truncated tree $\mathcal{T}_{(\rho,\eta)}$ as η decreases, namely it is $\mathcal{O}(\eta^{-p})$. Our key estimates on variance and sample complexity in Lemma 15 indicate that the natural measure of the complexity of $\mathcal{T}_{(\rho,\eta)}$ is the weighted tree complexity measure $\sum_{j\geq j_{\min}} 2^{-2j}\#_j\mathcal{T}_{(\rho,\eta)}$ in the definition above. First of all, the class $\mathcal{B}_s$ is indeed larger than $\mathcal{A}_s^\infty$ (see appendix A.4 for a proof):

Lemma 6 *$\mathcal{B}_s$ is a more general model class than $\mathcal{A}_s^\infty$: if $\rho \in \mathcal{A}_s^\infty$, then $\rho \in \mathcal{B}_s$ and $|\rho|_{\mathcal{B}_s} \lesssim |\rho|_{\mathcal{A}_s^\infty}$.*

Example 3 *The volume measures on the d-dim ($d \geq 3$) S-manifold and Z-manifold are in $\mathcal{B}_2$ and $\mathcal{B}_{1.5(d-2)/(d-3)}$ respectively (see appendix B). In numerical experiments, s can be approximated by the negative of the slope in the log-log plot of $\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_\eta}X\|^{d-2}$ versus the weighted complexity of the truncated tree according to Eq. (9): see numerical examples in Figure 5.*



(a) S-manifold (b) Z-manifold (c) Error vs partition size, S-manifold

(d) Adaptive approx. of S (e) Adaptive approx. of Z (f) Error vs partition size, Z-manifold

Figure 4: (a,b): $\log_{10} \|\widehat{\mathcal{P}}_j(x_i) - \widehat{\mathcal{P}}_{j+1}(x_i)\|$ from the coarsest scale (top) to the finest scale (bottom), with columns indexed by points, which, for visualization purposes only, are sorted roughly from "left to right" on the manifold. (d,e): adaptive approximations: for the S-manifold the adaptive approximation is close to a uniform approximation, but for the Z-manifold it contains few large pieces near the almost-flat regions, and several small pieces near the "corners". (c,f): log-log plot of the approximation error versus the partition size in GMRA and Adaptive GMRA respectively. Theoretically, the slope is $-2/d$ in both GMRA and Adaptive GMRA for the S-manifold. For the Z-manifold, the slope is $-1.5/d$ in GMRA and $-1.5/(d-1)$ in Adaptive GMRA (see appendix B).

We also need a quasi-orthogonality condition which says that the operators $\{\mathcal{Q}_{j,k}\}_{k\in\mathcal{K}_j, j\geq j_{\min}}$ applied on $\mathcal{M}$ are mostly orthogonal across scales and/or $\|\mathcal{Q}_{j,k}X\|$ quickly decays.

Definition 7 (Quasi-orthogonality) *There exists a constant $B_0 > 0$ such that for any proper subtree $\tilde{\mathcal{T}}$ of any master tree $\mathcal{T}$ satisfying Assumptions (A1-A5),*

$$\| \sum_{C_{j,k}\notin\tilde{\mathcal{T}}} \mathcal{Q}_{j,k}X\|^2 \leq B_0 \sum_{C_{j,k}\notin\tilde{\mathcal{T}}} \|\mathcal{Q}_{j,k}X\|^2. \tag{8}$$

We postpone further discussion of this condition to Section 5.2. One can show (see appendix D) that in the case $d \geq 3$, $\rho \in \mathcal{B}_s$ along with quasi-orthogonality implies a certain rate of

approximation of X by $\mathcal{P}_{\Lambda_{(\rho,\eta)}}X$, as $\eta \to 0^+$:

$$\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}}X\|^2 \leq B_{s,d}|\rho|^p_{\mathcal{B}_s}\eta^{2-p} \leq B_{s,d}|\rho|^2_{\mathcal{B}_s}\left(\sum_{j\geq j_{\min}} 2^{-2j}\#_j\mathcal{T}_{(\rho,\eta)}\right)^{-\frac{2s}{d-2}}, \tag{9}$$

where $s = \frac{(d-2)(2-p)}{2p}$ and $B_{s,d} := B_0 2^p/(1-2^{p-2})$.

The main result of this paper is the following performance analysis of empirical Adaptive GMRA (see the proof in Section 4.3).

Theorem 8 *Suppose ρ satisfies quasi-orthogonality and $\mathcal{M}$ is bounded: $\mathcal{M} \subset B_M(0)$. Let $\nu > 0$. There exists $\kappa_0(\theta_2, \theta_3, \theta_4, a_{\max}, d, \nu)$ such that if $\tau_n = \kappa\sqrt{(\log n)/n}$ with $\kappa \geq \kappa_0$, the following holds:*

(i) if $d \geq 3$ and $\rho \in \mathcal{B}_s$ for some $s > 0$, there are c_1 and c_2 such that

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}}X\| \geq c_1\left(\frac{\log n}{n}\right)^{\frac{s}{2s+d-2}}\right\} \leq c_2 n^{-\nu}. \tag{10}$$

(ii) if $d = 1$, there exist c_1 and c_2 such that

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}}X\| \geq c_1\left(\frac{\log n}{n}\right)^{\frac{1}{2}}\right\} \leq c_2 n^{-\nu}. \tag{11}$$

(iii) if $d = 2$ and

$$|\rho| := \sup_{\mathcal{T}}\sup_{\eta>0}\frac{1}{\log\frac{1}{\eta}}\sum_{j\geq j_{\min}} 2^{-2j}\#_j\mathcal{T}_{(\rho,\eta)} < +\infty,$$

then there exist c_1 and c_2 such that

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}}X\| \geq c_1\left(\frac{\log^2 n}{n}\right)^{\frac{1}{2}}\right\} \leq c_2 n^{-\nu}. \tag{12}$$

Notice that by choosing ν large enough, we have

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}}X\| \geq c_1\left(\frac{\log^\alpha n}{n}\right)^{\beta}\right\} \leq c_2 n^{-\nu} \Rightarrow \text{MSE} \leq c_1\left(\frac{\log^\alpha n}{n}\right)^{2\beta},$$

so we also have $\text{MSE} \lesssim (\log n/n)^{\frac{2s}{2s+d-2}}$ for $d \geq 3$ and $\text{MSE} \lesssim \log^d n/n$ for $d = 1, 2$.

The dependencies of the constants in Theorem 8 on the geometric constants are as follows:

$$\begin{aligned}
d \geq 3: &\quad c_1 = c_1(\theta_{2,3,4}, a_{\max}, d, s, \kappa, |\rho|_{\mathcal{B}_s}, B_0, \nu), &&c_2 = c_2(\theta_{2,3,4}, a_{\min}, a_{\max}, d, s, \kappa, |\rho|_{\mathcal{B}_s}, B_0).\\
d = 2: &\quad c_1 = c_1(\theta_{2,3,4}, a_{\max}, d, \kappa, |\rho|_{\mathcal{B}_s}, B_0, \nu), &&c_2 = c_2(\theta_{2,3,4}, a_{\min}, a_{\max}, d, \kappa, |\rho|_{\mathcal{B}_s}, B_0).\\
d = 1: &\quad c_1 = c_1(\theta_{1,2,3,4}, a_{\max}, d, \kappa, B_0, \nu), &&c_2 = c_2(\theta_{1,2,3,4}, a_{\min}, a_{\max}, d, \kappa, B_0).
\end{aligned}$$

Theorem 8 is more satisfactory than Theorem 4 for two reasons: (i) when $d \geq 3$, the same rate $(\log n/n)^{2s/(2s+d-2)}$ is proved for the model class $\mathcal{B}_s$ which is larger than $\mathcal{A}_s^\infty$; (ii) the threshold-based estimator is adaptive: it does not require a priori knowledge of the regularity s, since the choice of κ is independent of s, yet it achieves the rate as if it knew the optimal regularity parameter s.

From the perspective of dictionary learning, when $d \geq 3$, Adaptive GMRA provides a dictionary $\Phi_{\widehat{\Lambda}_{\tau_n}}$ associated with a tree of weighted complexity $(n/\log n)^{d-2/(2s+d-2)}$, so that every x sampled from ρ may be encoded by a vector with $d+1$ nonzero entries, among which one encodes the location of x in the adaptive partition and the other d entries are the local principal coefficients of x.

For a given accuracy ε, in order to achieve MSE $\lesssim \varepsilon^2$, the number of samples we need is $n_\varepsilon \gtrsim (1/\varepsilon)^{(2s+d-2)/s}\log(1/\varepsilon)$. When s is unknown, we can determine s as follows: we fix a small n_0 and run Adaptive GMRA with $n_0, 2n_0, 4n_0, \ldots, Cn_0$ samples. For each sample size, we evenly split data into a training set to construct Adaptive GMRA and a test set to evaluate the MSE. According to Theorem 8, the MSE scales like $[(\log n)/n]^{\frac{2s}{2s+d-2}}$ where n is the sample size. Therefore, the slope in the log-log plot of the MSE versus n gives an approximation of $-2s/(2s+d-2)$.

The threshold τ_n in our adaptive algorithm is independent of s since κ_0 does not depend on s, which means our adaptive algorithm does not require s as a priori information but rather will learn it from data.

Remark 9 *It would also be natural to consider another stopping criterion: $\mathcal{E}_{j,k}^2 := \frac{1}{\rho(C_{j,k})}\int_{C_{j,k}} \|\mathcal{P}_j x - x\|^2 d\rho \leq \eta^2$ which suggests stopping refinement to finer scales if the approximation error is below certain threshold. The reason why we do not adopt this stopping criterion is that in this case the threshold η would have to depend on s in order to guarantee the (adaptive) rate* MSE $\lesssim (\log n/n)^{2s/(2s+d-2)}$ *for $d \geq 3$. More precisely, for any threshold $\eta > 0$, let $\mathcal{T}_{(\rho,\eta)}^{\mathcal{E}}$ be the smallest proper subtree of $\mathcal{T}$ whose leaves satisfy $\mathcal{E}_{j,k}^2 \leq \eta^2$. The corresponding adaptive partition $\Lambda_{(\rho,\eta)}^{\mathcal{E}}$ consists of the leaves of $\mathcal{T}_{(\rho,\eta)}^{\mathcal{E}}$. This stopping criterion guarantees $\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}^{\mathcal{E}}} X\| \leq \eta$. It is natural to define the model class $\mathcal{F}_s$ in the case $d \geq 3$ to be the set of probability measures ρ supported on $\mathcal{M}$ such that $\sup_{\mathcal{T}} \sup_{\eta>0} \eta^{(d-2)/s} \sum_{j \geq j_{\min}} 2^{-2j} \#_j \Lambda_{(\rho,\eta)}^{\mathcal{E}} < \infty$ where $\mathcal{T}$ varies over the set of multiscale tree decompositions satisfying (A1-A5). One can show that $\mathcal{A}_s^\infty \subsetneq \mathcal{F}_s$. As an analogue of Theorem 8, we can prove that, there exists $\kappa_0 > 0$ such that if our adaptive algorithm adopts the stopping criterion $\widehat{\mathcal{E}}_{j,k} \leq \tau_n^{\mathcal{E}}$ where the threshold is chosen as $\tau_n^{\mathcal{E}} = \kappa(\log n/n)^{\frac{s}{2s+d-2}}$ with $\kappa \geq \kappa_0$, then the empirical approximation on the adaptive partition $\widehat{\Lambda}_{\tau_n^{\mathcal{E}}}$ satisfies* MSE $= \|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n^{\mathcal{E}}}} X\|^2 \lesssim (\log n/n)^{2s/(2s+d-2)}$. *With this stopping criterion, the threshold $\tau_n^{\mathcal{E}}$ would require knowing s, unlike in Theorem 8.*

Theorem 8 is stated when $\mathcal{M}$ is bounded. The assumption of the boundedness of $\mathcal{M}$ is largely irrelevant, and may be replaced by a weaker assumption on the decay of ρ.

Theorem 10 *Let $d \geq 3$, $s, \delta, \lambda, \mu > 0$. Assume that there exists C_1 such that*

$$\int_{B_R(0)^c} ||x||^2 d\rho \leq C_1 R^{-\delta}, \ \forall R \geq R_0.$$

Suppose ρ satisfies quasi-orthogonality. If ρ restricted on $B_R(0)$, denoted by $\rho_{|B_R(0)}$, is in $\mathcal{B}_s$ for every $R \geq R_0$ and $(|\rho_{|B_R(0)}|_{\mathcal{B}_s})^p \leq C_2 R^\lambda$ for some $C_2 > 0$, where $p = \frac{2(d-2)}{2s+d-2}$. Then there exists $\kappa_0(\theta_2, \theta_3, \theta_4, a_{\max}, d, \nu)$ such that if $\tau_n = \kappa\sqrt{\log n/n}$ with $\kappa \geq \kappa_0$, we have

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}} X\| \geq c_1 \left(\frac{\log n}{n}\right)^{\frac{s}{2s+d-2}\frac{\delta}{\delta+\max(\lambda,2)}}\right\} \leq c_2 n^{-\nu} \tag{13}$$

for some c_1, c_2 independent of n, where the estimator $\widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}} X$ is obtained by Adaptive GMRA within $B_{R_n}(0)$ where $R_n = \max(R_0, \mu(n/\log n)^{\frac{2s}{(2s+d-2)(\delta+\max(\lambda,2))}})$, and is equal to 0 for the points outside $B_{R_n}(0)$.

Theorem 10 is proved at the end of Section 4.3. It implies MSE $\lesssim (\log n/n)^{\frac{2s}{2s+d-2}\cdot\frac{\delta}{\delta+\max(\lambda,2)}}$. As δ increases, i.e., $\delta \to +\infty$, the MSE approaches $(\log n/n)^{\frac{2s}{2s+d-2}}$, which is consistent with Theorem 8 for bounded $\mathcal{M}$. Similar results, with similar proofs, would hold under different assumptions on the decay of ρ; for example for ρ decaying at least exponentially, only additional $\log n$ terms in the rate would be lost compared in Theorem 8.

Remark 11 *We claim that λ is not large in simple cases. For example, if $\rho \in \mathcal{A}_s^\infty$ and ρ decays in the radial direction in such a way that $\rho(C_{j,k}) \leq C2^{-jd}\|c_{j,k}\|^{-(d+1+\delta)}$, it is easy to show that $\rho_{|B_R(0)} \in \mathcal{B}_s$ for all $R > 0$ and $|\rho_{|B_R(0)}|_{\mathcal{B}_s}^p \leq R^\lambda$ with $\lambda = d - \frac{(d+1+\delta)(d-2)}{2s+d-2}$ (see the end of Section 4.3).*

Remark 12 *Suppose that ρ was supported in a tube of radius σ around a d-dimensional manifold $\mathcal{M}$, a model that can account both for (bounded) noise and situations where data is not exactly on a manifold, but close to it, as in Maggioni et al. (2016). Then Theorem 8 and Theorem 10 apply in this case, provided one stops the estimator at a scale j such that $2^{-j} \gtrsim \sigma$.*

Remark 13 *In these Theorems we are assuming that d is given because it can be estimated using existing techniques, see Little et al. (2017) and many references therein.*

2.5. Connection to previous works

The works by Allard et al. (2012) and Maggioni et al. (2016) are natural predecessors to this work. In Allard et al. (2012), GMRA and orthogonal GMRA were proposed as data-driven dictionary learning tools to analyze intrinsically low-dimensional point clouds in a high dimensional space. The bias $\|X - \mathcal{P}_j X\|$ were estimated for volume measures on $\mathcal{C}^s, s \in (1,2]$ manifolds . The performance of GMRA, including sparsity guarantees and computational costs, were systematically studied and tested on both simulated and real data. In Maggioni et al. (2016) the finite sample behavior of empirical GMRA was studied. A non-asymptotic probabilistic bound on the approximation error $\|X - \widehat{\mathcal{P}}_j X\|$ for the model class $\mathcal{A}_2$ (a special case of Theorem 4 with $s = 2$) was established. It was further proved that if the measure ρ is absolutely continuous with respect to the volume measure on a tube of a bounded $\mathcal{C}^2$ manifold with a finite reach, then ρ is in $\mathcal{A}_2$. Running the cover

tree algorithm on data gives rise to a family of multiscale partitions satisfying Assumption (A3-A5). The analysis in Maggioni et al. (2016) robustly accounts for noise and modeling errors as the probability measure is concentrated "near" a manifold. This work extends GMRA by introducing Adaptive GMRA, where low-dimensional linear approximations of $\mathcal{M}$ are built on adaptive partitions at different scales. The finite sample performance of Adaptive GMRA is proved for a large model class. Adaptive GMRA takes full advantage of the multiscale structure of GMRA in order to model data sets of varying complexity across locations and scales. We also generalize the finite sample analysis of empirical GMRA from $\mathcal{A}_2$ to $\mathcal{A}_s$, and analyze the finite sample behavior of orthogonal GMRA and adaptive orthogonal GMRA.

In a different direction, a popular learning algorithm for fitting low-dimensional planes to data is k-flats: let $\mathcal{F}_k$ be the collections of k flats (affine spaces) of dimension d. Given data $\mathcal{X}_n = \{x_1, \ldots, x_n\}$, k-flats solves the optimization problem

$$\min_{S \in \mathcal{F}_k} \frac{1}{n} \sum_{i=1}^{n} \mathrm{dist}^2(x_i, S) \tag{14}$$

where $\mathrm{dist}(x, S) = \inf_{y \in S} \|x - y\|$. Even though a global minimizer of (14) exists, it is hard to attain due to the non-convexity of the model class $\mathcal{F}_k$, and practitioners are aware that many local minima that are significantly worse than the global minimum exist. While often k is considered given, it may be in fact chosen from the data: for example Theorem 4 in Canas et al. (2012) implies that, given n samples from a probability measure that is absolutely continuous with respect to the volume measure on a smooth d-dimensional manifold $\mathcal{M}$, the expected (out-of-sample) L^2 approximation error of $\mathcal{M}$ by $k_n = C_1(\mathcal{M}, \rho) n^{\frac{d}{2(d+4)}}$ planes is of order $\mathcal{O}(n^{-\frac{2}{d+4}})$. This result is comparable with our Theorem 4 in the case $s = 2$ which says that the L^2 error by empirical GMRA at the scale j such that $2^j \asymp (n/\log n)^{\frac{1}{d+2}}$ achieves a faster rate $\mathcal{O}(n^{-\frac{2}{d+2}})$. So we not only achieve a better rate, but we do so with provable and fast algorithms, that are nonlinear but do not require non-convex optimization.

Multiscale adaptive estimation has been an intensive research area for decades. In the pioneering works by Donoho and Johnstone (see Donoho and Johnstone, 1994, 1995), soft thresholding of wavelet coefficients was proposed as a spatially adaptive method to denoise a function. In machine learning, Binev et al. addressed the regression problem with piecewise constant approximations (see Binev et al., 2005) and piecewise polynomial approximations (see Binev et al., 2007) supported on an adaptive subpartition chosen as the union of data-*independent* cells (e.g. dyadic cubes or recursively split samples). While the works above are in the context of function approximation/learning/denoising, a whole branch of geometric measure theory (following the seminal work by Jones (1990); David and Semmes (1993)) quantifies via multiscale least squares fits the rectifiability of sets and their approximability by multiple images of bi-Lipschitz maps of, say, a d-dimensional square. We can the view the current work as extending those ideas to the setting where data is random, possibly noisy, and guarantees on error on future data become one of the fundamental questions.

Theorem 8 can be viewed as a geometric counterpart of the adaptive function approximation in Binev et al. (2005, 2007). Our results are a "geometric counterpart" of sorts. We would like to point out two main differences between Theorem 8 and Theorem 3 in Binev et al. (2005): (i) In Binev et al. (2005, Theorem 3), there is an extra assumption that

the function is in $\mathcal{A}_\gamma$ with γ arbitrarily small. This assumption takes care of the error at the nodes in $\mathcal{T} \setminus \mathcal{T}^n$ where the thresholding criteria would succeed: these nodes should be added to the adaptive partition but have not been explored by our data. This assumption is removed in our Theorem 8 by observing that the nodes below the data master tree have small measure so their refinement criterion is smaller than $2^{-j}\tau_n$ with high probability. (ii) we consider scale-dependent thresholding criterion $\widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n$ unlike the criterion in Binev et al. (2005, 2007) that is scale-independent. This difference arises because at scale j our linear approximation is built on data within a ball of radius $\lesssim 2^{-j}$ and so the variance of PCA on a fixed cell at scale j is proportional to 2^{-2j}. For the same reason, we measure the complexity of $\mathcal{T}_{(\rho,\eta)}$ in terms of the weighted tree complexity instead of the cardinality since the former one gives an upper bound of the variance in piecewise linear approximation on partition via PCA (see Lemma 15). Using scale-dependent threshold and measuring tree complexity in this way give rise to the best rate of convergence. In contrast, if we use scale-independent threshold and define a model class Γ_s for whose elements $\#\mathcal{T}_{(\rho,\eta)} = \mathcal{O}(\eta^{-\frac{2d}{2s+d}})$ (analogous to the function class in Binev et al. (2005, 2007)), we can still show that $\mathcal{A}_s^\infty \subset \Gamma_s$, but the estimator only achieves MSE $\lesssim ((\log n)/n)^{\frac{2s}{2s+d}}$. However many elements[2] of Γ_s not in $\mathcal{A}_s^\infty$ are in $\mathcal{B}^{s'}$ with $\frac{2(d-2)}{2s'+d-2} = \frac{2d}{2s+d}$, and in Theorem 8 the estimator based on scaled thresholding achieves a better rate, which we believe is optimal.

We refer the reader to Maggioni et al. (2016) for a thorough discussion of further related work related to manifold and dictionary learning.

2.6. Construction of a multiscale tree decomposition

Our multiscale tree decomposition is constructed from a variation of the cover tree algorithm (see Beygelzimer et al., 2006) applied on half of the data denoted by $\mathcal{X}_n'$. In brief the cover tree $T(\mathcal{X}_n')$ on $\mathcal{X}_n'$ is a leveled tree where each level is a "cover" for the level beneath it. Each level is indexed by j and each node in $T(\mathcal{X}_n')$ is associated with a point in $\mathcal{X}_n'$. A point can be associated with multiple nodes in the tree but it can appear at most once at every level. Let $T_j(\mathcal{X}_n') \subset \mathcal{X}_n'$ be the set of nodes of T at level j. The cover tree obeys the following properties for all $j = j_{\min}, \ldots, j_{\max}$:

1. Nesting: $T_j(\mathcal{X}_n') \subset T_{j+1}(\mathcal{X}_n')$;
2. Separation: for all distinct $p, q \in T_j(\mathcal{X}_n')$, $\|p - q\| > 2^{-j}$;
3. Covering: for all $q \in T_{j+1}(\mathcal{X}_n')$, there is $p \in T_j(\mathcal{X}_n')$ such that $\|p - q\| < 2^{-j}$. The node at level j associated with p is a parent of the node at level $j+1$ associated with q.

In the third property, q is called a child of p. Each node can potentially have multiple parents satisfying the distance constraint in 3. above, but is only assigned to one of them in the tree. The properties above imply that for any $q \in \mathcal{X}_n'$, there exists $p \in T_j$ such that $\|p - q\| < 2^{-j+1}$. The authors in Beygelzimer et al. (2006) showed that cover tree always exists and that can be constructed in time $\mathcal{O}(C^d D n \log n)$.

2. For these elements, the average cell-wise refinement is monotone in the sense that for every $C_{j,k}$ and $C_{j+1,k'} \subset C_{j,k}$, we have $\Delta_{j+1,k'}/\sqrt{\rho(C_{j+1,k'})} \leq \Delta_{j,k}/\sqrt{\rho(C_{j,k})}$.

We now show that from a set of nets $\{T_j(\mathcal{X}_n')\}_{j=j_{\min},\dots,j_{\max}}$ as above we can construct a set of $C_{j,k}$'s with desired properties. (see Appendix A for the construction of $C_{j,k}$'s and the proof of Proposition 14). $\widetilde{\mathcal{M}}$ defined in (31) is equal to the union of the $C_{j,k}$'s up to a set with 0 empirical measure.

Proposition 14 *Assume ρ is a doubling probability measure on $\mathcal{M}$ with doubling constant C_1. Then $\{C_{j,k}\}_{k\in\mathcal{K}_j, j_{\min}\le j\le j_{\max}}$ constructed in Appendix A satisfies the Assumptions*

1. *(A1) with $a_{\max} \le C_1^2(24)^d$ and $a_{\min} = 1$.*
2. *For any $\nu > 0$,*
$$\mathbb{P}\left\{\rho(\mathcal{M}\setminus\widetilde{\mathcal{M}}) > \frac{28\nu\log n}{3n}\right\} \le 2n^{-\nu}; \tag{15}$$
3. *(A3) with $\theta_1 = C_1^{-1}4^{-d}$;*
4. *(A4) with $\theta_2 = 3$.*
5. *Additionally:*
   - *5a. if ρ satisfies the conditions in (A5) with $B_r(z)$, $z\in\mathcal{M}$, replacing $C_{j,k}$ with constants $\tilde\theta_3, \tilde\theta_4$ such that $\lambda_d(\mathrm{Cov}(\rho_{|_{B_r(z)}})) \ge \tilde\theta_3 r^2/d$ and $\lambda_{d+1}(\mathrm{Cov}(\rho_{|_{B_r(z)}})) \le \tilde\theta_4\lambda_d(\mathrm{Cov}(\rho_{|_{B_r(z)}}))$, then the conditions in (A5) are satisfied by the $C_{j,k}$'s we construct with $\theta_3 := \tilde\theta_3(4C_1)^{-2}12^{-d}$ and $\theta_4 := \tilde\theta_4/\tilde\theta_3 12^{2d+2}C_1^4$.*
   - *5b. if ρ is the volume measure on a closed $\mathcal{C}^s$ manifold isometrically embedded in $\mathbb{R}^D$, then the conditions in (A5) are satisfied by the $C_{j,k}$'s when j is sufficiently large.*

Even though the $\{C_{j,k}\}$ does not exactly satisfy Assumption (A2), we claim that (15) is sufficient for our performance guarantees in the case that $\mathcal{M}$ is bounded by M and $d\ge 3$, since simply approximating points on $\mathcal{M}\setminus\widetilde{\mathcal{M}}$ by 0 gives the error:

$$\mathbb{P}\left\{\int_{\mathcal{M}\setminus\widetilde{\mathcal{M}}}\|x\|^2 d\rho \ge \frac{28M^2\log n}{3n}\right\} \le 2n^{-\nu}. \tag{16}$$

The constants in Proposition 14 are extremely pessimistic, due to the generality of the assumptions on the space $\mathcal{M}$. Indeed when $\mathcal{M}$ is a nice manifold as in case (5b), the statement in the Proposition says that the constants for the $C_{j,k}$'s we construct are similar to those of the ideal $C_{j,k}$'s. In practice we use a much simpler and more efficient tree construction method and we experimentally obtain the properties above with $a_{\max} = C_1^2 4^d$ and $a_{\min} = 1$, at least for the vast majority of the points, and $\theta_{\{3,4\}} \approxeq \tilde\theta_{\{3,4\}}$. We describe this simpler construction for the multiscale partitions in Appendix A.3, together with experiments suggesting that at least in relatively simple cases one may expect $\theta_{\{3,4\}} \approxeq \tilde\theta_{\{3,4\}}$.

Besides cover trees, there are other methods that can be used in practice for the multiscale partition, such as METIS by Karypis and Kumar (1999) that is used in the numerical examples in Chen and Maggioni (2010) and Allard et al. (2012), iterated PCA (see some analysis in Szlam (2009)) or iterated k-means. These can be computationally more efficient than cover trees, with the downside being that they may lead to partitions not satisfying our usual assumptions (in theory, and perhaps in practice).

3. Numerical experiments

We conduct numerical experiments on both synthetic and real data to demonstrate the performance of our algorithms. Given $\{x_i\}_{i=1}^n$, we split them to training data for the constructions of empirical GMRA and Adaptive GMRA and test data for the evaluation of the approximation errors:

	L^2 error	L^∞ error
Absolute error	$\left(\frac{1}{n^{\text{test}}}\sum_{x_i\in\text{test set}}\|x_i-\widehat{\mathcal{P}}x_i\|^2\right)^{\frac12}$	$\max_{x_i\in\text{test set}}\|x_i-\widehat{\mathcal{P}}x_i\|$
Relative error	$\left(\frac{1}{n^{\text{test}}}\sum_{x_i\in\text{test set}}\|x_i-\widehat{\mathcal{P}}x_i\|^2/\|x_i\|^2\right)^{\frac12}$	$\max_{x_i\in\text{test set}}\|x_i-\widehat{\mathcal{P}}x_i\|/\|x_i\|$

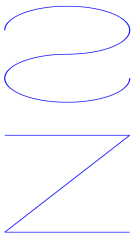
where n^{test} is the cardinality of the test set and $\widehat{\mathcal{P}}$ is the piecewise linear projection given by empirical GMRA or Adaptive GMRA. In our experiments we use absolute error for synthetic data, 3D shape and relative error for the MNIST digit data, natural image patches.

3.1. Synthetic data

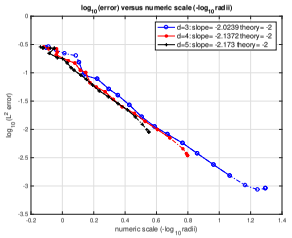
We take samples $\{x_i\}_{i=1}^n$ on the d-dim S and Z-manifolds, whose first two coordinates $x_{i,1}, x_{i,2}$ are on the S and Z curve and other coordinates $x_{i,k}\in[0,1], k=3,4,\dots,d+1$. We evenly split the samples to the training set and the test set. In the noisy case, training data are corrupted by Gaussian noise: $\tilde x_i^{\text{train}} = x_i^{\text{train}} + \frac{\sigma}{\sqrt{D}}\xi_i, i=1,\dots,\frac{n}{2}$ where $\xi_i\sim\mathcal{N}(0, I_{D\times D})$, but test data are noise-free. Test data error below the noise level implies that we are denoising the data.

3.1.1. Regularity parameter s in the $\mathcal{A}_s$ and $\mathcal{B}_s$ model

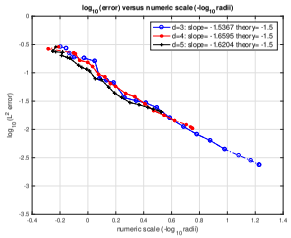
We sample 10^5 training points on the d-dim S- or Z-manifolds, for $d=3,4,5$. The measure on the S-manifold is not exactly the volume measure but is comparable with the volume measure. The log-log plot of the approximation error versus scale in Figure 5 (b) shows that volume measures on the d-dim S-manifold are in $\mathcal{A}_s$ with $s\approx 2.0, 2.1, 2.2$ when $d=3,4,5$, consistent with our theory which gives $s=2$. Figure 5 (c) shows that volume measures on the d-dim Z-manifold are in $\mathcal{A}_s$ with $s\approx 1.5, 1.7, 1.6$ when $d=3,4,5$, consistent with our theory which gives $s=1.5$. The log-log plot of the approximation error versus the weighted complexity of the adaptive partition in Figure 5 (d) and (e) gives rises to an approximation of the regularity parameter s in the $\mathcal{B}_s$ model in the table.



(a) Projection of S and Z-manifolds

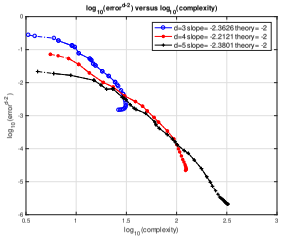


(b) S: error vs. scale

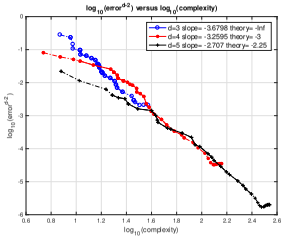


(c) Z: error vs. scale

	the S-manifold	
	theoretical s	numerical s
$d=3$	2	2.1 ± 0.2
$d=4$	2	2.1 ± 0.2
$d=5$	2	2.3 ± 0.1
	the Z-manifold	
	theoretical s	numerical s
$d=3$	$+\infty$	3.4 ± 0.3
$d=4$	3	3.1 ± 0.2
$d=5$	2.25	2.5 ± 0.2



(d) S: error vs. tree complexity

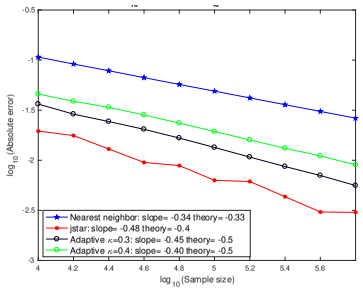


(e) Z: error vs. tree complexity

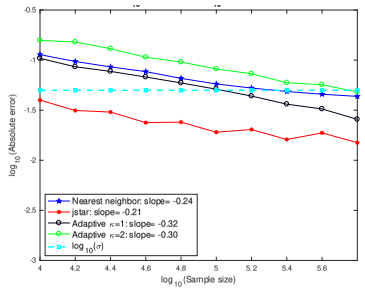
Figure 5: 10^5 training points are sampled on the d-dimensional S or Z-manifold ($d = 3, 4, 5$). In (b) and (c), we display $\log_{10} \|X - \widehat{\mathcal{P}}_j X\|_n$, versus scale j. The negative of the slope on the solid portion of the line approximates the regularity parameter s in the $\mathcal{A}_s$ model. In (d) and (e), we display the log-log plot of $\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_\eta} X\|_n^{d-2}$ versus the weighted complexity of the adaptive partition for the d-dimensional S and Z-manifold. The negative of the slope on the solid portion of the line approximates the regularity parameter s in the $\mathcal{B}_s$ model. Our five experiments give the s in the table. For the 3-dim Z-manifold, while $s = +\infty$ in the case of infinite samples, we do obtain a large s with 10^5 samples.

3.1.2. Error versus sample size n

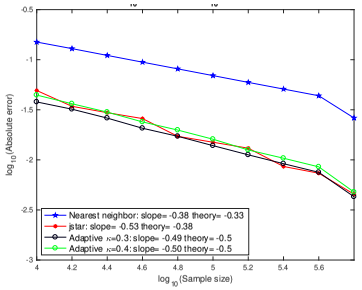
We take n samples on the 3-dim S- and Z-manifolds embedded in $\mathbb{R}^{100}$ ($d = 3, D = 100$). In Figure 6, we set the noise level $\sigma = 0$ (a,c) and $\sigma = 0.05$ (b,d), display the log-log plot of the average approximation error over 10 trails with respect to the sample size n for empirical GMRA at scale j^* which is chosen as per Theorem 4: $2^{-j^*} = [(\log n)/n]^{1/(2s+d-2)}$ with $d = 3$ and $s = 2$ for the S-manifold and $s = 1.5$ for the Z-manifold. For Adaptive GMRA, the ideal κ increases as σ increases. We let $\kappa \in \{0.3, 0.4\}$ when $\sigma = 0$ and $\kappa \in \{1, 2\}$ when $\sigma = 0.05$. We also test the Nearest Neighbor (NN) approximation. The negative of the slope, determined by least squared fit, gives rise to the rate of convergence:



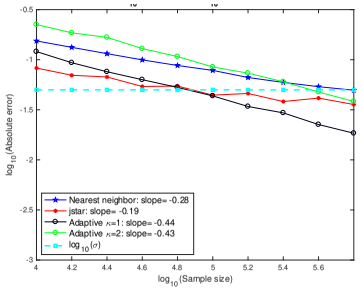
(a) the 3-dim S-manifold, $\sigma = 0$



(b) the 3-dim S-manifold, $\sigma = 0.05$



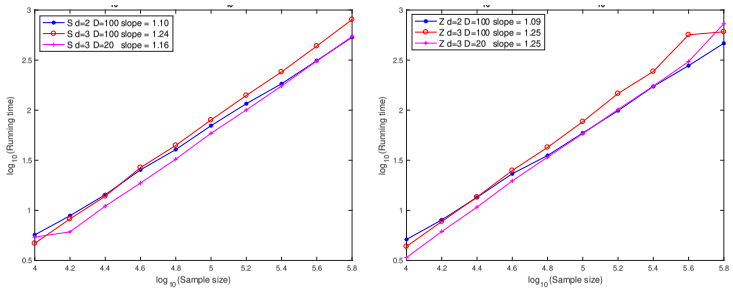
(c) the 3-dim Z-manifold, $\sigma = 0$



(d) the 3-dim Z-manifold, $\sigma = 0.05$

Figure 6: L^2 error versus the sample size n, for the 3-dim S and Z manifolds ($d = 3$), top and bottom rows respectively, of GMRA at the scale j^* chosen as per Theorem 4 (with the constant μ set, arbitrarily, equal to 1), and Adaptive GMRA with varied κ. We let $\kappa \in \{0.3, 0.4\}$ when $\sigma = 0$ (left column) and $\kappa \in \{1, 2\}$ when $\sigma = 0.05$ (right column).

L^2 error $\sim n^{\text{slope}}$. When $\sigma = 0$, the convergence rate for the nearest neighbor approximation should be $1/d = 1/3$. GMRA gives rise to a smaller error and a faster rate of convergence than the nearest neighbor approximation. When $\sigma = 0.05$, Adaptive GMRA yields a faster rate of convergence than GMRA, especially for the Z manifold. We note a de-noising effect when the approximation error falls below σ as n increases. In Adaptive GMRA, different values of κ do yield different errors up to a constant, but the rate of convergence is almost independent of κ, as predicted by Theorem 8.



(a) the S-manifold, $\sigma = 0$ (b) the Z-manifold, $\sigma = 0$

Figure 7: Average running time of GMRA in 10 experiments versus the sample size n for the S (a) and Z (b) manifolds. We set $d = 2, D = 100$ and $d = 3, D = 100$ and $d = 3, D = 20$ respectively.

3.1.3. Running time versus sample size n

The complexity of GMRA is $O(C^d Dn \log n)$. In Figure 7, we display the average running time of GMRA in 10 experiments for the S and Z manifolds when $d = 2, D = 100$ and $d = 3, D = 100$ and $d = 3, D = 20$. The running time of GMRA is almost linear in n. The running time increases as d and D increase since the complexity of GMRA is exponential in d and linear in D.

3.1.4. Robustness of GMRA and Adaptive GMRA

The robustness of the empirical GMRA and Adaptive GMRA is tested on the 3-dim S and Z-manifolds embedded in $\mathbb{R}^{100}$ while σ varies but n is fixed to be 10^5. Figure 8 shows that the average L^2 approximation error in 10 trails increases linearly with respect to σ for both uniform and Adaptive GMRA with $\kappa \in \{0.01, 0.05, 0.5\}$.

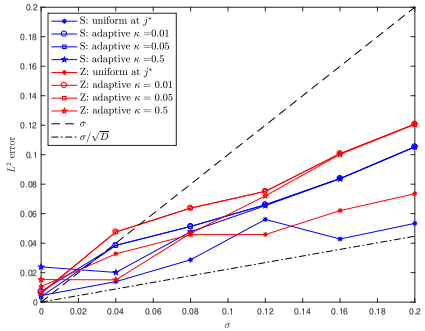
3.2. 3D shapes

We run GMRA and Adaptive GMRA on 3D points clouds on the teapot, armadillo and dragon in Figure 9. The teapot data are from the matlab toolbox and others are from the Stanford 3D Scanning Repository `http://graphics.stanford.edu/data/3Dscanrep/`.

Figure 9 shows that the adaptive partitions chosen by Adaptive GMRA matches our expectation that, at irregular locations, cells are selected at finer scales than at "flat" locations.

In Figure 10, we display the absolute L^2/L^∞ approximation error on test data versus scale and partition size. The left column shows the L^2 approximation error versus scale for GMRA and the center approximation. While the GMRA approximation is piecewise

Figure 8: The average L^2 approximation error in 10 trails versus σ for GMRA and Adaptive GMRA with $\kappa \in \{0.01, 0.05, 0.5\}$ on data sampled on the 3-dim S and Z-manifolds. This shows the error of approximation grows linearly with the noise size, suggesting robustness in the construction.



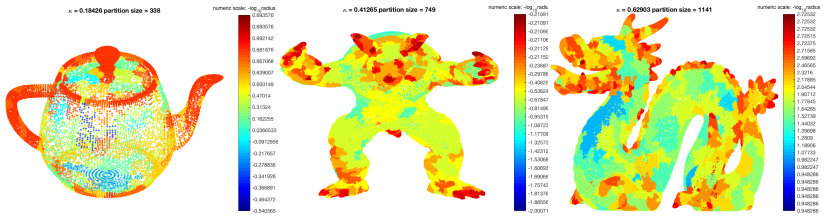
linear, the center approximation is piecewise constant. Both approximation errors decay from coarse to fine scales, but GMRA yields a smaller error than the approximation by local centers. In the middle column, we run GMRA and Adaptive GMRA with the L^2 refinement criterion defined in Table 2 with scale-dependent ($\Delta_{j,k} \geq 2^{-j}\tau_n$) and scale-independent ($\Delta_{j,k} \geq \tau_n$) threshold respectively, and display the log-log plot of the L^2 approximation error versus the partition size. Overall Adaptive GMRA yields the same L^2 approximation error as GMRA with a smaller partition size, but the difference is insignificant in the armadillo and dragon, as these 3D shapes are complicated and the L^2 error simply averages the error at all locations. Then we implement Adaptive GMRA with the L^∞ refinement criterion: $\widehat{\Delta}^\infty_{j,k} = \max_{x_i \in C_{j,k}} \|\widehat{\mathcal{P}}_{j+1}x_i - \widehat{\mathcal{P}}_j x_i\|$ and display the log-log plot of the L^∞ approximation error versus the partition size in the right column. In the L^∞ error, Adaptive GMRA saves a considerable number (about half) of cells in order to achieve the same approximation error as GMRA. In this experiment, scale-independent threshold is slightly better than scale-dependent threshold in terms of saving the partition size.

3.3. MNIST digit data

We consider the MNIST data set from `http://yann.lecun.com/exdb/mnist/`, which contains images of $60,000$ handwritten digits, each of size 28×28, grayscale. The intrinsic dimension of this data set varies for different digits and across scales, as it was observed in Little et al. (2017). We run GMRA by setting the diameter of cells at scale j to be $\mathcal{O}(0.9^j)$ in order to slowly zoom into the data at multiple scales.

We evenly split the digits to the training set and the test set. As the intrinsic dimension is not well-defined, we set GMRA to pick the dimension of $\widehat{V}_{j,k}$ adaptively, as the smallest dimension needed to capture 50% of the energy of the data in $C_{j,k}$. As an example, we display the GMRA approximations of the digit $0, 1, 2$ from coarse scales to fine scales in Figure 11. The histogram of the dimensions of the subspaces $\widehat{V}_{j,k}$ is displayed in (a). (b) represents $\log_{10} \|\widehat{\mathcal{P}}_{j+1}x_i - \widehat{\mathcal{P}}_j x_i\|$ from the coarsest scale (top) to the finest scale (bottom), with columns indexed by the digits, sorted from 0 to 9. We observe that 1 has more fine

(a) 41,472 points (b) 165,954 points (c) 437,645 points



(d) $\kappa \approx 0.18$, partition size $= 338$ (e) $\kappa \approx 0.41$, partition size $= 749$ (f) $\kappa \approx 0.63$, partition size $= 1141$

Figure 9: Top line: 3D shapes; bottom line: adaptive partitions selected with refinement criterion $\widehat{\Delta}_{j,k} \geq 2^{-j}\kappa\sqrt{(\log n)/n}$. Every cell is colored by scale. In the adaptive partition, at irregular locations cells are selected at finer scales than at "flat" locations.

scale information than the other digits. In (c), we display the log-log plot of the relative L^2 error versus scale in GMRA and the center approximation. The improvement of GMRA over center approximation is noticeable. Then we compute the relative L^2 error for GMRA and Adaptive GMRA when the partition size varies. Figure 11 (d) shows that Adaptive GMRA achieves the same accuracy as GMRA with fewer cells in the partition. Errors increase when the partition size exceeds 10^3 due to a large variance at fine scales. In this experiment, scale-dependent threshold and scale-independent threshold yield similar performances.

3.4. Natural image patches

It was argued in Peyré (2009) that many sets of patches extracted from natural images can be modeled a low-dimensional manifold. We use the Caltech 101 dataset from `https://www.vision.caltech.edu/Image_Datasets/Caltech101/` (see F. Li and Perona, 2006), take 40 images from four categories: accordion, airplanes, hedgehog and scissors and extract multiscale patches of size 8×8 from these images. Specifically, if the image is of size $m \times m$,

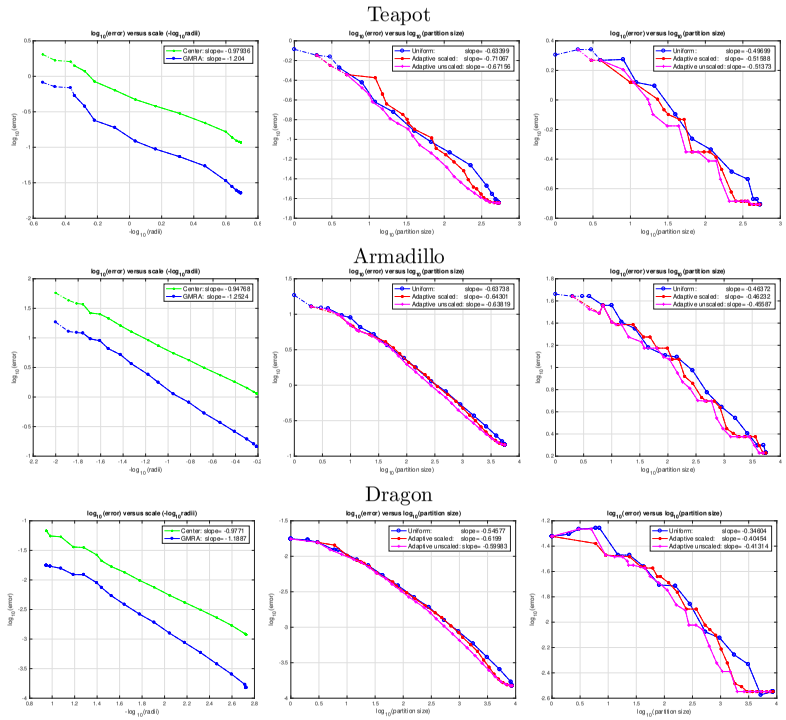
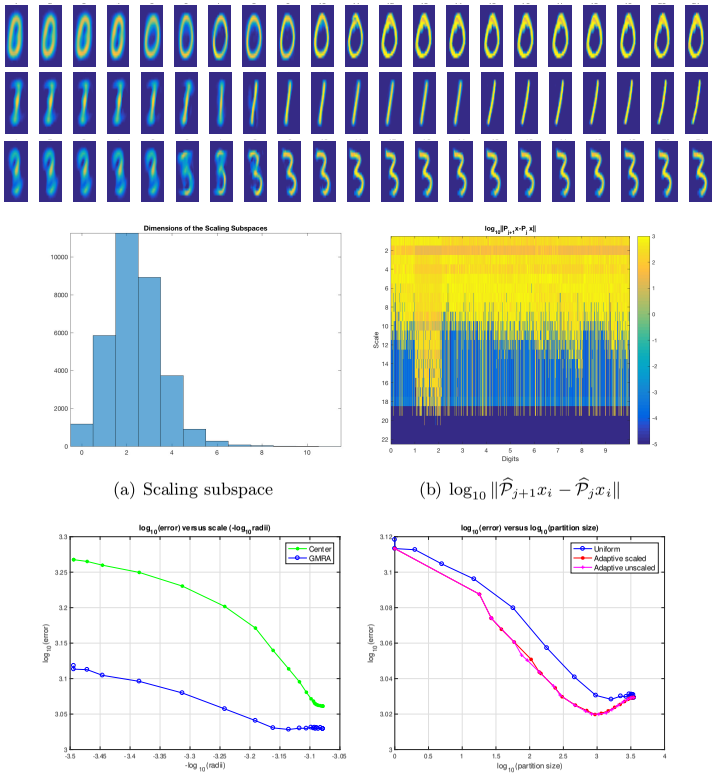


Figure 10: Left column: $\log_{10}(L^2$ error) versus scale for GMRA and center approximation; Middle column: log-log plot of the L^2 error versus partition size for GMRA and Adaptive GMRA with scale-dependent and scale-independent threshold under the L^2 refinement defined in Table 2; Right column: log-log plot of L^∞ error versus partition size for GMRA and Adaptive GMRA with scale-dependent and scale-independent threshold under the L^∞ refinement.

for $\ell = 1, \ldots, \log_2(m/8)$, we collect patches of size $2^\ell 8$, low-pass filter them and downsample them to become patches of size 8×8 (see Gerber and Maggioni (2013) for a discussion about dictionary learning on patches of multiple sizes using multiscale ideas). Then we randomly pick $200,000$ patches, evenly split them to the training set and the test set. In the construction of GMRA, we set the diameter of cells at scale j to be $\mathcal{O}(0.9^j)$ and the dimension of $\widehat{V}_{j,k}$ to be the smallest dimension needed to capture 50% of the energy of the data in $C_{j,k}$. We also run GMRA and Adaptive GMRA on the Fourier magnitudes of these image patches to take advantage of translation-invariance of the Fourier magnitudes. The



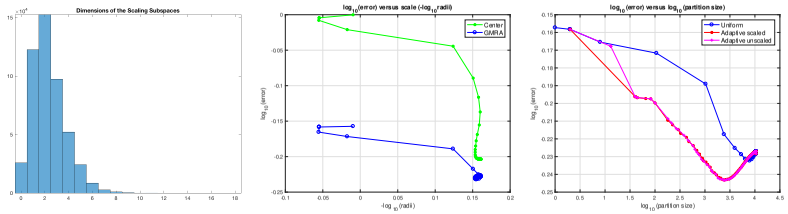
(a) Scaling subspace (b) $\log_{10} \|\widehat{\mathcal{P}}_{j+1}x_i - \widehat{\mathcal{P}}_j x_i\|$

(c) Error versus scale (d) Error versus partition size

Figure 11: The top three rows: multiscale approximations of the digit $0, 1, 2$ in the MNIST data set, from the coarsest scale (left) to the finest scale (right). (a) the histogram of dimensions of the subspaces $\widehat{V}_{j,k}$; (b) $\log_{10} \|\widehat{\mathcal{P}}_{j+1}x_i - \widehat{\mathcal{P}}_j x_i\|$ from the coarsest scale (top) to the finest scale (bottom), with columns indexed by the digits, sorted from 0 to 9; (c) log-log plot of the relative L^2 error versus scale in GMRA and the center approximation; (d) log-log plot of the relative L^2 error versus partition size for GMRA, Adaptive GMRA with scale-dependent and scale-independent threshold.

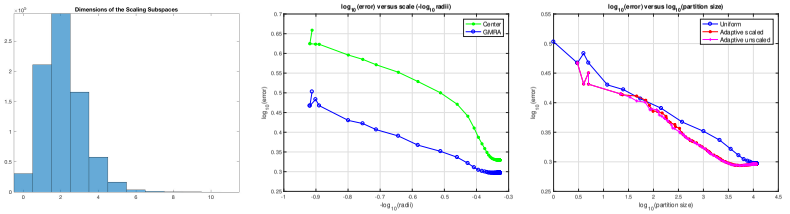
results are shown in Figure 13. The histograms of the dimensions of the subspaces $\widehat{V}_{j,k}$ are displayed in (a,d). Figure 13 (b) and (e) show the relative L^2 error versus scale for GMRA and the center approximation. We then compute the relative L^2 error for GMRA

Learning image patches



(a) Scaling subspace (b) Error versus scale (c) Error versus partition size

Learning the Fourier magnitudes of image patches



(d) Scaling subspace (e) Error versus scale (f) Error versus partition size

Figure 12: Caltech 101 image patches

Figure 13: Top line: learning on $200,000$ image patches; bottom line: results of learning the Fourier magnitudes of the same image patches. (a,d) histograms of the dimensions of the subspaces $\widehat{V}_{j,k}$; (b,e) relative L^2 error versus scale for GMRA and the center approximation; (c,f) relative L^2 error versus the partition size for GMRA, Adaptive GMRA with scale-dependent and scale-independent threshold.

and Adaptive GMRA when the partition size varies and display the log-log plot in (c) and (f). It is noticeable that Adaptive GMRA achieves the same accuracy as GMRA with a smaller partition size. We conducted similar experiments on $200,000$ multiscale patches from CIFAR 10 from `https://www.cs.toronto.edu/~kriz/cifar.html` (see Krizhevsky and Hinton, 2009) with extremely similar results (not shown).

4. Performance analysis of GMRA and Adaptive GMRA

This section is devoted to the performance analysis of empirical GMRA and Adaptive GMRA. We will start with the following stochastic error estimate on any partition.

4.1. Stochastic error on a fixed partition

Suppose $\tilde{\mathcal{T}}$ is a finite proper subtree of the data master tree $\mathcal{T}^n$. Let Λ be the partition consisting the outer leaves of $\tilde{\mathcal{T}}$. The piecewise affine projector on Λ and its empirical version are

$$\mathcal{P}_\Lambda = \sum_{C_{j,k}\in\Lambda} \mathcal{P}_{j,k}\mathbf{1}_{j,k} \quad \text{and} \quad \widehat{\mathcal{P}}_\Lambda = \sum_{C_{j,k}\in\Lambda} \widehat{\mathcal{P}}_{j,k}\mathbf{1}_{j,k}.$$

A non-asymptotic concentration bound on the stochastic error $\|\mathcal{P}_\Lambda X - \widehat{\mathcal{P}}_\Lambda X\|$ is given by:

Lemma 15 *Let Λ be the partition associated a finite proper subtree $\tilde{\mathcal{T}}$ of the data master tree $\mathcal{T}^n$. Suppose Λ contains $\#_j\Lambda$ cells at scale j. Then for any $\eta > 0$,*

$$\mathbb{P}\{\|\mathcal{P}_\Lambda X - \widehat{\mathcal{P}}_\Lambda X\| \geq \eta\} \leq \alpha d \cdot \#\Lambda \cdot e^{-\frac{\beta n \eta^2}{d^2 \sum_j 2^{-2j}\#_j\Lambda}} \tag{17}$$

$$\mathbb{E}\|\mathcal{P}_\Lambda X - \widehat{\mathcal{P}}_\Lambda X\|^2 \leq \frac{d^2 \log(\alpha d \#\Lambda) \sum_j 2^{-2j}\#_j\Lambda}{\beta n}$$

where $\alpha = \alpha(\theta_2, \theta_3)$ and $\beta = \beta(\theta_2, \theta_3, \theta_4)$.

Lemma 15 and Proposition 16 below are proved in appendix C .

4.2. Performance analysis of empirical GMRA on uniform partitions

According to Eq. (1), the approximation error of empirical GMRA is split into the squared bias and the variance. A corollary of Lemma 15 with $\Lambda = \Lambda_j$ results in an estimate of the variance term.

Proposition 16 *For any $\eta \geq 0$,*

$$\mathbb{P}\{\|\mathcal{P}_j X - \widehat{\mathcal{P}}_j X\| \geq \eta\} \leq \alpha d \#\Lambda_j e^{-\frac{\beta 2^{2j} n \eta^2}{d^2 \#\Lambda_j}} \tag{18}$$

$$\mathbb{E}\|\mathcal{P}_j X - \widehat{\mathcal{P}}_j X\|^2 \leq \frac{d^2 \#\Lambda_j \log[\alpha d \#\Lambda_j]}{\beta 2^{2j} n}. \tag{19}$$

In Eq. (1), the squared bias decays like $\mathcal{O}(2^{-2js})$ whenever $\rho \in \mathcal{A}_s$ and the variance scales like $\mathcal{O}(j2^{j(d-2)}/n)$. A proper choice of the scale j gives rise to Theorem 4 whose proof is given below.

Proof of Theorem 4

Proof [Proof of Theorem 4]

$$\begin{aligned}\mathbb{E}\|X - \widehat{\mathcal{P}}_j X\|^2 &\leq 2\|X - \mathcal{P}_j X\|^2 + 2\mathbb{E}\|\mathcal{P}_j X - \widehat{\mathcal{P}}_j X\|^2 \\ &\leq 2|\rho|^2_{\mathcal{A}_s} 2^{-2sj} + \frac{2d^2 \#\Lambda_j \log[\alpha d \#\Lambda_j]}{\beta 2^{2j} n} \leq 2|\rho|^2_{\mathcal{A}_s} 2^{-2sj} + \frac{2d^2 2^{j(d-2)}}{\theta_1 \beta n} \log \frac{\alpha d 2^{jd}}{\theta_1}\end{aligned}$$

as $\#\Lambda_j \leq 2^{jd}/\theta_1$ due to Assumption (A3).

Intrinsic dimension $d = 1$: In this case, both the squared bias and the variance decrease as j increases, so we should choose the scale j^* as large as possible as long as most cells at scale j^* have d points. We will choose j^* such that $2^{-j^*} = \mu\frac{\log n}{n}$ for some $\mu > 0$. After grouping Λ_{j^*} into light and heavy cells whose measure is below or above $\frac{28(\nu+1)\log n}{3n}$, we can show that the error on light cells is upper bounded by $C(\frac{\log n}{n})^2$ and all heavy cells have at least d points with high probability (see Lemma 17).

Lemma 17 *Suppose j^* is chosen such that $2^{-j^*} = \mu\frac{\log n}{n}$ with some $\mu > 0$. Then*

$$\|(X - \mathcal{P}_{j^*}X)\mathbf{1}_{\{C_{j^*,k}:\rho(C_{j^*,k})\leq\frac{28(\nu+1)\log n}{3n}\}}\|^2 \leq \frac{28(\nu+1)\theta_2^2\mu}{3\theta_1}\left(\frac{\log n}{n}\right)^2,$$

$$\mathbb{P}\left\{\textit{each } C_{j^*,k} \textit{ satisfying } \rho(C_{j^*,k}) > \frac{28(\nu+1)\log n}{3n} \textit{ has at least } d \textit{ points}\right\} \geq 1 - n^{-\nu}.$$

Lemma 17 is proved in appendix D. If j^* is chosen as above, The probability estimate in (5) follows from

$$\|X - \mathcal{P}_{j^*}X\| \leq |\rho|_{\mathcal{A}_s} 2^{-sj^*} \leq |\rho|_{\mathcal{A}_s}\mu^s\left(\frac{\log n}{n}\right)^s \leq |\rho|_{\mathcal{A}_s}\mu^s\frac{\log n}{n},$$

$$\mathbb{P}\left\{\|\mathcal{P}_{j^*}X - \widehat{\mathcal{P}}_{j^*}X\| \geq C_1\frac{\log n}{n}\right\} \leq \frac{\alpha}{\theta_1\mu}\left((\log n)/n\right)^{-1} e^{-\frac{\mu\beta\theta_1 C_1^2 \log n}{d^2}} \leq \frac{\alpha}{\theta_1\mu}\frac{n\, n^{-\frac{\mu\beta\theta_1 C_1^2}{d^2}}}{\log n} \leq C_2 n^{-\nu}$$

provided that C_1 is chosen such that $\mu\beta\theta_1 C_1^2/d^2 - 1 > \nu$.

Intrinsic dimension $d \geq 2$: When $d \geq 2$, the squared bias decreases but the variance increases as j gets large. We choose j^* such that $2^{-j^*} = \mu\left((\log n)/n\right)^{\frac{1}{2s+d-2}}$ to balance these two terms. We use the same technique as $d = 1$ to group Λ_{j^*} into light and heavy cells whose measure is below and above, repectively, $28/3 \cdot (\nu+1)(\log n)/n$, we can show that the error on light cells is upper bounded by $C((\log n)/n)^{\frac{2s}{2s+d-2}}$ and all heavy cells have at least d points with high probability (see Lemma 18).

Lemma 18 *Let j^* be chosen such that $2^{-j^*} = \mu\left((\log n)/n\right)^{\frac{1}{2s+d-2}}$ with some $\mu > 0$. Then*

$$\|(X - \mathcal{P}_{j^*}X)\mathbf{1}_{\{C_{j^*,k}:\rho(C_{j^*,k})\leq\frac{28(\nu+1)\log n}{3n}\}}\|^2 \leq \frac{28(\nu+1)\theta_2^2\mu^{2-d}}{3\theta_1}\left(\frac{\log n}{n}\right)^{\frac{2s}{2s+d-2}},$$

$$\mathbb{P}\left\{\forall\, C_{j^*,k} : \rho(C_{j^*,k}) > \frac{28(\nu+1)\log n}{3n},\ C_{j^*,k} \textit{ has at least } d \textit{ points}\right\} \geq 1 - n^{-\nu}.$$

Proof of Lemma 18 is omitted since it is the same as the proof of Lemma 17. The probability estimate in (6) follows from

$$\|X - \mathcal{P}_{j^*}X\| \leq |\rho|_{\mathcal{A}_s} 2^{-sj^*} \leq |\rho|_{\mathcal{A}_s}\mu^s\left(\frac{\log n}{n}\right)^{\frac{s}{2s+d-2}},$$

$$\mathbb{P}\left\{\|\mathcal{P}_{j^*}X - \widehat{\mathcal{P}}_{j^*}X\| \geq C_1\left(\frac{\log n}{n}\right)^{\frac{s}{2s+d-2}}\right\} \leq \frac{\alpha d\mu^{-d}}{\theta_1}\left(\frac{\log n}{n}\right)^{-\frac{d}{2s+d-2}} e^{-\frac{\beta\theta_1 C_1^2\mu^{d-2}\log n}{d^2}} \leq C_2 n^{-\nu}$$

provided that $\beta\theta_1 C_1^2\mu^{d-2}/d^2 - 1 > \nu$. ■

4.3. Performance analysis of empirical GMRA on adaptive partitions

Proof [Proof of Theorem 8] In the case that $\mathcal{M}$ is bounded by M, the minimum scale $j_{\min} = \log_2 \frac{\theta_2}{M}$. We first consider the case $d \geq 3$. In our proof C stands for constants that may vary at different locations, but it is independent of n and D. We will begin by defining several objects of interest:

- $\mathcal{T}^n$: the data master tree whose leaf contains at least d points in $\mathcal{X}_n$. It can be viewed as the part of a multiscale tree that our data have explored.
- $\mathcal{T}$: a complete multiscale tree containing $\mathcal{T}^n$. $\mathcal{T}$ can be viewed as the union $\mathcal{T}^n$ and some empty cells, mostly at fine scales with high probability, that our data have not explored.
- $\mathcal{T}_{(\rho,\eta)}$: the smallest subtree of $\mathcal{T}$ which contains $\{C_{j,k} \in \mathcal{T} : \Delta_{j,k} \geq 2^{-j}\eta\}$.
- $\mathcal{T}_\eta = \mathcal{T}_{(\rho,\eta)} \cap \mathcal{T}^n$.
- $\widehat{\mathcal{T}}_\eta$: the smallest subtree of $\mathcal{T}^n$ which contains $\{C_{j,k} \in \mathcal{T}^n : \widehat{\Delta}_{j,k} \geq 2^{-j}\eta\}$.
- $\Lambda_{(\rho,\eta)}$: the partition associated with $\mathcal{T}_{(\rho,\eta)}$.
- Λ_η : the partition associated with $\mathcal{T}_\eta$.
- $\widehat{\Lambda}_\eta$: the partition associated with $\widehat{\mathcal{T}}_\eta$.
- Suppose $\mathcal{T}^0$ and $\mathcal{T}^1$ are two subtrees of $\mathcal{T}$. If Λ^0 and Λ^1 are two adaptive partitions associated with $\mathcal{T}^0$ and $\mathcal{T}^1$ respectively, we denote by $\Lambda^0 \vee \Lambda^1$ and $\Lambda^0 \wedge \Lambda^1$ the partitions associated to the trees $\mathcal{T}^0 \cup \mathcal{T}^1$ and $\mathcal{T}^0 \cap \mathcal{T}^1$ respectively.

We also let $b = 2a_{\max} + 5$ where $a_{\max}$ is the maximal number of children that a node has in $\mathcal{T}$; $\kappa_0 = \max(\kappa_1, \kappa_2)$ where $b^2\kappa_1^2/(21\theta_2^2) = \nu + 1$ and $\alpha_2\kappa_2^2/b^2 = \nu + 1$ with α_2 defined in Lemma 20. In order the obtain the MSE bound, one can simply set $\nu = 1$.

The empirical Adaptive GMRA projection is given by $\widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}} = \sum_{C_{j,k} \in \widehat{\Lambda}_{\tau_n}} \widehat{\mathcal{P}}_{j,k} \mathbf{1}_{j,k}$. Using the triangle inequality, we split the error as follows:

$$\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}} X\| \leq e_1 + e_2 + e_3 + e_4$$

where

$$e_1 := \|X - \mathcal{P}_{\widehat{\Lambda}_{\tau_n} \vee \Lambda_{b\tau_n}} X\|, \qquad e_2 := \|\mathcal{P}_{\widehat{\Lambda}_{\tau_n} \vee \Lambda_{b\tau_n}} X - \mathcal{P}_{\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}} X\|$$
$$e_3 := \|\mathcal{P}_{\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}} X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}} X\|, \qquad e_4 := \|\widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}} X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}} X\|.$$

A similar split appears in the works of Binev et al. (2005, 2007). The partition built from those $C_{j,k}$'s satisfying $\widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n$ does not exactly coincide with the partition chosen based on those $C_{j,k}$ satisfying $\Delta_{j,k} \geq 2^{-j}\tau_n$. This is accounted by e_2 and e_4, corresponding to those $C_{j,k}$'s whose $\widehat{\Delta}_{j,k}$ is significantly larger or smaller than $\Delta_{j,k}$, which we will prove to be small with high probability. The remaining terms e_1 and e_3 correspond to the bias and variance of the approximations on the partition obtained by thresholding $\Delta_{j,k}$.

Term e_1: The first term e_1 is essentially the bias term. Since $\widehat{\Lambda}_{\tau_n} \vee \Lambda_{b\tau_n} \supseteq \Lambda_{b\tau_n}$,

$$e_1^2 = \|X - \mathcal{P}_{\widehat{\Lambda}_{\tau_n} \vee \Lambda_{b\tau_n}} X\|^2 \leq \|X - \mathcal{P}_{\Lambda_{b\tau_n}} X\|^2 \leq \underbrace{\|X - \mathcal{P}_{\Lambda_{(\rho,b\tau_n)}} X\|^2}_{e_{11}^2} + \underbrace{\|\mathcal{P}_{\Lambda_{(\rho,b\tau_n)}} X - \mathcal{P}_{\Lambda_{b\tau_n}} X\|^2}_{e_{12}^2}.$$

e_{11}^2 may be upper bounded deterministically from Eq. (9):

$$e_{11}^2 \leq B_{s,d}|\rho|_{\mathcal{B}_s}^p (b\tau_n)^{2-p} \leq B_{s,d}|\rho|_{\mathcal{B}_s}^{\frac{2(d-2)}{2s+d-2}} (b\kappa)^{\frac{4s}{2s+d-2}} \left(\frac{\log n}{n}\right)^{\frac{2s}{2s+d-2}}. \tag{20}$$

e_{12} encodes the difference between thresholding $\mathcal{T}$ and $\mathcal{T}^n$, but it is 0 with high probability:

Lemma 19 *For any $\nu > 0$, κ such that $\kappa > \kappa_1$, where $b^2\kappa_1^2/(21\theta_2^2) = \nu + 1$,*

$$\mathbb{P}\{e_{12} > 0\} \leq C(\theta_2, a_{\max}, a_{\min}, \kappa) n^{-\nu} \tag{21}$$

The proof is postponed, together with those of the Lemmata that follow, to appendix D). If $\mathcal{M}$ is bounded by M, then $e_{12}^2 \leq 4M^2$ and

$$\mathbb{E}e_{12}^2 \leq 4M^2\mathbb{P}\{e_{12} > 0\} \leq 4M^2 C n^{-\nu} \leq 4M^2 C \left(\frac{\log n}{n}\right)^{\frac{2s}{2s+d-2}} \tag{22}$$

if $\nu > 2s/(2s+d-2)$, for example $\nu = 1$.

Term e_3: e_3 corresponds to the variance on the partition $\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}$. For any $\eta > 0$,

$$\mathbb{P}\{e_3 > \eta\} \leq \alpha d \#(\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}) e^{-\frac{\beta n \eta^2}{d^2 \sum_{j \geq j_{\min}} 2^{-2j} \#_j(\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b})}}$$

according to Lemma 15. Since $\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b} \subset \mathcal{T}_{\tau_n/b}$, for any $j \geq 0$, regardless of $\widehat{\Lambda}_{\tau_n}$, we have $\#_j(\widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}) \leq \#_j \mathcal{T}_{\tau_n/b} \leq \#\mathcal{T}_{\tau_n/b}$. Therefore

$$\mathbb{P}\{e_3 > \eta\} \leq \alpha d \# \mathcal{T}_{\tau_n/b} e^{-\frac{\beta n \eta^2}{d^2 \sum_{j \geq j_{\min}} 2^{-2j} \#_j \mathcal{T}_{\tau_n/b}}} \leq \alpha d \# \mathcal{T}_{\tau_n/b} e^{-\frac{\beta n \eta^2}{d^2 |\rho|_{\mathcal{B}_s}^p (\tau_n/b)^{-p}}}, \tag{23}$$

which implies

$$\mathbb{E}e_3^2 = \int_0^{+\infty} \eta \mathbb{P}\{e_3 > \eta\} d\eta = \int_0^{+\infty} \eta \min\left(1, \alpha d \# \mathcal{T}_{\tau_n/b} e^{-\frac{\beta n \eta^2}{d^2 \sum_{j \geq j_{\min}} 2^{-2j} \#_j \mathcal{T}_{\tau_n/b}}}\right) d\eta$$
$$\leq \frac{d^2 \log \alpha d \# \mathcal{T}_{\tau_n/b}}{\beta n} \sum_{j \geq j_{\min}} 2^{-2j} \#_j \mathcal{T}_{\tau_n/b} \leq C \frac{\log n}{n} \left(\frac{\tau_n}{b}\right)^{-p} \leq C(\theta_2, \theta_3, d, \kappa, s, |\rho|_{\mathcal{B}_s}) \left(\frac{\log n}{n}\right)^{\frac{2s}{2s+d-2}}.$$

Term e_2 and e_4: These terms account for the difference of truncating the master tree based on $\Delta_{j,k}$'s and its empirical counterparts $\widehat{\Delta}_{j,k}$'s. We prove that $\widehat{\Delta}_{j,k}$'s concentrate near $\Delta_{j,k}$'s with high probability if there are sufficient samples.

Lemma 20 *For any $\eta > 0$ and any $C_{j,k} \in \mathcal{T}$*

$$\max\left\{\mathbb{P}\left\{\widehat{\Delta}_{j,k} \leq \eta \ \text{ and } \ \Delta_{j,k} \geq b\eta\right\}, \mathbb{P}\left\{\Delta_{j,k} \leq \eta \ \text{ and } \ \widehat{\Delta}_{j,k} \geq b\eta\right\}\right\} \leq \alpha_1 e^{-\alpha_2 2^{2j} n \eta^2} \tag{24}$$

for some constants $\alpha_1 := \alpha_1(\theta_2, \theta_3, a_{\max}, d)$ and $\alpha_2 := \alpha_2(\theta_2, \theta_3, \theta_4, a_{\max}, d)$.

This Lemma enables one to show that $e_2 = 0$ and $e_4 = 0$ with high probability:

Lemma 21 *Let α_1 and α_2 be the constants in Lemma 20. For any fixed $\nu > 0$,*

$$\mathbb{P}\{e_2 > 0\} + \mathbb{P}\{e_4 > 0\} \le \alpha_1 a_{\min} n^{-\nu} \tag{25}$$

when κ is chosen such that $\kappa > \kappa_2$, with $\alpha_2\kappa_2^2/b^2 = \nu + 1$.

Since $\mathcal{M}$ is bounded by M, we have $e_2^2 \le 4M^2$ so

$$\mathbb{E}e_2^2 \le 4M^2\mathbb{P}\{e_2 > 0\} \le 4M^2\alpha_1 a_{\min} n^{-\nu} \le 4M^2\alpha_1 a_{\min}\left(\frac{\log n}{n}\right)^{\frac{2s}{2s+d-2}}$$

if $\nu > 2s/(2s+d-2)$, for example $\nu = 1$. The same bound holds for e_4.

Finally, we complete the probability estimate (10): let $c_0^2 = B_{s,d}|\rho|_{\mathcal{B}_s}^{\frac{2(d-2)}{2s+d-2}}(b\kappa)^{\frac{4s}{2s+d-2}}$ such that $e_{11} \le c_0\left((\log n)/n\right)^{\frac{s}{2s+d-2}}$. We have

$$\begin{aligned}
&\mathbb{P}\left\{\|X - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}}X\| \ge c_1\left((\log n)/n\right)^{\frac{s}{2s+d-2}}\right\}\\
&\le \mathbb{P}\left\{e_3 > (c_1 - c_0)\left((\log n)/n\right)^{\frac{s}{2s+d-2}}\right\} + \mathbb{P}\{e_{12} > 0\} + \mathbb{P}\{e_2 > 0\} + \mathbb{P}\{e_4 > 0\}\\
&\le \mathbb{P}\left\{e_3 > (c_1 - c_0)\left((\log n)/n\right)^{\frac{s}{2s+d-2}}\right\} + Cn^{-\nu},
\end{aligned}$$

as long as κ is chosen such that $\kappa > \max(\kappa_1, \kappa_2)$ where $b^2\kappa_1^2/(21\theta_2^2) = \nu + 1$ and $\alpha_2\kappa_2^2/b^2 = \nu + 1$ according to (21) and (25). Applying (23) gives rise to

$$\begin{aligned}
&\mathbb{P}\left\{e_3 > (c_1 - c_0)\left((\log n)/n\right)^{\frac{s}{2s+d-2}}\right\} \le \alpha d\#\mathcal{T}_{\tau_n/b}e^{-\frac{\beta n}{|\rho|_{\mathcal{B}_s}^p(\tau_n/b)^{-p}}(c_1-c_0)^2\left((\log n)/n\right)^{\frac{2s}{2s+d-2}}}\\
&\le \alpha d\#\mathcal{T}_{\tau_n/b}n^{-\frac{\beta(c_1-c_0)^2\kappa^p}{b^p|\rho|_{\mathcal{B}_s}^p}} \le \alpha d a_{\min} n^{-\left(\frac{\beta(c_1-c_0)^2\kappa^p}{b^p|\rho|_{\mathcal{B}_s}^p}-1\right)} \le \alpha d a_{\min} n^{-\nu}
\end{aligned}$$

if c_1 is taken large enough such that $\frac{\beta(c_1-c_0)^2\kappa^p}{b^p|\rho|_{\mathcal{B}_s}^p} \ge \nu + 1$.

We are left with the cases $d = 1, 2$. When $d = 1$, for any distribution ρ satisfying quasi-orthogonality (8) and any $\eta > 0$, the tree complexity may be bounded as follows:

$$\sum_{j\ge j_{\min}} 2^{-2j}\#_j\mathcal{T}_{(\rho,\eta)} \le \sum_{j\ge j_{\min}} 2^{-2j}2^j/\theta_1 = 2/\theta_1 2^{-j_{\min}} = 2M/(\theta_1\theta_2),$$

so $\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}}X\|^2 \le 8MB_0\eta^2/(3\theta_1\theta_2)$. Hence

$$e_{11}^2 \le \tfrac{8MB_0}{3\theta_1\theta_2}(b\tau_n)^2 \le \tfrac{8MB_0b^2\kappa^2}{3\theta_1\theta_2}(\log n)/n, \quad \mathbb{P}\{e_3 > \eta\} \le \alpha d\#\mathcal{T}_{\tau_n/b}e^{-\frac{\theta_1\theta_2\beta n\eta^2}{2Md^2}},$$

which yield $\mathbb{E}e_3^2 \le 2Md^2\log\alpha d\#\mathcal{T}_{\tau_n/b}/(\theta_1\theta_2\beta n) \le C(\log n)/n$ and estimate (11).

When $d = 2$, for any distribution satisfying quasi-orthogonality and given any $\eta > 0$, we have $\sum_{j\ge j_{\min}} 2^{-2j}\#_j\mathcal{T}_{(\rho,\eta)} \le -|\rho|^{-1}\log\eta$, whence $\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}}X\|^2 \le -\frac{4}{3}B_0|\rho|\eta^2\log\eta$. Therefore

$$e_{11}^2 \le -\tfrac{4}{3}B_0|\rho|(b\tau_n)^2\log(b\tau_n) \le C(\log^2 n)/n \quad , \quad \mathbb{P}\{e_3 > \eta\} \le \alpha d\#\mathcal{T}_{\tau_n/b}e^{-\frac{2\beta n\eta^2}{d^2|\rho|\log n}},$$

which yield $\mathbb{E}e_3^2 \le d^2|\rho|\log\alpha d\#\mathcal{T}_{\tau_n/b}(\log n)/(2\beta n) \le C(\log^2 n)/n$ and the probability estimate (12). ■

Proof [Proof of Theorem 10] Let $R > 0$. If we run Adaptive GMRA on $B_R(0)$, and approximate points outside $B_R(0)$ by 0, the MSE of the Adaptive GMRA in $B_R(0)$ is

$$\|(\mathbb{I} - \widehat{\mathcal{P}}_{\widehat{\Lambda}_{\tau_n}})\mathbf{1}_{\{\|x\|\le R\}}X\|^2 \lesssim (|\rho|_{B_0(R)}|^p + R^2)\left((\log n)/n\right)^{\frac{2s}{2s+d-2}} \lesssim R^{\max(\lambda,2)}\left((\log n)/n\right)^{\frac{2s}{2s+d-2}}.$$

The squared error outside $B_R(0)$ is

$$\|\mathbf{1}_{\{\|x\|\ge R\}}X\|^2 = \int_{B_R(0)^c}||x||^2 d\rho \le CR^{-\delta}. \tag{26}$$

The total MSE is

$$\text{MSE} \lesssim R^{\max(\lambda,2)}\left((\log n)/n\right)^{\frac{2s}{2s+d-2}} + R^{-\delta}.$$

Minimizing over R suggests taking $R = R_n = \max(R_0, \mu(\log n/n)^{-\frac{2s}{(2s+d-2)(\delta+\max(2,\lambda))}})$, yielding MSE $\lesssim \left((\log n)/n\right)^{\frac{2s}{2s+d-2}\cdot\frac{\delta}{\delta+\max(\lambda,2)}}$. The probability estimate (13) follows from Eq. (26) and Eq. (10) in Theorem 8. ■

In Remark 11, we claim that λ is not large in simple cases. If $\rho \in \mathcal{A}_s^\infty$ and ρ decays such that $\rho(C_{j,k}) \le 2^{-jd}\|c_{j,k}\|^{-(d+1+\delta)}$, we have $\Delta_{j,k} \le 2^{-js}2^{-jd/2}\|c_{j,k}\|^{-(d+1+\delta)/2}$. Roughly speaking, for any $\eta > 0$, the cells of distance r to 0 satisfying $\Delta_{j,k} \ge 2^{-j}\eta$ will satisfy $2^{-j} \ge (\eta r^{\frac{d+1+\delta}{2}})^{\frac{2}{2s+d-2}}$. In other words, the cells of distance r to 0 are truncated at scale $j_{\max}$ such that $2^{-j_{\max}} = (\eta r^{\frac{d+1+\delta}{2}})^{\frac{2}{2s+d-2}}$, which gives rise to complexity $\le 2^{-2j_{\max}}r^{d-1}2^{j_{\max}d} \le \eta^{-\frac{2(d-2)}{2s+d-2}}r^{d-1-\frac{(d+1+\delta)(d-2)}{2s+d-2}}$. If we run Adaptive GMRA with threshold η on $B_R(0)$, the weighted complexity of the truncated tree is upper bounded by $\eta^{-\frac{2(d-2)}{2s+d-2}}r^{d-\frac{(d+1+\delta)(d-2)}{2s+d-2}}$. Therefore, $\rho_{|B_R(0)} \in \mathcal{B}_s$ for all $R > 0$ and $|\rho_{|B_R(0)}|_{\mathcal{B}_s}^p \le R^\lambda$ with $\lambda = d - \frac{(d+1+\delta)(d-2)}{2s+d-2}$.

5. Discussions and extensions

5.1. Computational complexity

The computational cost in GMRA and Adaptive GMRA may be split as follows:

Tree construction: Cover tree itself is an online algorithm where a single-point insertion or removal takes cost at most $O(\log n)$. The total computational cost of the cover tree algorithm is $C^dDn\log n$ where $C > 0$ is a constant (Beygelzimer et al., 2006).

Local PCA: At every scale j, we perform local PCA on the training data restricted to the $C_{j,k}$ for every $k \in \mathcal{K}_j$, using the random PCA algorithm (Halko et al., 2009). Recall that $\widehat{n}_{j,k}$ denotes the number of training points in $C_{j,k}$. The cost of local PCA at scale j is in the order of $\sum_{k\in\mathcal{K}_j} Dd\widehat{n}_{j,k} = Ddn$, and there are about $1/d\log n$ scales which gives rise to

the total cost of $Dn \log n$.

Adaptive approximation: To achieve an adaptive approximation, we need to compute the empirical geometric wavelet coefficients $\widehat{\Delta}_{j,k}$ for every $C_{j,k}$, which costs $2Dd\widehat{n}_{j,k}$ on $C_{j,k}$ and $2Dn \log n$ for the whole tree.

The computational costs of GMRA and Adaptive GMRA are summarized in Table 3.

Operations	Computational cost
Multiscale tree construction	$C^d Dn \log n$
Randomized PCA at all nodes	$Dn \log n$
Computing $\Delta_{j,k}$'s	$2Dn \log n$
GMRA	$C^d Dn \log n + Dn \log n$
Adaptive GMRA	$C^d Dn \log n + 3Dn \log n$
Compute $\mathcal{P}_j(x)$ for a test point	$\underbrace{D \log n}_{\text{find } C_{j,k} \text{ containing } x} + \underbrace{Dd}_{\text{compute } \mathcal{P}_{j,k}(x)} = D(\log n + d)$

Table 3: Computational cost

5.2. Quasi-orthogonality

A main difference between GMRA and orthonormal wavelet bases (see Daubechies, 1992; Mallat, 1998) is that $V_{j,x} \nsubseteq V_{j+1,x}$ where $(j,x) = (j,k)$ such that $x \in C_{j,k}$. Therefore the geometric wavelet subspace $\mathrm{Proj}_{V_{j,x}{}^{\perp}} V_{j+1,x}$ which encodes the difference between $V_{j+1,x}$ and $V_{j,x}$ is in general not orthogonal across scales.

Theorem 8 involves a quasi-orthogonality condition (8), which is satisfied if the operators $\{\mathcal{Q}_{j,k}\}$ applied on $\mathcal{M}$ are rapidly decreasing in norm or are orthogonal. When $\rho \in \mathcal{A}_1^\infty$ such that $\|\mathcal{Q}_{j,k}X\| \sim 2^{-j}\sqrt{\rho(C_{j,k})}$, quasi-orthogonality is guaranteed. In this case, for any node $C_{j,k}$ and $C_{j',k'} \subset C_{j,k}$, we have $\|\mathcal{Q}_{j',k'}X\|/\sqrt{\rho(C_{j',k'})} \lesssim 2^{-(j'-j)}\|\mathcal{Q}_{j,k}X\|/\sqrt{\rho(C_{j,k})}$, which implies $\sum_{C_{j',k'} \subset C_{j,k}} \langle \mathcal{Q}_{j,k}X, \mathcal{Q}_{j',k'}X\rangle \lesssim 2\|\mathcal{Q}_{j,k}X\|^2$. Therefore $B_0 \lesssim 2$. Another setting is when $\mathcal{Q}_{j',k'}$ and $\mathcal{Q}_{j,k}$ are orthogonal whenever $C_{j',k'} \subset C_{j,k}$, as guaranteed in orthogonal GMRA in Section 5.3, in which case exact orthogonality is automatically satisfied.

Quasi-orthogonality enters in the proof of Eq. (9). If quasi-orthogonality is violated, we still have a convergence result in Theorem 8 but the convergence rate will be worse: MSE $\lesssim [(\log n)/n]^{\frac{s}{2s+d-2}}$ when $d \geq 3$ and MSE $\lesssim [(\log^d n)/n]^{\frac{1}{2}}$ when $d = 1, 2$.

5.3. Orthogonal GMRA and adaptive orthogonal GMRA

A different construction, called orthogonal geometric multi-resolution analysis in Section 5 of Allard et al. (2012), follows the classical wavelet theory by constructing a sequence

of increasing subspaces and then the corresponding wavelet subspaces exactly encode the orthogonal complement across scales. Exact orthogonality is therefore satisfied.

5.3.1. Orthogonal GMRA

In the construction, we build the sequence of subspaces $\{\widehat{S}_{j,k}\}_{k \in \mathcal{K}_j, j \geq j_{\min}}$ with a coarse-to-fine algorithm in Table 4. For fixed x and j, (j,x) denotes (j,k) such that $x \in C_{j,k}$. In orthogonal GMRA the sequence of subspaces $S_{j,x}$ is increasing such that $S_{0,x} \subset S_{1,x} \subset \cdots S_{j,x} \subset S_{j+1,x} \cdots$ and the subspace $U_{j+1,x}$ exactly encodes the orthogonal complement of $S_{j,x}$ in $S_{j+1,x}$. Orthogonal GMRA with respect to the distribution ρ corresponds to affine projectors onto the subspaces $\{S_{j,k}\}_{k \in \mathcal{K}_j, j \geq j_{\min}}$.

	Orthogonal GMRA	Empirical orthogonal GMRA
Subpaces	$S_{0,x} = V_{0,x}$	$\widehat{S}_{0,x} = \widehat{V}_{0,x}$
	$U_{1,x} = \mathrm{Proj}_{S_{0,x}^{\perp}} V_{1,x},\ S_{1,x} = S_{0,x} \oplus U_{1,x}$	$\widehat{U}_{1,x} = \mathrm{Proj}_{\widehat{S}_{0,x}^{\perp}} \widehat{V}_{1,x},\ \widehat{S}_{1,x} = \widehat{S}_{0,x} \oplus \widehat{U}_{1,x}$
	$\cdots$	$\cdots$
	$U_{j+1,x} = \mathrm{Proj}_{S_{j,x}^{\perp}} V_{j+1,x}$	$\widehat{U}_{j+1,x} = \mathrm{Proj}_{\widehat{S}_{j,x}^{\perp}} \widehat{V}_{j+1,x}$
	$S_{j+1,x} = S_{j,x} \oplus U_{j+1,x}$	$\widehat{S}_{j+1,x} = \widehat{S}_{j,x} \oplus \widehat{U}_{j+1,x}$
Affine projectors	$\mathcal{S}_j := \sum_{k \in \mathcal{K}_j} \mathcal{S}_{j,k} \mathbf{1}_{j,k}$	$\widehat{\mathcal{S}}_j := \sum_{k \in \mathcal{K}_j} \widehat{\mathcal{S}}_{j,k} \mathbf{1}_{j,k}$
	$\mathcal{S}_{j,k}(x) := c_{j,k} + \mathrm{Proj}_{S_{j,k}}(x - c_{j,k})$	$\widehat{\mathcal{S}}_{j,k}(x) := \widehat{c}_{j,k} + \mathrm{Proj}_{\widehat{S}_{j,k}}(x - \widehat{c}_{j,k})$

Table 4: Orthogonal GMRA

For a fixed distribution ρ, the approximation error $\|X - \mathcal{S}_j X\|$ decays as j increases. We will consider the model class $\mathcal{A}_s^{\mathrm{o}}$ where $\|X - \mathcal{S}_j X\|$ decays like $\mathcal{O}(2^{-js})$.

Definition 22 *A probability measure ρ supported on $\mathcal{M}$ is in $\mathcal{A}_s^{\mathrm{o}}$ if*

$$|\rho|_{\mathcal{A}_s^{\mathrm{o}}} = \sup_{\mathcal{T}} \inf\{A_0^o : \|X - \mathcal{S}_j X\| \leq A_0^o 2^{-js}, \forall\, j \geq j_{\min}\} < \infty\,, \tag{27}$$

where $\mathcal{T}$ varies over the set, assumed non-empty, of multiscale tree decompositions satisfying Assumptions (A1-A5).

Notice that $\mathcal{A}_s \subset \mathcal{A}_s^{\mathrm{o}}$. We split the MSE into the squared bias and the variance as: $\mathbb{E}\|X - \widehat{\mathcal{S}}_j X\|^2 = \|X - \mathcal{S}_j X\|^2 + \mathbb{E}\|\mathcal{S}_j X - \widehat{\mathcal{S}}_j X\|^2$. The squared bias $\|X - \mathcal{S}_j X\|^2 \leq |\rho|^2_{\mathcal{A}_s^{\mathrm{o}}} 2^{-2js}$ whenever $\rho \in \mathcal{A}_s^{\mathrm{o}}$. In Lemma 34 we show $\mathbb{E}\|\mathcal{S}_j X - \widehat{\mathcal{S}}_j X\|^2 \leq \frac{d^2 j^4 \#\Lambda_j \log[\alpha j \#\Lambda_j]}{\beta 2^{2j} n} = \mathcal{O}\left(\frac{j^5 2^{j(d-2)}}{n}\right)$ where α and β are the constants in Lemma 15. A proper choice of the scale yields the following result:

Theorem 23 *Assume that $\rho \in \mathcal{A}_s^o$, $s \geq 1$. Let $\nu > 0$ be arbitrary and $\mu > 0$. If j^* is properly chosen such that*

$$2^{-j^*} = \begin{cases} \mu \frac{\log n}{n} & \text{for } d = 1 \\ \mu \left(\frac{\log^5 n}{n}\right)^{\frac{1}{2s+d-2}} & \text{for } d \geq 2 \end{cases},$$

then there exists a constant $C_1(\theta_1, \theta_2, \theta_3, \theta_4, d, \nu, \mu, s)$ such that

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{S}}_{j^*} X\| \geq (|\rho|_{\mathcal{A}_s^o} \mu^s + C_1) \frac{\log^5 n}{n}\right\} \leq C_2(\theta_1, \theta_2, \theta_3, \theta_4, d, \mu) n^{-\nu} \quad \text{for } d = 1,$$

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{S}}_{j^*} X\| \geq (|\rho|_{\mathcal{A}_s^o} \mu^s + C_1) \left(\frac{\log^5 n}{n}\right)^{\frac{s}{2s+d-2}}\right\} \leq C_2(\theta_1, \theta_2, \theta_3, \theta_4, d, \mu, s) n^{-\nu} \quad \text{for } d \geq 2\,. \tag{28}$$

Theorem 23 is proved in appendix E.1.

5.3.2. Adaptive Orthogonal GMRA

Definition (infinite sample)	Empirical version
$\Delta_{j,k}^o := \|(\mathcal{S}_j - \mathcal{S}_{j+1})\mathbf{1}_{j,k} X\|$ $= \left(\|(\mathbb{I} - \mathcal{S}_j)\mathbf{1}_{j,k} X\|^2 - \|(\mathbb{I} - \mathcal{S}_{j+1})\mathbf{1}_{j,k} X\|^2\right)^{\frac{1}{2}}$	$\widehat{\Delta}_{j,k}^o := \left\|(\widehat{\mathcal{S}}_j - \widehat{\mathcal{S}}_{j+1})\mathbf{1}_{j,k} X\right\|$ $= \left(\left\|(\mathbb{I} - \widehat{\mathcal{S}}_j)\mathbf{1}_{j,k} X\right\|^2 - \left\|(\mathbb{I} - \widehat{\mathcal{S}}_{j+1})\mathbf{1}_{j,k} X\right\|^2\right)^{\frac{1}{2}}$

Table 5: Refinement criterion in adaptive orthogonal GMRA

Orthogonal GMRA can be constructed adaptively to the data with the refinement criterion defined in Table 5. We let $\tau_n^o := \kappa(\log^5 n/n)^{\frac{1}{2}}$ where κ is a constant, truncate the data master tree $\mathcal{T}^n$ to the smallest proper subtree that contains all $C_{j,k} \in \mathcal{T}^n$ satisfying $\widehat{\Delta}_{j,k}^o \geq 2^{-j}\tau_n^o$, denoted by $\widehat{\mathcal{T}}_{\tau_n^o}$. Empirical adaptive orthogonal GMRA returns piecewise affine projectors on the adaptive partition $\widehat{\Lambda}_{\tau_n^o}$ consisting of the outer leaves of $\widehat{\mathcal{T}}_{\tau_n^o}$. Our algorithm is summarized in Algorithm 2.

If ρ is known, given any fixed threshold $\eta > 0$, we let $\mathcal{T}_{(\rho,\eta)}$ be the smallest proper tree of $\mathcal{T}$ that contains all $C_{j,k} \in \mathcal{T}$ for which $\Delta_{j,k}^o \geq 2^{-j}\eta$. This gives rise to an adaptive partition $\Lambda_{(\rho,\eta)}$ consisting the outer leaves of $\mathcal{T}_{(\rho,\eta)}$. We introduce a model class $\mathcal{B}_s^o$ for whose elements we can control the growth rate of the truncated tree $\mathcal{T}_{(\rho,\eta)}$ as η decreases.

Definition 24 *In the case $d \geq 3$, given $s > 0$, a probability measure ρ supported on $\mathcal{M}$ is in $\mathcal{B}_s^o$ if the following quantity is finite*

$$|\rho|_{\mathcal{B}_s^o}^p := \sup_{\mathcal{T}} \sup_{\eta > 0} \eta^p \sum_{j \geq j_{\min}} 2^{-2j} \#_j \mathcal{T}_{(\rho,\eta)} \text{ with } p = \frac{2(d-2)}{2s+d-2} \tag{29}$$

where $\mathcal{T}$ varies over the set, assumed non-empty, of multiscale tree decompositions satisfying Assumptions (A1-A5).

Algorithm 2 Empirical Adaptive Orthogonal GMRA

Input: data $\mathcal{X}_{2n} = \mathcal{X}_n' \cup \mathcal{X}_n$, intrinsic dimension d, threshold κ

Output: $\widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}}$: adaptive piecewise linear projectors

1: Construct $\mathcal{T}^n$ and $\{C_{j,k}\}$ from $\mathcal{X}_n'$

2: Compute $\widehat{\mathcal{S}}_{j,k}$ and $\widehat{\Delta}_{j,k}^o$ on every node $C_{j,k} \in \mathcal{T}^n$.

3: $\widehat{\mathcal{T}}_{\tau_n^o} \leftarrow$ smallest proper subtree of $\mathcal{T}^n$ containing all $C_{j,k} \in \mathcal{T}^n$: $\widehat{\Delta}_{j,k}^o \geq 2^{-j}\tau_n^o$ where $\tau_n^o = \kappa\sqrt{(\log^5 n)/n}$.

4: $\widehat{\Lambda}_{\tau_n^o} \leftarrow$ partition associated with the outer leaves of $\widehat{\mathcal{T}}_{\tau_n^o}$

5: $\widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}} \leftarrow \sum_{C_{j,k} \in \widehat{\Lambda}_{\tau_n^o}} \widehat{\mathcal{S}}_{j,k} \mathbf{1}_{j,k}$.

Notice that exact orthogonality is satisfied for orthogonal GMRA. One can show that, as long as $\rho \in \mathcal{B}_s^o$,

$$\|X - \mathcal{S}_{\Lambda_{(\rho,\eta)}} X\|^2 \leq B_{s,d}^o |\rho|_{\mathcal{B}_s^o}^p \eta^{2-p} \leq B_{s,d}^o |\rho|_{\mathcal{B}_s^o}^2 \left(\sum_{j \geq j_{\min}} 2^{-2j} \#_j \mathcal{T}_{(\rho,\eta)}\right)^{-\frac{2s}{d-2}},$$

where $B_{s,d}^o := 2^p/(1 - 2^{p-2})$. We can prove the following performance guarantee of the empirical adaptive orthogonal GMRA (see Appendix E.2):

Theorem 25 *Suppose $\mathcal{M}$ is bounded: $\mathcal{M} \subset B_M(0)$ and the multiscale tree satisfies $\rho(C_{j,k}) \leq \theta_0 2^{-jd}$ for some $\theta_0 > 0$. Let $d \geq 3$ and $\nu > 0$. There exists $\kappa_0(\theta_0, \theta_2, \theta_3, \theta_4, a_{\max}, d, \nu)$ such that if $\rho \in \mathcal{B}_s^o$ for some $s > 0$ and $\tau_n^o = \kappa\left[(\log^5 n)/n\right]^{\frac{1}{2}}$ with $\kappa \geq \kappa_0$, then there is a c_1 and c_2 such that*

$$\mathbb{P}\left\{\|X - \widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}} X\| \geq c_1 \left(\frac{\log^5 n}{n}\right)^{\frac{s}{2s+d-2}}\right\} \leq c_2 n^{-\nu}. \tag{30}$$

In Theorem 25, the constants are $c_1 := c_1(\theta_0, \theta_2, \theta_3, \theta_4, a_{\max}, d, s, \kappa, |\rho|_{\mathcal{B}_s^o}, \nu)$ and $c_2 := c_2(\theta_0, \theta_2, \theta_3, \theta_4, a_{\min}, a_{\max}, d, s, \kappa, |\rho|_{\mathcal{B}_s^o})$. Eq. (30) implies that MSE $\lesssim (\frac{\log^5 n}{n})^{\frac{2s}{2s+d-2}}$ for orthogonal Adaptive GMRA when $d \geq 3$. In the case of $d = 1, 2$, we can prove that MSE $\lesssim \frac{\log^{4+d} n}{n}$.

Acknowledgments

This research was partially funded by NSF-DMS-ATD-1222567, NSF-DMS-1708553, 1724979, 1818751, NSF-IIS-1546392 and AFOSR FA9550-14-1-0033. We are grateful to Duke University for donated computing time and equipment used for part of the numerical experiments of this paper.

Algorithm 3 Construction of a multiscale tree decomposition $\{C_{j,k}\}$

Input: data $\mathcal{X}_n'$

Output: A multiscale tree decomposition $\{C_{j,k}\}$

1: Run cover tree on $\mathcal{X}_n'$ to obtain a set of nets $\{T_j(\mathcal{X}_n')\}_{j\in[j_{\min},j_{\max}]}$

2: $j = j_{\min}$: $C_{j_{\min},0} = \widetilde{\mathcal{M}}$ defined in (31)

3: for $j = j_{\min}+1,\ldots,j_{\max}$: For every C_{j-1,k_0} at scale $j-1$, C_{j-1,k_0} has $\#(T_j(\mathcal{X}_n')\cap C_{j-1,k_0})$ children indexed by $a_{j,k}\in T_j(\mathcal{X}_n')\cap C_{j-1,k_0}$ with corresponding $C_{j,k}$'s constructed as follows:

$$C_{j,k}^{(j)} = \widetilde{\mathcal{M}} \bigcap \text{Voronoi}(a_{j,k}, T_j(\mathcal{X}_n')\cap C_{j-1,k_0})$$

and for $i = j+1,\ldots,j_{\max}$

$$C_{j,k}^{(i)} = \Big(\bigcup_{a_{i,k'}\in C_{j,k}^{(i-1)}} B_{\frac{1}{4}2^{-i}}(a_{i,k'})\Big)\bigcup C_{j,k}^{(i-1)}$$

Finally, let $C_{j,k} = C_{j,k}^{(j_{\max})}$.

Appendix A. Tree construction, regularity of geometric spaces

A.1. Tree construction

We now show that from a set of nets $\{T_j(\mathcal{X}_n')\}_{j\in[j_{\min},j_{\max}]}$ from the cover tree algorithm we can construct a set of $C_{j,k}$ with desired properties. Similar constructions are classical in harmonic analysis (Christ, 1990). Let $\{a_{j,k}\}_{k=1}^{N(j)}$ be the set of points in $T_j(\mathcal{X}_n')$. Given a set of points $\{z_1,\ldots,z_m\}\subset\mathbb{R}^D$, the Voronoi cell of z_ℓ with respect to $\{z_1,\ldots,z_m\}$ is defined as

$$\text{Voronoi}(z_\ell,\{z_1,\ldots,z_m\}) = \{x\in\mathbb{R}^D : \|x-z_\ell\|\le\|x-z_i\| \text{ for all } i\neq\ell\}.$$

Let

$$\widetilde{\mathcal{M}} = \bigcup_{j=j_{\min}}^{j_{\max}} \bigcup_{a_{j,k}\in T_j(\mathcal{X}_n')} B_{\frac{1}{4}2^{-j}}(a_{j,k})\,. \tag{31}$$

Our $C_{j,k}$'s are constructed in Algorithm 3. These $C_{j,k}$'s form a multiscale tree decomposition of $\widetilde{\mathcal{M}}$. We will prove that $\mathcal{M}\backslash\widetilde{\mathcal{M}}$ has a negligible measure and $\{C_{j,k}\}_{k\in\mathcal{K}_j,j\in[j_{\min},j_{\max}]}$ satisfies Assumptions (A1-A5). The key is that every $C_{j,k}$ is contained in a ball of radius $3\cdot 2^{-j}$ and also contains a ball of radius $2^{-j}/4$.

Lemma 26 *Every $C_{j,k}$ constructed in Algorithm 3 satisfies $B_{\frac{2^{-j}}{4}}(a_{j,k})\subseteq C_{j,k}\subseteq B_{3\cdot 2^{-j}}(a_{j,k})$*

Proof For any $x\in\mathbb{R}^D$ and any set $C\in\mathbb{R}^D$, the diameter of C with respect to x is defined as $\text{diam}(C,x) := \sup_{z\in C}\|z-x\|$. First, we prove that, for every j, $C_{j,k_1}\cap C_{j,k_2}=\emptyset$ whenever $k_1\neq k_2$. Take any $a_{j+1,k_1'}\in C_{j,k_1}$ and $a_{j+1,k_2'}\in C_{j,k_2}$. Our construction guarantees that

$$\text{diam}(C_{j+1,k_1'}, a_{j+1,k_1'}) \le \tfrac{1}{4}2^{-(j+1)} + \tfrac{1}{4}2^{-(j+2)} + \ldots < \tfrac{1}{2}2^{-(j+1)}$$

and similarly for $\text{diam}(C_{j+1,k_2'}, a_{j+1,k_2'})$. Since $\|a_{j+1,k_1'} - a_{j+1,k_2'}\| \ge 2^{-(j+1)}$, this implies that $C_{j+1,k_1'}\cap C_{j+1,k_2'}=\emptyset$. In our construction,

$$C_{j,k_1} = \Big(\bigcup_{a_{j+1,k_1'}\in C_{j,k_1}} C_{j+1,k_1'}\Big)\bigcup B_{\frac{2^{-j}}{4}}(a_{j,k_1}),\ C_{j,k_2} = \Big(\bigcup_{a_{j+1,k_2'}\in C_{j,k_2}} C_{j+1,k_2'}\Big)\bigcup B_{\frac{2^{-j}}{4}}(a_{j,k_2}).$$

Since $\|a_{j,k_1} - a_{j,k_2}\| \ge 2^{-j}$, we observe that $B_{\frac{1}{4}2^{-j}}(a_{j,k_1})\cap B_{\frac{1}{4}2^{-j}}(a_{j,k_2}) = \emptyset$, $C_{j+1,k_1'}\cap B_{\frac{1}{4}2^{-j}}(a_{j,k_2}) = \emptyset$ for every $a_{j+1,k_1'}\in C_{j,k_1}$, and $C_{j+1,k_2'}\cap B_{\frac{1}{4}2^{-j}}(a_{j,k_1}) = \emptyset$ for every $a_{j+1,k_2'}\in C_{j,k_2}$. Therefore $C_{j,k_1}\cap C_{j,k_2}=\emptyset$.

Our construction of $C_{j,k}$'s guarantees that every $C_{j,k}$ contains a ball of radius $\frac{1}{4}\cdot 2^{-j}$. Next we prove that every $C_{j,k}$ is contained in a ball of radius $3\cdot 2^{-j}$. The cover tree structure guarantees that $\mathcal{X}_n'\subset\cup_{a_{j,k}\in T_j(\mathcal{X}_n')}B_{2\cdot 2^{-j}}(a_{j,k})$ for every j. Hence, for every $a_{j,k}$ and every $a_{j+1,k'}\in C_{j,k}$, we obtain $\|a_{j+1,k'} - a_{j,k}\|\le 2\cdot 2^{-j}$ and the computation above yields $\text{diam}(C_{j+1,k'}, a_{j+1,k'})\le 2^{-j}/4$, and therefore $\text{diam}(C_{j,k}, a_{j,k})\le 2\cdot 2^{-j} + 2^{-j}/4 \le 3\cdot 2^{-j}$. In summary $C_{j,k}$ is contained in the ball of radius $3\cdot 2^{-j}$ centered at $a_{j,k}$. ■

The following Lemma will be useful when comparing comparing covariances of sets:

Lemma 27 *If $B\subseteq A$, then we have $\lambda_d(\text{cov}(\rho|_A))\ge\frac{\rho(B)}{\rho(A)}\lambda_d(\text{cov}(\rho|_B))$.*

Proof Without loss of generality, we assume both A and B are centered at x_0. Let V be the eigenspace associated with the largest d eigenvalues of $\text{cov}(\rho|_B)$. Then

$$\begin{aligned}
\lambda_d(\text{cov}(\rho|_A)) &= \max_{\dim U=d}\min_{u\in U}\frac{u^T\text{cov}(\rho|_A)u}{u^Tu} \ge \min_{v\in V}\frac{v^T\text{cov}(\rho|_A)v}{v^Tv}\\
&\ge \min_{v\in V}\frac{v^T\left(\int_A(x-x_0)(x-x_0)^Td\rho\right)v}{\rho(A)v^Tv}\\
&= \min_{v\in V}\left(\frac{v^T\left(\int_B(x-x_0)(x-x_0)^Td\rho\right)v}{\rho(A)v^Tv} + \frac{v^T\left(\int_{A\backslash B}(x-x_0)(x-x_0)^Td\rho\right)v}{\rho(A)v^Tv}\right)\\
&\ge \min_{v\in V}\frac{v^T\left(\int_B(x-x_0)(x-x_0)^Td\rho\right)v}{\rho(A)v^Tv} = \frac{\rho(B)}{\rho(A)}\lambda_d(\text{cov}(\rho|_B)).
\end{aligned}$$

■

A.2. Regularity of geometric spaces

To fix the ideas, consider the case where $\mathcal{M}$ is a manifold of class $\mathcal{C}^s$, $s\in\mathbb{R}^+\setminus\mathbb{Z}$, i.e. around every point x_0 there is a neighborhood U_{x_0} that is parametrized by a function $f: V\to U_{x_0}$, where V is an open connected set of $\mathbb{R}^d$, and $f\in\mathcal{C}^s$, i.e. f is $\lfloor s\rfloor$ times continuously differentiable and the $\lfloor s\rfloor$-th derivative $f^{\lfloor s\rfloor}$ is Hölder continuous of order $s-\lfloor s\rfloor$, i.e. $||f^{\lfloor s\rfloor}(x) - f^{\lfloor s\rfloor}(y)||\le||f^{\lfloor s\rfloor}||_{\mathcal{C}^{s-\lfloor s\rfloor}}||x-y||^{s-\lfloor s\rfloor}$. In particular, for $s\in(0,1)$, f is simply a Hölder function of order s. Without loss of generality, up to a (linear) change of coordinates we may assume $x = f(x^d)$ where $x^d\in V$.

If $\mathcal{M}$ is a manifold of class $\mathcal{C}^s, s \in (0,1)$, a constant approximation of f on a set I by the value $x_0 := f(x_0^d)$ on such set yields

$$\frac{1}{\rho(I)}\int_I |f(x^d) - f(x_0^d)|^2 d\rho(x) \le \frac{1}{\rho(I)}\int_I ||x^d - x_0^d||^{2s}||f||^2_{\mathcal{C}^s} d\rho(x) \le ||f||^2_{\mathcal{C}^s}\mathrm{diam}(I)^{2s}$$

where we used continuity of f. If I was a ball, we would obtain a bound which would be better by a multiplicative constant no larger than $1/d$. Moreover, the left hand side is minimized by the mean $\frac{1}{\rho(I)}\int_I f(y)d\rho(y)$ of f on I, and so the bound on the right hand side holds a fortiori by replacing $f(x_0^d)$ by the mean.

Next we consider the linear approximation of $\mathcal{M}$ on $I \subset \mathcal{M}$. Suppose there exits θ_0, θ_2 such that I is contained in a ball of radius $\theta_2 r$ and contains a ball of radius $\theta_0 r$. Let $x_0 \in I$ be the closest point on I to the mean. Then I is the graph of a $\mathcal{C}^s$ function $f\colon P_{T_{x_0}(I)} \to P_{T^\perp_{x_0}(I)}$ where $T_{x_0}(I)$ is the plane tangent to I at x_0 and $T^\perp_{x_0}(I)$ is the orthogonal complement of $T_{x_0}(I)$. Since all the quantities involved are invariant under rotations and translations, up to a change of coordinates, we may assume $x^d = (x_1, \dots, x_d)$ and $f = (f_1, \dots, f_{D-d})$ where $f_i := f_i(x^d)$, $i = d+1, \dots, D$. A linear approximation of $f = (f_{d+1}, \dots, f_D)$ based on Taylor expansion and an application of the mean value theorem yields the error estimates.

- Case 1: $s \in (1,2)$

$$\begin{aligned}
&\frac{1}{\rho(I)}\int_I \left\| f(x^d) - f(x_0^d) - \nabla f(x_0^d)\cdot(x^d - x_0^d)\right\|^2 d\rho \\
&= \sum_{i=d+1}^D \frac{1}{\rho(I)} \sup_{\xi_i \in \mathrm{domain}(f_i)} \int_I \left|\nabla f_i(\xi_i)(x^d - x_0^d) - \nabla f_i(x_0^d)\cdot(x^d - x_0^d)\right|^2 d\rho \\
&\le \sum_{i=d+1}^D \frac{1}{\rho(I)} \sup_{\xi_i \in \mathrm{domain}(f_i)} \int_{C_{j,k}} ||x^d - x_0^d||^2 ||\xi_i - x_0^d||^{2(s-\lfloor s\rfloor)} ||\nabla f_i||^2_{\mathcal{C}^{s-\lfloor s\rfloor}} d\rho \\
&\le D \max_{i=1,\dots,D-d} ||\nabla f_i||^2_{\mathcal{C}^{s-\lfloor s\rfloor}} \mathrm{diam}(I)^{2s}.
\end{aligned}$$

- Case 2: $s = 2$

$$\begin{aligned}
&\frac{1}{\rho(I)}\int_I \left\| f(x^d) - f(x_0^d) - \nabla f(x_0^d)\cdot(x^d - x_0^d)\right\|^2 d\rho \\
&= \sum_{i=d+1}^D \frac{1}{\rho(I)} \int_I \left\| f_i(x^d) - f_i(x_0^d) - \nabla f_i(x_0^d)\cdot(x^d - x_0^d)\right\|^2 d\rho \\
&\le \sum_{i=d+1}^D \frac{1}{\rho(I)} \sup_{\xi_i \in \mathrm{domain}(f_i)} \int_I \left\| \frac{1}{2}(\xi_i - x_0^d)^T D^2 f_i|_{x_0^d}(\xi_i - x_0^d) + o(\|\xi_i - x_0^d\|^2)\right\|^2 d\rho \\
&\le \frac{D}{2} \max_{i=1,\dots,D-d} \|D^2 f_i\| \mathrm{diam}(I)^4 + o(2^{-4j}).
\end{aligned}$$

$\mathcal{M}$ does not have boundaries, so the Taylor expansion in the computations above can be performed on the convex hull of $P_{T_{x_0}(I)}$, whose diameter is no larger than $\mathrm{diam}(I)$. Note

that this bound then holds for other linear approximations which are at least as good, in $L^2(\rho|_I)$, as Taylor expansion. One such approximation is, by definition, the linear least square fit of f in $L^2(\rho|_I)$. Let L_I be the least square fit to the function $x \mapsto f(x)$. Then

$$\begin{aligned}
&\sum_{i=d+1}^D \lambda_i(\mathrm{cov}(\rho|_I))^2 = \frac{1}{\rho(I)}\int_I ||f(x) - L_I(x)||^2 d\rho(x) \\
&\le \begin{cases} D\max_{i=1,\dots,D-d} ||\nabla f_i||^2_{\mathcal{C}^{s-\lfloor s\rfloor}} \mathrm{diam}(I)^{2s}, & s \in (1,2) \\ \frac{D}{2}\max_{i=1,\dots,D-d} \|D^2 f_i\| \mathrm{diam}(I)^4, & s = 2 \end{cases}.
\end{aligned} \tag{32}$$

Proof [Proof of Proposition 14] Claim (A1) follows by a simple volume argument: $C_{j,k}$ is contained in a ball of radius $3\cdot 2^{-j}$, and therefore has volume at most $C_1(3\cdot 2^{-j})^d$, and each child contains a ball of radius $2^{-(j+1)}/4$, and therefore volume at least $C_1^{-1}(2^{-(j+1)}/4)^d$. It follows that $a_{\max} \le C_1^2(3\cdot 2^{-j}/2^{-(j+1)}\cdot 4)^d$. Clearly $a_{\min} \ge 1$ since every $a_{j,k}$ belongs to both $T_j(\mathcal{X}_n')$ and $T_{j'}(\mathcal{X}_n')$ with $j' \ge j$. (A1),(A3), (A4) are straightforward consequences of the doubling assumption and Lemma 26. As for (A2), for any $\nu > 0$, we have

$$\begin{aligned}
&\mathbb{P}\left\{\rho(\mathcal{M}\setminus\widetilde{\mathcal{M}}) > \frac{28\nu\log n}{3n}\right\} = \mathbb{P}\left\{\widehat{\rho}(\mathcal{M}\setminus\widetilde{\mathcal{M}}) = 0 \text{ and } \rho(\mathcal{M}\setminus\widetilde{\mathcal{M}}) > \frac{28\nu\log n}{3n}\right\} \\
&\le \mathbb{P}\left\{|\widehat{\rho}(\mathcal{M}\setminus\widetilde{\mathcal{M}}) - \rho(\mathcal{M}\setminus\widetilde{\mathcal{M}})| > \frac{1}{2}\rho(\mathcal{M}\setminus\widetilde{\mathcal{M}}) \text{ and } \rho(\mathcal{M}\setminus\widetilde{\mathcal{M}}) > \frac{28\nu\log n}{3n}\right\} \\
&\le 2e^{-\frac{3}{28}n\rho(\mathcal{M}\setminus\widetilde{\mathcal{M}})} \le 2n^{-\nu}.
\end{aligned}$$

In order to prove the last statement about property (A5) in the case of 5a, observe that $B_{2^{-j}/4}(a_{j,k}) \subseteq C_{j,k} \subseteq B_{3\cdot 2^{-j}}(a_{j,k})$. By Lemma 27 we have

$$\frac{C_1^{-1}(2^{-j}/4)^d}{\rho(C_{j,k})}\lambda_d(\mathrm{cov}(\rho|_{B_{2^{-j}/4}(a_{j,k})}) \le \lambda_d(\mathrm{cov}(\rho|_{C_{j,k}}) \le \frac{C_1(3\cdot 2^{-j})^d}{\rho(C_{j,k})}\lambda_d(\mathrm{cov}(\rho|_{B_{3\cdot 2^{-j}}(a_{j,k})})$$

and therefore $\lambda_d(\mathrm{cov}(\rho|_{C_{j,k}}) \ge C_1^{-2}(1/12)^d\lambda_d(\mathrm{cov}(\rho|_{B_{2^{-j}/4}(a_{j,k})}) \ge C_1^{-2}(1/12)^d\tilde{\theta}_3(2^{-j}/4)^2/d$, so that (A5)-(i) holds with $\theta_3 = \tilde{\theta}_3(4C_1)^{-2}(1/12)^d$. Proceeding similarly for λ_{d+1}, we obtain from the upper bound above that

$$\lambda_{d+1}(\mathrm{cov}(\rho|_{C_{j,k}}) \le \frac{C_1(3\cdot 2^{-j})^d}{C_1^{-1}(2^{-j}/4)^d}\lambda_{d+1}(\mathrm{cov}(\rho|_{B_{3\cdot 2^{-j}}(a_{j,k})}) \le (12^d)^2\cdot 144C_1^4\tilde{\theta}_4/\tilde{\theta}_3\lambda_d(\mathrm{cov}(\rho|_{C_{j,k}})$$

so that (A5)-(ii) holds with $\theta_4 = (12^d)^2\cdot 144C_1^4\tilde{\theta}_4/\tilde{\theta}_3$.

In order to prove (A5) in the case of 5b, we use calculations as in Little et al. (2017); Maggioni et al. (2016) where one obtains that the first d eigenvalues of the covariance matrix of $\rho|_{B_r(z)}$ with $z \in \mathcal{M}$, is lower bounded by $\tilde{\theta}_3 r^2/d$ for some $\tilde{\theta}_3 > 0$. Then (A5)-(i) holds for $C_{j,k}$ with $\theta_3 = \tilde{\theta}_3(4C_1)^{-2}(1/12)^d$. The estimate of $\lambda_{d+1}(\mathrm{cov}(\rho|_{C_{j,k}}))$ follows from (32) such that

$$\sum_{i=d+1}^D \lambda_i(\mathrm{cov}(\rho|_{C_{j,k}}))^2 \le \begin{cases} D\max_{i=1,\dots,D-d} ||\nabla f_i||^2_{\mathcal{C}^{s-\lfloor s\rfloor}}(6\cdot 2^{-j})^{2s}, & s \in (1,2) \\ \frac{D}{2}\max_{i=1,\dots,D-d} \|D^2 f_i\|(6\cdot 2^{-j})^4, & s = 2 \end{cases}.$$

Therefore, there exists j_0 such that $\lambda_{d+1}(\mathrm{cov}(\rho|_{C_{j,k}})) < \theta_4 \lambda_d(\mathrm{cov}(\rho|_{C_{j,k}}))$ when $j \geq j_0$. The calculation above also implies that $\rho \in \mathcal{A}_s^\infty$ if $\max_{i=1,\ldots,D-d} ||\nabla f_i||^2_{\mathcal{C}^{s-\lfloor s \rfloor}}$ for $s \in (1,2)$ or $\max_{i=1,\ldots,D-d} \|D^2 f_i\|$ for $s = 2$ is uniformly upper bounded.

■

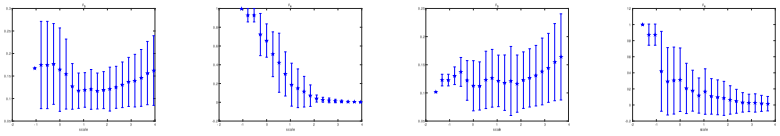
A.3. An alternative tree construction method

The $\{C_{j,k}\}$ constructed by Algorithm 3 is proved to satisfy Assumptions (A1-A5). In numerical experiments, we use a much simpler algorithm to construct $\{C_{j,k}\}$ as follows:

$$C_{j_{\max},k} = \mathrm{Voronoi}(a_{j_{\max},k}, T_{j_{\max}}(\mathcal{X}_n')) \cap B_{2^{-j_{\max}}}(a_{j_{\max},k}),$$

and for any $j < j_{\max}$, we define $C_{j,k} = \bigcup_{a_{j-1,k'} \text{ child of } a_{j,k}} C_{j-1,k'}$.

We observe that the vast majority of $C_{j,k}$'s constructed above satisfy Assumptions (A1-A5) in our numerical experiments. While it is not difficult to construct counterexamples in which the $C_{j,k}$'s thus construct fail to satisfy Assumptions (A1-A5). In Fig. 14, we will show that (A5) is satisfied when we experiment on volume measures on the 3-dim S and Z-manifold. Here we sample 10^5 training data, perform multiscale tree decomposition as stated above, and compute $\theta_3^{j,k}, \theta_4^{j,k}$ at every $C_{j,k}$. In Fig. 14, we display the mean of $\{\theta_3^{j,k}\}_{k \in \mathcal{K}_j}$ or $\{\theta_4^{j,k}\}_{k \in \mathcal{K}_j}$ versus scale j, with a vertical error bar representing the standard deviation of $\{\theta_3^{j,k}\}_{k \in \mathcal{K}_j}$ or $\{\theta_4^{j,k}\}_{k \in \mathcal{K}_j}$ at each scale. We observe that $\theta_3 = \min_{j,k} \theta_3^{j,k} \geq 0.05$ at all scales and $\theta_4 = \max_{j,k} \theta_4^{j,k} \leq 1/2$ except at very coarse scales, which demonstrates Assumption (A5) is satisfied here. Indeed θ_4 is not only bounded, but also decreases from coarse scales to fine scales.



(a) $\{\theta_3^{j,k}\}_{k \in \mathcal{K}_j}$ versus j of the 3-dim S-manifold (b) $\{\theta_4^{j,k}\}_{k \in \mathcal{K}_j}$ versus j of the 3-dim S-manifold (c) $\{\theta_3^{j,k}\}_{k \in \mathcal{K}_j}$ versus j of the 3-dim Z-manifold (d) $\{\theta_4^{j,k}\}_{k \in \mathcal{K}_j}$ versus j of the 3-dim Z-manifold

Figure 14: The mean of $\{\theta_3^{j,k}\}_{k \in \mathcal{K}_j}$ or $\{\theta_4^{j,k}\}_{k \in \mathcal{K}_j}$ versus scale j, with a vertical error bar representing the standard deviation of $\{\theta_3^{j,k}\}_{k \in \mathcal{K}_j}$ or $\{\theta_4^{j,k}\}_{k \in \mathcal{K}_j}$ at each scale.

We observe that, every $C_{j,k}$ constructed above is contained in a ball of radius $\theta_2 2^{-j}$ and contains a ball of radius $\theta_0 2^{-j}$, with $\theta_2/\theta_0 \in [1,2]$ for the majority of $C_{j,k}$'s. In Fig. 15, we take the volume measures on the 3-dim S and Z-manifold, and plot $\log_2$ of the outer-radius

and the statistics a lower bound for the in-radius[3] versus the scale of cover tree. Notice that the in-radius is a fraction of the outer-radius at all scales, and $\log_2 \theta_2 - \log_2 \theta_0 \leq 1$ for the majority of cells.

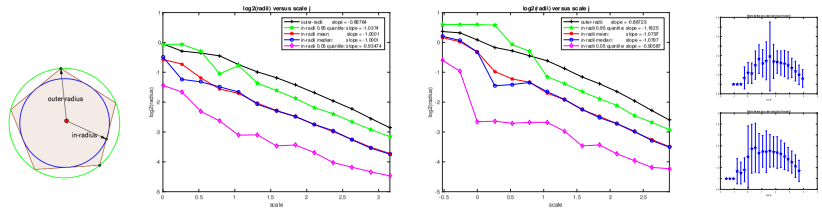


Figure 15: From left to right: the in-radius and outer-radius of a pentagon; $\log_2$ of the outer-radius and the statistics of the in-radius versus the scale of cover tree for the 3-dim S-manifold, and then the same plot for the 3-dim Z-manifold; ratio between outer-radii and in-radii, for the 3-dim S-manifold (top) and the 3-dim Z-manifold (bottom).

A.4. $\mathcal{A}_s^\infty \subset \mathcal{B}_s$

Proof [proof of Lemma 6] Assume $\rho(C_{j,k}) \asymp 2^{-jd}$. According to Definition 2, $\rho \in \mathcal{A}_s^\infty$ if $\|(X - \mathcal{P}_{j,k}X)\mathbf{1}_{j,k}\| \leq |\rho|_{\mathcal{A}_s^\infty} 2^{-js} \sqrt{\rho(C_{j,k})}$, $\forall k \in \mathcal{K}_j, j \geq j_{\min}$, which implies $\Delta_{j,k} \leq 2|\rho|_{\mathcal{A}_s^\infty} 2^{-js} \sqrt{\rho(C_{j,k})} \lesssim |\rho|_{\mathcal{A}_s^\infty} 2^{-j(s+\frac{d}{2})}$.

Let $\eta > 0$ and $\mathcal{T}_{(\rho,\eta)}$ be the smallest proper subtree of $\mathcal{T}$ that contains all $C_{j,k}$ for which $\Delta_{j,k} \geq 2^{-j}\eta$. All the nodes satisfying $\Delta_{j,k} \geq 2^{-j}\eta$ will satisfy $|\rho|_{\mathcal{A}_s^\infty} 2^{-j(s+\frac{d}{2})} \gtrsim 2^{-j}\eta$ which implies $2^{-j} \gtrsim (\eta/|\rho|_{\mathcal{A}_s^\infty})^{\frac{2}{2s+d-2}}$. Therefore, the truncated tree $\mathcal{T}_{(\rho,\eta)}$ is contained in $\mathcal{T}_{j^*} = \cup_{j \leq j^*} \Lambda_j$ with $2^{-j^*} \asymp (\eta/|\rho|_{\mathcal{A}_s^\infty})^{\frac{2}{2s+d-2}}$, so the entropy of $\mathcal{T}_{(\rho,\eta)}$ is upper bounded by the entropy of $\mathcal{T}_{j^*}$, which is $\sum_{j \leq j^*} 2^{-2j} \# \Lambda_j \asymp 2^{j^*(d-2)} \asymp (\eta/|\rho|_{\mathcal{A}_s^\infty})^{-\frac{2(d-2)}{2s+d-2}}$. Then $\rho \in \mathcal{B}_s$ and $|\rho|_{\mathcal{B}_s} \lesssim |\rho|_{\mathcal{A}_s^\infty}$ according to Definition 5. ■

Appendix B. S-manifold and Z-manifold

We consider volume measures on the d dimensional S-manifold and Z-manifold whose x_1 and x_2 coordinates are on the S curve and Z curve in Figure 5 (a) and $x_i, i = 3, \ldots, d+1$ are uniformly distributed in $[0,1]$.

3. The in-radius of $C_{j,k}$ is approximately computed as follows: we randomly pick a center, and evaluate the largest radius with which the ball contains at least 95% points from $C_{j,k}$. This procedure is repeated for two centers, and then we pick the maximal radius as an approximation of the in-radius.

B.1. S-manifold

Since S-manifold is smooth and has a bounded curvature, the volume measure on the S-manifold is in $\mathcal{A}_2^\infty$. Therefore, the volume measure on the S-manifold is in $\mathcal{A}_2$ and $\mathcal{B}_2$ when $d \geq 3$.

B.2. Z-manifold

B.2.1. The volume on the Z-manifold is in $\mathcal{A}_{1.5}$

The uniform distribution on the d dimensional Z-manifold is in $\mathcal{A}_1$ at two corners and satisfies $\|(X - \mathcal{P}_{j,k}X)\mathbf{1}_{j,k}\| = 0$ when $C_{j,k}$ is away from the corners. There exists $A_0 > 0$ such that $\|(X - \mathcal{P}_{j,k}X)\mathbf{1}_{j,k}\| \leq A_0 2^{-j}\sqrt{\rho(C_{j,k})}$ when $C_{j,k}$ intersects with the corners. At scale j, there are about 2^{jd} cells away from the corners and there are about $2^{j(d-1)}$ cells which intersect with the corners. As a result,

$$\|X - \mathcal{P}_j X\| \leq \mathcal{O}\left(\sqrt{2^{jd} \cdot 0 \cdot 2^{-jd} + 2^{j(d-1)} \cdot 2^{-2j} \cdot 2^{-jd}}\right) = \mathcal{O}(2^{-1.5j}),$$

so the volume measure on Z-manifold is in $\mathcal{A}_{1.5}$.

B.2.2. Model class $\mathcal{B}_s$

Assume $\rho(C_{j,k}) \asymp 2^{-jd}$. We compute the regularity parameter s in the $\mathcal{B}_s$ model class when $d \geq 3$. It is easy to see that $\Delta_{j,k} = 0$ when $C_{j,k}$ is away from the corners and $\Delta_{j,k} \leq 2A_0 2^{-j}\sqrt{\rho(C_{j,k})} \lesssim 2^{-j(\frac{d}{2}+1)}$ when $C_{j,k}$ intersects with the corners. Given any fixed threshold $\eta > 0$, in the truncated tree $\mathcal{T}_{(\rho,\eta)}$, the parent of the leaves intersecting with the corners satisfy $2^{-j(\frac{d}{2}+1)} \gtrsim 2^{-j}\eta$. In other words, at the corners the tree is truncated at a scale coarser than j^* such that $2^{-j^*} = \mathcal{O}(\eta^{\frac{2}{d}})$. Since $\Delta_{j,k} = 0$ when $C_{j,k}$ is away from the corners, the entropy of $\mathcal{T}_{(\rho,\eta)}$ is dominated by the nodes intersecting with the corners whose cardinality is $2^{j(d-1)}$ at scale j. Therefore

$$\text{Entropy of } \mathcal{T}_{(\rho,\eta)} \lesssim \sum_{j \leq j^*} 2^{-2j} 2^{j(d-1)} = \mathcal{O}\left(\eta^{-\frac{2(d-3)}{d}}\right),$$

which implies that $p \leq \frac{2(d-3)}{d}$ and $s \geq \frac{3(d-2)}{2(d-3)} > 1.5$.

Then we study the relation between the error $\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}}X\|$ and the partition size $\#\Lambda_{(\rho,\eta)}$, which is numerically verified in Figure 4. Since all the nodes in $\mathcal{T}_{(\rho,\eta)}$ that intersect with corners are at a scale coarser than j^*, $\#\Lambda_{(\rho,\eta)} \approx 2^{j^*(d-1)} \asymp \eta^{-\frac{2(d-1)}{d}}$. Therefore, $\eta \lesssim [\#\Lambda_{(\rho,\eta)}]^{-\frac{d}{2(d-1)}}$ and

$$\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}}X\| \lesssim \eta^{\frac{2-p}{2}} = \eta^{\frac{2s}{2s+d-2}} \lesssim [\#\Lambda_{(\rho,\eta)}]^{-\frac{2sd}{2(d-1)(2s+d-2)}} = [\#\Lambda_{(\rho,\eta)}]^{-\frac{3}{2(d-1)}}.$$

Appendix C. Proofs of Lemma 15 and Proposition 16

C.1. Concentration inequalities

We first recall a Bernstein inequality from Tropp (2014) which is an exponential inequality to estimate the spectral norm of a sum independent random Hermitian matrices of size

$D \times D$. It features the dependence on an intrinsic dimension parameter which is usually much smaller than the ambient dimension D. For a positive-semidefinite matrix A, the intrinsic dimension is the quantity

$$\text{intdim}(A) = \frac{\text{trace}(A)}{\|A\|}.$$

Proposition 28 (Theorem 7.3.1 in Tropp (2014)) *Let $\xi_1, \ldots, \xi_n$ be $D \times D$ independent random Hermitian matrices that satisfy*

$$\mathbb{E}\xi_i = 0 \text{ and } \|\xi_i\| \leq R, \ i = 1, \ldots, n.$$

Form the mean $\xi = \frac{1}{n}\sum_{i=1}^n \xi_i$. Suppose $\mathbb{E}(\xi^2) \preceq \Phi$. Introduce the intrinsic dimension parameter $d_{\text{in}} = \text{intdim}(\Phi)$. Then, for $nt \geq n\|\Phi\|^{1/2} + R/3$,

$$\mathbb{P}\{\|\xi\| \geq t\} \leq 4 d_{\text{in}} e^{-\frac{nt^2/2}{n\|\Phi\| + Rt/3}}.$$

We use the above inequalities to estimate the deviation of the empirical mean from the mean and the deviation of the empirical covariance matrix from the covariance matrix when the data $\mathcal{X}_{j,k} = \{x_1, \ldots, x_n\}$ (with a slight abuse of notations) are i.i.d. samples from the distribution $\rho_{|C_{j,k}}$.

Lemma 29 *Suppose $x_1, \ldots, x_n$ are i.i.d. samples from $\rho_{|C_{j,k}}$. Let the local mean and covariance, and their empirical counterparts, be defined as*

$$c_{j,k} = \int_{C_{j,k}} x d\rho_{|C_{j,k}} \quad , \quad \widehat{c}_{j,k} := \frac{1}{n}\sum_{i=1}^n x_i$$

$$\Sigma_{j,k} = \int_{C_{j,k}} (x - c_{j,k})(x - c_{j,k})^T d\rho_{|C_{j,k}} \quad , \quad \widehat{\Sigma}_{j,k} := \frac{1}{n}\sum_{i=1}^n (x_i - \widehat{c}_{j,k})(x_i - \widehat{c}_{j,k})^T$$

Then

$$\mathbb{P}\{\|\widehat{c}_{j,k} - c_{j,k}\| \geq t\} \leq 8e^{-\frac{3nt^2}{6\theta_2^2 2^{-2j} + 2\theta_2 2^{-j} t}}, \tag{33}$$

$$\mathbb{P}\{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \geq t\} \leq \left(\frac{4\theta_2^2}{\theta_3} d + 8\right) e^{-\frac{3nt^2}{24\theta_2^4 2^{-4j} + 8\theta_2^2 2^{-2j} t}}. \tag{34}$$

Proof We start by proving (33). We will apply Bernstein inequality with $\xi_i = x_i - c_{j,k} \in \mathbb{R}^D$. Clearly $\mathbb{E}\xi_i = 0$, and $\|\xi_i\| \leq \theta_2 2^{-j}$ due to Assumption (A4). We form the mean $\xi = \frac{1}{n}\sum_{i=1}^n \xi_i = \widehat{c}_{j,k} - c_{j,k}$ and compute the variance

$$\sigma^2 = n^2\|\mathbb{E}\xi^T\xi\| = \left\|\mathbb{E}\left(\sum_{i=1}^n x_i - c_{j,k}\right)^T \left(\sum_{i=1}^n x_i - c_{j,k}\right)\right\| = \left\|\sum_{i=1}^n \mathbb{E}(x_i - c_{j,k})^T(x_i - c_{j,k})\right\| \leq n\theta_2^2 2^{-2j}.$$

Then for $nt \geq \sigma + \theta_2 2^{-j}/3$,

$$\mathbb{P}\{\|\widehat{c}_{j,k} - c_{j,k}\| \geq t\} \leq 8e^{-\frac{n^2t^2/3}{\sigma^2 + \theta_2 2^{-j} nt/3}} \leq 8e^{-\frac{3nt^2}{6\theta_2^2 2^{-2j} + 2\theta_2 2^{-j} t}}.$$

We now prove (34). Define the intermediate matrix $\bar{\Sigma}_{j,k} = \frac{1}{n}\sum_{i=1}^{n}(x_i - c_{j,k})(x_i - c_{j,k})^T$. Since $\widehat{\Sigma}_{j,k} - \Sigma_{j,k} = \bar{\Sigma}_{j,k} - \Sigma_{j,k} - (\widehat{c}_{j,k} - c_{j,k})(\widehat{c}_{j,k} - c_{j,k})^T$, we have

$$\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \le \|\bar{\Sigma}_{j,k} - \Sigma_{j,k}\| + \|\widehat{c}_{j,k} - c_{j,k}\|^2 \le \|\bar{\Sigma}_{j,k} - \Sigma_{j,k}\| + \theta_2 2^{-j}\|\widehat{c}_{j,k} - c_{j,k}\|.$$

A sufficient condition for $\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| < t$ is $\|\bar{\Sigma}_{j,k} - \Sigma_{j,k}\| < t/2$ and $\|\widehat{c}_{j,k} - c_{j,k}\| < 2^j t/(2\theta_2)$. We apply Proposition 28 to estimate $\mathbb{P}\{\|\bar{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t/2\}$: let $\xi_i = (x_i - c_{j,k})(x_i - c_{j,k})^T - \Sigma_{j,k} \in \mathbb{R}^{D\times D}$. One can verify that $\mathbb{E}\xi_i = 0$ and $\|\xi_i\| \le 2\theta_2^2 2^{-2j}$. We form the mean $\xi = \frac{1}{n}\sum_{i=1}^{n}\xi_i = \bar{\Sigma}_{j,k} - \Sigma_{j,k}$, and then

$$\mathbb{E}\xi^2 = \mathbb{E}\left(\frac{1}{n^2}\sum_{i=1}^{n}\xi_i\sum_{i=1}^{n}\xi_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\xi_i^2 \preceq \frac{1}{n^2}\sum_{i=1}^{n}\theta_2^2 2^{-2j}\Sigma_{j,k} \preceq \frac{\theta_2^2 2^{-2j}}{n}\Sigma_{j,k},$$

which satisfies $\left\|\frac{\theta_2^2 2^{-2j}}{n}\Sigma_{j,k}\right\| \le \theta_2^4 2^{-4j}/n$. Meanwhile

$$d_{\text{in}} = \text{intdim}(\Sigma_{j,k}) = \frac{\text{trace}(\Sigma_{j,k})}{\|\Sigma_{j,k}\|} \le \frac{\theta_2^2 2^{-2j}}{\theta_3 2^{-2j}/d} = \frac{\theta_2^2}{\theta_3}d.$$

Then, Proposition 28 implies

$$\mathbb{P}\{\|\bar{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t/2\} \le \frac{4\theta_2^2}{\theta_3}d e^{\frac{-nt^2/8}{\theta_2^4 2^{-4j} + \frac{\theta_2^2 2^{-2j}t}{3}}} = \frac{4\theta_2^2}{\theta_3}d e^{\frac{-3nt^2}{24\theta_2^4 2^{-4j} + 8\theta_2^2 2^{-2j}t}}.$$

Combining with (33), we obtain

$$\begin{aligned}\mathbb{P}\{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t\} &\le \mathbb{P}\{\|\bar{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t/2\} + \mathbb{P}\left\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge \frac{2^j t}{2\theta_2}\right\} \\ &\le \left(\frac{4\theta_2^2}{\theta_3}d + 8\right) e^{-\frac{3nt^2}{24\theta_2^4 2^{-4j} + 8\theta_2^2 2^{-2j}t}}.\end{aligned}$$

■

In Lemma 29 data are assumed to be i.i.d. samples from the conditional distribution $\rho_{|C_{j,k}}$. Given $\mathcal{X}_n = \{x_1, \ldots, x_n\}$ which contains i.i.d. samples from ρ, we will show that the empirical measure $\widehat{\rho}(C_{j,k}) = \widehat{n}_{j,k}/n$ is close to $\rho(C_{j,k})$ with high probability.

Lemma 30 *Suppose $x_1, \ldots, x_n$ are i.i.d. samples from ρ. Let $\rho(C_{j,k}) = \int_{C_{j,k}} 1 d\rho$ and $\widehat{\rho}(C_{j,k}) = \widehat{n}_{j,k}/n$ where $\widehat{n}_{j,k}$ is the number of points in $C_{j,k}$. Then*

$$\mathbb{P}\{|\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| \ge t\} \le 2e^{-\frac{3nt^2}{6\rho(C_{j,k}) + 2t}} \tag{35}$$

for all $t \ge 0$. Setting $t = \frac{1}{2}\rho(C_{j,k})$ gives rise to

$$\mathbb{P}\left\{|\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| \ge \frac{1}{2}\rho(C_{j,k})\right\} \le 2e^{-\frac{3}{28}n\rho(C_{j,k})}. \tag{36}$$

Combining Lemma 29 and Lemma 30 gives rise to concentration bounds on $\|\widehat{c}_{j,k} - c_{j,k}\|$ and $\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\|$ where $c_{j,k}$, $\widehat{c}_{j,k}$, $\Sigma_{j,k}$ and $\widehat{\Sigma}_{j,k}$ are the conditional mean, empirical conditional mean, conditional covariance matrix and empirical conditional covariance matrix on $C_{j,k}$, respectively:

Lemma 31 *Suppose $x_1, \ldots, x_n$ are i.i.d. samples from ρ. Define $c_{j,k}, \Sigma_{j,k}$ and $\widehat{c}_{j,k}, \widehat{\Sigma}_{j,k}$ as in Table 1. Then given any $t > 0$,*

$$\mathbb{P}\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge t\} \le 2e^{-\frac{3}{28}n\rho(C_{j,k})} + 8e^{-\frac{3n\rho(C_{j,k})t^2}{12\theta_2^2 2^{-2j} + 4\theta_2 2^{-j}t}}, \tag{37}$$

$$\mathbb{P}\left\{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t\right\} \le 2e^{-\frac{3}{28}n\rho(C_{j,k})} + \left(\frac{4\theta_2^2}{\theta_3}d + 8\right)e^{-\frac{3n\rho(C_{j,k})t^2}{96\theta_2^4 2^{-4j} + 16\theta_2^2 2^{-2j}t}}. \tag{38}$$

Proof The number of samples on $C_{j,k}$ is $\widehat{n}_{j,k} = \sum_{i=1}^{n}\mathbf{1}_{j,k}(x_i)$. Clearly $\mathbb{E}[\widehat{n}_{j,k}] = n\rho(C_{j,k})$. Let $\mathcal{I} \subset \{1, \ldots, n\}$ and $|\mathcal{I}| = s$. Conditionally on the event $A_{\mathcal{I}} := \{x_i \in C_{j,k}$ for $i \in \mathcal{I}$ and $x_i \notin C_{j,k}$ for $i \notin \mathcal{I}\}$, the random variables $\{x_i, i \in \mathcal{I}\}$ are i.i.d. samples from $\rho_{|C_{j,k}}$. According to Lemma 30,

$$\begin{aligned}\mathbb{P}\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge t \mid \widehat{n}_{j,k} = s\} &= \sum_{\substack{\mathcal{I}\subset\{1,\ldots,n\} \\ |\mathcal{I}|=s}} \mathbb{P}\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge t \mid A_{\mathcal{I}}\}\frac{1}{\binom{n}{s}} \\ &= \mathbb{P}\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge t \mid A_{\{1,\ldots,s\}}\} \le 8e^{-\frac{3st^2}{6\theta_2^2 2^{-2j} + 2\theta_2 2^{-j}t}},\end{aligned}$$

and

$$\mathbb{P}\{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t \mid \widehat{n}_{j,k} = s\} \le \left(\frac{4\theta_2^2}{\theta_3}d + 8\right)e^{-\frac{3st^2}{24\theta_2^4 2^{-4j} + 8\theta_2^2 2^{-2j}t}}.$$

Furthermore $|\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| \le \frac{1}{2}\rho(C_{j,k})$ yields $\widehat{n}_{j,k} \ge \frac{1}{2}n\rho(C_{j,k})$ and then

$$\mathbb{P}\left\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge t \;\middle|\; |\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| \le \frac{1}{2}\rho(C_{j,k})\right\} \le 8e^{-\frac{3n\rho(C_{j,k})t^2}{12\theta_2^2 2^{-2j} + 4\theta_2 2^{-j}t}}, \tag{39}$$

$$\mathbb{P}\left\{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t \;\middle|\; |\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| \le \frac{1}{2}\rho(C_{j,k})\right\} \le \left(\frac{4\theta_2^2}{\theta_3}d + 8\right)e^{-\frac{3n\rho(C_{j,k})t^2}{48\theta_2^4 2^{-4j} + 16\theta_2^2 2^{-2j}t}}. \tag{40}$$

Eq. (39) (40) along with Lemma 30 gives rise to

$$\mathbb{P}\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge t\} \le 2e^{-\frac{3}{28}n\rho(C_{j,k})} + 8e^{-\frac{3n\rho(C_{j,k})t^2}{12\theta_2^2 2^{-2j} + 4\theta_2 2^{-j}t}},$$

$$\mathbb{P}\left\{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge t\right\} \le 2e^{-\frac{3}{28}n\rho(C_{j,k})} + \left(\frac{4\theta_2^2}{\theta_3}d + 8\right)e^{-\frac{3n\rho(C_{j,k})t^2}{48\theta_2^4 2^{-4j} + 16\theta_2^2 2^{-2j}t}}.$$

■

Given $\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\|$, we can estimate the angle between the eigenspace of $\widehat{\Sigma}_{j,k}$ and $\Sigma_{j,k}$ with the following proposition.

Proposition 32 (Davis and Kahan (1970) or Theorem 3 in Zwald and Blanchard (2006)) *Let $\delta_d(\Sigma_{j,k}) = \frac{1}{2}(\lambda_d^{j,k} - \lambda_{d+1}^{j,k})$. If $\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \le \frac{1}{2}\delta_d(\Sigma_{j,k})$, then*

$$\left\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\right\| \le \frac{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\|}{\delta_d(\Sigma_{j,k})}.$$

According to Assumption (A4) and (A5), $\delta_d(\Sigma_{j,k}) \ge \theta_3 2^{-2j}/(4d)$. An application of Proposition 32 yields

$$\begin{aligned}\mathbb{P}\left\{\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\| \ge t\right\} &\le \mathbb{P}\left\{\|\widehat{\Sigma}_{j,k} - \Sigma_{j,k}\| \ge \frac{\theta_3(1-\theta_4)t}{2d2^{2j}}\right\} \\ &\le 2e^{-\frac{3}{28}n\rho(C_{j,k})} + \left(\frac{4\theta_2^2}{\theta_3}d + 8\right) e^{-\frac{3\theta_3^2(1-\theta_4)^2 n\rho(C_{j,k})t^2}{384\theta_2^4 d^2 + 32\theta_2^2\theta_3(1-\theta_4)dt}}. \end{aligned} \tag{41}$$

Proof [Proof of Lemma 15] Since

$$\|\mathcal{P}_\Lambda X - \widehat{\mathcal{P}}_\Lambda X\|^2 = \sum_{C_{j,k}\in\Lambda}\int_{C_{j,k}} \|\mathcal{P}_{j,k}x - \widehat{\mathcal{P}}_{j,k}x\|^2 d\rho = \sum_j \sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \|\mathcal{P}_{j,k}x - \widehat{\mathcal{P}}_{j,k}x\|^2 d\rho,$$

we obtain the estimate

$$\mathbb{P}\left\{\|\mathcal{P}_\Lambda X - \widehat{\mathcal{P}}_\Lambda X\| \ge \eta\right\} \le \sum_j \mathbb{P}\left\{\sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \|\mathcal{P}_{j,k}x - \widehat{\mathcal{P}}_{j,k}x\|^2 d\rho \ge \frac{2^{-2j}\#_j\Lambda\eta^2}{\sum_{j\ge j_{\min}} 2^{-2j}\#_j\Lambda}\right\}. \tag{42}$$

Next we prove that, for any fixed scale j,

$$\mathbb{P}\left\{\sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \|\mathcal{P}_{j,k}x - \widehat{\mathcal{P}}_{j,k}x\|^2 d\rho \ge t^2\right\} \le \alpha\#_j\Lambda e^{-\frac{\beta 2^{2j}nt^2}{\#_j\Lambda}}. \tag{43}$$

Then Lemma 15 is proved by setting $t^2 = 2^{-2j}\#_j\Lambda\eta^2/(\sum_{j\ge 0} 2^{-2j}\#_j\Lambda)$.

The proof of (43) starts with the following calculation:

$$\begin{aligned}&\sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \|\mathcal{P}_{j,k}x - \widehat{\mathcal{P}}_{j,k}x\|^2 d\rho \\ &= \sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \|c_{j,k} + \mathrm{Proj}_{V_{j,k}}(x - c_{j,k}) - \widehat{c}_{j,k} - \mathrm{Proj}_{\widehat{V}_{j,k}}(x - \widehat{c}_{j,k})\|^2 d\rho \\ &\le \sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \|(\mathbb{I} - \mathrm{Proj}_{\widehat{V}_{j,k}})(c_{j,k} - \widehat{c}_{j,k}) + (\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}})(x - c_{j,k})\|^2 d\rho \\ &\le 2\sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \left[\|c_{j,k} - \widehat{c}_{j,k}\|^2 + \|(\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}})(x - c_{j,k})\|^2\right] d\rho \\ &\le 2\sum_{k:C_{j,k}\in\Lambda}\int_{C_{j,k}} \left[\|c_{j,k} - \widehat{c}_{j,k}\|^2 + \theta_2^2 2^{-2j}\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\|^2\right] d\rho\end{aligned}$$

For any fixed j and given $t > 0$, we divide Λ into light cells $\Lambda_{j,t}^-$ and heavy cells $\Lambda_{j,t}^+$, where

$$\Lambda_{j,t}^- := \left\{C_{j,k} \in \Lambda : \rho(C_{j,k}) \le \frac{t^2}{20\theta_2^2 2^{-2j}\#_j\Lambda}\right\} \text{ and } \Lambda_{j,t}^+ := \Lambda \setminus \Lambda_{j,t}^-.$$

Since $\int_{C_{j,k}} \left[\|c_{j,k} - \widehat{c}_{j,k}\|^2 + \theta_2^2 2^{-2j}\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\|^2\right] d\rho \le 5\theta_2^2 2^{-2j}\rho(C_{j,k})$, for light sets we have

$$2\sum_{k:C_{j,k}\in\Lambda_{j,t}^-}\int_{C_{j,k}} \left[\|c_{j,k} - \widehat{c}_{j,k}\|^2 + \theta_2^2 2^{-2j}\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\|^2\right] d\rho \le \frac{t^2}{2}. \tag{44}$$

Next we consider $C_{j,k} \in \Lambda_{j,t}^+$. We have

$$\begin{aligned}&\mathbb{P}\left\{\|\widehat{c}_{j,k} - c_{j,k}\| \ge \frac{t}{\sqrt{8\rho(C_{j,k})\#_j\Lambda}}\right\} \\ &\le 2\exp\left(-\frac{3}{28}n\rho(C_{j,k})\right) + 8e^{-\frac{3n\rho(C_{j,k})\frac{t^2}{8\rho(C_{j,k})\#_j\Lambda}}{12\theta_2^2 2^{-2j} + 4\theta_2 2^{-j}\frac{t}{\sqrt{8\rho(C_{j,k})\#_j\Lambda}}}} \le C_1 e^{-C_2\frac{2^{2j}nt^2}{\#_j\Lambda}},\end{aligned} \tag{45}$$

and

$$\begin{aligned}&\mathbb{P}\left\{\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\| \ge \frac{2^j t}{\theta_2\sqrt{8\rho(C_{j,k})\#_j\Lambda}}\right\} \\ &\le 2e^{-\frac{3}{28}n\rho(C_{j,k})} + \left(\frac{4\theta_2^2}{\theta_3}d + 8\right) e^{-\frac{3\theta_3^2(1-\theta_4)^2 n\rho(C_{j,k})\frac{2^{2j}t^2}{8\theta_2^2\rho(C_{j,k})\#_j\Lambda}}{384\theta_2^4 d^2 + 32\theta_2^2\theta_3(1-\theta_4)d\frac{2^j t}{\theta_2\sqrt{8\rho(C_{j,k})\#_j\Lambda}}}} \le C_3 d e^{-C_4\frac{2^{2j}nt^2}{d^2\#_j\Lambda}}\end{aligned} \tag{46}$$

where positive constants C_1, C_2, C_3, C_4 depend on θ_2 and θ_3. Combining (44), (45) and (46) gives rise to (43) with $\alpha = \max(C_1, C_3)$ and $\beta = \min(C_2, C_4)$.

■

Proof [Proof of Proposition 16] The bound (18) follows directly from Lemma 15 applied to $\Lambda = \Lambda_j$; (19) follows from (18) by integrating the probability over η:

$$\begin{aligned}\mathbb{E}\|\mathcal{P}_j X - \widehat{\mathcal{P}}_j X\|^2 &= \int_0^{+\infty} \eta\mathbb{P}\left\{\|\mathcal{P}_j X - \widehat{\mathcal{P}}_j X\| \ge \eta\right\} d\eta \\ &\le \int_0^{+\infty} \eta\min\left\{1, \alpha d\#\Lambda_j e^{-\frac{\beta 2^{2j}n\eta^2}{d^2\#\Lambda_j}}\right\} d\eta = \int_0^{\eta_0} \eta d\eta + \int_{\eta_0}^{+\infty} \alpha d\eta\#\Lambda_j e^{-\frac{\beta 2^{2j}n\eta^2}{d^2\#\Lambda_j}} d\eta\end{aligned}$$

where $\alpha d\#\Lambda_j e^{-\frac{\beta 2^{2j}n\eta_0^2}{d^2\#\Lambda_j}} = 1$. Then

$$\mathbb{E}\|\mathcal{P}_j X - \widehat{\mathcal{P}}_j X\|^2 = \frac{1}{2}\eta_0^2 + \frac{\alpha}{2\beta}\cdot\frac{\#\Lambda_j^2}{2^{2j}n}e^{-\beta\frac{2^{2j}n\eta_0^2}{\#\Lambda_j}} \le \frac{d^2\#\Lambda_j\log[\alpha d\#\Lambda_j]}{\beta 2^{2j}n}.$$

■

Appendix D. Proof of Eq. (9), Lemma 17, 19, 20, 21

Proof [Proof of Eq. (9)] Let $\Lambda^{+0}_{(\rho,\eta)} = \Lambda_{(\rho,\eta)}$ and $\Lambda^{+n}_{(\rho,\eta)}$ be the partition consisting of the children of $\Lambda^{+(n-1)}_{(\rho,\eta)}$ for $n = 1, 2, \ldots$. Then

$$\begin{aligned}\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}} X\| &= \|\sum_{\ell=0}^{n-1} (\mathcal{P}_{\Lambda^{+\ell}_{(\rho,\eta)}} X - \mathcal{P}_{\Lambda^{+(\ell+1)}_{(\rho,\eta)}} X) + \mathcal{P}_{\Lambda^{+n}(\rho,\eta)} X - X\| \\ &= \|\sum_{\ell=0}^{\infty} (\mathcal{P}_{\Lambda^{+\ell}_{(\rho,\eta)}} X - \mathcal{P}_{\Lambda^{+(\ell+1)}_{(\rho,\eta)}} X) + \lim_{n\to\infty} \mathcal{P}_{\Lambda^{+n}_{(\rho,\eta)}} X - X\| \\ &\le \|\sum_{C_{j,k}\notin\mathcal{T}_{(\rho,\eta)}} \mathcal{Q}_{j,k} X\| + \|\lim_{n\to\infty} \mathcal{P}_{\Lambda^{+n}_{(\rho,\eta)}} X - X\|.\end{aligned}$$

We have $\|\lim_{n\to\infty} \mathcal{P}_{\Lambda^{+n}_{(\rho,\eta)}} X - X\| = 0$ due to Assumption (A4). Therefore,

$$\begin{aligned}&\|X - \mathcal{P}_{\Lambda_{(\rho,\eta)}} X\|^2 \le \|\sum_{C_{j,k}\notin\mathcal{T}_{(\rho,\eta)}} \mathcal{Q}_{j,k} X\|^2 \le \sum_{C_{j,k}\notin\mathcal{T}_{(\rho,\eta)}} B_0 \|\mathcal{Q}_{j,k} X\|^2 = B_0 \sum_{C_{j,k}\notin\mathcal{T}_{(\rho,\eta)}} \Delta_{j,k}^2 \\ &\le B_0 \sum_{\ell\ge 0} \sum_{C_{j,k}\in\mathcal{T}_{(\rho,2^{-(\ell+1)}\eta)}\setminus\mathcal{T}_{(\rho,2^{-\ell}\eta)}} \Delta_{j,k}^2 \le B_0 \sum_{\ell\ge 0}\sum_{j\ge j_{\min}} (2^{-j}2^{-\ell}\eta)^2 \#_j \mathcal{T}_{(\rho,2^{-(\ell+1)}\eta)} \\ &\le B_0 \sum_{\ell\ge 0} 2^{-2\ell}\eta^2 \sum_{j\ge j_{\min}} 2^{-2j} \#_j \mathcal{T}_{(\rho,2^{-(\ell+1)}\eta)} \le B_0 \sum_{\ell\ge 0} 2^{-2\ell}\eta^2 |\rho|^p_{\mathcal{B}_s} [2^{-(\ell+1)}\eta]^{-p} \\ &\le B_0 2^p \left(\sum_{\ell\ge 0} 2^{-\ell(2-p)}\right) |\rho|^p_{\mathcal{B}_s} \eta^{2-p} \le B_{s,d} |\rho|^p_{\mathcal{B}_s} \eta^{2-p}.\end{aligned}$$

■

Proof [of Lemma 17]

$$\begin{aligned}&\left\|(X - \mathcal{P}_{j^*} X)\mathbf{1}_{\left\{C_{j^*,k}:\rho(C_{j^*,k})\le\frac{28(\nu+1)\log n}{3n}\right\}}\right\|^2 \le \sum_{\left\{C_{j^*,k}:\rho(C_{j^*,k})\le\frac{28(\nu+1)\log n}{3n}\right\}} \int_{C_{j^*,k}} \|x - \mathcal{P}_{j^*,k}\|^2 d\rho \\ &\le \#\left\{C_{j^*,k} : \rho(C_{j^*,k}) \le \frac{28(\nu+1)\log n}{3n}\right\} \theta_2^2 2^{-2j^*} \frac{28(\nu+1)\log n}{3n} \\ &\le \frac{28(\nu+1)\theta_2^2}{3\theta_1} 2^{j^*(d-2)} (\log n)/n \le \frac{28(\nu+1)\theta_2^2\mu}{3\theta_1} \left((\log n)/n\right)^2.\end{aligned}$$

For every $C_{j^*,k}$, we have

$$\begin{aligned}&\mathbb{P}\left\{\rho(C_{j^*,k}) > \tfrac{28}{3}(\nu+1)(\log n)/n \text{ and } \widehat{\rho}(C_{j^*,k}) < d/n\right\} \\ &\le \mathbb{P}\left\{|\widehat{\rho}(C_{j^*,k}) - \rho(C_{j^*,k})| > \rho(C_{j^*,k})/2 \text{ and } \rho(C_{j^*,k}) > \tfrac{28}{3}(\nu+1)(\log n)/n\right\} \\ &\qquad \text{for } n \text{ so large that } 14(\nu+1)\log n > 3d \\ &\le 2e^{-\frac{3}{28}n\rho(C_{j^*,k})} \le 2n^{-\nu-1}.\end{aligned}$$

Then

$$\begin{aligned}&\mathbb{P}\left\{\text{each } C_{j^*,k} \text{ satisfying } \rho(C_{j^*,k}) > \tfrac{28}{3}(\nu+1)(\log n)/n \text{ has at most } d \text{ points}\right\} \\ &\le \#\left\{C_{j^*,k} : \rho(C_{j^*,k}) < \tfrac{28}{3}(\nu+1)(\log n)/n\right\} 2n^{-\nu-1} \le \#\Lambda_{j^*} 2n^{-\nu-1} \le 2n^{-\nu}/(\theta_1\mu\log n) < n^{-\nu},\end{aligned}$$

when n is so large that $\theta_1\mu\log n > 2$. ■

Proof [of Lemma 19] Since $\mathcal{T}_{b\tau_n} \subset \mathcal{T}_{(\rho,b\tau_n)}$, $\mathbb{P}\{e_{12} > 0\}$ if and only if there exists $C_{j,k} \in \mathcal{T}_{(\rho,b\tau_n)} \setminus \mathcal{T}_{b\tau_n}$. In other words, $\mathbb{P}\{e_{12} > 0\}$ if and only if there exists $C_{j,k} \in \mathcal{T}_{(\rho,b\tau_n)}$ such that $\widehat{\rho}(C_{j,k}) < d/n$ and $\Delta_{j,k} > 2^{-j} b\tau_n$. Therefore,

$$\begin{aligned}&\mathbb{P}\{e_{12} > 0\} \le \sum_{C_{j,k}\in\mathcal{T}_{(\rho,b\tau_n)}} \mathbb{P}\{\widehat{\rho}(C_{j,k}) < d/n \text{ and } \Delta_{j,k} > 2^{-j} b\tau_n\} \\ &\le \sum_{C_{j,k}\in\mathcal{T}_{(\rho,b\tau_n)}} \mathbb{P}\left\{\widehat{\rho}(C_{j,k}) < d/n \text{ and } \rho(C_{j,k}) > \tfrac{4b^2\tau_n^2}{9\theta_2^2}\right\} \quad \left(\text{since } \Delta_{j,k} \le \tfrac{3}{2}\theta_2 2^{-j}\sqrt{\rho(C_{j,k})}\right) \\ &\le \sum_{C_{j,k}\in\mathcal{T}_{(\rho,b\tau_n)}} \mathbb{P}\left\{|\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| > \rho(C_{j,k})/2 \text{ and } \rho(C_{j,k}) > \tfrac{4b^2\tau_n^2}{9\theta_2^2}\right\} \\ &\qquad (\text{for } n \text{ large enough so that } 2b^2\kappa^2\log n > 9\theta_2^2 d) \\ &\le \sum_{C_{j,k}\in\mathcal{T}_{(\rho,b\tau_n)}} 2e^{-\frac{3}{28}n\cdot\frac{4b^2\kappa^2\log n}{9\theta_2^2 n}} \le 2n^{-\frac{b^2\kappa^2}{21\theta_2^2}} \#\mathcal{T}_{(\rho,b\tau_n)}.\end{aligned}$$

The leaves of $\mathcal{T}_{(\rho,b\tau_n)}$ satisfy $\rho(C_{j,k}) > 4b^2\tau_n^2/(9\theta_2^2)$. Since $\rho(\mathcal{M}) = 1$, there are at most $9\theta_2^2/(4b^2\tau_n^2)$ leaves in $\mathcal{T}_{(\rho,b\tau_n)}$. Meanwhile, since every node in $\mathcal{T}$ has at least $a_{\min}$ children, $\#\mathcal{T}_{(\rho,b\tau_n)} \le 9\theta_2^2 a_{\min}/(4b^2\tau_n^2)$. Then for a fixed but arbitrary $\nu > 0$,

$$\mathbb{P}\{e_{12} > 0\} \le \tfrac{18\theta_2^2 a_{\min}}{4b^2\tau_n^2} n^{-\frac{b^2\kappa^2}{21\theta_2^2}} \le \tfrac{18\theta_2^2 a_{\min}}{4b^2\kappa^2} n^{1-\frac{b^2\kappa^2}{21\theta_2^2}} \le C(\theta_2, a_{\max}, a_{\min}, \kappa) n^{-\nu},$$

if κ is chosen such that $\kappa > \kappa_1$ where $b^2\kappa_1^2/(21\theta_2^2) = \nu + 1$. ■

Proof [of Lemma 20] We first prove (24). Introduce the intermediate variable

$$\bar{\Delta}_{j,k} := \|\mathcal{Q}_{j,k}\|_n = \|(\mathcal{P}_j - \mathcal{P}_{j+1})\mathbf{1}_{j,k} X\|_n$$

and then observe that

$$\begin{aligned}\mathbb{P}\left\{\widehat{\Delta}_{j,k} \le \eta \text{ and } \Delta_{j,k} \ge b\eta\right\} &\le \mathbb{P}\left\{\widehat{\Delta}_{j,k} \le \eta \text{ and } \bar{\Delta}_{j,k} \ge (a_{\max}+2)\eta\right\} \\ &\quad + \mathbb{P}\left\{\bar{\Delta}_{j,k} \le (a_{\max}+2)\eta \text{ and } \Delta_{j,k} \ge (2a_{\max}+5)\eta\right\}. \quad (47)\end{aligned}$$

The bound in Eq. (24) is proved in the following three steps. In Step One, we show that $\Delta_{j,k} \ge b\eta$ implies $\rho(C_{j,k}) \ge \mathcal{O}(2^{2j}\eta^2)$. Then we estimate $\mathbb{P}\left\{\widehat{\Delta}_{j,k} \le \eta \text{ and } \bar{\Delta}_{j,k} \ge (a_{\max}+2)\eta\right\}$ in Step Two and $\mathbb{P}\left\{\bar{\Delta}_{j,k} \le (a_{\max}+2)\eta \text{ and } \bar{\Delta}_{j,k} \ge (2a_{\max}+5)\eta\right\}$ in Step Three.

Step One: Notice that $\Delta_{j,k} \le \frac{3}{2}\theta_2 2^{-j}\sqrt{\rho(C_{j,k})}$. As a result, $\Delta_{j,k} \ge b\eta$ implies

$$\rho(C_{j,k}) \ge \frac{4b^2 2^{2j}\eta^2}{9\theta_2^2}. \tag{48}$$

Step Two:

$$\mathbb{P}\left\{\widehat{\Delta}_{j,k} \le \eta \ \text{ and } \ \bar{\Delta}_{j,k} \ge (a_{\max}+2)\eta\right\} \le \mathbb{P}\left\{|\widehat{\Delta}_{j,k} - \bar{\Delta}_{j,k}| \ge (a_{\max}+1)\eta\right\}. \tag{49}$$

We can write

$$\begin{aligned}
|\widehat{\Delta}_{j,k} - \bar{\Delta}_{j,k}| &\le \left\|(\mathcal{P}_{j,k} - \widehat{\mathcal{P}}_{j,k})\mathbf{1}_{j,k}X\right\|_n + \sum_{C_{j+1,k'}\in\mathscr{C}(C_{j,k})} \left\|(\widehat{\mathcal{P}}_{j+1,k'} - \mathcal{P}_{j+1,k'})\mathbf{1}_{j+1,k'}X\right\|_n \\
&\le \underbrace{\left(\|c_{j,k} - \widehat{c}_{j,k}\| + \theta_2 2^{-j}\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\|\right)\sqrt{\widehat{\rho}(C_{j,k})}}_{e_1} \\
&\quad + \underbrace{\sum_{C_{j+1,k'}\in\mathscr{C}(C_{j,k})} \left(\|c_{j+1,k'} - \widehat{c}_{j+1,k'}\| + \theta_2 2^{-(j+1)}\|\mathrm{Proj}_{V_{j+1,k'}} - \mathrm{Proj}_{\widehat{V}_{j+1,k'}}\|\right)\sqrt{\widehat{\rho}(C_{j+1,k'})}}_{e_2}.
\end{aligned} \tag{50}$$

Term e_1: We will estimate $\mathbb{P}\{e_1 > \eta\}$. Conditional on the event that $\{|\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| \le \frac{1}{2}\rho(C_{j,k})\}$, we have $e_1 \le \frac{3}{2}\left(\|c_{j,k} - \widehat{c}_{j,k}\| + \theta_2 2^{-j}\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\|\right)\sqrt{\rho(C_{j,k})}$. A similar argument to the proof of Lemma 15 along with (48) give rise to

$$\mathbb{P}\left\{\frac{3}{2}\left(\|c_{j,k} - \widehat{c}_{j,k}\| + \theta_2 2^{-j}\|\mathrm{Proj}_{V_{j,k}} - \mathrm{Proj}_{\widehat{V}_{j,k}}\|\right)\sqrt{\rho(C_{j,k})} > \eta\right\} \le \tilde{\gamma}_1 e^{-\tilde{\gamma}_2 2^{2j} n\eta^2}$$

where $\tilde{\gamma}_1 := \tilde{\gamma}_1(\theta_2, \theta_3, d)$ and $\tilde{\gamma}_2 := \tilde{\gamma}_2(\theta_2, \theta_3, \theta_4, d)$; otherwise $\mathbb{P}\left\{|\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| > \frac{1}{2}\rho(C_{j,k})\right\} \le 2e^{-\frac{3}{28}n\rho(C_{j,k})} \le 2e^{-\frac{b^2 2^{2j} n\eta^2}{21\theta_2^2}}$. Therefore

$$\mathbb{P}\{e_1 > \eta\} \le \max(\tilde{\gamma}_1, 2)e^{-\min(\tilde{\gamma}_2, \frac{b^2}{21\theta_2^2})2^{2j}n\eta^2} \tag{51}$$

Term e_2: We will estimate $\mathbb{P}\{e_2 > a_{\max}\eta\}$. Let $\Lambda^- = \left\{C_{j+1,k'} \in \mathscr{C}(C_{j,k}) : \rho(C_{j+1,k'}) \le \frac{2^{2j}\eta^2}{8\theta_2^2}\right\}$ and $\Lambda^+ = \mathscr{C}(C_{j,k}) \setminus \Lambda^-$. For every $C_{j+1,k'} \in \Lambda^-$, when we condition on the event that $\left\{\rho(C_{j+1,k'}) \le \frac{2^{2j}\eta^2}{8\theta_2^2} \text{ and } \widehat{\rho}(C_{j+1,k'}) \le \frac{2^{2j}\eta^2}{4\theta_2^2}\right\}$, we obtain

$$\begin{aligned}
&\sum_{C_{j+1,k'}\in\Lambda^-} \left(\|c_{j+1,k'} - \widehat{c}_{j+1,k'}\| + \theta_2 2^{-(j+1)}\|\mathrm{Proj}_{V_{j+1,k'}} - \mathrm{Proj}_{\widehat{V}_{j+1,k'}}\|\right)\sqrt{\widehat{\rho}(C_{j+1,k'})} \\
&\le \sum_{C_{j+1,k'}\in\Lambda^-} \theta_2 2^{-j}\sqrt{\widehat{\rho}(C_{j,k})} \le a_{\max}\eta/2;
\end{aligned} \tag{52}$$

otherwise,

$$\begin{aligned}
&\mathbb{P}\left\{\rho(C_{j+1,k'}) \le \tfrac{2^{2j}\eta^2}{8\theta_2^2} \text{ and } \widehat{\rho}(C_{j+1,k'}) > \tfrac{2^{2j}\eta^2}{4\theta_2^2}\right\} \\
&\le \mathbb{P}\left\{\rho(C_{j+1,k'}) \le \tfrac{2^{2j}\eta^2}{8\theta_2^2} \ \text{ and } \ |\widehat{\rho}(C_{j+1,k'}) - \rho(C_{j+1,k'})| \ge \tfrac{2^{2j}\eta^2}{8\theta_2^2}\right\} \\
&\le 2e^{-\left(3n\left(\frac{2^{2j}\eta^2}{8\theta_2^2}\right)^2\right)\Big/\left(6\rho(C_{j+1,k'}) + 2\frac{2^{2j}\eta^2}{8\theta_2^2}\right)} \le 2e^{-\frac{3\cdot 2^{2j}n\eta^2}{64\theta_2^2}}.
\end{aligned} \tag{53}$$

For $C_{j+1,k'} \in \Lambda^+$, a similar argument to e_1 gives rise to

$$\begin{aligned}
&\mathbb{P}\left\{\sum_{C_{j+1,k'}\in\Lambda^+} \left(\|c_{j+1,k'} - \widehat{c}_{j+1,k'}\| + \theta_2 2^{-(j+1)}\|\mathrm{Proj}_{V_{j+1,k'}} - \mathrm{Proj}_{\widehat{V}_{j+1,k'}}\|\right)\sqrt{\widehat{\rho}(C_{j+1,k'})} > a_{\max}\eta/2\right\} \\
&\le \sum_{C_{j+1,k'}\in\Lambda^+} \mathbb{P}\left\{\left(\|c_{j+1,k'} - \widehat{c}_{j+1,k'}\| + \theta_2 2^{-(j+1)}\|\mathrm{Proj}_{V_{j+1,k'}} - \mathrm{Proj}_{\widehat{V}_{j+1,k'}}\|\right)\sqrt{\widehat{\rho}(C_{j+1,k'})} \ge \eta/2\right\} \\
&\le \tilde{\gamma}_3 e^{-\tilde{\gamma}_4 2^{2j} n\eta^2}
\end{aligned} \tag{54}$$

where $\tilde{\gamma}_3 := \tilde{\gamma}_3(\theta_2, \theta_3, a_{\max}, d)$ and $\tilde{\gamma}_4 := \tilde{\gamma}_4(\theta_2, \theta_3, \theta_4, a_{\max}, d)$.

Finally combining (49), (50), (51), (52), (53) and (54) yields

$$\begin{aligned}
\mathbb{P}\left\{\widehat{\Delta}_{j,k} \le \eta \ \text{ and } \ \bar{\Delta}_{j,k} \ge (a_{\max}+2)\eta\right\} &\le \mathbb{P}\left\{|\widehat{\Delta}_{j,k} - \bar{\Delta}_{j,k}| \ge (a_{\max}+1)\eta\right\} \\
&\le \mathbb{P}\{e_1 > \eta\} + \mathbb{P}\{e_2 > a_{\max}\eta\} \le \tilde{\gamma}_5 e^{-\tilde{\gamma}_6 2^{2j} n\eta^2}
\end{aligned} \tag{55}$$

for some constants $\tilde{\gamma}_5 := \tilde{\gamma}_5(\theta_2, \theta_3, a_{\max}, d)$ and $\tilde{\gamma}_6 := \tilde{\gamma}_6(\theta_2, \theta_3, \theta_4, a_{\max}, d)$.

Step Three: The probability $\mathbb{P}\left\{\bar{\Delta}_{j,k} \le (a_{\max}+2)\eta \ \text{ and } \ \Delta_{j,k} \ge (2a_{\max}+5)\eta\right\}$ is estimated as follows. For a fixed $C_{j,k}$, we define the function

$$f(x) = \|(\mathcal{P}_j - \mathcal{P}_{j+1})\,\mathbf{1}_{j,k}x\|, \ x \in \mathcal{M}.$$

Observe that $|f(x)| \le \frac{3}{2}\theta_2 2^{-j}$ for any $x \in \mathcal{M}$. We define $\|f\|^2 = \int_{\mathcal{M}} f^2(x)d\rho$ and $\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^n f^2(x_i)$. Then

$$\begin{aligned}
&\mathbb{P}\left\{\bar{\Delta}_{j,k} \le (a_{\max}+2)\eta \ \text{ and } \ \Delta_{j,k} \ge (2a_{\max}+5)\eta\right\} \\
&\le \mathbb{P}\left\{\Delta_{j,k} - 2\bar{\Delta}_{j,k} \ge \eta\right\} = \mathbb{P}\left\{\|f\| - 2\|f\|_n \ge \eta\right\} \le 3e^{-\frac{2^{2j}n\eta^2}{648\theta_2^2}}
\end{aligned} \tag{56}$$

where the last inequality follows from Györfi et al. (2002, Theorem 11.2). Combining (47), (55) and (56) yields (24).

Next we turn to the bound in Eq. (24), which corresponds to the case that $\Delta_{j,k} \le \eta$ and $\widehat{\Delta}_{j,k} \ge b\eta$. In this case we have $\widehat{\Delta}_{j,k} \le \frac{3}{2}\theta_2 2^{-j}\sqrt{\widehat{\rho}(C_{j,k})}$ which implies

$$\widehat{\rho}(C_{j,k}) \ge \frac{4b^2 2^{2j}\eta^2}{9\theta_2^2}, \tag{57}$$

instead of (48). We shall use the fact that $\rho(C_{j,k}) \geq (2b^2 2^{2j}\eta^2)/(9\theta_2^2)$ given (57) with high probability, by writing

$$\begin{aligned}\mathbb{P}\left\{\Delta_{j,k} \leq \eta \text{ and } \widehat{\Delta}_{j,k} \geq b\eta\right\} \leq & \mathbb{P}\left\{\Delta_{j,k} \leq \eta \text{ and } \widehat{\Delta}_{j,k} \geq b\eta \,\middle|\, \rho(C_{j,k}) \geq \tfrac{2b^2 2^{2j}\eta^2}{9\theta_2^2}\right\} \\ & + \mathbb{P}\left\{\rho(C_{j,k}) \leq \tfrac{2b^2 2^{2j}\eta^2}{9\theta_2^2} \text{ and } \widehat{\rho}(C_{j,k}) \geq \tfrac{4b^2 2^{2j}\eta^2}{9\theta_2^2}\right\} \qquad (58)\end{aligned}$$

where the first term is estimated as above and the second one is estimated through Eq. (35) in Lemma 30:

$$\begin{aligned}& \mathbb{P}\left\{\rho(C_{j,k}) \leq \tfrac{2b^2 2^{2j}\eta^2}{9\theta_2^2} \text{ and } \widehat{\rho}(C_{j,k}) \geq \tfrac{4b^2 2^{2j}\eta^2}{9\theta_2^2}\right\} \\ \leq & \mathbb{P}\left\{\rho(C_{j,k}) \leq \frac{2b^2 2^{2j}\eta^2}{9\theta_2^2} \text{ and } |\widehat{\rho}(C_{j,k}) - \rho(C_{j,k})| \geq \frac{2b^2 2^{2j}\eta^2}{9\theta_2^2}\right\} \\ \leq & 2e^{-\left(3n(\frac{2b^2 2^{2j}\eta^2}{9\theta_2^2})^2\right)\Big/\left(6\rho(C_{j,k})+2\frac{2b^2 2^{2j}\eta^2}{9\theta_2^2}\right)} \leq 2e^{-\frac{3b^2 2^{2j} n\eta^2}{36\theta_2^2}}.\end{aligned}$$

Using the estimate in (58), we obtain the bound (24) which concludes the proof. ■

Proof [Proof of Lemma 21] We will show how Lemma 20 implies Eq. (25). Clearly $e_2 = 0$ if $\widehat{\Lambda}_{\tau_n} \vee \Lambda_{b\tau_n} = \widehat{\Lambda}_{\tau_n} \wedge \Lambda_{\tau_n/b}$, or equivalently $\widehat{\mathcal{T}}_{\tau_n} \cup \mathcal{T}_{b\tau_n} = \widehat{\mathcal{T}}_{\tau_n} \cap \mathcal{T}_{\tau_n/b}$. In the case $e_2 > 0$, the inclusion $\widehat{\mathcal{T}}_{\tau_n} \cap \mathcal{T}_{\tau_n/b} \subset \widehat{\mathcal{T}}_{\tau_n} \cup \mathcal{T}_{b\tau_n}$ is strict, i.e. there exists $C_{j,k} \in \mathcal{T}^n$ such that either $C_{j,k} \in \widehat{\mathcal{T}}_{\tau_n}$ and $C_{j,k} \notin \mathcal{T}_{\tau_n/b}$, or $C_{j,k} \in \mathcal{T}_{b\tau_n}$ and $C_{j,k} \notin \widehat{\mathcal{T}}_{\tau_n}$. In other words, there exists $C_{j,k} \in \mathcal{T}^n$ such that either $\Delta_{j,k} < 2^{-j}\tau_n/b$ and $\widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n$, or $\Delta_{j,k} \geq b2^{-j}\tau_n$ and $\widehat{\Delta}_{j,k} < 2^{-j}\tau_n$. As a result,

$$\begin{aligned}\mathbb{P}\{e_2 > 0\} \leq & \sum_{C_{j,k}\in\mathcal{T}^n} \mathbb{P}\left\{\widehat{\Delta}_{j,k} < 2^{-j}\tau_n \text{ and } \Delta_{j,k} \geq b2^{-j}\tau_n\right\} \qquad (59) \\ & + \sum_{C_{j,k}\in\mathcal{T}^n} \mathbb{P}\left\{\Delta_{j,k} < 2^{-j}\tau_n/b \text{ and } \widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n\right\}.\end{aligned}$$

$$\mathbb{P}\{e_4 > 0\} \leq \sum_{C_{j,k}\in\mathcal{T}^n} \mathbb{P}\left\{\Delta_{j,k} < 2^{-j}\tau_n/b \text{ and } \widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n\right\}. \qquad (60)$$

We apply (24) in Lemma 20 to estimate the first term in (59):

$$\begin{aligned}& \sum_{C_{j,k}\in\mathcal{T}^n} \mathbb{P}\left\{\widehat{\Delta}_{j,k} < 2^{-j}\tau_n \text{ and } \Delta_{j,k} \geq b2^{-j}\tau_n\right\} \leq \sum_{C_{j,k}\in\mathcal{T}^n} \alpha_1 e^{-\alpha_2 n 2^{2j}\cdot 2^{-2j}\kappa^2\frac{\log n}{n}} \\ & = \alpha_1 \#\mathcal{T}^n n^{-\alpha_2\kappa^2} \leq \alpha_1 a_{\min} n n^{-\alpha_2\kappa^2} \leq \alpha_1 a_{\min} n^{1-\alpha_2\kappa^2} = \alpha_1 a_{\min} n^{-(\alpha_2\kappa^2-1)}.\end{aligned}$$

Using (24), we estimate the second term in (59) and (60) as follows

$$\sum_{C_{j,k}\in\mathcal{T}^n} \mathbb{P}\left\{\Delta_{j,k} < 2^{-j}\tau_n/b \text{ and } \widehat{\Delta}_{j,k} \geq 2^{-j}\tau_n\right\} \leq \sum_{C_{j,k}\in\mathcal{T}^n} \alpha_1 e^{-\alpha_2 n 2^{2j}\cdot\frac{2^{-2j}}{b^2}\kappa^2\frac{\log n}{n}} \leq \alpha_1 a_{\min} n^{-(\alpha_2\kappa^2/b^2-1)}.$$

We therefore obtain (25) by choosing $\kappa > \kappa_2$ with $\alpha_2\kappa_2^2/b^2 = \nu + 1$. ■

Appendix E. Proofs for orthogonal GMRA

E.1. Performance analysis of orthogonal GMRA

The proofs of Theorem 23 and Theorem 25 are resemblant to the proofs of Theorem 4 and Theorem 8. The main difference lies in the variance term, which results in the extra log factors in the convergence rate of orthogonal GMRA. Let Λ be the partition associated with a finite proper subtree $\tilde{\mathcal{T}}$ of the data master tree $\mathcal{T}^n$, and let

$$\mathcal{S}_\Lambda = \sum_{C_{j,k}\in\Lambda} \mathcal{S}_{j,k}\mathbf{1}_{j,k} \quad \text{and} \quad \widehat{\mathcal{S}}_\Lambda = \sum_{C_{j,k}\in\Lambda} \widehat{\mathcal{S}}_{j,k}\mathbf{1}_{j,k}.$$

Lemma 33 *Let Λ be the partition associated with a finite proper subtree $\tilde{\mathcal{T}}$ of the data master tree $\mathcal{T}^n$. Suppose Λ contains $\#_j\Lambda$ cells at scale j. Then for any $\eta > 0$,*

$$\mathbb{P}\{\|\mathcal{S}_\Lambda X - \widehat{\mathcal{S}}_\Lambda X\| \geq \eta\} \leq \alpha d\left(\sum_{j\geq j_{\min}} j\#_j\Lambda\right) e^{-\frac{\beta n\eta^2}{d^2\sum_{j\geq j_{\min}} j^4 2^{-2j}\#_j\Lambda}} \qquad (61)$$

where α and β are the constants in Lemma 15.

Proof [Proof of Lemma 33] The increasing subspaces $\{S_{j,x}\}$ in the construction of orthogonal GMRA may be written as

$$\begin{aligned}S_{0,x} &= V_{0,x} \\ S_{1,x} &= V_{0,x} \oplus V_{0,x}^\perp V_{1,x} \\ S_{2,x} &= V_{0,x} \oplus V_{0,x}^\perp V_{1,x} \oplus V_{1,x}^\perp V_{0,x}^\perp V_{2,x} \\ &\cdots \\ S_{j,x} &= V_{0,x} \oplus V_{0,x}^\perp V_{1,x} \oplus \ldots \oplus V_{j-1,x}^\perp \cdots V_{1,x}^\perp V_{0,x}^\perp V_{j,x}.\end{aligned}$$

Therefore $\|\mathrm{Proj}_{S_{j,x}} - \mathrm{Proj}_{\widehat{S}_{j,x}}\| \leq \sum_{\ell=0}^{j}(j+1-\ell)\|\mathrm{Proj}_{V_{\ell,x}} - \mathrm{Proj}_{\widehat{V}_{\ell,x}}\|$, and then

$$\mathbb{P}\left\{\|\mathrm{Proj}_{S_{j,x}} - \mathrm{Proj}_{\widehat{S}_{j,x}}\| \geq t\right\} \leq \sum_{\ell=0}^{j} \mathbb{P}\left\{\|\mathrm{Proj}_{V_{\ell,x}} - \mathrm{Proj}_{\widehat{V}_{\ell,x}}\| \geq t/j^2\right\}. \qquad (62)$$

The rest of the proof is almost the same as the proof of Lemma 15 in appendix C with a slight modification of (41) substituted by (62). ■

The corollary of Lemma 33 with $\Lambda = \Lambda_j$ results in the following estimate of the variance in empirical orthogonal GMRA.

Lemma 34 *For any $\eta \geq 0$,*

$$\mathbb{P}\{\|\mathcal{S}_j X - \widehat{\mathcal{S}}_j X\| \geq \eta\} \leq \alpha dj\#\Lambda_j e^{-\frac{\beta 2^{2j} n\eta^2}{d^2 j^4\#\Lambda_j}}, \qquad (63)$$

$$\mathbb{E}\|\mathcal{S}_j X - \widehat{\mathcal{S}}_j X\|^2 \leq \frac{d^2 j^4\#\Lambda_j \log[\alpha dj\#\Lambda_j]}{\beta 2^{2j} n}. \qquad (64)$$

Proof [Proof of Theorem 23]

$$\mathbb{E}\|X-\widehat{\mathcal{S}}_jX\|^2 \le \|X-\mathcal{S}_jX\|^2+\mathbb{E}\|\mathcal{S}_jX-\widehat{\mathcal{S}}_jX\|^2$$
$$\le |\rho|^2_{\mathcal{A}^o_s}2^{-2sj}+\frac{d^2j^4\#\Lambda_j\log[\alpha dj\#\Lambda_j]}{\beta 2^{2j}n}\le |\rho|^2_{\mathcal{A}^o_s}2^{-2sj}+\frac{d^2j^4 2^{j(d-2)}}{\theta_1\beta n}\log\frac{\alpha dj2^{jd}}{\theta_1}.$$

When $d\ge 2$, We choose j^* such that $2^{-j^*}=\mu\left((\log^5 n)/n\right)^{\frac{1}{2s+d-2}}$. By grouping Λ_{j^*} into light and heavy cells whose measure is below or above $\frac{28}{3}(\nu+1)\log^5 n/n$, we can show that the error on light cells is upper bounded by $C((\log^5 n)/n)^{\frac{2s}{2s+d-2}}$ and all heavy cells have at least d points with high probability.

Lemma 35 *Suppose j^* is chosen such that $2^{-j^*}=\mu\left(\frac{\log^5 n}{n}\right)^{\frac{1}{2s+d-2}}$ with some $\mu>0$. Then*

$$\|(X-\mathcal{P}_{j^*}X)\mathbf{1}_{\{C_{j^*,k}:\rho(C_{j^*,k})\le\frac{28(\nu+1)\log^5 n}{3n}\}}\|^2\le\frac{28(\nu+1)\theta_2^2\mu^{2-d}}{3\theta_1}\left(\frac{\log^5 n}{n}\right)^{\frac{2s}{2s+d-2}},$$
$$\mathbb{P}\left\{\forall C_{j^*,k}:\rho(C_{j^*,k})>\frac{28(\nu+1)\log^5 n}{3n},\ C_{j^*,k}\ \textit{has at least } d \textit{ points}\right\}\ge 1-n^{-\nu}.$$

Proof of Lemma 35 is omitted since it is the same as the proof of Lemma 17. Lemma 35 guarantees that a sufficient amount of cells at scale j^* has at least d points. The probability estimate in (28) follows from

$$\mathbb{P}\left\{\|\mathcal{S}_{j^*}X-\widehat{\mathcal{S}}_{j^*}X\|\ge C_1\left(\frac{\log^5 n}{n}\right)^{\frac{s}{2s+d-2}}\right\}\le C_2\log n\left(\frac{\log^5 n}{n}\right)^{-\frac{d}{2s+d-2}}e^{-\beta\theta_1\mu^{d-2}C_1^2(2s+d-2)^4/d^2\log n}$$
$$\le C_2(\log n)\,n^{\frac{d}{2s+d-2}}n^{-\beta\theta_1\mu^{d-2}C_1^2(2s+d-2)^4/d^2}\le C_2n^{1-\beta\theta_1\mu^{d-2}C_1^2(2s+d-2)^4/d^2}\le C_2n^{-\nu}$$

provided C_1 is chosen such that $\beta\theta_1\mu^{d-2}C_1^2(2s+d-2)^4/d^2-1>\nu$. The proof when $d=1$ is completely analogous to that of Theorem 4. ■

E.2. Performance analysis of adaptive orthogonal GMRA

Proof [Proof of Theorem 25] Empirical adaptive orthogonal GMRA is given by $\widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}}=\sum_{C_{j,k}\in\widehat{\Lambda}_{\tau_n^o}}\widehat{\mathcal{S}}_{j,k}\mathbf{1}_{j,k}$. Using triangle inequality, we have

$$\|X-\widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}}X\|\le e_1+e_2+e_3+e_4$$

with each term given by

$$e_1:=\|X-\mathcal{S}_{\widehat{\Lambda}_{\tau_n^o}\vee\Lambda_{b\tau_n^o}}X\|\qquad e_2:=\|\mathcal{S}_{\widehat{\Lambda}_{\tau_n^o}\vee\Lambda_{b\tau_n^o}}X-\mathcal{S}_{\widehat{\Lambda}_{\tau_n^o}\wedge\Lambda_{\tau_n^o/b}}X\|$$
$$e_3:=\|\mathcal{S}_{\widehat{\Lambda}_{\tau_n^o}\wedge\Lambda_{\tau_n^o/b}}X-\widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}\wedge\Lambda_{\tau_n^o/b}}X\|\qquad e_4:=\|\widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}\wedge\Lambda_{\tau_n^o/b}}X-\widehat{\mathcal{S}}_{\widehat{\Lambda}_{\tau_n^o}}X\|$$

where $b=2a_{\max}+5$. We will prove the case $d\ge 3$. Here one proceeds in the same way as in the proof of Theorem 8. A slight difference lies in the estimates of e_3, e_2 and e_4.

Term e_3: $\mathbb{E}e_3^2$ is the variance. One can verify that $\mathcal{T}_{(\rho,\tau_n^o/b)}\subset\mathcal{T}_{j_0}:=\cup_{j\le j_0}\Lambda_j$ where j_0 is the largest integer satisfying $2^{j_0d}\le 9b^2\theta_0\theta_2^2/(4\tau_n^{o2})$. The reason is that $\Delta^o_{j,k}\le\frac{3}{2}\theta_2 2^{-j}\sqrt{\theta_0 2^{-jd}}$ so $\Delta^o_{j,k}\ge 2^{-j}\tau_n^o/b$ implies $2^{j_0d}\le 9b^2\theta_0\theta_2^2/(4\tau_n^{o2})$. For any $\eta>0$,

$$\mathbb{P}\{e_3>\eta\}\le\alpha dj_0\#\mathcal{T}_{\tau_n^o/b}e^{-\frac{\beta n\eta^2}{j_0^4\sum_{j\ge j_{\min}}2^{-2j}\#_j\mathcal{T}_{\tau_n^o/b}}}\le\alpha dj_0\#\mathcal{T}_{\tau_n^o/b}e^{-\frac{\beta n\eta^2}{j_0^4|\rho|^p_{\mathcal{B}^o_s}(\tau_n^o/b)^{-p}}}$$

The estimate of $\mathbb{E}e_3^2$ follows from

$$\mathbb{E}e_3^2=\int_0^{+\infty}\eta\mathbb{P}\{e_3>\eta\}\,d\eta=\int_0^{+\infty}\eta\min\left(1,\alpha dj_0\#\mathcal{T}_{\tau_n^o/b}e^{-\frac{\beta n\eta^2}{j_0^4\sum_{j\ge j_{\min}}2^{-2j}\#_j\mathcal{T}_{\tau_n^o/b}}}\right)d\eta$$
$$\le\frac{j_0^4\log\alpha j_0\#\mathcal{T}_{\tau_n^o/b}}{\beta n}\sum_{j\ge j_{\min}}2^{-2j}\#_j\mathcal{T}_{\tau_n^o/b}\le C\frac{\log^5 n}{n}(\tau_n^o/b)^{-p}\le C(\theta_0,\theta_2,\theta_3,a_{\max},\kappa,d,s)\left(\frac{\log^5 n}{n}\right)^{\frac{2s}{2s+d-2}}.$$

Term e_2 and e_4: These two terms are analyzed with Lemma 36 stated below such that for any fixed but arbitrary $\nu>0$,

$$\mathbb{P}\{e_2>0\}+\mathbb{P}\{e_4>0\}\le\beta_1a_{\min}/dn^{-\nu}$$

if κ is chosen such that $\kappa>\kappa_2$ with $d^4\beta_2\kappa_2^2/b^2=\nu+1$.

Lemma 36 *$b=2a_{\max}+5$. For any $\eta>0$ and any $C_{j,k}\in\mathcal{T}$*

$$\max\left(\mathbb{P}\left\{\widehat{\Delta}^o_{j,k}\le\eta\ \textit{and}\ \Delta^o_{j,k}\ge b\eta\right\},\mathbb{P}\left\{\Delta^o_{j,k}\le\eta\ \textit{and}\ \widehat{\Delta}^o_{j,k}\ge b\eta\right\}\right)\le\beta_1je^{-\beta_2n2^{2j}\eta^2/j^4},$$

with positive constants $\beta_1:=\beta_1(\theta_2,\theta_3,\theta_4,a_{\max},d)$ and $\beta_2:=\beta_2(\theta_2,\theta_3,\theta_4,a_{\max},d)$.

■

References

M. Aharon, M. Elad, and A. Bruckstein. K-SVD: Design of dictionaries for sparse representation. In *Proceedings OF SPARS*, pages 9–12, 2005.

W. K. Allard, G. Chen, and M. Maggioni. Multi-scale geometric methods for data sets II: Geometric multi-resolution analysis. *Applied and Computational Harmonic Analysis*, 32 (3):435–462, 2012.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *International Conference on Machine Learning*, pages 97–104, 2006.

P. Binev, A. Cohen, W. Dahmen, R.A. DeVore, and V. Temlyakov. Universal algorithms for learning theory part i: piecewise constant functions. *Journal of Machine Learning Research*, 6:1297–1321, 2005.

P. Binev, A. Cohen, W. Dahmen, R.A. DeVore, and V. Temlyakov. Universal algorithms for learning theory part ii: piecewise polynomial functions. *Constructive Approximation*, 26(2):127–152, 2007.

G. Canas, T. Poggio, and L. Rosasco. Learning manifolds with k-means and k-flats. In *Advances in Neural Information Processing Systems*, 2012.

E. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, pages 2313–2351, 2007. math.ST/0506081.

G. Chen and G. Lerman. Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Foundations of Computational Mathematics*, 9:517–558, 2009.

G. Chen and M. Maggioni. Multiscale geometric and spectral analysis of plane arrangements. In *Conference on Computer Vision and Pattern Recognition*, 2011.

G. Chen, M. Iwen, S. Chin, and M. Maggioni. A fast multiscale framework for data in high-dimensions: Measure estimation, anomaly detection, and compressive measurements. In *Visual Communications and Image Processing (VCIP), 2012 IEEE*, pages 1–6, 2012.

Guangliang Chen and Mauro Maggioni. Multiscale geometric wavelets for the analysis of point clouds. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2010.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

Michael Christ. A t(b) theorem with remarks on analytic capacity and the cauchy integral. *Colloquium Mathematicae*, 60-61(2):601–628, 1990.

A. Cohen, I. Daubechies, O. G. Guleryuz, and M. T. Orchard. On the importance of combining wavelet-based nonlinear approximation with coding strategies. *IEEE Transactions on Information Theory*, 48(7):1895–1921, 2002.

R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–7431, 2005a.

R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7432–7438, 2005b.

I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992.

G. David and S. Semmes. *Analysis of and on uniformly rectifiable sets*, volume 38 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 1993.

C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

D. Deng and Y. Han. *Harmonic analysis on spaces of homogeneous type*, volume 19. Springer Science and Business Media, 2008.

D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

D. L Donoho and C. Grimes. Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, pages 5591–5596, March 2003.

D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

D. L. Donoho and J. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

A. Eftekhari and M. B. Wakin. Sparse subspace clustering. *Applied and Computational Harmonic Analysis*, 39(1):67–109, 2015.

E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, June 2009.

R. Fergus F. Li and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

S. Gerber and M. Maggioni. Multiscale dictionaries, transforms, and learning in high-dimensions. In *Proc. SPIE conference Optics and Photonics*, 2013.

R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert. Sample complexity of dictionary learning and other matrix factorizations. *IEEE Transactions on Information Theory*, 61(6):3469–3486, 2015.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A distribution-free theory of nonparametric regression*. New York: Springer, 2002.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. 2009.

J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 11–18, 2003.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(4):17–441,498–520, 1933.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

M. A. Iwen and M. Maggioni. Approximation of points on low-dimensional manifolds via random linear projections. *Inference and Information*, 2(1):1–31, 2013. arXiv:1204.3337v1, 2012.

P. W. Jones. Rectifiable sets and the traveling salesman problem. *Inventiones Mathematicae*, 102(1):1–15, 1990.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.

K. Kreutz-Delgado, J. F. Murray, B. D. Rao, Kjersti Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, February 2003.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

M.S. Lewicki, T.J. Sejnowski, and H. Hughes. Learning overcomplete representations. *Neural Computation*, 12:337–365, 1998.

A. V. Little, M. Maggioni, and L. Rosasco. Multiscale geometric methods for data sets i: Multiscale svd, noise and curvature. *Applied and Computational Harmonic Analysis*, 43 (3):504–567, 2017.

A.V. Little, Y.-M. Jung, and M. Maggioni. Multiscale estimation of intrinsic dimensionality of data sets. In *Proc. AAAI Fall Symposium Series*, 2009a.

A.V. Little, J. Lee, Y.-M. Jung, and M. Maggioni. Estimation of intrinsic dimensionality of samples from noisy low-dimensional manifolds in high dimensions with multiscale SVD. In *Proc. IEEE/SP 15th Workshop on Statistical Signal Processing*, 2009b.

G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. *Proceedings of the 27th International Conference on Machine Learning*, 2010.

Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1–17, 2007.

Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008.

M. Maggioni, S. Minsker, and N. Strawn. Multiscale dictionary learning: Non-asymptotic bounds and robustness. *Journal of Machine Learning Research*, 17(1):43–93, January 2016.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

S. G. Mallat. *A wavelet tour in signal processing*. Academic Press, 1998.

A. Maurer and M. Pontil. K-dimensional coding schemes in hilbert spaces. *IEEE Transactions on Information Theory*, 56(11):5839–5846, 2010.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

G. Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2):249–260, 2009.

G. Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34 (1):17–31, 2009.

M. Protter and M. Elad. Sparse and redundant representations and motion-estimation-free algorithm for video denoising, 2007.

M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi. Determination of reaction coordinates via locally scaled diffusion map. *Journal of Chemical Physics*, (134):124116, 2011.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

A. Szlam. Asymptotic regularity of subdivisions of euclidean domains by iterated PCA and iterated 2-means. *Applied and Computational Harmonic Analysis*, 27(3):342–350, 2009.

J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

J. A. Tropp. User-friendly tools for random matrices: An introduction. NIPS version, 2014. URL `http://users.cms.caltech.edu/~jtropp/`.

D. Vainsencher, S. Mannor, and A. M. Bruckstein. The sample complexity of dictionary learning. *The Journal of Machine Learning Research*, 12:3259–3281, 2011.

R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *European conference on computer vision*, volume 4, pages 94–106, 2006.

T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Randomized hybrid linear modeling by local best-fit flats. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 2010.

Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Journal of Shanghai University*, 8(4):406–424, 2004.

W. Zheng, M. A. Rohrdanz, M. Maggioni, and C. Clementi. Polymer reversal rate calculated via locally scaled diffusion map. *Journal of Chemical Physics*, (134):144108, 2011.

L. Zwald and G. Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *Advances in Neural Information Processing Systems*, pages 1649–1656, 2006.

Journal of Machine Learning Research 20 (2019) 1-51 Submitted 6/17; Revised 5/19; Published 6/19

Change Surfaces for Expressive Multidimensional Changepoints and Counterfactual Prediction

William Herlands HERLANDS@CMU.EDU
Event and Pattern Detection Laboratory
H.J. Heinz III College and Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Daniel B. Neill DANIEL.NEILL@NYU.EDU
Center for Urban Science and Progress, NYU Wagner School of Public Service, and NYU Courant Department of Computer Science
New York University
Brooklyn, NY 11201, USA

Hannes Nickisch HANNES@NICKISCH.ORG
Digital Imaging
Philips Research Hamburg
Röntgenstraße 24-26
22335 Hamburg, Germany

Andrew Gordon Wilson ANDREW@CORNELL.EDU
Operations Research and Information Engineering
Cornell University
Ithaca, NY 14853, USA

Editor: Andreas Krause

Abstract

Identifying changes in model parameters is fundamental in machine learning and statistics. However, standard changepoint models are limited in expressiveness, often addressing unidimensional problems and assuming instantaneous changes. We introduce *change surfaces* as a multidimensional and highly expressive generalization of changepoints. We provide a model-agnostic formalization of change surfaces, illustrating how they can provide variable, heterogeneous, and non-monotonic rates of change across multiple dimensions. Additionally, we show how change surfaces can be used for counterfactual prediction. As a concrete instantiation of the change surface framework, we develop Gaussian Process Change Surfaces (GPCS). We demonstrate counterfactual prediction with Bayesian posterior mean and credible sets, as well as massive scalability by introducing novel methods for additive non-separable kernels. Using two large spatio-temporal datasets we employ GPCS to discover and characterize complex changes that can provide scientific and policy relevant insights. Specifically, we analyze twentieth century measles incidence across the United States and discover previously unknown heterogeneous changes after the introduction of the measles vaccine. Additionally, we apply the model to requests for lead testing kits in New York City, discovering distinct spatial and demographic patterns.

Keywords: Change surface, changepoint, counterfactual, Gaussian process, scalable inference, kernel method

©2019 William Herlands, Daniel B. Neill, Hannes Nickisch, and Andrew Gordon Wilson

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-352.html.

1. Introduction

Detecting and modeling changes in data is critical in statistical theory, scientific discovery, and public policy. For example, in epidemiology, detecting changes in disease dynamics can provide information about when and where a vaccination program becomes effective. In dangerous professions such as coal mining, changes in accident occurrence patterns can indicate which regulations impact worker safety. In city governance, policy makers may be interested in how requests for health services change across space and over time.

Changepoint models have a long history in statistics, beginning in the mid-twentieth century, when methods were first developed to identify changes in a data generating process (Page, 1954; Horváth and Rice, 2014). The primary goal of these models is to determine if a change in the distribution of the data has occurred, and then to locate one or more points in the domain where such changes occur. While identifying these changepoints is an important result in itself, changepoint methods are also frequently applied to other problems such as outlier detection or failure analysis (Reece et al., 2015; Tartakovsky et al., 2013; Kapur et al., 2011). Different changepoint methods are distinguished by the diversity of changepoints they are able to detect and the complexity of the underlying data. The simplest models consider mean shifts between functional regimes (Chernoff and Zacks, 1964; Killick et al., 2012), while others consider changes in the covariance structure or higher order moments (Keshavarz et al., 2018; Ross, 2013; James and Matteson, 2013). A *regime* is a particular data generating process or underlying function that is separated from other underlying processes or functions by changepoints. Additionally, there is a fundamental distinction between changepoint models that identify changes sequentially using online algorithms, and those that analyze data retrospectively to find one or more changes in past data (Brodsky and Darkhovsky, 2013; Chen and Gupta, 2011). Finally, changepoint methods may be fully parametric, semi-parametric, or nonparametric (Ross, 2013; Guan, 2004). For additional discussion of changepoints beyond the scope of this paper, readers may consider the literature reviews in Aue and Horváth (2013), Ivanoff and Merzbach (2010), and Aminikhanghahi and Cook (2017).

Yet nearly all changepoint methods described in the statistics and machine learning literature consider system perturbations as discrete changepoints. This literature seeks to identify instantaneous differences in parameter distributions. The advantage of such models is that they provide definitive assessments of the location of one or more changepoints. This approach is reasonable, for instance, when considering catastrophic events in a mechanical system, such as the effect of a car crash on various embedded sensor readings. Yet the challenge with these models is that real world systems rarely exhibit a clear binary transition between regimes. Indeed, in many applications, such as in biological science, instantaneous changes may be physically impossible. While a handful of approaches consider non-discrete changepoints (e.g., Wilson et al., 2012; Wilson, 2014; Lloyd et al., 2014) they still require linear, monotonic, one-dimensional, and, in practice, relatively quick changes. Existing models do not provide the expressiveness necessary to model complex changes.

Additionally, applying changepoints to multiple dimensions, such as spatio-temporal data, is theoretically and practically non-trivial. Previous literature, exemplified by Guinness et al. (2013), use jump processes for changes in spatio-temporal data. Jump processes can model abrupt, non-discrete changes in the multivariate data. Yet as Guinness et al.

(2013) note, their model requires careful, application-dependent parametric choices which severely limit generalizability. Alternatively, Majumdar et al. (2005) model discrete spatio-temporal changepoints with three additive Gaussian processes: one for $t \leq t_0$, one for $t > t_0$, and one for all t. Nicholls and Nunn (2010) use a Bayesian onset-field process on a lattice to model the spatio-temporal distribution of human settlement on the Fiji islands. However, both the models in these two papers are limited to considering discrete changepoints.

1.1. Main contributions

In this paper, we introduce *change surfaces* as expressive, multidimensional generalizations of changepoints. We present a model-agnostic formulation of change surfaces and instantiate this framework with scalable Gaussian process models. The resulting model is capable of automatically learning expressive covariance functions and a sophisticated continuous change surface. Additionally, we derive massively scalable inference procedures, as well as counterfactual prediction techniques. Finally, we apply the proposed methods to a wide variety of numerical data and complex human systems. In particular, we:

1. Introduce change surfaces as multidimensional and highly flexible generalizations of changepoint modeling.
2. Introduce a procedure which allows one to specify background functions and change functions, for more powerful inductive biases and added interpretability.
3. Provide a new framework for counterfactual prediction using change surfaces.
4. Present the Gaussian Process Change Surface model (GPCS) which models change surfaces with highly flexible Random Kitchen Sink (Rahimi and Recht, 2007) features.
5. Develop massively scalable additive, non-stationary, non-separable kernels by using the Weyl inequality (Weyl, 1912) and novel Kronecker methods. In addition we integrate our approach into the recent KISS-GP framework (Wilson and Nickisch, 2015). The resulting approach is the first scalable Gaussian process multidimensional changepoint model.
6. Describe a novel initialization method for spectral mixture kernels (Wilson and Adams, 2013) by fitting a Gaussian mixture model to the Fourier transform of the data. This method provides good starting values for hyperparameters of expressive stationary kernels, allowing for successful optimization over a multimodal parameter space.
7. Demonstrate that the GPCS approach is robust to misspecification, and automatically discourages extraneous model complexity, leading to the discovery of interpretable generative hypotheses for the data.
8. Perform counterfactual prediction in complex real world data with posterior mean and covariance estimates for each point in the input domain.
9. Use GPCS for discovering and characterizing continuous changes in large observational data. We demonstrate our approach on a recently released public health dataset providing new insight that suggests how the effect of the 1963 measles vaccine may have varied over space and time in the United States. Additionally, we apply the model to requests for lead testing kits in New York City from 2014-2016. The results illustrate distinct spatial patterns in increased concern about lead-tainted water.

1.2. Outline

The paper is divided into three main units.

Section 2 formally introduces the notion of change surfaces as a multidimensional, expressive generalization of changepoints. We discuss a variant of change surfaces in section 2.1 and detail how to use change surfaces for counterfactual prediction in section 2.2. The discussion of change surfaces in this unit is method-agnostic, and should be relevant to experts from a wide variety of statistical and machine learning disciplines. We emphasize the novel contribution of this framework to the general field of change detection.

Section 3 presents the Gaussian Process Change Surface (GPCS) as a scalable method for change surface modeling. We review Gaussian process basics in section 3.1. We specify the GPCS model in section 3.2. Counterfactual predictions with GPCS are derived in section 3.3. Scalable inference using novel Kronecker methods are presented in section 3.4, and we describe a novel initialization technique for expressive Gaussian process kernels in section 3.5.

Section 4 demonstrates GPCS on *out-of-class* numerical data and complex spatio-temporal data. We describe our numerical setup in section 4.1 presenting results for posterior prediction, change surface identification, and counterfactual prediction. We present a one-dimensional application of GPCS on coal mining data in section 4.2 including a comparison to state-of-the-art changepoint methods. Moving to spatio-temporal data, we apply GPCS to model requests for lead testing kits in New York City in section 4.3 and discuss the policy relevant conclusions. Additionally, we use GPCS to model measles incidence in the United States in section 4.4 and discuss scientifically relevant insights.

Finally, we conclude with summary remarks in section 5.

2. Change surfaces

In human systems and scientific phenomena we are often confronted with changes or perturbations which may not immediately disrupt an entire system. Instead, changes such as policy interventions and natural disasters take time to affect deeply ingrained habits or trickle through a complex bureaucracy. The dynamics of these changes are non-trivial, with sophisticated distributions, rates, and intensity functions. Using expressive models to fully characterize such changes is essential for accurate predictions and scientifically meaningful results. For example, in the spatio-temporal domain, changes are often heterogeneously distributed across space and time. Capturing the complexity of these changes provides useful insights for future policy makers enabling them to better target or structure policy interventions.

In order to provide the expressive capability for such models, we introduce the notion of a *change surface* as a generalization of changepoints. We assume data are (x, y), where $x = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^D$, are inputs or covariates, and $y = \{y_1, \ldots, y_n\}$, $y_i \in \mathbb{R}$, are outputs or response variables indexed by x. A change surface defines transitions between latent functions $f_1, \ldots, f_r$ defining r regimes in the data. Unlike with changepoints, we

do not require that the transitions be discrete. Instead we define r warping functions $s(x) = [s_1(x), \ldots, s_r(x)]$ where $s_i(x) : \mathbb{R}^D \to [0, 1]$, which have support over the entire domain of x. Importantly, these warping functions have an inductive bias towards $\{0, 1\}$ creating a soft mutual exclusivity between the functions. We define the canonical form of a change surface as

$$\begin{aligned} y(x) &= s_1(x)f_1(x) + \cdots + s_r(x)f_r(x) + \epsilon \\ &s.t. \\ &\sum_{i=1}^{r} s_i(x) = 1 \\ &s_i(x) \geq 0 \end{aligned} \tag{1}$$

where $\epsilon(x)$ is noise. Each $s_i(x)$ defines how the coverage of $f_i(x)$ varies over the input domain. Where $s_i(x) \approx 1$, $f_i(x)$ dominates and primarily describes the relationship between x and y. In cases where there is no i such that $s_i(x) \approx 1$, a number of functions are dominant in defining the relationship between x and y. Since $s(x)$ has a strong inductive bias towards 1 or 0, the regions with multiple dominant functions are transitory and often the areas of interest. Therefore, we can interpret how the change surface develops and where different regimes dominate by evaluating each $s(x)$ over the input domain.

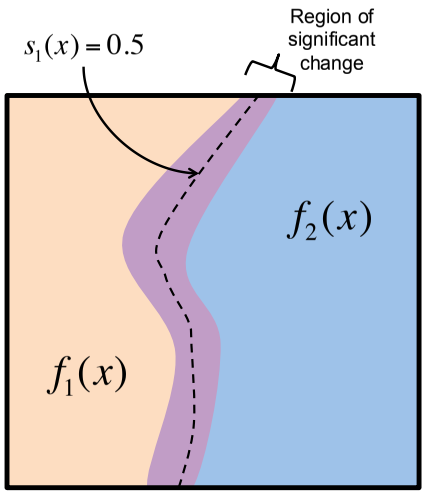


Figure 1: Two-dimensional depiction of the change surface model where $f_1(x)$ is drawn in orange and $f_2(x)$ is drawn in blue. The region in purple depicts an area of transition between the two functions. The dashed line represents the domain where $s_1(x) = 0.5$.

Figure 1 depicts a two-dimensional change surface model where latent $f_1(x)$ is drawn in orange and latent $f_2(x)$ is drawn in blue. In those areas the first warping function, $s_1(x)$, is

nearly 1 and 0 respectively. The region in purple depicts an area of transition between the two functions. We would expect that $s_1(x) \approx 0.5$ in this region since both latent functions are active.

In many applications we can imagine that a latent background function, $f_0(x)$, exists that is common to all data regimes. One could reparametrize the model in Eq. (1) by letting each latent regime be a sum of two functions: $f_0(x) + f_i(x)$. Thus each regime compartmentalizes into $f_0(x)$, a common background function, and $f_i(x)$, a regime-specific latent function. This provides a generalized change surface model,

$$y(x) = f_0(x) + s_1(x)f_1(x) + \cdots + s_r(x)f_r(x) + \epsilon(x). \tag{2}$$

et even with the $f_0(x)$ background function, the inductive bias towards $\{0, 1\}$ is still critical to ensure that each function in the change surface models a distinct regime in the data. At the boundary when $s_i(x)$ is 0 or 1 for any x then the model describes discrete multivariate changepoints (see more about comparing change surfaces to changepoints in the section below). Alternatively, when $s_i(x)$ is a constant value for all x then the model describes a constant mixture without change regions.

Change surfaces can be considered particular types of *adaptive* mixture models (e.g., Wilson et al., 2012), where $s(x)$ are mixture weights in a simplex that have a strong inductive bias towards discretization. There are multiple ways to induce this bias towards discretization. For example, one can choose warping functions $s(x)$ which have sharp transitions between 0 and 1, such as the logistic sigmoid function. With multiple functions, $r \geq 2$, we can also explicitly penalize the warping functions from having similar values. Since each of these warping functions are constrained to be in $[0, 1]$ this penalty would tend move their values towards 0 or 1. More generally, in the case of multiple functional regimes, we can penalize $s(x)$ from being far from $\{0, 1\}$. For example, we could place a prior over $s(x)$ with a heavy weight on 1 and 0.

Comparison to changepoint models: The flexibility of $s(x)$ defines the complexity of the change surface. In the simplest case, $x_i \in \mathbb{R}^1, s(x) \in \{0, 1\}$, and the change surface reduces to a univariate changepoint used in much of the changepoint literature. Alternatively, if we consider $x \in \mathbb{R}^1, s(x) = \sigma(x)$ the change surface is a smooth univariate changepoint with a fixed rate of change. Such a model only permits a monotonic rate of change and single changepoint.

We illustrate the difference between the warping functions, $s(x)$, of a change surface model and standard changepoint methods in Figure 2. The top plot shows unidimensional data with a clear change between two sinusoids. The subsequent plots represent the changes modeled in a discrete changepoint, sigmoid changepoint, and change surface model respectively. The changepoint model can only identify a change at a point in time, and the sigmoid changepoint is a special case of a change surface constrained to a fixed rate of change. However, a general change surface can model gradual changes as well as non-monotonic changes, providing a much richer representation of the data's dynamics, and seamlessly extending to multidimensional data.

Expressive change surfaces consider regimes as overlapping elements in the domain. They can illustrate if certain changes occur more slowly or quickly, vary over particular subpopulations, or change rapidly in certain regions of the input domain. Such insights are

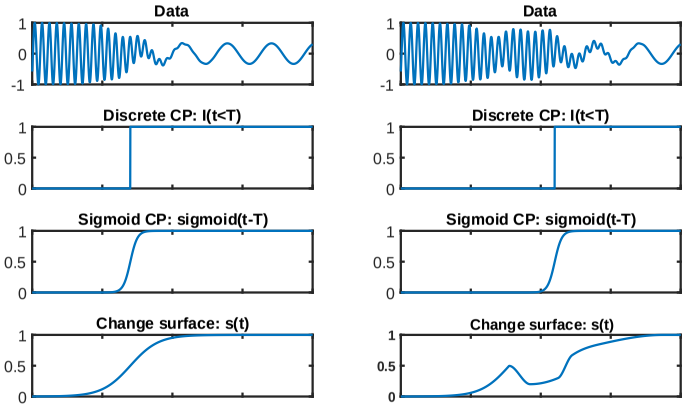


Figure 2: Unidimensional comparison of changepoint and change surface methods. In each column, the top plot shows unidimensional data with a clear change between two sinusoids. The subsequent plots represent the warping functions of a discrete changepoint, sigmoid changepoint, and change surface model.

not provided by standard changepoint models but are critical for understanding policy interventions or scientific processes. Table 1 compares some of the limitations of changepoints with the added flexibility of change surfaces.

Table 1: Comparison of changepoint limitations to change surface flexibility.

Changepoints limited by:	**Change surfaces allow for:**
Considering unidimensional, often temporal-only problems	Multidimensional inputs with heterogeneous changes across the input dimensions. Indeed, we apply change surfaces to 3-dimensional, spatio-temporal problems in section 4.
Detecting discrete or near-discrete changes in parameter distribution	Warping functions, $s(x)$, can be defined flexibly to allow for discrete or continuous changes with variable, and even non-monotonic rates of change.
Not simultaneously modeling the latent functional regimes	Learning $s_i(x)$ and $f_i(x)$ in Equation (1) to simultaneously model the change surface and underlying functional regimes.

Yet the flexibility required by change surfaces as applied to real data sets might seem difficult to instantiate with any particular model. Indeed, machine learning methods are

often desired to be expressive, interpretable, and scalable to large data. To address this challenge we introduce the Gaussian Process Change Surface (GPCS) in section 3 which uses Gaussian process priors with flexible kernels to provide rich modeling capability, and a novel scalable inference scheme to permit the method to scale to massive data.

2.1. Change surface background model

In certain applications we are interested in modeling how a change occurs concurrent with a background function which is common to all regimes. For example, consider urban crime. If a police department staged a prolonged intervention in one sector of the city, we expect that some of the crime dynamics in that sector might change. However, seasonal and other weather-related patterns may remain the same throughout the entire city. In this case we want a model to identify and isolate those general background patterns as well as one or more clearly interpretable functions representing regions of change from the background distribution.

We can accommodate such a model as a special case of the generalized change surface from Eq. (2). Each latent function is modeled as $f_0(x) + f_i(x)$ where $f_0(x)$ models "background" dynamics, and $f_i(x)$ models each *change* function. Since changes do not necessarily persist over the entire domain, we fix $f_r(x) = 0$, and allow $\sum_{i=1}^{r-1} s_i(x) \leq 1$. This approach results in the following *change surface background model*:

$$\begin{aligned} y(x) &= f_0(x) + s_1(x)f_1(x) + \cdots + s_{r-1}(x)f_{r-1}(x) + \epsilon \\ &s.t. \\ &\sum_{i=1}^{r-1} s_i(x) \leq 1 \\ &s_i(x) \geq 0 \end{aligned} \tag{3}$$

Figure 3 presents a two-dimensional representation of the change surface and change surface background models. The data depicted comes from the numerical experiments in section 4.1.

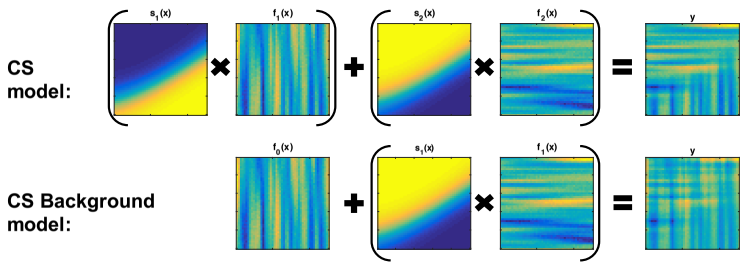


Figure 3: Two-dimensional representation of the change surface model (Eq. 1) and change surface background model (Eq. 3).

The explicit decomposition into background and change functions is valuable, for instance, if we wish to model *counterfactuals*: we want to know what the data in a region might look like had there been no change. The decomposition also enables us to interpret the precise effect of each change. Moreover, from a statistical perspective, the decomposition allows us to naturally encode inductive biases into the change surface model, allowing meaningful *a priori* statistical dependencies between each region. In the particular case of $r = 2$, the change surface background model has the form $y(x) = f_0(x) + s_1(x)f_1(x)$, where $f_1(x)$ is the only change function modulated by a change surface, $s_1(x) \in [0, 1]$. This corresponds to observation studies or natural experiments where a single change is observed in the data. We explore this special case further in our discussion of counterfactual prediction, in section 2.2.

Finally, for any change surface or change surface background model, it is critical that the model not overfit the data due to a proliferation of parameters, which could lead to erroneously detected changes even when no dynamic change is present. We discuss one strategy for preventing overfitting through the use of Gaussian processes in section 3.

2.2. Counterfactual prediction

By simultaneously characterizing the change surface, $s(x)$, and the underlying generative functions, $f(x)$, change surface models allow us to ask questions about how the data would have looked had there been only one latent function. In other words, change surface models allow us to consider counterfactual questions.

For example, in section 4.4 we consider measles disease incidence in the United States in the twentieth century. The measles vaccine was introduced in 1963, radically changing the dynamics of disease incidence. Counterfactual studies such as van Panhuis et al. (2013) attempt to estimate how many cases of measles there would have been in the absence of the vaccine. To be clear, since change surface models do not consider explicit indicators of an intervention, they do not directly estimate the counterfactual with respect to a particular treatment variable such as vaccination. Instead, they identify and characterize changes in the data generating process that may or may not correspond to a known intervention. The change surface counterfactuals estimate the y values for each functional regime in the absence of the change identified by the change surface model. In cases where the discovered change surface does correspond to a known intervention of interest, domain experts may interpret the change surface predictions as a counterfactual "what if" that intervention and any contemporaneous changes in the data generating process (note that we cannot disentangle these causal factors without explicit intervention labels) did not occur.

Counterfactuals are typically studied in econometrics. In observational studies econometricians try to measure the effect of a "treatment" over some domain. Econometric models often measure simple features of the intervention effect, such as the expected value of the treatment over the entire domain, also known as the *average treatment effect*. A nascent body of work considers machine learning approaches to provide counterfactual prediction in complex data (Athey and Imbens, 2006; Brodersen et al., 2015; Johansson et al., 2016; Hartford et al., 2016), as well as richer measures of the intervention effect (Athey and Imbens, 2006; McFowland et al., 2016). Recent work by Schulam and Saria (2017) uses Gaussian processes for trajectory counterfactual prediction over time. However, these methods generally follow a common framework using the potential outcomes model, which assumes that each observation is observed with a discrete treatment (Rubin, 2005; Holland, 1986). With discrete treatments a unit, x, is either intervened upon or not intervened upon — there are no partial interventions. For example, in a medical study a patient may be given a vaccination, or given a placebo. Such discretization is similar to a traditional changepoint model where $s(x) \in \{0, 1\}$ can only be in one of two states. Yet discrete states prove challenging in practical applications where units may be partially treated or affected through spillover. For example, there may be herd effects in vaccinations whereby a person's neighbor being vaccinated reduces the risk of infection to the person. Certain econometric models attempt to account for partial treatment such as treatment eligibility (Abadie et al., 2002), where partial treatments are induced by defining proportions of the population that could potentially be treated. Yet a model that directly enables and estimates continuous levels of treatment may be more natural in such cases.

Counterfactuals using change surfaces. Change surface models enable counterfactual prediction in potentially complex data through the expressive parameterization of the latent functions, $f_1(x), \ldots, f_r(x)$. Determining the individual function value $f_i(x)$ over the input domain is equivalent to determining the counterfactual of $f_i(x)$ in the absence of all other latent functions. We can compute counterfactual estimates for latent functions in either the regular change surface model or the change surface background model. In the latter case, if $r = 2$ recall that the model takes the form $y = f_0(x) + s_1(x)f_1(x) + \epsilon$. Determining the counterfactual for $f_0(x)$ provides an estimate for the data without the detected change, while the counterfactual for $f_1(x)$ estimates the effect of that change across the entire regime.

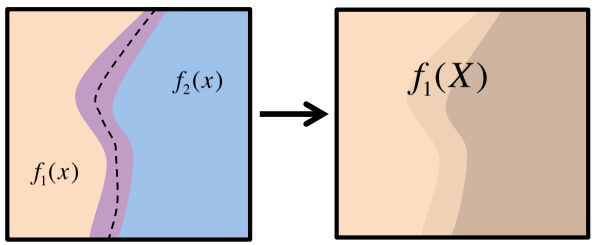


Figure 4: Two-dimensional depiction of change surface counterfactual prediction. The left panel illustrates the change surface of Figure 1. The right image depicts the counterfactual of f_1 over the entire domain, X, representing what the observed data could look like in the *absence* of an intervention. The darker shading of the picture depicts larger posterior uncertainty.

For example, Figure 4 depicts the counterfactual of f_1 from Figure 1, where f_1 is predicted over the entire regime, X. The darker shading of the picture depicts larger posterior uncertainty. As we move toward the right portion of the plot, away from data regions where f_1 was active, we have greater uncertainty in our counterfactual predictions.

Computing counterfactuals for each $f_i(x)$ provides insight into the effect of a change on the various regimes. When combined with domain expertise, these models may also be useful for estimating the treatment effect of specific variables. Additionally, given a Bayesian formulation of the change surface, such as that proposed in section 3, we can compute the full posterior distribution over the counterfactual prediction rather than just a point estimate. Finally, since change surfaces model all data points as a combination of latent functions, we do not assume that observed data comes from a particular treatment or control. Rather we learn the contribution of each functional regime to each data point.

Some simple changepoint models could, in theory, provide the ability for counterfactual prediction between regimes. But since changepoint models consider each regime either completely or nearly independently of other regimes, there is no information shared between regimes. This lack of information sharing across regimes makes accurate counterfactual prediction challenging without strong assumptions about the data generating process. Indeed, to our knowledge there is no previous literature using changepoint models for counterfactual prediction.

Assumptions in change surface counterfactuals: Change surface models identify changing data dynamics without explicitly considering intervention labels. Instead, counterfactuals of the functional regimes are computed with respect to the change surface labels, $s(x)$. Thus these counterfactuals estimate the value of functional regimes in the absence of those changes but do not necessarily represent counterfactual estimates of any particular variable. The interpretation of these counterfactuals as estimates for each functional regime in the absence of a specific known intervention requires identification of the correct change surface, i.e.:

- The intervention induces a change in the data generation process that cannot be modeled with a single latent functional regime.
- The magnitude of the change is large enough to be detected.
- The change surface model is sufficiently flexible to accurately characterize this change.
- The change surface model does not overfit the data to erroneously identify a change.

Moreover, the resulting counterfactual estimates do not rule out the possibility that other changes in the data generating process occur contemporaneously with the intervention of interest. As such, these counterfactuals are most naturally interpreted as estimating what the data would look like in the absence of the intervention and any other contemporaneous changes. Disentangling these multiple potential causal factors would require additional data about both the intervention and other potential causes.

Change surface counterfactual predictions can provide immense value in practical settings. Although in some datasets explicit intervention labels are available, many observational datasets do not have such labels. Learning a change surface effectively provides a real-valued label that can be used to predict counterfactuals. Even when the approximate boundaries of an intervention are known, change surface modeling can still provide an important advantage since the intervention labels may not capture the true complexity of the data. For example, knowing the date that the measles vaccine was introduced does not

account for regional variation in vaccine distribution and uptake (see section 4.4). Both observational studies and randomized control trials suffer from partial treatment or spillover, where an intervention on one agent or region secondarily affects a non-intervened agent or region. For example, increasing policing in one area of the city may displace crime from the intervened region to other areas of the city (Verbitsky-Savitz and Raudenbush, 2012). This effect violates the Stable Unit Treatment Value Assumption, which is the basis for many estimation techniques in economics (Rubin, 1986). By using the assumed boundaries of an intervention as a prior over $s(x)$, a change surface model can discover if, and where, spillover occurs. This spillover will be captured as a non-discrete change and can aid both in interpretability of the results and counterfactual prediction. In all these cases change surface counterfactuals may lead to more believable counterfactual predictions by using a real valued change surface to directly model spillover and interventions.

3. Gaussian Process Change Surfaces (GPCS)

We exemplify the general concept of change surfaces using Gaussian processes (e.g., Rasmussen and Williams, 2006). We emphasize that our change surface formulations from section 2 are not limited to a certain class of models. Yet Gaussian processes offer a compelling instantiation of change surfaces since they can flexibly model non-linear functions, seamlessly extend to multidimensional and irregularly sampled data, and provide naturally interpretable parameters. Perhaps most importantly, due to the Bayesian Occam's Razor principle (Rasmussen and Ghahramani, 2001; MacKay, 2003; Rasmussen and Williams, 2006; Wilson et al., 2014), Gaussian processes do not in general overfit the data, and extraneous model components are automatically pruned. Indeed, even though we develop a rich change surface model with multiple mixture parameters, our results below demonstrate that the model does not spuriously identify change surfaces in data.

Gaussian processes have been previously used for nonparametric changepoint modeling. Saatçi et al. (2010) extend the sequential Bayesian Online Changepoint Detection algorithm (Adams and MacKay, 2007) by using a Gaussian process to model temporal covariance within a particular regime. Similarly, Garnett et al. (2009) provide Gaussian processes for sequential changepoint detection with mutually exclusive regimes. Moreover, Keshavarz et al. (2018) prove asymptotic convergence bounds for a class of Gaussian process changepoint detection but are restricted to considering a single abrupt change in one-dimensional data. Focusing on anomaly detection, Reece et al. (2015) develop a non-stationary kernel that could conceivably be used to model a changepoint in covariance structure. However, as with most of the changepoint models discussed in section 1, these models all focus on discrete changepoints, where regimes defined by distinct Gaussian processes change instantaneously.

A small collection of pioneering work has briefly considered the possibility of Gaussian processes with sigmoid changepoints (Wilson, 2014; Lloyd et al., 2014). Yet these models rely on sigmoid transformations of linear functions which are restricted to fixed rates of change, and are demonstrated exclusively on small, one-dimensional time series data. They cannot expressively characterize non-linear changes or feasibly operate on large multidimensional data.

The limitations of these models reflect a common criticism that Gaussian processes are unable to convincingly respond to changes in covariance structure. We propose addressing this deficiency by modeling change surfaces with Gaussian processes. Thus our work both demonstrates a generalization of changepoint models and an enhancement to the expressive power of Gaussian processes.

3.1. Gaussian processes overview

We provide a brief review of Gaussian processes. More detail can be found in Rasmussen and Williams (2006), Schölkopf and Smola (2002), and MacKay (1998).

Consider data, (x, y), as in section 2, where $x = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^D$, are inputs or covariates, and $y = \{y_1, \ldots, y_n\}$, $y_i \in \mathbb{R}$ are outputs or response variables indexed by x. We assume that y is generated from x by a latent function with a Gaussian process prior (GP) and Gaussian noise. In particular,

$$y = f(x) + \epsilon \tag{4}$$
$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')) \tag{5}$$
$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{6}$$

A Gaussian process is a nonparametric prior over functions completely specified by mean and covariance functions. The mean function, $\mu(x)$, is the prior expectation of $f(x)$, while the covariance function, $k(x, x')$, is a positive semidefinite kernel that defines the covariance between function values $f(x)$ and $f(x')$.

$$\mu(x) = \mathbb{E}[f(x)] \tag{7}$$
$$k(x, x') = \mathrm{cov}(f(x), f(x')) \tag{8}$$

Any finite collection of function values is normally distributed $[f(x_1)...f(x_p)] \sim \mathcal{N}(\mu(x), K)$ where $p \times p$ matrix $K_{i,j} = k(x_i, x_j)$. Thus we can draw samples from a Gaussian process at a finite set of points by sampling from a multivariate Gaussian distribution. In this paper we generally consider $\mu(x) = 0$ and concentrate on the covariance function. The choice of kernel is particularly important in Gaussian process applications since the kernel defines the types of correlations encoded in the Gaussian process. For example, a common kernel choice is a Radial Basis Function (RBF), also known as a Gaussian kernel,

$$k(x, x') = s^2 \exp[-(x - x')^T V^{-1}(x - x')/2] \tag{9}$$

where s^2 is the signal variance and V is a diagonal matrix of bandwidths. The RBF kernel implies that nearby values are more highly correlated. While this may be true in many applications, it would be inappropriate for data with significant periodicity. In such cases a periodic kernel would be more fitting. We consider more expressive kernel representations in section 3.2.2. This formulation of Gaussian processes naturally accommodates inputs x of arbitrary dimensionality.

Prediction with Gaussian processes Given a set of kernel hyperparameters, θ, and data, (x, y), we can derive a closed form expression for the predictive distribution of $f(x^*)$

evaluated at points x^*,

$$\begin{aligned} f(x^*)|\theta, x, y, x^* \sim \mathcal{N}\Big(& k(x^*, x)[k(x, x) + \sigma_\epsilon^2 I]^{-1}(y - \mu(x)) + \mu(x^*), \\ & k(x^*, x^*) - k(x^*, x)[k(x, x) + \sigma_\epsilon^2 I]^{-1} k(x, x^*) \Big) \end{aligned} \tag{10}$$

The predictive distribution provides posterior mean and variance estimates that can be used to define Bayesian credible sets. Thus Gaussian process prediction is useful both for estimating the value of a function at new points, x^*, and for deriving a function's distribution in the domain, x, for which we have data.

Learning Gaussian process hyperparameters In order to learn kernel hyperparameters we often desire to optimize the marginal likelihood of the data conditioned on the kernel hyperparameters, θ, and inputs, x.

$$p(y|\theta, x) = \int p(y|f, x)p(f|\theta)df \tag{11}$$

Thus we choose the kernel which maximizes the likelihood that the observed data is generated by the Gaussian process prior with hyperparameters θ. In the case of a Gaussian observation model we can express the log marginal likelihood as,

$$\log p(y|\theta, x) = -\frac{1}{2}\log|K + \sigma_\epsilon^2 I| - \frac{1}{2}(y - \mu(x))^T (K + \sigma_\epsilon^2 I)^{-1}(y - \mu(x)) + \text{constant} \tag{12}$$

However, solving linear systems and log determinants involving the $n \times n$ covariance matrix K which incurs $\mathcal{O}(n^3)$ computations and $\mathcal{O}(n^2)$ memory, for n training points, using standard approaches based on the Cholesky decomposition (Rasmussen and Williams, 2006). These computational requires are prohibitive for many applications, particularly in public policy — the focus of this paper — where it is normal to have more than few thousand training points. Accordingly, we develop alternative scalable inference procedures, presented in section 3.4, which enable tractability on much larger datasets.

3.2. Model specification

Change surface data consists of latent functions $f_1, \ldots, f_r$ defining r regimes in the data. The change surface defines the transitions between these functions. We could initially consider an input-dependent mixture model such as in Wilson et al. (2012),

$$y(x) = w_1(x)f_1(x) + \cdots + w_r(x)f_r(x) + \epsilon \tag{13}$$

where the weighting functions, $w_i(x) : \mathbb{R}^D \to \mathbb{R}^1$, describe the mixing proportions over the input domain. However, for data with changing regimes we are particularly interested in latent functions that exhibit some amount of mutual exclusivity.

We induce this partial discretization with $\sigma(z) : \mathbb{R}^r \to [0, 1]^r$. These functions have support over the entire real line, but a range in $[0, 1]$ and concentrated towards 0 and 1. Thus, each $w_i(x)$ in Eq. (13) becomes $\sigma_i(w(x))$, where $w(x) = [w_1(x), .., w_r(x)]$. Additionally, we choose $\sigma(z)$ such that it produces a convex combination over the weighting functions, $\sum_{i=1}^r \sigma_i(w(x)) = 1$. In this way, each $w_i(x)$ defines the strength of latent f_i over

the domain, while $\sigma(z)$ normalizes these weights to induce weak mutual exclusivity. Thus considering the general model of change surfaces in Eq. (1) we define each warping function as $s_i(x) = \sigma_i(w(x))$.

A natural choice for flexible change surfaces is to let $\sigma(z)$ be the softmax function. In this way the change surface can approximate a Heaviside step function, corresponding to the sharp transitions of standard changepoints, or more gradual changes. For r latent functions, the resulting warping function is:

$$s_i(x) = \sigma_i(w(x)) = \text{softmax}(w(x))_i = \frac{\exp(w_i(x))}{\sum_{j=1}^{r} \exp(w_j(x))} \tag{14}$$

The Gaussian process change surface (GPCS) model is thus

$$y(x) = \sigma_1(w(x))f_1(x) + \cdots + \sigma_r(w(x))f_r(x) + \epsilon \tag{15}$$

where each f_i is drawn from a Gaussian process. Importantly, we expect that each Gaussian process, $f_i(x)$, will have different hyperparameter values corresponding to different dynamics in the various regimes.

Since a sum of Gaussian processes is a Gaussian process, we can re-write Eq. (15) as $y(x) = f(x) + \epsilon$, where $f(x)$ has a single Gaussian process prior with covariance function,

$$k(x, x') = \sigma_1(w(x))k_1(x, x')\sigma_1(w(x')) + \cdots + \sigma_r(w(x))k_r(x, x')\sigma_r(w(x')) \tag{16}$$

In this form we can see that $\sigma_1(w(x)) \ldots \sigma_r(w(x))$ induce non-stationarity since they are dependent on the input x. Thus, even if we use stationary kernels for all k_i, GPCS observations follow a Gaussian process with a flexible, non-stationary kernel.

3.2.1. Design choices for $w(x)$

The functional form of $w(x)$ determines how changes can occur in the data, and how many can occur. For example, a linear parametric weighting function,

$$w(x) = \beta_0 + \beta_1^T x \tag{17}$$

only permits a single linear change surface in the data. Yet even this simple model is more expressive than discrete changepoints since it permits flexibility in the rate of change and extends to change regions in $\mathbb{R}^D$.

In order to develop a general framework, we introduce a flexible $w(x)$ that is formed as a finite sum of Random Kitchen Sink (RKS) features which map the D dimensional input x to an m dimensional feature space. We use RKS features from a Fourier basis expansion with Gaussian parameters and employ marginal likelihood optimization to learn the parameters of this expansion. Similar expansions have been used to efficiently approximate flexible non-parametric Gaussian processes (Lázaro-Gredilla et al., 2010; Rahimi and Recht, 2007).

Using m RKS features, $w(x)$ is defined as,

$$w(x) = \sum_{i=1}^{m} a_i \cos(\omega_i^T x_i + b_i) \tag{18}$$

where we initially sample,

$$a_i \sim \mathcal{N}(0, \frac{\sigma_0}{m} I) \tag{19}$$

$$\omega_i \sim \mathcal{N}(0, \frac{1}{4\pi^2} \Lambda^{-1}) \tag{20}$$

$$b_i \sim \text{Uniform}(0, 2\pi) \tag{21}$$

Initialization of hyperparameters σ_0 and diagonal matrix of length-scales, $\Lambda = \text{diag}(l_1^2, \ldots, l_D^2)$, is discussed in section 3.5.

Experts with domain knowledge can specify a parametric form for $w(x)$ other than RKS features. Such specification can be advantageous, requiring relatively few, highly interpretable parameters to optimize. For example, in an industrial setting where we are modeling failure of parts in a factory we could define $w(x)$ such that it was monotonically increasing since machine parts do not self-repair. This bias could take the form of a linear function as in Equation (17). Note that since parameters are learned from data, the functional form of $w(x)$ does not require prior knowledge about if or where changes occur.

3.2.2. Kernel specification

Each latent function is specified by a kernel with its own set of hyperparameters. By design, each k_i may be of a different form. For example, one function may have a Matérn kernel, another a periodic kernel, and a third an exponential kernel. Such specification is useful when domain knowledge provides insight into the covariance structure of the various regimes.

In order to maintain maximal generality and expressivity, we develop GPCS using multidimensional spectral mixture kernels (Wilson and Adams, 2013) where $x \in \mathbb{R}^D$.

$$k_{\text{SM}}(x, x') = \sum_{q=1}^{Q} \omega_q \cos(2\pi(x - x')^T \mu_q) \prod_{d=1}^{D} \exp(-2\pi^2 (x^{(d)} - x'^{(d)})^2 v_q^{(d)}) \tag{22}$$

This kernel is derived via spectral densities that are scale-location mixtures of Q Gaussians. Each component in this mixture has mean $\mu_q \in \mathbb{R}^D$, covariance matrix $\text{diag}(v_q^{(1)}, ..., v_q^{(D)})$, and signal variance parameter $\omega_q \in \mathbb{R}^1$. With a sufficiently large Q, spectral mixture kernels can approximate any stationary kernel, providing the flexibility to capture complex patterns over multiple dimensions. These kernels have been used in pattern prediction, outperforming complex combinations of standard stationary kernels (Wilson et al., 2014).

Previous work on Gaussian processes changepoint modeling has typically been restricted to RBF (Saatçi et al., 2010; Garnett et al., 2009) or exponential kernels (Majumdar et al., 2005). However, expressive covariance functions are particularly critical for modelling multidimensional and spatio-temporal data – a key application for change surfaces – where structure is often complex and unknown a priori.

Initializing and training expressive kernels is often challenging. We propose a practical initialization procedure in section 3.5, which can be used quite generally to help learn flexible kernels.

3.2.3. GPCS background model

Following section 2.1 we extend GPCS to the "GPCS background model." For this model we add a latent background function, $f_0(x)$, with an independent Gaussian process prior. Using the same choices for expressive $w(x)$ and covariance functions, we define the GPCS background model as,

$$y(x) = f_0(x) + \sigma_1(w(x))f_1(x) + \cdots + \sigma_{r-1}(w(x))f_{r-1}(x) + \epsilon \tag{23}$$

Recall that in this model we set $f_r(x) = 0$. Additionally, since we continue to enforce $\sum_{i=1}^{r} \sigma_i(w(x)) = 1$, thus $\sum_{i=1}^{r-1} \sigma_i(w(x)) \leq 1$.

This model effectively places different priors on the background and change regions, as opposed to the the standard GPCS model which places the same GP prior on each regime. The different priors in the GPCS background model reflect an intentional inductive bias which could be advantageous in certain domain settings, such as policy interventions, as discussed in section 2.1 above.

3.3. GPCS Counterfactual Prediction

We consider counterfactuals when using two latent functions in a GPCS, $f_1(x)$ and $f_2(x)$. This two-function setup addresses a typical setting for counterfactual prediction when considering two alternatives. The derivations below can be extended to multiple functional regimes. As discussed above, we note that change surface counterfactuals are only valid with respect to the regimes of the data as identified by GPCS. Subsequent analysis and domain expertise are necessary to make any further claims about the relationship between an identified change surface and some latent intervention.

In counterfactual prediction we wish to infer the value of $f_1(x)$ and $f_2(x)$ in the absence of the other function. Therefore we condition on the observations, (x, y), and GPCS model parameters in order to compute the conditional distribution $p\big([f_1(x), f_2(x)]|y\big)$ from the multivariate Gaussian joint distribution $p\big([f_1(x), f_2(x)], y\big)$. For notational convenience we omit explicit reference to the model parameters in the subsequent derivations but note that all distributions are conditional on these parameters.

To recall, for two latent functions, $f_1(x)$ and $f_2(x)$, GPCS specifies

$$y(x) = \sigma_1 f_1(x) + \sigma_2 f_2(x) + \epsilon \tag{24}$$
$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{25}$$
$$f_1(x) \sim \mathcal{GP}(0, K_1) \tag{26}$$
$$f_2(x) \sim \mathcal{GP}(0, K_2) \tag{27}$$

where for notational simplicity we let $K_1 = k_1(x, x')$, $K_2 = k_2(x, x')$, $\sigma_1 = \sigma_1(w(x))$, and $\sigma_2 = \sigma_2(w(x))$.

We consider the most general case when we want to predict counterfactuals for both $f_1(x)$ and $f_2(x)$ over the domain X. No restrictions are placed over X. It can include the entire original domain, parts of the original domain, or different inputs entirely. We concatenate $f(X)$ and $g(X)$ together,

$$u = [f(X), g(X)]\,. \tag{28}$$

Since in section 3.2 we assumed that $f_1(x)$ and $f_2(x)$ have independent Gaussian process priors, we know that,

$$u \sim \mathcal{N}\Big(0, \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}\Big) \tag{29}$$

Considering the observed data, y, we know that u and y are jointly Gaussian,

$$\begin{bmatrix} u \\ y \end{bmatrix} \sim \mathcal{N}\Big(0, \begin{bmatrix} \Sigma_{u,u} & \Sigma_{u,y} \\ \Sigma_{u,y}^T & \Sigma_{y,y} \end{bmatrix}\Big) \tag{30}$$

and using multivariate Gaussian identities, we find that u has the conditional Gaussian distribution

$$u|y \sim \mathcal{N}\Big(\Sigma_{u,y}\Sigma_{y,y}^{-1}y, \Sigma_{u,u} - \Sigma_{u,y}\Sigma_{y,y}^{-1}\Sigma_{u,y}^T\Big) \tag{31}$$

Thus in order to derive counterfactuals for both $f(X)$ and $g(X)$ we only need to compute $\Sigma_{u,y}, \Sigma_{y,y}$, and $\Sigma_{u,u}$. Note that with respect to $\Sigma_{u,u}$ we have already derived the covariance structure for u in Equation (29).

Computation for $\Sigma_{u,y}$ In order to compute $\Sigma_{u,y}$, we expand the multiplication noting that y is defined to be a two-function GPCS,

$$\Sigma_{u,y} = E[uy^T] \tag{32}$$
$$= \mathbb{E}[\begin{bmatrix} f_1(x_1) \\ \cdots \\ f_1(x_n) \\ f_2(x_1) \\ \cdots \\ f_2(x_n) \end{bmatrix} \begin{bmatrix} \sigma_1(x_1)f_1(x_1) + \sigma_2(x_1)f_2(x_1) + \epsilon \\ \cdots \\ \sigma_1(x_n)f_1(x_n) + \sigma_2(x_n)f_2(x_n) + \epsilon \end{bmatrix}^T] \tag{33}$$

Multiplying these elements is assisted by the following identities. Recall that kernels K_1 and K_2 define the covariance among function values in f and g respectively,

$$\mathbb{E}[f_1(x_i)f_1(x_j)] = k_1(i, j) \tag{34}$$
$$\mathbb{E}[f_2(x_i)f_2(x_j)] = k_2(i, j) \tag{35}$$

Additionally, since $f_1(x)$ and $f_2(x)$ have independent Gaussian process priors, $\mathbb{E}[f_1(x_i)f_2(x_j)] = 0$. Furthermore, because ϵ is distributed with mean zero, $\mathbb{E}[\epsilon_i] = 0$. Finally, since $\sigma_1(x)$ and $\sigma_2(x)$ are constant (conditional on hyperparameters) $\mathbb{E}[\sigma_1(x_i)] = \sigma_1(x_i)$ and $\mathbb{E}[\sigma_2(x_i)] =$

$\sigma_2(x_i)$. Thus we can conclude that

$$\Sigma_{u,y} = \begin{bmatrix} \sigma_1(x_1)k_1(1,1) & \sigma_1(x_2)k_1(1,2) & ... & \sigma_1(x_n)k_1(1,n) \\ \sigma_1(x_1)k_1(2,1) & \sigma_1(x_2)k_1(2,2) & ... & \sigma_1(x_n)k_1(2,n) \\ ... \\ \sigma_1(x_1)k_1(n,1) & \sigma_1(x_2)k_1(n,2) & ... & \sigma_1(x_n)k_1(n,n) \\ \sigma_2(x_1)k_2(1,1) & \sigma_2(x_2)k_2(1,2) & ... & \sigma_2(x_n)k_2(1,n) \\ \sigma_2(x_1)k_2(2,1) & \sigma_2(x_2)k_2(2,2) & ... & \sigma_2(x_n)k_2(2,n) \\ ... \\ \sigma_2(x_1)k_2(n,1) & \sigma_2(x_2)k_2(n,2) & ... & \sigma_2(x_n)k_2(n,n) \end{bmatrix} \tag{36}$$

$$= \begin{bmatrix} K_1 \odot \mathbb{1}\sigma_1^T \\ K_2 \odot \mathbb{1}\sigma_2^T \end{bmatrix} \tag{37}$$

where $\odot$ is elementwise multiplication.

Computation for $\Sigma_{y,y}$ The computation for $\Sigma_{y,y}$ is very similar to that of $\Sigma_{u,y}$ so we omit its expansion for the sake of brevity. The slight difference is that we must consider $\mathbb{E}[\epsilon_i \epsilon_i]$ which equals σ_ϵ^2.

Thus,

$$\Sigma_{y,y} = E[yy^T] \tag{38}$$

$$= K_1 \odot [\sigma_1 \sigma_1^T] + K_2 \odot [\sigma_2 \sigma_2^T] + I_n \sigma_\epsilon^2 \tag{39}$$

3.3.1. GPCS background model counterfactuals

The counterfactual derivations above directly apply to the GPCS background model with $r = 2$, where $y(x) = f_0(x) + \sigma_1(w(x))f_1(x)$. Recall that as we discussed in section 2.1, this is a special case of the GPCS background model where $f_1(x)$ is an additive change function. In this case, the counterfactual for $f_0(x)$ estimates what would have occurred in the absence of the identified change. The counterfactual for $f_1(x)$ models how the change would have affected the entire domain.

If we let $u = [f_0(X), f_1(X)]$ we can derive counterfactuals for the GPCS background model by setting $\sigma_0 = 1$ in the equations for $\Sigma_{u,u}$, $\Sigma_{u,y}$, and $\Sigma_{y,y}$ above. Explicitly,

$$\Sigma_{u,u} = \begin{bmatrix} K_0 & 0 \\ 0 & K_1 \end{bmatrix} \tag{40}$$

$$\Sigma_{u,y} = \begin{bmatrix} K_0 \\ K_1 \odot \mathbb{1}\sigma_1^T \end{bmatrix} \tag{41}$$

$$\Sigma_{y,y} = K_0 + K_1 \odot [\sigma_1 \sigma_1^T] + I_n \sigma_\epsilon^2 \tag{42}$$

3.4. Scalable inference

Analytic optimization and inference for Gaussian processes requires computation of the log marginal likelihood from Eq. (12). Yet solving linear systems and computing log determinants over $n \times n$ covariance matrices, using standard approaches such as the Cholesky decomposition, requires $O(n^3)$ computations and $O(n^2)$ memory, which is impractical for large datasets. Recent advances in scalable Gaussian processes (Wilson, 2014) have reduced

this computational burden by exploiting Kronecker structure under two assumptions: (1) the inputs lie on a grid formed by a Cartesian product, $x \in X = X^{(1)} \times ... \times X^{(D)}$; and, (2) the kernel is multiplicative across each dimension. Multiplicative kernels are commonly employed in spatio-temporal Gaussian process modeling (Martin, 1990; Majumdar et al., 2005; Flaxman et al., 2015), corresponding to a soft a priori assumption of independence across input dimensions, without ruling out posterior correlations. The popular RBF and ARD kernels, for instance, already have this multiplicative structure. Under these assumptions, the $n \times n$ covariance matrix $K = K_1 \otimes \cdots \otimes K_D$, where each K_d is $n_d \times n_d$ such that $\prod_1^D n_d = n$.

Using efficient Kronecker algebra, Saatçi (2011) shows how one can solve linear systems and compute log determinants in $O(Dn^{\frac{D+1}{D}})$ operations using $O(Dn^{\frac{2}{D}})$ memory. Furthermore, Wilson et al. (2014) extends the Kronecker methods for incomplete grids. Yet for additive compositions of kernels, such as those needed for change surface modeling in Eq. (16), the resulting sum of matrix Kronecker products does not decompose as a Kronecker product. Thus, the standard Kronecker approaches for scalable inference and learning are inapplicable. Instead, solving linear systems for the kernel inverse can be efficiently carried out through linear conjugate gradients as in Flaxman et al. (2015) that only rely on matrix vector multiplications, which can be performed efficiently with sums of Kronecker matrices.

However, there is no exact method for efficient computation of the log determinant of the sum of Kronecker products. Instead, Flaxman et al. (2015) upper bound the log determinant using the Fiedler bound (Fiedler, 1971) which says that for $n \times n$ Hermitian matrices A and B with sorted eigenvalues $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_n$ respectively,

$$\log(|A + B|) \leq \sum_{i=1}^{n} \log(\alpha_i + \beta_{n-i+1}) \tag{43}$$

While efficient, the Fiedler bound does not generalize to more than two matrices.

3.4.1. Weyl bound

In order to achieve scalable computations for an arbitrary additive composition of Kronecker matrices, we propose to bound the log determinant of the sum of multiple covariance matrices using Weyl's inequality (Weyl, 1912) which states that for $n \times n$ Hermitian matrices, $M = A + B$, with sorted eigenvalues $\mu_1, \ldots, \mu_n$, $\alpha_1, \ldots, \alpha_n$, and $\beta_1, \ldots, \beta_n$ respectively,

$$\mu_{i+j-1} \leq \alpha_i + \beta_j \quad \forall i, j \geq 1 \tag{44}$$

Since $\log(|A + B|) = \log(|M|) = \sum_{i=1}^{n} \log(\mu_i)$ we can bound the log determinant by $\sum_{i+j-1=1}^{n} \log(\alpha_i + \beta_j)$. Furthermore, we can use the Weyl bound iteratively over pairs of matrices to bound the sum of r covariance matrices $K_1, \ldots, K_r$.

As the bound indicates, there is flexibility in the choice of which eigenvalue pair $\{\alpha_i, \beta_j\}$ to use for bounding μ_{i+j-1}. Thus for each eigenvalue, μ_k, we wish to choose i, j that minimizes $\alpha_i + \beta_j$ subject to $k = i + j - 1$. One might be tempted to minimize over all possible pairs for each eigenvalue, $\mu_1, \ldots, \mu_n$, in order to obtain the tightest bound on the log determinant. Unfortunately, such a procedure requires $O(n^2)$ computations. Instead we explore two possible alternatives:

1. For each μ_{i+j-1} we choose the "middle" pair, $\{\alpha_i, \beta_j\}$, such that $i = j$ when possible, and $i = j + 1$ otherwise. This "middle" heuristic requires $O(n)$ computations.

2. We employ a greedy search to choose the minimum of v possible pairs of eigenvalues. Using the previous i' and j', we consider $\{\alpha_i, \beta_j\}$ for all $i = i' - \frac{v}{2}, ..., i' + \frac{v}{2}$ and the corresponding j values. Setting $v = 1$ corresponds to the middle heuristic. Setting $v = n$ corresponds to the exact Weyl bound. The greedy search requires $O(vn)$ computations.

In addition to bounding the sum of kernels, we must also deal with the scaling functions, $\sigma_i(w(x))$. We can rewrite Eq. (16) in matrix notation,

$$K = S_1 K_1 S_1' + \cdots + S_r K_r S_r' \tag{45}$$

where $S_i = \text{diag}(\sigma_i(w(x)))$ and $S_i' = \text{diag}(\sigma_i(w(x')))$. Employing the bound on eigenvalues of matrix products (Bhatia, 2013),

$$\text{sort}(\text{eig}(AB)) \leq \text{sort}(\text{eig}(A))\text{sort}(\text{eig}(B)) \tag{46}$$

we can bound the log determinant of K in Eq. (45) with an iterative Weyl approximation over $[\{s_{i,l} k_{i,l} s_{i,l}'\}_{l=1}^{n}]_{i=1}^{r}$ where $s_{i,l}$, $k_{i,l}$, and $s_{i,l}'$ are the lth largest eigenvalue of S_i, K_i, and S_i' respectively.

We empirically evaluate the exact Weyl bound, middle heuristic, and greedy search with $v = 80$ pairs of eigenvalue indexes to search above and below the previous index. All experiments are evaluated using GPCS with synthetic data generated according to the procedure in section 4.1. We also compare these results against the Fiedler bound in the case of two kernels.

Figure 5 depicts the ratio of each approximation to the true log determinant, and the time to compute each approximation over increasing number of observations for two kernels. While the Fiedler approximation is more accurate than any Weyl approach, all approximations perform quite similarly (note the fine grained axis scale) and converge to ≈ 0.85 of the true log determinant. In terms of computation time, the exact Weyl bound scales poorly with data size as expected. Yet both approximate Weyl bounds scale well. In practice, we use the middle heuristic described above, since it provides the fastest results, nearly equivalent to the Fiedler bound.

Figure 6 depicts the same quantities as Figure 5 but using three additive kernels. Since the Fiedler approximation is only valid for two kernels it is excluded from these plots. While the log determinant approximation ratios are less accurate for small datasets, as the data size increases all Weyl approximations converge to ≈ 0.8.

In addition to enabling scalable change surface kernels, the Weyl bound method permits scalable additive kernels in general. When applied to the spatio-temporal domain this yields the first scalable Gaussian process model which is non-separable in space and time.

3.4.2. Massively Scalable Inference

We further extend the scalability and flexibility of the Weyl bound method by leveraging a structured kernel interpolation methodology from the KISS-GP framework (Wilson and

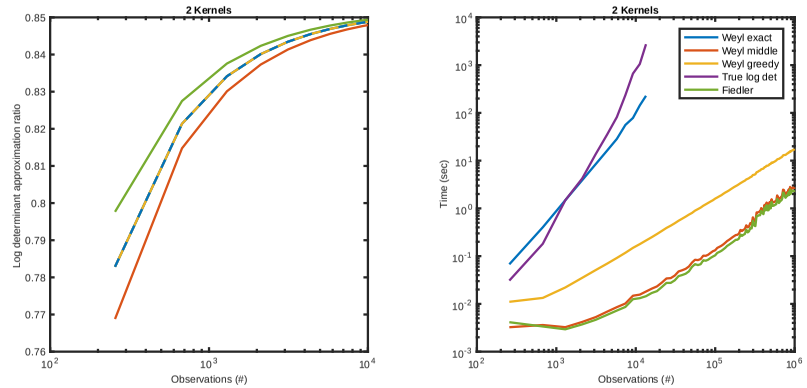


Figure 5: Left plot shows the ratio of log determinant approximations to the true log determinant of two additive kernels. Note that the y-axis is scaled to a relatively narrow band. The dashed line indicates that both the Weyl exact and Weyl greedy method performed similarly. Right plot shows the time to compute each approximation and the true log determinant.

Nickisch, 2015). Although many spatio-temporal policy relevant applications naturally have near-grid structure, such as readings over a nearly dense set of latitudes, longitudes, and times, this integration with KISS-GP further relaxes the dependencies on grid assumptions. The resulting approach scales to much larger problems by interpolating data to a smaller, user-defined grid. In particular, with local cubic interpolation, the error in the kernel approximation is upper bounded $O(1/m^3)$ for m latent grid points, and m can be very large because the kernel matrices in this space are structured. These scalable approaches are thus very generally applicable as demonstrated in an extensive range of previously published experiments in Wilson et al. (2016b,a) based on these techniques. Additionally, KISS-GP enables the Weyl bound approximation methods to apply to arbitrary, non-grid data.

We empirically demonstrate the advantages of integration with KISS-GP by evaluating an additive GPCS on the two-dimensional data described above. Although the original data lies on a grid, we use KISS-GP interpolation to compute the negative log likelihood on four grids of increasingly smaller size. Figure 7 depicts the negative log likelihood and the computation time for these experiments using the Weyl middle heuristic. The plot legend indicates the size of the induced grid size. For example, 'KISS-GP 75%' is 75% the size of the original grid. Note that the time and log likelihood scales in Figure 7 are different from those in Figures 5 and 6 since we are now computing full inference steps as opposed to just computing the log determinant. The results indicate that with minimal error in negative log likelihood accuracy we can substantially reduce the time for inference.

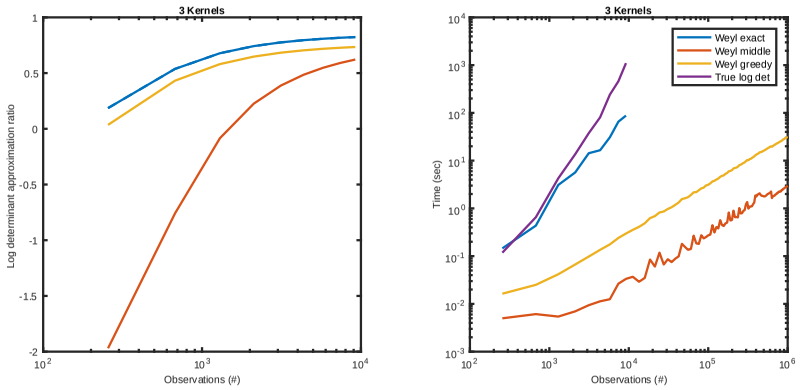


Figure 6: Left plot shows the ratio of approximations to the true log determinant of 3 additive kernels. Note that the y-axis has a much larger scale than in Figure 5. Right plot shows the time to compute each approximation and the true log determinant of 3 additive kernels.

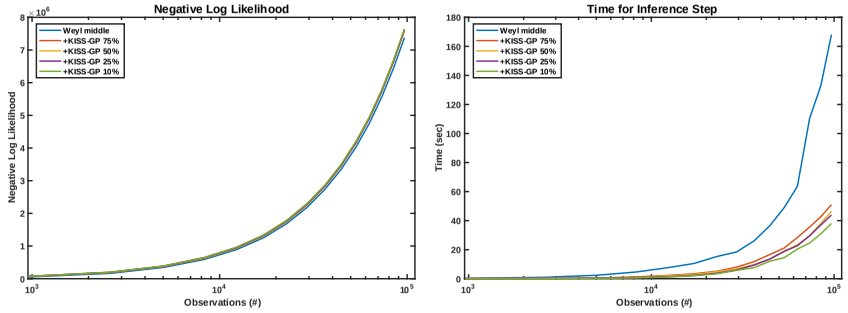


Figure 7: Plots showing negative log likelihood and time for inference on two additive kernels using the Weyl bound on grids of decreasing size. For example, 'KISS-GP 75%' computes the Weyl middle bound on a grid which is 75% the size of the original grid used to compute the first line.

3.5. Initialization

Since GPCS uses flexible spectral mixture kernels, as well as RKS features for the change surface, the parameter space is highly multimodal. Therefore, it is essential to initialize the model hyperparameters appropriately. Below we present an approach where we first initialize the $w(x)$ RKS features and then use those values in a novel initialization method for

the spectral mixture kernels. Like most GP optimization problems, GPCS hyperparameter optimization is non-convex and there are no provable guarantees that the proposed initialization will result in optimal solutions. However, it is our experience that this initialization procedure works well in practice for the GPCS as well as spectral mixture kernels in general.

To initialize $w(x)$ defined by RKS features we first simplify the change surface model by assuming that each latent function, $f_1, ..., f_r$, from Eq. (15) is drawn from a Gaussian process with an RBF kernel. Since RBF kernels have far fewer hyperparameters than spectral mixture kernels, starting with RBF kernels helps our approach find good starting vaglues for $w(x)$. Algorithm 1 provides the procedure for initializing this simplified change surface model. Note that depending on the application domain, a model with latent functions defined by RBF kernels may be sufficient as a terminal model.

Algorithm 1 Initialize RKS $w(x)$ by optimizing a simplified model with RBF kernels

for $i = 1 : m_1$ **do**
  Draw a, ω, b for RKS features in $w(x)$
  Draw m_2 sets of hyperparameter values for RBF kernels, $\{\theta_1, ..., \theta_{m_2}\}$
  Choose the best hyperparameter set, $\theta^{(i)} = \texttt{max-likelihood}(\theta_1, ..., \theta_{m_2})$
  Partial optimization of $\{a, \omega, b, \theta\} \rightarrow \Theta^{(i)}$
end for
Choose the best set of hyperparameters, $\Theta = \texttt{max-likelihood}(\Theta^{(1)}, ..., \Theta^{(m_1)})$
Optimize Θ until convergence

In the algorithm, we test multiple possible sets of values for $w(x)$ by drawing the hyperparameters a, ω, and b from their respective prior distributions (see section 3.2.1) m_1 number of times. We set reasonable values for hyperparameters in those prior distributions. Specifically, we let $\Lambda = (\frac{\text{range}(x)}{2})^2$, $\sigma_0 = \text{std}(y)$, and $\sigma_n = \frac{\text{mean}(|y|)}{10}$. These choices are similar to those employed in Lázaro-Gredilla et al. (2010).

For each sampled set of $w(x)$ hyperparameters, we sample m_2 sets of hyperparameters for the RBF kernels and select the set with the highest marginal likelihood. Then we run an abbreviated optimization procedure over the combined $w(x)$ and RBF hyperparameters and select the joint set that achieves the highest marginal likelihood. Finally, we optimize the resulting hyperparameters until convergence.

In order to initialize the spectral mixture kernels, we use the initialized $w(x)$ from above to define the subset $\{x : \sigma_i(w(x)) > 0.5\}$ where each latent function, f_i from Eq. (15), is dominant. We then take a Fourier transform of $y(x)$ over each dimension, $x^{(d)}$, of $\{x : \sigma_i(w(x)) > 0.5\}$ to obtain the empirical spectrum in that dimension. Note that we consider each dimension of x individually since we have a multiplicative Q-component spectral mixture kernel over each dimension (Wilson, 2014). Since spectral mixture kernels model the spectral density with Q Gaussians on $\mathbb{R}^1$, we fit a 1-dimensional Gaussian mixture model,

$$p(x) = \sum_{q=1}^{Q} \phi_q \mathcal{N}(\mu_q, v_q) \tag{47}$$

to the empirical spectrum for each dimension. Using the learned mixture model we initialize the parameters of the spectral mixture kernels for $f_i(x)$.

Algorithm 2 Initialize spectral mixture kernels

for $k_i : i = 1 : r$ **do**
  for $d = 1 : D$ **do**
    Compute $x^{(d)} \in \{x : \sigma_i(w(x)) > 0.5\}$
    Sample $s \sim |\text{FFT}(\text{sort}(y(x^{(d)})))|^2$
    Fit Q component GMM as $p(s) = \sum_{q=1}^{Q} \phi_q^{(d)} \mathcal{N}(\mu_q^{(d)}, v_q^{(d)})$
    Initialize $\omega_q = \text{std}(y(x^{(d)})) * \phi_q$
  end for
end for

After initializing $w(x)$ and spectral mixture hyperparameters, we jointly optimize the entire model using marginal likelihood and non-linear conjugate gradients (Rasmussen and Nickisch, 2010).

4. Experiments

We demonstrate the power and flexibility of GPCS by applying the model to a variety of numerical simulations and complex human settings. We begin with 2-dimensional numerical data in section 4.1, and show that GPCS is able to correctly model out-of-class polynomial change surfaces, and that it provides higher accuracy regressions than other comparable methods. Additionally we compute highly accurate counterfactual predictions for both GPCS and GPCS background models and discuss how the posterior distribution varies over the prediction domain as a function of the change surface.

We next consider coal mining, epidemiological, and urban policy data to provide additional analytical evidence for the effectiveness of GPCS and to demonstrate how GPCS results can be used to provide novel policy-relevant and scientifically-relevant insights. The ground truth against which GPCS is evaluated are the domain specific interventions in these case studies.

In order to compare GPCS to standard changepoint models, we use a 1-dimensional dataset on the frequency of coal mining accidents. After fitting GPCS, we show that the change surface is able to identify a region of change similar to other changepoint methods. However, unlike changepoint methods that only identify a single moment of change, GPCS models how the data changes over time.

We then employ GPCS to analyze two complex spatio-temporal settings involving policy and scientific questions. First we examine requests for residential lead testing kits in New York City between 2014-2016, during a time of heightened concern about lead-tainted water. GPCS identifies a spatially and temporally varying change surface around the period when issues of water contamination were being raised in the news. We conduct a regression analysis on the resulting change surface features to better understand demographic factors that may have affected residents' concerns about lead-tainted water.

Second, we apply GPCS to model state-level measles incidence in the United States during the twentieth century. GPCS identifies a substantial change around the introduction of the measles vaccine in 1963. However, the shape of the change surface varies over time for each state, indicating possible spatio-temporal heterogeneity in the adoption and effectiveness of the vaccination program during its initial years. We use regression analysis on the change surface features to explore possible institutional and demographic factors that may have influenced the impacts of the measles vaccination program. Finally, we estimate the counterfactual of measles incidence without vaccination by filtering out the detected change function and examining the inferred latent background function.

4.1. Numerical Experiments

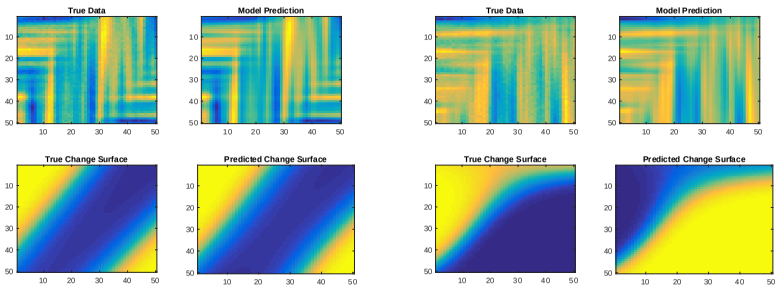
We generate a 50×50 grid of synthetic data by drawing independently from two latent functions, $f_1(x)$ and $f_2(x)$. Each function is characterized by an independent Gaussian process with a two-dimensional RBF kernel of different length-scales and signal variances. The synthetic change surface between the functions is defined by $\sigma(w_{\text{poly}}(x))$ where $w_{\text{poly}}(x) = \sum_{i=0}^{3} \beta_i^T x^i$, $\beta_i \sim \mathcal{N}(0, 3I_D)$. We chose a polynomial change surface because it generates complex change patterns but is out-of-class when we use RKS features for $w(x)$, thus testing the robustness of GPCS to change surface misspecification.

4.1.1. GPCS model

Using the synthetic data generation technique described above we simulate data as $y = \sigma(w_{poly}(x))f_1(x) + (1 - \sigma(w_{\text{poly}}(x)))f_2(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. We apply GPCS with two latent functions, spectral mixture kernels, and $w(x)$ defined by RKS features. We do not provide the model with prior information about the change surface or latent functions. As emphasized in section 3.5, successful convergence is dependent on reasonable initialization. Therefore, we use $m_1 = 100$ and $m_2 = 20$ for Algorithm 1. Figure 8 depicts two typical results using the initialization procedure followed by analytic optimization. The model captures the change surface and produces an appropriate regression over the data. Note that in Figure 8b the predicted change surface is flipped since the order of functions is not important in GPCS.

To demonstrate that the initialization method from section 3.5 provides consistent results, we consider a numeric example and run GPCS 30 times with different random seeds. Figure 9 provides the true data and change surface as well as the mean and standard deviation over the 30 experimental results using the section 3.5 initialization procedure. For the predicted change surface we manually normalized the orientation of the change surface before computing summary statistics. The results illustrate that the initialization procedure provides accurate and consistent results for both y and $\sigma(w(x))$ across multiple runs. Indeed, when we repeat these experiments with random initialization, instead of the procedure from section 3.5, the MSE between the predicted and true change surface is 58% greater than when using our initialization procedure. Additionally, the results have a 17% larger standard deviation of $\sigma(w(x_i))$ over the 30 runs, demonstrating that the procedure we propose provides more consistent and accurate results.

Using synthetic data, we create a predictive test by splitting the data into training and testing sets. We compare GPCS to three other expressive, scalable methods: sparse spectrum Gaussian process with 500 basis functions (Lázaro-Gredilla et al., 2010), sparse spectrum Gaussian process with fixed spectral points with 500 basis functions (Lázaro-Gredilla et al., 2010), and a Gaussian process with multiplicative spectral mixture kernels in each dimension. For each method we average the results for 10 random restarts. For



(a) (b)

Figure 8: Two numerical data experiments. In each of (a) and (b) the top-left plot depicts the data (e.g., observations indexed by two dimensional spatial inputs); the bottom-left shows the true change surface with the range from blue to yellow depicting $\sigma_1(w(x))$. The top-right depicts the predicted output; the bottom-right shows the predicted change surface. Note that the predicted change surface in plot (b) is flipped since the order of functions is not important.

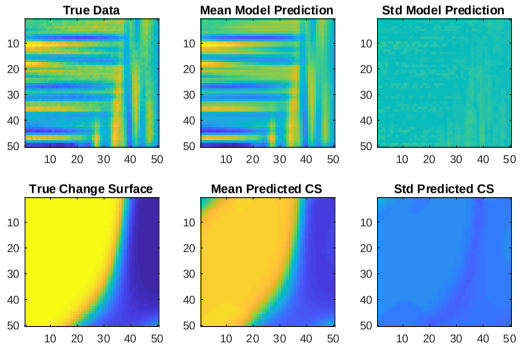


Figure 9: Consistency results across 30 runs with different random seeds. True data and change surface are on the left, while the mean and standard deviation of the predicted results are in center and right panels.

each method Table 2 shows the normalized mean squared error (NMSE),

$$\text{NMSE} = \frac{\|y_{test} - y_{pred}\|_2^2}{\|y_{test} - \bar{y}_{train}\|_2^2} \tag{48}$$

Table 2: Comparison of prediction accuracy (normalized mean squared error) using flexible and scalable Gaussian process methods on synthetic multidimensional change-surface data.

Method	**NMSE**
GPCS	0.00078
SSGP	0.01530
SSGP fixed	0.02820
Spectral mixture	0.00200

where $\bar{y}_{train}$ is the mean of the training data.

GPCS performed best due to the expressive non-stationary covariance function that fits to the different functional regimes in the data. Although the other methods can flexibly adapt to the data, they must account for the change in covariance structure by setting a shorter global length-scale over the data, thus underestimating the correlation of points in each regime. Thus their predictive accuracy is lower than GPCS, which can accommodate changes in covariance structure across the boundary of a change surface while retaining reasonable covariance length-scales within each regime.

We use GPCS to compute counterfactual predictions on the numerical data. In the previous experiments we used the data, (x, y), to fit the parameters of GPCS, θ. Now we condition on (x, y, θ) to infer the individual latent functions $f_1(x)$ and $f_2(x)$ over the entire domain, x. By employing the marginalization procedure described in section 3.3 we derive posterior distributions for both $f_1(x)$ and $f_2(x)$. Since we have synthetic data we can then compare the counterfactual predictions to the true latent function values. Specifically, we use (x, y, θ) from Figure 8b to infer the posterior counterfactual mean and variance for both $f_1(x)$ and $f_2(x)$ and show the results in Figure 10. Note how the posterior mean predictions of $f_1(x)$ and $f_2(x)$ are quite similar to the true values. Moreover, the posterior uncertainty estimates are very reasonable. For both $f_1(x)$ and $f_2(x)$ the posterior variance varies over the two-dimensional domain, x, as a function of the change surface. Where $s_1(x) \approx 1$ the posterior variance of $f_1(x) \approx 0$ while the posterior variance of $f_2(x)$ is large. In areas where $s_2(x) \approx 1$ the posterior variance of $f_1(x)$ is large, while the posterior variance of $f_2(x) \approx 0$. The uncertainty is also evident in the squared error, $\frac{1}{n}\sum(f_i(x) - \hat{f_i(x)})$, where, as expected, each function has larger error in areas of high posterior variance.

As discussed in section 3, the underlying probabilistic Gaussian process model behind GPCS automatically *discourages* extraneous complexity, favoring the simplest explanations consistent with the data (MacKay, 2003; Rasmussen and Ghahramani, 2001; Rasmussen and Williams, 2006; Wilson et al., 2014; Wilson, 2014). This property enables GPCS to discover interpretable generative hypothesis for the data, which is crucial for public policy applications. This Bayesian Occam's razor principle is a cornerstone of many probabilistic approaches, such as automatically determining the intrinsic dimensionality for probabilistic PCA (Minka, 2001), or hypothesis testing for Bayesian neural network architectures (MacKay, 2003). In the absence of such automatic complexity control, these methods would always favour the highest intrinsic dimensionality or the largest neural network respectively.

To demonstrate this Occam's razor principle in our context, we generate numeric data from a single GP without any change surface by setting $\sigma(w_{\text{poly}}(x)) = 0$, and fit a *misspec-*

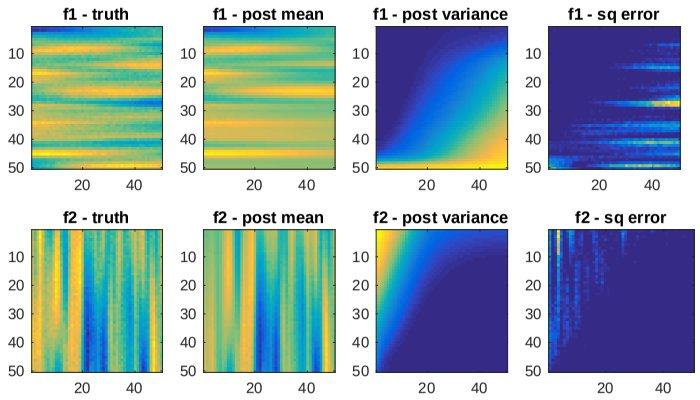
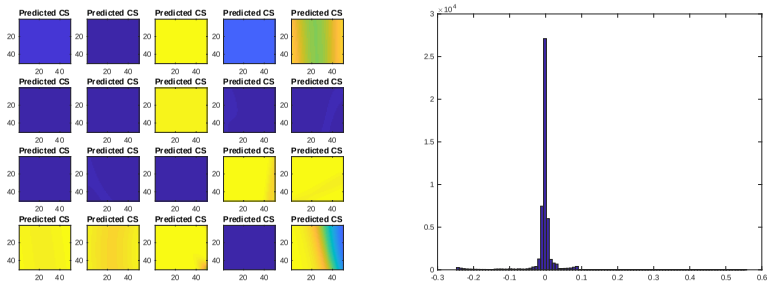


Figure 10: Posterior counterfactual predictions using hyperparameters derived from GPCS model. We plot the true latent function as well as the posterior mean and variance estimates for each function. Additionally, we plot the squared error between the true and posterior mean values.

ified GPCS model assuming two latent regimes. Figure 11 depicts the predicted change surfaces for 20 experiments of such data. The left panel illustrates pictorially that the



(a) (b)

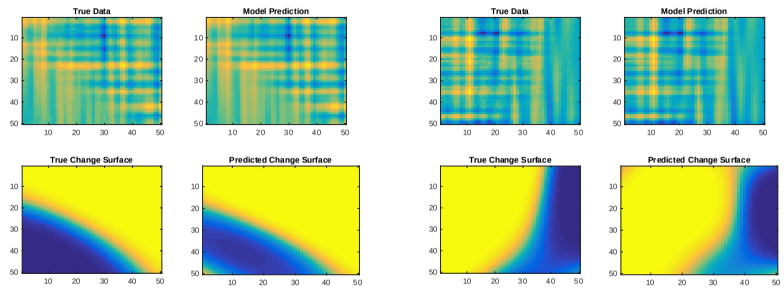
Figure 11: Data without any change surface, $\sigma(w_{\text{poly}}(x)) = 0$. The left panel depicts $\sigma_1(w(x))$ for each experiment. The right panel provides a histogram of the mean centered change surfaces values, $\sigma_1(w(x)) - \sum_{i \in n} \sigma_1(w(x_i))$.

change surfaces are nearly all flat at either $\sigma_1(w(x)) = 0$ or $\sigma_1(w(x)) = 1$ for these experiments. Specifically, $\text{std}[\sigma_1(w(x))] < 0.03$ for all but two runs. This finding indicates that GPCS discovers that no dynamic transition exists and does not overfit the data, despite the added flexibility afforded by multiple mixture components. Only one of the 20 results (bottom-right) indicates a change, and even in that case the magnitude of the transition is markedly subdued as compared to the results in Figures 8 and 12. While the upper-right result appears to have a large transition, in fact it has a flat change surface with $\text{std}[\sigma_1(w(x))] = 0.07$. The right panel provides a histogram of the mean centered change surface values for all experiments, $\sigma_1(w(x)) - \sum_{i \in n} \sigma_1(\omega(x_i))$, again demonstrating that GPCS learns very flat change surfaces and does not erroneously identify a change.

4.1.2. GPCS background model

We test the GPCS background model with a similar setup. Using the synthetic data generation technique described above, we simulate data as $y = f_0(x) + \sigma(w_{\text{poly}}(x)) f_1(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. We again note that the polynomial change surface is out-of-class.

We apply the GPCS background model with one background function and one latent function scaled by a change surface. Both Gaussian process priors use spectral mixture kernels, and $w(x)$ is defined by RKS features. We do not provide the model with prior information about the change surface or latent functions. Figure 12 depicts two typical results using the initialization procedure followed by analytic optimization. The model captures the change surface and produces an appropriate regression over the data.



(a) (b)

Figure 12: Two numerical data experiments. In each of (a) and (b) the top-left plot depicts the data; the bottom-left shows the true change surface with the range from blue to yellow depicting $\sigma_1(w(x))$. The top-right depicts the predicted output; the bottom-right shows the predicted change surface.

We use the GPCS background model to compute counterfactual predictions on the data from Figure 12b. Conditioning on (x, y, θ) we employ the marginalization procedure described in section 3.3 to infer posterior distributions for the background function, $f_0(x)$, and the change function, $f_1(x)$, over the entire domain, x. The results are shown in Figure

13. Note how the posterior mean predictions of both the background and change functions

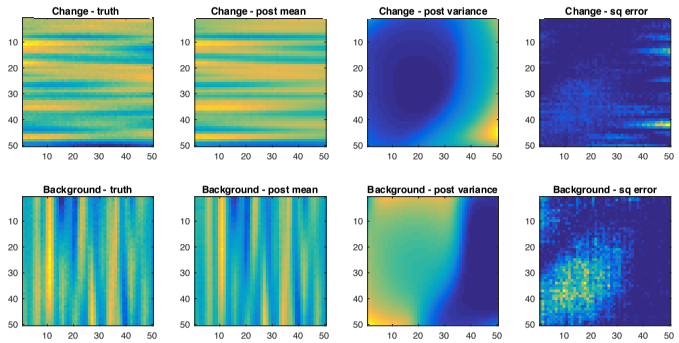


Figure 13: Posterior counterfactual predictions using hyperparameters derived from GPCS background model. We plot the true latent function as well as the posterior mean and variance estimates for each function. Additionally, we plot the squared error between the true and posterior mean values.

are quite similar to the true values. As in the case of GPCS, the posterior variance for each function varies over the two-dimensional domain, x, as a function of the change surface, $\sigma(w_{\text{poly}}(x))$.

4.1.3. Log Gaussian Cox Process

The numerical experiments above demonstrate the consistency of GPCS for identifying out-of-sample change surfaces and modeling complex data for high accuracy prediction. To further demonstrate the flexibility of the model, we apply GPCS to data generated by a log-Gaussian Cox process (Møller et al., 1998; Flaxman et al., 2015). This inhomogeneous Poisson process is modulated by a stochastic intensity defined as a GP,

$$\lambda = f \tag{49}$$

$$f \sim \mathcal{GP}(\mu, K) \tag{50}$$

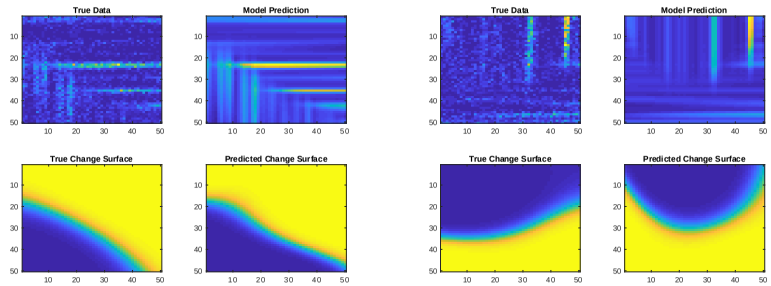
Conditional on λ, and letting s denote a region in space-time, the resulting small-area count data are non-negative integers distributed as

$$y(s) \mid \lambda \sim \text{Poisson}\big(\exp \int_s \lambda(x)dx\big). \tag{51}$$

We let this GP model be a convex combination of two GPs with an out-of-sample change surface, as described in section 4.1. Thus we generated data from this model as

$$y \mid f_1(x_i), f_2(x_i) \sim \text{Poisson}\Big(\exp\big[\sigma(w_{poly}(x))f_1(x) + (1-\sigma(w_{\text{poly}}(x)))f_2(x) + \epsilon\big]\Big). \tag{52}$$

Such data substantially departs from the type of data that GPCS is designed to model. Indeed, while custom approaches are often created to handle inhomogeneous Poisson data (Flaxman et al., 2015; Shirota and Gelfand, 2016), we use GPCS to demonstrate its flexibility and applicability to complex non-Gaussian data. The results are shown in Figure 14. The



(a) (b)

Figure 14: Two numerical data experiments with data from a log-Gaussian Cox process. In each of (a) and (b) the top-left plot depicts the data (e.g., observations indexed by two dimensional spatial inputs); the bottom-left shows the true change surface with the range from blue to yellow depicting $\sigma_1(w(x))$. The top-right depicts the predicted output; the bottom-right shows the predicted change surface.

model provides accurate change surfaces and predictions even though the data is substantially out-of-class – even beyond the out-of-class change surface data from sections 4.1.1 and 4.1.2. The precise location of change surfaces deviates slightly in GPCS, particularly on the left edge of Figure 14b where the raw data is highly stochastic. Additionally, the model predictions are smoothed versions of the true latent data, which reflects the fundamental difference between Gaussian and Poisson models.

4.2. British Coal Mining Data

British coal mining accidents from 1861 to 1962 have been well studied as a benchmark in the point process and changepoint literature (Raftery and Akman, 1986; Carlin et al., 1992; Adams and MacKay, 2007). We use yearly counts of accidents from Jarrett (1979). Adams and MacKay (2007) indicate that the Coal Mines Regulation Act of 1887 affected the underlying process of coal mine accidents. This act limited child labor in mines, detailed inspection procedures, and regulated construction standards (Mining, 2017). We apply GPCS to show that it can detect changes corresponding to policy interventions in data while providing additional information beyond previous changepoint approaches.

We consider GPCS with two latent functions, spectral mixture kernels, and $w(x)$ defined by RKS features. We do not provide the model with prior information about the 1887 legislation date. Figure 15 depicts the cumulative data and predicted change surface. The

Table 3: Comparing methods for estimating the date of change in coal mining data.

Method	Estimated date
GPCS $\sigma(w(x)) = 0.5$	1888.8
PELT mean change	1886.5
PELT variance change	1882.5
ecp	1887.0
Student-t test	1886.5
Bartlett test	1947.5
Mann-Whitney test	1891.5
Kolmogorov-Smirnov test	1896.5

red line marks the year 1887 and the magenta line marks $x : \sigma(w(x)) = 0.5$. GPCS correctly identified the change region and suggests a gradual change that took 5.6 years to transition from $\sigma(w(x)) = 0.25$ to $\sigma(w(x)) = 0.75$.

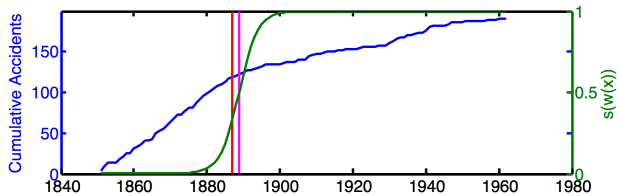


Figure 15: British coal mining accidents from 1851 to 1962. The blue line depicts cumulative annual accidents, the green line plots $\sigma(w(x))$, the vertical red line marks the Coal Mines Regulation Act of 1887, and the vertical magenta line indicates $\sigma(w(x)) = 0.5$.

Using the coal mining data we apply a number of well known univariate changepoint methods using their standard settings. We compared Pruned Exact Linear Time (PELT) (Killick et al., 2012) for changes in mean and variance and a nonparametric method named "ecp" (James and Matteson, 2013). Additionally, we tested the batch changepoint method described in Ross (2013) with Student-t and Bartlett tests for Gaussian data as well as Mann-Whitney and Kolmogorov-Smirnov tests for nonparametric changepoint estimation (Sharkey and Killick, 2014). Figure 3 compares the dates of change identified by these methods to the midpoint date where $\sigma(w(x)) = 0.5$ in GPCS.

Most of the methods identified a midpoint date between 1886 and 1895. While each method provides a point estimate of the change, only GPCS provides a clear, quantitative description of the development of this change. Indeed the 5.6 years during which the change surface transitions between $\sigma(w(x)) = 0.25$ to $\sigma(w(x)) = 0.75$ nicely encapsulate most of the point estimate method results.

4.3. New York City Lead Data

In recent years there has been heightened concern about lead-tainted water in major United States metropolitan areas. For example, concerns about lead poisoning in Flint, Michigan's water supply garnered national attention in 2015 and 2016, leading to Congressional hearings. Similar lead contamination issues have been reported in a spate of United States cities such as Cleveland, OH, New York, NY, and Newark, NJ (Editorial Board, 2016). Lead concerns in New York City have focused on lead-tainted water in schools and public housing projects, prompting reporting in some local and national media (Gay, 2016).

In order to understand the evolving dynamics of New York City residents' concerns about lead-tainted water, we analyzed requests for residential lead testing kits in New York City. These kits can be freely ordered by any resident of New York City and allow individuals to test their household's water for elevated levels of lead (City, 2016). We considered weekly requests for each zip code in New York City from January 2014 through April 2016. This provides a proxy for measuring the concern about lead tainted water. Figure 16 shows the aggregated requests over the entire city for lead testing kits during the observation period. It could be argued that this is an imperfect reflection of citizen concern since is unlikely that a household will request more than one testing kit within a relatively short period of time. Thus a reduction in requests may be due to saturation in demand for kits rather than a decrease in concern. However, we contend that since there were only 28,057 requests for lead testing kits over the entire observation period, and New York City contains approximately 3,148,067 households, there is a substantial pool of households in New York City that are able to signal their concern through requesting a lead testing kit (Census Bureau, 2014a).

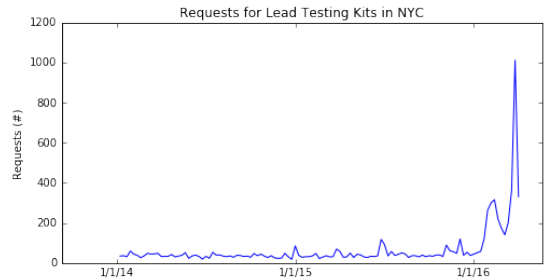


Figure 16: Requests for residential lead testing kits in New York City aggregated at a weekly level across the entire city.

While there is a distinct uptick in requests for kits towards the middle and end of the observation period, there is no ground truth change point, unlike the coal mining example in section 4.2 and the measles incidence example in section 4.4. We apply GPCS with two latent functions, spectral mixture kernels, and $w(x)$ defined by RKS features. Note that the inputs are three dimensional, $x \in \mathbb{R}^3$, with two spatial dimensions representing centroids of each zipcode and one temporal dimension.

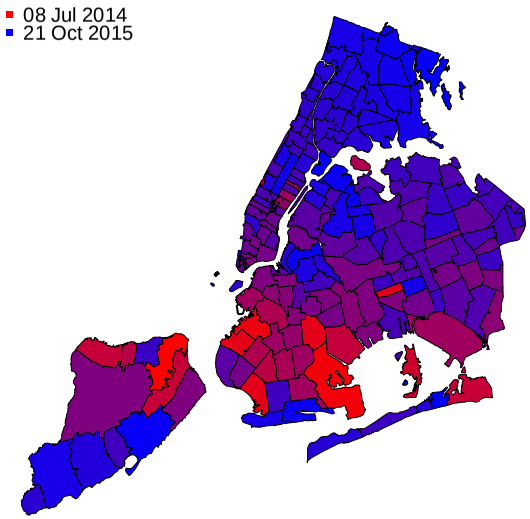


Figure 17: NYC zip codes colored by the date where $\sigma(w(x_{\text{zip}})) = 0.5$. Red indicates earlier dates, with Bulls Head in Staten Island being the earliest. Blue indicates later dates, with New Hyde Park at the eastern edge of Queens being the latest.

The model suggests that residents' concerns about lead tainted water had distinct spatial and temporal variation. In Figure 17 we depict the midpoint, $\sigma(w(x_{\text{zip}})) = 0.5$, for each zip code. We illustrate the spatial variation in the midpoint date by shading zip codes with an early midpoint in red and zip codes with later midpoint in blue. Regions in Staten Island and Brooklyn experienced the earliest midpoints, with Bulls Head in Staten Island (zip code 10314) being the first area to reach $\sigma(w(x_{\text{zip}})) = 0.5$ and New Hyde Park at the eastern edge of Queens (zip code 11040) being the last. The model detects certain zip codes changing in mid to late 2014, which somewhat predates the national publicity surrounding the Flint water crisis. However, most zip codes have midpoint dates sometime in 2015.

In Figure 18 we depict the change surface slope from $\sigma(w(x_{\text{zip}})) = 0.25$ to $\sigma(w(x_{\text{zip}})) = 0.75$ for each zip code to estimate the rate of change. We illustrate the variation in slope by shading zip codes with flatter change slopes in red and the steeper change slopes in blue. The flattest change surface occurred in Mariner's Harbor in Staten Island (zip code 10303) while the steepest change surface occurred in Woodlawn Heights in the Bronx (zip code 10470). We find that some zip codes had approximately four times the rate of change as others.

Regression analysis: The variations in the change surface indicate that the concerns about lead-tainted water may have varied heterogeneously over space and time. In order

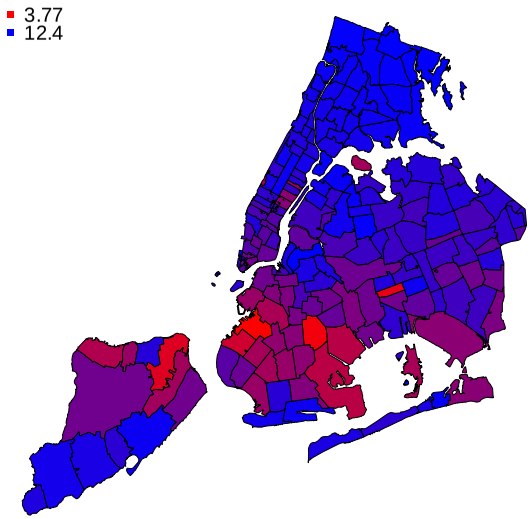


Figure 18: NYC zip codes colored by the slope of $\sigma(w(x_{\text{zip}}))$ from 0.25 to 0.75. Red indicates flatter slopes, with Mariner's Harbor in Staten Island being the flattest. Blue indicates steeper slopes, with Woodlawn Heights in the Bronx being the steepest.

to better understand these patterns we considered demographic and housing characteristics that may have contributed to differential concern among residents in New York City. Specifically we examined potential factors influencing the midpoint date between the two regimes. All data were taken from the 2014 American Community Survey 5 year average at the zip code level (Census Bureau, 2014b). Factors considered included information about residents such as education of householder, whether the householder was the home owner, previous year's annual income of household, number of people per household, and whether a minor or senior lived in the household. Additionally, we considered information about when the homes were built.

Results of a linear regression over all factors can be seen in Table 4. Five variables were statistically significant at a p-value < 0.05: median annual household income, percentage of houses built 1940-1959, percentage of householders with high school equivalent education, percentage of householders with at least a college education, and percentage of owner occupied households. Median annual household income had a positive correlation with the change date, suggesting that higher household income is associated with later midpoint dates. People with lower incomes may tend to live in housing that is less well maintained, or is perceived to be less well maintained. Thus they may require less "activation energy" to request lead testing kits when faced with possible environmental hazards. Education

Table 4: Results from a linear regression to the NYC lead midpoint date, $\sigma(w(x_{\text{zip}})) = 0.5$. Variables are listed on the left while their coefficients, with standard errors in parentheses, are listed on the right. Asterisks indicate statistically significant variables.

	Dependent variable:
	Midpoint date
Log median household income	21.916**
	(7.912)
% homes built after 2010	0.549
	(0.724)
% homes built 2000-2009	0.061
	(0.164)
% homes built 1980-1999	−0.070
	(0.153)
% homes built 1960-1979	0.027
	(0.094)
% homes built 1940-1959	0.667**
	(0.092)
% education high school equivalent	−1.609**
	(0.331)
% education some college	0.143
	(0.312)
% education college and above	−0.864**
	(0.303)
% households owner occupied	−0.310*
	(0.126)
Average family size	9.507
	(6.453)
% households with member 18 or younger	−0.020
	(0.282)
% households with member 60 or older	0.202
	(0.215)
% households with only one member	0.283
	(0.227)
Constant	−149.602
	(77.036)
Observations	176
R^2	0.420
Adjusted R^2	0.370
Note:	*p<0.05; **p<0.01

levels were compared to a base value of householders with less than a high school education. Thus zip codes with more educated householders tended to have earlier midpoint dates, and more concern about lead-tainted water. Similarly, owner occupied households had a negative correlation with the midpoint date. Since owner occupiers may tend to have more knowledge about their home infrastructure and may expect to remain in a location for longer than renters – perhaps even over generations – they could have a greater interest in ensuring low levels of water-based lead. The positive correlation of homes built between 1940-1959 may be due to a geographic anomaly since zip codes with the highest proportion of these homes are all in Eastern Queens. This region has very high median incomes which may ultimately explain the later midpoint dates.

This analysis indicates that more education and outreach to lower-income families by the New York City Department of Environmental Protection could be an effective means of addressing residents' concerns about future health risks. Additionally, it suggests an information disparity between renters and owner-occupiers that may be of interest to policy makers. Beyond the statistical analysis of demographic data, we also qualitatively examined media coverage related to the Flint water crisis as detailed by the Flint Water Study (Water Study, 2015). While some articles and news reports were reported in 2014, the vast majority began in 2015. The increased rate and national scope of this coverage in 2015 and 2016 may explain why zip codes with later midpoint dates shifted more rapidly. Additionally, it may be that residents with lower incomes identified earlier with those in Flint and thus were more concerned about potentially contaminated water than their more affluent neighbors.

In addition to the regression factors, there is a significant positive correlation between change slope and midpoint date with a p-value of 4×10^{-4}. The positive correlation between midpoint date and change slope is evident from a visual inspection of Figures 17 and 18. This relation indicates that in zip codes that changed later, their changes were relatively quicker perhaps due to the prevalence of news coverage at that later time.

4.4. United States Measles Data

Measles was nearly eradicated in the United States following the introduction of the measles vaccine in 1963. However, due to the vast geographic, ethnic, bureaucratic, and socioeconomic heterogeneity in the United States we may expect differential effectiveness of the vaccination program, particularly in its initial years. We analyze monthly incidence data for measles from 1935 to 2003 in each of the continental United States and the District of Columbia. Incidence rates per 100,000 population based on historical population estimates are made publicly available by Project Tycho (van Panhuis et al., 2013). We fit the model to $\approx 33,000$ data points where $x \in \mathbb{R}^3$ with two spatial dimensions representing centroids of each state and one temporal dimension.

We apply GPCS with two latent functions, spectral mixture kernels, and $w(x)$ defined by RKS features. We do not provide prior information about the 1963 vaccination date. Results for three states are shown in Figure 19 along with the predicted change surface for each state. The red line marks the vaccine year of 1963, while the magenta line marks where $\sigma(w(x_{\text{state}})) = 0.5$.

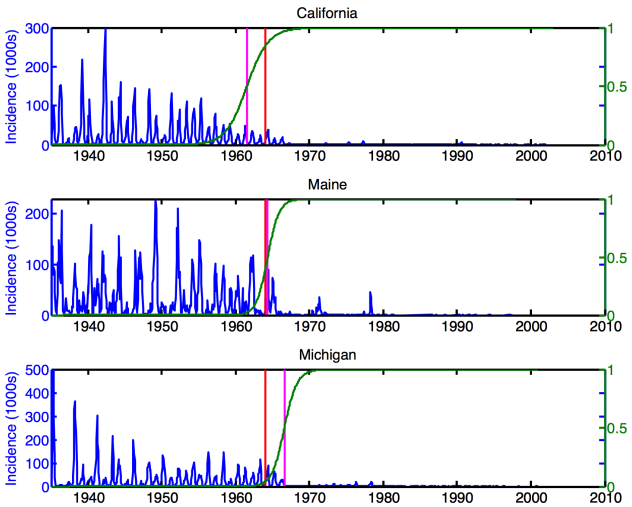


Figure 19: Measles incidence levels from three states, 1935 to 2003. The green line plots $\sigma(w(x_{\text{state}}))$, the vertical red line indicates the vaccine in 1963, and the magenta line indicates $\sigma(w(x_{\text{state}})) = 0.5$.

GPCS correctly identified the time frame when the measles vaccine was released in the United States. Additionally, the model suggests that the effect of the measles vaccine varied both temporally and spatially. This finding again demonstrates the effectiveness of GPCS to detect changes in real world data while providing important insight into the change's dynamics that are not ascertainable through existing models. In Figure 20 we depict the midpoint, $\sigma(w(x_{\text{state}})) = 0.5$, for each state. We illustrate the spatial variation in the change surface midpoint by shading states with an early midpoint in red and states with a later midpoint in blue. We discover that there is an approximately 6 year range of midpoints between states, with California being the earliest and North Dakota being the latest.

In Figure 21 we depict the change surface slope from $\sigma(w(x_{\text{state}})) = 0.25$ to $\sigma(w(x_{\text{state}})) = 0.75$ for each state to estimate the rate of change. We illustrate the variation in slope by shading states with the flatter change regions in red and the steeper change regions in blue. Here we find that some states had approximately twice the rate of change as others, with Arizona having the flattest slope and Maine the steepest.

Regression analysis: These variations in the change surface indicate that the measles vaccine may have affected states heterogeneously over space and time. In order to better understand these dynamics we considered demographic information that may have contributed to differences in measles vaccine program implementation and effectiveness. Specifically we examined potential factors influencing the midpoint shift date between the two regimes, $\sigma(w(x_{\text{state}})) = 0.5$. Since the change surface shifts primarily during the 1960s and the

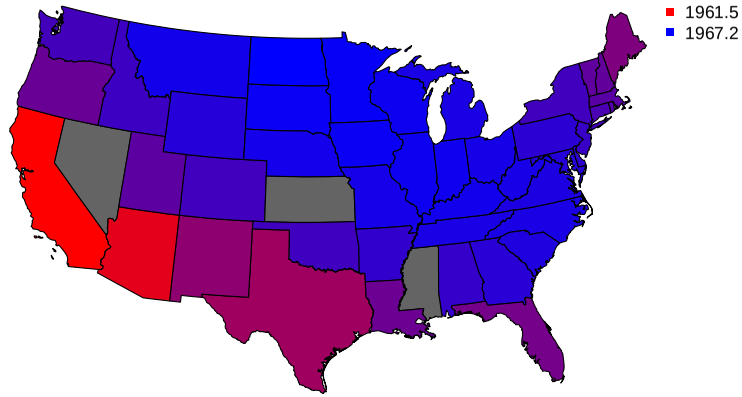


Figure 20: U.S. states colored by the date where $\sigma(w(x_{\text{state}})) = 0.5$. Red indicates earlier dates, with California being the earliest. Blue indicates later dates, with North Dakota being the latest. Grayed out states were missing in the dataset.

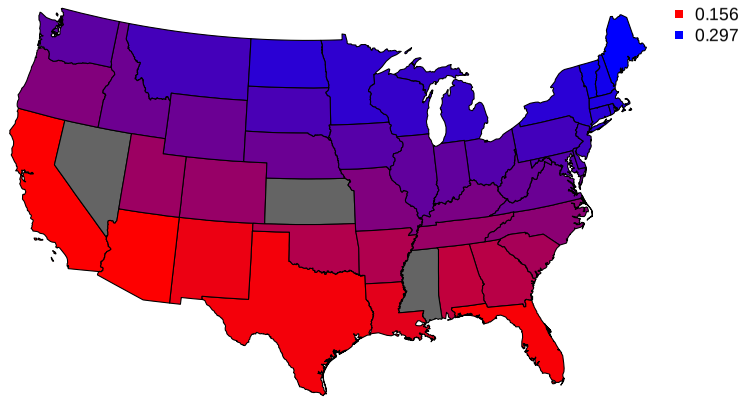


Figure 21: U.S. states colored by the slope of $\sigma(w(x_{\text{state}}))$ from 0.25 to 0.75. Red indicates flatter slopes, with Arizona being the lowest. Blue indicates steeper slopes, with Maine being the highest. Grayed out states were missing in the dataset.

measles vaccine is introduced in 1963, we consider historical census data only from 1960-1962 (Census Bureau, 1999). Factors included annual birth rate, death rates of different age segments, and population in each state. Since measles is often contracted by children and

people are rarely diagnosed for the disease twice in their life (it is a permanently immunizing disease), previous literature has shown that birth rates and the size of a young non-immune population is important for understanding the pre-vaccination dynamics of measles (Earn et al., 2000). Indeed, before the measles vaccine 5-9 year olds comprised 50% of disease incidence (Control and Prevention, 2016). We also consider median household income and household income inequality for each state. Finally, we also consider the average annual temperature in each state.

The results of a linear regression over all factors can be seen in Table 5. Four variables were statistically significant at a p-value < 0.05: the Gini coefficient of annual family income per state, average annual temperature, death rate of people aged 10+, and proportion of population aged 0-9. The Gini coefficient had a relatively large, positive correlation suggesting that wider family income inequality is associated with later dates of switching to the post-vaccine regime. One potential explanation of this phenomenon may be that states with higher Gini coefficients may have had large socio-economically depressed communities as well as substantial rural populations. Inoculation and vaccination education may have been more difficult in those communities and regions, thus delaying the midpoint date in those states. For example, Arkansas, Alabama, Kentucky, and Tennessee are all relatively rural states and have among the highest Gini coefficients. These states all have relatively late midpoint dates sometime in 1966. Another interesting example is the District of Columbia, which had the highest Gini coefficient. Although Washington D.C. is an urban center, it had also been an area of poverty and substandard local government, which may have contributed to its late change. Warmer temperatures are correlated with early midpoint dates perhaps due to biological mechanisms underlying the contagion of measles. Additionally, measles is spread through human contact which may also be affected by weather patterns. Death rates of people aged 10+ and relatively larger populations of children aged 0-9 were associated with later midpoint dates. Both of these factors indicate higher density of young children who may never have been affected by measles. This in turn may have increased the prevalence of the virus and delayed the midpoint date.

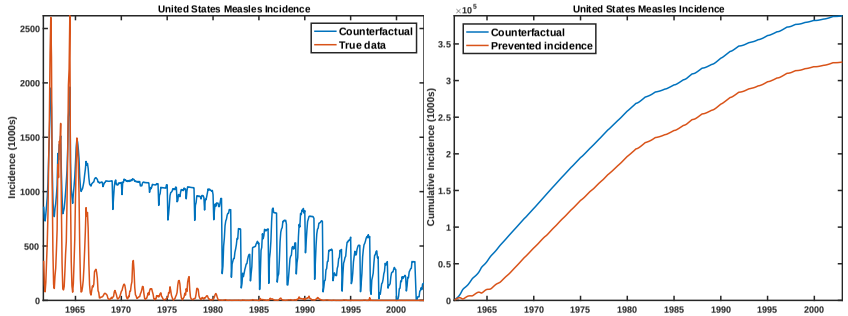
In addition to the regression factors, there is a significant positive correlation between change slope and midpoint date with a p-value $< 2.2 \times 10^{-16}$, suggesting that states with later changes transition more quickly from the pre-vaccine regime to the post-vaccine regime. The steeper change slope may be due to other states already having inoculated their residents. Fewer measles cases nationwide could have enabled states with later midpoint dates to more effectively contain the disease in their borders.

While this analysis does not provide conclusive results about underlying causal mechanisms, it suggests that further scientific research is warranted to understand the political and demographic factors that contributed to differential effectiveness in the early years of the measles vaccine program. Indeed, one challenge in analyzing measles at a state-level aggregation is that measles disease dynamics may vary between cities even within states (Dalziel et al., 2016). Nevertheless, the results indicate that future vaccination programs should particularly consider how to quickly and effectively provide vaccinations to rural areas and provide additional resources to socioeconomically disadvantaged communities. Additionally, care should be taken when accounting for the effects of weather patterns and population dynamics.

Table 5: Results from a linear regression to the measles incidence midpoint date, $\sigma(w(x_{\text{state}})) = 0.5$. Variables are listed on the left while their coefficients, with standard errors in parentheses, are listed on the right. Asterisks indicate statistically significant variables.

	Dependent variable:
	Midpoint date
Log death rate aged 0-4	-1.614
	(2.186)
Log death rate aged 5-9	5.023
	(2.640)
Log death rate aged 10+	7.651^{**}
	(2.632)
Log birth rate	-10.932
	(5.472)
Gini of family income	48.503^{**}
	(17.461)
Log median household income	4.997
	(2.620)
Log population	0.117
	(0.228)
Proportion of population aged 0-9	84.757^{*}
	(32.784)
Average temperature (°F)	-0.093^{*}
	(0.035)
Constant	$1,980.509^{**}$
	(24.237)
Observations	46
R^2	0.396
Adjusted R^2	0.245
Note:	*p<0.05; **p<0.01

Counterfactual analysis: Using the counterfactual GPCS framework, we inferred the incidence of measles in the absence of the change surface identified by GPCS. We used the latent function that is dominant in the data before the measles vaccine to compute posterior estimates for measles incidence between the earliest detected midpoint date in 1961 and the end of the data in 2003. This estimation is inspired by the counterfactual estimation described in van Panhuis et al. (2013). We argue that GPCS provides more believable counterfactual estimates than simple interpolations or regressions because GPCS is a more expressive model for measles dynamics and explicitly considers data variation both before and after the start of the measles vaccine program. Figure 22 depicts the aggregated counterfactual posterior mean estimates over the entire United States. The left plot shows true and counterfactual monthly incidence, while the right plot depicts the cumulative counterfactual incidence. Under the assumption that the change surface reflects the causal effect of the vaccine program intervention, we also estimate how many cases were "prevented" through the vaccination program. Since disease dynamics may have many causal factors, we cannot disentangle the introduction of the measles vaccine from any contemporaneous societal or policy changes that may have impacted measles incidence. Thus these findings are a starting point for more extensive epidemiological research. Additionally, while we plot the posterior mean estimates, note that our confidence in these estimates diminishes as we consider counterfactual estimates far from the change region.



(a) (b)

Figure 22: Counterfactual posterior mean estimates for measles incidence. Plot (a) depicts the aggregated counterfactual posterior mean estimates over the entire United States. Plot (b) depicts the cumulative counterfactual incidence over the entire United States as well as estimating how many cases were "prevented" through the vaccination program under the assumption that the change surface corresponds to the vaccine intervention.

5. Conclusion

We presented change surfaces as an expressive generalization of changepoints that are able to model complex, multidimensional data with varying rates of change between latent functional regimes. Additionally, we showed how change surfaces can be used for counterfactual prediction. Yet we believe that change surfaces are not only a generalization of the statistical properties of change points, but truly a conceptual shift for modeling of distributional changes in data. Instead of attempting to discover discrete moments of change, change surfaces offer a more realistic framework for modeling complex data. Indeed, the change surface analyses presented in this paper demonstrate that they can provide scientific and public policy insights.

As an instantiation of change surfaces, we presented GPCS, which uses independent Gaussian process priors and flexible RKS basis functions to enable a highly expressive change surface model. We derived counterfactual prediction methods for GPCS that naturally provide counterfactual posterior mean and variance estimates. We also demonstrate that probabilistic inference within GPCS automatically discourages extraneous complexity, naturally leading to interpretable generative hypotheses for our observations. To support GPCS we also created a novel scalable inference method for multiple additive kernels using the Weyl bound. This result extends far beyond change surfaces, enabling scalable Gaussian processes with non-separable covariance structures over multiple dimensions. Additionally, we developed an effective approach for initializing expressive spectral mixture kernels. Future work may consider combining the Weyl bound approach with recent developments in automated computation of Gaussian process log determinants (Gardner et al., 2018; Wang et al., 2019). In particular, integrating the Weyl bound methodology with new MVM approaches in Dong et al. (2017) may provide important computational benefits.

Using change surfaces we are able to model complex, spatio-temporal data with expressivity and clarity. We studied requests for lead testing kits in New York City between 2014-2016, a period of heightened concern regarding water quality around the United States. GPCS identified a change in the dynamics of requests, but perhaps more importantly, it illustrated how that change developed over time and varied over space. The spatio-temporal heterogeneity modeled by GPCS enabled further investigation into demographic factors that may have influenced the behavior of residents in various parts of the city. This analysis is only possible with a change surface model because standard changepoint approaches are only able to provide single, point-in-time estimates of a midpoint date. Policy makers are often interested in learning how public health risks or legal regulations affect various populations. Our results demonstrate that change surfaces can be a particularly effective method for policy makers to understand how changes develop and are distributed over a multidimensional domain.

We also used GPCS to model measles incidence in the United States over the course of the twentieth century. In addition to identifying a change in regimes around the introduction of the measles vaccine in 1963, we used the fitted change surface to illuminate heterogeneity across states. The differential change rates and midpoint dates in each state could have important scientific and policy implications for vaccination campaigns. To this end, we provide a regression analysis of institutional and demographic factors that may have influenced the impact of the measles vaccination program.

Finally, we are excited about how the introduction of change surfaces could inspire further research into expressive modeling of complex changes. As we emphasized in Section 3, our use of Gaussian processes in GPCS presents but one approach to modeling change surfaces. Future work may provide alternative methods for characterizing change surfaces

using statistical approaches beyond Gaussian processes. For example, other instantiations of change surfaces could utilize penalty terms to enforce the soft mutual exclusivity between the functional regimes, or else employ decision-tree like structures to divide the domain. Another fruitful methodological avenue could extend the retrospective analysis in this paper to address online or sequential change surface detection. Additionally, change surfaces may be further used for causal inference in conjuction with natural experiments, which are often used by econometricians for causal inference in observational data. For example, change surfaces may help discover regression discontinuity designs (Herlands et al., 2018) or identify heterogeneous treatment effects in real-valued data.

Acknowledgments

Thank you to Wilbert Van Panhuis and Seth Flaxman for providing much appreciated insight and suggestions. This material is based upon work supported by NSF Graduate Research Fellowship DGE-1252522 as well as NSF awards IIS-0953330 and IIS-1563887.

References

Alberto Abadie, Joshua Angrist, and Guido Imbens. Instrumental variables estimates of the effect of subsidized training on the quantiles of trainee earnings. *Econometrica*, 70 (1):91–117, 2002.

Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *stat*, 1050:19, 2007.

Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

Susan Athey and Guido W Imbens. Identification and inference in nonlinear difference-in-differences models. *Econometrica*, 74(2):431–497, 2006.

Alexander Aue and Lajos Horváth. Structural breaks in time series. *Journal of Time Series Analysis*, 34(1):1–16, 2013.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L. Scott. Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, 9:247–274, 2015.

E Brodsky and Boris S Darkhovsky. *Nonparametric methods in change point problems*, volume 243. Springer Science & Business Media, 2013.

Bradley P Carlin, Alan E Gelfand, and Adrian FM Smith. Hierarchical bayesian analysis of changepoint problems. *Applied statistics*, pages 389–405, 1992.

U. S. Census Bureau. United states historical census data. `https://www.census.gov/hhes/www/income/data/historical/state/`, 1999. Accessed: 2016-4-10.

U. S. Census Bureau. American community survey 1-year estimates. `http://factfinder.census.gov/`, 2014a. Accessed: 2016-4-10.

U. S. Census Bureau. American community survey 5-year estimates. `http://factfinder.census.gov/`, 2014b. Accessed: 2016-4-10.

Jie Chen and Arjun K Gupta. *Parametric statistical change point analysis: With applications to genetics, medicine, and finance*. Springer Science & Business Media, 2011.

Herman Chernoff and Shelemyahu Zacks. Estimating the current mean of a normal distribution which is subjected to changes in time. *The Annals of Mathematical Statistics*, pages 999–1018, 1964.

New York City. Water lead test kit request. `http://www1.nyc.gov/nyc-resources/service/1266/water-lead-test-kit-request`, 2016. Accessed: 2016-4-10.

Centers For Disease Control and Prevention. Epidemiology and prevention of vaccine-preventable diseases, 2016. URL `https://www.cdc.gov/vaccines/pubs/pinkbook/meas.html`.

Benjamin D Dalziel, Ottar N Bjørnstad, Willem G van Panhuis, Donald S Burke, C Jessica E Metcalf, and Bryan T Grenfell. Persistent chaos of measles epidemics in the prevaccination united states caused by a small change in seasonal transmission patterns. *PLoS Comput Biol*, 12(2):e1004655, 2016.

Kun Dong, David Eriksson, Hannes Nickisch, David Bindel, and Andrew G Wilson. Scalable log determinants for gaussian process kernel learning. In *Advances in Neural Information Processing Systems*, pages 6327–6337, 2017.

David J. D. Earn, Pejman Rohani, Benjamin M. Bolker, and Bryan T. Grenfell. A simple model for complex dynamical transitions in epidemics. *Science*, 287(5453):667–670, 2000. ISSN 0036-8075. doi: 10.1126/science.287.5453.667. URL `http://science.sciencemag.org/content/287/5453/667`.

The Editorial Board. Poisoned water in newark schools. *New York Times*, March 2016.

Miroslav Fiedler. Bounds for the determinant of the sum of hermitian matrices. *Proceedings of the American Mathematical Society*, pages 27–31, 1971.

Seth R Flaxman, Andrew Gordon Wilson, Daniel B Neill, Hannes Nickisch, and Alexander J Smola. Fast kronecker inference in gaussian processes with non-gaussian likelihoods. *International Conference on Machine Learning 2015*, 2015.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

Roman Garnett, Michael A Osborne, and Stephen J Roberts. Sequential bayesian prediction in the presence of changepoints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 345–352. ACM, 2009.

Mara Gay. Elevated levels of lead found in water of some vacant public-housing apartments. *Wall Street Journal*, 2016.

Zhong Guan. A semiparametric changepoint model. *Biometrika*, pages 849–862, 2004.

Joseph Guinness, Michael L Stein, et al. Interpolation of nonstationary high frequency spatial–temporal temperature data. *The Annals of Applied Statistics*, 7(3):1684–1708, 2013.

Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. Counterfactual prediction with deep instrumental variables networks. *arXiv preprint arXiv:1612.09596*, 2016.

William Herlands, Andrew Wilson, Hannes Nickisch, Seth Flaxman, Daniel Neill, Wilbert Van Panhuis, and Eric Xing. Scalable gaussian processes for characterizing multidimensional change surfaces. In *Artificial Intelligence and Statistics*, pages 1013–1021, 2016.

William Herlands, Edward McFowland III, Andrew Gordon Wilson, and Daniel B Neill. Automated local regression discontinuity design discovery. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1512–1520. ACM, 2018.

Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

Lajos Horváth and Gregory Rice. Extensions of some classical methods in change point analysis. *Test*, 23(2):219–255, 2014.

B Gail Ivanoff and Ely Merzbach. Optimal detection of a change-set in a spatial poisson process. *The Annals of Applied Probability*, pages 640–659, 2010.

Nicholas A James and David S Matteson. ecp: An r package for nonparametric multiple change point analysis of multivariate data. *arXiv preprint arXiv:1309.3295*, 2013.

RG Jarrett. A note on the intervals between coal-mining disasters. *Biometrika*, pages 191–193, 1979.

Fredrik D Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. *arXiv preprint arXiv:1605.03661*, 2016.

P. K. Kapur, H. Pham, A. Gupta, and P. C. Jha. *Change-Point Models*, pages 171–213. Springer London, London, 2011. ISBN 978-0-85729-204-9. doi: 10.1007/978-0-85729-204-9_5. URL `http://dx.doi.org/10.1007/978-0-85729-204-9_5`.

Hossein Keshavarz, Clayton Scott, and XuanLong Nguyen. Optimal change point detection in gaussian processes. *Journal of Statistical Planning and Inference*, 193:151–178, 2018.

Rebecca Killick, Paul Fearnhead, and IA Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500): 1590–1598, 2012.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.

James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

David JC MacKay. Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Anandamayee Majumdar, Alan E Gelfand, and Sudipto Banerjee. Spatio-temporal changepoint modeling. *Journal of Statistical Planning and Inference*, 130(1):149–166, 2005.

RJ Martin. The use of time-series models and methods in the analysis of agricultural field trials. *Communications in Statistics-Theory and Methods*, 19(1):55–81, 1990.

Ed McFowland, Sriram Somanchi, and Daniel B. Neill. Efficient discovery of heterogeneous treatment effects in randomized experiments via anomalous pattern detection. *Working paper*, 2016.

Scottish Mining. Coal mines regulation act, 2017. URL `http://www.scottishmining.co.uk/256.html`.

Thomas P Minka. Automatic choice of dimensionality for pca. In *Advances in neural information processing systems*, pages 598–604, 2001.

Jesper Møller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. Log gaussian cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

Geoff K Nicholls and Patrick D Nunn. On building and fitting a spatio-temporal change-point model for settlement and growth at bourewa, fiji islands. *arXiv preprint arXiv:1006.5575*, 2010.

ES Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

AE Raftery and VE Akman. Bayesian analysis of a poisson process with a change-point. *Biometrika*, pages 85–89, 1986.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

Carl Rasmussen and Chris Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. In *Advances in neural information processing systems*, pages 294–300, 2001.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010.

Steven Reece, Roman Garnett, Michael Osborne, and Stephen Roberts. Anomaly detection and removal using non-stationary gaussian processes. *arXiv preprint arXiv:1507.00566*, 2015.

Gordon J Ross. Parametric and nonparametric sequential change detection in r: The cpm package. *Journal of Statistical Software*, page 78, 2013.

Donald B Rubin. Comment: Which ifs have causal answers. *Journal of the American Statistical Association*, 81(396):961–962, 1986.

Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

Yunus Saatçi. *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge, 2011.

Yunus Saatçi, Ryan D Turner, and Carl E Rasmussen. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 927–934, 2010.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. In *Advances in Neural Information Processing Systems*, pages 1697–1708, 2017.

Paul Sharkey and Rebecca Killick. Nonparametric methods for online changepoint detection. Technical Report STOR601, Lancaster University, 2014.

Shinichiro Shirota and Alan E Gelfand. Inference for log gaussian cox processes using an approximate marginal posterior. *arXiv preprint arXiv:1611.10359*, 2016.

Alexander G Tartakovsky, Aleksey S Polunchenko, and Grigory Sokolov. Efficient computer network anomaly detection by changepoint detection methods. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):4–11, 2013.

Willem G van Panhuis, John Grefenstette, Su Yon Jung, Nian Shong Chok, Anne Cross, Heather Eng, Bruce Y Lee, Vladimir Zadorozhny, Shawn Brown, Derek Cummings, et al. Contagious diseases in the united states from 1888 to the present. *The New England journal of medicine*, 369(22):2152, 2013.

Natalya Verbitsky-Savitz and Stephen W Raudenbush. Causal inference under interference in spatial settings: a case study evaluating community policing program in chicago. *Epidemiological Methods*, 1:106–130, 2012.

Ke Alexander Wang, Geoff Pleiss, Jacob R Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. *arXiv preprint arXiv:1903.08114*, 2019.

Flint Water Study. Flint water study: Articles in the press. `http://flintwaterstudy.org/articles-in-the-press/`, 2015. Accessed: 2016-4-10.

Hermann Weyl. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1067–1075, 2013.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1775–1784, 2015.

Andrew Wilson, Zoubin Ghahramani, and David A Knowles. Gaussian process regression networks. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 599–606, 2012.

Andrew Wilson, Elad Gilboa, John P Cunningham, and Arye Nehorai. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*, pages 3626–3634, 2014.

Andrew G Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016a.

Andrew Gordon Wilson. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, PhD thesis, University of Cambridge, 2014.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016b.

Journal of Machine Learning Research 20 (2019) 1-31 Submitted 8/17; Revised 5/19; Published 7/19

Hamiltonian Monte Carlo with Energy Conserving Subsampling

Khue-Dung Dang DOANKHUEDUNG.DANG@UNSW.EDU.AU
UNSW Business School, School of Economics, University of New South Wales.
ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS)

Matias Quiroz QUIROZ.MATIAS@GMAIL.COM
School of Mathematical and Physical Sciences, University of Technology Sydney.
ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS)

Robert Kohn R.KOHN@UNSW.EDU.AU
UNSW Business School, School of Economics, University of New South Wales.
ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS)

Minh-Ngoc Tran MINH-NGOC.TRAN@SYDNEY.EDU.AU
Discipline of Business Analytics, University of Sydney.
ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS)

Mattias Villani MATTIAS.VILLANI@LIU.SE
Division of Statistics and Machine Learning, Linköping University.
Department of Statistics, Stockholm University.
ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS)

Editor: Ryan Adams

Abstract

Hamiltonian Monte Carlo (HMC) samples efficiently from high-dimensional posterior distributions with proposed parameter draws obtained by iterating on a discretized version of the Hamiltonian dynamics. The iterations make HMC computationally costly, especially in problems with large data sets, since it is necessary to compute posterior densities and their derivatives with respect to the parameters. Naively computing the Hamiltonian dynamics on a subset of the data causes HMC to lose its key ability to generate distant parameter proposals with high acceptance probability. The key insight in our article is that efficient subsampling HMC for the parameters is possible if both the dynamics and the acceptance probability are computed from the same data subsample in each complete HMC iteration. We show that this is possible to do in a principled way in a HMC-within-Gibbs framework where the subsample is updated using a pseudo marginal MH step and the parameters are then updated using an HMC step, based on the current subsample. We show that our subsampling methods are fast and compare favorably to two popular sampling algorithms that use gradient estimates from data subsampling. We also explore the current limitations of subsampling HMC algorithms by varying the quality of the variance reducing control variates used in the estimators of the posterior density and its gradients.

Keywords: Bayesian inference, Big Data, Markov chain Monte Carlo, Estimated likelihood, Stochastic gradient Hamiltonian Monte Carlo, Stochastic Gradient Langevin Dynamics

©2019 Khue-Dung Dang, Matias Quiroz, Robert Kohn, Minh-Ngoc Tran, Mattias Villani.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-452.html.

1. Introduction

Bayesian inference relies on computing expectations with respect to the posterior density of the model parameters given the data. The functional form of the posterior density often does not correspond to a known density and hence obtaining independent samples to compute the expectation by Monte Carlo integration is difficult, especially when the dimension of the model parameter is moderate to large. Markov Chain Monte Carlo (MCMC) is a generic sampling algorithm that produces correlated draws from the posterior density.

Metropolis-Hastings (MH) (Metropolis et al., 1953; Hastings, 1970) is arguably the most popular MCMC algorithm. Its most common implementation uses a random walk proposal, in which a new sample is proposed based on the current state of the Markov chain. While Random walk MH is easy to implement, it explores the posterior very slowly in high-dimensional problems and gives highly correlated draws and imprecise estimators of posterior integrals.

Hamiltonian Monte Carlo (HMC) (Duane et al., 1987) can produce distant proposals while maintaining a high acceptance probability (Neal, 2011; Betancourt, 2017). HMC augments the target posterior by adding fictitious momentum variables and carries out the sampling on an extended target density. The extended target is proportional to the exponential of a Hamiltonian function that describes the total energy of the system, which is the sum of the potential energy (negative log posterior) and the kinetic energy (negative log density of the momentum variables). The Hamiltonian dynamics describes how the total energy evolves through time. One particularly interesting feature of the Hamiltonian is that it conserves energy as time evolves, a property that is approximately maintained even when the dynamics is approximated in discrete time. Hence, a MH proposal obtained by simulating the dynamics has approximately the same value of the extended target density as that of the current draw, resulting in a high acceptance probability, even when the proposed draw is far from the current draw. This typically avoids the inherently slow exploration of the parameter space evident in random walk proposals (Betancourt, 2017). HMC simulates the evolution of Hamiltonian dynamics for a given period of time. Up to a point, the longer the integration time the more effectively the dynamics explore the posterior distribution, while small integration times approach diffusive Langevin methods (Roberts and Rosenthal, 1998; Roberts and Stramer, 2002). In practice, the simulation of the dynamics is implemented with a numerical integrator. The more accurate the integrator, the larger the step size and the fewer total steps needed to simulate the dynamic for a chosen time. Smaller step sizes typically require larger integration times to achieve the same efficient exploration of the posterior distribution. In either case, HMC requires computing the gradient of the log-posterior in each step when simulating the dynamics, and in practice a large number of steps may be performed. The extra computations are often worthwhile as HMC greatly improves the sampling efficiency of the generated samples compared to a proposal which does not use gradient information. Even with the improved performance of HMC, however, the cost can be too prohibitive for the limited computational resources available in a given application. Consequently, some algorithms use subsampling of the data to reduce the cost of computing the posterior density and its gradients. Unfortunately, naive subsampling methods are often problematic and in particular they compromise many of the

features that give HMC its scalability (Betancourt, 2015). In this article we introduce a subsampling HMC scheme that can tackle very large data sets and maintain the scalability.

Our article speeds up computation by using subsets of the data to compute both the dynamics and the subsequent MH correction performed when deciding to accept a proposal. More precisely, we propose a HMC-within-Gibbs algorithm that alternates i) sampling small subsets of data using a pseudo-marginal step and ii) sampling parameters using the HMC dynamics and the MH correction based on the current subset of the data. We will here focus on HMC algorithms where the Hamiltonian dynamics are used to generate a proposal which is subsequently accepted or rejected using a MH step, which we refer to as HMC. Extensions to other HMC algorithms that use the entire trajectory generated by the dynamics (Hoffman and Gelman, 2014; Betancourt, 2017) are interesting future research direction discussed in Section 7.

We propose two different subsampling versions of the HMC algorithm. In the first *perturbed* approach we use a slightly biased likelihood estimator and show that the algorithm targets a perturbed posterior which gets close to the true posterior density as the subsample size and the number of observations of the data set become large; see Section 4.2 for details. The second approach uses an unbiased but possibly negative likelihood estimator which allows us to obtain simulation consistent estimators of the posterior expectation of any function of the parameters. However, this approach is harder to implement efficiently than the perturbed approach and is typically slower computationally.

We compare our algorithms to Stochastic Gradient Langevin Dynamics (Welling and Teh, 2011, SGLD) and Stochastic Gradient Hamiltonian Monte Carlo (Chen et al., 2014, SG-HMC), two of the most popular subsampling algorithms that use gradient information in machine learning. To make the comparison more challenging we implement both methods with control variates for improved performance (Baker et al., 2017). We demonstrate that our algorithms compare favorably to SGLD and SG-HMC. It is by now well known that all proposed subsampling MCMC and HMC algorithms need accurate control variates to lower the variance of the estimated posterior and gradients, and we explore the robustness of the algorithms when the control variates are degraded.

The paper is organized as follows. Section 2 reviews previous research and the methods we compare against. Section 3 presents our methodology using a general likelihood estimator and argues that it circumvents the incompatibility of data subsampling and HMC raised in recent literature. Sections 4 and 5 present two algorithms based on two specific likelihood estimators. Finally, Section 6 demonstrates the usefulness of the methods on two large data sets and compares with alternative approaches. This section also explores the limitation of subsampling approaches by experimenting with successively degrading the quality of the control variates used for variance reduction. Section 7 concludes and discusses future research.

2. Related work

2.1. High-dimensional MCMC for large data sets

There has recently been a surge of interest in developing posterior simulation algorithms that scale with respect to both the number of observations n and the number of parameters d. Since simulation methods have the ambitious goal of exploring all regions in parameter space with sizable probability mass, they naturally require many more iterations than posterior optimization algorithms. Posterior optimization is computationally attractive for big data, but does not quantify the posterior uncertainty, which is often a central task in science. Although there exist optimization-based methods that aim to approximate the entire posterior distribution, e.g. variational Bayes (Blei et al., 2017), Laplace approximations (Bernardo and Smith, 2001, Chapter 5) or integrated nested Laplace approximations (Rue et al., 2009), in practice it is nearly impossible to know how they perform without comparing the results to a posterior simulation method. It is thus important to develop posterior simulation methods that:

i) remain computationally efficient when n is large and

ii) explore the posterior distribution efficiently when d is large.

Two distinct approaches exist to resolve i). The first is to use parallel computing by dividing the n data observations into K parts, performing independent posterior simulation on each of the K subposteriors and subsequently merge the draws to represent the full data posterior. See, for example, Scott et al. (2013); Neiswanger et al. (2014); Wang and Dunson (2014); Minsker et al. (2014); Nemeth and Sherlock (2018). The second approach, which is the focus of our article, is to work with subsamples of m observations to estimate the full data posterior (Maclaurin and Adams, 2014; Korattikara et al., 2014; Bardenet et al., 2014, 2017; Maire et al., 2018; Bierkens et al., 2019; Quiroz et al., 2019, 2018b,a; Gunawan et al., 2018) or its gradient (Welling and Teh, 2011; Chen et al., 2014; Ma et al., 2015; Shang et al., 2015; Baker et al., 2017). The rest of this section reviews samplers which use gradient information about the posterior density.

The primary problem confronted in ii) is how to generate proposals which maintain a high acceptance probability and are also distant enough to avoid a highly persistent Markov chain for the model parameter. A useful approach is to simulate a discretized Langevin diffusion (Roberts and Rosenthal, 1998; Nemeth et al., 2016) or, more generally, Hamilton's equations (Duane et al., 1987) and use the simulated draw as a proposal in the MH algorithm to correct for the bias introduced by the discretization (Neal, 2011). HMC provides a solution to ii) (Neal, 2011; Betancourt, 2017), but when combined with i), the algorithm becomes computationally intractable since simulating the Hamiltonian dynamics requires a large number of costly gradient evaluations for every proposed parameter value.

2.2. Subsampling HMC algorithms and related approaches

A computationally attractive way to accelerate HMC is to use a fixed subsample of the data to unbiasedly estimate the computationally costly gradients in each step of the discretized Hamiltonian trajectory, and skip the MH correction to avoid evaluating the posterior on the full data. Betancourt (2015) demonstrates that this simple strategy produces highly biased trajectories, where the bias depends upon the quality of the gradient estimator. Moreover, Betancourt (2015) shows that attempts to average out the bias by renewing the subsample in each step of the trajectory still perform poorly; see also the *naive stochastic gradient HMC* in Chen et al. (2014). Betancourt (2015) illustrates that adding a MH correction step based on the full data to fully correct for the biased trajectories leads to a rapid deterioration of the acceptance probability of HMC as d increases, and concludes that there is a fundamental

incompatibility of HMC and data subsampling. As a remedy to the poor performance by the naive stochastic gradient HMC, Chen et al. (2014) propose adding a friction term to the dynamics to correct for the bias. For the rest of our article, we refer to the method using the friction term in the dynamics as Stochastic Gradient Hamiltonian Monte Carlo (SG-HMC). Chen et al. (2014) omit the MH correction step and the resulting bias in the posterior depends on the quality of the discretization of the continuous dynamics. Hence, in order to traverse the parameter space effectively, the integrator potentially needs a large number of steps to compensate for the small step size used when discretizing the dynamics, which may be very costly.

A different approach, but related in the sense that it uses an estimated gradient, is Stochastic Gradient Langevin Dynamics (Welling and Teh, 2011, SGLD). SGLD combines Stochastic Gradient Optimization (Robbins and Monro, 1951, SGO) and Langevin dynamics (Roberts and Rosenthal, 1998), by allowing the initial iterates to resemble a SGO and gradually traverse to Langevin Dynamics so as to not collapse to the posterior mode. SGLD avoids a costly MH correction by arguing that it is not needed because the discretization step size of the Langevin dynamics is decreased as a function of the iterates (Welling and Teh, 2011). However, this decreases the rate of convergence of estimators based on its output to $R^{-1/3}$, where R is the number of samples from the posterior (Teh et al., 2016), as opposed to $R^{-1/2}$ for MCMC (Roberts and Rosenthal, 2004), and in particular HMC. Practical implementations of SGLD use a sequence of step sizes that does not decrease to zero in the limit, and Vollmer et al. (2016) show that the posterior approximated by SGLD can then be quite far from the true posterior; see also Brosse et al. (2018). Bardenet et al. (2017) also demonstrate that SGLD can be accurate for the posterior mode, but gives a poor approximation of the full posterior distribution on a toy example with $d = 2$ and highly redundant data, i.e. superfluous amounts of data in relation to the complexity of the model. Recently, Dubey et al. (2016) improve SGLD using control variates, see also Baker et al. (2017) who, in addition, use control variates in the SG-HMC algorithm proposed in Chen et al. (2014). We implement both SGLD and SG-HMC with highly efficient control variates when comparing them to our method. We also implement the methods without control variates as it has been shown that sometimes variance reduction may be detrimental (Chatterji et al., 2018).

All the problems discussed above with subsampling in HMC and related algorithms stem from the fact that subsampling disconnects the Hamiltonian from its own dynamics. This disconnect causes HMC proposals to lose their energy conserving property and their attractive ability to sample efficiently in high dimensions. The next section presents a new energy conserving approach to subsampling in HMC that keeps the connection intact by estimating both the Hamiltonian and its corresponding dynamics from the same subsample. By updating the subsample in a separate pseudo-marginal step we make sure that the HMC algorithm still targets the posterior based on all data. Put differently, our new approach creates a Hamiltonian system with corresponding dynamics for a given subset of data. This allows for the scalability of HMC to be maintained for each subsample, as is demonstrated in Section 6.7.

3. Energy conserving subsampling in HMC

This section presents our new approach to subsampling in HMC and discusses why our approach avoids the pitfalls described in Betancourt (2015). In order to not distract from the main ideas, we first present our approach for a general unbiased and almost surely positive likelihood estimator from data subsampling. Sections 4 and 5 then present practical algorithms based on likelihood estimators proposed in Quiroz et al. (2019) and Quiroz et al. (2018b), respectively.

3.1. Pseudo-marginal MCMC

Denote the model parameter vector by $\theta \in \Theta \subset \mathbb{R}^d$, where $\mathbb{R}^d$ is the space of d-dimensional real vectors, and let $\pi(\theta)$ be its posterior density given a data set y with n observations. We first briefly describe pseudo-marginal MCMC (Andrieu and Roberts, 2009), which serves as inspiration for one building block in our subsampling HMC approach. Pseudo-marginal algorithms target the augmented posterior

$$\overline{\pi}_m(\theta, u) \propto \widehat{L}_m(\theta) p_\Theta(\theta) p_U(u), \tag{1}$$

where $\widehat{L}_m(\theta)$ is an unbiased and non-negative estimator of the likelihood $L(\theta)$ based on m auxiliary variables, u, with density $p_U(u)$. In the particular application to subsampling, $u \in \{1, \ldots, n\}^m$, contains the indices for the data observations used in estimating the likelihood and m denotes the subsample size, see Section 4.1 for details. Note that $\widehat{L}_m$ and any quantity included in its definition depend on n, but this is suppressed in the notation for conciseness. We can now design an MCMC chain to sample θ and u jointly from (1) and, since $\widehat{L}_m(\theta)$ is unbiased, the θ iterates are samples from $\pi(\theta)$.

The choice of m is crucial for pseudo-marginal methods. An m that is too small results in a noisy likelihood estimator and the Markov chain may get stuck due to severely overestimating the likelihood at the current draw, subsequently rejecting nearly all proposals. Conversely, taking m too large wastes useful computational resources. A natural aim is to choose an m that minimizes the Computational Time (CT) needed to generate the equivalent of a single independent draw from the posterior, with

$$\text{CT} \coloneqq \text{IF} \times \text{Total number of density and gradient evaluations}, \tag{2}$$

where the Inefficiency Factor (IF) is proportional to the asymptotic variance when estimating a posterior functional based on the MCMC output, and is interpreted as the number of samples needed to obtain the equivalent of a single independent sample. The second term is proportional to the computing time for a single draw, a measure that is independent of the implementation. Starting with Pitt et al. (2012), there is a large literature showing that CT is minimized by an m that targets a variance of the log of the likelihood estimator around 1 (Doucet et al., 2015; Sherlock et al., 2015; Tran et al., 2017; Deligiannidis et al., 2018; Schmon et al., 2018). Recent developments in pseudo-marginal methods induce dependence in the auxiliary variables u over the MCMC iterations such that the likelihood estimates over the iterations become dependent (Deligiannidis et al., 2018). This makes it possible to tolerate a substantially larger variance of the likelihood estimator, and hence smaller subsamples in our context. We follow Tran et al. (2017) and induce the dependence over

the iterations by partioning the u's into blocks and only update a subset of the blocks in each iteration. The optimal subsample size m is then obtained by targeting a certain value for the conditional variance of the likelihood estimator for a given induced correlation ρ (Tran et al., 2017).

3.2. An energy conserving HMC-within-Gibbs framework

Following the standard HMC algorithm, our subsampling HMC algorithm introduces a fictitious continuous momentum vector $\vec{p} \in \mathcal{P} \subset \mathbb{R}^d$ of the same dimension as the continuous parameter vector θ. The extended target in (1) is then further augmented by $\vec{p}$ to

$$\overline{\pi}_m(\theta, \vec{p}, u) \propto \exp\left(-\widehat{\mathcal{H}}(\theta, \vec{p})\right) p_U(u), \quad \widehat{\mathcal{H}}(\theta, \vec{p}) = \widehat{\mathcal{U}}(\theta) + \mathcal{K}(\vec{p}) \tag{3}$$

with

$$\widehat{\mathcal{U}}(\theta) = -\log \widehat{L}_m(\theta) - \log p_\Theta(\theta) \quad \text{and } \mathcal{K}(\vec{p}) = \frac{1}{2}\vec{p}\,' M^{-1}\vec{p}, \tag{4}$$

where M is a symmetric positive-definite matrix. In (3) we assume that the Hamiltonian $\widehat{\mathcal{H}}$ is separable. We propose a HMC-within-Gibbs method to sample from (3), alternating sampling from

1. $u|\theta, \vec{p}, y$ - Pseudo-marginal MH update (Section 3.3)
2. $\theta, \vec{p}\,|u, y$ - HMC update given u from Step 1 (Section 3.4).

This scheme has (3) as its invariant distribution. Integrating out the momentum variables yields $\overline{\pi}_m(\theta, u)$ in (1) and, further integrating out u, yields $\pi(\theta)$ if the likelihood estimator $\widehat{L}_m(\theta)$ is unbiased. Lindsten and Doucet (2016) propose the related pseudo-marginal HMC sampler, in which a momentum vector is also introduced for the auxiliary variables u. That scheme, however, is not applicable here as the pseudo-marginal variables we employ are discrete and not amenable to Hamiltonian dynamics themselves.

The next subsections describe in detail the two updates of our algorithm and explain why our approach does not compromise the Hamiltonian flow.

3.3. Updating the data subset

Given $\theta^{(j-1)}, \vec{p}^{\,(j-1)}$ and $u^{(j-1)}$, at iteration j we propose $u' \sim p_U(u)$ and set $u^{(j)} = u'$ with probability

$$\alpha_u = \min\left\{1, \frac{\widehat{L}_m(\theta^{(j-1)}; u')}{\widehat{L}_m(\theta^{(j-1)}; u^{(j-1)})}\right\}, \tag{5}$$

where the notation emphasizes that the estimators are based on different data subsets. If the proposal is rejected we set $u^{(j)} = u^{(j-1)}$.

Since u' is proposed independently of $u^{(j-1)}$, the log of the ratio in (5) can be highly variable, possibly getting the sampler stuck when the numerator is significantly overestimated. To prevent this, we implement the block update of u for data subsampling in Quiroz et al. (2019, 2018b) with G blocks, which gives a correlation ρ of roughly $1 - 1/G$ between the $\log \widehat{L}_m$ at the current and proposed draws (Tran et al., 2017; Quiroz et al., 2018b). Setting $G = 100$, gives $\rho \approx 0.99$, which helps the chain to mix well.

3.4. Updating the parameters

Given $u^{(j)}$, we use Hamilton's equations

$$\frac{d\theta_l}{dt} = \frac{\partial \widehat{\mathcal{H}}(\theta, \vec{p})}{\partial \vec{p}_l}, \quad \frac{d\vec{p}_l}{dt} = -\frac{\partial \widehat{\mathcal{H}}(\theta, \vec{p})}{\partial \theta_l}, \quad l = 1, \ldots, d, \tag{6}$$

to propose θ and $\vec{p}$. Note that this trajectory follows the Hamiltonian flow for $\widehat{\mathcal{H}}$ viewed as a function of θ and $\vec{p}$ for a given data subset selected by $u^{(j)}$, since $u^{(j)}$ is fixed through time t. We obtain the proposal as in standard HMC, using a leapfrog integrator with integration time ϵL, but with $\widehat{\mathcal{H}}$ in place of $\mathcal{H}$. Specifically, at iteration j, given the data subset $u^{(j)}$, if the leapfrog integrator starts at $(\theta^{(j-1)}, \vec{p}_0)$ with $\vec{p}_0 \sim \mathcal{K}(\vec{p})$ and ends at $(\theta_L, -\vec{p}_L)$, we let $(\theta^{(j)}, \vec{p}^{\,(j)}) = (\theta_L, -\vec{p}_L)$ with probability

$$\alpha_{\theta,\vec{p}} = \min\left\{1, \exp\left(-\widehat{\mathcal{H}}(\theta_L, -\vec{p}_L) + \widehat{\mathcal{H}}(\theta^{(j-1)}, \vec{p}_0)\right)\right\}, \tag{7}$$

with $\widehat{\mathcal{H}}$ in (3). If $(\theta_L, -\vec{p}_L)$ is rejected, we set $(\theta^{(j)}, \vec{p}^{\,(j)}) = (\theta^{(j-1)}, \vec{p}_0)$. In practice, it is unnecessary to store the sampled momentum.

Using the terminology in Betancourt (2015), we can think of the dynamics in (6) as generating a trajectory following a *modified level set* ($\widehat{\mathcal{H}}$), as opposed to the *exact level set* obtained using dynamics that do not subsample the data ($\mathcal{H}$). A key property of our framework is that the *same* estimate $\widehat{\mathcal{H}}$ is used in generating the discretized leapfrog trajectory as in the acceptance probability in (7). The connection between the modified level set $\widehat{\mathcal{H}}$ and its dynamics is kept intact, and thus the original energy conserving property of HMC remains, even for distant proposals. We therefore name our algorithm Hamiltonian Monte Carlo with Energy Conserving Subsampling (HMC-ECS). Note that energy conservation is only possible because of the pseudo-marginal mechanism where we update the subsample in each Gibbs iteration, thereby guaranteeing that our samples are from the target posterior based on all the data.

Betancourt (2015) illustrates the problems with using $\widehat{\mathcal{H}}$ for the dynamics, but $\mathcal{H}$ in the acceptance probability. Given a sensible step length ϵ, discretizing the Hamiltonian with a symplectic integrator introduces an error of $O(\epsilon^2)$ (Neal, 2011) relative to the modified level set and hence the discretization error is very small. Betancourt (2015) notes that the modified level set and the discretized trajectory based on it might be very far from the exact level set, resulting in low acceptance probabilities no matter how small ϵ is. SG-HMC (Chen et al., 2014) deliberately circumvents the disconnect problem by generating proposals from the trajectories based on a modified Hamiltonian, but skip the rejection step. The disadvantage of SG-HMC is therefore that the bias in the targeted posterior now grows with the step length ϵ. Keeping ϵ small makes SG-HMC very computationally demanding since a very large number of leapfrog steps are needed for distant proposals. In contrast, the dynamics of HMC-ECS target the subsampled Hamiltonian, and so maintain a high acceptance probability even for a large ϵ. The bias introduced by the subsampling is then confined to the pseudo-marginal step, which is chosen so that the bias is very small as our theoretical analysis below shows.

Algorithm 1 shows one iteration of our proposed HMC-ECS algorithm based on the leapfrog integrator using the estimated likelihood $\widehat{L}_m(\theta)$. The next two sections consider previously proposed likelihood estimators and show how we use them in HMC-ECS.

Algorithm 1: One iteration of HMC-ECS.

Input: Current position $u^{(j-1)}$, $\theta^{(j-1)}$, stepsize ϵ and integrating time ϵL

Propose $u' \sim p_U(u)$

Set $u^{(j)} \leftarrow u'$ with probability

$$\alpha_u = \min\left\{1, \frac{\widehat{L}_m(\theta^{(j-1)}; u')}{\widehat{L}_m(\theta^{(j-1)}; u^{(j-1)})}\right\},$$

else $u^{(j)} \leftarrow u^{(j-1)}$

Given $u^{(j)}$:

$\vec{p}_0 \sim \mathcal{K}(\vec{p})$; $\theta_0 \leftarrow \theta^{(j-1)}$; $\widehat{\mathcal{H}}_0 \leftarrow \widehat{\mathcal{H}}(\theta^{(j-1)}, \vec{p}_0)$

$\vec{p}_0 \leftarrow \vec{p}_0 - \frac{\epsilon}{2}\nabla_\theta \widehat{\mathcal{U}}(\theta_0)$

for $l = 1$ **to** L **do**

  $\theta_l \leftarrow \theta_{l-1} + \epsilon M^{-1}\vec{p}_{l-1}$

  if $i < L$ **then** $\vec{p}_l \leftarrow \vec{p}_{l-1} - \epsilon\nabla_\theta \widehat{\mathcal{U}}(\theta_l)$;

  else $\vec{p}_L \leftarrow \vec{p}_{L-1} - \frac{\epsilon}{2}\nabla_\theta \widehat{\mathcal{U}}(\theta_l)$;

end

$\vec{p}_L \leftarrow -\vec{p}_L$

$\widehat{\mathcal{H}}_L \leftarrow \widehat{\mathcal{H}}(\theta_L, \vec{p}_L)$

Set $(\theta^{(j)}, \vec{p}^{\,(j)}) \leftarrow (\theta_L, \vec{p}_L)$ with probability

$$\alpha_{\theta,\vec{p}} = \min\left\{1, \exp\left(-\widehat{\mathcal{H}}(\theta_L, -\vec{p}_L) + \widehat{\mathcal{H}}(\theta^{(j-1)}, \vec{p}_0)\right)\right\},$$

else $(\theta^{(j)}, \vec{p}^{\,(j)}) \leftarrow (\theta^{(j-1)}, \vec{p}_0)$

Output: $u^{(j)}$, $\theta^{(j)}$

4. Perturbed HMC-ECS

4.1. Efficient estimators of the log-likelihood

Quiroz et al. (2019) propose sampling m observations with replacement and estimate an additive log-likelihood

$$\ell(\theta) = \log L(\theta) = \sum_{k=1}^{n} \ell_k(\theta), \quad \ell_k(\theta) = \log p(y_k|\theta)$$

by the unbiased difference estimator

$$\widehat{\ell}_m(\theta) = \sum_{k=1}^{n} q_k(\theta) + \widehat{d}_m(\theta), \tag{8}$$

where

$$\widehat{d}_m(\theta) = \frac{1}{m}\sum_{i=1}^{m} \frac{\ell_{u_i}(\theta) - q_{u_i}(\theta)}{\omega_{u_i}(\theta)}, \quad u_i \in \{1, \ldots, n\} \text{ iid with } \Pr(u_i = k) = \omega_k, \tag{9}$$

and $q_k(\theta)$ are control variates. We continue to suppress dependence on n in the notation for many quantities introduced in this section. If the $q_k(\theta)$ approximate the $\ell_k(\theta)$ reasonably well, then we obtain an efficient estimator by taking $\omega_k = 1/n$ for all k. Quiroz et al. (2019) estimate $\sigma^2(\theta) = \mathrm{V}\left[\widehat{\ell}_m(\theta)\right]$ by

$$\widehat{\sigma}^2_m(\theta) = \frac{n^2}{m^2}\sum_{i=1}^{m}\left(d_{u_i}(\theta) - \overline{d}_u(\theta)\right)^2, \quad \text{with } d_{u_i}(\theta) = \ell_{u_i}(\theta) - q_{u_i}(\theta) \tag{10}$$

where $\overline{d}_u$ denotes the mean of the d_{u_i} in the sample $u = (u_1, \ldots, u_m)$.

To obtain efficient control variates, Quiroz et al. (2019) follow Bardenet et al. (2017) and let $q_k(\theta)$ be a second order Taylor approximation around a fixed central value $\theta^\star$,

$$q_k(\theta) = \ell_k(\theta^\star) + \nabla_\theta \ell_k(\theta^\star)^\top(\theta - \theta^\star) + \frac{1}{2}(\theta - \theta^\star)^\top H_k(\theta^\star)(\theta - \theta^\star), \quad H_k(\theta^\star) := \nabla_\theta \nabla_\theta^\top \ell_k(\theta^\star). \tag{11}$$

After processing the full data once before the MCMC to compute simple summary statistics, $\sum_{k=1}^{n} q_k(\theta)$ can be computed in $O(1)$ time (Bardenet et al., 2017).

4.2. Efficient estimators of the likelihood and its gradient

Quiroz et al. (2019) use the likelihood estimator

$$\widehat{L}_m(\theta) = \exp\left(\widehat{\ell}_m(\theta) - \frac{1}{2}\widehat{\sigma}^2_m(\theta)\right) \tag{12}$$

first proposed by Ceperley and Dewing (1999) and Nicholls et al. (2012). The motivation for this estimator is that it is unbiased for the likelihood if i) $\widehat{\ell}_m(\theta) \sim \mathcal{N}(\ell(\theta), \sigma^2(\theta))$ (justified by the central limit theorem) and ii) $\widehat{\sigma}^2_m(\theta)$ in (12) is replaced by the population quantity

$\sigma^2(\theta)$. However, $\sigma^2(\theta)$ is not available in practice and the estimator in (12) is biased. This bias makes the MCMC algorithm in Quiroz et al. (2019) target a slightly perturbed posterior $\overline{\pi}_m(\theta)$. Assuming that the expansion point $\theta^\star$ in the control variates is the posterior mode based on all the data, Quiroz et al. (2019) prove that

$$\int_\Theta |\overline{\pi}_m(\theta) - \pi(\theta)| d\theta = O\left(\frac{1}{nm^2}\right) \text{ and } |\mathrm{E}_{\overline{\pi}_m}[h(\theta)] - \mathrm{E}_\pi[h(\theta)]| = O\left(\frac{1}{nm^2}\right), \tag{13}$$

where $\overline{\pi}_m(\theta) = \mathrm{E}_{p_U}\left[\widehat{L}_m(\theta)\right]$ is the perturbed marginal for θ when using the likelihood estimator in (12). These results carry over to our Hamiltonian approach straightforwardly as we obtain the augmented target in Quiroz et al. (2019) after integrating out the momentum in (3) and using (12). Hence, the θ iterates from HMC-ECS converge to a perturbed posterior which may get arbitrarily close to the true posterior by increasing the subsample size m, or by increasing n and letting $m = O(n^\nu)$ for some $\nu > 0$. For example, if $\nu = 1/2$, then the above orders are $O(1/n^2)$ with respect to n. However, this extremely rapidly vanishing perturbation is usually not practically attainable since the result in (13) assumes that $\theta^\star$ is the posterior mode based on *all* data. Corollary 1 in Quiroz et al. (2019) proves rates under the more realistic assumption that $\theta^\star$ is the posterior mode based on a fixed subset of $\tilde{n} \ll n$ observations. If, for example, $\tilde{n} = O(\sqrt{n})$ then the rates in (13) become $O(1/\sqrt{n})$. Importantly, the optimal subsample size in this case becomes $m = O(\sqrt{n})$, which shows that HMC-ECS scales well with the size of the data. See Quiroz et al. (2019) for suggestions on how to get closer to the rates in (13) in a computationally tractable way.

In addition to estimating the likelihood and the log-likelihood, our Hamiltonian approach also needs to estimate a gradient. It is straightforward to modify (8) to instead provide an unbiased estimator of the gradient of the log-likelihood. With $\omega_k = 1/n$ and $\nabla_\theta q_k(\theta) = \nabla_\theta \ell_k(\theta^\star) + H_k(\theta^\star)(\theta - \theta^\star)$,

$$\nabla_\theta \widehat{\ell}_m(\theta) = A(\theta^\star) + B(\theta^\star)(\theta - \theta^\star) + \frac{n}{m}\sum_{i=1}^m \left(\nabla_\theta \ell_{u_i}(\theta) - \nabla_\theta q_{u_i}(\theta)\right), \tag{14}$$

where

$$A(\theta^\star) := \sum_{k=1}^n \nabla_\theta \ell_k(\theta^\star) \in \mathbb{R}^d \quad \text{and } B(\theta^\star) := \sum_{k=1}^n H_k(\theta^\star) \in \mathbb{R}^{d\times d}$$

are obtained at the cost of computing over the full data set once before the MCMC since $\theta^\star$ is fixed. It is also straightforward to compute $\nabla_\theta \widehat{\sigma}^2_m(\theta)$ using (10) with this choice of control variate.

It is important to note that the perturbation in the targeted posterior in perturbed HMC-ECS is independent of the step length in the leapfrog iterations, ϵ. Hence, we can generate distant proposals from a small number of leapfrog steps without increasing the posterior bias.

5. Signed HMC-ECS

We now present an alternative HMC-ECS algorithm based on the Block-Poisson estimator in Quiroz et al. (2018b). This algorithm gives simulation consistent estimates of expectations with respect to the true posterior density without any perturbation.

5.1. The block-Poisson estimator

Quiroz et al. (2018b) propose the block-Poisson estimator, formed by sampling $\mathcal{X}_l \sim \mathrm{Pois}(1)$ for $l = 1, \ldots \lambda$, and computing $\widehat{d}_m^{(h,l)}$, $h = 1, \ldots, \mathcal{X}_l$, using (9) based on a mini-batch sample size m, and then estimate the likelihood by

$$\widehat{L}_m(\theta) = \exp\left(\sum_{k=1}^n q_k(\theta)\right)\prod_{l=1}^\lambda \xi_l, \qquad \xi_l = \exp\left(\frac{a+\lambda}{\lambda}\right)\prod_{h=1}^{\mathcal{X}_l}\left(\frac{\widehat{d}_m^{(h,l)}(\theta) - a}{\lambda}\right), \tag{15}$$

where λ is a positive integer, a is a real number and $\xi_l = \exp\left((a+\lambda)/\lambda\right)$ if $\mathcal{X}_l = 0$. Pois(1) denotes the Poisson distribution with mean 1. Note that the total subsample size $m\lambda\mathcal{X}_l$ is random with mean $m\lambda$ in a given MCMC iteration.

Since $\widehat{L}_m(\theta)$ is unbiased for $L(\theta)$, defining the augmented density as in (3) gives

$$\int_U \int_{\mathcal{P}} \overline{\pi}_m(\theta, \vec{p}, u) d\vec{p} du = \pi(\theta).$$

Hence, if (3) is a proper density using (15), we obtain samples from the desired marginal for θ. However, (3) is a proper density only if $\tau := \Pr\left(\widehat{L}_m(\theta) \geq 0\right) = 1$, which requires that a in (15) is a lower bound of $\widehat{d}_m^{(h,l)}$ (Jacob and Thiery, 2015) which results in a prohibitively costly estimator (Quiroz et al., 2018b). Instead, we follow Quiroz et al. (2018b) who use the approach of Lyne et al. (2015) for exact inference on an expectation of an arbitrary function $\psi(\theta)$ with respect to $\pi(\theta)$. For our Hamiltonian approach, this entails defining a proper augmented density,

$$\widetilde{\pi}_m(\theta, \vec{p}, u) \propto \left|\widehat{L}_m(\theta)\right| p_\Theta(\theta) p_U(u) \exp\left(-\mathcal{K}(\vec{p})\right), \tag{16}$$

and writing

$$\mathrm{E}_\pi[\psi] = \frac{\int_\Theta \psi(\theta) L(\theta) p_\Theta(\theta) d\theta}{\int_\Theta L(\theta) p_\Theta(\theta) d\theta} = \frac{\int_{\mathcal{U}}\int_{\mathcal{P}}\int_\Theta \psi(\theta) S(\theta,u)\widetilde{\pi}(\theta,\vec{p},u) d\theta d\vec{p} du}{\int_{\mathcal{U}}\int_{\mathcal{P}}\int_\Theta S(\theta,u)\widetilde{\pi}(\theta,\vec{p},u) d\theta d\vec{p} du} = \frac{\mathrm{E}_{\widetilde{\pi}}[\psi S]}{\mathrm{E}_{\widetilde{\pi}}[S]}, \tag{17}$$

where $S(\theta, u) = \mathrm{sign}\left(\widehat{L}_m(\theta)\right)$ and $\mathrm{sign}(\cdot) = 1$ if $\widehat{L}_m(\theta) \geq 0$ and $\mathrm{sign}(\cdot) = -1$ otherwise. Equation (17) suggests running the HMC-ECS sampler outlined in Section 3.2 on the target (16), and then estimating (17) by

$$\widehat{I}_R = \frac{\sum_{j=1}^R \psi(\theta^{(j)}) s^{(j)}}{\sum_{j=1}^R s^{(j)}},$$

where $s^{(j)}$ is the sign of the estimate at the jth iteration. We follow Quiroz et al. (2018b) and use the term *signed PM* for any pseudo-marginal algorithm that uses the technique in Lyne et al. (2015) where a pseudo-marginal sampler is run on the absolute value of the estimated posterior and subsequently sign-corrected by importance sampling. Similarly, we call the algorithm described in this section *signed HMC-ECS*.

The block-Poisson estimator in (15) has more tuning parameters than the estimator in (12). Quiroz et al. (2018b) extend the optimal tuning approach in Pitt et al. (2012) to the

signed pseudo-marginal algorithm with the block-Poisson estimator. The Computational Time (CT) measure in (2) now becomes

$$\text{CT} := \frac{\text{IF}}{(2\tau-1)^2} \times \text{Total number of density and gradient evaluations}, \tag{18}$$

where $\tau := \Pr\left(\widehat{L}_m(\theta) \geq 0\right)$. Quiroz et al. (2018b) derive analytical expressions for both $\text{V}[\log|\widehat{L}_m|]$ and $\Pr(\widehat{L}_m \geq 0)$ needed to optimize CT. The fact that Quiroz et al. (2018b) take $\Pr(\widehat{L}_m \geq 0)$ into account when tuning the algorithm avoids the instability from changing signs in signed PMMH. Quiroz et al. (2018b) also consider optimal tuning when correlating the estimators of the log of the likelihood at the current and proposed values of the MCMC. This correlation, ρ, is achieved by only updating u for a subset of the products in (15) in each iteration, keeping the others fixed. Quiroz et al. (2018b) show that if u is updated in κ products, then $\rho \approx 1 - \kappa/\lambda$.

6. Applications

6.1. Model

We consider the logistic regression

$$p(y_k|x_k,\theta) = \left(\frac{1}{1+\exp(-x_k^\top\theta)}\right)^{y_k}\left(\frac{1}{1+\exp(x_k^\top\theta)}\right)^{1-y_k}, \quad \text{with } p_\Theta(\theta) = \mathcal{N}(\theta|0,\lambda_\theta^{-2}I),$$

where λ_θ is a global shrinkage factor which we treat as constant for simplicity. We estimate the model on two large data sets described below.

6.2. Competing algorithms and performance measure

We compare the performance of both the perturbed and the signed HMC-ECS algorithms against SGLD and SG-HMC. All subsampling methods use the same control variates based on a second order Taylor expansion for comparability; see Section 6.8 for experiments with control variates based on lower order Taylor expansions. The expansion point $\theta^\star$ is unique to each experiment and is discussed later.

Following Vollmer et al. (2016); Baker et al. (2017), we implement SGLD using a fixed small step size ϵ instead of decreasing it (which gives worse results). This gives the following dynamics after discretization

$$\theta_i = \theta_{i-1} - \frac{\epsilon}{2}\nabla_\theta\widehat{\mathcal{U}}(\theta_i) + \zeta_i, \quad i = 1,\ldots,R,$$

where $\widehat{\mathcal{U}}(\theta) = -\nabla_\theta\widehat{\ell}_m(\theta) - \log p_\Theta(\theta)$ with $\nabla_\theta\widehat{\ell}_m(\theta)$ in (14) and $\zeta_i \sim \mathcal{N}(0,\epsilon I)$.

Following Chen et al. (2014), we implement SG-HMC using a discretized dynamics with momentum $\vec{p}$ with covariance matrix M of the form

$$\begin{aligned}\theta_l &= \theta_{l-1} + \epsilon M^{-1}\vec{p}_{l-1}\\ \vec{p}_l &= \vec{p}_{l-1} - \epsilon\nabla_\theta\widehat{\mathcal{U}}(\theta_l) - \epsilon C M^{-1}\vec{p}_{l-1} + \zeta_l, \quad l = 1,\ldots,L,\end{aligned}$$

where $\widehat{\mathcal{U}}(\theta) = -\nabla_\theta\widehat{\ell}_m(\theta) - \log p_\Theta(\theta)$ with $\nabla_\theta\widehat{\ell}_m(\theta)$ in (14), $\zeta_l \sim \mathcal{N}(0, 2(C-\widehat{B})\epsilon)$. We set $\widehat{B} = 0$ (Chen et al., 2014) and $C = I$ (Ma et al., 2015).

The algorithms are compared with respect to CT as defined in (2) when the likelihood estimator is non-negative, where IF is computed using the CODA package in R (Plummer et al., 2006). For signed HMC-ECS, the CT is given by (18) and we estimate τ by the fraction of positive signs. Note that CT does not take into account that some of the algorithms can give a substantially biased estimate of the posterior, and we assess the bias separately.

The Relative Computational Time (RCT) between algorithm $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined as

$$\text{RCT}_{\mathcal{A}_1,\mathcal{A}_2} = \frac{\text{CT}_{\mathcal{A}_2}}{\text{CT}_{\mathcal{A}_1}}. \tag{19}$$

6.3. Tuning and settings of our algorithms

We first outline how our algorithms are tuned. These settings are used for all the experiments unless stated otherwise.

The choice of the positive-definite mass matrix M in (4) is crucial for the performance of any HMC type algorithm: an M that closely resembles the covariance of the posterior facilitates sampling, especially when θ is highly correlated in the posterior (Neal, 2011; Betancourt, 2017). In logistic regression, we set $M = -\Sigma^{-1}(\theta^\star)$, where $\Sigma(\theta^\star)$ is the Hessian of the log posterior evaluated at some $\theta^\star$. We initialize $M = I$ and, during a burn-in period of 1,000 iterations, update M every 200 iterations based on the new $\theta^\star$. We use the same tuning of M in HMC-ECS when updating θ and $\vec{p}$ conditional on the data subsample u. However, since it is impractical to compute the Hessian of the conditional posterior at each iteration, we use the full data set when evaluating M (and include the cost in the CT), which performs well in practice although we stress that it is not optimal.

To select the step size ϵ in the leapfrog integrator, we use the dual averaging approach of Hoffman and Gelman (2014), which requires a predetermined trajectory length ϵL. We find that $\epsilon L = 1.2$ is sensible for our examples. The dual averaging algorithm uses this trajectory length and adaptively changes ϵ during the burn-in period in order to achieve a desired level of acceptance rate δ. We follow Hoffman and Gelman (2014) and set $\delta = 0.8$. The tuning for the logistic regression is relatively simple since Σ can be computed analytically, which is useful for both setting M and ϵ. More complex models are harder to tune, but this is unlikely to influence the comparisons between HMC-ECS and the other algorithms here, which is our primary concern. We stress that these tuning issues are inherent to HMC itself and not due to our subsampling approach. This strategy gives $\epsilon = 0.2$ and $L = 6$ for our algorithm which is the default in our experiments unless otherwise stated.

In HMC-ECS we generate subsamples with a correlation of $\rho = 0.99$. The subsample size m in the perturbed HMC-ECS approach is set according to the guidelines in Quiroz et al. (2019), see Section 4. In the signed HMC-ECS approach, we follow Quiroz et al. (2018b) who set the mini-batch size $m = 30$ and set λ optimally to minimize CT according to the formulas in Quiroz et al. (2018b). For our examples, $\lambda = 100$ and $\lambda = 200$, for the HIGGS and bankruptcy data sets, respectively.

6.4. Tuning and settings of the competing algorithms

We run full data HMC using the tuning strategy for ϵ and M outlined in Section 6.3 and use $M = \Sigma^{-1}(\theta^\star)$ with $\theta^\star$ as the posterior mean in SG-HMC. This favours SG-HMC over HMC-ECS, which needs to learn $\theta^\star$ as the algorithm progresses.

We are unaware of any tuning strategies for setting ϵ in SGLD and SG-HMC and therefore use trial and error. We find that the value of ϵ depends on which data set or algorithm was considered; see Sections 6.5 and 6.6. For SG-HMC we also need to set the number of steps L and we explore two choices. First, notice that HMC-ECS uses the trajectory length $\epsilon L = 1.2$. We thus set $L = 1.2/\epsilon$ so that SG-HMC can traverse the space as swiftly as HMC-ECS. We also compare with SG-HMC using a value of L which gives the same number of likelihood and gradient evaluations as HMC-ECS. For SGLD, we run the algorithm for R iterations that correspond to the number of gradient evaluations used post burn-in in HMC-ECS. For example, $R = 12{,}000$ iterations if $L = 6$ and HMC-ECS performs $2,000$ iterations.

Finally, we use the same m as HMC-ECS for SGLD and SG-HMC, as it is outside the scope of this paper to derive optimality results for those algorithms.

6.5. Results for the HIGGS data

Baldi et al. (2014) use the HIGGS data set, which contains 11 million observations, with a binary response *detected particle* predicted by 29 covariates. We use 10.5 millions observations for training and 500,000 for testing.

Unless stated otherwise, we start all algorithms at a $\theta^\star$ obtained as the posterior mean from running HMC on 1% of a randomly chosen subset of the data. This $\theta^\star$ is also used to initialize the control variates. We first run the algorithms using a full mass matrix M, chosen as explained above. The subsample size was set to $m = 1,300$ for all methods. For SG-HMC and SGLD, $\epsilon = 0.06$ and $\epsilon = 0.000001$ are used, respectively. The resulting L is 20 for SG-HMC.

Table 1 displays the CT for each algorithm compared to the perturbed HMC-ECS algorithm. The table shows the minimum, median and maximum RCT across all parameters. The best algorithm is perturbed HMC-ECS closely followed by signed HMC-ECS, both have RCTs that are roughly three times better than the best competitor SG-HMC. Although our metrics do not allow for direct comparison between biased (our approaches, SG-HMC and SGLD) methods and unbiased methods (HMC), we note that implementing HMC using the full data in this example is, for the perturbed and exact approaches respectively, 642.8 and 554.1 times more expensive in terms of posterior density and gradient evaluations.

RCT does not take the bias into account, however. Figure 1 displays kernel density estimates of the marginal posterior of four randomly selected parameters, and Figure 2 plots the posterior mean and variance for all the parameters from each algorithm against the true posterior mean and variance obtained from HMC based on the full data set. Figures 1 and 2 clearly show that both HMC-ECS algorithms and SG-HMC do a very good job in approximating the posterior, while SGLD gives biased estimates. We have also added the results for SG-HMC and SGLD without control variates as variance reduction is not always optimal for these algorithms (Chatterji et al., 2018). In this example, control variates

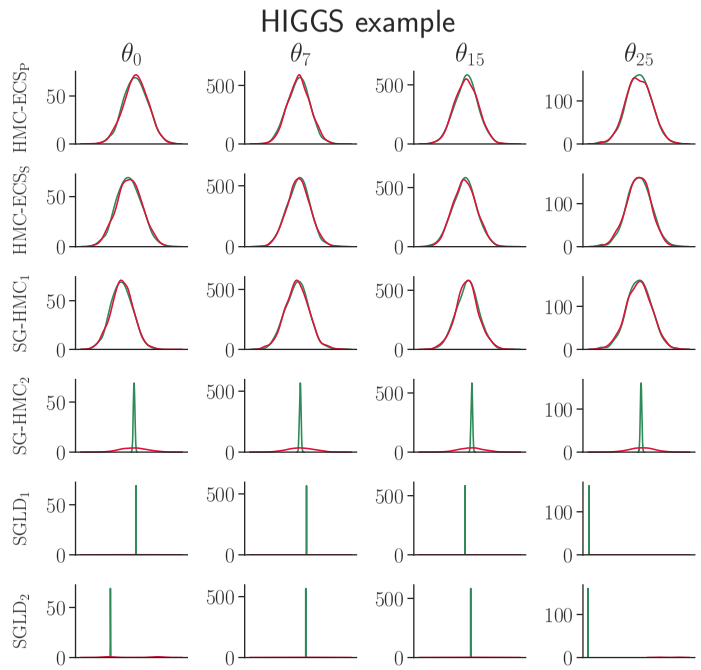


Figure 1: HIGGS data. The figure shows kernel density estimates of the posterior from the compared subsampling algorithms for four randomly selected parameters (green lines) and the corresponding posterior from HMC on the full data set (red lines). All algorithms use the negative inverse Hessian from all the data as mass matrix. HMC-ECS$_P$ and HMC-ECS$_S$ denote, respectively, the perturbed and signed HMC-ECS$_S$. For SGLD and SG-HMC, subscript 1 refers to the second order control variate and subscript 2 refers to the version without control variates.

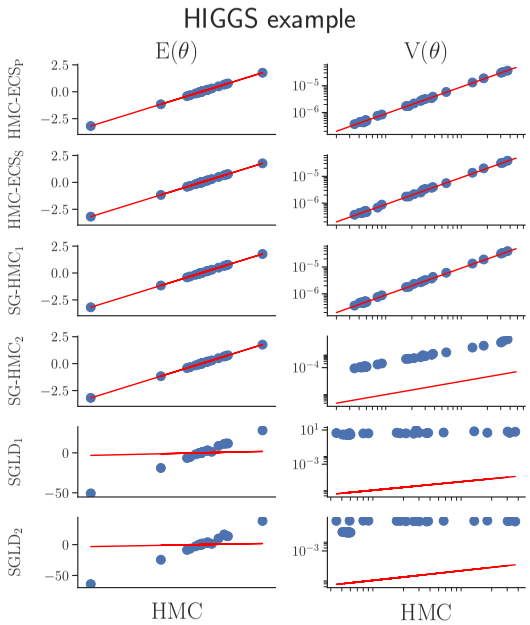


Figure 2: HIGGS data. The figure plots the estimated posterior mean and variance for all parameters from the subsampling algorithm against their true values obtained by HMC on the full data set. All algorithms use the negative inverse Hessian from all the data as mass matrix. HMC-ECS$_{\mathrm{P}}$ and HMC-ECS$_{\mathrm{S}}$ denote, respectively, the perturbed and signed HMC-ECS$_{\mathrm{S}}$. For SGLD and SG-HMC, subscript 1 refers to the second order control variate and subscript 2 refers to the version without control variates.

RCT	Signed HMC-ECS	SG-HMC	SGLD
min	0.8	2.43	3.58
median	1.15	2.97	12.46
max	1.95	4.08	326.80

Table 1: HIGGS data. Relative computational time compared to perturbed HMC-ECS. The computational cost is (number of likelihood/gradient evaluations)$\times$IF. For HMC-ECS and signed HMC-ECS the cost is computed for the entire run including training and warmup period. The RCT for the stochastic gradient methods are based on post-burnin iterations only.

	Perturbed HMC-ECS	Signed HMC-ECS
$\alpha_{\theta,\vec{p}}$	0.980	0.979
α_u	1	0.993
L	6	6
IF	2.185	2.192
ESS	927	922
$\widehat{\tau}$		1
$100(m/n)$	0.012	0.029

Table 2: HIGGS data. Summary of settings and efficiencies of HMC-ECS. The table shows the average acceptance probabilities (as a benchmark HMC has 0.980) in the post burn-in period for the two Gibbs steps, the number of steps L in the integrator used to obtain a predetermined trajectory length $\epsilon L = 1.2$, the average Inefficiency Factor (IF) (as a benchmark HMC has 2.084), the Effective Sample Size ESS $= R/$IF, the estimated probability of a positive likelihood estimator τ and the percentage of data used by each of the algorithms.

are indeed helpful, except for SGLD which does not provide an accurate approximation regardless of which control variate is used.

We conclude this example by demonstrating that HMC-ECS can safely be used for obtaining the predictive distribution. Figure 3 shows that the Receiver Operating Characteristic (ROC) curve for the $500,000$ test observations obtained with either of the two HMC-ECS algorithms are indistinguishable from the ROC curve obtained with HMC on the full data set.

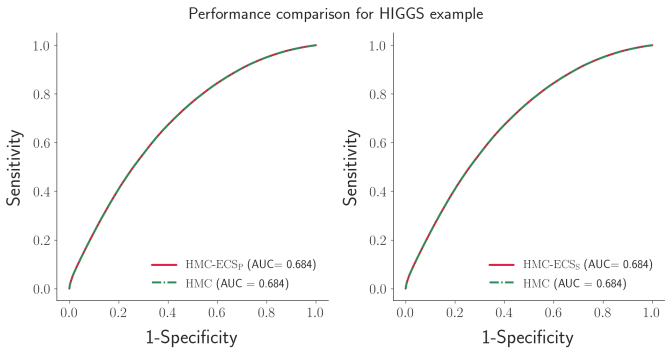


Figure 3: Prediction performance for the HIGGS data. The figure shows the Receiver Operating Characteristic (ROC) curves for the 500,000 test observations with HMC and HMC-ECS$_P$ (left panel, perturbed HMC-ECS) and HMC-ECS$_S$ (right panel, signed HMC-ECS).

6.6. Results for the Bankruptcy data

This data set contains annual observations on the bankruptcy status (binary y) of Swedish firms in the time period 1991-2008. We follow Giordani et al. (2014) and model the log odds of the firm failure probability as a non-linear function of six firm-specific financial ratios and two macroeconomic variables using an additive spline model. Giordani et al. (2014) estimate the model using frequentist methods. We use a Bayesian approach with an 81 dimensional posterior distribution (an intercept and 10 basis spline functions for each covariate) given the $n = 4{,}748{,}089$ firm-year observations.

Experimentation shows that $\epsilon = 0.02$ is a sensible choice for SG-HMC. Using the same trajectory length as in HMC-ECS gives $L = 60$ for SG-HMC. We also compare with SG-HMC using $L = 7$ for which SG-HMC has the same number of gradient evaluations as HMC-ECS has gradient and likelihood evaluations. For SGLD, $\epsilon = 0.00002$ is a sensible choice.

The subsample size for perturbed HMC-ECS was initially set to the optimal $m = 62{,}000$ following the guidelines in Quiroz et al. (2019) based on the initial value for $\theta^\star$. We then ran perturbed HMC-ECS for 100 iterations to obtain a better $\theta^\star$ and recalibrated to the now optimal $m = 1{,}000$ for this improved $\theta^\star$. This improved $\theta^\star$ is also used to tune λ in signed HMC-ECS, as explained in Section 5. All iterations used for tuning are included in the computational cost.

Figures 4 and 5 clearly show that the two HMC-ECS algorithms give better posterior approximations. The figures also show that SG-HMC with $L = 60$ leapfrog steps performs reasonably well in terms of accuracy and much better than SGLD, although some expectation estimates are biased. SG-HMC with $L = 7$ has substantially higher inefficiency factors and these figures show the degraded accuracy of the algorithm.

RCT	HMC-ECS$_S$	SG-HMC$_1$	SG-HMC$_2$	SGLD
min	1.2	6.8	48.6	53.9
median	1.6	9.5	100.2	230.1
max	2.6	682.3	246.7	2784.2

Table 3: Bankruptcy data. Relative computational time compared to perturbed HMC-ECS. For the two HMC-ECS algorithms, the cost is computed for the entire run including training and warmup period. The RCT with respect to stochastic gradient methods are based on post-burnin iteration only. HMC-ECS$_S$ is the signed HMC-ECS. SG-HMC$_1$ and SG-HMC$_2$ denote, respectively, the SG-HMC with $L = 60$ and $L = 7$ leapfrog steps.

	Perturbed HMC-ECS	Signed HMC-ECS
$\alpha_{\theta,\vec{p}}$	0.967	0.962
α_u	0.994	0.964
L	6	6
IF	2.202	2.31
ESS	912	871
$\widehat{\tau}$		1
$100(m/n)$	0.021	0.126

Table 4: Bankruptcy data. Summary of settings and efficiencies of HMC and HMC-ECS. The table shows the average acceptance probabilities (as a benchmark HMC has 0.966) in the post burn-in period for the two Gibbs steps, the number of steps L in the integrator used to obtain a predetermined trajectory length $\epsilon L = 1.2$, the average Ineffiency Factor (IF) (as a benchmark HMC has 2.195), the Effective Sample Size ESS $= R/$IF, the estimated probability of a positive likelihood estimator τ and the percentage of data used by each of the algorithms.

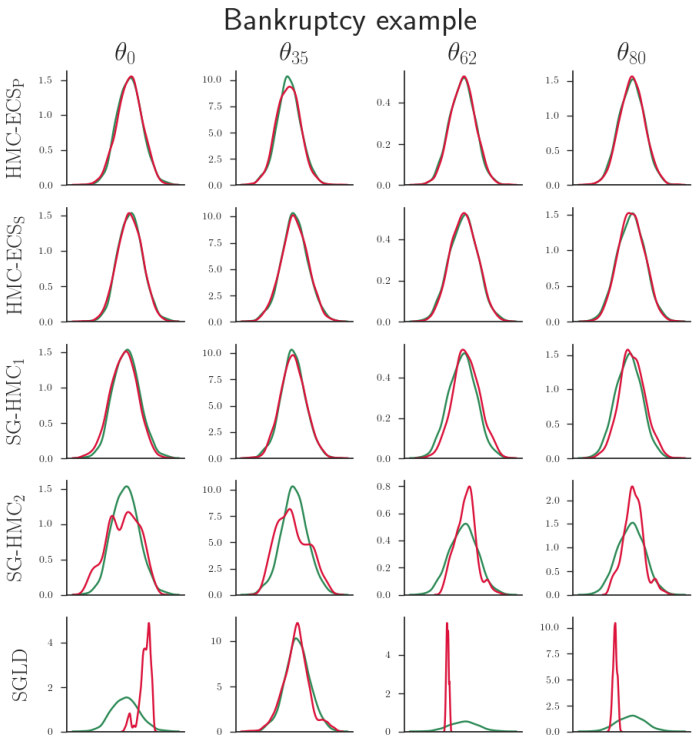


Figure 4: Bankruptcy data. The figure shows kernel density estimates of the posterior from the compared subsampling algorithms for four randomly selected parameters (green lines) and the corresponding posterior from HMC on the full data set (red lines). HMC-ECS$_P$ and HMC-ECS$_S$ denote, respectively, the perturbed and signed HMC-ECS$_S$. SG-HMC$_1$ and SG-HMC$_2$ denote, respectively, the SG-HMC with $L = 60$ and $L = 7$ leapfrog steps.

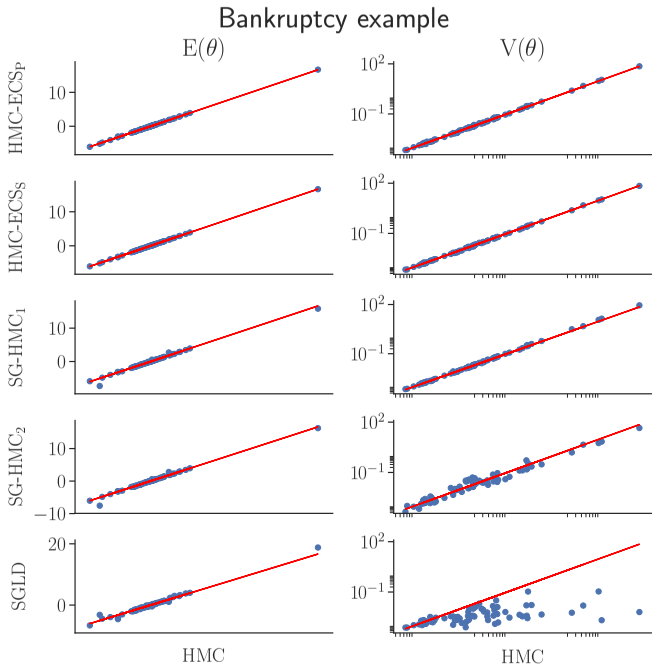


Figure 5: Bankruptcy data. The figure plots the estimated posterior mean and variance for all parameters from the subsampling algorithm against their true values obtained by HMC on the full data set. HMC-ECS$_P$ and HMC-ECS$_S$ denote, respectively, the perturbed and signed HMC-ECS$_S$. SG-HMC$_1$ and SG-HMC$_2$ denote, respectively, the SG-HMC with $L = 60$ and $L = 7$ leapfrog steps.

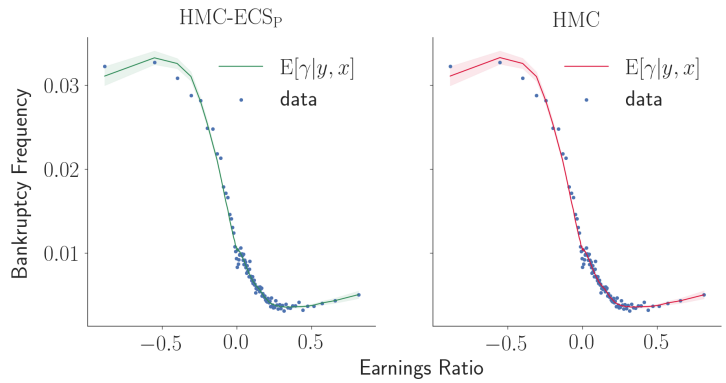


Figure 6: Realized and predicted bankruptcy probabilities as a function of the covariate *Earnings ratio* from the perturbed HMC-ECS (left panel) and HMC (right panel). Realized mean bankruptcy probabilities (blue dots) are computed by dividing the data into 100 equally sized groups based on the earnings ratio variable and estimating the bankruptcy probability by the fraction of bankrupt firms in each group. The solid line is the predictive mean and the shaded regions are point wise 90% equal tail posterior credible intervals.

Figure 6 shows the probability of bankruptcy for the fitted model and the empirical bankruptcy frequencies as a function of one of the covariates, Earnings ratio, with details in the caption. The posterior mean and posterior predictive intervals obtained by HMC and the perturbed HMC-ECS are indistinguishable.

Finally, we note that implementing HMC using the full data in this example is, for the perturbed and exact approach respectively, 478.7 and 311.5 times more expensive in terms of posterior density and gradient evaluations.

6.7. Scalability of HMC-ECS

Beskos et al. (2013) show that, in an optimally tuned HMC algorithm, the step size ϵ needs to be scaled as $O(d^{-1/4})$ to keep the acceptance probability constant as the dimension d increases. This is more favorable than the rate $O(d^{-1/3})$ of Langevin Monte Carlo (Roberts and Rosenthal, 1998). We have argued that since our algorithm is performing a HMC step using a Hamiltonian based on a subset of the data it should scale with dimension similarly to HMC.

We set out to test this hypothesis empirically as follows. First, we consider a sequence of d, obtained as $d = 2^h$, $h = 1, \ldots, 8$ and obtain eight simulated data sets with $n = 10,000$ each. For each d, we run the dual averaging algorithm as described in Section 6.3 to find the optimal $\epsilon$and check if $\epsilon = O(d^{-1/4})$ is reasonable. In agreement with Beskos

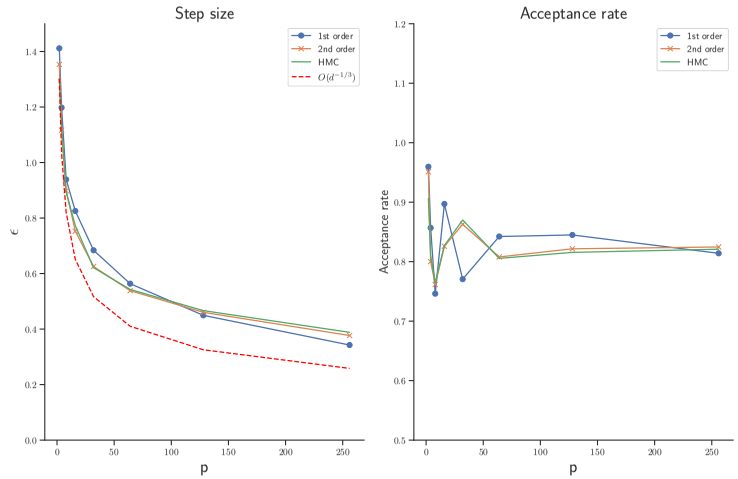


Figure 7: Empirical illustration of the scaling of HMC-ECS, see Section 6.7 for the experimental settings. The left figure shows the optimal step size as a function of the dimension. HMC ($O(d^{-1/4})$) and Langevin ($O(d^{-1/3})$) are plotted for reference. The right figure shows the acceptance rate targeted for optimality. The IF of our methods are close to 1, regardless of if the 1st or 2nd control variate is used. The variance of the log-likelihood estimator for the second order control variate is nearly zero (quadratic target) and for the first order control variate is kept around 1 (Pitt et al., 2012) by selecting m appropriately.

et al. (2013) we set M optimally from the curvature of the conditional target posterior. This is easily achieved by considering a Gaussian regression model where we set the prior $p_\Theta(\theta) = \mathcal{N}(0, 5^2 I_d)$, such that the optimal M for the conditional target is, assuming the bias-correction term to be negligible,

$$M = \frac{n}{m} \sum_{i=1}^{m} \mathbf{X}_{u_i}^\top \mathbf{X}_{u_i} + \frac{1}{5^2},$$

where $\mathbf{X}_{u_i}$ denotes the u_ith row of the design matrix. We note that, since the Gaussian model is quadratic in its log-density, the second order control variate will yield a perfect fit, i.e. the variance of $\widehat{l}_m(\theta)$ is zero. Thus, we also experiment with a first order control variate. We scale m to maintain the variance around 1 (Pitt et al., 2012). Figure 7 shows the results and we deduce that the algorithm does indeed maintain scalability.

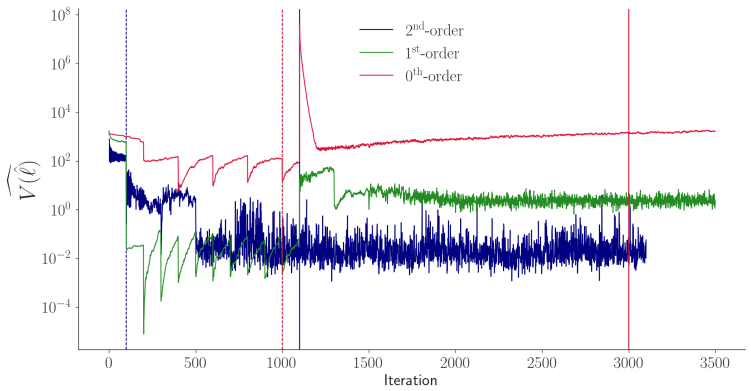


Figure 8: Variance of $\widehat{l}_m$. The figure shows the estimated variance as a function of the iterations for three different orders in the Taylor expansion for the control variates. The dashed vertical lines correspond to the end of the training period used the solid vertical line corresponds to the end of the burn-in period.

6.8. Limitations of subsampling HMC

Variance reduction by control variates is crucial in any subsampling MCMC algorithm. This subsection explores the role of control variates by successively degrading the quality of the control variates by lowering the order of their Taylor approximation. Figure 8 shows the estimated variance of $\widehat{l}_m(\theta)$ as a function of the iterates when the control variates are based on a Taylor series expansions of different orders. We note that the algorithm survives a substantial variance during the training iterations when applying the first order control variates, and this variance eventually settles down once a sensible $\theta^\star$ is found. Figure 9 shows that the expectation and variance estimates are very accurate for the first order control variates. Figures 8 and 9 also shows that control variates of zero order are too crude for HMC-ECS in this example. While it seems that the competing methods are more robust to the quality of the control variates, we again stress that all competitors are placed in the unrealistically favorable scenario of having M equal to the inverse Hessian evaluated at the posterior mean.

The control variates (Bardenet et al., 2017) used in this paper are quadratic in θ centered around $\theta^\star$. This means that they are expected to work well for any model which has a log-density that is reasonably quadratic in θ in a neighborhood of $\theta^\star$. Examples are Poisson regression (Quiroz et al., 2018c) and Student-t regression (Quiroz et al., 2019). While the locally approximately quadratic feature is found in many models, there are clearly models in which this local approximation can be poor, for example deep neural nets. To find efficient control variates, which are also computationally feasible, in a class of complex models remains an open challenging problem. We stress however that this is a problem for

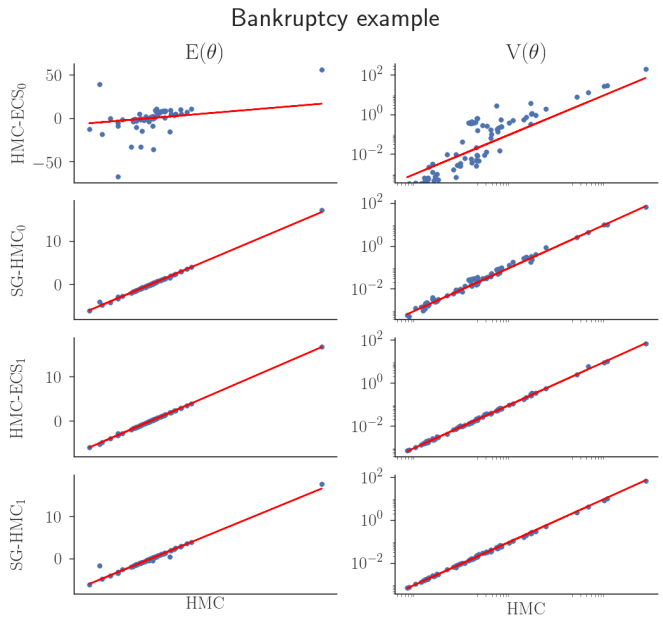


Figure 9: Results for the zeroth and first order control variate. The figure shows the estimates of posterior expectations and posterior variances using the zeroth order control variate (upper panel) and first order control variate (lower panel). All comparisons are versus HMC which represents the ground truth.

all existing subsampling MCMC approaches, and any progress on improved control variates can be straightforwardly incorporated into HMC-ECS.

7. Conclusions and Future Research

We propose a method to speed up Bayesian inference while maintaining high sampling efficiency in moderately high-dimensional parameter spaces by combining data subsampling and Hamiltonian Monte Carlo such that the energy is conserved. We show how to implement the method using two estimators of the likelihood. The first implementation, which we refer to as perturbed HMC-ECS, produces iterates from a perturbed density that will get arbitrarily close to the true density, as measured by the total variation metric, at the rate $O(n^{-1}m^{-2})$. The second implementation, which we refer to as signed HMC-ECS, gives iterates which are then used in an importance sampling estimator to obtain a simulation consistent estimator of the expectation of any posterior functional.

We apply the methods to simulate from the posterior distribution in two data sets, with $d = 29$ and $d = 81$ dimensions, respectively. Our two HMC-ECS algorithms perform highly accurate inference, comparable to HMC without subsampling, but are computationally much faster. This is a major step forward since Bardenet et al. (2017) and Quiroz et al. (2019) demonstrate that most subsampling approaches cannot even beat standard MH without subsampling on toy examples with $d = 2$ and highly redundant data. We also show that HMC-ECS is very competitive against SGLD and SG-HMC, both in terms of sampling efficiency and accuracy.

Control variates to reduce the variance of subsampling estimators are well known to be crucial for any subsampling MCMC algorithm. We use very efficient control variates based on a second order Taylor expansion in our applications, but explore the effects of less accurate control variates. We find that HMC-ECS still performs well with a cruder first order approximation, but that a Taylor approximation of order zero is too crude and gives a too large variance for HMC-ECS.

Similarly to HMC, HMC-ECS is difficult to tune. Self-tuning algorithms such as the no-U-Turn sampler (Hoffman and Gelman, 2014) have been proposed for HMC and it would be interesting to see if our ideas can be applied there. It would also be interesting to consider Riemann Manifold HMC (Girolami and Calderhead, 2011), which has been demonstrated to be very effective when a high-dimensional posterior exhibits strong correlations. Scaling up such an algorithm opens up the possibility of simulating the posterior density of highly complex models with huge data sets. Finally, until recently, one of the limitations of HMC was its inability to cope with discrete parameters. Nishimura et al. (2017) overcomes this limitation and extending HMC-ECS in this direction would be an interesting undertaking.

Acknowledgments

We thank the action editor and two reviewers for constructive comments that improved both the content and presentation of the paper.

Matias Quiroz and Robert Kohn were partially supported by Australian Research Council Center of Excellence grant CE140100049. Mattias Villani was partially supported by Swedish Foundation for Strategic Research (Smart Systems: RIT 15-0097).

References

Christophe Andrieu and Gareth O Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

Jack Baker, Paul Fearnhead, Emily B Fox, and Christopher Nemeth. Control variates for stochastic gradient MCMC. *arXiv preprint arXiv:1706.05439*, 2017.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5(4308), 2014.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In *Proceedings of The 31st International Conference on Machine Learning*, pages 405–413, 2014.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(47):1–43, 2017.

José M Bernardo and Adrian FM Smith. *Bayesian Theory*. John Wiley & Sons, 2001.

Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19(5A):1501–1534, 2013.

Michael Betancourt. The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling. In *International Conference on Machine Learning*, pages 533–540, 2015.

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Joris Bierkens, Paul Fearnhead, and Gareth Roberts. The zig-zag process and super-efficient sampling for Bayesian analysis of big data. *The Annals of Statistics*, 47(3):1288–1320, 2019.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Nicolas Brosse, Alain Durmus, and Eric Moulines. The promises and pitfalls of stochastic gradient Langevin dynamics. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8277–8287. Curran Associates, Inc., 2018.

DM Ceperley and M Dewing. The penalty method for random walks with uncertain energies. *The Journal of Chemical Physics*, 110(20):9812–9820, 1999.

Niladri Chatterji, Nicolas Flammarion, Yian Ma, Peter Bartlett, and Michael Jordan. On the theory of variance reduction for stochastic gradient Monte Carlo. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 764–773. PMLR, 2018.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

George Deligiannidis, Arnaud Doucet, and Michael K Pitt. The correlated pseudomarginal method. *Journal of the Royal Statistical Society. Series B (Methodological)*, 80(5):839–870, 2018.

Arnaud Doucet, MK Pitt, George Deligiannidis, and Robert Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

Kumar Avinava Dubey, Sashank J Reddi, Sinead A Williamson, Barnabas Poczos, Alexander J Smola, and Eric P Xing. Variance reduction in stochastic gradient Langevin dynamics. In *Advances in Neural Information Processing Systems*, pages 1154–1162, 2016.

Paolo Giordani, Tor Jacobson, Erik von Schedvin, and Mattias Villani. Taking the twists into account: Predicting firm bankruptcy risk with splines of financial ratios. *Journal of Financial and Quantitative Analysis*, 49(4):1071–1099, 2014.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

David Gunawan, Robert Kohn, Matias Quiroz, Khue-Dung Dang, and Minh-Ngoc Tran. Subsampling sequential Monte Carlo for static Bayesian models. *arXiv preprint arXiv:1805.03317*, 2018.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Matthew D Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

Pierre E Jacob and Alexandre H Thiery. On nonnegative unbiased estimators. *The Annals of Statistics*, 43(2):769–784, 2015.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 31st International Conference on Machine Learning*, pages 181–189, 2014.

Fredrik Lindsten and Arnaud Doucet. Pseudo-marginal Hamiltonian Monte Carlo. *arXiv preprint arXiv:1607.02516*, 2016.

Anne-Marie Lyne, Mark Girolami, Yves Atchade, Heiko Strathmann, and Daniel Simpson. On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical Science*, 30(4):443–467, 2015.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

Dougal Maclaurin and Ryan P Adams. Firefly Monte Carlo: Exact MCMC with subsets of data. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI 2014).*, 2014.

Florian Maire, Nial Friel, and Pierre Alquier. Informed sub-sampling MCMC: approximate Bayesian inference for large datasets. *Statistics and Computing*, To appear, 2018.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David Dunson. Scalable and robust Bayesian inference via the median posterior. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1656–1664, 2014.

Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):113–162, 2011.

Willie Neiswanger, Chong Wang, and Eric Xing. Asymptotically exact, embarrassingly parallel MCMC. *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI 2014)*, pages 623–632, 2014.

Christopher Nemeth and Chris Sherlock. Merging MCMC subposteriors through Gaussian-process approximations. *Bayesian Analysis*, 13(2):507–530, 2018.

Christopher Nemeth, Chris Sherlock, and Paul Fearnhead. Particle Metropolis-adjusted Langevin algorithms. *Biometrika*, 103(3):701–717, 2016.

Geoff K Nicholls, Colin Fox, and Alexis Muir Watt. Coupled MCMC with a randomized acceptance probability. *arXiv preprint arXiv:1205.6857*, 2012.

Akihiko Nishimura, David Dunson, and Jianfeng Lu. Discontinuous Hamiltonian Monte Carlo for sampling discrete parameters. *arXiv preprint arXiv:1705.08510*, 2017.

Michael K Pitt, Ralph dos Santos Silva, Paolo Giordani, and Robert Kohn. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151, 2012.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

Matias Quiroz, Minh-Ngoc Tran, Mattias Villani, and Robert Kohn. Speeding up MCMC by delayed acceptance and data subsampling. *Journal of Computational and Graphical Statistics*, 27(1):12–22, 2018a.

Matias Quiroz, Minh-Ngoc Tran, Mattias Villani, Robert Kohn, and Khue-Dung Dang. The block-Poisson estimator for optimally tuned exact subsampling MCMC. *arXiv preprint arXiv:1603.08232v5*, 2018b.

Matias Quiroz, Mattias Villani, Robert Kohn, Minh-Ngoc Tran, and Khue-Dung Dang. Subsampling MCMC - A introduction for the survey statistician. *Sankhya A*, 80(1): 33–69, 2018c.

Matias Quiroz, Robert Kohn, Mattias Villani, and Minh-Ngoc Tran. Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, 114(526): 831–843, 2019.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

Gareth O Roberts and Jeffrey S Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.

Gareth O Roberts and Osnat Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and Computing in Applied Probability*, 4(4):337–357, 2002.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009.

Sebastian M Schmon, George Deligiannidis, Arnaud Doucet, and Michael K Pitt. Large sample asymptotics of the pseudo-marginal method. *arXiv preprint arXiv:1806.10060*, 2018.

Steven L Scott, Alexander W Blocker, Fernando V Bonassi, H Chipman, E George, and R McCulloch. Bayes and big data: the consensus Monte Carlo algorithm. In *EFaBBayes 250" conference*, volume 16, 2013.

Xiaocheng Shang, Zhanxing Zhu, Benedict Leimkuhler, and Amos J Storkey. Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling. In *Advances in Neural Information Processing Systems*, pages 37–45, 2015.

Chris Sherlock, Alexandre H Thiery, Gareth O Roberts, and Jeffrey S Rosenthal. On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics*, 43(1):238–275, 2015.

Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer. Consistency and fluctuations for stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 17: 1–33, 2016.

M-N Tran, R Kohn, M Quiroz, and M Villani. The block pseudo-marginal sampler. *arXiv preprint arXiv:1603.02485v5*, 2017.

Sebastian J. Vollmer, Konstantinos C. Zygalakis, and Yee Whye Teh. Exploration of the (non-)asymptotic bias and variance of stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 17(159):1–48, 2016.

Xiangyu Wang and David B Dunson. Parallel MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605v2*, 2014.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688, 2011.

Journal of Machine Learning Research 20 (2019) 1-43 Submitted 9/17; Revised 3/19; Published 6/19

Low Permutation-rank Matrices: Structural Properties and Noisy Completion

Nihar B. Shah NIHARS@CS.CMU.EDU
Machine Learning Department and Computer Science Department
Carnegie Mellon University

Sivaraman Balakrishnan SIVA@STAT.CMU.EDU
Department of Statistics
Carnegie Mellon University

Martin J. Wainwright WAINWRIG@STAT.BERKELEY.EDU
EECS Department and Department of Statistics
University of California, Berkeley

Editor: Tong Zhang

Abstract

We consider the problem of noisy matrix completion, in which the goal is to reconstruct a structured matrix whose entries are partially observed in noise. Standard approaches to this underdetermined inverse problem are based on assuming that the underlying matrix has low rank, or is well-approximated by a low rank matrix. In this paper, we propose a richer model based on what we term the "permutation-rank" of a matrix. We first describe how the classical non-negative rank model enforces restrictions that may be undesirable in practice, and how and these restrictions can be avoided by using the richer permutation-rank model. Second, we establish the minimax rates of estimation under the new permutation-based model, and prove that surprisingly, the minimax rates are equivalent up to logarithmic factors to those for estimation under the typical low rank model. Third, we analyze a computationally efficient singular-value-thresholding algorithm, known to be optimal for the low-rank setting, and show that it also simultaneously yields a consistent estimator for the low-permutation rank setting. Finally, we present various structural results characterizing the uniqueness of the permutation-rank decomposition, and characterizing convex approximations of the permutation-rank polytope.

Keywords: Non-negative matrix completion, recommender systems, permutation-based model, minimax theory, oracle inequalities.

1. Introduction

In the problem of matrix completion, the goal is to reconstruct a matrix based on observations of a subset of its entries (Laurent, 2001). Matrix completion has a variety of applications, including recommender systems (Koren et al., 2009), image understanding (Lee and Seung, 1999), credit risk monitoring (Vandendorpe et al., 2008), fluorescence spectroscopy (Gobinet et al., 2004), and modeling signal-adaptive audio effects (Sarver and Klapuri, 2011). We refer the reader to the surveys by Gillis (2014); Davenport and Romberg (2016) for an overview of the vast literature on this topic. Throughout this paper, in order to provide a running example for our modeling, it will be convenient to refer back to

©2019 Shah, Balakrishnan, Wainwright.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-545.html.

a particular variant of a recommender system application. More concretely, suppose that there are $n \geq 2$ users and $d \geq 2$ items, as well as an unknown matrix $M^* \in [0,1]^{n \times d}$ that captures the users' preferences for the items. Specifically, the $(i,j)^{th}$ entry of M^* represents the probability that user i likes item j. The problem is to estimate this preference matrix $M^* \in [0,1]^{n \times d}$ from observing users' likes or dislikes for some subset of the items.

Following a long line of past work in this area (e.g., Chen et al., 2013; Gross, 2011; Srebro et al., 2005; Candes and Recht, 2009; Candès and Tao, 2010; Keshavan et al., 2010; Recht, 2011; Chatterjee, 2014), we consider the following form of random design observation model. For a given parameter $p_{\text{obs}} \in (0,1]$ and for any user-item pair (i,j), we observe user i's rating for item j with probability p_{obs}. We assume that when an entry (i,j) is observed, we observe a binary value—for instance, {like, dislike} or $\{0,1\}$—which arises as a Bernoulli realization of the true preference M^*_{ij}.[1] More formally, we observe a matrix $Y \in \{0, \frac{1}{2}, 1\}^{n \times d}$, where

$$Y_{ij} = \begin{cases} 1 & \text{with probability } p_{\text{obs}} M^*_{ij} & \text{(user } i \text{ likes item } j) \\ 0 & \text{with probability } p_{\text{obs}}(1 - M^*_{ij}) & \text{(user } i \text{ dislikes item } j) \\ \frac{1}{2} & \text{with probability } 1 - p_{\text{obs}} & \text{(no data available),} \end{cases} \tag{1}$$

for every $(i,j) \in [n] \times [d]$. The goal is to estimate the underlying matrix M^* based on the observed matrix Y.

It is clear that, if no structural conditions are imposed on the underlying matrix M^*, then this problem is ill-posed. A classical approach is to impose on a bound on either the rank or the non-negative rank of the matrix. We begin by describing the approach based on the non-negative rank, before turning to the alternative approach based on permutation rank that is the focus of this paper.

Non-negative rank: In the problem of non-negative low-rank matrix completion, the matrix M^* is assumed to have a factorization of the form

$$M^* = UV^T,$$

for some matrices $U \in \mathbb{R}_+^{n \times r}$ and $V \in \mathbb{R}_+^{d \times r}$. Here the integer $r \in \{1, \ldots, \min\{d,n\}\}$ is known as the *non-negative rank* of the matrix. (As a corner case, we also have that the zero matrix is the only matrix with a non-negative rank of $r = 0$.) It is often assumed that the non-negative rank r is a known quantity, but in this paper, we make no such assumptions. For any value of $r \in \{1, \ldots, \min\{d,n\}\}$, we let $\mathbb{C}_{\text{NR}}(r)$ denote the set of all matrices with a non-negative factorization of rank at most r—that is

$$\mathbb{C}_{\text{NR}}(r) := \left\{ M \in [0,1]^{n \times d} \mid M = UV^T,\ U \in \mathbb{R}_+^{n \times r},\ V \in \mathbb{R}_+^{d \times r} \right\}.$$

For any matrix M, the smallest value of r such that $M \in \mathbb{C}_{\text{NR}}(r)$ is termed its *non-negative rank*, and is denoted by $\overline{r}(M)$.

1. Our results readily extend to any rating scheme with bounded values, such as five-star ratings. We focus on the binary case for purposes of brevity.

In order to gain some intuition for the meaning of the non-negative rank, note that any matrix $M \in \mathbb{C}_{\text{NR}}(r)$ can be written as a sum of the form

$$M = \sum_{\ell=1}^{r} u^\ell (v^\ell)^T.$$

Here $u^\ell \in \mathbb{R}_+^n$ and $v^\ell \in \mathbb{R}_+^d$ are vectors such that $u^\ell (v^\ell)^T \in [0,1]^{n \times d}$ for every $\ell \in [r]$. Such a decomposition can be interpreted as the existence of r features, indexed by $\ell \in [r]$. The d entries of vector v^ℓ represent the contribution of feature ℓ to the d respective items, and the n entries of vector u^ℓ represent the amounts by which the n respective users are influenced by feature ℓ. The popular overview article by Koren et al. (2009) provides an explanation for this assumption:

> *"Latent factor models are an alternative approach that tries to explain the ratings by characterizing both items and users on, say, 20 to 100 factors inferred from the ratings patterns. ... For movies, the discovered factors might measure obvious dimensions such as comedy versus drama, amount of action, or orientation to children; less well-defined dimensions such as depth of character development or quirkiness; or completely uninterpretable dimensions. For users, each factor measures how much the user likes movies that score high on the corresponding movie factor."*

Concerns with non-negative rank: Let us now delve deeper into this model[2], continuing in the context of movie recommendations for the sake of concreteness. Suppose there are r features that govern the movie watching experience; examples of such features include the amount of comedy content or the depth of character development. For any user $i \in [n]$ and any feature $\ell \in [r]$, we let $u_i^\ell \in \mathbb{R}_+$ denote the "affinity" of user i towards feature ℓ, and for any movie $j \in [d]$, we let $v_j^\ell \in \mathbb{R}_+$ denote the amount of content associated to feature ℓ in movie j. The conventional low non-negative rank model then assumes that the affinity of user i towards movie j conditioned on feature ℓ is given by $u_i^\ell v_j^\ell$. Consequently, for given feature ℓ, the entire behavior of each user and movie is governed by a pair of parameters, namely u_i^ℓ and v_j^ℓ for user i and item j respectively. Such an assumption has some unnatural implications. For instance, consider any two movies, say A and B, and any two users, say X and Y. Then conditioned on any feature ℓ, we have the implication

$$\frac{\text{Preference of user } X \text{ for movie } A}{\text{Preference of user } X \text{ for movie } B} = \frac{\text{Preference of user } Y \text{ for movie } A}{\text{Preference of user } Y \text{ for movie } B}.$$

In words, the low non-negative rank model inherently imposes a condition that is potentially unrealistic—namely, that for any given feature, the ratio of preferences for any pair of movies is identical for all users. Likewise, for any given feature, the ratio of preferences of any pair of users is identical for all movies. With the goal of circumventing this possibly troublesome condition, let us now describe a generalization that we call permutation rank, which is the main focus of our paper.

2. A slightly different, alternative interpretation is discussed in Appendix A.

Permutation rank: As with the ordinary rank, the permutation rank of the all-zeros matrix is zero. Otherwise, for any non-zero matrix, the permutation rank ρ takes values in the set $\{1, \ldots, \min\{n,d\}\}$. We begin by describing the set $\mathbb{C}_{\text{PR}}(1)$ of matrices with permutation rank one:

$$\begin{aligned}\mathbb{C}_{\text{PR}}(1) := \{ & M \in [0,1]^{n \times d} \mid \exists \text{ permutations } \pi_1 : [n] \to [n] \text{ and } \pi_2 : [d] \to [d] \text{ such that} \\ & M_{ij} \geq M_{i'j'} \text{ for every quadruple } (i,j,i',j') \text{ such that } \pi_1(i) \geq \pi_1(i') \text{ and } \pi_2(j) \geq \pi_2(j') \ \}.\end{aligned}$$

In words, a non-zero matrix is said to have a permutation rank of 1 if there exists a permutation of its rows and columns such that the entries of the resulting matrix are non-decreasing down any column and to the right along any row. Observe that any matrix with the conventional (non-negative) rank equal to 1 also belongs to the set $\mathbb{C}_{\text{PR}}(1)$. However, a matrix in $\mathbb{C}_{\text{PR}}(1)$ can have any non-negative rank, meaning the set of matrices with a permutation-rank of 1 also includes some matrices with a full non-negative rank.

We now extend the definition of the permutation rank to any integer $\rho \in \{1, \ldots, \min\{n,d\}\}$. In particular, the set of matrices with permutation rank at most ρ is given by

$$\mathbb{C}_{\text{PR}}(\rho) := \left\{ M \in [0,1]^{n \times d} \;\middle|\; M = \textstyle\sum_{\ell=1}^{\rho} Q^\ell \text{ for some matrices } Q^1, \ldots, Q^\rho \in \mathbb{C}_{\text{PR}}(1) \right\},$$

Note that this definition reduces to $\mathbb{C}_{\text{PR}}(1)$ in the special case $\rho = 1$. Otherwise, for $\rho > 1$, the permutations defining membership of each constituent matrix Q^ℓ in $\mathbb{C}_{\text{PR}}(\rho)$ are allowed to be different. For any matrix M, the smallest value of ρ such that $M \in \mathbb{C}_{\text{PR}}(\rho)$ is termed its *permutation rank*, and is denoted by $\overline{\rho}(M)$.

Revisiting the example of movie recommendations, the interpretation of this more general permutation-rank model is that conditioned on any feature $\ell \in [r]$, the preference ordering across movies continues to be consistent for different users, but the values of these preferences *need not* be identical scalings of each other. Observe that the conventional non-negative matrix-completion setting $\mathbb{C}_{\text{NR}}(r)$ is a special case of the permutation-rank matrix-completion setting where each matrix Q^ℓ is restricted to be of rank one. Whenever $r < \min\{d,n\}$, we have the strict inclusion $\mathbb{C}_{\text{NR}}(r) \subset \mathbb{C}_{\text{PR}}(r)$.

Outline and main contributions: Having discussed the limitations of the non-negative rank model and introduced the permutation rank, we now outline the remainder of the paper. In Section 2, we present our main results on the problem of estimating the matrix M^* (in the Frobenius norm) from partial and noisy observations. Specifically, we present a certain regularized least squares estimator, and prove that it achieves (nearly) minimax-optimal rates for estimation over the permutation-rank model. We also show that surprisingly, even if one considers the more restrictive non-negative rank model, and even if the rank is known, no estimator can achieve lower estimation error up to logarithmic factors. We also analyze the computationally efficient Singular Value Thresholding (SVT) algorithm, and show that it yields consistent estimates over the permutation-rank model, in addition to yielding the optimal estimate under the non-negative rank model. In Section 3, we establish some interesting properties of the permutation-rank model, and also derive certain relationships of this model with the non-negative rank model. In Section 4 we present the proofs of our results. We conclude the paper with a discussion in Section 5.

The paper also contains two appendices. Appendix A describes an alternative interpretation of the non-negative rank model. Appendix B is devoted to negative results, where we show that certain intuitive algorithms provably fail.

2. Main results on estimating M^*

We begin by considering the problem of estimating a low permutation rank matrix M^* based on noisy and partial observations. We first analyze a computationally expensive estimator, based on regularizing the least-squares cost with a multiple of the permutation rank, and show that it achieves minimax-optimal rates up to logarithmic factors. We then turn to a polynomial-time algorithm based on nuclear norm regularization, which is equivalent singular value thresholding in the current set-up.

2.1. Optimal oracle inequalities for estimation

Suppose that we collect an observation matrix Y of the form (1), where the unknown matrix M^* belongs $\mathbb{C}_{\rm PR}$. In this section, we analyze a regularized form of least-squares estimation, as applied to the recentered matrix

$$Y' := \frac{1}{p_{\rm obs}} Y - \frac{1-p_{\rm obs}}{2p_{\rm obs}} 11^T. \tag{2a}$$

We perform this recentering in order to obtain an unbiased estimate Y' of the true matrix M^* in the presence of missing observations, which is used in the least-squares estimator described below. As a sanity check, observe that when $p_{\rm obs} = 1$, we have the direct relation $Y' = Y$.

Letting $\overline{\rho}(M)$ denote the permutation-rank of any matrix M, we then consider the estimator

$$\widehat{M}_{\rm LS} \in \underset{M \in [0,1]^{n\times d}}{\arg\min} \Big(\|Y' - M\|_{\rm F}^2 + \frac{\overline{\rho}(M) \max\{n,d\} \log^{2.01} d}{p_{\rm obs}} \Big). \tag{2b}$$

Observe that importantly, the estimator $\widehat{M}_{\rm LS}$ does *not* need to know the value of the true permutation-rank of the underlying matrix. Moreover, while the estimator (as stated) is based on a known value of $p_{\rm obs}$, this assumption is not critical, since $p_{\rm obs}$ can be estimated accurately from the observed matrix Y.

We now turn to some theoretical guarantees on the performance of this estimator. Rather than assuming that, for a given rank ρ, the target matrix M^* has permutation rank exactly equal to ρ, we instead provide bounds that depend on distances to the set of all matrices with a given permutation rank. More precisely, for any given tolerance $\epsilon \geq 0$, define the set

$$B^{\rm P}(\rho, \epsilon) := \big\{ M \in [0,1]^{n\times d} \mid \exists M' \in [0,1]^{n\times d} \text{ s.t. } \overline{\rho}(M') \leq \rho \text{ and } \|M - M'\|_{\rm F} \leq \epsilon \big\},$$

corresponding to the set of all matrices that are at most ϵ distant from the set of matrices with permutation rank ρ. Similarly, we define the set

$$B^{\rm N}(r, \epsilon) := \big\{ M \in [0,1]^{n\times d} \mid \exists M' \in [0,1]^{n\times d} \text{ s.t. } \overline{r}(M') \leq r \text{ and } \|M - M'\|_{\rm F} \leq \epsilon \big\},$$

corresponding to matrices that are at most ϵ away from some matrix with non-negative-rank r.

In stating the following theorem, as well as throughout the remainder of the paper, we use c, c', c_1 etc. to denote positive universal constants. The values of these constants may differ from line to line.

Theorem 1 *(a) For any matrix $M^* \in [0,1]^{n\times d}$ and any integer $\rho \in [\min\{n,d\}]$, the regularized least squares estimator $\widehat{M}_{LS}$ satisfies the upper bound*

$$\frac{1}{nd}\|\widehat{M}_{LS} - M^*\|_F^2 \leq c_l \min\Big\{1, \quad \min_{M \in \mathbb{C}_{PR}(\rho)} \frac{\|M - M^*\|_F^2}{nd} + \frac{\rho \ \log^{2.01}(nd)}{\min\{n,d\} p_{\rm obs}}\Big\}, \tag{3a}$$

with probability at least $1 - e^{-c_0 \max\{n,d\}\log(nd)}$.
(b) Conversely, for any integer $\rho \in [\min\{n,d\}]$, any scalar $\epsilon \geq 0$, and any estimator $\widehat{M}$, there exists a matrix $M^* \in B^N(\rho, \epsilon)$ such that

$$\mathbb{E}\big[\frac{1}{nd}\|\widehat{M} - M^*\|_F^2\big] \geq c_2 \min\Big\{1, \quad \frac{\epsilon^2}{nd} + \frac{\rho}{\min\{n,d\} p_{\rm obs}}\Big\}. \tag{3b}$$

See Section 4.1 for the proof of this claim.

Interpretation as oracle inequality: The upper bound (3a) is an instance of an *oracle inequality*: it provides a family of upper bounds, one for each choice of the integer $\rho \in [\min\{n,d\}]$, on the estimation error associated with an arbitrary matrix $M^* \in [0,1]^{n\times d}$. For each choice of ρ, the upper bound (3a) consists of two terms. The first term, involving the minimum over $M \in \mathbb{C}_{\rm PR}(\rho)$, is a form of approximation error: it measures how well the unknown matrix M^* can be approximated with a matrix M of permutation rank at most ρ. The second term is a form of estimation error, measuring the difficulty of estimating a matrix that has permutation rank at most ρ. Since one such an upper bound holds for each choice of ρ, the bound (3a) shows that the estimator mimicks the behavior of an "oracle", which is allowed to choose ρ so as to optimize the trade-off between the approximation and estimation error.

Sandwiching of the risk: The upper bound (3a) of Theorem 1 can equivalently be stated in the following manner. For any integer $\rho \in [\max\{n,d\}]$ and any scalar $\epsilon \geq 0$ such that $M^* \in B^{\rm P}(\rho, \epsilon)$, the regularized least squares estimator $\widehat{M}_{\rm LS}$ satisfies the upper bound

$$\frac{1}{nd}\|\widehat{M}_{\rm LS} - M^*\|_{\rm F}^2 \leq c_1 \min\Big\{\frac{\epsilon^2}{nd} + \frac{\rho \ \log^{2.01}(nd)}{\min\{n,d\} p_{\rm obs}}, 1\Big\}, \tag{4a}$$

with probability at least $1 - e^{-c_0 \max\{n,d\}\log(\max\{nd\})}$. On the other hand, since $B^{\rm N}(\rho,\epsilon) \subseteq B^{\rm P}(\rho,\epsilon)$ for every value of ρ and ϵ, the lower bound (3b) of Theorem 1 implies the following result. For any integer $\rho \in [\min\{n,d\}]$, any scalar $\epsilon \geq 0$, and any estimator $\widehat{M}$, there exists a matrix $M^* \in B^{\rm N}(\rho,\epsilon)$ such that

$$\mathbb{E}\big[\frac{1}{nd}\|\widehat{M} - M^*\|_{\rm F}^2\big] \geq c_2 \min\Big\{1, \quad \frac{\epsilon^2}{nd} + \frac{\rho}{\min\{n,d\} p_{\rm obs}}\Big\}. \tag{4b}$$

Comparing the bounds (4a) and (4b), we see that our results are sharp up to logarithmic factors.

Specialization to minimax risk: When suitably specialized to matrices that have some fixed permutation (or non-negative) rank, Theorem 1 leads to sharp upper and lower bounds on the minimax risks for the problems of matrix completion over the sets $\mathbb{C}_{\text{NR}}$ and $\mathbb{C}_{\text{PR}}$. In order for a clear comparison between the two bounds, let us index both the non-negative rank and the permutation-rank using a generic notation k—the meaning of the notation will be clear from the context.

Part (a) of Theorem 1 implies that for any value $k \in [\min\{n, d\}]$ and any matrix $M^* \in \mathbb{C}_{\text{PR}}(k)$, the regularized least squares estimator $\widehat{M}_{\text{LS}}$ satisfies the bound

$$\frac{1}{dn}\|\widehat{M}_{\text{LS}} - M^*\|_{\text{F}}^2 \leq c_1 \min\Big\{\frac{k\ \log^{2.01}(nd)}{\min\{n,d\}p_{\text{obs}}}, 1\Big\},$$

with probability at least $1 - e^{-c_0 \max\{n,d\}\log(nd)}$. Within the set of matrices $[0,1]^{n\times d}$ under consideration, we have the deterministic upper bound $\frac{1}{nd}\|\widehat{M}_{\text{LS}} - M^*\|_{\text{F}}^2 \leq 1$. Consequently, our high probability upper bound also implies a uniform bound on the mean-squared error over the set $\mathbb{C}_{\text{PR}}(k)$—that is

$$\sup_{M^*\in\mathbb{C}_{\text{PR}}(k)} \frac{1}{dn}\mathbb{E}[\|\widehat{M}_{\text{LS}} - M^*\|_{\text{F}}^2] \leq c_1' \min\Big\{\frac{k\ \log^{2.01}(nd)}{\min\{n,d\}p_{\text{obs}}}, 1\Big\}. \tag{5a}$$

Since $\mathbb{C}_{\text{PR}}(k)$ is a superset of $\mathbb{C}_{\text{NR}}(k)$, the same upper bound holds for the minimax risk over $\mathbb{C}_{\text{NR}}(k)$:

$$\sup_{M^*\in\mathbb{C}_{\text{NR}}(k)} \frac{1}{dn}\mathbb{E}[\|\widehat{M}_{\text{LS}} - M^*\|_{\text{F}}^2] \leq c_1' \min\Big\{\frac{\rho\ \log^{2.01}(nd)}{\min\{n,d\}p_{\text{obs}}}, 1\Big\}. \tag{5b}$$

Conversely, part (b) of Theorem 1 implies that for any $k \in [\max\{n, d\}]$, the error incurred by any estimator $\widehat{M}$ over the set $\mathbb{C}_{\text{NR}}(k)$ is error lower bounded as

$$\sup_{M^*\in\mathbb{C}_{\text{NR}}(k)} \frac{1}{dn}\mathbb{E}[\|\widehat{M} - M^*\|_{\text{F}}^2] \geq c_2 \min\Big\{\frac{k}{\min\{n,d\}p_{\text{obs}}}, 1\Big\}. \tag{5c}$$

Since $\mathbb{C}_{\text{NR}}(k) \subseteq \mathbb{C}_{\text{PR}}(k)$, the error incurred by any estimator $\widehat{M}$ over the set $\mathbb{C}_{\text{PR}}(k)$ is also lower bounded as

$$\sup_{M^*\in\mathbb{C}_{\text{PR}}(k)} \frac{1}{dn}\mathbb{E}[\|\widehat{M} - M^*\|_{\text{F}}^2] \geq c_2 \min\Big\{\frac{k}{\min\{n,d\}p_{\text{obs}}}, 1\Big\}. \tag{5d}$$

We have thus characterized the minimax risk over both the families $\mathbb{C}_{\text{PR}}$ or $\mathbb{C}_{\text{NR}}$, with bounds (5) that are matching up to logarithmic factors.

A win-win for permutation-based models: An important consequence of our oracle and minimax results is the multi-fold benefit of moving from the restrictive non-negative-rank assumptions to the more general permutation-rank assumptions. Fitting a permutation-rank k model when the true matrix actually has a non-negative rank of k leads to relatively little additional (overfitting) error. On the other hand, we show later in the paper that fitting a non-negative rank k model when the true matrix actually has a permutation-rank of k can lead to very high error, due to model mismatch.

Resolving an open problem in estimation from pairwise comparisons: The aforementioned results resolve an important open problem in the area of estimation from pairwise comparisons. The pairwise-comparison setting involves a number of items (such as a number of cars or football teams), and the observed data comprises noisy comparisons between various pairs of these items. The classical literature in this area assumes that the probability of any item beating another in a pairwise comparison between them follows a restrictive parameter-based model. Recently, the paper Shah et al. (2017) established minimax rates of estimating these pairwise comparison probabilities under a strictly and significantly more general model called strong stochastic transitivity (SST). Understanding minimax rates when the pairwise comparison probabilities follow mixtures of SST models remained open.

The pairwise-comparison setting under the SST model is a special case of our present problem and corresponds to the setting where the value of ρ is known and equal to 1, the matrix M^* is square with $n = d$, and all entries of M^* satisfy the shifted-skew-symmetry condition $M^*_{ij} + M^*_{ji} = 1$. Theorem 1 provides guarantees on the estimation of mixtures of different SST models, thereby resolving an important open problem on this topic. In addition, the results of Theorem 1 are obtained in the framework of oracle inequalities, which is more general than the minimax framework of Shah et al. (2017), and sheds light on situations when the true matrix does not follow the assumed model.

2.2. Computationally efficient estimator

At this point, we do not know how to compute the regularized least squares estimator (2b) in an efficient manner, and we suspect that it may be computationally intractable to do so. Consequently, in this section, we turn to analyzing a different method based on singular value thresholding (SVT). Singular value thresholding has been used either directly or as a subroutine in several past papers on the conventional low-rank matrix completion problem (see, for example, the papers Cai et al., 2010; Donoho et al., 2014; Chatterjee, 2014). This approach is appealing due to its computational simplicity, involving only computation of the singular value decomposition, followed by a single pointwise non-linearity; see Cai and Osher (2010) for a fast algorithm. In the context of the permutation rank completion problem, we show here that the SVT estimator is consistent for estimation under the permutation-rank model, albeit with a rate that is suboptimal by a factor of $\sqrt{\min\{n,d\}p_{\text{obs}}}$. Note that these guarantees hold without the estimator needing to know that the matrix is drawn from a permutation-rank model, nor the value p_{obs} of the permutation rank.

The SVT estimator is straightforward to describe. From the observation matrix $Y \in \{0, \frac{1}{2}, 1\}^{n\times d}$, we first obtain the transformed observation matrix Y' as in equation (2a). Applying the singular value decomposition yields the representation $Y' = UDV^T$, where the $(n \times d)$ matrix D is diagonal, whereas the $(n \times n)$ and $(d \times d)$ matrices U and V respectively are both orthonormal. For a threshold $\lambda > 0$ to be specified, define another diagonal matrix $\widetilde{D}_\lambda$ with entries

$$[\widetilde{D}_\lambda]_{jj} = \begin{cases} 0 & \text{if } D_{jj} < \lambda \\ D_{jj} - \lambda & \text{if } D_{jj} \geq \lambda \end{cases} \qquad \text{for each } j \in [\min\{n,d\}]. \tag{6}$$

Finally, the SVT estimator is given by

$$\widehat{M}_{\text{SVT}} = U\widetilde{D}_\lambda V^T.$$

The following theorem now establishes guarantees for the singular value thresholding estimator.

Theorem 2 *Suppose that* $p_{\text{obs}} \geq \frac{1}{\min\{n,d\}}\log^7(nd)$. *Then for any matrix* $M^* \in [0,1]^{n\times d}$, *the SVT estimator* $\widehat{M}_{SVT}$ *with threshold* $\lambda = 2.1\sqrt{\frac{n+d}{p_{\text{obs}}}}$ *satisfies the bound*

$$\frac{1}{nd}\|\widehat{M}_{SVT} - M^*\|_F^2 \leq c_l \min_{M\in[0,1]^{n\times d}}\Big(\min\Big\{\frac{\overline{\rho}(M)}{\sqrt{\min\{n,d\}p_{\text{obs}}}}, \frac{\overline{r}(M)}{\min\{n,d\}p_{\text{obs}}}\Big\} + \frac{1}{nd}\|M^* - M\|_F^2\Big), \tag{7}$$

with probability at least $1 - e^{-c_0\max\{n,d\}}$.

Observe that the bound (7) on the risk of the SVT estimator has the term $\sqrt{\min\{n,d\}}$ in the denominator of the first expression in the minimum, as opposed to the $\min\{n,d\}$ in the upper bound (3a) from Theorem 1. This form of "$\sqrt{n}$-suboptimality" arises in several permutation-based problems of this type studied in recent papers (e.g., Shah et al., 2017, 2019; Chatterjee and Mukherjee, 2019; Shah et al., 2016; Flammarion et al., 2016; Pananjady et al., 2016). In some cases, this gap—between the performance of any polynomial-time algorithm and the best algorithm—is known to be unavoidable (Shah et al., 2019) conditional on the planted clique conjecture. It is interesting to speculate whether such a computational complexity gap exists in the context of the permutation-rank model.

The proof techniques underlying Theorem 2 can also be used to establish previously known guarantees (Koltchinskii et al., 2011; Chatterjee, 2014) for the non-negative rank model. In order to contrast with the permutation-rank model, let us state one such guarantee here. It is known from previous results (Koltchinskii et al., 2011; Chatterjee, 2014) for the non-negative rank model that for any matrix $M^* \in [0,1]^{n\times d}$, the SVT estimator incurs an error upper bounded by

$$\frac{1}{nd}\|\widehat{M}_{\text{SVT}} - M^*\|_{\text{F}}^2 \leq c_1' \frac{\overline{r}(M^*)}{\min\{n,d\}p_{\text{obs}}}, \tag{8}$$

with high probability. On the other hand, our permutation-based modeling approach yields a stronger guarantee for the classical SVT estimator—namely, setting $M = M^*$ in our upper bound (8) of Theorem 2 guarantees that the SVT estimator $\widehat{M}_{\text{SVT}}$ with threshold $\lambda = 2.1\sqrt{\frac{n+d}{p_{\text{obs}}}}$ satisfies the upper bound

$$\frac{1}{nd}\|\widehat{M}_{\text{SVT}} - M^*\|_{\text{F}}^2 \leq c_1 \min\Big\{\frac{\overline{r}(M^*)}{\min\{n,d\}p_{\text{obs}}}, \frac{\overline{\rho}(M^*)}{\sqrt{\min\{n,d\}p_{\text{obs}}}}\Big\}, \tag{9}$$

with high probability. The bound (9) can yield results that are much sharper as compared to what may be obtained from the previously known guarantees (8) for the SVT estimator.

For example, suppose $n = d$ and $p_{\text{obs}} = 1$. Consider the matrix $M^* \in [0,1]^{n\times d}$ given by

$$M^*_{ij} = \begin{cases} 1 & \text{if } i > j \\ 0 & \text{if } i < j \\ \frac{1}{2} & \text{if } i = j. \end{cases}$$

Then we have $\overline{r}(M^*) = n$ and $\overline{\rho}(M^*) = 1$. Consequently, the bound (8) from past literature yields an upper bound of

$$\frac{1}{nd}\|\widehat{M}_{\text{SVT}} - M^*\|_{\text{F}}^2 \leq c_1',$$

whereas in contrast, our analysis (9) yields the sharper bound

$$\frac{1}{nd}\|\widehat{M}_{\text{SVT}} - M^*\|_{\text{F}}^2 \leq \frac{c_1}{\sqrt{n}}, \tag{10}$$

with high probability. Moreover, in our earlier work (Shah et al., 2017), we have shown that for this choice of M^*, the bound (10) is the best possible up to logarithmic factors for the SVT estimator with any fixed threshold λ.

3. Properties of permutation-rank models

In the previous section, we established some motivating properties of permutation-based models from the perspective of statistical estimation, in this section, we derive some more insights on the permutation-rank model.

3.1. Comparing permutation-rank and non-negative-rank

We begin by comparing the permutation-rank model with the conventional non-negative rank model. To this end, first observe that the definitions of the two models immediately imply that the permutation-rank of any matrix is always upper bounded by its non-negative rank, that is, for any matrix M, we have $\overline{\rho}(M) \leq \overline{r}(M)$. A natural question that now arises is whether there is any additional general condition beyond this simple relation that constrains the two notions of the matrix rank. The following proposition shows that there is no other guaranteed relation between the two notions of matrix rank.

Proposition 3 *For any values* $0 < \rho \leq r \leq \min\{n,d\}$, *there exist matrices whose permutation-rank is* ρ *and non-negative rank is* r.

A particular instance that underlies part of the proof of Proposition 3, associated to any pair of values (ρ, r), is the following block matrix $M_{\rho,r}$ of size $(n \times d)$:

$$M_{\rho,r} := \begin{bmatrix} J_{r-\rho+1} & 0 & 0 \\ 0 & I_{\rho-1} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where for any value k, J_k denotes an upper triangular matrix of size $(k \times k)$ with all entries on and above the diagonal set as 1, and let I_k denote the identity matrix of size $(k \times k)$.

By construction, the matrix $M_{\rho,r}$ has a non-negative rank $\overline{r}(M_{\rho,r}) = r$ and a permutation rank $\overline{\rho}(M_{\rho,r}) = \rho$.

We now investigate a second relation between the two models. Recall from our discussion earlier that the assumptions of the permutation-rank model are much less restrictive than the assumptions of the non-negative rank model. With this context, a natural question that arises is to quantify the bias of an estimator that fits a matrix of non-negative rank of k when the true underlying matrix instead has a permutation rank of k. We answer this question using the notion of the Hausdorff distance: For any two sets $\mathbb{S}_1, \mathbb{S}_2 \in \mathbb{R}^{n\times d}$, the Hausdorff distance $\mathcal{H}(\mathbb{S}_1, \mathbb{S}_2)$ between the two sets in the squared Frobenius norm is defined as

$$\mathcal{H}(\mathbb{S}_1, \mathbb{S}_2) := \max\Big\{ \sup_{M\in\mathbb{S}_1} \inf_{M'\in\mathbb{S}_2} \|M - M'\|_{\mathrm{F}}^2 \,,\; \sup_{M'\in\mathbb{S}_2} \inf_{M\in\mathbb{S}_1} \|M - M'\|_{\mathrm{F}}^2 \Big\} \tag{11}$$

The following proposition quantifies the Hausdorff distance between non-negative-rank and permutation-rank models.

Proposition 4 *For any positive integer $k \le \frac{1}{2}\min\{d, n\}$, the Hausdorff distance between the sets $\mathbb{C}_{NR}(k)$ and $\mathbb{C}_{PR}(k)$ is lower bounded as*

$$\mathcal{H}(\mathbb{C}_{NR}(k), \mathbb{C}_{PR}(k)) \ge c_3 \frac{nd}{k}. \tag{12}$$

Proposition 4 helps quantify the bias on fitting a non-negative rank model when the true matrix follows the permutation-rank model as follows. Consider any positive integer $k \le \frac{1}{2}\min\{d, n\}$, and any estimator $\widetilde{M}_k$ that outputs a matrix in $\mathbb{C}_{\mathrm{NR}}(k)$. Then since $\mathbb{C}_{\mathrm{NR}}(k) \subseteq \rho(k)$, the error incurred by this estimator when the true matrix lies in the set $\mathbb{C}_{\mathrm{PR}}(k)$ is lower bounded as

$$\sup_{M^*\in\mathbb{C}_{\mathrm{PR}}(k)} \frac{1}{nd}\|M - \widetilde{M}_k\|_{\mathrm{F}}^2 \ge \frac{1}{nd}\mathcal{H}(\mathbb{C}_{\mathrm{NR}}(k), \mathbb{C}_{\mathrm{PR}}(k)) \ge c_3 \frac{1}{k}, \tag{13}$$

with probability 1.

Observe that when k is a constant (but n and d are allowed to grow), the right hand side of the bound (12) becomes a constant, and this is the largest possible order-wise gap between any pair of matrices in $[0, 1]^{n\times d}$. Likewise, the right hand side of (13) becomes a constant, and this is the largest possible order-wise error for any estimator that outputs matrices in $[0, 1]^{n\times d}$.

3.2. No "good" convex approximation

In this section, we investigate a question about an important property of the permutation-based set, and in particular, its primitive $\mathbb{C}_{\mathrm{PR}}(1)$. There are various estimators including our regularized least squares estimator (2b) as well as those studied in the literature (Shah et al., 2017, 2019, 2016) which require solving a an optimization problem over the set $\mathbb{C}_{\mathrm{PR}}(1)$. With this goal in mind, a natural question that arises is: Is the set $\mathbb{C}_{\mathrm{PR}}(1)$ convex? If not, then does it at least have a "good" convex approximation? The following proposition answers these questions in the negative using the notion of the Hausdorff distance between sets (11).

Proposition 5 *The Hausdorff distance* (11) *between the set of matrices with permutation-rank one and any arbitrary convex set $\mathbb{C} \subseteq \mathbb{R}^{n\times n}$ is lower bounded as*

$$\frac{1}{nd}\mathcal{H}(\mathbb{C}_{PR}(1), \mathbb{C}) \ge c,$$

where $c > 0$ is a universal constant.

A specific example of a convex set $\mathbb{C}$ is the convex hull of $\mathbb{C}_{\mathrm{PR}}(1)$. Then by definition we have the relation $\sup_{M_1\in\mathbb{C}_{\mathrm{PR}}(1)} \inf_{M_2\in\mathbb{C}} \|M_1 - M_2\|_{\mathrm{F}}^2 = 0$. Consequently, Proposition 5 implies that

$$\sup_{M_2\in\mathbb{C}} \inf_{M_1\in\mathbb{C}_{\mathrm{PR}}(1)} \|M_1 - M_2\|_{\mathrm{F}}^2 = \Theta(nd),$$

thus showing that the convex hull of $\mathbb{C}_{\mathrm{PR}}(1)$ is a much larger set than $\mathbb{C}_{\mathrm{PR}}(1)$ itself.

The proof of Proposition 5 relies on a more general result that we derive, one which relates a certain notion of inherent (lack of) convexity of a set to the Hausdorff distance between that set and any convex approximation. Note that this result does not preclude the possibility that an optimization procedure over a convex approximation to $\mathbb{C}_{\mathrm{PR}}(1)$ converges close enough to some element of $\mathbb{C}_{\mathrm{PR}}(1)$ itself. We leave the investigation of this possibility to future work.

3.3. On the uniqueness of decomposition

In this section, we investigate conditions for the uniqueness of the decomposition of any matrix into its constituent components that have a permutation-rank of one. In the conventional setting of low non-negative rank matrix completion, several past works (Donoho and Stodden, 2003; Theis et al., 2005; Laurberg et al., 2008; Gillis, 2012; Arora et al., 2012) investigate the conditions required for uniqueness of the decomposition of matrices into their constituent non-negative rank-one matrices. Here we consider an analogous question in the setting of permutation rank. More precisely, consider any matrix $M \in [0, 1]^{n\times d}$ with a permutation-rank decomposition of the form

$$M = \sum_{\ell=1}^{\overline{\rho}(M)} M^{(\ell)}, \tag{14}$$

where $M^{(\ell)} \in \mathbb{C}_{\mathrm{PR}}(1)$ for every $\ell \in [\overline{\rho}(M)]$. Under what conditions on the matrix M is the set $\{M^{(1)}, \ldots, M^{(\overline{\rho}(M))}\}$ of constituent matrices unique? The following result provides a necessary condition for uniqueness. In order to state the result, we use the notation $\mathbf{1}$ to denote the indicator function, that is, $\mathbf{1}\{x\} = 1$ if x is true and $\mathbf{1}\{x\} = 0$ if x is false.

Proposition 6 *A necessary condition for the uniqueness of a permutation-rank decomposition* (14) *for any matrix M is that for every coordinate $(i, j) \in [n] \times [d]$, there is at most one $\ell \in [\overline{\rho}(M)]$ such that $M_{ij}^{(\ell)}$ is non-zero and distinct from all other entries of $M^{(\ell)}$, that is,*

$$\sum_{\ell\in[\overline{\rho}(M)]} \mathbf{1}\Big\{ M_{ij}^{\ell} \notin \{0\} \cup \{M_{i'j'}^{\ell}\}_{\substack{i'\in[n], j'\in[d],\\ (i',j')\neq(i,j)}} \Big\} \le 1 \qquad \textit{for every } (i, j) \in [n] \times [d].$$

We note that the necessary condition continues to hold even if we restrict attention to only symmetric matrices. The necessary condition provided by Proposition 6 indicates that *any* sufficient condition for uniqueness of the decomposition must be quite strong. Moreover, we believe that the conditions for sufficiency may be significantly stronger than those necessitated by Proposition 6. The reason for such drastic requirements for uniqueness is the high-degree of flexibility offered by the permutation-rank model.

Let us illustrate the necessary condition from Proposition 6 with a simple example. Consider the following matrix M with $n = d = 2$ and $\overline{\rho}(M) = 2$ and decomposition into $M^{(1)}, M^{(2)} \in \mathbb{C}_{\text{PR}}(1)$:

$$M := \begin{bmatrix} 1 & .6 \\ .6 & 1 \end{bmatrix} = \begin{bmatrix} 0 & .3 \\ .3 & .9 \end{bmatrix} + \begin{bmatrix} 1 & .3 \\ .3 & .1 \end{bmatrix}$$

Observe that the necessary condition obtained in Proposition 6 is required to hold for every coordinate of the matrix. Let us first evaluate this condition for the coordinate $(1,1)$ of the matrices. Since $M^{(1)}_{11} = 0$, there is at most one $\ell \in \{1,2\}$ such that $M^{(\ell)}_{11}$ is non-zero, and hence the coordinate $(1,1)$ satisfies the necessary condition. Moving on to coordinate $(1,2)$, we have $M^{(1)}_{12} = M^{(1)}_{21}$ and $M^{(2)}_{12} = M^{(2)}_{21}$; hence the coordinate $(1,2)$ also passes the necessary condition. The argument for coordinate $(1,2)$ also applies to coordinate $(2,1)$ since the matrices involved are symmetric. We finally test coordinate $(2,2)$. Observe that $M^{(1)}_{22} \notin \{0, M^{(1)}_{11}, M^{(1)}_{12}, M^{(1)}_{21}\}$ and $M^{(2)}_{22} \notin \{0, M^{(2)}_{11}, M^{(2)}_{12}, M^{(2)}_{21}\}$. As a consequence, for both $\ell = 1$ and $\ell = 2$, we have that $M^{(\ell)}_{22}$ is non-zero and distinct from all other entries of $M^{(\ell)}$. The condition necessary for uniqueness is thus violated. Indeed, as guaranteed by Proposition 6, there exist other decompositions of M with permutation-rank 2—for instance, the decomposition

$$M = \begin{bmatrix} 1 & .6 \\ .6 & 1 \end{bmatrix} = \begin{bmatrix} 0 & .4 \\ .4 & .9 \end{bmatrix} + \begin{bmatrix} 1 & .2 \\ .2 & .1 \end{bmatrix}$$

is another example.

Finally, we put the negative result on the decomposition into some practical perspective with an analogy to tensor decompositions. The canonical polyadic (CP) decomposition of a tensor (Hitchcock, 1927) is not unique unless strong non-degeneracy conditions are imposed (Kruskal, 1977). From a theoretical perspective in many applications (for instance, in estimating latent variable models (Hsu and Kakade, 2013)) the CP decomposition is most useful or interpretable when it is unique. Furthermore, even when the decomposition is unique, computing it is NP-hard in the worst-case (Håstad, 1990). However, in practice the CP decomposition is often computed via ad-hoc methods that generate useful results (Kolda and Bader, 2009).

4. Proofs

We now turn to the proofs of our main results. In all our proofs, we assume that the values of n and d are larger than certain universal constants, so as to avoid subcases having to do with small values of (n, d). We will also ignore floors and ceilings wherever they are not critical. These assumptions entail no ultimate loss of generality, since our results continue

to hold for all values with different constant prefactors. Throughout these and other proofs, we use the notation $\{c, c', c_0, c_1, C, C'\}$ and so on to denote positive constants whose values may change from line to line.

4.1. Proof of Theorem 1(a)

The proof of this theorem involves generalizing an argument used in our past work (Shah et al., 2017, Theorem 1). In particular, the problem setting of our past work (Shah et al., 2017, Theorem 1) is a special case of the present problem, restricted to the case of square matrices ($n = d$) and permutation rank $\rho = 1$. Here we develop a number of additional techniques in order to handle the generalization to non-square matrices and arbitrary permutation rank.

Throughout the proof, we assume without loss of generality that $n \leq d$; otherwise, we can apply the same argument to the matrix transposes. Now observe that we must have $\frac{1}{nd}\|\widehat{M}_{\text{LS}} - M^*\|_{\text{F}}^2 \leq 1$ deterministically, and hence it suffices to show that

$$\frac{1}{nd}\|\widehat{M}_{\text{LS}} - M^*\|_{\text{F}}^2 \leq c_1 \frac{1}{nd} \min_{M \in [0,1]^{n \times d}} \Big(\|M - M^*\|_{\text{F}}^2 + \frac{\overline{\rho}(M) d \log^{2.01}(nd)}{p_{\text{obs}}} \Big), \tag{15}$$

with high probability.

In order to succinctly capture the right hand side of (15), we define a matrix M_0 as

$$M_0 \in \operatorname*{arg\,min}_{M \in [0,1]^{n \times d}} \Big(\|M - M^*\|_{\text{F}}^2 + \frac{\overline{\rho}(M) d \log^{2.01} d}{p_{\text{obs}}} \Big).$$

One can alternatively interpret the matrix M_0 as the estimate that $\widehat{M}_{\text{LS}}$ would produce if the observation matrix Y' was replaced by the true matrix M^* in the optimization problem specified in (2b).

A key challenge in this analysis is to handle the complex structure of the permutation rank set. Specifically, any matrix in the set $\mathbb{C}_{\text{PR}}(k)$, for some $k \in [n]$, is defined to be the sum of k matrices each in the set $\mathbb{C}_{\text{PR}}(1)$. Each of these k constituent matrices can have *different and arbitrary* permutations of their rows/columns. In order to handle this complexity, we decompose this set into "simpler" sets of matrices, where each set of matrices is associated to a fixed set of permutations. We introduce some additional notation to this end. For any pair of permutations $\pi : [n] \to [n]$ and $\sigma : [d] \to [d]$, we first define the set

$$\mathbb{C}_{\text{PR}}(1; \pi, \sigma) := \{M \in \mathbb{C}_{\text{PR}}(1) \mid \text{rows and columns of } M \text{ are ordered according to } \pi \text{ and } \sigma \text{ respectively}\}.$$

In words, $\mathbb{C}_{\text{PR}}(1; \pi, \sigma)$ is the set of all matrices with permutation rank 1 and where the rows are ordered according to the permutation π and the columns are ordered according to the permutation σ.

Next we extend the notation $\mathbb{C}_{\text{PR}}(1; \pi, \sigma)$ to matrices with a more general permutation rank. Let Π denote the set of all possible permutations of d items, and let Σ denote the set of all possible permutations of the n users. Consider any value $k \in [n]$, any sequence

$\widetilde{\Pi}^{(k)} := (\pi_1, \ldots, \pi_k) \in \Pi^k$ and any sequence $\widetilde{\Sigma}^{(k)} := (\sigma_1, \ldots, \sigma_k) \in \Sigma^k$. Define the set

$$\mathbb{C}_{\mathrm{PR}}(k; \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}) := \Big\{ M = \sum_{\ell=1}^{k} M^{(\ell)} \Big| \; M^{(\ell)} \in \mathbb{C}_{\mathrm{PR}}(1; \pi_\ell, \sigma_\ell) \text{ for every } \ell \in [k] \Big\}.$$

The set $\mathbb{C}_{\mathrm{PR}}(k; \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})$ is the set of all matrices with permutation rank k, and which can be expressed as the constituent permutation-rank-1 matrices following permutations $(\pi_1, \sigma_1), \ldots, (\pi_k, \sigma_k)$ respectively.

Having decomposed the permutation-rank set, we now apply a similar treatment to the estimator $\widehat{M}_{\mathrm{LS}}$. The estimator $\widehat{M}_{\mathrm{LS}}$ as defined in (2b) minimizes a certain objective over all matrices in $[0,1]^{n \times d}$. We will decompose this estimator in a certain manner in terms of multiple estimators, each of which optimize over a particular subset of $[0,1]^{n \times d}$. Specifically, for each set $\mathbb{C}_{\mathrm{PR}}(k; \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})$ defined above, let us define an associated estimator

$$M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} \in \underset{M \in \mathbb{C}_{\mathrm{PR}}(k; \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})}{\arg\min} \|Y' - M\|_{\mathrm{F}}^2. \tag{16}$$

Note that in defining the objective of minimization (16) for $M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}$, we have omitted the regularization term $\frac{\overline{\rho}(M) d \log^{2.01} d}{p_{\mathrm{obs}}}$ from (2b) because the value of the rank $\overline{\rho}(M)$ is the same for all matrices in the set $\mathbb{C}_{\mathrm{PR}}(k; \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})$. Observe that the regularized least squares estimate $\widehat{M}_{\mathrm{LS}}$ can now be thought of as choosing the matrix $M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}$ across all possible values of $(k; \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})$ that minimizes the regularized least squares objective in (2b).

Finally, we define a set $\widetilde{\Gamma}$ as

$$\widetilde{\Gamma} := \Big\{ (k, \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}) \in [n] \times \Pi^k \times \Sigma^k \Big| \|Y' - M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}\|_{\mathrm{F}}^2 + \frac{k d \log^{2.01} d}{p_{\mathrm{obs}}} \leq \|Y' - M_0\|_{\mathrm{F}}^2 + \frac{\overline{\rho}(M_0) d \log^{2.01} d}{p_{\mathrm{obs}}} \Big\}.$$

The set $\widetilde{\Gamma}$ is the set of all tuples of k and sets of permutations $(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})$ such that the value of the objective in (2b) under the estimator $M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}$ is no more than that obtained if one sets the estimate to M_0. The proof will use these quantities in the following manner. First, the matrix M_0 can intuitively be thought of as a "reasonable" regularized least squares estimate (although impractical since computing M_0 requires knowledge of M^*). Treating the objective value in the optimization problem (2b) attained by M_0 as a baseline, the proof can primarily focus on only those estimates (16) whose objective value in (2b) is at most that of M_0. The set of tuples $(k, \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})$ satisfying this condition precisely comprises the set $\widetilde{\Gamma}$.

Note that the set $\widetilde{\Gamma}$ is guaranteed to be non-empty since the parameter and permutations corresponding to M_0 always lie in $\widetilde{\Gamma}$. We now state a lemma which forms a key component of this proof.

Lemma 7 *For any $(k, \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}) \in \widetilde{\Gamma}$, it must be that*

$$\mathbb{P}\Big(\|M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} - M_0\|_F^2 \leq c_l \frac{\overline{\rho}(M_0) d \log^{2.01} d}{p_{\mathrm{obs}}} + 4\|M_0 - M^*\|_F^2 \Big) \geq 1 - e^{-4kd \log d}, \tag{17}$$

for some positive universal constant c_l.

Under our assumption of $d \geq n$, for any value of k the cardinality of the set $\widetilde{\Gamma}$ restricted to any k is at most $e^{2kd \log d}$. Hence a union bound over all $k \in [n]$ and all permutations—applied to equation (17)—yields

$$\mathbb{P}\Big(\max_{(k, \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}) \in \widetilde{\Gamma}} \|M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} - M_0\|_{\mathrm{F}}^2 \leq c_l \frac{\overline{\rho}(M_0) d \log^{2.01} d}{p_{\mathrm{obs}}} + 4\|M_0 - M^*\|_{\mathrm{F}}^2 \Big) \geq 1 - e^{-d \log d}. \tag{18}$$

From the definition of the regularized least squares estimator $\widehat{M}_{\mathrm{LS}}$ in equation (2b) and the definition of set $\widetilde{\Gamma}$ above, we have that $\widehat{M}_{\mathrm{LS}}$ must equal $M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}$ for some $(k, \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}) \in \widetilde{\Gamma}$. As a consequence, the tail bound (18) ensures that

$$\mathbb{P}\Big(\|\widehat{M}_{\mathrm{LS}} - M_0\|_{\mathrm{F}}^2 \leq c_l \frac{\overline{\rho}(M_0) d \log^{2.01} d}{p_{\mathrm{obs}}} + 4\|M_0 - M^*\|_{\mathrm{F}}^2 \Big) \geq 1 - e^{-d \log d}.$$

Finally, applying the triangle inequality yields the claimed result

$$\mathbb{P}\Big(\|\widehat{M}_{\mathrm{LS}} - M^*\|_{\mathrm{F}}^2 \leq 10\|M^* - M_0\|_{\mathrm{F}}^2 + 2c_l \frac{\overline{\rho}(M_0) d \log^{2.01} d}{p_{\mathrm{obs}}} \Big) \geq 1 - e^{-d \log d}.$$

Proof of Lemma 7 We begin by writing an equivalent form of the (random) observation and noise matrices that will aid in our analysis. It is straightforward to verify that the observation matrix Y' can equivalently be written in the linearized form

$$Y' = M^* + \frac{1}{p_{\mathrm{obs}}} W', \tag{19a}$$

where W' has entries that are independent, and are distributed as

$$[W']_{ij} = \begin{cases} p_{\mathrm{obs}}(\frac{1}{2} - [M^*]_{ij}) + \frac{1}{2} & \text{with probability } p_{\mathrm{obs}}[M^*]_{ij} \\ p_{\mathrm{obs}}(\frac{1}{2} - [M^*]_{ij}) - \frac{1}{2} & \text{with probability } p_{\mathrm{obs}}(1 - [M^*]_{ij}) \\ p_{\mathrm{obs}}(\frac{1}{2} - [M^*]_{ij}) & \text{with probability } 1 - p_{\mathrm{obs}}. \end{cases} \tag{19b}$$

By definition, any $(k, \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}) \in \widetilde{\Gamma}$ must satisfy the inequality

$$\|Y' - M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}\|_{\mathrm{F}}^2 + \frac{k d \log^{2.01} d}{p_{\mathrm{obs}}} \leq \|Y' - M_0\|_{\mathrm{F}}^2 + \frac{\overline{\rho}(M_0) d \log^{2.01} d}{p_{\mathrm{obs}}}.$$

Using the linearized form (19a), some algebraic manipulations yield the inequality

$$\|M^* - M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}\|_{\mathrm{F}}^2 + \frac{2}{p_{\mathrm{obs}}} \langle\langle W', \; M^* - M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} \rangle\rangle + \frac{k d \log^{2.01} d}{p_{\mathrm{obs}}} \leq \|M^* - M_0\|_{\mathrm{F}}^2 + \frac{2}{p_{\mathrm{obs}}} \langle\langle W', \; M^* - M_0 \rangle\rangle + \frac{\overline{\rho}(M_0) d \log^{2.01} d}{p_{\mathrm{obs}}}. \tag{20}$$

Applying the inequality $\|M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} - M_0\|_{\mathrm{F}}^2 \leq 2\|M^* - M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}\|_{\mathrm{F}}^2 + 2\|M^* - M_0\|_{\mathrm{F}}^2$ and denoting $\widehat{\Delta}_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} := (M_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} - M_0)$, some algebraic manipulations yield the inequality

$$\frac{1}{2}\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}}\|_{\mathrm{F}}^2 \leq \frac{2}{p_{\mathrm{obs}}} \langle\langle W', \; \widehat{\Delta}_{\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}} \rangle\rangle + \frac{(\overline{\rho}(M_0) - k) d \log^{2.01} d}{p_{\mathrm{obs}}} + 2\|M_0 - M^*\|_{\mathrm{F}}^2. \tag{21}$$

Now consider the set of matrices

$$\mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0) := \Big\{\alpha(M - M_0) \mid M \in \mathbb{C}_{\text{PR}}(k; \widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}),\ \alpha \in [0,1]\Big\}, \tag{22}$$

and note that $\mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0) \subseteq [-1,1]^{n\times d}$. For each choice of radius $t > 0$, define the random variable

$$Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(t) := \sup_{\substack{M_{\text{DIFF}} \in \mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}; M_0),\\ \|M_{\text{DIFF}}\|_{\text{F}} \le t}} \frac{2}{p_{\text{obs}}} \langle\!\langle W',\ M_{\text{DIFF}} \rangle\!\rangle. \tag{23}$$

Using the inequality (21), the Frobenius norm error $\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}}$ then satisfies the bound

$$\frac{1}{2}\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}}^2 \le\ Z_{\widetilde{\Pi},\widetilde{\Sigma}}\big(\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}}\big) + \frac{(\overline{\rho}(M_0) - k) d \log^{2.01} d}{p_{\text{obs}}} + 2\|M_0 - M^*\|_{\text{F}}^2. \tag{24}$$

Thus, in order to obtain our desired bound, we need to understand the behavior of the random quantity $Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(t)$.

Dy definition, the set $\mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0)$ is "star-shaped", meaning that $\alpha M_{\text{DIFF}} \in \mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)})$ for every $\alpha \in [0,1]$ and every $M_{\text{DIFF}} \in \mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0)$. Using this star-shaped property, we are guaranteed that $\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)]$ grows at most linearly with δ. We are then in turn guaranteed the existence of some scalar $\delta_c > 0$ satisfying the critical inequality

$$\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c)] \le \frac{\delta_c^2}{2}. \tag{25}$$

Our interest is in an upper bound to the smallest (strictly) positive solution δ_c to the critical inequality (25), and moreover, our goal is to show that for every $t \ge \delta_c$, we have $\|\widehat{\Delta}\|_{\text{F}} \le c\sqrt{t\delta_c}$ with high probability. To this end, define a "bad" event $\mathcal{A}_t$ as

$$\mathcal{A}_t = \big\{\exists \Delta \in \mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0) \mid \|\Delta\|_{\text{F}} \ge \sqrt{t\delta_c} \quad \text{and} \quad \frac{2}{p_{\text{obs}}}\langle\!\langle W',\ \Delta \rangle\!\rangle \ge 2\|\Delta\|_{\text{F}}\sqrt{t\delta_c}\big\}. \tag{26}$$

Using the star-shaped property of $\mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0)$, it follows by a rescaling argument that

$$\mathbb{P}[\mathcal{A}_t] \le \mathbb{P}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c) \ge 2\delta_c\sqrt{t\delta_c}] \qquad \text{for all } t \ge \delta_c.$$

The following lemma helps control the behavior of the random variable $Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c)$.

Lemma 8 *For any $\delta > 0$, the mean of $Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)$ is bounded as*

$$\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)] \le c_1 \frac{\max\{k, \overline{\rho}(M_0)\} d}{p_{\text{obs}}} \log^2 d,$$

and for every $u > 0$, its tail probability is bounded as

$$\mathbb{P}\Big(Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta) > \mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)] + u\Big) \le \exp\Big(\frac{-c_2 u^2 p_{\text{obs}}}{\delta^2 + \mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)] + u}\Big),$$

where c_1 and c_2 are positive universal constants.

From this lemma, we have the tail bound

$$\mathbb{P}\Big(Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c) > \mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c)] + \delta_c\sqrt{t\delta_c}\Big) \le \exp\Big(\frac{-c_2(\delta_c\sqrt{t\delta_c})^2 p_{\text{obs}}}{\delta_c^2 + \mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c)] + (\delta_c\sqrt{t\delta_c})}\Big),$$

for all $t > 0$. By the definition of δ_c in (25), we have $\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c)] \le \delta_c^2 \le \delta_c\sqrt{t\delta_c}$ for all $t \ge \delta_c$, and consequently

$$\mathbb{P}[\mathcal{A}_t] \le \mathbb{P}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta_c) \ge 2\delta_c\sqrt{t\delta_c}] \ \le\ \exp\Big(\frac{-c_2(\delta_c\sqrt{t\delta_c})^2 p_{\text{obs}}}{3\delta_c\sqrt{t\delta_c}}\Big), \quad \text{for all } t \ge \delta_c. \tag{27}$$

Now we must have either $\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}} \le \sqrt{t\delta_c}$, or we have $\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}} > \sqrt{t\delta_c}$. In the latter case, conditioning on the complement $\mathcal{A}_t^c$, our basic inequality (21) implies that

$$\frac{1}{2}\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}}^2 \le 2\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}}\sqrt{t\delta_c} + \frac{(\overline{\rho}(M_0) - k) d \log^{2.01} d}{p_{\text{obs}}} + 2\|M_0 - M^*\|_{\text{F}}^2.$$

Using the identity $x^2 \le ax + b \Rightarrow x^2 \le 4a^2 + 2b$ for every non-negative triplet of scalars x, a and b, we obtain

$$\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}}^2 \le 16t\delta_c + 2\frac{(\overline{\rho}(M_0) - k) d \log^{2.01} d}{p_{\text{obs}}} + 4\|M_0 - M^*\|_{\text{F}}^2, \tag{28}$$

with probability at least $\mathbb{P}(\mathcal{A}_t^c) \ge 1 - \exp\Big(\frac{-c_2(\delta_c\sqrt{t\delta_c})^2 p_{\text{obs}}}{3\delta_c\sqrt{t\delta_c}}\Big)$ for all $t \ge \delta_c$ (from (27)). Finally, from the bound on the expected value of $Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(t)$ in Lemma 8, we see that the critical inequality (25) is satisfied for

$$\delta_c = \sqrt{\frac{c_1 \max\{\overline{\rho}(M_0), k\} d}{p_{\text{obs}}}} \log d.$$

Setting $t = c'\delta_c$ in (28) for a large enough constant c' yields

$$\mathbb{P}\Big(\|\widehat{\Delta}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}\|_{\text{F}}^2 \le \frac{c_1'\overline{\rho}(M_0) d}{p_{\text{obs}}} \log^{2.01} d + 4\|M_0 - M^*\|_{\text{F}}^2\Big) \ge 1 - \exp\Big(-4\max\{\overline{\rho}(M_0), k\} d \log d\Big),$$

for some constant $c_1' > 0$, thus yielding the claim of the lemma.

Proof of Lemma 8 We break our proof into two parts, corresponding to bounds on the mean of the random variable $Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)$ followed by control of its tail behavior.

<u>Bounding the mean</u>: We begin by establishing an upper bound on the mean $\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)]$. In order to obtain the desired upper bound, in the proof we carefully account for the rank k and the different permutations associated to any matrix in $\mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0)$.

For convenience of analysis, we introduce a new random variable

$$\widetilde{Z}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}} := \sup_{M_{\text{DIFF}} \in \mathbb{C}_{\text{DIFF}}(\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}; M_0)} \langle\!\langle W',\ M_{\text{DIFF}} \rangle\!\rangle.$$

Then by definition, we have $\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)] \leq \frac{1}{p_{\rm obs}}\mathbb{E}[\widetilde{Z}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}]$ for every $\delta > 0$. In addition, since $M_0 \in \mathbb{C}_{\rm PR}(k)$, it can be decomposed as $M_0 = \sum_{\ell=1}^{k} M_0^{(\ell)}$, for some matrices $M_0^{(1)}, \ldots, M_0^{(k)} \in \mathbb{C}_{\rm PR}(1)$.

We introduce some additional notation for ease of exposition. If $\overline{\rho}(M_0) < k$, then let $M_0^{(\overline{\rho}(M_0)+1)}, \ldots, M_0^{k}$ denote all-zero matrices. Hence we can write $M_0 = \sum_{\ell=1}^{\max\{\overline{\rho}(M_0),k\}} M_0^{(\ell)}$. If $\overline{\rho}(M_0) > k$ then let $\pi_{k+1}, \ldots, \pi_{\overline{\rho}(M_0)}$ be arbitrary (but fixed) permutations of n items and $\sigma_{k+1}, \ldots, \sigma_{\overline{\rho}(M_0)}$ be arbitrary (but fixed) permutations of d items. With this notation in place, we have the following deterministic upper bound on the value of the random variable $\widetilde{Z}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}$:

$$\widetilde{Z}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}} \leq \sum_{\ell=1}^{\max\{\overline{\rho}(M_0),k\}} \sup_{[M_{\rm DIFF}]_\ell \in \mathbb{C}_{\rm DIFF}(\{\pi_\ell\},\{\sigma_\ell\};M_0^{(\ell)})} \langle\!\langle W', \, [M_{\rm DIFF}]_\ell \rangle\!\rangle.$$

We also recall our assumption that $d \geq n$ without loss of generality. Now let $\log N(\epsilon, \mathbb{C}, \|\cdot\|_{\rm F})$ denote the ϵ metric entropy of class $\mathbb{C} \subset \mathbb{R}^{n\times d}$ in the Frobenius norm metric $\|\cdot\|_{\rm F}$. Then the truncated form of Dudley's entropy integral inequality yields[3]

$$\mathbb{E}[\widetilde{Z}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}] \leq \sum_{\ell=1}^{\max\{\overline{\rho}(M_0),k\}} c\left\{d^{-8} + \int_{\frac{1}{2}d^{-9}}^{2d} \sqrt{\log N(\epsilon, \mathbb{C}_{\rm DIFF}(\{\pi_\ell\},\{\sigma_\ell\};M_0^{(\ell)}), \|.\|_{\rm F})}(\Delta\epsilon)\right\}, \tag{29}$$

where we have used the fact that the diameter of the set $\mathbb{C}_{\rm DIFF}(\{\pi_\ell\},\{\sigma_\ell\};M_0^{(\ell)})$ is at most $2d$ in the Frobenius norm.

In our past work (Shah et al., 2017, Lemma 2), we derived a bound on the metric entropy of the set $\mathbb{C}_{\rm DIFF}(\{\pi_\ell\},\{\sigma_\ell\};M_0^{(\ell)})$ as:

$$\log N\Big(\epsilon, \mathbb{C}_{\rm DIFF}(\{\pi_\ell\},\{\sigma_\ell\};M_0^{(\ell)}), \|\cdot\|_{\rm F}\Big) \leq 16\frac{d^2}{\epsilon^2}\big(\log\frac{d}{\epsilon}\big)^2,$$

for any $\epsilon > 0$ and $\ell \in [k]$. Substituting this bound on the metric entropy into the Dudley bound (29) yields

$$\mathbb{E}[\widetilde{Z}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}] \leq c' \max\{\overline{\rho}(M_0), k\} d \log^2 d.$$

The inequality $\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)] \leq \frac{1}{p_{\rm obs}}\mathbb{E}[\widetilde{Z}_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}]$ then yields the claimed result.

Bounding the tail: In order to establish the claimed tail bound on the deviations of $Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)$ above its mean, we use a Bernstein-type bound on the supremum of empirical processes due to Klein and Rio (2005, Theorem 1.1c), which we state in a simplified form here.

Lemma 9 (Klein and Rio) *Let $X := (X_1, \ldots, X_m)$ be any sequence of zero-mean, independent random variables, each taking values in $[-1, 1]$. Let $\mathcal{V} \subset [-1, 1]^m$ be any measurable set of m-length vectors. Then for any $u > 0$, the supremum $X^\dagger = \sup_{v\in\mathcal{V}}\langle X, v\rangle$ satisfies the upper tail bound*

$$\mathbb{P}\big(X^\dagger > \mathbb{E}[X^\dagger] + u\big) \leq \exp\Big(\frac{-u^2}{2\sup_{v\in\mathcal{V}}\mathbb{E}[\langle v, X\rangle^2] + 4\mathbb{E}[X^\dagger] + 3u}\Big).$$

We now call upon Lemma 9 setting $\mathcal{V} = \{M_{\rm DIFF} \in \mathbb{C}_{\rm DIFF}(\widetilde{\Pi}^{(k)}, \widetilde{\Sigma}^{(k)}; M_0) \mid \|M_{\rm DIFF}\|_{\rm F} \leq \delta\}$, $X = W'$, and $X^\dagger = p_{\rm obs} Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)$. The entries of the matrix W' are mutually independent, have a mean of zero, and are bounded by 1 in absolute value. Then we have $\mathbb{E}[X^\dagger] = p_{\rm obs}\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)]$ and $\mathbb{E}[\langle\!\langle M_{\rm DIFF}, \, W'\rangle\!\rangle^2] \leq 4p_{\rm obs}\|M_{\rm DIFF}\|_{\rm F}^2 \leq 4p_{\rm obs}\delta^2$ for every $M_{\rm DIFF} \in \mathcal{V}$. With these assignments, and some algebraic manipulations, we obtain that for every $u > 0$,

$$\mathbb{P}\big(Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta) > \mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)] + u\big) \leq \exp\Big(\frac{-u^2 p_{\rm obs}}{8\delta^2 + 4\mathbb{E}[Z_{\widetilde{\Pi}^{(k)},\widetilde{\Sigma}^{(k)}}(\delta)] + 3u}\Big),$$

as claimed.

4.2. Proof of Theorem 1(b)

Assume without loss of generality that $d \geq n$. Throughout the proof, we ignore floor and ceiling conditions as these are not critical to the proof and affect the lower bound by only a constant factor.

Our proof is based on the framework of Fano's inequality which is a standard method in minimax analysis (see Cover and Thomas, 2012 or Tsybakov, 2008). Here we state the inequality in the context of our problem:

Lemma 10 (Fano's inequality) *Consider any integer $\eta \geq 2$ and any set of η matrices $G^1, \ldots, G^\eta \in B^N(r, \epsilon)$. For each $\ell \in [\eta]$, let $\mathbb{P}^\ell$ denote the probability distribution of the matrix Y obtained by setting $M^* = G^\ell$. Setting ξ_1 and ξ_2 as*

$$\xi_1 = \max_{\ell\neq\ell'\in[\eta]} D_{\rm KL}(\mathbb{P}^\ell \| \mathbb{P}^{\ell'}) \qquad \text{and} \qquad \xi_2 = \min_{\ell\neq\ell'\in[\eta]} \|G^\ell - G^{\ell'}\|_F^2,$$

the minimax risk of estimating M^* from Y is lower bounded as

$$\inf_{\widehat{M}} \sup_{M^*\in B^N(r,\epsilon)} \mathbb{E}\big[\|M^* - \widehat{M}\|_2^2\big] \geq \frac{\xi_2}{2}\Big(1 - \frac{\xi_1 + \log 2}{\log \eta}\Big). \tag{30}$$

In order to obtain the desired bound using Fano's inequality, we must construct a set of matrices with suitable values of η, ξ_1 and ξ_2. Our first step towards this goal is to call upon the Gilbert-Varshamov bound (Gilbert, 1952; Varshamov, 1957) from coding theory. Here we state this bound in the form of a special case that is required for our problem. As done throughout this manuscript, the notation c represents a positive universal constant.

Lemma 11 (Gilbert-Varshamov bound) *For any integer $\tau \geq 2$, there exists a set of at least $\exp(c\tau)$ binary vectors, each of length τ, such that the Hamming distance between any pair vectors in this set is at least $\tau/10$.*

3. Here we use $(\Delta\epsilon)$ to denote the differential of ϵ, so as to avoid confusion with the number of columns d.

We now use the framework of Fano and Gilbert-Varshamov to establish our claimed result.

Designing candidate set of matrices for Fano's inequality: In what follows, we use the Gilbert-Varshamov bound to construct a set of binary vectors with certain desired properties, and then use it to in turn obtain a set of matrices for use in Fano's inequality. Let $\tau = dr + p_{\text{obs}}\epsilon^2$. Then the Gilbert-Varshamov bound guarantees the existence of

$$\eta := \exp\Big(c(dr + p_{\text{obs}}\epsilon^2)\Big)$$

binary vectors $g^1, \ldots, g^\eta$, each of length $(dr + p_{\text{obs}}\epsilon^2)$, such that the Hamming distance between any pair of vectors in this set is lower bounded by $\frac{dr+p_{\text{obs}}\epsilon^2}{10}$.

For some $\delta \in (0, \frac{1}{4})$ whose value is specified later, define a related set of vectors $\widetilde{g}^1, \ldots, \widetilde{g}^\eta$ as

$$\widetilde{g}^\ell_j = \begin{cases} \frac{1}{2} + \delta & \text{if } g^\ell_j = 1 \\ \frac{1}{2} - \delta & \text{if } g^\ell_j = 0, \end{cases}$$

for every $\ell \in [\eta]$ and $j \in [dr + p_{\text{obs}}\epsilon^2]$. Next define a set of "low rank" matrices $G^1, \ldots, G^\eta \in [0,1]^{n\times d}$ where the matrix G^ℓ is obtained as follows. For each $\ell \in [\eta]$, arrange the first rd entries of vector $\widetilde{g}^\ell$ as the entries of an $(r \times d)$ matrix—this arrangement may be done in an arbitrary manner as long as it is consistent across every $\ell \in [\eta]$. Now append a $(\frac{p_{\text{obs}}\epsilon^2}{d} \times d)$ matrix at the bottom, whose entries comprise the last $p_{\text{obs}}\epsilon^2$ entries of the vector $\widetilde{g}^\ell$—again, this arrangement may be done in an arbitrary manner as long as it is consistent across every $\ell \in [\eta]$. Now stack $\frac{1}{p_{\text{obs}}}$ copies of the resulting $\big((r + \frac{p_{\text{obs}}\epsilon^2}{d}) \times d\big)$ matrix on top of each other to form a $\big((\frac{r}{p_{\text{obs}}} + \frac{\epsilon^2}{d}) \times d\big)$ matrix. Note that our assumption $\epsilon^2 + \frac{r\max\{n,d\}}{p_{\text{obs}}} \leq nd$, along with the assumption $d \geq n$, implies that $n \geq \frac{r}{p_{\text{obs}}} + \frac{\epsilon^2}{d}$. Append $(n - (\frac{r}{p_{\text{obs}}} + \frac{\epsilon^2}{d}))$ rows of all zeros at the bottom of this matrix, and denote the resultant $(n \times d)$ matrix as G^ℓ. The set of matrices $\{G^1, \ldots, G^\eta\}$ is our candidate set for use in Fano's inequality.

Applying Fano's inequality: In order to apply Fano's inequality, we first establish certain properties of the candidate set of matrices constructed above.

Lemma 12 *The set of matrices $\{G^1, \ldots, G^\eta\}$ satisfies the following three properties:*
(a) $G^\ell \in B^N(r, \epsilon)$ for every $\ell \in [\eta]$.
(b) For every pair $\ell_1 \neq \ell_2 \in [\eta]$, it must be that

$$\|G^{\ell_1} - G^{\ell_2}\|_F^2 \geq \frac{\delta^2}{10}\big(\frac{dr}{p_{\text{obs}}} + \epsilon^2\big).$$

(c) For every $\ell \in [\eta]$, let $\mathbb{P}^\ell$ denotes the probability distribution of the matrix Y obtained by setting $M^* = G^\ell$. Then for every pair $\ell_1 \neq \ell_2 \in [\eta]$, it must be that

$$D_{\text{KL}}(\mathbb{P}^\ell_1 \| \mathbb{P}^{\ell_2}) \leq c'\delta^2 p_{\text{obs}}\big(\frac{dr}{p_{\text{obs}}} + \epsilon^2\big).$$

Part (a) of Lemma 12 ensures that the condition of membership in $B^{\text{N}}(r, \epsilon)$ in the statement of Lemma 10 is satisfied. Then parts (b) and (c) of Lemma 12 allow us to set $\xi_1 = c'\delta^2 p_{\text{obs}}\big(\frac{dr}{p_{\text{obs}}} + \epsilon^2\big)$ and $\xi_2 = \frac{\delta^2}{10}\big(\frac{dr}{p_{\text{obs}}} + \epsilon^2\big)$ in Lemma 10. Substituting these relations in Lemma 10 yields a lower bound on any estimator $\widehat{M}$ for M^* as

$$\begin{aligned}\inf_{\widehat{M}} \sup_{M^* \in B^{\text{N}}(r,\epsilon)} \mathbb{E}[\|\widehat{M} - M^*\|_{\text{F}}^2] &\geq \frac{\delta^2}{20}\Big(\frac{dr}{p_{\text{obs}}} + \epsilon^2\Big)\Big(1 - \frac{c'\delta^2 p_{\text{obs}}\big(\frac{dr}{p_{\text{obs}}} + \epsilon^2\big) + \log 2}{c(dr + p_{\text{obs}}\epsilon^2)}\Big) \\ &\overset{(i)}{\geq} c''\big(\frac{dr}{p_{\text{obs}}} + \epsilon^2\big),\end{aligned}$$

where inequality (i) is obtained by choosing δ^2 as a small enough constant (which depends only on c and c'). Recalling our assumption $d \geq n$, and consequently replacing d by $\max\{n, d\}$ in the bound yields the claimed result.

Proof of Lemma 12 Part (a): We show that the matrix $G^\ell \in [0,1]^{n\times d}$ can be decomposed into a sum of a low-rank matrix (of non-negative rank at most r) and a sparse matrix (number of non-zero entries at most ϵ^2). To this end, we first show that $G^\ell \in B^{\text{N}}(r, \epsilon)$ for every $\ell \in [\eta]$, that is, we show that the matrix $G^\ell \in [0,1]^{n\times d}$ can be decomposed into a sum of a low-rank matrix (of non-negative rank at most r) and a sparse matrix (number of non-zero entries at most ϵ^2). First we set to zero the entries in G^ℓ which correspond to the last $p_{\text{obs}}\epsilon^2$ entries of the vector $\widetilde{g}^\ell$. Let us denote the resulting matrix as $\widetilde{G}^\ell$. Each row of the matrix $\widetilde{G}^\ell$ is either all zero or is identical to one among the first r rows of G^ℓ. Consequently we have $\overline{r}(\widetilde{G}^\ell) \leq r$. Also observe that in the matrix $(G^\ell - \widetilde{G}^\ell)$, the number of non-zero entries is at most $\frac{1}{p_{\text{obs}}} \times p_{\text{obs}}\epsilon^2 = \epsilon^2$, and furthermore, each of these entries lie in the interval $[0, 1]$. Hence we have $\|G^\ell - \widetilde{G}^\ell\|_{\text{F}}^2 \leq \epsilon^2$. The matrix G^ℓ thus satisfies all the requirements for membership in the set $B^{\text{N}}(r, \epsilon)$.

Part (b): The claim pertaining to the lower bound on the Frobenius norm difference is easily obtained via simple algebra:

$$\|G^{\ell_1} - G^{\ell_2}\|_{\text{F}}^2 = \frac{1}{p_{\text{obs}}}\|\widetilde{g}^{\ell_1} - \widetilde{g}^{\ell_2}\|_2^2 = \frac{4\delta^2}{p_{\text{obs}}} D_H(g^{\ell_1}, g^{\ell_2}),$$

where $D_H(g^{\ell_1}, g^{\ell_2})$ denotes the Hamming distance between the two binary vectors g^{ℓ_1} and g^{ℓ_2}. From our construction, the Hamming distance between any pair of vectors in the set $\{g^1, \ldots, g^\eta\}$ is at least $\frac{dr+p_{\text{obs}}\epsilon^2}{10}$, thereby yielding the claimed result.

Part (c): Since the entries of Y are mutually independent, we have the decomposition

$$D_{\text{KL}}(\mathbb{P}^{\ell_1} \| \mathbb{P}^{\ell_2}) = \sum_{i\in[n], j\in[d]} D_{\text{KL}}(\mathbb{P}^{\ell_1}_{ij} \| \mathbb{P}^{\ell_2}_{ij}),$$

where we have slightly abused notation to let $\mathbb{P}^\ell_{ij}$ denote the (marginal) probability distribution induced on Y_{ij} under $\mathbb{P}^\ell$. Now, the structure of the matrices $G^1, \ldots, G^\eta$ helps simplify

this bound to

$$D_{\mathrm{KL}}(\mathbb{P}^{\ell_1}\|\mathbb{P}^{\ell_2}) = \frac{1}{p_{\mathrm{obs}}} \sum_{i\in[r+\frac{p_{\mathrm{obs}}\epsilon^2}{d}],j\in[d]} D_{\mathrm{KL}}(\mathbb{P}^{\ell_1}_{ij}\|\mathbb{P}^{\ell_2}_{ij})$$
$$\overset{(i)}{\leq} \frac{1}{p_{\mathrm{obs}}}\Big(\frac{p_{\mathrm{obs}}\delta^2(dr+p_{\mathrm{obs}}\epsilon^2)}{(1/4-\delta^2)}\Big)$$
$$\overset{(ii)}{\leq} c'\delta^2 p_{\mathrm{obs}}\big(\frac{dr}{p_{\mathrm{obs}}}+\epsilon^2\big),$$

where inequality (i) follows from the elementary identity $a\log\frac{a}{b} \leq (a-b)\frac{a}{b}$ for all $a,b \in (0,1)$ along with some algebraic simplifications, and inequality (ii) is due to the restriction $\delta \in (0,\frac{1}{4})$. We thus obtain the claimed result.

4.3. Proof of Theorem 2

We now turn to analysis of the singular-value thresholding (SVT) estimator. This proof is based on the framework of a proof from our earlier work (Shah et al., 2017, Theorem 2), which can be seen as a particular case with $n = d$ and $\rho = 1$. We introduce certain additional tricks in order to generalize the proof for general values of ρ and to obtain a sharp dependence on ρ. As in our previous proofs, we may assume without loss of generality that $n \leq d$.

The high-level idea behind the proof is as follows. The goal is to bound the term $\|\widehat{M}_{\mathrm{SVT}} - M^*\|_{\mathrm{F}}^2$. Observe that $\widehat{M}_{\mathrm{SVT}}$ depends on the true matrix M^*, the realization of the noise Y and the threshold λ. The first step is to bound the target term in a manner that explicitly constitutes these components of $\widehat{M}_{\mathrm{SVT}}$. Such a decomposition will then allow for a simplified analysis where we can separately analyze the individual constituents. The terms corresponding to M^* are then bounded in an algebraic manner, and the noise term is bounded using tools from random matrix theory. These individual results are then put together by means of some careful algebra in order to obtain the claimed result.

Let us now present the complete proof. Recall from equation (19a) that we can write our observation model as $Y' = M^* + \frac{1}{p_{\mathrm{obs}}}W'$, where $W' \in [-1,1]^{n\times d}$ is a zero-mean matrix with mutually independent entries. Also recall that these entries follow the distribution

$$[W']_{ij} = \begin{cases} p_{\mathrm{obs}}(\frac{1}{2}-[M^*]_{ij})+\frac{1}{2} & \text{with probability } p_{\mathrm{obs}}[M^*]_{ij} \\ p_{\mathrm{obs}}(\frac{1}{2}-[M^*]_{ij})-\frac{1}{2} & \text{with probability } p_{\mathrm{obs}}(1-[M^*]_{ij}) \\ p_{\mathrm{obs}}(\frac{1}{2}-[M^*]_{ij}) & \text{with probability } 1-p_{\mathrm{obs}}. \end{cases} \tag{31}$$

For any matrix A, let $\sigma_1(A), \sigma_2(A), \ldots$ denote its singular values in descending order.

Our proof of the upper bound is based on three lemmas. The first lemma is a result from our earlier work (Shah et al., 2017). The key idea behind this lemma is to bound the term $\|\widehat{M}_{\mathrm{SVT}} - M^*\|_{\mathrm{F}}^2$ in a manner that explicitly captures the terms constituting $\widehat{M}_{\mathrm{SVT}}$.

Lemma 13 *(Shah et al., 2017, Lemma 3) If $\lambda \geq \frac{1.01\|W'\|_{op}}{p_{\mathrm{obs}}}$, then*

$$\|\widehat{M}_{SVT} - M^*\|_F^2 \leq c\sum_{j=1}^{n}\min\{\lambda^2, \sigma_j^2(M^*)\} \tag{32}$$

with probability at least $1 - c_1 e^{-c'n}$, where c, c_1 and c' are positive universal constants.

Our second lemma now helps to bound the right hand side of the inequality (32). The lemma presents an approximation-theoretic result that bounds the tail of the singular values of any matrix with a given permutation-rank or non-negative rank. The proof of this lemma builds on a construction due to Chatterjee (2014).

Lemma 14 *(a) For any matrix $M \in \mathbb{C}_{PR}(\rho)$ and any $s \in \{1,2,\ldots,n-1\}$, we have*

$$\sum_{j=s+1}^{n}\sigma_j^2(M) \leq \frac{nd\rho^2}{s}.$$

(b) For any matrix $M \in \mathbb{C}_{NR}(r)$ and any $s \in \{1,2,\ldots,n-1\}$, we have

$$\sum_{j=s+1}^{n}\sigma_j^2(M) \leq nd\max\{\frac{r-s}{r}, 0\}.$$

Observe that the result (32) applies under the condition on the noise matrix W' stated in Lemma 13. Our third lemma hence controls the noise term W' in a manner that will allow us to apply Lemma 13.

Lemma 15 *Suppose that $p_{\mathrm{obs}} \geq \frac{1}{\min\{n,d\}}\log^7(nd)$. Then given a random matrix W' with entries distributed according to the distribution* (31)*, we have*

$$\mathbb{P}\big(\|W'\|_{op} > 2.01\sqrt{p_{\mathrm{obs}}(n+d)}\big) \leq e^{-c'\max\{n,d\}}.$$

Based on these three lemmas, we now complete the proof of the theorem. From Lemma 15 we see that the choice $\lambda = 2.1\sqrt{\frac{n+d}{p_{\mathrm{obs}}}}$ guarantees that $\lambda \geq \frac{1.01\|W'\|_{\mathrm{op}}}{p_{\mathrm{obs}}}$ with probability at least $1 - e^{-c'\max\{n,d\}}$. Consequently, the condition required for an application of Lemma 13 is satisfied, and applying this lemma then yields the upper bound

$$\|\widehat{M}_{\mathrm{SVT}} - M^*\|_{\mathrm{F}}^2 \leq c\sum_{j=1}^{n}\min\{\frac{d}{p_{\mathrm{obs}}}, \sigma_j^2(M^*)\} \tag{33}$$

with probability at least $1 - e^{-c'd}$, where we have also used our assumption that $n \leq d$.

Now consider any matrix $M_0 \in \mathbb{R}^{n\times d}$. In what follows, we convert the bound (33) into a bound that depends on the properties of M_0, namely $\overline{\rho}(M_0)$, $\overline{r}(M_0)$ and $\|M^* - M_0\|_{\mathrm{F}}$. Let $\rho_0 = \overline{\rho}(M_0)$ and $r_0 = \overline{r}(M_0)$.

We first have the following deterministic upper bound

$$\begin{aligned}\sum_{j=1}^{n}\min\{\frac{d}{p_{\mathrm{obs}}}, \sigma_j^2(M^*)\} &\leq 2\sum_{j=1}^{n}\min\{\frac{d}{p_{\mathrm{obs}}}, \sigma_j^2(M_0)\} \\ &\quad + 2\sum_{j=1}^{n}\Big(\min\{\sqrt{\frac{d}{p_{\mathrm{obs}}}}, \sigma_j(M^*)\} - \min\{\sqrt{\frac{d}{p_{\mathrm{obs}}}}, \sigma_j(M_0)\}\Big)^2 \\ &\leq 2\sum_{j=1}^{n}\min\{\frac{d}{p_{\mathrm{obs}}}, \sigma_j^2(M_0)\} + 2\sum_{j=1}^{n}\big(\sigma_j(M^*) - \sigma_j(M_0)\big)^2, \end{aligned} \tag{34}$$

where the inequality (34) is a consequence of the more general result that $(\min\{a, b_1\} - \min\{a, b_2\})^2 \leq (b_1 - b_2)^2$ for any three real numbers a, b_1, and b_2. In what follows, we bound the two terms on the right hand side of (34) separately.

Bounding the second term on the right hand side of (34): In order to bound the second term, we call upon a more general relation pertaining to matrices established in Schönemann (1968).

Lemma 16 (Schönemann, 1968) *For any pair of matrices $A, B \in \mathbb{R}^{n\times d}$ with singular value decompositions $A = U_1 D_1 V_1^T$ and $B = U_2 D_2 V_2^T$, it must be that*

$$(U_1 U_2^T,\ V_1 V_2^T) \in \underset{U\in\mathbb{R}^{n\times n},\ V\in\mathbb{R}^{d\times d}}{\arg\min} \|A - UBV^T\|_F^2 \quad \text{such that } U^T = U^{-1},\ V^T = V^{-1}. \tag{35}$$

To bound the second term on the right hand side of (34), we use Lemma 16 with the choices $A = M^*$ and $B = M_0$. With this choice, some simple algebra yields that the minimum value of the objective $\|A - UBV^T\|_F^2$ in (35) equals $\sum_{j=1}^n (\sigma_j(M^*) - \sigma_j(M_0))^2$. On the other hand, the choice of U and V as identity matrices is feasible for (35), and the associated value of the objective equals $\|M^* - M_0\|_F^2$. Consequently, we have the inequality

$$\sum_{j=1}^n (\sigma_j(M^*) - \sigma_j(M_0))^2 \leq \|M^* - M_0\|_F^2. \tag{36}$$

Bounding the first term on the right hand side of (34): As for the first term of (34), an application of Lemma 14(a) to the matrix M_0 yields the bound

$$\begin{aligned}\sum_{j=1}^n \min\{\frac{d}{p_{\text{obs}}}, \sigma_j^2(M_0)\} &\leq \min_{s\in[n]} (\frac{sd}{p_{\text{obs}}} + \frac{(\rho_0)^2 nd}{s}) \\ &\leq 3\rho_0 d\sqrt{\frac{n}{p_{\text{obs}}}},\end{aligned} \tag{37}$$

where inequality (37) is obtained with the choice $s = \lceil \rho_0\sqrt{p_{\text{obs}} n}\rceil$. Separately, an application of Lemma 14(b) to the matrix M_0 yields

$$\begin{aligned}\sum_{j=1}^n \min\{\frac{d}{p_{\text{obs}}}, \sigma_j^2(M_0)\} &\leq \min_{s\in[n]} \big(\frac{sd}{p_{\text{obs}}} + nd\max\{1 - \frac{s}{r_0}, 0\}\big) \\ &\leq \frac{r_0 d}{p_{\text{obs}}},\end{aligned} \tag{38}$$

where the inequality (38) is obtained with the choice $s = r_0$.

Putting it all together: Combining the bounds (33), (34), (36), (37) and (38), we obtain the result that the inequality

$$\|\widehat{M}_{\text{SVT}} - M^*\|_F^2 \leq 2\min\Big\{\frac{3\rho_0 d\sqrt{n}}{\sqrt{p_{\text{obs}}}}, \frac{r_0 d}{p_{\text{obs}}}\Big\} + 2\|M^* - M_0\|_F^2,$$

must hold with probability at least $1 - e^{c'd}$. Finally, recalling our assumption that $d \geq n$ and substituting $n = \min\{n, d\}$ and $d = \max\{n, d\}$ yields the claimed result.

Proof of Lemma 14 Part (a): Without loss of generality, assume that $d \geq n$.

We begin with an upper bound on the tail of the singular values of any matrix in $\mathbb{C}_{\text{PR}}(1)$, that is, of any matrix that has a permutation-rank of 1. The proof of this bound uses a construction due to Chatterjee (2014) for a rank $\widetilde{s}$ approximation of any matrix in $\mathbb{C}_{\text{PR}}(1)$, for any value $\widetilde{s} \in [n]$. We first reproduce Chatterjee's construction.

For a given matrix $M \in \mathbb{C}_{\text{PR}}(1)$, define the vector $\tau \in \mathbb{R}^d$ of column sums—namely, with entries $\tau_j = \sum_{i=1}^n [M]_{ij}$ for $j \in [d]$. Using this vector, define a rank $\widetilde{s}$ approximation $\widetilde{M}$ to M by grouping the columns according to the vector τ according to the following procedure:

- Observing that each $\tau_j \in [0, n]$, divide the full interval $[0, n]$ into $\widetilde{s}$ groups—say of the form
$$[0, n/\widetilde{s}), [n/\widetilde{s}, 2n/\widetilde{s}), \ldots [(\widetilde{s} - 1)n/\widetilde{s}, n].$$
If τ_j falls into the interval α for some $\alpha \in [\widetilde{s}]$, then map column j to the group G_α of indices.
- For each $\alpha \in [\widetilde{s}]$ such that group G_α is non-empty, choose a particular column index $j' \in G_\alpha$ in an *arbitrary* fashion. For every other column index $j \in G_\alpha$, set $\widetilde{M}_{ij} = M_{ij'}$ for all $i \in [n]$.

By construction, the matrix $\widetilde{M}$ has at most $\widetilde{s}$ distinct rows, and hence rank at most $\widetilde{s}$. Now consider any column $j \in [d]$ and suppose that $j \in G_\alpha$ for some $\alpha \in [\widetilde{s}]$. Let j' denote the column chosen for the group G_α in the second step of the construction. Since $M \in \mathbb{C}_{\text{PR}}(1)$, we must either have $M_{ij} \geq M_{ij'} = \widetilde{M}_{ij}$ for every $i \in [n]$, or $M_{ij} \leq M_{ij'} = \widetilde{M}_{ij}$ for every $i \in [n]$. Then we are guaranteed that

$$\sum_{i=1}^n |\widetilde{M}_{ij} - M_{ij}| = |\sum_{i=1}^n (\widetilde{M}_{ij} - M_{ij})| = |\tau_{j'} - \tau_j| \leq \frac{n}{\widetilde{s}}, \tag{39}$$

where we have used the fact the pair $(\tau_j, \tau_{j'})$ must lie in an interval of length at most $n/\widetilde{s}$. This completes the description of Chatterjee's construction.

In what follows, we use Chatterjee's result in order to obtain our claimed bound on the tail of the spectrum of any matrix $M \in \mathbb{C}_{\text{PR}}(\rho)$. We modify the result in a careful manner that allows us to obtain the desired dependence on the parameter ρ. Recall that any matrix $M \in \mathbb{C}_{\text{PR}}(\rho)$ can be decomposed as

$$M = \sum_{\ell=1}^{\rho} M^{(\ell)},$$

for some matrices $M^{(1)}, \ldots, M^{(\rho)} \in \mathbb{C}_{\text{PR}}(1)$. Let $\widetilde{s} = \frac{s}{\rho}$. For every $\ell \in [r]$, let $\widetilde{M}^{(\ell)}$ be a rank $\widetilde{s} = \frac{s}{\rho}$ approximation of $M^{(\ell)}$ obtained from Chatterjee's construction above, but with the following additional detail. Observe that in Chatterjee's construction, the choice of column j' from group G_α is arbitrary. For our construction, we will make a specific choice of this column: we choose the column whose entries have the smallest values among all columns in the group G_α. With this choice, we have the property

$$\widetilde{M}_{ij}^{(\ell)} \leq M_{ij}^{(\ell)} \qquad \text{for every } \ell \in [\rho],\ i \in [n], j \in [d]. \tag{40}$$

Now let $\widetilde{M} := \sum_{\ell=1}^{\rho} \widetilde{M}^{(\ell)}$. Since every entry of every matrix $\widetilde{M}^{(\ell)}$ is non-negative, we have that every entry of $\widetilde{M}$ is also non-negative. We also claim that

$$\widetilde{M}_{ij} = \sum_{\ell=1}^{\rho} \widetilde{M}_{ij}^{(\ell)} \overset{(i)}{\leq} \sum_{\ell=1}^{\rho} M_{ij}^{(\ell)} = M_{ij} \leq 1,$$

where the inequality (i) is a consequence of the set of inequalities (40). Thus we have that $\widetilde{M} \in [0,1]^{n\times d}$, $M \in [0,1]^{n\times d}$, and that the rank of $\widetilde{M}$ is at most $\rho\widetilde{s}$. This result then yields the bound

$$\begin{aligned}\sum_{j=\rho\widetilde{s}+1}^{n} \sigma_j^2(M) \leq \|M - \widetilde{M}\|_{\mathrm{F}}^2 \leq \sum_{i=1}^{n}\sum_{j=1}^{d} |M_{ij} - \widetilde{M}_{ij}| &= \sum_{i=1}^{n}\sum_{j=1}^{d} |\sum_{\ell=1}^{\rho}(M_{ij}^{(\ell)} - \widetilde{M}_{ij}^{(\ell)})| \\ &\overset{(i)}{\leq} \sum_{i=1}^{n}\sum_{j=1}^{d}\sum_{\ell=1}^{\rho} |M_{ij}^{(\ell)} - \widetilde{M}_{ij}^{(\ell)}|,\end{aligned}$$

where inequality (i) follows from the triangle inequality. Now recall that every matrix $\widetilde{M}^{(\ell)}$ is obtained from Chatterjee's construction, and rewriting Chatterjee's result (39) for the matrices $\widetilde{M}^{(\ell)}$ presently under consideration, we obtain

$$\sum_{i=1}^{n} |\widetilde{M}_{ij}^{(\ell)} - M_{ij}^{(\ell)}| \leq \frac{n}{\widetilde{s}},$$

for every $\ell \in [\rho]$. As a consequence, we have

$$\sum_{j=\rho\widetilde{s}+1}^{n} \sigma_j^2(M) \leq \frac{\rho n d}{\widetilde{s}} = \frac{\rho^2 n d}{s},$$

where we have substituted the relation $\widetilde{s} = \frac{s}{\rho}$ to obtain the final result.

Part (b): This result follows directly from the facts that the rank of M is at most r, and the square of its Frobenius norm is at most nd.

Proof of Lemma 15 Define an $((n+d)\times(n+d))$ matrix W'' as

$$W'' = \frac{1}{\sqrt{p_{\mathrm{obs}}}} \begin{bmatrix} 0 & W' \\ (W')^T & 0 \end{bmatrix}.$$

From (31) and the construction above, we have that the matrix W'' is symmetric, and has mutually independent entries above the diagonal that have a mean of zero and a variance upper bounded by 1. Consequently, known results in random matrix theory (e.g., see Chatterjee, 2014, Theorem 3.4 or Tao, 2012, Theorem 2.3.21) yield the bound $\|W''\|_{\mathrm{op}} \leq 2.01\sqrt{n+d}$ with probability at least $1 - e^{-c\max\{n,d\}}$, under the assumption $p_{\mathrm{obs}} \geq \frac{1}{\min\{n,d\}}\log^7(nd)$. One can also verify that $\|W''\|_{\mathrm{op}} = \frac{1}{\sqrt{p_{\mathrm{obs}}}}\|W'\|_{\mathrm{op}}$, yielding the claimed result.

4.4. Proof of Proposition 3

We recall that for any integer $k \geq 0$, the notation J_k denotes an upper triangular matrix of size $(k\times k)$ with all entries on and above the diagonal set as 1, and I_k denotes the identity matrix of size $(k\times k)$. Consider an $(n\times d)$ matrix M with the following block structure:

$$M := \begin{bmatrix} J_{r-\rho+1} & 0 & 0 \\ 0 & I_{\rho-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In the remainder of the proof, we show that $\overline{r}(M) = r$ and $\overline{\rho}(M) = \rho$. Using the ideas in the construction of M and the associated proof to follow, one can construct many other matrices that have a non-negative rank of r and a permutation-rank of ρ, for any given value $1 \leq \rho \leq r \leq \min\{n,d\}$.

We partition the proof into four parts.

Proof of $\overline{r}(M) \leq r$: One can write M as a sum of r matrices, each having a non-negative rank of one: for each non-zero row, consider a component matrix comprising that row and zeros elsewhere. Consequently, we have $\overline{r}(M) \leq r$.

Proof of $\overline{r}(M) \geq r$: Observe that the (conventional) rank of M^* equals r. Since the rank of any matrix is a lower bound on its non-negative rank, we have that $\overline{r}(M) \geq r$. We have thus established that the non-negative rank of this matrix equals exactly r.

Proof of $\overline{\rho}(M) \leq \rho$: Observe that the $(n\times d)$ matrix with $J_{r-\rho+1}$ as its top-left submatrix and zeros elsewhere has a permutation-rank of 1. Moreover, any $(n\times d)$ matrix with exactly one entry as 1 and the remaining entries 0 also has a permutation-rank of 1, and hence a $(n\times d)$ matrix with $I_{\rho-1}$ as its submatrix and zeros elsewhere has a permutation-rank of at most $(\rho-1)$. Putting these arguments together, we obtain the bound $\overline{\rho}(M) \leq \rho$.

Proof that $\overline{\rho}(M) \geq \rho$: First observe that the matrix

$$I_{2\times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

does not belong to $\mathbb{C}_{\mathrm{PR}}(1)$. It follows that any matrix containing $I_{2\times 2}$ as a submatrix cannot belong to the set $\mathbb{C}_{\mathrm{PR}}(1)$. It further follows that for any positive integer k, the matrix $I_{k\times k}$ must have a permutation rank of at least k. Finally, observe that the matrix M contains $I_{\rho\times\rho}$ as its submatrix (given by the intersection of rows $\{r-\rho,\ldots,r\}$ with the columns $\{r-\rho,\ldots,r\}$). It follows that M must have a permutation rank of at least ρ, thereby proving the claim.

4.5. Proof of Proposition 4

We assume for ease of exposition that n and d are divisible by k. Otherwise, since $k \leq \frac{1}{2}\min\{n,d\}$, one may take floors or ceilings which will change the result only by a constant

factors. Since $\mathbb{C}_{\rm NR}(k) \subseteq \mathbb{C}_{\rm PR}(k)$, we have $\sup_{M \in \mathbb{C}_{\rm NR}(k)} \inf_{M' \in \mathbb{C}_{\rm PR}(k)} \|M - M'\|_{\rm F}^2 = 0$. In what follows, we show that $\sup_{M \in \mathbb{C}_{\rm PR}(k)} \inf_{M' \in \mathbb{C}_{\rm NR}(k)} \|M - M'\|_{\rm F}^2 \geq \frac{cnd}{k}$.

Consider the block matrix $\widetilde{M} \in [0,1]^{\frac{n}{k} \times \frac{d}{k}}$:

$$\widetilde{M} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \tag{41}$$

where each of the four blocks is of size $(\frac{n}{2k} \times \frac{d}{2k})$. The following lemma shows that the best rank-1 approximation to $\widetilde{M}$ has a large approximation error:

Lemma 17 *For the matrix $\widetilde{M}$ defined in (41), for any vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$, it must be that*

$$\|\widetilde{M} - uv^T\|_F^2 \geq c\frac{nd}{k^2},$$

where $c > 0$ is a universal constant.

We now use the matrix $\widetilde{M}$ defined in (41) to build the following block matrix $M \in \mathbb{C}_{\rm PR}(k)$:

$$M := \begin{bmatrix} \widetilde{M} & 0 & \cdots & 0 \\ 0 & \widetilde{M} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \widetilde{M} \end{bmatrix}.$$

In words, the matrix M is a block-diagonal matrix where the diagonal has k copies of $\widetilde{M}$.

Due to the block diagonal structure of M, the singular values of M are simply k copies of the singular values of its constituent matrix $\widetilde{M}$. Consequently, we have that for any matrix $M' \in \mathbb{C}_{\rm NR}(k)$:

$$\|M - M'\|_{\rm F}^2 \geq k(\|\widetilde{M}\|_{\rm F}^2 - \|\widetilde{M}\|_{\rm op}^2) \overset{(i)}{\geq} c\frac{nd}{k},$$

as claimed, where the inequality (i) is a consequence of Lemma 17.

Proof of Lemma 17 Consider any value $i \in [\frac{n}{2k}]$ and $j \in [\frac{d}{2k}]$. Then we claim that

$$(\widetilde{M}_{i,j} - [uv^T]_{i,j})^2 + (\widetilde{M}_{i+\frac{n}{2k},j} - [uv^T]_{i+\frac{n}{2k},j})^2 + (\widetilde{M}_{i,j+\frac{d}{2k}} - [uv^T]_{i,j+\frac{d}{2k}})^2 + (\widetilde{M}_{i+\frac{n}{2k},j+\frac{d}{2k}} - [uv^T]_{i+\frac{n}{2k},j+\frac{d}{2k}})^2 \geq 0.01. \tag{42}$$

If not, then for the choice of $\widetilde{M}$ in (41), we must have $[uv^T]_{i,j} \in (0.9, 1.1)$, $[uv^T]_{i+\frac{n}{2k},j} \in (0.9, 1.1)$, $[uv^T]_{i,j+\frac{d}{2k}} \in (0.9, 1.1)$ and $[uv^T]_{i+\frac{n}{2k},j+\frac{d}{2k}} < 0.1$. However, since $[uv^T]_{i',j'} = u_{i'}v_{j'}$ for every coordinate (i', j'), we also have

$$[uv^T]_{i,j} \times [uv^T]_{i+\frac{n}{2k},j+\frac{d}{2k}} = [uv^T]_{i+\frac{n}{2k},j} \times [uv^T]_{i,j+\frac{d}{2k}},$$

which contradicts the required ranges of the individual coordinates. Summing the bound (42) over all values of $i \in [\frac{n}{2k}]$ and $j \in [\frac{d}{2k}]$ yields the claimed result.

4.6. Proof of Proposition 5

Consider any set $\mathbb{S}$ and any convex set $\mathbb{C}$. We begin with a key lemma that establishes a relation between $\mathcal{H}(\mathbb{S}, \mathbb{C})$ and a proposed notion of the inherent convexity of $\mathbb{S}$.

Lemma 18 *For any set $\mathbb{S} \subseteq [0,1]^{n \times d}$ and any convex set $\mathbb{C} \subseteq [0,1]^{n \times d}$, it must be that*

$$\mathcal{H}(\mathbb{S}, \mathbb{C}) \geq \frac{2}{9} \sup_{M_1 \in \mathbb{S},\ M_2 \in \mathbb{S}} \inf_{M_0 \in \mathbb{S}} \|\frac{1}{2}(M_1 + M_2) - M_0\|_F^2. \tag{43}$$

The left hand side of inequality (43) is the Hausdorff distance between the sets $\mathbb{S}$ and $\mathbb{C}$ in terms of the squared Frobenius norm. The right hand side of the inequality represents a notion of the inherent convexity of the set $\mathbb{S}$.

With this lemma in place, we now complete the remainder of the proof. To this end, we set $\mathbb{S} = \mathbb{C}_{\rm PR}(1)$, and let $\mathbb{C}$ be any convex set of $[0,1]$-valued $(n \times d)$ matrices.

We now construct a pair of matrices $M_1 \in \mathbb{C}_{\rm PR}(1)$ and $M_2 \in \mathbb{C}_{\rm PR}(1)$ that we use to lower bound the right hand side of (43). Define matrices $M_1 \in \mathbb{C}_{\rm PR}(1)$ and $M_2 \in \mathbb{C}_{\rm PR}(1)$ as

$$[M_1]_{ij} = \begin{cases} 1 & \text{if } i \leq \frac{n}{2},\ j \leq \frac{d}{2} \\ 0 & \text{otherwise,} \end{cases} \qquad \text{and} \qquad [M_2]_{ij} = \begin{cases} 1 & \text{if } i > \frac{n}{2},\ j > \frac{d}{2} \\ 0 & \text{otherwise.} \end{cases}$$

It follows that the entries of the matrix $\frac{1}{2}(M_1 + M_2)$ are given by:

$$[\frac{1}{2}(M_1 + M_2)]_{ij} = \begin{cases} \frac{1}{2} & \text{if } (i \leq \frac{n}{2},\ j \leq \frac{d}{2}) \text{ or } (i > \frac{n}{2},\ j > \frac{d}{2}) \\ 0 & \text{otherwise.} \end{cases}$$

Now consider any pair of integers $(i, j) \in [\lfloor n/2 \rfloor] \times [\lfloor d/2 \rfloor]$. Then the (2×2) submatrix of $\frac{1}{2}(M_1 + M_2)$ formed by its entries (i, j), $(i + \lceil n/2 \rceil, j)$, $(i, j + \lceil d/2 \rceil)$ and $(i + \lceil n/2 \rceil, j + \lceil d/2 \rceil)$ equals

$$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}.$$

It is easy to verify that there is a constant $c > 0$ such that the squared Frobenius norm distance between this rescaled identity matrix and any (2×2) matrix in $\mathbb{C}_{\rm PR}(1)$ is at least c. Since this argument holds for any choice of $(i, j) \in [\lfloor n/2 \rfloor] \times [\lfloor d/2 \rfloor]$, summing up the errors across each of these sets of entries yields

$$\|\frac{1}{2}(M_1 + M_2) - M\|_{\rm F}^2 \geq c'nd, \qquad \text{for every matrix } M \in \mathbb{C}_{\rm PR}(1),$$

where $c' > 0$ is a universal constant. Finally, substituting this bound in Lemma 18 yields the claimed result.

It remains to prove Lemma 18.

Proof of Lemma 18. Consider any matrices $M_1 \in \mathbb{S}$ and $M_2 \in \mathbb{S}$. From the definition of the Hausdorff distance $\mathcal{H}$, we know that there exist matrices $\widetilde{M}_1 \in \mathbb{C}$ and $\widetilde{M}_2 \in \mathbb{C}$ such that

$$\|M_i - \widetilde{M}_i\|_{\mathrm{F}}^2 \leq \mathcal{H}(\mathbb{S}, \mathbb{C}), \quad \text{for } i \in \{1, 2\}. \tag{44}$$

Since $\mathbb{C}$ is a convex set, we also have $\frac{1}{2}(\widetilde{M}_1 + \widetilde{M}_2) \in \mathbb{C}$. Then from the definition of $\mathcal{H}$, we also know that there exists a matrix $M_0 \in \mathbb{S}$ such that

$$\|\frac{1}{2}(\widetilde{M}_1 + \widetilde{M}_2) - M_0\|_{\mathrm{F}}^2 \leq \mathcal{H}(\mathbb{S}, \mathbb{C}). \tag{45}$$

Finally, applying the triangle inequality to the bounds (44) and (45) yields

$$\begin{aligned}\|\frac{1}{2}(M_1 + M_2) - M_0\|_{\mathrm{F}}^2 &\leq 3\|\frac{1}{2}(\widetilde{M}_1 + \widetilde{M}_2) - M_0\|_{\mathrm{F}}^2 + \frac{3}{4}\|M_1 - \widetilde{M}_1\|_{\mathrm{F}}^2 + \frac{3}{4}\|M_2 - \widetilde{M}_2\|_{\mathrm{F}}^2 \\ &\leq \frac{9}{2}\mathcal{H}(\mathbb{S}, \mathbb{C}).\end{aligned}$$

4.7. Proof of Proposition 6

Suppose there exists a coordinate pair (i, j) such the stated condition is violated. Then there must exist two distinct values $\ell_1 \in [\overline{\rho}(M)]$ and $\ell_2 \in [\overline{\rho}(M)]$ that satisfy the following three conditions:

(a) $M_{ij}^{(\ell_1)} > 0$ and $M_{ij}^{(\ell_2)} > 0$,

(b) The value of $M_{ij}^{(\ell_1)}$ is different from all other entries in $M^{(\ell_1)}$, and

(c) The value of $M_{ij}^{(\ell_2)}$ is different from all other entries in $M^{(\ell_2)}$.

In addition, the fact that $M_{ij}^{(\ell_1)} + M_{ij}^{(\ell_2)} \in (0, 1)$ for every coordinate (i, j), along with condition (a) above, imply a fourth condition:

(d) $M_{ij}^{(\ell_1)} < 1$ and $M_{ij}^{(\ell_2)} < 1$.

Now for any $\epsilon > 0$, define $\widetilde{M}_\epsilon^{(\ell_1)}$ and $\widetilde{M}_\epsilon^{(\ell_2)}$ to be matrices obtained by replacing the $(i, j)^{th}$ entries of the matrices $M^{(\ell_1)}$ and $M^{(\ell_2)}$ with $(M_{ij}^{(\ell_1)} + \epsilon)$ and $(M_{ij}^{(\ell_2)} - \epsilon)$ respectively. Now, conditions (b)–(d) in tandem imply that there exists some value $\epsilon > 0$ such that *all* of the following properties hold:

(i) $[M_\epsilon^{(\ell_1)}]_{ij} \in [0, 1]$,

(ii) $[M_\epsilon^{(\ell_2)}]_{ij} \in [0, 1]$, and

(iii) The relative ordering of the entries of $M^{(\ell_1)}$ is identical to the relative ordering of the entries of $M_\epsilon^{(\ell_1)}$; the relative ordering of the entries of $M^{(\ell_1)}$ is identical to the relative ordering of the entries of $M_\epsilon^{(\ell_1)}$.

Properties (i) and (ii) imply that $\widetilde{M}_\epsilon^{(\ell_1)} \in [0, 1]^{n \times d}$ and $\widetilde{M}_\epsilon^{(\ell_2)} \in [0, 1]^{n \times d}$. Combined with property (iii), we also have $\widetilde{M}_\epsilon^{(\ell_1)} \in \mathbb{C}_{\mathrm{PR}}(1)$ and $\widetilde{M}_\epsilon^{(\ell_2)} \in \mathbb{C}_{\mathrm{PR}}(1)$. Finally, from the construction of the matrices $\widetilde{M}_\epsilon^{(\ell_1)}$ and $\widetilde{M}_\epsilon^{(\ell_2)}$, it is easy to see the relation

$$M = \widetilde{M}_\epsilon^{(\ell_1)} + \widetilde{M}_\epsilon^{(\ell_2)} + \sum_{i \in [\overline{\rho}(M)] \setminus \{\ell_1, \ell_2\}} M^{(i)}.$$

This decomposition of M is a different, valid permutation-rank decomposition of M.

5. Discussion and future work

We posit that the conventional low-rank models for matrix completion and denoising are equivalent to "parametric" assumptions with undesirable implications. We propose a new permutation-rank approach and argue, by means of a conceptual discussion as well as theoretical guarantees, that this approach offers significant benefits at little additional cost. Our work also contributes to a growing body of literature (Shah, 2017, Part 1), (Shah et al., 2017; Shah and Wainwright, 2018; Shah et al., 2016, 2019; Heckel et al., 2016; Chatterjee and Mukherjee, 2019; Flammarion et al., 2016; Chen et al., 2017) on moving towards more flexible models based on permutations that provide robustness to model mismatches.

In this paper we considered a binary-valued observation matrix. More generally, one may have observations that are real valued and bounded, in which case, our results continue to hold. Specifically, if we have every entry of Y and M^* is bounded by a constant, and if the noise is assumed to be of zero mean, then the statistical results presented here continue to hold (with possibly different constant factors which may depend on the bounding constant). We note that the bounded-entries assumption holds in most practical rating schemes deployed today.

We studied a low non-negative rank framework inspired by its interpretation discussed in Appendix A. It will be of interest to extend this framework to a regular low-rank matrix completion problem setting. The permutation-rank version of this setting would be identical to that defined in this paper, but where the constituent rank one components are not restricted to have entries $[0, 1]$. In terms of computational aspects of this model, we are optimistic for the following reason. Presently, given any matrix, we do not know how to compute its permutation rank (both non-negative and conventional) in a computationally efficient manner. However, the (classical) rank of a matrix is easily computable efficiently, whereas the non-negative rank is known to be computationally hard. We hope that the relative ease of computing the classical rank may carry over to the permutation-rank setting. In terms of statistical aspects, we expect some of our present results to carry over to the permutation-rank version of regular low-rank matrix completion, including Theorem 1(b) and Theorem 2 with some modifications. However, at this point we do not know if the proof of the information-theoretic upper bound of Theorem 1(a) can be readily translated (with or without any additional assumptions), and we leave this question open for future work.

Our work gives rise to some useful open problems that we hope to address in future work. In this paper, we established benefits of the permutation-based approach for the matrix completion problem under the random design observation setting. In the literature, the classical low (non-negative) rank matrix completion problem is more recently also studied under other observation models such as weighted random sampling (Negahban and Wainwright, 2012), fixed design (Jain et al., 2013; Klopp, 2014), streaming/active learning (Yun et al., 2015; Jin et al., 2016; Balcan and Zhang, 2016), or biased observation models (Hsieh et al., 2015), which are also of interest in the context of permutation-rank matrix completion. A second open problem is to close the gap between the statistically optimal minimax rate of estimation and the best known rate for polynomial-time computable algorithms for the permutation-rank model. Any solution to this problem may also contribute to the understanding of some other open problems in the literature (e.g., see Shah et al., 2017; Flammarion et al., 2016; Shah et al., 2016) on the gap between the statistical

and computational aspects of estimation under an unknown permutation. Finally, we hope that the insights and techniques for flexible models we developed can be useful in problems of broader interest in the domain of "learning from people", to handle issues like miscalibration (Wang and Shah, 2019), subjectivity (Noothigattu et al., 2018), biases (Stelmakh et al., 2019), noise (Stelmakh et al., 2018), and strategic behavior (Xu et al., 2019), where models need to be flexible in order to accommodate complex human behavior.

Acknowledgments

The work of NBS was partially supported by NSF grants 1755656 and 1763734. The work of MJW and NBS was partially supported by DOD Advanced Research Projects Agency W911NF-16-1-0552 and National Science Foundation grant NSF-DMS-1612948. The work of SB was supported by NSF grants DMS-1713003 and CIF-1763734.

References

Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM, 2012.

Maria-Florina F Balcan and Hongyang Zhang. Noise-tolerant life-long matrix completion via adaptive sampling. In *Advances In Neural Information Processing Systems*, 2016.

Jian-Feng Cai and Stanley Osher. Fast singular value thresholding without singular value decomposition. *UCLA CAM Report*, 5, 2010.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, December 2009.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inf. Theor.*, 56(5):2053–2080, May 2010.

Sabyasachi Chatterjee and Sumit Mukherjee. Estimation in tournaments and graphs under monotonicity constraints. *IEEE Transactions on Information Theory*, 2019.

Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2014.

Xi Chen, Sivakanth Gopi, Jieming Mao, and Jon Schneider. Competitive analysis of the top-k ranking problem. In *ACM-SIAM Symposium on Discrete Algorithms*, 2017.

Yudong Chen, Ali Jalali, Sujay Sanghavi, and Constantine Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Trans. Information Theory*, 59(7):4324–4337, 2013.

Thomas Cover and Joy Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4): 608–622, 2016.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, 2003.

David Donoho, Matan Gavish, et al. Minimax risk of matrix denoising by singular value thresholding. *The Annals of Statistics*, 42(6):2413–2440, 2014.

Nicolas Flammarion, Cheng Mao, and Philippe Rigollet. Optimal rates of statistical seriation. *arxiv:1607.02435*, 2016.

Edgar N Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31 (3):504–522, 1952.

Nicolas Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13(Nov):3349–3386, 2012.

Nicolas Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257), 2014.

Cyril Gobinet, Eric Perrin, and Régis Huez. Application of non-negative matrix factorization to fluorescence spectroscopy. In *European Signal Processing Conference*, 2004.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans. Information Theory*, 57(3):1548–1566, 2011.

Johan Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

Reinhard Heckel, Nihar B Shah, Kannan Ramchandran, and Martin J Wainwright. Active ranking from pairwise comparisons and when parametric assumptions don't help. *arxiv:1606.08842*, 2016.

Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Studies in Applied Mathematics*, 6(1-4):164–189, 1927.

Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit Dhillon. PU learning for matrix completion. In *International Conference on Machine Learning*, pages 2445–2453, 2015.

Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *ACM Innovations in Theoretical Computer Science*, 2013.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *ACM symposium on Theory of computing*, 2013.

Chi Jin, Sham M Kakade, and Praneeth Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2016.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010.

Thierry Klein and Emmanuel Rio. Concentration around the mean for maxima of empirical processes. *The Annals of Probability*, 33(3):1060–1077, 2005.

Olga Klopp. Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*, 20(1):282–303, 2014.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Vladimir Koltchinskii, Karim Lounici, and Alexandre B Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, pages 2302–2329, 2011.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

Hans Laurberg, Mads Græsbøll Christensen, Mark D Plumbley, Lars Kai Hansen, and Søren Holdt Jensen. Theorems on positive data: On the uniqueness of NMF. *Computational intelligence and neuroscience*, 2008, 2008.

M. Laurent. Matrix completion problems. In *The Encyclopedia of Optimization*, pages 221—229. Kluwer Academic, 2001.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Sahand Negahban and Martin J Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13(May):1665–1697, 2012.

Ritesh Noothigattu, Nihar Shah, and Ariel Procaccia. Choosing how to choose papers. *arXiv preprint arxiv:1808.09057*, 2018.

Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Linear regression with an unknown permutation: Statistical and computational limits. *arxiv:1608.02902*, 2016.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011.

Ryan Sarver and Anssi Klapuri. Application of nonnegative matrix factorization to signal-adaptive audio effects. In *Proc. DAFx*, pages 249–252, 2011.

Peter H Schönemann. On two-sided orthogonal procrustes problems. *Psychometrika*, 33(1): 19–33, 1968.

Nihar Shah. *Learning From People*. PhD thesis, EECS Department, University of California, Berkeley, Jul 2017.

Nihar B Shah and Martin J Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *Journal of machine learning research*, 2018.

Nihar B Shah, Sivaraman Balakrishnan, and Martin J Wainwright. A permutation-based model for crowd labeling: Optimal estimation and robustness. *arxiv:1606.09632*, 2016.

Nihar B Shah, Sivaraman Balakrishnan, Adityanand Guntuboyina, and Martin J Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *IEEE Transactions on Information Theory*, 2017.

Nihar B Shah, Sivaraman Balakrishnan, and Martin J Wainwright. Feeling the Bern: Adaptive estimators for Bernoulli probabilities of pairwise comparisons. *IEEE Transactions on Information Theory*, 2019.

Nathan Srebro, Noga Alon, and Tommi S Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances In Neural Information Processing Systems*, 2005.

Ivan Stelmakh, Nihar Shah, and Aarti Singh. PeerReview4All: Fair and accurate reviewer assignment in peer review. *arXiv preprint arxiv:1806.06237*, 2018.

Ivan Stelmakh, Nihar Shah, and Aarti Singh. On testing for biases in peer review. In *ACM EC Workshop on Mechanism Design for Social Good*, 2019.

Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Society Providence, RI, 2012.

Fabian J Theis, Kurt Stadlthanner, and Toshihisa Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *European Signal Processing Conference*, 2005.

A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. 2008.

Antoine Vandendorpe, Ngoc-Diep Ho, Steven Vanduffel, and Paul Van Dooren. On the parameterization of the creditrisk+ model for estimating credit portfolio risk. *Insurance: Mathematics and Economics*, 42(2):736–745, 2008.

RR Varshamov. Estimate of the number of signals in error correcting codes. In *Dokl. Akad. Nauk SSSR*, 1957.

Jingyan Wang and Nihar B Shah. Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings. In *AAMAS*, 2019.

Yichong Xu, Han Zhao, Xiaofei Shi, and Nihar Shah. On strategyproof conference review. In *IJCAI*, 2019.

Se-Young Yun, Marc Lelarge, and Alexandre Proutiere. Streaming, memory limited matrix completion with noise. *arxiv:1504.03156*, 2015.

Appendix

Appendix A. Alternative interpretation of the non-negative rank model

In the non-negative rank model described in the introduction, one may wonder why the affinity of a user to a movie conditioned on a feature must be modeled as the product $u_i^\ell v_j^\ell$ of the separate connections of the user and movie to the feature. Secondly, one may also wonder why the net affinity of a user to a movie is the sum of the affinities across the features $\sum_{\ell=1}^r u_i^\ell v_j^\ell$. These two modeling assumptions may sometimes be confusing, and hence in what follows, we present an alternative interpretation of the low non-negative rank model for the recommender systems application.

Consider any feature $\ell \in [r]$. The affinities of users towards movies conditioned on this feature is a matrix, say $X^{(\ell)} \in [0,1]^{n\times d}$. The matrix $X^{(\ell)}$ is assumed to have a (non-negative) rank of 1. Hence the probability that user i likes movie j, when asked to judge only based on feature ℓ, equals $X_{ij}^{(\ell)}$.

Now, every user is assumed to have their own way of weighing features to decide which movies they like. Specifically, any user $i \in [n]$ is associated to values $\alpha_i^{(1)}, \ldots, \alpha_i^{(r)}$ such that $\alpha_i^{(\ell)} \geq 0$ for every $\ell \in [r]$ and $\sum_{\ell=1}^r \alpha_i^{(\ell)} = 1$. The probability that user i likes any movie j is assumed to be the convex combination

$$\sum_{\ell=1}^r \alpha_i^{(\ell)} X_{ij}^{(\ell)}.$$

This completes the description of the model.

Let us verify that the resulting user-movie matrix has a non-negative rank of r. Recall the assumption that $X^{(\ell)}$ has a non-negative rank of 1, and let $X^{(\ell)} = u^\ell (v^\ell)^T$ for some vectors u^ℓ and v^ℓ. Then the ith row of the overall user-movie matrix equals $\sum_{\ell=1}^r \alpha_i^{(\ell)} u_i^\ell (v^\ell)^T$, and hence the overall user-movie matrix equals

$$\sum_{\ell=1}^r \widetilde{u}^\ell (v^\ell)^T, \qquad \text{where} \quad \widetilde{u}^\ell = \begin{bmatrix} \alpha_1^{(\ell)} u_1^\ell \\ \vdots \\ \alpha_n^{(\ell)} u_n^\ell \end{bmatrix}.$$

This completes the alternative description of the non-negative rank model.

One can observe that the restriction $\sum_{\ell=1}^r \alpha_i^{(\ell)} = 1$ makes this model slightly more restrictive than the non-negative rank model described earlier in the main text. However, all of our results on estimation for the non-negative rank model described in Section 2.1 continue to apply to this model as well.

Appendix B. Intuitive algorithms that provably fail

In this section, we present two intuitive polynomial-time computable algorithms for the permutation-rank setting—one for estimating M^* from Y, and one for decomposing M^* into its constituent permutation-rank-one matrices—and show that these algorithms provably fail. Our goal in describing these negative results is as a complement to the positive results provided in the main text, and with the hope that the points of failure of these algorithms may form starting points for subsequent research.

B.1. An intuitive polynomial-time estimator for M^* from Y

In this section, we consider the problem of estimating the matrix M^* from noisy and partial observations Y as defined earlier in equation (1). For simplicity, we assume that $p_{\text{obs}} = 1$. For any vector $z \in \mathbb{R}^m$, we let vector $z_+ \in \mathbb{R}^m$ with entries $[z_+]_i = \max\{z_i, 0\}$ represent the positive component of z, and vector $z_- \in \mathbb{R}^m$ with entries $[z_-]_i = \min\{z_i, 0\}$ represent the negative component of z.

We first provide some intuition and background to motivate the estimator we study in this section, and then present a formal definition. Denote the permutation-rank of matrix M^* as $\rho^* := \overline{\rho}(M^*)$, and assume that the value of ρ^* is known. The goal is to obtain an estimate $\widehat{M} \in \mathbb{C}_{\text{PR}}(\rho^*)$ from the observed matrix Y such that the error $\frac{1}{nd}\|M^* - \widehat{M}\|_{\text{F}}^2$ is as small as possible. For any such matrix $\widehat{M}$, let us use the following notation for its permutation-rank decomposition: $\widehat{M} = \sum_{\ell=1}^{\rho^*} \widehat{M}^{(\ell)}$ where $\widehat{M}^{(\ell)} \in \mathbb{C}_{\text{PR}}(1)$ for every $\ell \in [\rho^*]$. Further, we let $\widehat{\pi}^{(\ell)}$ and $\widehat{\sigma}^{(\ell)}$ respectively denote the permutation of the rows and columns of $\widehat{M}^{(\ell)}$.

Past literature on computationally-efficient estimation for such problems provides us with estimators with the following two distinct goals: (i) to estimate the permutations of the constituent matrices in the permutation-rank decomposition, and (ii) estimators to compute the entries of the constituent matrices given the permutations. For each of these two goals, we describe a natural estimator below from past works.

Estimating the permutations via singular value decomposition: Compute the singular value decomposition $Y = \sum_{\ell=1}^{\min\{n,d\}} a^{(\ell)}[b^{(\ell)}]^T$ such that the vectors $\{a^{(1)}, \ldots, a^{(\min\{n,d\})}\}$ are mutually orthogonal, the vectors $\{b^{(1)}, \ldots, b^{(\min\{n,d\})}\}$ are mutually orthogonal, and $[a^{(1)}]^T b^{(1)} \geq \ldots \geq [a^{(\min\{n,d\})}]^T b^{(\min\{n,d\})}$. In order to resolve a global sign ambiguity, we also mandate the condition $\|a_+^{(\ell)}\|_2 \geq \|a_-^{(\ell)}\|_2$ for every $\ell \in [\min\{n,d\}]$. Finally, for each $\ell \in [\rho^*]$, set $\widehat{\pi}^{(\ell)}$ and $\widehat{\sigma}^{(\ell)}$ as the ordering of the entries of $a^{(\ell)}$ and $b^{(\ell)}$ respectively.

From past works (Shah et al., 2016), this estimator for the permutations is known to possess appealing properties for the case when $\rho^* = 1$. For instance, it is not hard to see that in a noiseless setting where $Y = M^*$, the estimator will yield exactly the row and column permutations of $M^* \in \mathbb{C}_{\text{PR}}(1)$. This fact is employed in Shah et al. (2016) to obtain consistent estimates of the permutations associated to an unknown matrix in $\mathbb{C}_{\text{PR}}(1)$ in the context of a "crowd labeling" problem. It is also not hard to verify that the estimator is can be computed in a computationally-efficient manner.

Estimating the entries via least squares, when given the permutations: Given some estimate $\widehat{\pi}^{(1)}, \widehat{\sigma}^{(1)}, \ldots, \widehat{\pi}^{(\rho^*)}, \widehat{\sigma}^{(\rho^*)}$ of the permutations associated to the permutation-rank decomposition of M^*, the following estimator $\widehat{M}$ provides an estimate of the matrix M^* as well as the matrices in its permutation-rank decomposition.

$$\widehat{M} \in \underset{M \in [0,1]^{n \times d}}{\arg\min} \; \|Y - M\|_{\mathrm{F}}^2 \tag{46}$$

$$\text{such that} \quad M = \sum_{\ell=1}^{\rho^*} \widetilde{M}^{(\ell)}, \text{ and}$$

$$\widetilde{M}^{(\ell)} \in \mathbb{C}_{\mathrm{PR}}(1) \text{ with rows and columns ordered by } (\widehat{\pi}^{(\ell)}, \widehat{\sigma}^{(\ell)}), \text{ for every } \ell \in [\rho^*].$$

The aforementioned estimator $\widehat{M}$ is a natural extension of the least-squares estimators studied in past works (Shah et al., 2017, 2019) for the case when $M^* \in \mathbb{C}_{\mathrm{PR}}(1)$. The estimator is known to have appealing properties from both the statistical and computational perspectives. From a computational standpoint, all of the constraints in the optimization program (46) can be expressed as a set of (polynomial number of) linear inequalities, thereby making the optimization problem computationally tractable. From a statistical standpoint, if the given permutations $(\widehat{\pi}^{(1)}, \widehat{\sigma}^{(1)}, \ldots, \widehat{\pi}^{(\rho^*)}, \widehat{\sigma}^{(\rho^*)})$ are exactly (or approximately) equal to the permutations associated to a permutation-rank decomposition of M^*, the proofs of the results in Shah et al. (2017, 2019) as well as Theorem 1 in the present paper imply that the estimator $\widehat{M}$ is minimax optimal for estimating M^* from Y. The estimator $\widehat{M}$ continues to remain statistically efficient if the permutations are known up to a reasonable approximation.

Given the two intuitive estimators discussed above, a natural means to estimate M^* from Y is to concatenate these two estimators to obtain the following two-step estimator: Step 1: From Y, obtain an estimate $(\widehat{\pi}^{(1)}, \widehat{\sigma}^{(1)}, \ldots, \widehat{\pi}^{(\rho^*)}, \widehat{\sigma}^{(\rho^*)})$ of the permutations of the decomposition of M^* via the singular value decomposition-based estimator described above. Step 2: Using the estimates of the permutations, obtain an estimate $\widehat{M}$ of M^* via the least squares projection (46).

We believe that when $\rho^* = 1$, this estimator is not only consistent, but it has an expected error decaying at the rate $\mathcal{O}(\min\{n, d\}^{-1/2})$. We now show that in fact as soon as one moves to the setting of $\rho^* > 1$, this estimator is no longer even consistent—even if there is no noise.

Proposition 19 *There exists a matrix $M^* \in \mathbb{C}_{PR}(2)$ such that when $Y = M^*$, the two-step estimator $\widehat{M}$ has an error lower bounded as*

$$\frac{1}{nd} \|M^* - \widehat{M}\|_F^2 \geq c_2,$$

with probability 1.

The proof of this result is provided in Appendix B.3.1. The proof also demonstrates an identical negative result for the following modified estimation algorithm: In computing $\widehat{M}$ as above, instead of taking only the permutations of the top ρ^* singular vectors, collect permutations from singular vectors until you obtain ρ^* *distinct* permutations; then apply the least squares projection step to these ρ^* distinct permutations.

B.2. An intuitive greedy algorithm for permutation-rank decomposition

Consider any matrix any matrix $M \in [0,1]^{n \times d}$. The singular value decomposition of M into components having a (conventional) rank of one can be performed with the following greedy algorithm:

- Let $\widehat{k} = 1$
- While $M \neq \sum_{\ell=1}^{\widehat{k}-1} \widehat{M}^{(\ell)}$:
  - Let $\widehat{M}^{(\widehat{k})} \in \underset{\substack{M' = ab^T \\ (a,b) \in \mathbb{R}^n \times \mathbb{R}^d}}{\arg\min} \; \|M - \sum_{\ell=1}^{\widehat{k}-1} \widehat{M}^{(\ell)} - M'\|_F$
  - $\widehat{k} = \widehat{k} + 1$
- Output $\widehat{k}$ as the rank of M and $\{\widehat{M}^{(1)}, \ldots, \widehat{M}^{(\widehat{k})}\}$ as its singular value decomposition.

An obvious question that arises is whether a similar greedy algorithm works to obtain a permutation-rank decomposition.

To this end, consider any value $q \geq 1$, and for any matrix M, let $\|M\|_q$ denote its entry-wise norm $\|M\|_q := \left(\sum_{i,j} (M_{ij})^q\right)^{\frac{1}{q}}$. Then the natural analogue of the aforedescribed algorithm in the context of permutation-rank decomposition is as follows:

- Let $\widehat{\rho} = 1$
- While $M \neq \sum_{\ell=1}^{\widehat{\rho}-1} \widehat{M}^{(\ell)}$:
  - Let $\widehat{M}^{(\widehat{\rho})} \in \underset{M' \in \mathbb{C}_{\mathrm{PR}}(1)}{\arg\min} \; \|M - \sum_{\ell=1}^{\widehat{\rho}-1} \widehat{M}^{(\ell)} - M'\|_q$
  - $\widehat{\rho} = \widehat{\rho} + 1$
- Output $\widehat{\rho}$ as the permutation-rank of M and $\{\widehat{M}^{(1)}, \ldots, \widehat{M}^{(\widehat{\rho})}\}$ as its permutation-decomposition

The following proposition investigates whether such an algorithm will work.

Proposition 20 *For any values of n, d and $\rho \geq 2$, there exists an $(n \times d)$ matrix $M \in \mathbb{C}_{PR}(\rho)$ such that the above algorithm outputs a decomposition of permutation-rank at least $(\rho + 1)$.*

The guaranteed incorrectness of the permutation rank of the output of the algorithm also directly implies that the decomposition is also incorrect.

B.3. Proofs

We now present the proofs of the negative results introduced in this section.

B.3.1. Proof of Proposition 19

In what follows, for clarity of exposition, we ignore issues pertaining to floors and ceilings of numbers, as they affect the results only by a constant factor.

We begin by defining a matrix $M^* \in \mathbb{C}_{\text{PR}}(2)$ as

$$M^* = M^{(1)} + M^{(2)},$$

with

$$M^{(1)} = a^{(1)}(b^{(1)})^T + a^{(2)}(b^{(2)})^T \quad \text{and} \quad M^{(2)} = a^{(3)}(b^{(3)})^T.$$

Set

$$\begin{aligned} a^{(1)} &= [1 \;\; .9 \;\cdots\; .9 \quad .8 \;\cdots\; .8 \quad 0 \;\cdots\; 0 \;]^T \\ a^{(2)} &= [0 \;\; .2 \;\cdots\; .2 \quad -.1 \;\cdots\; -.1 \quad 0 \;\cdots\; 0 \;]^T \\ a^{(3)} &= [0 \;\; \underbrace{0 \;\cdots\; 0}_{\nu_1(n-1)} \quad \underbrace{0 \;\cdots\; 0}_{\nu_2(n-1)} \quad \underbrace{1 \;\cdots\; 1}_{\nu_3(n-1)} \;]^T, \end{aligned}$$

and

$$\begin{aligned} b^{(1)} &= [1 \;\; .9 \;\cdots\; .9 \quad .8 \;\cdots\; .8 \quad 0 \;\cdots\; 0 \;]^T \\ b^{(2)} &= [0 \;\; .2 \;\cdots\; .2 \quad -.1 \;\cdots\; -.1 \quad 0 \;\cdots\; 0 \;]^T \\ b^{(3)} &= [0 \;\; \underbrace{0 \;\cdots\; 0}_{\nu_1(d-1)} \quad \underbrace{0 \;\cdots\; 0}_{\nu_2(d-1)} \quad \underbrace{1 \;\cdots\; 1}_{\nu_3(d-1)} \;]^T, \end{aligned}$$

where $\nu_1 = .684$, $\nu_2 = .304$, and $\nu_3 = .012$. It is easy to verify that all entries of the matrices $M^*, M^{(1)}, M^{(2)}$ lie in the interval $[0, 1]$ and that $M^{(1)} \in \mathbb{C}_{\text{PR}}(1)$ and $M^{(2)} \in \mathbb{C}_{\text{PR}}(1)$ and $M^* \in \mathbb{C}_{\text{PR}}(2) \backslash \mathbb{C}_{\text{PR}}(1)$.

One can further verify the following properties of this construction:

1. $\langle a^{(\ell)}, a^{(\ell')} \rangle = 0$ and $\langle b^{(\ell)}, b^{(\ell')} \rangle = 0$ for every $\ell \neq \ell' \in \{1, 2, 3\}$.
2. $\|a^{(1)}\|_2 > \|a^{(2)}\|_2 > \|a^{(3)}\|_2$ and $\|b^{(1)}\|_2 > \|b^{(2)}\|_2 > \|b^{(3)}\|_2$.
3. The (conventional) rank of M^* is 3.
4. $a^{(1)}$, $a^{(2)}$ and $a^{(3)}$ have different permutations of their entries; likewise, $b^{(1)}$, $b^{(2)}$ and $b^{(3)}$ have different permutations of their entries.
5. $\|a_+^{(\ell)}\|_2 \geq \|a_-^{(\ell)}\|_2$ and $\|b_+^{(\ell)}\|_2 \geq \|b_-^{(\ell)}\|_2$ for every $\ell \in [3]$.

The five properties listed above imply that the following decomposition of M^*,

$$M^* = \sum_{\ell=1}^{3} a^{(\ell)}(b^{(\ell)})^T,$$

is a valid singular value decomposition with the global signs of the constituent vectors satisfying the conditions of Step 1 of the algorithm. Consequently, the $\rho^* = 2$ estimated

permutations in Step 1 of the algorithm are those given by the respective orderings of the entries of the vectors $\{a^{(1)}, b^{(1)}\}$ and $\{a^{(2)}, b^{(2)}\}$.

Observe that the entries 2 to $(1+\nu_1(n-1))$ of both $a^{(1)}$ and $a^{(2)}$ have values higher than the last $\nu_3(n-1)$ entries of these vectors, and hence this ordering is reflected in the respective permutations derived from these vectors. Likewise, the entries 2 through $(1+\nu_1(d-1))$ are ranked higher than the last $\nu_3(d-1)$ entries in the permutation derived from the vectors $b^{(1)}$ and $b^{(2)}$. Due to this collection of inequalities, the least squares program in Step 2 of the algorithm must mandate that

$$M_{ij} \geq M_{i'j'} \geq M_{i''j''}, \tag{47}$$

whenever $2 \leq i, i' \leq 1 + \nu_1(n-1)$; $n - \nu_3(n-1) < i'' \leq n$; $2 \leq j \leq 1 + \nu_1(d-1)$; and $d - \nu_3(d-1) < j', j'' \leq d$. However, for each coordinate in this range, we also have $M^*_{ij} = .85, M^*_{i'j'} = 0, M^*_{i''j''} = 1$. Consequently, any triplet of values $(M_{ij}, M_{i'j'}, M_{i''j''})$ that follows the ordering (47) must necessarily incur an error lower bounded as

$$(M_{ij} - M^*_{ij})^2 + (M_{i'j'} - M^*_{i'j'})^2 + (M_{i''j''} - M^*_{i''j''})^2 \geq c,$$

for some universal constant $c > 0$. Summing up the errors over all the entries of the matrix in the aforementioned coordinate set yields that any matrix M satisfying the constraints of the least squares problem must have squared Frobenius error at least $\|M - M^*\|_{\text{F}}^2 \geq c'nd$, for some universal constant $c' > 0$.

B.3.2. Proof of Proposition 20

First let $n = d = \rho = 2$. Consider the (2×2) matrix M defined as

$$M := \begin{bmatrix} 0 & .6 \\ .6 & .4 \end{bmatrix}.$$

It is easy to verify that $M \in \mathbb{C}_{\text{PR}}(2) \backslash \mathbb{C}_{\text{PR}}(1)$.

Let us now investigate the operation of the proposed algorithm on this matrix M. The following lemma controls the first step of the algorithm.

Lemma 21 *When the input matrix M is as defined above, the algorithm will select*

$$\widehat{M}^{(1)} = \begin{bmatrix} 0 & .4 \\ .4 & .4 \end{bmatrix}$$

in the first iteration.

As a consequence of this lemma, we have the following residual that is used for the subsequent iterations of the algorithm:

$$M - \widehat{M}^{(1)} = \begin{bmatrix} 0 & .2 \\ .2 & 0 \end{bmatrix}.$$

It is easy to see the that the residual matrix $(M - \widehat{M}^{(1)}) \in \mathbb{C}_{\text{PR}}(2) \backslash \mathbb{C}_{\text{PR}}(1)$. Also observe that in each iteration, the algorithm subtracts out a matrix in $\mathbb{C}_{\text{PR}}(1)$ from the residual.

Consequently, the algorithm will require at least two more iterations to terminate. The algorithm thus necessarily outputs a decomposition with $\widehat{\rho} \geq 3$, as claimed.

Next we extend these arguments to any arbitrary values of $n \geq 2, d \geq 2, \rho \geq 2$. Consider matrix M with entries:

- $M_{11} = 0$, $M_{12} = M_{21} = .6$, $M_{22} = .4$
- $M_{ii} = 1$ for every $i \in \{3, \ldots, \rho\}$
- $M_{ij} = 0$ for every other coordinate (i, j).

The matrix M has a block-diagonal structure with the top-left (2×2) block as one non-zero component and $(\rho - 2)$ other entries on the diagonal as $(\rho - 2)$ additional non-zero components.

The rest of the proof is partitioned into two cases:
Case I: Suppose that at some step $\widehat{\rho}$ of the algorithm, some entry of the residual matrix $(M - \sum_{\ell=1}^{\widehat{\rho}} \widehat{M}^{(\ell)}$ is strictly negative. Then the algorithm will never terminate because every subsequent candidate matrix in the minimization step of the algorithm must lie in $\mathbb{C}_{\text{PR}}(1)$ and hence must have non-negative entries. The algorithm will thus output $\widehat{\rho} = \infty$.

Case II: Now suppose that the residual matrices always have non-negative entires. Then given the block-diagonal structure of M, any matrix $\widehat{M}^{(1)}, \widehat{M}^{(2)}, \ldots$ in the iterations of the algorithm can be non-zero in exactly one of these diagonal components. As a result, the overall decomposition yielded by the algorithm decouples into ρ individual decompositions of the ρ respective blocks, each of which will contribute a permutation-rank of at least 1. Moreover, from the arguments for the case of $n = d = 2$ above, we also have that the top-left (2×2) block will induce a decomposition of permutation-rank 3 in the algorithm. Putting the pieces together, we see that the matrix M will induce the proposed algorithm to output a decomposition of permutation-rank at least $(\rho + 1)$, whereas $\overline{\rho}(M) = \rho$.

Proof of Lemma 21 Since $M_{11} = 0$, we must have $\widehat{M}_{11}^{(1)} = 0$. Now suppose the column ordering of $\widehat{M}^{(1)}$ is such that the first column is greater than the second column. Then we must have $\widehat{M}_{12}^{(1)} = 0$. Since $M^{(1)}$ is the minimizer of the optimization program in the algorithm we must then have $\widehat{M}_{21}^{(1)} = M_{21}$ and $\widehat{M}_{22}^{(1)} = M_{22}$, and consequently $\|M - \widehat{M}^{(1)}\|_q = \widehat{M}_{12}^{(1)} = .6$. An analogous argument holds if the first row is greater than the second row in the permutation of $\widehat{M}^{(1)}$. Finally, suppose that in the permutations of $M^{(1)}$, the second column is greater than the first column and the second row is greater than the first row. Then we must have $.4 \geq \widehat{M}_{22}^{(1)} \geq \max\{\widehat{M}_{12}^{(1)}, \widehat{M}_{21}^{(1)}\}$. With this condition, one can see that the minimizer of the optimization program is $\widehat{M}_{12}^{(1)} = \widehat{M}_{21}^{(1)} = \widehat{M}_{22}^{(1)} = .4$. Consequently, we have $\|M - \widehat{M}^{(1)}\|_q = .2 \times 2^{\frac{1}{q}} < .6$ for any $q \geq 1$. Thus the algorithm chooses

$$\widehat{M}^{(1)} = \begin{bmatrix} 0 & .4 \\ .4 & .4 \end{bmatrix}.$$

Journal of Machine Learning Research 20 (2019) 1-56 Submitted 10/17; Revised 11/18; Published 4/19

Non-Convex Matrix Completion and Related Problems via Strong Duality

Maria-Florina Balcan NINAMF@CS.CMU.EDU
Carnegie Mellon University

Yingyu Liang YLIANG@CS.WISC.EDU
University of Wisconsin-Madison

Zhao Song ZHAOS@G.HARVARD.EDU
UT-Austin & Harvard University

David P. Woodruff DWOODRUF@CS.CMU.EDU
Carnegie Mellon University

Hongyang Zhang* HONGYANZ@CS.CMU.EDU
Carnegie Mellon University & TTIC

Editor: Moritz Hardt

Abstract

This work studies the *strong duality of non-convex matrix factorization problems*: we show that under certain dual conditions, these problems and the dual have the same optimum. This has been well understood for convex optimization, but little was known for non-convex problems. We propose a novel analytical framework and prove that under certain dual conditions, the optimal solution of the matrix factorization program is the same as that of its bi-dual and thus the global optimality of the non-convex program can be achieved by solving its bi-dual which is convex. These dual conditions are satisfied by a wide class of matrix factorization problems, although matrix factorization is hard to solve in full generality. This analytical framework may be of independent interest to non-convex optimization more broadly.

We apply our framework to two prototypical matrix factorization problems: matrix completion and robust Principal Component Analysis. These are examples of efficiently recovering a hidden matrix given limited reliable observations. Our framework shows that exact recoverability and strong duality hold with nearly-optimal sample complexity for the two problems.

Keywords: strong duality, non-convex optimization, matrix factorization, matrix completion, robust principal component analysis, sample complexity

1. Introduction

Non-convex matrix factorization problems have been an emerging object of study in theoretical computer science (Balcan et al., 2019, 2018; Arora et al., 2012; Jain et al., 2013; Hardt, 2014; Sun and Luo, 2015; Moitra, 2016; Razenshteyn et al., 2016; Song et al., 2017, 2019), optimization (Wen et al., 2012; Shen et al., 2014), machine learning (Bhojanapalli et al., 2016b; Ge et al., 2016, 2015; Jain et al., 2010; Li et al., 2016; Wang and Xu, 2012), and many other domains (Bouwmans et al., 2018). In theoretical computer science and optimization, the study of such models has led to significant advances in provable algorithms that converge to local minima in linear time (Jain et al., 2013; Hardt, 2014; Sun and Luo, 2015; Agarwal et al., 2017; Allen-Zhu, 2017). In machine learning, matrix factorization serves as a building block for large-scale prediction and recommendation systems, e.g., the winning submission for the Netflix prize (Koren et al., 2009). The matrix factorization problems can be stated as finding a target matrix $\mathbf{X}^*$ in the form of $\mathbf{X}^* = \mathbf{AB}$, by minimizing the objective function $H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2$ or $H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2$ over factor matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times n_2}$ with a known value of $r \ll \min\{n_1, n_2\}$, where $H(\cdot)$ is some function that characterizes the desired properties of $\mathbf{X}^*$. Two prototypical examples are matrix completion and robust Principal Component Analysis (PCA).

This work develops a novel framework to analyze a class of non-convex matrix factorization problems and show their strong duality, which leads to exact recoverability for matrix completion and robust PCA via the solutions to convex optimization problems. Strong duality is well understood for convex optimization, but very few non-convex problems were known to have this property. The results in this work thus significantly expand the set of non-convex problems with strong duality. Furthermore, our framework also shows exact recoverability of the two prototypical examples matrix completion and robust PCA with nearly-optimal sample complexity.

Our work is motivated by several promising areas where our analytical framework for non-convex matrix factorizations is applicable. The first area is low-rank matrix completion. It has been shown that a low-rank matrix can be exactly recovered by finding a solution of the form $\mathbf{AB}$ that is consistent with the observed entries (assuming that it is incoherent) (Jain et al., 2013; Sun and Luo, 2015; Ge et al., 2016). This problem has received a tremendous amount of attention due to its important role in optimization and its wide applicability in many areas such as quantum information theory and collaborative filtering (Hardt, 2014; Zhang et al., 2016; Balcan and Zhang, 2016). The second area is robust PCA, a fundamental problem of interest in data processing. It aims at recovering both the low-rank and the sparse components exactly from their superposition (Candès et al., 2011; Netrapalli et al., 2014; Gu et al., 2016; Zhang et al., 2015a, 2016; Yi et al., 2016), where the low-rank component corresponds to the product of $\mathbf{A}$ and $\mathbf{B}$ while the sparse component is captured by a proper choice of function $H(\cdot)$, e.g., the ℓ_1 norm (Candès et al., 2011; Awasthi et al., 2016). Besides these two areas, we believe that our analytical framework can be potentially applied to other non-convex problems more broadly, e.g., matrix sensing (Tu et al., 2016), dictionary learning (Sun et al., 2017b), weighted low-rank approximation (Razenshteyn et al., 2016; Li et al., 2016), and deep linear neural network (Kawaguchi, 2016), which may be of independent interest.

Without assumptions on the structure of the objective function, direct formulations of matrix factorization problems are NP-hard to optimize in general (Hardt et al., 2014; Zhang et al., 2013). With standard assumptions on the structure of the problem and with sufficiently many samples, these optimization problems can be solved efficiently, e.g., by convex relaxation (Candès and Recht, 2009; Chen, 2015; Foygel and Srebro, 2011). Some other methods run local search algorithms given an initialization close enough to the global solution in the basin of attraction (Jain et al., 2013; Hardt, 2014; Sun and Luo, 2015; Ge et al., 2015; Jin et al., 2017). However, these methods have sample complexity significantly larger than the information-theoretic lower bound; see Table 1.1 for a comparison. The problem becomes even more challenging when the number of samples is small enough that the sample-based initialization can be far from the desired solution, in which case the algorithm can run into a local minimum or a saddle point.

Another line of work has focused on studying the loss surface of matrix factorization problems, providing positive results for approximately achieving global optimality. One nice property in this line of research is that there is no spurious local minima for specific applications such as matrix

*. Corresponding author.

©2019 Maria-Florina Balcan, Yingyu Liang, Zhao Song, David P. Woodruff, and Hongyang Zhang. JMLR 20(102):1-56, 2019
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-611.html.

completion (Ge et al., 2016), matrix sensing (Bhojanapalli et al., 2016b), dictionary learning (Sun et al., 2017b), phase retrieval (Sun et al., 2016), linear deep neural networks (Kawaguchi, 2016), etc. However, these results are based on concrete forms of objective functions. Also, even when any local minimum is guaranteed to be globally optimal, in general it remains NP-hard to escape high-order saddle points (Anandkumar and Ge, 2016), and additional arguments are needed to show the achievement of a local minimum. Most importantly, all existing results rely on strong assumptions on the sample size.

1.1. Our Results

Our work studies a variety of non-convex matrix factorization problems, and the goal is to provide a unified framework to analyze a large class of matrix factorization problems and to provide efficient algorithms to achieve global optimum. Our main results show that although matrix factorization problems are hard to optimize in general, *under certain dual conditions the duality gap is zero*, and thus the problem can be converted to an equivalent convex program.

To state the main theorem of our framework, recall that a function $H(\cdot)$ is closed if for each $\alpha \in \mathbb{R}$, the sub-level set $\{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2} : H(\mathbf{X}) \leq \alpha\}$ is a closed set. Also, recall the nuclear norm (a.k.a. trace norm) of a matrix $\mathbf{X}$ is $\|\mathbf{X}\|_* = \sum_{i=1}^r \sigma_i(\mathbf{X})$. Define the r^*-norm to be $\|\mathbf{X}\|_{r*} = \max_{\mathbf{M}} \langle \mathbf{M}, \mathbf{X} \rangle - \frac{1}{2}\|\mathbf{M}\|_r^2$ where $\|\mathbf{M}\|_r^2 = \sum_{i=1}^r \sigma_i^2(\mathbf{M})$ is the sum of the first r largest squared singular values. Note that both $\|\mathbf{X}\|_*$ and $\|\mathbf{X}\|_{r*}$ are convex functions. Our main results are as follows.

Theorems 3 and 4 (Strong Duality. Informal). *Under certain dual conditions, strong duality holds for the non-convex optimization problem*

$$(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \operatorname*{argmin}_{\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{r \times n_2}} H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2, \tag{1}$$

where $H(\cdot)$ is convex and closed. In other words, problem (1) *and its bi-dual problem*

$$\widetilde{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}} H(\mathbf{X}) + \|\mathbf{X}\|_{r*}, \tag{2}$$

have exactly the same optimal solutions in the sense that $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \widetilde{\mathbf{X}}$.

Similarly, under certain dual conditions, strong duality holds for the non-convex optimization problem

$$(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname*{argmin}_{\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{r \times n_2}} H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2, \tag{3}$$

where $H(\cdot)$ is convex and closed. In other words, problem (1) *and its bi-dual problem*

$$\bar{\mathbf{X}} = \operatorname*{argmin}_{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}} H(\mathbf{X}) + \|\mathbf{X}\|_*, \tag{4}$$

have exactly the same optimal solutions in the sense that $\bar{\mathbf{A}}\bar{\mathbf{B}} = \bar{\mathbf{X}}$.

Description of Dual Conditions. Intuitively, the dual conditions in the above-mentioned theorems state that the angle between $\partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$ and the row and column spaces of $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$ is small. In other words, there is a matrix in the sub-differential set $\partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$ which has almost the same row and

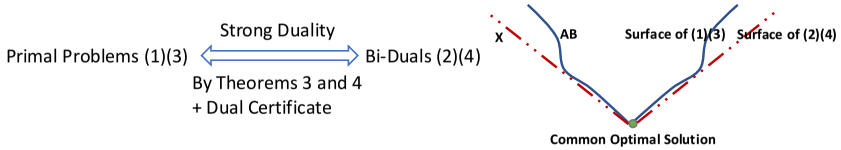


Figure 1: Strong duality of matrix factorizations.

column spaces as matrix $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$. For example, we have $\partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) = \Omega$ for the matrix completion problem, where Ω represents the subspace of matrices supported on the observed indices. Then the dual conditions require that there is a matrix which is supported on the observed indices and shares almost the same row and column spaces as $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$.

Theorem 3 connects the non-convex programs (1) to its convex counterpart (2) via strong duality; see Figure 1. Note that strong duality rarely holds in the non-convex optimization region: low-rank matrix approximation (Overton and Womersley, 1992; Zhang et al., 2019) and quadratic optimization with two quadratic constraints (Beck and Eldar, 2006) are among the few paradigms that enjoy such a nice property. Given strong duality, the computational issues of the original problem can be overcome by solving the convex bi-dual problem (2). The theorem connects the regularization $\frac{1}{2}\|\mathbf{AB}\|_F^2$ to the r^* norm $\|\mathbf{X}\|_{r*}$. This regularization is of special interest to many matrix factorization problems. For example, when $H(\mathbf{AB}) = \frac{1}{2}\|\mathbf{X}\|_F^2 - \langle \mathbf{X}, \mathbf{AB} \rangle$, problem (1) reduces to the PCA problem: $\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{X} - \mathbf{AB}\|_F^2$. When $H(\mathbf{AB}) = \frac{1}{2}\|\mathbf{X}\|_F^2 - \langle \mathbf{X}, \mathbf{AB} \rangle + \gamma\|\mathbf{A}\|_F^2 + \gamma\|\mathbf{B}\|_F^2$, problem (1) reduces to the quadratically regularized PCA problem (Udell et al., 2016): $\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{X} - \mathbf{AB}\|_F^2 + \gamma\|\mathbf{A}\|_F^2 + \gamma\|\mathbf{B}\|_F^2$. Our framework of strong duality is then applicable to all these problems.

Furthermore, Theorem 4 also connects the non-convex programs (3) to its convex counterpart (4): the theorem connects $\frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2$ to the nuclear norm $\|\mathbf{X}\|_*$. This gives new insights for the nuclear norm relaxation technique commonly used for optimization problems with low rank constraints from the perspective of strong duality.

The positive results of our framework are complemented by a lower bound to formalize the hardness of the above problem in general. Assuming that the random 4-SAT problem (Razenshteyn et al., 2016) is hard (see Conjecture 32), we give a strong negative result for deterministic algorithms. If also $\mathsf{BPP} = \mathsf{P}$ (see Section 7 for a discussion), then the same conclusion holds for randomized algorithms succeeding with probability at least $2/3$.

Theorem 10 (Hardness Statement. Informal). *Assuming that random 4-SAT is hard on average, there is a problem in the form of* (1) *such that any deterministic algorithm achieving* $(1+\epsilon)\mathsf{OPT}$ *in the objective function value with $\epsilon \leq \epsilon_0$ requires $2^{\Omega(n_1+n_2)}$ time, where* OPT *is the optimum and $\epsilon_0 > 0$ is an absolute constant. If* $\mathsf{BPP} = \mathsf{P}$*, then the same conclusion holds for randomized algorithms succeeding with probability at least* $2/3$.

Now we turn to the application of our framework. This only requires the verification of the dual conditions in Theorem 3. We will show that two prototypical problems, matrix completion and robust PCA, obey the conditions. They belong to the linear inverse problems of form (1) with a proper choice of function $H(\cdot)$, which aim at exactly recovering a hidden matrix $\mathbf{X}^*$ with $\mathsf{rank}(\mathbf{X}^*) \leq r$ given a limited number of linear observations of it.

For matrix completion, the linear measurements are of the form $\{\mathbf{X}^*_{ij} : (i,j) \in \Omega\}$, where Ω is the support set which is uniformly distributed among all subsets of $[n_1] \times [n_2]$ of cardinality m. With strong duality, we can either study the exact recoverability of the primal problem (1), or investigate the validity of its convex dual (or bi-dual) problem (2). Here we study the former with tools from geometric functional analysis. Recall that in the analysis of matrix completion, one typically requires a μ-incoherence condition for a given rank-r matrix $\mathbf{X}^*$ with skinny SVD $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ (Recht, 2011; Candès and Tao, 2010):

$$\|\mathbf{U}^T\mathbf{e}_i\|_2 \leq \sqrt{\frac{\mu r}{n_1}} \quad \text{for all } i \in [n_1], \qquad \text{and} \qquad \|\mathbf{V}^T\mathbf{e}_i\|_2 \leq \sqrt{\frac{\mu r}{n_2}} \quad \text{for all } i \in [n_2], \tag{5}$$

where $\mathbf{e}_i$'s are basis vectors with i-th entry equal to 1 and other entries equal to 0. The incoherence condition claims that information spreads throughout the left and right singular vectors and is standard in the matrix completion literature. Under this standard condition, we have the following results.

Theorems 6, 7, and 8 (Matrix Completion. Informal). *There exist optimization problems for matrix completion in the forms of* (1) *and* (2) *that enjoy strong duality with each other and exactly recovers* $\mathbf{X}^*$ *with high probability, provided that* $m = \mathcal{O}(\kappa^2\mu(n_1+n_2)r\log(n_1+n_2)\log_{2\kappa}(n_1+n_2))$ *(for the formulation in Theorem 8) or* $m = \mathcal{O}(\mu(n_1+n_2)r\log^2(n_1+n_2))$ *(for the formulation in Theorem 7), where* κ *is the condition number of* $\mathbf{X}^*$. *The sample complexity lower bound is* $\Omega(\mu r(n_1+n_2)\log(n_1+n_2))$.

To the best of our knowledge, our result is the first to connect convex matrix completion to non-convex matrix completion, two parallel lines of research that have received significant attention in the past few years. Table 1.1 compares our results with prior results. Ours match the best known results but further provide strong duality. Also, our results are achieved by a clean framework for a class of related problems.

For robust PCA, instead of studying exact recoverability of problem (1) as for matrix completion, we investigate problem (2) directly. The robust PCA problem is to recover an incoherent low-rank component $\mathbf{X}^*$ and a sparse component $\mathbf{S}^*$ from their sum (Candès et al., 2011; Agarwal et al., 2012). We obtain the following theorem for robust PCA.

Theorems 9 (Robust PCA. Informal). *There exists a convex optimization formulation for robust PCA in the form of problem* (2) *that exactly recovers the incoherent matrix* $\mathbf{X}^* \in \mathbb{R}^{n_1\times n_2}$ *and* $\mathbf{S}^* \in \mathbb{R}^{n_1\times n_2}$ *with high probability, even if* $\mathsf{rank}(\mathbf{X}^*) = \Theta\left(\frac{\min\{n_1,n_2\}}{\mu\log^2\max\{n_1,n_2\}}\right)$ *and the size of the support of* $\mathbf{S}^*$ *is* $m = \Theta(n_1n_2)$, *where the support set of* $\mathbf{S}^*$ *is uniformly distributed among all sets of cardinality* m, *and the incoherence parameter* μ *satisfies the incoherence condition* (5) *and* $\|\mathbf{X}^*\|_\infty \leq \sqrt{\frac{\mu r}{n_1n_2}}\sigma_r(\mathbf{X}^*)$.

The bounds in Theorem 9 match the best known results in the robust PCA literature when the supports of $\mathbf{S}^*$ are uniformly sampled (Candès et al., 2011), while our assumption is arguably more intuitive; see Section 6. Note that our results hold even when $\mathbf{X}^*$ is close to full rank and a constant fraction of the entries have noise.

Independently of our work, Ge et al. (2017) developed a framework to analyze the loss surface of low-rank problems, and applied the framework to matrix completion and robust PCA. For matrix

1. This lower bound is information-theoretic (Candès and Tao, 2010).

Work	Sample Complexity	Incoherence
Jain et al. (2013)	$\mathcal{O}\left(\kappa^4\mu^2r^{4.5}n_{(1)}\log n_{(1)}\log\left(\frac{r\lVert\mathbf{X}^*\rVert_F}{\epsilon}\right)\right)$	(5)
Hardt (2014)	$\mathcal{O}\left(\mu r n_{(1)}(r+\log\left(\frac{n_{(1)}\lVert\mathbf{X}^*\rVert_F}{\epsilon}\right)\frac{\lVert\mathbf{X}^*\rVert_F^2}{\sigma_r^2}\right)$	(5)
Ge et al. (2016)	$\mathcal{O}(\max\{\mu^6\kappa^{16}r^4, \mu^4\kappa^4r^6\}n_{(1)}\log^2 n_{(1)})$	$\lVert\mathbf{X}^*_{i:}\rVert_2 \leq \frac{\mu\lVert\mathbf{X}^*\rVert_F}{\sqrt{n_{(2)}}}$
Sun and Luo (2015)	$\mathcal{O}(rn_{(1)}\kappa^2\max\left\{\mu\log n_{(2)}, \sqrt{\frac{n_{(1)}}{n_{(2)}}}\mu^2r^6\kappa^4\right\}$	(5)
Zheng and Lafferty (2016)	$\mathcal{O}(\mu r^2 n_{(1)}\kappa^2\max(\mu, \log n_{(1)}))$	(5)
Gamarnik et al. (2017)	$\mathcal{O}\left(\left(\mu^2r^4\kappa^2+\mu r\log\left(\frac{\lVert\mathbf{X}^*\rVert_F}{\epsilon}\right)\right)n_{(1)}\log\left(\frac{\lVert\mathbf{X}^*\rVert_F}{\epsilon}\right)\right)$	(5)
Zhao et al. (2015)	$\mathcal{O}\left(\mu r^3 n_{(1)}\log n_{(1)}\log\left(\frac{1}{\epsilon}\right)\right)$	(5)
Keshavan et al. (2010a)	$\mathcal{O}\left(n_{(2)}r\sqrt{\frac{n_{(1)}}{n_{(2)}}}\kappa^2\max\left\{\mu\log n_{(2)}, \mu^2r\sqrt{\frac{n_{(1)}}{n_{(2)}}}\kappa^4\right\}\right)$	(5) and (20)
Gross (2011)	$\mathcal{O}(\mu r n_{(1)}\log^2 n_{(1)})$	(5) and (20)
Chen (2015)	$\mathcal{O}(\mu r n_{(1)}\log^2 n_{(1)})$	(5)
Ours, Theorem 7	$\mathcal{O}(\kappa^2\mu r n_{(1)}\log(n_{(1)})\log_{2\kappa}(n_{(1)}))$	(5)
Ours, Theorem 8	$\mathcal{O}(\mu r n_{(1)}\log^2 n_{(1)})$	(5)
Lower Bound[1]	$\Omega(\mu r n_{(1)}\log n_{(1)})$	(5)

Table 1: Comparison of matrix completion methods. Here $\kappa = \sigma_1(\mathbf{X}^*)/\sigma_r(\mathbf{X}^*)$ is the condition number of $\mathbf{X}^* \in \mathbb{R}^{n_1\times n_2}$, ϵ is the accuracy such that the output $\widetilde{\mathbf{X}}$ obeys $\|\widetilde{\mathbf{X}} - \mathbf{X}^*\|_F \leq \epsilon$, $n_{(1)} = \max\{n_1, n_2\}$ and $n_{(2)} = \min\{n_1, n_2\}$.

completion, their sample complexity is $\mathcal{O}(\kappa^6\mu^4r^6(n_1+n_2)\log(n_1+n_2))$, significantly larger than our bound. For robust PCA, the number of the outlier entries that their method can tolerate is $\mathcal{O}\left(\frac{n_1n_2}{\mu r\kappa^3}\right)$, but their result is for deterministic outlier entries and thus are not directly comparable to ours. Zhang et al. (2017) also studied the robust PCA problem using non-convex optimization, where the outlier entries are also deterministic and the number of outliers that their algorithm can tolerate is $\mathcal{O}\left(\frac{n_1n_2}{r\kappa}\right)$.

1.2. Our Techniques

Reduction to Low-Rank Approximation. Our results are inspired by the low-rank approximation problem:

$$\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} \frac{1}{2}\|-\widetilde{\mathbf{\Lambda}} - \mathbf{A}\mathbf{B}\|_F^2. \tag{6}$$

We know that all local solutions of (6) are globally optimal (see Lemma 1) and that strong duality holds for any given matrix $-\widetilde{\mathbf{\Lambda}} \in \mathbb{R}^{n_1\times n_2}$ (Grussler et al., 2016). To extend this property to our more general problem (1), our main insight is to reduce problem (1) to the form of (6) using the ℓ_2-regularization term. While some prior work attempted to apply a similar reduction, their conclusions either depended on unrealistic conditions on local solutions, e.g., all local solutions are rank-deficient (Haeffele et al., 2014; Grussler et al., 2016), or their conclusions relied on strong assumptions on the objective functions, e.g., that the objective functions are twice-differentiable (Haeffele and Vidal, 2015). For example, the conditions that all local solutions are rank-deficient break down even

for the PCA problem, and the assumptions that the objective function is twice-differential preclude $H(\cdot)$ in (1) and (3) from encoding hard constraints. Instead, our general results formulate strong duality via the existence of a dual certificate $\widetilde{\mathbf{\Lambda}}$. For concrete applications, the existence of a dual certificate is then converted to mild assumptions, e.g., that the number of measurements is sufficiently large and the positions of measurements are randomly distributed. We will illustrate the importance of randomness below.

The Blessing of Randomness. The desired dual certificate $\widetilde{\mathbf{\Lambda}}$ may not exist in the deterministic world. A hardness result (Razenshteyn et al., 2016) shows that for the problem of weighted low-rank approximation, which can be cast in the form of (1), without some randomization in the measurements made on the underlying low rank matrix, it is NP-hard to achieve a good objective value, not to mention to achieve strong duality. A similar result was shown for deterministic matrix completion (Hardt and Moitra, 2013). Thus we should utilize randomness to analyze the existence of a dual certificate. For specific applications such as matrix completion, the assumption that the measurements are random is standard, under which, the angle between the space Ω (the space of matrices which are consistent with observations) and the space $\mathcal{T}$ (the space of matrices which are low-rank) is small with high probability, namely, $\mathbf{X}^*$ is almost the unique low-rank matrix that is consistent with the measurements. Thus, our dual certificate can be represented as another form of a convergent Neumann series concerning the projection operators on the spaces Ω and $\mathcal{T}$; otherwise, the same construction of Neumann series may diverge as the norm concerning the projection operators on the spaces Ω and $\mathcal{T}$ is larger than 1 in the deterministic worst case. The remainder of the proof is to show that such a construction obeys the dual conditions. To show this, we use the fact that the subspace Ω and the complement space $\mathcal{T}^\perp$ are almost orthogonal when the sample size is sufficiently large. This implies the projection of our dual certificate on the space $\mathcal{T}^\perp$ has a very small norm, which exactly matches the dual conditions.

Non-Convex Geometric Analysis. Strong duality implies that the primal problem (1) and its bi-dual problem (2) have exactly the same solutions in the sense that $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \widetilde{\mathbf{X}}$. Thus, to show exact recoverability of linear inverse problems such as matrix completion and robust PCA, it suffices to study either the non-convex primal problem (1) or its convex counterpart (2). Here we do the former analysis for matrix completion. We mention that traditional techniques Candès and Tao (2010); Recht (2011); Chandrasekaran et al. (2012) for convex optimization break down for our non-convex problem, since the subgradient of a non-convex objective function may not even exist Boyd and Vandenberghe (2004). Instead, we apply tools from geometric analysis Vershynin (2009) to analyze the geometry of problem (1). Our non-convex geometric analysis is in stark contrast to prior techniques of convex geometric analysis Vershynin (2015) where convex combinations of non-convex constraints were used to define the Minkowski functional (e.g., in the definition of atomic norm) while our method uses the non-convex constraint itself.

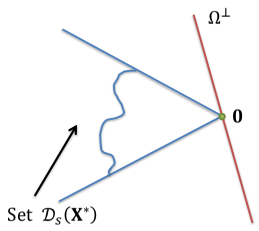


Figure 2: Feasibility.

For matrix completion, problem (1) has two hard constraints: (a) the rank of the output matrix should be no larger than r, as implied by the form of $\mathbf{AB}$; (b) the output matrix should be consistent with the sampled measurements, i.e., $\mathcal{P}_\Omega(\mathbf{AB}) = \mathcal{P}_\Omega(\mathbf{X}^*)$. We study the feasibility condition of

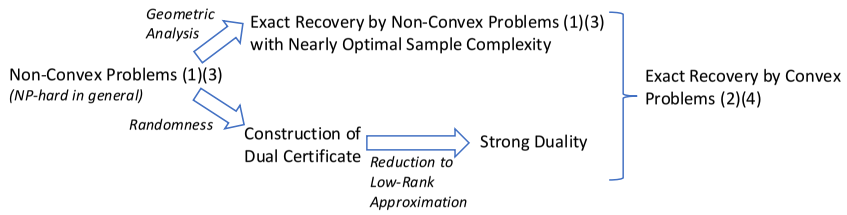


Figure 3: New analytical framework in this paper.

problem (1) from a geometric perspective: $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$ is the unique optimal solution to problem (1) if and only if starting from $\mathbf{X}^*$, either the rank of $\mathbf{X}^* + \mathbf{D}$ or $\|\mathbf{X}^* + \mathbf{D}\|_F$ increases for all directions $\mathbf{D}$'s in the constraint set $\Omega^\perp = \{\mathbf{D} \in \mathbb{R}^{n_1 \times n_2} : \mathcal{P}_\Omega(\mathbf{X}^* + \mathbf{D}) = \mathcal{P}_\Omega(\mathbf{X}^*)\}$. This can be geometrically interpreted as the requirement that the set $\mathcal{D}_\mathcal{S}(\mathbf{X}^*) = \{\mathbf{X} - \mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2} : \mathsf{rank}(\mathbf{X}) \le r, \|\mathbf{X}\|_F \le \|\mathbf{X}^*\|_F\}$ and the constraint set $\Omega^\perp$ must intersect uniquely at $\mathbf{0}$ (see Figure 2). This can then be shown by a dual certificate argument.

Putting Things Together. We summarize our new analytical framework with Figure 3.

Other Techniques. An alternative method is to investigate the exact recoverability of problem (2) via standard convex analysis. We find that the sub-differential of our induced function $\|\cdot\|_{r*}$ has similar properties as that of the nuclear norm. With this observation, we prove the validity of robust PCA in the form of (2) by combining this property of $\|\cdot\|_{r*}$ with standard techniques from (Candès et al., 2011).

1.3. Paper Organization

Section 2 reviews related work and Section 3 defines the notations. The main theorems for our framework are presented in Section 4. Their applications to matrix completion are in Section 5, and those to robust PCA are in Section 6. The computational aspects of the problem are discussed in Section 7, and experimental results on synthetic and real data sets are presented in Section 8. Section 9 concludes the paper. For clarity of presentation, some proofs in our analysis are deferred to the appendix.

2. Related Work

Non-convex matrix factorization is a popular topic studied in theoretical computer science (Jain et al., 2013; Hardt, 2014; Sun and Luo, 2015; Razenshteyn et al., 2016), machine learning (Bhojanapalli et al., 2016b; Ge et al., 2016, 2015; Jain et al., 2010; Li et al., 2016), and optimization (Wen et al., 2012; Shen et al., 2014). We review several lines of research on studying the global optimality of such optimization problems.

Global Optimality of Matrix Factorization. While lots of matrix factorization problems have been shown to have no spurious local minima, they either require additional conditions on the local minima, or are based on particular forms of the objective function. Specifically, Bach et al.

(2008) and Journée et al. (2010) proved that $\mathbf{X} = \mathbf{A}\mathbf{A}^T$ is a global minimizer of $F(\mathbf{X})$, if $\mathbf{A}$ is a rank-deficient local minimizer of $F(\mathbf{A}\mathbf{A}^T)$ and $F(\mathbf{X})$ is a twice differentiable convex function. Haeffele and Vidal (2015) further extended this result by allowing a more general form of objective function $F(\mathbf{X}) = G(\mathbf{X}) + H(\mathbf{X})$, where G is a twice differentiable convex function with compact level set and H is a proper convex function such that F is lower semi-continuous. However, a major drawback of this line of research is that these result fails when the local minimizer is of full rank.

Matrix Completion. Matrix completion is a prototypical example of matrix factorization. One line of work on matrix completion builds on convex relaxation (e.g., Srebro and Shraibman (2005); Candès and Recht (2009); Candès and Tao (2010); Recht (2011); Chandrasekaran et al. (2012); Negahban and Wainwright (2012)), which achieve nearly optimal sample complexity. However, the computational complexity is relatively high compared with the non-convex methods. Recently, Ge et al. (2016) showed that matrix completion has no spurious local optimum, when $|\Omega|$ is sufficiently large and the matrix $\mathbf{Y}$ is incoherent. The result is only for positive semi-definite matrices and their sample complexity is not optimal. Independently of our work, Ge et al. (2017) developed a framework to analyze the loss surface of low-rank problems, and applied the framework to matrix completion and robust PCA. For matrix completion, their sample complexity is $\mathcal{O}(\kappa^6\mu^4 r^6(n_1+n_2)\log(n_1+n_2))$, significantly larger than our bound.

Another line of work is built upon good initialization for global convergence. Recent attempts showed that one can first compute some form of initialization (e.g., by singular value decomposition) that is close to the global minimizer and then use non-convex approaches to reach global optimality, such as alternating minimization, block coordinate descent, and gradient descent (Keshavan et al., 2010b,a; Jain et al., 2013; Keshavan, 2012; Hardt, 2014; Bhojanapalli et al., 2016a; Zheng and Lafferty, 2015; Zhao et al., 2015; Tu et al., 2016; Chen and Wainwright, 2015; Sun and Luo, 2015). The advantage of such line of research is that the computational complexity is low compared with convex relaxation based approach, but the sample complexity is relatively high. In our result, we try to bridge these two lines of research, under the same sample complexity as the best-known result of matrix completion (Chen, 2015).

Robust PCA. Robust PCA is also a prototypical example of matrix factorization. The goal is to recover both the low-rank and the sparse components exactly from their superposition (Chandrasekaran et al., 2011; Candès et al., 2011; Netrapalli et al., 2014; Gu et al., 2016; Zhang et al., 2015a, 2016; Yi et al., 2016). It has been widely applied to various tasks, such as video denoising, background modeling, image alignment, photometric stereo, texture representation, subspace clustering, and spectral clustering.

There are typically two settings in the robust PCA literature: (a) the support set of the sparse matrix is uniformly sampled (Chandrasekaran et al., 2011; Candès et al., 2011; Zhang et al., 2016); b) the support set of the sparse matrix is deterministic, but the non-zero entries in each row or column of the matrix cannot be too large (Yi et al., 2016; Ge et al., 2017). In this work, we discuss the first case. Our framework provides results that match the best known work in setting (a) (Candès et al., 2011). For setting b), Ge et al. (2017) studied the robust PCA and the number of the outlier entries that their method can tolerate is $\mathcal{O}\left(\frac{n_1 n_2}{\mu r \kappa^5}\right)$, but their result is for deterministic outlier entries and thus are not directly comparable to ours. Zhang et al. (2017) also studied the robust PCA problem using non-convex optimization, where the outlier entries are also deterministic and the number of outliers that their algorithm can tolerate is $\mathcal{O}\left(\frac{n_1 n_2}{r\kappa}\right)$.

Other Matrix Factorization Problems. Matrix sensing is another typical matrix factorization problem (Chandrasekaran et al., 2012; Jain et al., 2013; Zhao et al., 2015). Bhojanapalli et al. (2016b), Park et al. (2017) and Tu et al. (2016) showed that the matrix recovery model $\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathcal{A}(\mathbf{A}\mathbf{B} - \mathbf{Y})\|_F^2$, achieves optimality for every local minimum, if the operator $\mathcal{A}$ satisfies the restricted isometry property. They further gave a lower bound and showed that the unstructured operator $\mathcal{A}$ may easily lead to a local minimum which is not globally optimal.

Some other matrix factorization problems are also shown to have nice geometric properties such as the property that all local minima are global minima. Examples include dictionary learning (Sun et al., 2017b), phase retrieval (Sun et al., 2016), and linear deep neural networks (Kawaguchi, 2016). In multi-layer linear neural networks where the goal is to learn a multi-linear projection $\mathbf{X}^* = \prod_i \mathbf{W}_i$, each $\mathbf{W}_i$ represents the weight matrix that connects the hidden units in the i-th and $(i+1)$-th layers. The study of such linear models can help the theoretical understanding of the loss surface of deep neural networks with non-linear activation functions (Kawaguchi, 2016; Choromanska et al., 2015). In dictionary learning, we aim to recover a dictionary matrix $\mathbf{A}$ from a given signal $\mathbf{X}$ in the form of $\mathbf{X} = \mathbf{A}\mathbf{B}$, provided that the representation coefficient $\mathbf{B}$ is sufficiently sparse. This problem centers around solving a non-convex matrix factorization problem with a sparsity constraint on the representation coefficient $\mathbf{B}$ (Bach et al., 2008; Sun et al., 2017b,a; Arora et al., 2014). Other high-impact examples of matrix factorization models range from the classic unsupervised learning problems like PCA, independent component analysis, and clustering, to the more recent problems such as non-negative matrix factorization, weighted low-rank matrix approximation, sparse coding, tensor decomposition (Bhaskara et al., 2014; Anandkumar et al., 2014), subspace clustering (Zhang et al., 2015b, 2014), etc. Applying our framework to these other problems is left for future work.

Atomic Norms. The atomic norm is a recently proposed function for linear inverse problems (Chandrasekaran et al., 2012). Many well-known norms, e.g., the ℓ_1 norm and the nuclear norm, serve as special cases of atomic norms. It has been widely applied to the problems of compressed sensing (Tang et al., 2013), low-rank matrix recovery (Candès and Recht, 2013), blind deconvolution (Ahmed et al., 2014), etc. The norm is defined by the Minkowski functional associated with the convex hull of a set $\mathcal{A}$: $\|\mathbf{X}\|_{\mathcal{A}} = \inf\{t > 0 : \mathbf{X} \in t\mathcal{A}\}$. In particular, if we set $\mathcal{A}$ to be the convex hull of the infinite set of unit-ℓ_2-norm rank-one matrices, then $\|\cdot\|_{\mathcal{A}}$ equals to the nuclear norm. We mention that our objective term $\|\mathbf{A}\mathbf{B}\|_F$ in problem (1) is similar to the atomic norm, but with slight differences: unlike the atomic norm, we set $\mathcal{A}$ to be the infinite set of unit-ℓ_2-norm matrices for $\mathsf{rank}(\mathbf{X}) \le r$. This leads to better sample complexity guarantees than the atomic-norm based methods.

Comparison with Our Work. Despite a large amount of works on matrix factorizations and their convex relaxation, few works study the connection between them. Hereby, we emphasize the difference between exact recoverability and tightness of convex relaxation: exact recoverability studies whether one algorithm can exactly recover the underlying matrix, while the tightness of convex relaxation characterizes the connection between non-convex problem and its convex counterpart. The standard proof technique only focuses on the former problem for both matrix completion and robust PCA. In contrast, this work explores the latter problem and bridges the gap between the non-convex matrix factorization and its convex counterpart.

3. Notations

We will use calligraphy to represent a set, bold capital letters to represent a matrix, bold lower-case letters to represent a vector, and lower-case letters to represent scalars. Specifically, we denote by $\mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2}$ the underlying matrix. We use $\mathbf{X}_{:t} \in \mathbb{R}^{n_1 \times 1}$ ($\mathbf{X}_{t:} \in \mathbb{R}^{1 \times n_2}$) to indicate the t-th column (row) of $\mathbf{X}$. The entry in the i-th row, j-th column of $\mathbf{X}$ is represented by $\mathbf{X}_{ij}$. The condition number of $\mathbf{X}$ is $\kappa = \sigma_1(\mathbf{X})/\sigma_r(\mathbf{X})$. We let $n_{(1)} = \max\{n_1, n_2\}$ and $n_{(2)} = \min\{n_1, n_2\}$. For SVD $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ of matrix $\mathbf{X}$, we define $\mathsf{svd}_r(\mathbf{X}) = \mathbf{U}_{:,1:r}\mathbf{\Sigma}_{(1:r):(1:r)}\mathbf{V}^\top_{:,1:r}$. For a function $H(\mathbf{M})$ on an input matrix $\mathbf{M}$, its conjugate function H^* is defined by $H^*(\mathbf{\Lambda}) = \max_{\mathbf{M}} \langle \mathbf{\Lambda}, \mathbf{M} \rangle - H(\mathbf{M})$. Furthermore, let H^{**} denote the conjugate function of H^*.

We will frequently use $\mathsf{rank}(\mathbf{X}) \leq r$ to constrain the rank of $\mathbf{X}$. This can be equivalently represented as $\mathbf{X} = \mathbf{A}\mathbf{B}$, by restricting the number of columns of $\mathbf{A}$ and rows of $\mathbf{B}$ to be r. For norms, denote by $\|\mathbf{X}\|_F = \sqrt{\sum_{ij} \mathbf{X}_{ij}^2}$ the Frobenius norm of matrix $\mathbf{X}$. Let $\sigma_1(\mathbf{X}) \geq \sigma_2(\mathbf{X}) \geq \ldots \geq \sigma_r(\mathbf{X})$ be the non-zero singular values of $\mathbf{X}$. The nuclear norm (a.k.a. trace norm) of $\mathbf{X}$ is defined by $\|\mathbf{X}\|_* = \sum_{i=1}^r \sigma_i(\mathbf{X})$, and the operator norm of $\mathbf{X}$ is $\|\mathbf{X}\| = \sigma_1(\mathbf{X})$. Denote by $\|\mathbf{X}\|_\infty = \max_{ij} |\mathbf{X}_{ij}|$. Define the r^*-norm to be $\|\mathbf{X}\|_{r*} = \max_{\mathbf{M}} \langle \mathbf{M}, \mathbf{X} \rangle - \frac{1}{2}\|\mathbf{M}\|_r^2$ where $\|\mathbf{M}\|_r^2 = \sum_{i=1}^r \sigma_i^2(\mathbf{M})$ is the sum of the first r largest squared singular values. For two matrices $\mathbf{A}$ and $\mathbf{B}$ of equal dimensions, we denote by $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{ij} \mathbf{A}_{ij}\mathbf{B}_{ij}$. We denote by $\partial H(\mathbf{X}) = \{\mathbf{\Lambda} \in \mathbb{R}^{n_1 \times n_2} : H(\mathbf{Y}) \geq H(\mathbf{X}) + \langle \mathbf{\Lambda}, \mathbf{Y} - \mathbf{X} \rangle \text{ for any } \mathbf{Y}\}$ the sub-differential of function H evaluated at $\mathbf{X}$. We define the indicator function of convex set $\mathcal{C}$ by $\mathbf{I}_{\mathcal{C}}(\mathbf{X}) = \begin{cases} 0, & \text{if } \mathbf{X} \in \mathcal{C}; \\ +\infty, & \text{otherwise.} \end{cases}$ For any non-empty set $\mathcal{C}$, denote by $\mathsf{cone}(\mathcal{C}) = \{t\mathbf{X} : \mathbf{X} \in \mathcal{C},\ t \geq 0\}$.

We denote by Ω the set of indices of observed entries, and $\Omega^\perp$ its complement. Without confusion, Ω also indicates the linear subspace formed by matrices with entries in $\Omega^\perp$ being 0. We denote by $\mathcal{P}_\Omega : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ the orthogonal projector to the subspace Ω. We will consider a single norm for these operators, namely, the operator norm denoted by $\|\mathcal{A}\|$ and defined by $\|\mathcal{A}\| = \sup_{\|\mathbf{X}\|_F = 1} \|\mathcal{A}(\mathbf{X})\|_F$. For any orthogonal projection operator $\mathcal{P}_\mathcal{T}$ to any subspace $\mathcal{T}$, we know that $\|\mathcal{P}_\mathcal{T}\| = 1$ whenever $\mathsf{dim}(\mathcal{T}) \neq 0$. For distributions, denote by $\mathcal{N}(0, 1)$ the standard Gaussian distribution, $\mathsf{Uniform}(m)$ the uniform distribution of cardinality m, and $\mathsf{Ber}(p)$ the Bernoulli distribution with success probability p.

4. Strong Duality of Matrix Factorizations: A New Analytical Framework

This section develops a novel framework to analyze non-convex matrix factorization problems. The framework can be applied to different specific problems and leads to nearly optimal sample complexity guarantees.

4.1. $\frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2$ Regularization

We first study the $\frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2$-regularized matrix factorization problem

$$(\mathbf{P}) \qquad \min_{\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{r \times n_2}} F(\mathbf{A}, \mathbf{B}) = H(\mathbf{A}\mathbf{B}) + \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2,\ H(\cdot) \text{ is convex and closed.}$$

We will show in this section that under suitable conditions the duality gap between (**P**) and its dual (bi-dual) problem is zero, so problem (**P**) can be converted to an equivalent convex problem.

First, we consider an easy case where $H(\mathbf{A}\mathbf{B}) = \frac{1}{2}\|\widehat{\mathbf{Y}}\|_F^2 - \langle \widehat{\mathbf{Y}}, \mathbf{A}\mathbf{B} \rangle$ for a fixed $\widehat{\mathbf{Y}}$, leading to the objective function $\frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{A}\mathbf{B}\|_F^2$. For this case, we establish the following lemma. Its proof is basically to calculate the gradient of $f(\mathbf{A}, \mathbf{B})$ and let it equal to zero (Srebro, 2004); see Appendix A for details.

Lemma 1 *For any given matrix $\widehat{\mathbf{Y}}$, any local minimum of $f(\mathbf{A}, \mathbf{B}) = \frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{A}\mathbf{B}\|_F^2$ is globally optimal, given by $\mathsf{svd}_r(\widehat{\mathbf{Y}})$. The objective function $f(\mathbf{A}, \mathbf{B})$ around any saddle point has a negative second-order directional curvature. Moreover, $f(\mathbf{A}, \mathbf{B})$ has no local maximum.*

Now we turn to the general case. Given Lemma 1, we can reduce $F(\mathbf{A}, \mathbf{B})$ to the form $\frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{A}\mathbf{B}\|_F^2$ for some $\widehat{\mathbf{Y}}$ plus an extra term:

$$\begin{aligned} F(\mathbf{A}, \mathbf{B}) &= \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2 + H(\mathbf{A}\mathbf{B}) \\ &= \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2 + H^{**}(\mathbf{A}\mathbf{B}) \\ &= \max_{\mathbf{\Lambda}} \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2 + \langle \mathbf{\Lambda}, \mathbf{A}\mathbf{B} \rangle - H^*(\mathbf{\Lambda}) \\ &= \max_{\mathbf{\Lambda}} \frac{1}{2}\| - \mathbf{\Lambda} - \mathbf{A}\mathbf{B}\|_F^2 - \frac{1}{2}\|\mathbf{\Lambda}\|_F^2 - H^*(\mathbf{\Lambda}) \\ &\triangleq \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}), \end{aligned} \tag{7}$$

where we define

$$L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) \triangleq \frac{1}{2}\| - \mathbf{\Lambda} - \mathbf{A}\mathbf{B}\|_F^2 - \frac{1}{2}\|\mathbf{\Lambda}\|_F^2 - H^*(\mathbf{\Lambda})$$

as the Lagrangian of problem (**P**),[2] and the second equality in Eqn. (7) holds because H is closed and convex with respect to the argument $\mathbf{A}\mathbf{B}$.

By Lemma 1 and the definition of $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$, for any fixed value of $\mathbf{\Lambda}$, any local minimum of $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ is globally optimal, because minimizing $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ is equivalent to minimizing $\frac{1}{2}\| - \mathbf{\Lambda} - \mathbf{A}\mathbf{B}\|_F^2$ for a fixed $\mathbf{\Lambda}$.

So the remaining part of our analysis is to choose a proper $\widetilde{\mathbf{\Lambda}}$ for a solution $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ of problem (**P**), such that $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$, which then implies strong duality. For this, we introduce the following condition.

Condition 1 *For a solution $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ to problem (**P**), there exists a $\widetilde{\mathbf{\Lambda}} \in \partial_{\mathbf{X}} H(\mathbf{X})|_{\mathbf{X} = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}}$ such that*

$$-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\widetilde{\mathbf{B}}^T = \widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{B}}^T \quad \textit{and} \quad \widetilde{\mathbf{A}}^T(-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) = \widetilde{\mathbf{A}}^T\widetilde{\mathbf{\Lambda}}. \tag{8}$$

Explanation of Condition 1. We note that $\nabla_{\mathbf{A}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = \mathbf{A}\mathbf{B}\mathbf{B}^T + \mathbf{\Lambda}\mathbf{B}^T$ and $\nabla_{\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = \mathbf{A}^T\mathbf{A}\mathbf{B} + \mathbf{A}^T\mathbf{\Lambda}$ for a fixed $\mathbf{\Lambda}$. In particular, if we set $\mathbf{\Lambda}$ to be the $\widetilde{\mathbf{\Lambda}}$ in (8), then $\nabla_{\mathbf{A}} L(\mathbf{A}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})|_{\mathbf{A} = \widetilde{\mathbf{A}}} = \mathbf{0}$ and $\nabla_{\mathbf{B}} L(\widetilde{\mathbf{A}}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})|_{\mathbf{B} = \widetilde{\mathbf{B}}} = \mathbf{0}$. So Condition 1 implies that $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ is either a saddle point or a local minimizer of $L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$ as a function of $(\mathbf{A}, \mathbf{B})$ for the fixed $\widetilde{\mathbf{\Lambda}}$. The following lemma states that if $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ is indeed a local minimizer, then $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ and strong duality holds.

2. One can easily check that $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = \min_{\mathbf{M}} L'(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{\Lambda})$, where $L'(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{\Lambda})$ is the Lagrangian of the constraint optimization problem $\min_{\mathbf{A}, \mathbf{B}, \mathbf{M}} \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2 + H(\mathbf{M}),\ \text{s.t. } \mathbf{M} = \mathbf{A}\mathbf{B}$. With a little abuse of notation, we call $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ the Lagrangian of the unconstrained problem (**P**) as well.

Lemma 2 (Dual Certificate) *Let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ be a solution to problem* (**P**)*. If there exists a dual certificate $\widetilde{\mathbf{\Lambda}}$ satisfying Condition 1 and the pair $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ is a local minimizer of $L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$ for the fixed $\widetilde{\mathbf{\Lambda}}$, then strong duality holds. Moreover, we have the relation $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}})$.*

Proof We begin by showing that $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$. By the assumption of the lemma, $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ is a local minimizer of $L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}) = \frac{1}{2}\| - \widetilde{\mathbf{\Lambda}} - \mathbf{A}\mathbf{B}\|_F^2 + c(\widetilde{\mathbf{\Lambda}})$, where $c(\widetilde{\mathbf{\Lambda}})$ is a function that is independent of $\mathbf{A}$ and $\mathbf{B}$. So according to Lemma 1, $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$, namely, $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ globally minimizes $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ when $\mathbf{\Lambda}$ is fixed to $\widetilde{\mathbf{\Lambda}}$. Furthermore, $\widetilde{\mathbf{\Lambda}} \in \partial_{\mathbf{X}} H(\mathbf{X})|_{\mathbf{X}=\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}}$ implies that $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} \in \partial_{\mathbf{\Lambda}} H^*(\mathbf{\Lambda})|_{\mathbf{\Lambda}=\widetilde{\mathbf{\Lambda}}}$ by the convexity of function H, meaning that $\mathbf{0} \in \partial_{\mathbf{\Lambda}} L(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \mathbf{\Lambda})$. So $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \mathbf{\Lambda})$ due to the concavity of $L(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \mathbf{\Lambda})$ w.r.t. variable $\mathbf{\Lambda}$. Thus $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$.

Now we prove the strong duality. By the fact that $F(\mathbf{A}, \mathbf{B}) = \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ and that $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \mathbf{\Lambda})$, we have

$$F(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = L(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{\Lambda}}) \leq L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}), \quad \forall \mathbf{A}, \mathbf{B}.$$

where the inequality holds because $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of L. So on the one hand, we have

$$\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = F(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) \leq \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}) \leq \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}).$$

On the other hand, by weak duality,

$$\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) \geq \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}).$$

Therefore, $\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$, i.e., strong duality holds.

Finally,

$$\begin{aligned}
\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} &= \operatorname*{argmin}_{\mathbf{A}\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}) \\
&= \operatorname*{argmin}_{\mathbf{A}\mathbf{B}} \frac{1}{2}\| - \widetilde{\mathbf{\Lambda}} - \mathbf{A}\mathbf{B}\|_F^2 - \frac{1}{2}\|\widetilde{\mathbf{\Lambda}}\|_F^2 - H^*(\widetilde{\mathbf{\Lambda}}) \\
&= \operatorname*{argmin}_{\mathbf{A}\mathbf{B}} \frac{1}{2}\| - \widetilde{\mathbf{\Lambda}} - \mathbf{A}\mathbf{B}\|_F^2 \\
&= \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}}),
\end{aligned}$$

as desired. ■

This lemma then leads to the following theorem.

Theorem 3 *Let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ denote an optimal solution of problem* (**P**)*. Define a matrix space*

$$\mathcal{T} \triangleq \{\widetilde{\mathbf{A}}\mathbf{X}^T + \mathbf{Y}\widetilde{\mathbf{B}}, \ \mathbf{X} \in \mathbb{R}^{n_2 \times r}, \ \mathbf{Y} \in \mathbb{R}^{n_1 \times r}\}. \tag{9}$$

Then strong duality holds for problem (**P**)*, provided that*

$$(1)\ \widetilde{\mathbf{\Lambda}} \in \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \triangleq \Psi, \qquad (2)\ \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}, \qquad (3)\ \|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}). \tag{10}$$

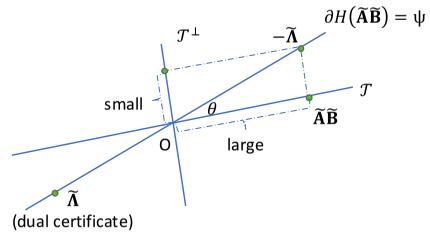


Figure 4: Geometry of dual condition (10) for general matrix factorization problems.

Proof The proof idea is to construct a dual certificate $\widetilde{\mathbf{\Lambda}}$ so that the conditions in Lemma 2 hold. It suffices for $\widetilde{\mathbf{\Lambda}}$ to satisfy the following:

$$\begin{aligned}
&(a) \quad \widetilde{\mathbf{\Lambda}} \in \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}), && \text{(by Condition 1)} \\
&(b) \quad (\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}})\widetilde{\mathbf{B}}^T = \mathbf{0} \quad \text{and} \quad \widetilde{\mathbf{A}}^T(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}}) = \mathbf{0}, && \text{(by Condition 1)} \\
&(c) \quad \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}}). && \text{(by the local minimizer assumption and Lemma 1)}
\end{aligned} \tag{11}$$

It turns out that for any matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, $\mathcal{P}_{\mathcal{T}^\perp}\mathbf{M} = (\mathbf{I} - \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^\dagger)\mathbf{M}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{B}}^\dagger)$ and so $\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{M}\| \leq \|\mathbf{M}\|$, a fact that we will frequently use in the sequel. Denote by $\mathcal{U}$ the left singular space of $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$ and $\mathcal{V}$ the right singular space. Then the linear space $\mathcal{T}$ can be equivalently represented as $\mathcal{T} = \mathcal{U} + \mathcal{V}$. Therefore, $\mathcal{T}^\perp = (\mathcal{U} + \mathcal{V})^\perp = \mathcal{U}^\perp \cap \mathcal{V}^\perp$.

With this, we will show the conditions in (11) are equivalent to those in (10). Condition (b) implies $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}} \in \mathsf{Null}(\widetilde{\mathbf{A}}^T) = \mathsf{Col}(\widetilde{\mathbf{A}})^\perp$ and $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}} \in \mathsf{Row}(\widetilde{\mathbf{B}})^\perp$ (so $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \widetilde{\mathbf{\Lambda}} \in \mathcal{T}^\perp$), and vice versa. Condition (c) $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}})$ implies that for an orthogonal decomposition $-\widetilde{\mathbf{\Lambda}} = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} + \mathbf{E}$, where $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} \in \mathcal{T}$, and $\mathbf{E} \in \mathcal{T}^\perp$, we have $\|\mathbf{E}\| < \sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$. Conversely, $\|\mathbf{E}\| < \sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$ and condition (b) imply $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathsf{svd}_r(-\widetilde{\mathbf{\Lambda}})$. Therefore, the conditions in (10) are equivalent to the conditions in (11), which imply the conditions in Lemma 2 and give strong duality as desired. ■

To show the dual condition in Theorem 3, intuitively, we need to show that the angle θ between the subspaces $\mathcal{T}$ and Ψ is small (see Figure 4) for a specific function $H(\cdot)$. In Section 5, we will demonstrate applications that, with randomness, obey this dual condition with high probability.

4.2. $\frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2$ Regularization

We now study another class of matrix factorization problems

$$(\mathbf{P'}) \qquad \min_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} F(\mathbf{A}, \mathbf{B}) = H(\mathbf{A}\mathbf{B}) + \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2, \ H(\cdot) \text{ is convex and closed.}$$

The analysis is similar to the analysis in Section 4.1. To see this, we first note that $\frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2$ has a natural variational form of the nuclear norm:

$$\|\mathbf{A}\mathbf{B}\|_* = \min_{\mathbf{A}'\in\mathbb{R}^{n_1\times r}, \mathbf{B}'\in\mathbb{R}^{r\times n_2}, \mathbf{A}'\mathbf{B}'=\mathbf{A}\mathbf{B}} \frac{1}{2}\|\mathbf{A}'\|_F^2 + \frac{1}{2}\|\mathbf{B}'\|_F^2.$$

Therefore, problem **(P')** is equivalent to

$$\textbf{(P'')} \qquad (\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname*{argmin}_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} F(\mathbf{A},\mathbf{B}) = H(\mathbf{AB}) + \|\mathbf{AB}\|_*, \ H(\cdot) \text{ is convex and closed.}$$

Thus we can analyze problem **(P'')** as in Section 4.1. More specifically, we have the following result.

Theorem 4 *Let $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$ denote an optimal solution of problem* **(P'')**. *Let $\mathbf{U\Sigma V}^\top$ denote the skinny SVD of $\bar{\mathbf{A}}\bar{\mathbf{B}}$. Define a matrix space*

$$\mathcal{T} \triangleq \{\mathbf{UX}^T + \mathbf{YV}^T, \ \mathbf{X}\in\mathbb{R}^{n_2\times r}, \ \mathbf{Y}\in\mathbb{R}^{n_1\times r}\}. \tag{12}$$

If there exists a dual certificate $\widetilde{\mathbf{\Lambda}}$ such that

$$\begin{aligned} &(a) \qquad \widetilde{\mathbf{\Lambda}} \in \partial H(\bar{\mathbf{A}}\bar{\mathbf{B}}) \triangleq \Psi, \\ &(b) \qquad \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \mathbf{UV}^\top, \\ &(c) \qquad \|\mathcal{P}_{\mathcal{T}^\perp}(-\widetilde{\mathbf{\Lambda}})\| < \frac{1}{2}, \end{aligned} \tag{13}$$

then strong duality holds for problems **(P'')** *and* **(P')**.

Remark 5 *There is a key difference between Theorem 4 and the standard optimality result by Candès and Recht (2009). On one hand, Theorem 4 shows that under certain conditions, the non-convex matrix completion and its convex counterpart share a common global optimality, i.e., the strong duality. On the other hand, the standard optimality result in Lemma 3.1 of (Candès and Recht, 2009) studies the conditions under which the convex matrix completion exactly recovers the underlying low-rank matrix. Their lemma does not concern about the non-convex matrix completion formulation.*

Proof We note that

$$\begin{aligned} F(\mathbf{A},\mathbf{B}) &= \|\mathbf{AB}\|_* + H(\mathbf{AB}) \\ &= \|\mathbf{AB}\|_* + H^{**}(\mathbf{AB}) \\ &= \|\mathbf{AB}\|_* + \max_{\mathbf{\Lambda}\in\mathbb{R}^{n_1\times n_2}} \langle \mathbf{\Lambda}, \mathbf{AB}\rangle - H^*(\mathbf{\Lambda}) \\ &= \max_{\mathbf{\Lambda}\in\mathbb{R}^{n_1\times n_2}} \|\mathbf{AB}\|_* + \langle \mathbf{\Lambda}, \mathbf{AB}\rangle - H^*(\mathbf{\Lambda}) \\ &= \max_{\mathbf{\Lambda}\in\mathbb{R}^{n_1\times n_2}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda}), \end{aligned}$$

where the second equality holds because $H(\cdot)$ is closed and convex with respect to the argument $\mathbf{AB}$ and the third equality holds by the definition of conjugate function. To prove Theorem 4, we need the following two claims.

Claim 1 $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$.

Proof We only need to show $\mathbf{0} \in \partial_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}}) = \bar{\mathbf{A}}\bar{\mathbf{B}} - \partial_{\mathbf{\Lambda}} H^*(\widetilde{\mathbf{\Lambda}})$, because $L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$ is a concave function of $\mathbf{\Lambda}$ for the fixed $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$. Note that $\bar{\mathbf{A}}\bar{\mathbf{B}} \in \partial_{\mathbf{\Lambda}} H^*(\widetilde{\mathbf{\Lambda}})$ is implied by $\widetilde{\mathbf{\Lambda}} \in \partial H(\bar{\mathbf{A}}\bar{\mathbf{B}})$ by the convexity of function $H(\cdot)$. Therefore, condition (a) immediately implies $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$. ■

Claim 2 $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$.

Proof First, we have

$$\begin{aligned} \operatorname*{argmin}_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}}) &= \operatorname*{argmin}_{\mathbf{A},\mathbf{B}} \|\mathbf{AB}\|_* - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{AB}\rangle - H^*(\widetilde{\mathbf{\Lambda}}) \\ &= \operatorname*{argmin}_{\mathbf{A},\mathbf{B}} \|\mathbf{AB}\|_* - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{AB}\rangle \\ &= \operatorname*{argmin}_{\mathbf{A},\mathbf{B}} \sum_{i=1}^r \sigma_i(\mathbf{AB}) - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{AB}\rangle, \end{aligned}$$

where the first step follows by definition of $L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$, the second step follows since $\mathbf{\Lambda}$ is fixed, the third step follows by definition of nuclear norm.

Using conditions (b) and (c) in (13), $-\widetilde{\mathbf{\Lambda}}$ can be rewritten as

$$-\widetilde{\mathbf{\Lambda}} = \underbrace{\mathbf{UV}^T}_{\text{in space } \mathcal{T} \text{ with eigenvalues } 1} + \underbrace{\mathcal{P}_{\mathcal{T}^\perp}(-\widetilde{\mathbf{\Lambda}})}_{\text{in space } \mathcal{T}^\perp \text{ with eigenvalues } <1/2},$$

which implies $\forall i \in [r], \sigma_i(-\widetilde{\mathbf{\Lambda}}) = 1$.

Using von Neumann's trace inequality (see Lemma 25), we can show

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^r \sigma_i(\mathbf{AB}) - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{AB}\rangle &\geq \min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^r \sigma_i(\mathbf{AB}) - \sum_{i=1}^r \sigma_i(-\widetilde{\mathbf{\Lambda}})\sigma_i(\mathbf{AB}) \\ &= \min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^r \sigma_i(\mathbf{AB}) - \sum_{i=1}^r \sigma_i(\mathbf{AB}) \qquad \text{by } \sigma_i(-\mathbf{\Lambda}) = 1, \forall i \in [r] \\ &= 0. \end{aligned}$$

Therefore, on the one hand, we already have

$$\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}}) \geq -H^*(\widetilde{\mathbf{\Lambda}}).$$

On the other hand, according to definition of $\bar{\mathbf{A}}, \bar{\mathbf{B}}$, and $-\widetilde{\mathbf{\Lambda}}$,

$$\begin{aligned} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}}) &= \|\bar{\mathbf{A}}\bar{\mathbf{B}}\|_* - \langle -\widetilde{\mathbf{\Lambda}}, \bar{\mathbf{A}}\bar{\mathbf{B}}\rangle - H^*(\widetilde{\mathbf{\Lambda}}) \\ &= -H^*(\widetilde{\mathbf{\Lambda}}). \end{aligned}$$

Thus, we can conclude that $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$. ■

Claim 1 and Claim 2 together show that $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of the Lagrangian $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$.

We now prove the strong duality. By the fact that $F(\mathbf{A},\mathbf{B}) = \max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$ and that $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$, we have

$$F(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}}) \leq L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}), \ \ \forall \mathbf{A}, \mathbf{B},$$

where the inequality holds because $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\cdot,\cdot,\cdot)$. So on the one hand, we have

$$\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda}) = F(\bar{\mathbf{A}}, \bar{\mathbf{B}}) \leq \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\widetilde{\mathbf{\Lambda}}) \leq \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda}).$$

On the other hand, by weak duality,

$$\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda}) \geq \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda}).$$

Therefore, $\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda}) = \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\mathbf{\Lambda})$, i.e., strong duality holds. ■

Similarly as for Theorem 3, to show the dual condition in Theorem 4, we need to show that the angle θ between the subspaces $\mathcal{T}$ and Ψ is small for a specific function $H(\cdot)$. We will also demonstrate how randomness implies this dual condition with high probability in concrete applications.

5. Matrix Completion

In matrix completion, there is a hidden matrix $\mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2}$ with rank at most r. We are given measurements $\{\mathbf{X}^*_{ij} : (i,j) \in \Omega\}$, where $\Omega \sim \mathsf{Uniform}(m)$, i.e., Ω is sampled uniformly at random from all subsets of $[n_1] \times [n_2]$ of cardinality m. The goal is to exactly recover $\mathbf{X}^*$ with high probability. Here we apply our unified framework in Section 4 to matrix completion, by setting $H(\cdot) = \mathbf{I}_{\{\mathbf{M}:\mathcal{P}_\Omega(\mathbf{M})=\mathcal{P}_\Omega(\mathbf{X}^*)\}}(\cdot)$. (Recall that $\mathbf{I}_{\mathcal{C}}$ is the indicator function of the set $\mathcal{C}$.)

A quantity governing the difficulties of matrix completion is the incoherence parameter μ. Intuitively, matrix completion is possible only if the information spreads evenly throughout the low-rank matrix. This intuition is captured by the incoherence conditions. Formally, denote by $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ the skinny SVD of a fixed $n_1 \times n_2$ matrix $\mathbf{X}$ of rank r. The μ-incoherence condition (5) was introduced for the low-rank matrix $\mathbf{X}$ (Candès et al., 2011; Candès and Recht, 2009; Recht, 2011; Zhang et al., 2016). For this condition, it can be shown that $1 \leq \mu \leq \frac{n_{(1)}}{r}$. It holds for many random matrices with incoherence parameter μ about $\sqrt{r \log n_{(1)}}$ (Keshavan et al., 2010a).

We have two positive results for matrix completion, which are achieved via efficient algorithms by applying Theorem 3 and Theorem 4, respectively.

Before proceeding, we first cite a lower bound from prior work. It shows that our two positive results are nearly optimal.

Theorem 6 (Information-Theoretic Lower Bound. Candès and Tao (2010), Theorem 1.7) *Denote by $\Omega \sim \mathsf{Uniform}(m)$ the support set uniformly distributed among all sets of cardinality m. Suppose that $m \leq c\mu n_{(1)} r \log n_{(1)}$ for an absolute constant c. Then there exist infinitely many $n_1 \times n_2$ matrices $\mathbf{X}'$ of rank at most r obeying μ-incoherence* (5) *such that $\mathcal{P}_\Omega(\mathbf{X}') = \mathcal{P}_\Omega(\mathbf{X}^*)$, with probability at least $1 - n_{(1)}^{-10}$.*

Our first positive result converts a non-convex rank-constrained problem to a convex optimization problem, which can be *efficiently* solved. The proof is simply by applying Theorem 3, so the details are deferred to Appendix D.

Theorem 7 (Efficient Matrix Completion I) *Let $\Omega \sim \mathsf{Uniform}(m)$ be the support set uniformly distributed among all sets of cardinality m. Denote the condition number of matrix $\mathbf{X}^*$ as $\kappa = \sigma_1(\mathbf{X}^*)/\sigma_r(\mathbf{X}^*)$. Let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) \in \mathbb{R}^{n_1 \times r} \times \mathbb{R}^{r \times n_2}$ denote the output of the non-convex matrix factorization problem*

$$\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{A}\mathbf{B}) = \mathcal{P}_\Omega(\mathbf{X}^*). \tag{14}$$

Then there are absolute constants c and c_0 such that with probability at least $1 - cn^{-10}$, the strong duality $\mathbf{X}^* = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \widetilde{\mathbf{X}}$ holds, provided that $m \geq c\kappa^2 \mu r n_{(1)} \log_{2\kappa}(n_{(1)}) \log(n_{(1)})$ and $\mathbf{X}^*$ obeys μ-incoherence (5)*, where $\widetilde{\mathbf{X}}$ is the output of the nuclear norm minimization*

$$\min_{\mathbf{X}\in\mathbb{R}^{n_1\times n_2}} \|\mathbf{X}\|_{r*}, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*). \tag{15}$$

Proof Sketch. We first show that the problem

$$(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \operatorname*{argmin}_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{A}\mathbf{B}) = \mathcal{P}_\Omega(\mathbf{X}^*),$$

exactly recovers $\mathbf{X}^*$, i.e., $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$, with the optimal sample complexity (see Appendix D). So if strong duality holds, this non-convex optimization problem can be equivalently converted to the convex program (15), proving Theorem 7.

It now suffices to apply our unified framework in Section 4 to prove the strong duality. We show that the dual condition in Theorem 3 holds with high probability by the following arguments. Let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ be a global solution to problem (15). For $H(\mathbf{X}) = \mathbf{I}_{\{\mathbf{M}\in\mathbb{R}^{n_1\times n_2}:\ \mathcal{P}_\Omega\mathbf{M}=\mathcal{P}_\Omega\mathbf{X}^*\}}(\mathbf{X})$, we have

$$\begin{aligned}\Psi = \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) &= \{\mathbf{G} \in \mathbb{R}^{n_1\times n_2} : \langle \mathbf{G}, \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\rangle \geq \langle \mathbf{G}, \mathbf{Y}\rangle, \text{ for any } \mathbf{Y} \in \mathbb{R}^{n_1\times n_2} \text{ s.t. } \mathcal{P}_\Omega\mathbf{Y} = \mathcal{P}_\Omega\mathbf{X}^*\} \\ &= \{\mathbf{G} \in \mathbb{R}^{n_1\times n_2} : \langle \mathbf{G}, \mathbf{X}^*\rangle \geq \langle \mathbf{G}, \mathbf{Y}\rangle, \text{ for any } \mathbf{Y} \in \mathbb{R}^{n_1\times n_2} \text{ s.t. } \mathcal{P}_\Omega\mathbf{Y} = \mathcal{P}_\Omega\mathbf{X}^*\} = \Omega,\end{aligned}$$

where the third equality holds since $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$. Then we only need to show

$$(1)\ \widetilde{\mathbf{\Lambda}} \in \Omega, \qquad (2)\ \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}, \qquad (3)\ \|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \frac{2}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}). \tag{16}$$

It is interesting to see that dual condition (16) can be satisfied if the angle θ between subspace Ω and subspace $\mathcal{T}$ is very small; see Figure 4. When the sample size $|\Omega|$ becomes larger, the angle θ becomes smaller (e.g., when $|\Omega| = n_1 n_2$, the angle θ is zero as $\Omega = \mathbb{R}^{n_1\times n_2}$). We show that the sample size $m \geq \Omega(\kappa^2 \mu r n_{(1)} \log_{2\kappa}(n_{(1)}) \log(n_{(1)}))$ is a sufficient condition for condition (16) to hold. □

Our second result further improves the sample complexity of Theorem 7 by applying Theorem 4. The reduced sample complexity matches the best known result, which was achieved by nuclear norm minimization. However, the strong duality is new here, which illustrates the reason why nuclear norm works for matrix completion from another viewpoint of non-convex optimization. The proof is deferred to Appendix E.

Theorem 8 (Efficient Matrix Completion II) *Let $\Omega \sim \mathsf{Uniform}(m)$ be the support set uniformly distributed among all sets of cardinality m. Let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) \in \mathbb{R}^{n_1\times r} \times \mathbb{R}^{r\times n_2}$ denote the output of the non-convex matrix factorization problem*

$$\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{A}\mathbf{B}) = \mathcal{P}_\Omega(\mathbf{X}^*), \tag{17}$$

Then there are absolute constants c and c_0 such that with probability at least $1 - cn^{-10}$, the strong duality $\mathbf{X}^* = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \widetilde{\mathbf{X}}$ holds, provided that $m \geq c_0 \mu r n_{(1)} \log^2 n_{(1)}$ and $\mathbf{X}^*$ obeys μ-incoherence (5)*, where $\widetilde{\mathbf{X}}$ is the output of the nuclear norm minimization*

$$\min_{\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}} \|\mathbf{X}\|_*, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*). \tag{18}$$

6. Robust Principal Component Analysis

This section develops our theory for robust PCA based on our framework. In the problem of robust PCA, we are given an observed matrix of the form $\mathbf{D} = \mathbf{X}^* + \mathbf{S}^*$, where $\mathbf{X}^*$ is the ground-truth matrix which is incoherent and low rank, and $\mathbf{S}^*$ is the corruption matrix which is sparse. The goal is to recover the hidden matrices $\mathbf{X}^*$ and $\mathbf{S}^*$ from the observation $\mathbf{D}$. We set $H(\mathbf{X}) = \lambda \|\mathbf{D} - \mathbf{X}\|_1$.

To make the information spreads evenly throughout the matrix, the matrix cannot have one entry whose absolute value is significantly larger than other entries. Candès et al. (2011) introduced an extra incoherence condition (Recall that $\mathbf{X}^* = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the skinny SVD of $\mathbf{X}^*$)

$$\|\mathbf{U}\mathbf{V}^T\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}. \tag{19}$$

In this work, we make the following incoherence assumption for robust PCA instead of (19):

$$\|\mathbf{X}^*\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}} \sigma_r(\mathbf{X}^*). \tag{20}$$

Note that condition (20) is very similar to the incoherence condition (19) for the robust PCA problem, but the two notions are not directly comparable. On the other hand, the condition (20) has an intuitive explanation, namely, that the entries must scatter almost uniformly across the low-rank matrix.

We have the following results for robust PCA.

Theorem 9 (Robust PCA) *Suppose $\mathbf{X}^*$ is an $n_1 \times n_2$ matrix of rank r, and obeys incoherence* (5) *and* (20)*. Assume that the support set Ω of $\mathbf{S}^*$ is uniformly distributed among all sets of cardinality m. Then with probability at least $1 - cn_{(1)}^{-10}$, the output of the optimization problem*

$$(\widetilde{\mathbf{X}}, \widetilde{\mathbf{S}}) = \operatorname*{argmin}_{\mathbf{X},\mathbf{S}} \|\mathbf{X}\|_{r*} + \lambda \|\mathbf{S}\|_1, \quad \text{s.t.} \quad \mathbf{D} = \mathbf{X} + \mathbf{S},$$

with $\lambda = \frac{\sigma_r(\mathbf{X}^*)}{\sqrt{n_{(1)}}}$ is exact, namely, $\widetilde{\mathbf{X}} = \mathbf{X}^*$ and $\widetilde{\mathbf{S}} = \mathbf{S}^*$, provided that $\mathsf{rank}(\mathbf{X}^*) \leq \rho_r \frac{n_{(2)}}{\mu \log^2 n_{(1)}}$ and $m \leq \rho_s n_1 n_2$, where c, ρ_r, and ρ_s are all positive absolute constants, and function $\|\cdot\|_{r*}$ is given by (21)*.*

The bounds on the rank of $\mathbf{X}^*$ and the sparsity of $\mathbf{S}^*$ in Theorem 9 match the best known results for robust PCA in prior work when we assume the support set of $\mathbf{S}^*$ is sampled uniformly (Candès et al., 2011).

7. Computational Aspects

Computational Efficiency. We discuss our computational efficiency given that we have strong duality. We note that the dual and bi-dual of primal problem (**P**) are given by (see Appendix H.1)

$$\begin{aligned} &(\textbf{Dual, D1}) \quad \max_{\mathbf{\Lambda} \in \mathbb{R}^{n_1 \times n_2}} -H^*(\mathbf{\Lambda}) - \frac{1}{2}\|\mathbf{\Lambda}\|_r^2, \quad \text{where } \|\mathbf{\Lambda}\|_r^2 = \sum_{i=1}^r \sigma_i^2(\mathbf{\Lambda}), \\ &(\textbf{Bi-Dual, D2}) \quad \min_{\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}} H(\mathbf{M}) + \|\mathbf{M}\|_{r*}, \quad \text{where } \|\mathbf{M}\|_{r*} = \max_{\mathbf{X}} \langle \mathbf{M}, \mathbf{X} \rangle - \frac{1}{2}\|\mathbf{X}\|_r^2. \end{aligned} \tag{21}$$

and the dual and bi-dual of primal problem (**P''**) are given by (see Appendix H.2)

$$\begin{aligned} &(\textbf{Dual, D1''}) \quad \max_{\|\mathbf{\Lambda}\| \leq 1} -H^*(\mathbf{\Lambda}), \\ &(\textbf{Bi-Dual, D2''}) \quad \min_{\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}} H(\mathbf{M}) + \|\mathbf{M}\|_*. \end{aligned}$$

Problems (**D1**), (**D1''**) and (**D2**), (**D2''**) can be solved efficiently due to their convexity. In particular, Grussler et al. (2016) provided a computationally efficient algorithm to compute the proximal operators of functions $\frac{1}{2}\|\cdot\|_r^2$ and $\|\cdot\|_{r*}$. Hence, there exist algorithms that can find global minimum up to an ϵ error in function value in time $\mathsf{poly}(1/\epsilon)$, e.g., the Douglas-Rachford algorithm (He and Yuan, 2012).

Computational Lower Bounds. Unfortunately, strong duality does not always hold for general non-convex problems (**P**) (and (**P'**)). Here we present a very strong lower bound based on the random 4-SAT hypothesis. This is by now a fairly standard conjecture in complexity theory (Feige, 2002) and gives us constant factor inapproximability of problem (**P**) (and (**P'**)) for deterministic algorithms, even those running in exponential time.

If we additionally assume that $\mathsf{BPP} = \mathsf{P}$, where BPP is the class of problems which can be solved in probabilistic polynomial time, and P is the class of problems which can be solved in deterministic polynomial time, then the same conclusion holds for randomized algorithms. This is also a standard conjecture in complexity theory, as it is implied by the existence of certain strong pseudo-random generators or if any problem in deterministic exponential time has exponential size circuits (Impagliazzo and Wigderson, 1997). Therefore, any sub-exponential time algorithm achieving a sufficiently small constant factor approximation to problem (**P**) (and (**P'**)) in general would imply a major breakthrough in complexity theory.

The lower bound is proved by a reduction from the Maximum Edge Bi-clique problem (Ambühl et al., 2011). The details are presented in Appendix G.

Theorem 10 (Computational Lower Bound) *Assume the hardness of Random 4-SAT (See Conjecture 32 in Appendix G). Then there exists an absolute constant $\epsilon_0 > 0$ for which any deterministic algorithm achieving $(1+\epsilon)\mathsf{OPT}$ in the objective function value for problem* (**P**) *(and* (**P'**)*) with $\epsilon \leq \epsilon_0$, requires $2^{\Omega(n_1+n_2)}$ time, where OPT is the optimum. If in addition, $\mathsf{BPP} = \mathsf{P}$, then the same conclusion holds for randomized algorithms succeeding with probability at least $2/3$.*

It is clear that the bi-dual is typically solvable in polynomial time. Thus, the hardness result implies that (a) under certain circumstances, strong duality does not hold, and (b) the original problem is hard, while the bi-dual is easy.

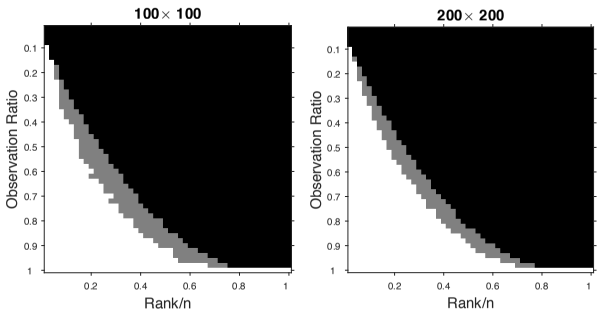


Figure 5: Exact recoverability of matrix completion with varying ranks and sample sizes. **White Region:** nuclear norm minimization succeeds. **White and Gray Regions:** r^* minimization succeeds. **Black Region:** both algorithms fail. It shows that the success region of r^* minimization slightly contains that of the nuclear minimization method.

8. Experiments

In this section, we will present our experimental results.

8.1. Experiments on Synthetic Data

We verify the exact recoverability of the r^* minimization (15) and the nuclear norm minimization (18) on the matrix completion problem by experiments on the synthetic data. The synthetic data are generated as follows. We construct the ground-truth matrix $\mathbf{X}^* = \mathbf{AB}$ as a product of matrices $\mathbf{A}$ of size $n \times r$ and $\mathbf{B}$ of size $r \times n$, whose entries are i.i.d. $\mathcal{N}(0, 1)$. We then uniformly sample m entries from $\mathbf{X}^*$ as the observations. For each size of the problem ($\mathbf{X}^*$ is 100×100 or 200×200), we test with different rank ratios r/n and observation ratios m/n^2. Each set of parameters is run 5 times, and the algorithm is said to succeed if $\|\widetilde{\mathbf{X}} - \mathbf{X}^*\|_F / \|\mathbf{X}^*\|_F \leq 10^{-3}$ for all five experiments, where $\widetilde{\mathbf{X}}$ is the output of the algorithms. We set the parameter r in r^* minimization (15) as the true rank, and use the Augmented Lagrange Multiplier Method (Chen et al., 2009) for optimization, where the proximal map of r^* norm is computed as in (Grussler et al., 2016).

The two figures in Figure 5 plots the fraction of exact recoveries: the white region represents the exact recovery by nuclear norm minimization, the white+gray region represents the exact recovery by r^* minimization (15), and the black region indicates the failure for both algorithms. It is clear that both algorithms succeed for a wide range of parameters. The success region of r^* minimization is slightly larger and contains the success region of the nuclear norm minimization for both 100×100 and 200×200 matrix completion problems.

8.2. Experiments on Real Data

To verify the performance of the algorithms on real data, we conduct experiments on the Hopkins 155 data set. This data set consists of 155 tasks/matrices, each of which consists of multiple data points

		$m = 0.05n_1n_2$		$m = 0.1n_1n_2$	
#Task	Size	Nuclear	r^*	Nuclear	r^*
Average over all 155 tasks	–	0.8249	0.8114	0.5689	0.5409
#1	59×459	0.7438	0.5948	0.5115	0.5117
#2	49×482	0.8235	0.6564	0.6371	0.5919
#3	49×153	0.7803	0.9174	0.5386	0.5386
#4	49×379	0.8500	0.9583	0.7287	0.7691
#5	49×432	0.8174	0.6353	0.4476	0.4477

Table 2: Relative error by matrix completion algorithms on the Hopkins 155 data set.

drawn from 2 or 3 moving objects. The trajectory of each object lies in a low-dimensional subspace, so the matrix for each task is supposed to be approximately low rank. We uniformly sample m entries from the matrix as our observations and run the matrix completion algorithms. The parameter r in the r^* minimization is set as the number of moving objects which is known to us in the data set.

Table 8.2 shows the the relative errors $\|\widehat{\mathbf{X}} - \mathbf{X}^*\|_F / \|\mathbf{X}^*\|_F$ of the nuclear norm minimization (18) and r^* minimization (15). On average, r^* minimization slightly outperforms the competitor, while sometimes the nuclear norm minimization is better. Table 8.2 also shows the errors on the first five tasks in the data set. It shows that when the number of observations is relatively large (10% observations or higher), the performance of the two algorithms are competitive to each other. When the number of observations is small (5% observed entries), there is a larger variance, but on average r^* minimization has an slight advantage.

9. Conclusions

This paper studied the strong duality of non-convex matrix factorization problems. It was shown that under certain dual conditions, a wide class of non-convex matrix factorization problems and their dual have the same optimum. This strong duality phenomenon is rarely discovered in the previous work. Hardness results were provided to show that strong duality is impossible without the randomness of sampling. The analytical framework may be of independent interest to non-convex optimization more broadly.

The proposed framework was applied to two prototypical matrix factorization problems in the machine learning community: matrix completion and robust PCA. This gave several efficient algorithms with nearly optimal sample complexity bounds which match the best-known previous results and also illustrated why the nuclear norm technique works from the new viewpoint of non-convex optimization.

Acknowledgments

We thank the anonymous reviewers for valuable comments, and Rina Foygel Barber, Rong Ge, Jason D. Lee, Zhouchen Lin, Guangcan Liu, Tengyu Ma, Benjamin Recht and Tuo Zhao for useful discussions. This work was supported in part by NSF grants NSF CCF-1422910, NSF CCF-1535967, NSF CCF-1451177, NSF IIS-1618714, NSF CCF-1527371, a Sloan Research Fellowship, a Microsoft Research Faculty Fellowship, DMS-1317308, Simons Investigator Award, Simons Collaboration

Grant, and ONR-N00014-16-1-2329. We would like to also thank Christian Grussler, Anders Rantzer, and Pontus Giselsson who kindly provided the code for computing the proximal map of the r^* function.

A. Proof of Lemma 1

Lemma 1 (Restated). *For any given matrix $\widehat{\mathbf{Y}}$, any local minimum of $f(\mathbf{A},\mathbf{B}) = \frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{AB}\|_F^2$ is globally optimal, given by $\mathsf{svd}_r(\widehat{\mathbf{Y}})$. The objective function $f(\mathbf{A},\mathbf{B})$ around any saddle point has a negative second-order directional curvature. Moreover, $f(\mathbf{A},\mathbf{B})$ has no local maximum.*

Proof We have that $(\mathbf{A},\mathbf{B})$ is a critical point of $f(\mathbf{A},\mathbf{B})$ if and only if $\nabla_{\mathbf{A}} f(\mathbf{A},\mathbf{B}) = \mathbf{0}$ and $\nabla_{\mathbf{B}} f(\mathbf{A},\mathbf{B}) = \mathbf{0}$, or equivalently,

$$\mathbf{ABB}^T = \widehat{\mathbf{Y}}\mathbf{B}^T \quad \text{and} \quad \mathbf{A}^T\mathbf{AB} = \mathbf{A}^T\widehat{\mathbf{Y}}. \tag{22}$$

Note that for any fixed matrix $\mathbf{A}$ (respectively $\mathbf{B}$), the function $f(\mathbf{A},\mathbf{B})$ is convex in the coefficients of $\mathbf{B}$ (respectively $\mathbf{A}$).

To prove the desired lemma, we need the following claim.

Claim 3 *If two matrices $\mathbf{A}$ and $\mathbf{B}$ define a critical point of $f(\mathbf{A},\mathbf{B})$, then the global mapping $\mathbf{M} = \mathbf{AB}$ is of the form*

$$\mathbf{M} = \mathcal{P}_{\mathbf{A}}\widehat{\mathbf{Y}},$$

with $\mathbf{A}$ satisfying

$$\mathbf{AA}^\dagger\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T = \mathbf{AA}^\dagger\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T\mathbf{AA}^\dagger = \widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T\mathbf{AA}^\dagger. \tag{23}$$

Proof If $\mathbf{A}$ and $\mathbf{B}$ define a critical point of $f(\mathbf{A},\mathbf{B})$, then (22) holds and the general solution to (22) satisfies

$$\mathbf{B} = (\mathbf{A}^T\mathbf{A})^\dagger\mathbf{A}^T\widehat{\mathbf{Y}} + (\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{L}, \tag{24}$$

for some matrix $\mathbf{L}$. So $\mathbf{M} = \mathbf{AB} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^\dagger\mathbf{A}^T\widehat{\mathbf{Y}} = \mathbf{AA}^\dagger\widehat{\mathbf{Y}} = \mathcal{P}_{\mathbf{A}}\widehat{\mathbf{Y}}$ by the property of the Moore-Penrose pseudo-inverse $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^\dagger\mathbf{A}^T$.

By (22), we also have

$$\mathbf{ABB}^T\mathbf{A}^T = \widehat{\mathbf{Y}}\mathbf{B}^T\mathbf{A}^T \quad \text{or equivalently} \quad \mathbf{MM}^T = \widehat{\mathbf{Y}}\mathbf{M}^T.$$

Plugging in the relation $\mathbf{M} = \mathbf{AA}^\dagger\widehat{\mathbf{Y}}$, (A) can be rewritten as

$$\mathbf{AA}^\dagger\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T\mathbf{AA}^\dagger = \widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T\mathbf{AA}^\dagger.$$

Note that the matrix $\mathbf{AA}^\dagger\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T\mathbf{AA}^\dagger$ is symmetric. Thus

$$\mathbf{AA}^\dagger\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T = \mathbf{AA}^\dagger\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T\mathbf{AA}^\dagger,$$

as desired. ■

To prove Lemma 1, we also need the following claim.

Claim 4 *Denote by $\mathcal{I} = \{i_1, i_2, ..., i_r\}$ any ordered r-index set (ordered by $\lambda_{i_j}, j \in [r]$ from the largest to the smallest) and λ_i, $i \in [n_1]$, the eigenvalues of $\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T \in \mathbb{R}^{n_1\times n_1}$ with p distinct values. Let $\mathbf{U}_{\mathcal{I}} = [\mathbf{u}_{i_1}, \mathbf{u}_{i_2}, ..., \mathbf{u}_{i_r}]$ denote the matrix formed by the orthonormal eigenvectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{n_1}]$ of $\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T \in \mathbb{R}^{n_1\times n_1}$ associated with the ordered p eigenvalues, whose multiplicities are $m_1, m_2, ..., m_p$ ($m_1 + m_2 + ... + m_p = n_1$). Then two matrices $\mathbf{A}$ and $\mathbf{B}$ define a critical point of $f(\mathbf{A},\mathbf{B})$ if and only if there exists an ordered r-index set $\mathcal{I}$, an invertible matrix $\mathbf{C}$, and an $r \times n$ matrix $\mathbf{L}$ such that*

$$\mathbf{A} = (\mathbf{UD})_{:\mathcal{I}}\mathbf{C} \quad \textit{and} \quad \mathbf{B} = \mathbf{A}^\dagger\widehat{\mathbf{Y}} + (\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{L}, \tag{25}$$

where $\mathbf{D}$ is a p-block-diagonal matrix with each block equal to the orthogonal projector of dimension m_i. For such a critical point, we have

$$\mathbf{AB} = \mathcal{P}_{\mathbf{A}}\widehat{\mathbf{Y}},$$

$$f(\mathbf{A},\mathbf{B}) = \frac{1}{2}\left(\mathsf{tr}(\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T) - \sum_{i\in\mathcal{I}}\lambda_i\right) = \frac{1}{2}\sum_{i\notin\mathcal{I}}\lambda_i. \tag{26}$$

Proof Note that $\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T$ is a real symmetric covariance matrix. So it can always be represented as $\mathbf{U\Lambda U}^T$, where $\mathbf{U} \in \mathbb{R}^{n_1\times n_1}$ is an orthonormal matrix consisting of eigenvectors of $\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T$ and $\mathbf{\Lambda} \in \mathbb{R}^{n_1\times n_1}$ is a diagonal matrix with non-increasing eigenvalues of $\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T$.

If $\mathbf{A}$ and $\mathbf{B}$ satisfy (25) for some $\mathbf{C}$, $\mathbf{L}$, and $\mathcal{I}$, then

$$\mathbf{ABB}^T = \widehat{\mathbf{Y}}\mathbf{B}^T \quad \text{and} \quad \mathbf{A}^T\mathbf{AB} = \mathbf{A}^T\widehat{\mathbf{Y}},$$

which is (22). So $\mathbf{A}$ and $\mathbf{B}$ define a critical point of $f(\mathbf{A},\mathbf{B})$.

For the converse, notice that

$$\mathcal{P}_{\mathbf{U}^T\mathbf{A}} = \mathbf{U}^T\mathbf{A}(\mathbf{U}^T\mathbf{A})^\dagger = \mathbf{U}^T\mathbf{AA}^\dagger\mathbf{U} = \mathbf{U}^T\mathcal{P}_{\mathbf{A}}\mathbf{U},$$

or equivalently, $\mathcal{P}_{\mathbf{A}} = \mathbf{U}\mathcal{P}_{\mathbf{U}^T\mathbf{A}}\mathbf{U}^T$. Thus (23) yields

$$\mathbf{U}\mathcal{P}_{\mathbf{U}^T\mathbf{A}}\mathbf{U}^T\mathbf{U\Lambda U}^T = \mathbf{U\Lambda U}^T\mathbf{U}\mathcal{P}_{\mathbf{U}^T\mathbf{A}}\mathbf{U}^T,$$

or equivalently, $\mathcal{P}_{\mathbf{U}^T\mathbf{A}}\mathbf{\Lambda} = \mathbf{\Lambda}\mathcal{P}_{\mathbf{U}^T\mathbf{A}}$. Notice that $\mathbf{\Lambda} \in \mathbb{R}^{n_1\times n_1}$ is a diagonal matrix with p distinct eigenvalues of $\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T$. So $\mathcal{P}_{\mathbf{U}^T\mathbf{A}}$ is a block-diagonal matrix with p blocks, each of which is an orthogonal projector of dimension m_i, corresponding to the eigenvalues λ_i, $i \in [p]$. Therefore, there exists an index set $\mathcal{I}$ such that $\mathcal{P}_{\mathbf{U}^T\mathbf{A}} = \mathbf{D}_{:\mathcal{I}}\mathbf{D}_{:\mathcal{I}}^T$, where $\mathbf{D}$ is a block-diagonal matrix. It follows that

$$\mathcal{P}_{\mathbf{A}} = \mathbf{U}\mathcal{P}_{\mathbf{U}^T\mathbf{A}}\mathbf{U}^T = \mathbf{UD}_{:\mathcal{I}}\mathbf{D}_{:\mathcal{I}}^T\mathbf{U}^T = (\mathbf{UD})_{:\mathcal{I}}(\mathbf{UD})_{:\mathcal{I}}^T.$$

Since the column space of $\mathbf{A}$ coincides with the column space of $(\mathbf{UD})_{:\mathcal{I}}$, $\mathbf{A}$ is of the form $\mathbf{A} = (\mathbf{UD})_{:\mathcal{I}}\mathbf{C}$, and $\mathbf{B}$ is given by (24). Thus $\mathbf{AB} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^\dagger\mathbf{A}^T\widehat{\mathbf{Y}} + \mathbf{A}(\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{L} = \mathcal{P}_{\mathbf{A}}\widehat{\mathbf{Y}}$

and

$$\begin{aligned} f(\mathbf{A},\mathbf{B}) &= \frac{1}{2}\|\widehat{\mathbf{Y}} - \mathbf{A}\mathbf{B}\|_F^2 \\ &= \frac{1}{2}\|\widehat{\mathbf{Y}} - \mathcal{P}_{\mathbf{A}}\widehat{\mathbf{Y}}\|_F^2 \\ &= \frac{1}{2}\|\mathcal{P}_{\mathbf{A}^\perp}\widehat{\mathbf{Y}}\|_F^2 \\ &= \frac{1}{2}\sum_{i\notin\mathcal{I}}\lambda_i. \end{aligned}$$

■

So the local minimizer of $f(\mathbf{A},\mathbf{B})$ is given by (25) with $\mathcal{I} = \Phi$, which is globally optimal according to (26), where Φ is the index set corresponding to the r largest eigenvalues of $\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^T$. We then show that when $\mathcal{I}$ consists of other combinations of indices of eigenvalues, i.e., $\mathcal{I} \neq \Phi$, the corresponding pair $(\mathbf{A},\mathbf{B})$ given by (25) is a strict saddle point.

Claim 5 *If $\mathcal{I} \neq \Phi$, then the pair $(\mathbf{A},\mathbf{B})$ given by* (25) *is a strict saddle point.*

Proof Let $i \in \Phi$ but $i \notin \mathcal{I}$, and denote by $\mathbf{U}\mathbf{D} = \mathbf{R}$. It is enough to slightly perturb the column space of $\mathbf{A}$ towards the direction of an eigenvector of λ_i. More precisely, fix two indices i and j such that $i \in \Phi$, $i \notin \mathcal{I}$, and j is the largest index in $\mathcal{I}$. For any ϵ, let $\widetilde{\mathbf{R}}_{:j} = (1+\epsilon^2)^{-1/2}(\mathbf{R}_{:j} + \epsilon\mathbf{R}_{:i})$. Notice that $i \notin \mathcal{I}$. Thus $\widetilde{\mathbf{R}}_{:j}^T\widetilde{\mathbf{R}}_{:j} = \mathbf{I}$. Let $\widetilde{\mathbf{A}} = \widetilde{\mathbf{R}}_{\mathcal{I}}\mathbf{C}$ and $\widetilde{\mathbf{B}} = \widetilde{\mathbf{A}}^\dagger\mathbf{Y} + (\mathbf{I} - \widetilde{\mathbf{A}}^\dagger\widetilde{\mathbf{A}})\mathbf{L}$. A direct calculation shows that

$$f(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}}) = f(\mathbf{A},\mathbf{B}) - \epsilon^2(\lambda_i - \lambda_j)/(2+2\epsilon^2).$$

Hence,

$$\lim_{\epsilon\to 0}\frac{f(\widetilde{\mathbf{A}},\widetilde{\mathbf{B}}) - f(\mathbf{A},\mathbf{B})}{\epsilon^2} = -\frac{1}{2}(\lambda_i - \lambda_j) < 0.$$

■

Note that all critical points of $f(\mathbf{A},\mathbf{B})$ are in the form of (25), and if $\mathcal{I} \neq \Phi$, the pair $(\mathbf{A},\mathbf{B})$ given by (25) is a strict saddle point, while if $\mathcal{I} = \Phi$, then the pair $(\mathbf{A},\mathbf{B})$ given by (25) is a local minimum. We conclude that $f(\mathbf{A},\mathbf{B})$ has no local maximum. The proof is completed. ■

B. Existence of Dual Certificate for Matrix Completion I

Let $\widetilde{\mathbf{A}} \in \mathbb{R}^{n_1\times r}$ and $\widetilde{\mathbf{B}} \in \mathbb{R}^{r\times n_2}$ such that $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$. Then we have the following lemma.

Lemma 11 *Let $\Omega \sim \mathsf{Uniform}(m)$ be the support set uniformly distributed among all sets of cardinality m. Suppose that $m \geq c\kappa^2\mu n_{(1)} r\log n_{(1)}\log_{2\kappa} n_{(1)}$ for an absolute constant c and $\mathbf{X}^*$ obeys*

μ-incoherence (5)*. Then there exists $\widetilde{\mathbf{\Lambda}}$ such that*

$$\begin{aligned} &(1) \quad \widetilde{\mathbf{\Lambda}} \in \Omega, \\ &(2) \quad \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}, \\ &(3) \quad \|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \frac{2}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}). \end{aligned} \tag{27}$$

with probability at least $1 - n_{(1)}^{-10}$.

The rest of the section is devoted to the proof of Lemma 11. We begin with the following lemma.

Lemma 12 *If we can construct an $\mathbf{\Lambda}$ such that*

$$\begin{aligned} &\text{(a)} \quad \mathbf{\Lambda} \in \Omega, \\ &\text{(b)} \quad \|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_F \leq \sqrt{\frac{r}{3n_{(1)}^2}}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}), \\ &\text{(c)} \quad \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\| < \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}), \end{aligned} \tag{28}$$

then we can construct an $\widetilde{\mathbf{\Lambda}}$ such that Eqn. (27) *holds with probability at least $1 - n_{(1)}^{-10}$.*

Proof To prove the lemma, we first claim the following theorem.

Theorem 13 (Candès and Recht (2009), Theorem 4.1) *Assume that Ω is sampled according to the Bernoulli model with success probability $p = \Theta(\frac{m}{n_1 n_2})$, and incoherence condition* (5) *holds. Then there is an absolute constant C_R such that for $\beta > 1$, we have*

$$\|p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}} - \mathcal{P}_{\mathcal{T}}\| \leq C_R\sqrt{\frac{\beta\mu n_{(1)} r\log n_{(1)}}{m}} \triangleq \epsilon,$$

with probability at least $1 - 3n^{-\beta}$ provided that $C_R\sqrt{\frac{\beta\mu n_{(1)} r\log n_{(1)}}{m}} < 1$.

Suppose that Condition (28) holds. Let $\mathbf{Y} = \widetilde{\mathbf{\Lambda}} - \mathbf{\Lambda} \in \Omega$ be the perturbation matrix between $\mathbf{\Lambda}$ and $\widetilde{\mathbf{\Lambda}}$ such that $\mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$. Such a $\mathbf{Y}$ exists by setting $\mathbf{Y} = \mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}(\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$. So $\|\mathcal{P}_{\mathcal{T}}\mathbf{Y}\|_F \leq \sqrt{\frac{r}{3n_{(1)}^2}}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$. We now prove Condition (3) in Eqn. (27). Observe that

$$\begin{aligned} \|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| &\leq \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\| + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| \\ &\leq \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\|. \end{aligned} \tag{29}$$

So we only need to show $\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| \leq \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$.

Before proceeding, we begin by introducing a normalized version $\mathcal{Q}_{\Omega} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^{n_1\times n_2}$ of $\mathcal{P}_{\Omega}$:

$$\mathcal{Q}_{\Omega} = p^{-1}\mathcal{P}_{\Omega} - \mathcal{I}.$$

With this, we have

$$\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}} = p\mathcal{P}_{\mathcal{T}}(\mathcal{I}+\mathcal{Q}_{\Omega})\mathcal{P}_{\mathcal{T}}.$$

Note that for any operator $\mathcal{P}:\mathcal{T}\to\mathcal{T}$, we have

$$\mathcal{P}^{-1} = \sum_{k\geq 0}(\mathcal{P}_{\mathcal{T}}-\mathcal{P})^k \text{ whenever } \|\mathcal{P}_{\mathcal{T}}-\mathcal{P}\|<1.$$

So according to Theorem 13, the operator $p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}$ can be represented as a *convergent* Neumann series

$$p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1} = \sum_{k\geq 0}(-1)^k(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}})^k,$$

because $\|\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}\|\leq\epsilon<\frac{1}{2}$ once $m\geq C\mu n_{(1)}r\log n_{(1)}$ for a sufficiently large absolute constant C. We also note that

$$p(\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}) = \mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}},$$

because $\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\mathcal{T}}=0$. Thus

$$\begin{aligned}
\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| &= \|\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}(\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda})-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\| \\
&= \|\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}})^{-1}((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda})-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\| \\
&= \|\sum_{k\geq 0}(-1)^k\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}})^k((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda})-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\| \\
&\leq \sum_{k\geq 0}\|(-1)^k\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_{\Omega}(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}})^k((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda})-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\|_F \\
&\leq \|\mathcal{Q}_{\Omega}\|\sum_{k\geq 0}\|\mathcal{P}_{\mathcal{T}}\mathcal{Q}_{\Omega}\mathcal{P}_{\mathcal{T}}\|^k\|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda})-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}))\|_F \\
&\leq \frac{4}{p}\|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda})-\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})\|_F \\
&\leq \Theta\left(\frac{n_1n_2}{m}\right)\sqrt{\frac{r}{3n_{(1)}^2}}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) \\
&\leq \frac{1}{3}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})
\end{aligned}$$

with high probability. The proof is completed. ■

It thus suffices to construct a dual certificate $\mathbf{\Lambda}$ such that all conditions in (28) hold. To this end, partition $\Omega=\Omega_1\cup\Omega_2\cup...\cup\Omega_b$ into b partitions of size q. By assumption, we may choose

$$q\geq\frac{128}{3}C\beta\kappa^2\mu r n_{(1)}\log n_{(1)} \quad\text{and}\quad b\geq\frac{1}{2}\log_{2\kappa}\left(24^2n_{(1)}^2\kappa^2\right)$$

for a sufficiently large constant C. Let $\Omega_j\sim\mathsf{Ber}(q)$ denote the set of indices corresponding to the j-th partitions. Define $\mathbf{W}_0=\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$ and set $\mathbf{\Lambda}_k=\frac{n_1n_2}{q}\sum_{j=1}^k\mathcal{P}_{\Omega_j}(\mathbf{W}_{j-1})$, $\mathbf{W}_k=\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}-\mathcal{P}_{\mathcal{T}}(\mathbf{\Lambda}_k)$ for $k=1,2,...,b$. Then by Theorem 13,

$$\begin{aligned}
\|\mathbf{W}_k\|_F &= \left\|\mathbf{W}_{k-1}-\frac{n_1n_2}{q}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega_k}(\mathbf{W}_{k-1})\right\|_F = \left\|\left(\mathcal{P}_{\mathcal{T}}-\frac{n_1n_2}{q}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega_k}\mathcal{P}_{\mathcal{T}}\right)(\mathbf{W}_{k-1})\right\|_F \\
&\leq \frac{1}{2\kappa}\|\mathbf{W}_{k-1}\|_F.
\end{aligned}$$

So it follows that $\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}-\mathcal{P}_{\mathcal{T}}(\mathbf{\Lambda}_b)\|_F=\|\mathbf{W}_b\|_F\leq(2\kappa)^{-b}\|\mathbf{W}_0\|_F\leq(2\kappa)^{-b}\sqrt{r}\sigma_1(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})\leq\sqrt{\frac{r}{24^2n_{(1)}^2}}\sigma_r(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}})$.

The following lemma together implies the strong duality of (15) straightforwardly.

Lemma 14 *Under the assumptions of Theorem 7, the dual certification $\mathbf{\Lambda}_b$ obeys the dual condition* (28) *with probability at least* $1-n_{(1)}^{-10}$.

Proof It is well known that for matrix completion, the Uniform model $\Omega\sim\mathsf{Uniform}(m)$ is equivalent to the Bernoulli model $\Omega\sim\mathsf{Ber}(p)$, where each element in $[n_1]\times[n_2]$ is included with probability $p=\Theta(m/(n_1n_2))$ independently; see Section I for a brief justification. By the equivalence, we can suppose $\Omega\sim\mathsf{Ber}(p)$.

To prove Lemma 14, as a preliminary, we need the following lemmas.

Lemma 15 (Chen (2015), Lemma 2) *Suppose $\mathbf{Z}$ is a fixed matrix. Suppose $\Omega\sim\mathsf{Ber}(p)$. Then with high probability,*

$$\|(\mathcal{I}-p^{-1}\mathcal{P}_{\Omega})\mathbf{Z}\|\leq C_0'\left(\frac{\log n_{(1)}}{p}\|\mathbf{Z}\|_\infty+\sqrt{\frac{\log n_{(1)}}{p}}\|\mathbf{Z}\|_{\infty,2}\right),$$

where $C_0'>0$ is an absolute constant and

$$\|\mathbf{Z}\|_{\infty,2}=\max\left\{\max_i\sqrt{\sum_b\mathbf{Z}_{ib}^2},\max_j\sqrt{\sum_a\mathbf{Z}_{aj}^2}\right\}.$$

Lemma 16 (Candès et al. (2011), Lemma 3.1) *Suppose $\Omega\sim\mathsf{Ber}(p)$ and $\mathbf{Z}$ is a fixed matrix. Then with high probability,*

$$\|\mathbf{Z}-p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathbf{Z}\|_\infty\leq\epsilon\|\mathbf{Z}\|_\infty,$$

provided that $p\geq C_0\epsilon^{-2}(\mu r\log n_{(1)})/n_{(2)}$ for some absolute constant $C_0>0$.

Lemma 17 (Chen (2015), Lemma 3) *Suppose that $\mathbf{Z}$ is a fixed matrix and $\Omega\sim\mathsf{Ber}(p)$. If $p\geq c_0\mu r\log n_{(1)}/n_{(2)}$ for some c_0 sufficiently large, then with high probability,*

$$\|(p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}-\mathcal{P}_{\mathcal{T}})\mathbf{Z}\|_{\infty,2}\leq\frac{1}{2}\sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{Z}\|_\infty+\frac{1}{2}\|\mathbf{Z}\|_{\infty,2}.$$

Observe that by Lemma 16,

$$\|\mathbf{W}_j\|_\infty\leq\left(\frac{1}{2}\right)^j\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty,$$

and by Lemma 17,

$$\|\mathbf{W}_j\|_{\infty,2}\leq\frac{1}{2}\sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{W}_{j-1}\|_\infty+\frac{1}{2}\|\mathbf{W}_{j-1}\|_{\infty,2}.$$

So

$$\begin{aligned}
&\|\mathbf{W}_j\|_{\infty,2} \\
&\le \left(\frac{1}{2}\right)^j \sqrt{\frac{n_{(1)}}{\mu r}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + \frac{1}{2}\|\mathbf{W}_{j-1}\|_{\infty,2} \\
&\le j\left(\frac{1}{2}\right)^j \sqrt{\frac{n_{(1)}}{\mu r}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + \left(\frac{1}{2}\right)^j \|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_{\infty,2}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}_b\| \\
&\le \sum_{j=1}^{b} \|\frac{n_1 n_2}{q}\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega_j}\mathbf{W}_{j-1}\| \\
&= \sum_{j=1}^{b} \|\mathcal{P}_{\mathcal{T}^\perp}(\frac{n_1 n_2}{q}\mathcal{P}_{\Omega_j}\mathbf{W}_{j-1} - \mathbf{W}_{j-1})\| \\
&\le \sum_{j=1}^{b} \|(\frac{n_1 n_2}{q}\mathcal{P}_{\Omega_j} - \mathcal{I})(\mathbf{W}_{j-1})\|.
\end{aligned}$$

Let p denote $\Theta\left(\frac{q}{n_1 n_2}\right)$. By Lemma 15,

$$\begin{aligned}
&\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}_b\| \\
&\le C_0'\frac{\log n_{(1)}}{p}\sum_{j=1}^{b}\|\mathbf{W}_{j-1}\|_\infty + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sum_{j=1}^{b}\|\mathbf{W}_{j-1}\|_{\infty,2} \\
&\le C_0'\frac{\log n_{(1)}}{p}\sum_{j=1}^{b}\left(\frac{1}{2}\right)^j\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty \\
&\quad + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sum_{j=1}^{b}\left[j\left(\frac{1}{2}\right)^j\sqrt{\frac{n_{(1)}}{\mu r}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + \left(\frac{1}{2}\right)^j\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_{\infty,2}\right] \\
&\le C_0'\frac{\log n_{(1)}}{p}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + 2C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sqrt{\frac{n_{(1)}}{\mu r}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_\infty + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\|\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\|_{\infty,2}.
\end{aligned}$$

Setting $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$, we note the facts that (we assume WLOG $n_2 \ge n_1$)

$$\|\mathbf{X}^*\|_{\infty,2} = \max_i \|\mathbf{e}_i^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\|_2 \le \max_i \|\mathbf{e}_i^T\mathbf{U}\|\sigma_1(\mathbf{X}^*) \le \sqrt{\frac{\mu r}{n_1}}\sigma_1(\mathbf{X}^*) \le \sqrt{\frac{\mu r}{n_1}}\kappa\sigma_r(\mathbf{X}^*),$$

and that

$$\begin{aligned}
\|\mathbf{X}^*\|_\infty &= \max_{ij}\langle\mathbf{X}^*, \mathbf{e}_i\mathbf{e}_j^T\rangle = \max_{ij}\langle\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \mathbf{e}_i\mathbf{e}_j^T\rangle = \max_{ij}\langle\mathbf{e}_i^T\mathbf{U}\mathbf{\Sigma}, \mathbf{e}_j^T\mathbf{V}\rangle \\
&\le \max_{ij}\|\mathbf{e}_i^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\|_2\|\mathbf{e}_j^T\mathbf{V}\|_2 \le \max_j\|\mathbf{X}^*\|_{\infty,2}\|\mathbf{e}_j^T\mathbf{V}\|_2 \le \frac{\mu r\kappa}{\sqrt{n_1 n_2}}\sigma_r(\mathbf{X}^*).
\end{aligned}$$

Substituting $p = \Theta\left(\frac{\kappa^2\mu r n_{(1)}\log(n_{(1)})\log_{2\kappa}(n_{(1)})}{n_1 n_2}\right)$, we obtain $\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}_b\| < \frac{1}{3}\sigma_r(\mathbf{X}^*)$. The proof is completed. ■

C. Subgradient of the $r*$ Function

Lemma 18 *Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the skinny SVD of matrix $\mathbf{X}^*$ of rank r. The subdifferential of $\|\cdot\|_{r*}$ evaluated at $\mathbf{X}^*$ is given by*

$$\partial\|\mathbf{X}^*\|_{r*} = \{\mathbf{X}^* + \mathbf{W} : \mathbf{U}^T\mathbf{W} = \mathbf{0}, \mathbf{W}\mathbf{V} = \mathbf{0}, \|\mathbf{W}\| \le \sigma_r(\mathbf{X}^*)\}. \tag{30}$$

Proof Note that for any fixed function $f(\cdot)$, the set of all optimal solutions of the problem

$$f^*(\mathbf{X}^*) = \max_{\mathbf{Y}}\langle\mathbf{X}^*, \mathbf{Y}\rangle - f(\mathbf{Y}) \tag{31}$$

form the subdifferential of the conjugate function $f^*(\cdot)$ evaluated at $\mathbf{X}^*$. Set $f(\cdot)$ to be $\frac{1}{2}\|\cdot\|_r^2$ and notice that the function $\frac{1}{2}\|\cdot\|_r^2$ is unitarily invariant. By Von Neumann's trace inequality, the optimal solutions to problem (31) are given by

$$[\mathbf{U}, \mathbf{U}^\perp]\mathsf{Diag}([\sigma_1(\mathbf{Y}), ..., \sigma_r(\mathbf{Y}), \sigma_{r+1}(\mathbf{Y}), ..., \sigma_{n_{(2)}}(\mathbf{Y})])[\mathbf{V}, \mathbf{V}^\perp]^T,$$

where $\{\sigma_i(\mathbf{Y})\}_{i=r+1}^{n_{(2)}}$ can be any value no larger than $\sigma_r(\mathbf{Y})$ and $\{\sigma_i(\mathbf{Y})\}_{i=1}^r$ are given by the optimal solution to the problem

$$\max_{\{\sigma_i(\mathbf{Y})\}_{i=1}^r}\sum_{i=1}^r\sigma_i(\mathbf{X}^*)\sigma_i(\mathbf{Y}) - \frac{1}{2}\sum_{i=1}^r\sigma_i^2(\mathbf{Y}).$$

The solution is unique such that $\sigma_i(\mathbf{Y}) = \sigma_i(\mathbf{X}^*)$, $i = 1, 2, ..., r$. The proof is complete. ■

D. Proof of Theorem 7

We will prove Theorem 7 in this section.

D.1. Exact Recoverability of Non-Convex Formulation

Theorem 19 (Uniqueness of Solution) *Let $\Omega \sim \mathsf{Uniform}(m)$ be the support set uniformly distributed among all sets of cardinality m. Suppose that $m \ge c\kappa^2\mu n_{(1)} r\log n_{(1)}\log_{2\kappa} n_{(1)}$ for an absolute constant c and $\mathbf{X}^*$ obeys μ-incoherence* (5)*. Then $\mathbf{X}^*$ is the unique solution of non-convex optimization*

$$\min_{\mathbf{A},\mathbf{B}}\frac{1}{2}\|\mathbf{A}\mathbf{B}\|_F^2, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{A}\mathbf{B}) = \mathcal{P}_\Omega(\mathbf{X}^*),$$

with probability at least $1 - n_{(1)}^{-10}$.

Proof We note that a recovery result under the Bernoulli model automatically implies a corresponding result for the uniform model Candès et al. (2011); see Section I for the details. So in the following, we assume the Bernoulli model.

Consider the feasibility of the matrix completion problem:

$$\text{Find a matrix } \mathbf{X} \in \mathbb{R}^{n_1 \times n_2} \text{ such that } \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*), \quad \|\mathbf{X}\|_F \le \|\mathbf{X}^*\|_F, \quad \mathsf{rank}(\mathbf{X}) \le r. \tag{32}$$

Note that if $\mathbf{X}^*$ is the unique solution of (32), then $\mathbf{X}^*$ is the unique solution of (14). We now show the former. Our proof first identifies a feasibility condition for problem (32), and then shows that $\mathbf{X}^*$ is the only matrix that obeys this feasibility condition when the sample size is large enough. We denote by

$$\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) = \{\mathbf{X} - \mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2} : \mathsf{rank}(\mathbf{X}) \le r,\ \|\mathbf{X}\|_F \le \|\mathbf{X}^*\|_F\},$$

and

$$\mathcal{T} = \{\mathbf{U}\mathbf{X}^T + \mathbf{Y}\mathbf{V}^T,\ \mathbf{X} \in \mathbb{R}^{n_2 \times r},\ \mathbf{Y} \in \mathbb{R}^{n_1 \times r}\},$$

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the skinny SVD of $\mathbf{X}^*$.

We have the following proposition for the feasibility of problem (32).

Proposition 20 (Feasibility Condition) *$\mathbf{X}^*$ is the unique feasible solution to problem* (32) *if $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$.*

Proof Notice that problem (32) is equivalent to another feasibility problem

$$\text{Find a matrix } \mathbf{D} \in \mathbb{R}^{n_1 \times n_2} \text{ such that } \quad \mathsf{rank}(\mathbf{X}^* + \mathbf{D}) \le r,\ \ \|\mathbf{X}^* + \mathbf{D}\|_F \le \|\mathbf{X}^*\|_F,\ \ \mathbf{D} \in \Omega^\perp.$$

Suppose that $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$. Since $\mathsf{rank}(\mathbf{X}^* + \mathbf{D}) \le r$ and $\|\mathbf{X}^* + \mathbf{D}\|_F \le \|\mathbf{X}^*\|_F$ are equivalent to $\mathbf{D} \in \mathcal{D}_{\mathcal{S}}(\mathbf{X}^*)$, and note that $\mathbf{D} \in \Omega^\perp$, we have $\mathbf{D} = \mathbf{0}$, which means $\mathbf{X}^*$ is the unique feasible solution to problem (32). ■

The remainder of the proof is to show $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$. To proceed, we note that

$$\begin{aligned}
\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) &= \left\{\mathbf{X} - \mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2} : \mathsf{rank}(\mathbf{X}) \le r,\ \frac{1}{2}\|\mathbf{X}\|_F^2 \le \frac{1}{2}\|\mathbf{X}^*\|_F^2\right\} \\
&\subseteq \{\mathbf{X} - \mathbf{X}^* \in \mathbb{R}^{n_1 \times n_2} : \|\mathbf{X}\|_{r*} \le \|\mathbf{X}^*\|_{r*}\} \quad \left(\text{since } \frac{1}{2}\|\mathbf{Y}\|_F^2 = \|\mathbf{Y}\|_{r*} \text{ for any rank-}r \text{ matrix}\right) \\
&\triangleq \mathcal{D}_{\mathcal{S}_*}(\mathbf{X}^*).
\end{aligned}$$

We now show that

$$\mathcal{D}_{\mathcal{S}_*}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}, \tag{33}$$

when $m \ge c\kappa^2 \mu r n_{(1)} \log_{2\kappa}(n_{(1)}) \log(n_{(1)})$, which will prove $\mathcal{D}_{\mathcal{S}}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$ as desired.

By Lemma 11, there exists a $\mathbf{\Lambda}$ such that

$$\begin{aligned}
&(1) \qquad \mathbf{\Lambda} \in \Omega, \\
&(2) \qquad \mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) = \mathbf{X}^*, \\
&(3) \qquad \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\| < \frac{2}{3}\sigma_r(\mathbf{X}^*).
\end{aligned}$$

Consider any $\mathbf{D} \in \Omega^\perp$ such that $\mathbf{D} \ne \mathbf{0}$. By Lemma 18, for any $\mathbf{W} \in \mathcal{T}^\perp$ and $\|\mathbf{W}\| \le \sigma_r(\mathbf{X}^*)$,

$$\|\mathbf{X}^* + \mathbf{D}\|_{r*} \ge \|\mathbf{X}^*\|_{r*} + \langle \mathbf{X}^* + \mathbf{W}, \mathbf{D}\rangle.$$

Since $\langle \mathbf{W}, \mathbf{D}\rangle = \langle \mathcal{P}_{\mathcal{T}^\perp}\mathbf{W}, \mathbf{D}\rangle = \langle \mathbf{W}, \mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\rangle$, we can choose $\mathbf{W}$ such that

$$\langle \mathbf{W}, \mathbf{D}\rangle = \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_*.$$

Then

$$\begin{aligned}
\|\mathbf{X}^* + \mathbf{D}\|_{r*} &\ge \|\mathbf{X}^*\|_{r*} + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* + \langle \mathbf{X}^*, \mathbf{D}\rangle \\
&= \|\mathbf{X}^*\|_{r*} + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* + \langle \mathbf{X}^* + \mathbf{\Lambda}, \mathbf{D}\rangle \quad (\text{since } \mathbf{\Lambda} \in \Omega \text{ and } \mathbf{D} \in \Omega^\perp) \\
&= \|\mathbf{X}^*\|_{r*} + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* + \langle \mathbf{X}^* + \mathcal{P}_{\mathcal{T}}\mathbf{\Lambda}, \mathbf{D}\rangle + \langle \mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}, \mathbf{D}\rangle \\
&= \|\mathbf{X}^*\|_{r*} + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* + \langle \mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}, \mathbf{D}\rangle \quad (\text{by condition (2)}) \\
&= \|\mathbf{X}^*\|_{r*} + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* + \langle \mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}, \mathbf{D}\rangle \\
&= \|\mathbf{X}^*\|_{r*} + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* + \langle \mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}, \mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\rangle \\
&\ge \|\mathbf{X}^*\|_{r*} + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* - \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\|\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* \quad (\text{by Hölder's inequality}) \\
&\ge \|\mathbf{X}^*\|_{r*} + \frac{1}{3}\sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{D}\|_* \quad (\text{by condition (3)}).
\end{aligned}$$

So if $\mathcal{T} \cap \Omega^\perp = \{\mathbf{0}\}$, since $\mathbf{D} \in \Omega^\perp$ and $\mathbf{D} \ne \mathbf{0}$, we have $\mathbf{D} \notin \mathcal{T}$. Therefore,

$$\|\mathbf{X}^* + \mathbf{D}\|_{r*} > \|\mathbf{X}^*\|_{r*}$$

which then leads to $\mathcal{D}_{\mathcal{S}_*}(\mathbf{X}^*) \cap \Omega^\perp = \{\mathbf{0}\}$.

The rest of proof is to show that $\mathcal{T} \cap \Omega^\perp = \{\mathbf{0}\}$. We have the following lemma.

Lemma 21 *Assume that $\Omega \sim \mathsf{Ber}(p)$ and the incoherence condition* (5) *holds. Then with probability at least $1 - n_{(1)}^{-10}$, we have $\|\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\| \le \sqrt{1 - p + \epsilon p}$, provided that $p \ge C_0\epsilon^{-2}(\mu r \log n_{(1)})/n_{(2)}$, where C_0 is an absolute constant.*

Proof If $\Omega \sim \mathsf{Ber}(p)$, we have, by Theorem 13, that with high probability

$$\|\mathcal{P}_{\mathcal{T}} - p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}}\| \le \epsilon,$$

provided that $p \ge C_0\epsilon^{-2}\frac{\mu r \log n_{(1)}}{n_{(2)}}$. Note, however, that since $\mathcal{I} = \mathcal{P}_\Omega + \mathcal{P}_{\Omega^\perp}$,

$$\mathcal{P}_{\mathcal{T}} - p^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}} = p^{-1}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}} - (1-p)\mathcal{P}_{\mathcal{T}})$$

and, therefore, by the triangle inequality

$$\|\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\| \le \epsilon p + (1-p).$$

Since $\|\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\|^2 \le \|\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\|$, the proof is completed. ■

We note that $\|\mathcal{P}_{\Omega^\perp}\mathcal{P}_{\mathcal{T}}\| < 1$ implies $\Omega^\perp \cap \mathcal{T} = \{\mathbf{0}\}$. The proof of latter part of the theorem is completed. ■

D.2. Strong Duality

We have shown in Theorem 19 that the problem $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} \frac{1}{2}\|\mathbf{AB}\|_F^2$, s.t.$\mathcal{P}_\Omega(\mathbf{AB}) = \mathcal{P}_\Omega(\mathbf{X}^*)$, exactly recovers $\mathbf{X}^*$, i.e., $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$, with nearly optimal sample complexity. So if strong duality holds, this non-convex optimization problem can be equivalently converted to the convex program (15). Then Theorem 7 is straightforward from strong duality.

It now suffices to apply our unified framework in Section 4 to prove the strong duality. Let

$$H(\mathbf{X}) = \mathbf{I}_{\{\mathbf{M}\in\mathbb{R}^{n_1\times n_2}:\ \mathcal{P}_\Omega\mathbf{M}=\mathcal{P}_\Omega\mathbf{X}^*\}}(\mathbf{X})$$

in Problem (**P**), and let $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ be a global solution to the problem. Then by Theorem 19, $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$. For Problem (**P**) with this special $H(\mathbf{X})$, we have

$$\begin{aligned}\Psi = \partial H(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}) &= \{\mathbf{G}\in\mathbb{R}^{n_1\times n_2} : \langle \mathbf{G}, \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}\rangle \geq \langle \mathbf{G}, \mathbf{Y}\rangle,\ \text{for any } \mathbf{Y}\in\mathbb{R}^{n_1\times n_2} \text{ s.t. } \mathcal{P}_\Omega\mathbf{Y} = \mathcal{P}_\Omega\mathbf{X}^*\}\\ &= \{\mathbf{G}\in\mathbb{R}^{n_1\times n_2} : \langle \mathbf{G}, \mathbf{X}^*\rangle \geq \langle \mathbf{G}, \mathbf{Y}\rangle,\ \text{for any } \mathbf{Y}\in\mathbb{R}^{n_1\times n_2} \text{ s.t. } \mathcal{P}_\Omega\mathbf{Y} = \mathcal{P}_\Omega\mathbf{X}^*\} = \Omega,\end{aligned}$$

where the third equality holds since $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \mathbf{X}^*$. Combining with Lemma 11 shows that the dual condition in Theorem 3 holds with high probability, which leads to strong duality and thus proving Theorem 7.

E. Proof of Theorem 8

Given support set $\Omega \subset [n_1]\times[n_2]$ and rank-r matrix $\mathbf{X}^* \in \mathbb{R}^{n_1\times n_2}$, define the following optimization problem

$$\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r},\mathbf{B}\in\mathbb{R}^{r\times n_2}} \|\mathbf{AB}\|_*, \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathbf{AB}) = \mathcal{P}_\Omega(\mathbf{X}^*). \tag{34}$$

We have the following lemma.

Lemma 22 *For any fixed support set Ω and rank-r matrix $\mathbf{X}^* \in \mathbb{R}^{n_1\times n_2}$, problem* (17) *and the non-convex problem* (34) *have the same solution, i.e.,*

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \bar{\mathbf{A}}\bar{\mathbf{B}},$$

where $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}$ is the optimal solution of (17) *and $\bar{\mathbf{A}}\bar{\mathbf{B}}$ is the optimal solution of* (34).

Proof First, introduce another optimization problem as a connection between the two in the lemma. For any matrix $\mathbf{Z}\in\mathbb{R}^{n_1\times n_2}$ of rank at most r, we consider the following optimization problem

$$\begin{aligned}\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r},\mathbf{B}\in\mathbb{R}^{r\times n_2}} \quad & \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2,\\ \text{s.t.} \quad & \mathbf{AB} = \mathbf{Z},\\ & \mathcal{P}_\Omega(\mathbf{AB}) = \mathcal{P}_\Omega(\mathbf{X}^*).\end{aligned} \tag{35}$$

We want to show that the optimal cost of objective function (35) is $\|\mathbf{Z}\|_*$ and the minimum is achieved when $\|\mathbf{A}\|_F = \|\mathbf{B}\|_F$. Denote by $\mathbf{U\Sigma V}^\top$ the skinny SVD of matrix $\mathbf{Z} = \mathbf{AB}$.

On one hand, we have

$$\begin{aligned}\|\mathbf{Z}\|_* &= \mathsf{tr}(\mathbf{\Sigma})\\ &= \mathsf{tr}(\mathbf{U}^\top\mathbf{U\Sigma V}^\top\mathbf{V}) && (\text{ by } \mathbf{U}^\top\mathbf{U} = \mathbf{I}, \mathbf{V}^\top\mathbf{V} = \mathbf{I}\in\mathbb{R}^{r\times r})\\ &= \mathsf{tr}(\mathbf{U}^\top\mathbf{ABV}) && (\text{ by } \mathbf{U\Sigma V}^\top = \mathbf{Z} = \mathbf{AB})\\ &\leq \sum_{i=1}^r \sigma_i(\mathbf{U}^\top\mathbf{A})\cdot\sigma_i(\mathbf{BV}) && (\text{ by trace inequality})\\ &\leq \frac{1}{2}\|\mathbf{U}^\top\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{BV}\|_F^2 && (\forall a, b : ab \leq \frac{1}{2}(a^2+b^2))\\ &= \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2.\end{aligned}$$

Therefore, $\|\mathbf{Z}\|_* \leq \min_{\mathbf{A},\mathbf{B},\text{ s.t. } \mathbf{AB}=\mathbf{Z},\mathcal{P}_\Omega(\mathbf{AB})=\mathcal{P}_\Omega(\mathbf{X}^*)} \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2$ for any matrix $\mathbf{Z}\in\mathbb{R}^{n_1\times n_2}$ satisfies that $\mathsf{rank}(\mathbf{Z}) \leq r$ and $\mathcal{P}_\Omega(\mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{X}^*)$.

On the other hand, taking $\mathbf{A} = \mathbf{U\Sigma}^{1/2}$ and $\mathbf{B} = \mathbf{\Sigma}^{1/2}\mathbf{V}^\top$, we can verify these properties

$$\begin{aligned}&\mathbf{AB} = \mathbf{U\Sigma}^{1/2}\mathbf{\Sigma}^{1/2}\mathbf{V}^\top = \mathbf{U\Sigma V}^\top = \mathbf{Z},\\ &\mathcal{P}_\Omega(\mathbf{AB}) = \mathcal{P}_\Omega(\mathbf{X}^*),\\ &\|\mathbf{A}\|_F = \|\mathbf{\Sigma}^{1/2}\|_F = \|\mathbf{B}\|_F,\\ &\frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{B}\|_F^2 = \|\mathbf{\Sigma}^{1/2}\|_F^2 = \|\mathbf{Z}\|_*.\end{aligned}$$

Thus (35) holds and the minimum is achieved when $\|\mathbf{A}\|_F = \|\mathbf{B}\|_F$.

We now prove Lemma 22 by contradiction. Assume for contradiction that $(\mathbf{A}^*, \mathbf{B}^*)$ is a solution to (34) such that $\|\mathbf{A}^*\|_F = \|\mathbf{B}^*\|_F$ (for any solution $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$ to problem (34), we can always find the pair $(\mathbf{A}^*, \mathbf{B}^*)$ such that $\mathbf{A}^*\mathbf{B}^* = \bar{\mathbf{A}}\bar{\mathbf{B}}$ and $\|\mathbf{A}^*\|_F = \|\mathbf{B}^*\|_F$), while it is not a solution to (17). So there exists $(\mathbf{A}', \mathbf{B}')$ such that $\mathcal{P}_\Omega(\mathbf{A}'\mathbf{B}') = \mathcal{P}_\Omega(\mathbf{X}^*)$ and

$$\begin{aligned}\|\mathbf{A}'\mathbf{B}'\|_* &= \frac{1}{2}\|\mathbf{A}'\|_F^2 + \frac{1}{2}\|\mathbf{B}'\|_F^2 && (\text{by Problem 35})\\ &< \frac{1}{2}\|\mathbf{A}^*\|_F^2 + \frac{1}{2}\|\mathbf{B}^*\|_F^2 && (\text{by } (\mathbf{A}', \mathbf{B}') \text{ is not a solution to Problem 17})\\ &= \|\mathbf{A}^*\mathbf{B}^*\|_*, && (\text{by Problem 35})\end{aligned}$$

which is contradictory with the optimality of $(\mathbf{A}^*, \mathbf{B}^*)$ to problem (34).

Similarly, assume for contradiction that $(\mathbf{A}^*, \mathbf{B}^*)$ is a solution to (17), while it is not a solution to (34). So there exists $(\mathbf{A}', \mathbf{B}')$ such that $\mathcal{P}_\Omega(\mathbf{A}'\mathbf{B}') = \mathcal{P}_\Omega(\mathbf{X}^*)$ and

$$\begin{aligned}\frac{1}{2}\|\mathbf{A}'\|_F^2 + \frac{1}{2}\|\mathbf{B}'\|_F^2 &= \|\mathbf{A}'\mathbf{B}'\|_*\\ &< \|\mathbf{A}^*\mathbf{B}^*\|_*\\ &= \frac{1}{2}\|\mathbf{A}^*\|_F^2 + \frac{1}{2}\|\mathbf{B}^*\|_F^2,\end{aligned}$$

which is contradictory with the optimality of $(\mathbf{A}^*, \mathbf{B}^*)$ to problem (17). The proof is completed. ■

Before proceeding, we provide the definition of a matrix space.

Definition 23 *Given matrices $\bar{\mathbf{A}} \in \mathbb{R}^{n_1 \times r}$ and $\bar{\mathbf{B}} \in \mathbb{R}^{r \times n_2}$, we denote by $\mathcal{T} \triangleq \{\bar{\mathbf{A}}\mathbf{X}^\top + \mathbf{Y}\bar{\mathbf{B}} : \mathbf{X} \in \mathbb{R}^{n_2 \times r},\ \mathbf{Y} \in \mathbb{R}^{n_1 \times r}\}$ a matrix space. For any matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, we have*

$$\mathcal{P}_{\mathcal{T}}\mathbf{M} = \bar{\mathbf{A}}\bar{\mathbf{A}}^\dagger\mathbf{M} + \mathbf{M}\bar{\mathbf{B}}\bar{\mathbf{B}}^\dagger - \bar{\mathbf{A}}\bar{\mathbf{A}}^\dagger\mathbf{M}\bar{\mathbf{B}}\bar{\mathbf{B}}^\dagger,$$

and

$$\mathcal{P}_{\mathcal{T}^\perp}\mathbf{M} = (\mathbf{I} - \bar{\mathbf{A}}\bar{\mathbf{A}}^\dagger)\mathbf{M}(\mathbf{I} - \bar{\mathbf{B}}\bar{\mathbf{B}}^\dagger).$$

Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the skinny SVD of matrix $\bar{\mathbf{A}}\bar{\mathbf{B}}$. Denote by $\mathcal{U} = \{\mathbf{U}\mathbf{X}^\top : \mathbf{X} \in \mathbb{R}^{n_2 \times r}\}$ and $\mathcal{V} = \{\mathbf{Y}\mathbf{V}^\top : \mathbf{Y} \in \mathbb{R}^{n_1 \times r}\}$. Then the linear space $\mathcal{T}$ can be equivalently represented as $\mathcal{T} = \mathcal{U} + \mathcal{V}$. Therefore, $\mathcal{T}^\perp = (\mathcal{U} + \mathcal{V})^\perp = \mathcal{U}^\perp \cap \mathcal{V}^\perp$.

The following lemma is a restated result of Theorem 4 for the matrix completion problem. For completeness, we include its proof here.

Lemma 24 (Dual Certificate) *Let $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$ be a solution of problem* (34). *Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ denote the skinny SVD of $\bar{\mathbf{A}}\bar{\mathbf{B}}$. If there exists a dual certificate $\widetilde{\mathbf{\Lambda}}$ such that*

$$\begin{aligned} &(a) \quad && \widetilde{\mathbf{\Lambda}} \in \Omega, \\ &(b) \quad && \mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \mathbf{U}\mathbf{V}^\top, \\ &(c) \quad && \|\mathcal{P}_{\mathcal{T}^\perp}(-\widetilde{\mathbf{\Lambda}})\| < \frac{1}{2}, \end{aligned} \tag{36}$$

then strong duality holds, i.e., problem (18) *and problem* (17) *have the same solution.*

Proof For any convex set $\mathcal{C}$, we define indicator function $\mathbf{I}_{\mathcal{C}} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}$ such that

$$\mathbf{I}_{\mathcal{C}}(\mathbf{X}) = \begin{cases} 0, & \text{if } \mathbf{X} \in \mathcal{C}; \\ +\infty, & \text{otherwise.} \end{cases}$$

We set $\mathcal{C}$ to be $\{\mathbf{X} : \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{X}^*), \mathbf{X} \in \mathbb{R}^{n_1 \times n_2}\}$, and define function $H : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}$ such that for any $\mathbf{X}$, $H(\mathbf{X}) = \mathbf{I}_{\mathcal{C}}(\mathbf{X})$.

Then problem (34) can be equivalently represented by

$$(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname*{argmin}_{\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{r \times n_2}} F(\mathbf{A}, \mathbf{B}) = \|\mathbf{A}\mathbf{B}\|_* + H(\mathbf{A}\mathbf{B}).$$

Define $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ to be $\|\mathbf{A}\mathbf{B}\|_* + \langle \mathbf{\Lambda}, \mathbf{A}\mathbf{B} \rangle - H^*(\mathbf{\Lambda})$. Then we have

$$\begin{aligned} F(\mathbf{A}, \mathbf{B}) &= \|\mathbf{A}\mathbf{B}\|_* + H(\mathbf{A}\mathbf{B}) \\ &= \|\mathbf{A}\mathbf{B}\|_* + H^{**}(\mathbf{A}\mathbf{B}) \\ &= \|\mathbf{A}\mathbf{B}\|_* + \max_{\mathbf{\Lambda} \in \mathbb{R}^{n_1 \times n_2}} \langle \mathbf{\Lambda}, \mathbf{A}\mathbf{B} \rangle - H^*(\mathbf{\Lambda}) \\ &= \max_{\mathbf{\Lambda} \in \mathbb{R}^{n_1 \times n_2}} \|\mathbf{A}\mathbf{B}\|_* + \langle \mathbf{\Lambda}, \mathbf{A}\mathbf{B} \rangle - H^*(\mathbf{\Lambda}) \\ &= \max_{\mathbf{\Lambda} \in \mathbb{R}^{n_1 \times n_2}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}), \end{aligned}$$

where the second equality holds because $H(\cdot)$ is closed and convex with respect to the argument $\mathbf{A}\mathbf{B}$ and the third equality holds by the definition of the conjugate function.

We first show that $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of the Lagrangian $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$, namely, $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$. Using Claim 6, we have $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$. Using Claim 7, we have $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$. Therefore, $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of the Lagrangian $L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$.

We now prove the strong duality. By the fact that $F(\mathbf{A}, \mathbf{B}) = \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$ and that $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$, we have

$$F(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}}) \leq L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}), \ \ \forall \mathbf{A}, \mathbf{B}.$$

where the inequality holds because $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}})$ is a primal-dual saddle point of $L(\cdot, \cdot, \cdot)$. So on the one hand, we have

$$\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = F(\bar{\mathbf{A}}, \bar{\mathbf{B}}) \leq \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}) \leq \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}).$$

On the other hand, by weak duality,

$$\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) \geq \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}).$$

Therefore, $\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda}) = \max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \mathbf{\Lambda})$, i.e., strong duality holds. ■

Claim 6 $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\Lambda} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$.

Proof We only need to show $\mathbf{0} \in \partial_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}}) = \bar{\mathbf{A}}\bar{\mathbf{B}} - \partial_{\mathbf{\Lambda}} H^*(\widetilde{\mathbf{\Lambda}})$, because $L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$ is a concave function of $\mathbf{\Lambda}$ for the fixed $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$. Note that $\bar{\mathbf{A}}\bar{\mathbf{B}} \in \partial_{\mathbf{\Lambda}} H^*(\widetilde{\mathbf{\Lambda}})$ is implied by $\widetilde{\mathbf{\Lambda}} \in \partial H(\bar{\mathbf{A}}\bar{\mathbf{B}})$ by the convexity of function $H(\cdot)$, and

$$\begin{aligned} \partial H(\bar{\mathbf{A}}\bar{\mathbf{B}}) &= \{\mathbf{G} \in \mathbb{R}^{n_1 \times n_2} : \langle \mathbf{G}, \bar{\mathbf{A}}\bar{\mathbf{B}} \rangle \geq \langle \mathbf{G}, \mathbf{Y} \rangle, \text{ for any } \mathbf{Y} \in \mathbb{R}^{n_1 \times n_2} \text{ s.t. } \mathcal{P}_\Omega \mathbf{Y} = \mathcal{P}_\Omega \mathbf{X}^*\} \\ &= \{\mathbf{G} \in \mathbb{R}^{n_1 \times n_2} : \langle \mathbf{G}, \bar{\mathbf{A}}\bar{\mathbf{B}} \rangle \geq \langle \mathbf{G}, \mathbf{Y} \rangle, \text{ for any } \mathbf{Y} \in \mathbb{R}^{n_1 \times n_2} \text{ s.t. } \mathcal{P}_\Omega \mathbf{Y} = \mathcal{P}_\Omega(\bar{\mathbf{A}}\bar{\mathbf{B}})\} \\ &= \Omega. \end{aligned}$$

Therefore, condition (a) immediately implies $\widetilde{\mathbf{\Lambda}} = \operatorname{argmax}_{\mathbf{\Lambda}} L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{\Lambda})$. ■

Claim 7 $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$.

Proof To see $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$ for the fixed $\widetilde{\mathbf{\Lambda}}$ with conditions (b) and (c), we have

$$\begin{aligned} \operatorname*{argmin}_{\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{r \times n_2}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}) &= \operatorname*{argmin}_{\mathbf{A},\mathbf{B}} \|\mathbf{A}\mathbf{B}\|_* - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{A}\mathbf{B} \rangle - H^*(\widetilde{\mathbf{\Lambda}}) \\ &= \operatorname*{argmin}_{\mathbf{A},\mathbf{B}} \|\mathbf{A}\mathbf{B}\|_* - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{A}\mathbf{B} \rangle \\ &= \operatorname*{argmin}_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{r} \sigma_i(\mathbf{A}\mathbf{B}) - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{A}\mathbf{B} \rangle \end{aligned}$$

where the first step follows by definition of $L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$, the second step follows by $\mathbf{\Lambda}$ is fixed, the third step follows by definition of nuclear norm.

Using conditions (b) and (c) in Eqn. (36), we can rewrite $-\widetilde{\mathbf{\Lambda}}$ as follows,

$$-\widetilde{\mathbf{\Lambda}} = \underbrace{\mathbf{U}\mathbf{V}^\top}_{\text{in space } \mathcal{T} \text{ with eigenvalues } 1} + \underbrace{\mathcal{P}_{\mathcal{T}^\perp}(-\widetilde{\mathbf{\Lambda}})}_{\text{in space } \mathcal{T}^\perp \text{ with eigenvalues } <1/2},$$

which implies $\forall i \in [r], \sigma_i(-\widetilde{\mathbf{\Lambda}}) = 1$.

We also need the following von Neumann's trace inequality.

Lemma 25 (von Neumann's trace inequality, Lin and Zhang (2017)) *For any matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ $(m \le n)$, $\mathsf{tr}(\mathbf{A}^\top\mathbf{B}) \le \sum_{i=1}^m \sigma_i(\mathbf{A})\sigma_i(\mathbf{B})$. The equality holds if and only if there exists column orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{A} = \mathbf{U}\mathit{Diag}(\sigma(\mathbf{A}))\mathbf{V}^\top$ and $\mathbf{B} = \mathbf{U}\mathit{Diag}(\sigma(\mathbf{B}))\mathbf{V}^\top$ are the SVDs of $\mathbf{A}$ and $\mathbf{B}$, simultaneously.*

Using von Neumann's trace inequality, we can show

$$\begin{aligned}
\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^r \sigma_i(\mathbf{AB}) - \langle -\widetilde{\mathbf{\Lambda}}, \mathbf{AB}\rangle &\ge \min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^r \sigma_i(\mathbf{AB}) - \sum_{i=1}^r \sigma_i(-\widetilde{\mathbf{\Lambda}})\sigma_i(\mathbf{AB}) \\
&= \min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^r \sigma_i(\mathbf{AB}) - \sum_{i=1}^r \sigma_i(\mathbf{AB}) \qquad (\text{by } \sigma_i(-\mathbf{\Lambda}) = 1, \forall i \in [r]) \\
&= 0,
\end{aligned}$$

Therefore, on one hand, we already have

$$\min_{\mathbf{A}\in\mathbb{R}^{n_1\times r}, \mathbf{B}\in\mathbb{R}^{r\times n_2}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}}) \ge -H^*(\widetilde{\mathbf{\Lambda}}).$$

On the other hand, according to definition of $\mathbf{A}, \mathbf{B}$ and $-\widetilde{\mathbf{\Lambda}}$, we can show

$$\begin{aligned}
L(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \widetilde{\mathbf{\Lambda}}) &= \|\bar{\mathbf{A}}\bar{\mathbf{B}}\|_* - \langle -\widetilde{\mathbf{\Lambda}}, \bar{\mathbf{A}}\bar{\mathbf{B}}\rangle - H^*(\widetilde{\mathbf{\Lambda}}) \\
&= -H^*(\widetilde{\mathbf{\Lambda}}).
\end{aligned}$$

Thus, we can conclude $(\bar{\mathbf{A}}, \bar{\mathbf{B}}) = \operatorname{argmin}_{\mathbf{A},\mathbf{B}} L(\mathbf{A}, \mathbf{B}, \widetilde{\mathbf{\Lambda}})$. ■

To show the dual condition in Lemma 24, intuitively, we need to show that the angle θ between subspace $\mathcal{T}$ and Ψ is small (see Figure 4) for a specific function $H(\cdot)$.

The proof of the dual conditions is similar to that in Appendix D, with some slight differences. We include the proof for completeness.

Lemma 26 *If we can construct an $\mathbf{\Lambda}$ such that*

$$\begin{aligned}
&\text{(a)} \quad \mathbf{\Lambda} \in \Omega, \\
&\text{(b)} \quad \|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top\|_F \le \sqrt{\frac{r}{3n_{(1)}^2}}, \\
&\text{(c)} \quad \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\| < \frac{1}{3},
\end{aligned} \tag{37}$$

then we can construct an $\widetilde{\mathbf{\Lambda}}$ such that (36) *holds with probability at least* $1 - n_{(1)}^{-10}$.

Proof Suppose that Condition (37) holds. Let $\mathbf{Y} = \widetilde{\mathbf{\Lambda}} - \mathbf{\Lambda} \in \Omega$ be the perturbation matrix between $\mathbf{\Lambda}$ and $\widetilde{\mathbf{\Lambda}}$ such that $\mathcal{P}_{\mathcal{T}}(-\widetilde{\mathbf{\Lambda}}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{B}}$. Such a $\mathbf{Y}$ exists by setting $\mathbf{Y} = \mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}})^{-1}(\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top)$. So $\|\mathcal{P}_{\mathcal{T}}\mathbf{Y}\|_F \le \sqrt{\frac{r}{3n_{(1)}^2}}$. We now prove Condition (3) in (36). Observe that

$$\begin{aligned}
\|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| &\le \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}\| + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| \\
&\le \frac{1}{3}\sigma_r(\mathbf{U}\mathbf{V}^\top) + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\|.
\end{aligned} \tag{38}$$

So we only need to show $\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| \le \frac{1}{3}$.

Before proceeding, we begin by introducing $\mathcal{Q}_\Omega : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^{n_1\times n_2}$, which is a normalized version of $\mathcal{P}_\Omega$, defined as

$$\mathcal{Q}_\Omega = p^{-1}\mathcal{P}_\Omega - \mathcal{I}.$$

With this, we have

$$\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}} = p\mathcal{P}_{\mathcal{T}}(\mathcal{I} + \mathcal{Q}_\Omega)\mathcal{P}_{\mathcal{T}}.$$

Note that for any operator $\mathcal{P} : \mathcal{T} \to \mathcal{T}$, we have

$$\mathcal{P}^{-1} = \sum_{k\ge 0}(\mathcal{P}_{\mathcal{T}} - \mathcal{P})^k \text{ whenever } \|\mathcal{P}_{\mathcal{T}} - \mathcal{P}\| < 1.$$

So according to Theorem 13, the operator $p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}})^{-1}$ can be represented as a *convergent* Neumann series

$$p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}})^{-1} = \sum_{k\ge 0}(-1)^k(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_\Omega\mathcal{P}_{\mathcal{T}})^k,$$

because $\|\mathcal{P}_{\mathcal{T}}\mathcal{Q}_\Omega\mathcal{P}_{\mathcal{T}}\| \le \epsilon < \frac{1}{2}$ once $m \ge C\mu n_{(1)} r \log n_{(1)}$ for a sufficiently large absolute constant C. We also note that

$$p(\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_\Omega\mathcal{P}_{\mathcal{T}}) = \mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}},$$

because $\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\mathcal{T}} = 0$. Thus

$$\begin{aligned}
\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}\| &= \|\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}}(\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}})^{-1}(\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top))\| \\
&= \|\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_\Omega\mathcal{P}_{\mathcal{T}}p(\mathcal{P}_{\mathcal{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}})^{-1}((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top))\| \\
&= \|\sum_{k\ge 0}(-1)^k\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_\Omega(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_\Omega\mathcal{P}_{\mathcal{T}})^k((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top))\| \\
&\le \sum_{k\ge 0}\|(-1)^k\mathcal{P}_{\mathcal{T}^\perp}\mathcal{Q}_\Omega(\mathcal{P}_{\mathcal{T}}\mathcal{Q}_\Omega\mathcal{P}_{\mathcal{T}})^k((\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top))\|_F \\
&\le \|\mathcal{Q}_\Omega\|\sum_{k\ge 0}\|\mathcal{P}_{\mathcal{T}}\mathcal{Q}_\Omega\mathcal{P}_{\mathcal{T}}\|^k\|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top))\|_F \\
&\le \frac{4}{p}\|\mathcal{P}_{\mathcal{T}}(-\mathbf{\Lambda}) - \mathbf{U}\mathbf{V}^\top)\|_F \\
&\le \Theta\left(\frac{n_1 n_2}{m}\right)\sqrt{\frac{r}{3n_{(1)}^2}} \\
&\le \frac{1}{3}
\end{aligned}$$

with high probability. The proof is completed. ■

It thus suffices to construct a dual certificate $\mathbf{\Lambda}$ such that all conditions in (37) hold. The proof follows from (Recht, 2011). Partition $\Omega = \Omega_1 \cup \Omega_2 \cup ... \cup \Omega_b$ into b partitions of size q. By assumption, we may choose

$$q \geq \frac{128}{3} C\beta\mu r n_{(1)} \log n_{(1)} \quad \text{and} \quad b \geq \frac{1}{2} \log\left(24^2 n_{(1)}^2\right)$$

for a sufficiently large constant C. Let $\Omega_j \sim \mathsf{Ber}(q)$ denote the set of indices corresponding to the j-th partitions. Define $\mathbf{W}_0 = \mathbf{U}\mathbf{V}^\top$ and set $\mathbf{\Lambda}_k = \frac{n_1 n_2}{q} \sum_{j=1}^k \mathcal{P}_{\Omega_j}(\mathbf{W}_{j-1})$, $\mathbf{W}_k = \mathbf{U}\mathbf{V}^\top - \mathcal{P}_{\mathcal{T}}(\mathbf{\Lambda}_k)$ for $k = 1, 2, ..., b$. Then by Theorem 13,

$$\begin{aligned}\|\mathbf{W}_k\|_F &= \left\|\mathbf{W}_{k-1} - \frac{n_1 n_2}{q}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega_k}(\mathbf{W}_{k-1})\right\|_F = \left\|\left(\mathcal{P}_{\mathcal{T}} - \frac{n_1 n_2}{q}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega_k}\mathcal{P}_{\mathcal{T}}\right)(\mathbf{W}_{k-1})\right\|_F \\ &\leq \frac{1}{2\kappa}\|\mathbf{W}_{k-1}\|_F.\end{aligned}$$

So it follows that $\|\mathbf{W}_b\|_F \leq (2\kappa)^{-b}\|\mathbf{W}_0\|_F \leq (2\kappa)^{-b}\sqrt{r} \leq \sqrt{\frac{r}{24^2 n_{(1)}^2}}$.

The following lemma together implies the strong duality of (18) straightforwardly.

Lemma 27 *Under the assumptions of Theorem 8, the dual certification* $\mathbf{W}_b$ *obeys the dual condition* (37) *with probability at least* $1 - n_{(1)}^{-10}$.

Proof It is well known that for matrix completion, the Uniform model $\Omega \sim \mathsf{Uniform}(m)$ is equivalent to the Bernoulli model $\Omega \sim \mathsf{Ber}(p)$, where each element in $[n_1] \times [n_2]$ is included with probability $p = \Theta(m/(n_1 n_2))$ independently; see Appendix I for a brief justification. By the equivalence, we can suppose $\Omega \sim \mathsf{Ber}(p)$.

Observe that by Lemma 16,

$$\|\mathbf{W}_j\|_\infty \leq \left(\frac{1}{2}\right)^j \|\mathbf{U}\mathbf{V}^\top\|_\infty,$$

and by Lemma 17,

$$\|\mathbf{W}_j\|_{\infty,2} \leq \frac{1}{2}\sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{W}_{j-1}\|_\infty + \frac{1}{2}\|\mathbf{W}_{j-1}\|_{\infty,2}.$$

Combining the two leads to

$$\begin{aligned}&\|\mathbf{W}_j\|_{\infty,2} \\ &\leq \left(\frac{1}{2}\right)^j \sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{U}\mathbf{V}^\top\|_\infty + \frac{1}{2}\|\mathbf{W}_{j-1}\|_{\infty,2} \\ &\leq j\left(\frac{1}{2}\right)^j \sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{U}\mathbf{V}^\top\|_\infty + \left(\frac{1}{2}\right)^j \|\mathbf{U}\mathbf{V}^\top\|_{\infty,2}.\end{aligned}$$

Therefore,

$$\begin{aligned}&\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}_b\| \\ &\leq \sum_{j=1}^b \|\frac{n_1 n_2}{q}\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega_j}\mathbf{W}_{j-1}\| \\ &= \sum_{j=1}^b \|\mathcal{P}_{\mathcal{T}^\perp}(\frac{n_1 n_2}{q}\mathcal{P}_{\Omega_j}\mathbf{W}_{j-1} - \mathbf{W}_{j-1})\| \\ &\leq \sum_{j=1}^b \|(\frac{n_1 n_2}{q}\mathcal{P}_{\Omega_j} - \mathcal{I})(\mathbf{W}_{j-1})\|.\end{aligned}$$

Let p denote $\Theta\left(\frac{q}{n_1 n_2}\right)$. By Lemma 15,

$$\begin{aligned}&\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{\Lambda}_b\| \\ &\leq C_0'\frac{\log n_{(1)}}{p}\sum_{j=1}^b \|\mathbf{W}_{j-1}\|_\infty + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sum_{j=1}^b \|\mathbf{W}_{j-1}\|_{\infty,2} \\ &\leq C_0'\frac{\log n_{(1)}}{p}\sum_{j=1}^b \left(\frac{1}{2}\right)^j \|\mathbf{U}\mathbf{V}^\top\|_\infty \\ &\quad + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sum_{j=1}^b \left[j\left(\frac{1}{2}\right)^j \sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{U}\mathbf{V}^\top\|_\infty + \left(\frac{1}{2}\right)^j \|\mathbf{U}\mathbf{V}^\top\|_{\infty,2}\right] \\ &\leq C_0'\frac{\log n_{(1)}}{p}\|\mathbf{U}\mathbf{V}^\top\|_\infty + 2C_0'\sqrt{\frac{\log n_{(1)}}{p}}\sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{U}\mathbf{V}^\top\|_\infty + C_0'\sqrt{\frac{\log n_{(1)}}{p}}\|\mathbf{U}\mathbf{V}^\top\|_{\infty,2}.\end{aligned}$$

We note the facts that (assume without loss of generality that $n_2 \geq n_1$)

$$\|\mathbf{U}\mathbf{V}^\top\|_{\infty,2} = \max_i \|\mathbf{e}_i^T\mathbf{U}\mathbf{V}^T\|_2 = \max_i \|\mathbf{e}_i^T\mathbf{U}\|_2 \leq \sqrt{\frac{\mu r}{n_1}},$$

and that

$$\begin{aligned}\|\mathbf{U}\mathbf{V}^\top\|_\infty &= \max_{ij}\langle\mathbf{U}\mathbf{V}^\top, \mathbf{e}_i\mathbf{e}_j^T\rangle = \max_{ij}\langle\mathbf{e}_i^T\mathbf{U}, \mathbf{e}_j^T\mathbf{V}\rangle \\ &\leq \max_{ij}\|\mathbf{e}_i^T\mathbf{U}\|_2\|\mathbf{e}_j^T\mathbf{V}\|_2 \leq \frac{\mu r}{\sqrt{n_1 n_2}}.\end{aligned}$$

Substituting $p = \Theta\left(\frac{\mu r n_{(1)}\log^2(n_{(1)})}{n_1 n_2}\right)$, we obtain $\|\mathcal{P}_{\mathcal{T}^\perp}\widetilde{\mathbf{\Lambda}}\| < \frac{1}{3}$. The proof is completed. ■

We note that the bi-dual problem of problem (34) is problem (18); see Appendix H.2 for the details. By Lemmas 24, 26 and 27, we have $\widetilde{\mathbf{A}}\widetilde{\mathbf{B}} = \widetilde{\mathbf{X}}$. We also notice from Theorem 1 of Chen (2015) that $\widetilde{\mathbf{X}} = \mathbf{X}^*$. The proof of Theorem 8 is then completed.

F. Proof of Theorem 9

Theorem 9 (Robust PCA. Restated). *Suppose $\mathbf{X}^*$ is $n_1 \times n_2$, obeys incoherence* (5) *and* (20). *Assume that the support set Ω of $\mathbf{S}^*$ is uniformly distributed among all sets of cardinality m. Then with probability at least $1 - cn_{(1)}^{-10}$, the output of the optimization problem*

$$(\widetilde{\mathbf{X}}, \widetilde{\mathbf{S}}) = \operatorname*{argmin}_{\mathbf{X},\mathbf{S}} \|\mathbf{X}\|_{r*} + \lambda\|\mathbf{S}\|_1, \quad \text{s.t.} \quad \mathbf{D} = \mathbf{X} + \mathbf{S}, \tag{39}$$

with $\lambda = \frac{\sigma_r(\mathbf{X}^*)}{\sqrt{n_{(1)}}}$ is exact, i.e., $\widetilde{\mathbf{X}} = \mathbf{X}^*$ and $\widetilde{\mathbf{S}} = \mathbf{S}^*$, provided that $\mathsf{rank}(\mathbf{X}^*) \le \frac{\rho_r n_{(2)}}{\mu \log^2 n_{(1)}}$ and $m \le \rho_s n_1 n_2$, where c, ρ_r, and ρ_s are all positive absolute constants, and function $\|\cdot\|_{r*}$ is given by (21).

F.1. Dual Certificates

Lemma 28 *Assume that $\|\mathcal{P}_\Omega \mathcal{P}_\mathcal{T}\| \le 1/2$ and $\lambda < \sigma_r(\mathbf{X}^*)$. Then $(\mathbf{X}^*, \mathbf{S}^*)$ is the unique solution to problem* (9) *if there exists a pair $(\mathbf{W}, \mathbf{F})$ for which*

$$\mathbf{X}^* + \mathbf{W} = \lambda(\mathsf{sign}(\mathbf{S}^*) + \mathbf{F} + \mathcal{P}_\Omega \mathbf{K}),$$

where $\mathbf{W} \in \mathcal{T}^\perp$, $\|\mathbf{W}\| \le \frac{\sigma_r(\mathbf{X}^*)}{2}$, $\mathbf{F} \in \Omega^\perp$, $\|\mathbf{F}\|_\infty \le \frac{1}{2}$, and $\|\mathcal{P}_\Omega \mathbf{K}\|_F \le \frac{1}{4}$.

Proof Let $(\mathbf{X}^* + \mathbf{H}, \mathbf{S}^* - \mathbf{H})$ be any optimal solution to problem (39). By the definition of the subgradient, the inequality follows

$$\begin{aligned}
&\|\mathbf{X}^* + \mathbf{H}\|_{r*} + \lambda\|\mathbf{S}^* - \mathbf{H}\|_1 \\
&\ge \|\mathbf{X}^*\|_{r*} + \lambda\|\mathbf{S}^*\|_1 + \langle \mathbf{X}^* + \mathbf{W}^*, \mathbf{H}\rangle - \lambda\langle \mathsf{sign}(\mathbf{S}^*) + \mathbf{F}^*, \mathbf{H}\rangle \\
&= \|\mathbf{X}^*\|_{r*} + \lambda\|\mathbf{S}^*\|_1 + \langle \mathbf{X}^* - \lambda\mathsf{sign}(\mathbf{S}^*), \mathbf{H}\rangle + \langle \mathbf{W}^*, \mathbf{H}\rangle - \lambda\langle \mathbf{F}^*, \mathbf{H}\rangle \\
&= \|\mathbf{X}^*\|_{r*} + \lambda\|\mathbf{S}^*\|_1 + \langle \mathbf{X}^* - \lambda\mathsf{sign}(\mathbf{S}^*), \mathbf{H}\rangle + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_* + \lambda\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_1 \\
&= \|\mathbf{X}^*\|_{r*} + \lambda\|\mathbf{S}^*\|_1 + \langle \lambda\mathbf{F} + \lambda\mathcal{P}_\Omega\mathbf{K} - \mathbf{W}, \mathbf{H}\rangle + \sigma_r(\mathbf{X}^*)\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_* + \lambda\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_1 \\
&= \|\mathbf{X}^*\|_{r*} + \lambda\|\mathbf{S}^*\|_1 + \frac{\sigma_r(\mathbf{X}^*)}{2}\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_* + \frac{\lambda}{2}\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_1 - \frac{\lambda}{4}\|\mathcal{P}_\Omega\mathbf{H}\|_F.
\end{aligned}$$

We note that

$$\begin{aligned}
\|\mathcal{P}_\Omega\mathbf{H}\|_F &\le \|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathbf{H}\|_F + \|\mathcal{P}_\Omega\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_F \\
&\le \frac{1}{2}\|\mathbf{H}\|_F + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_F \\
&\le \frac{1}{2}\|\mathcal{P}_\Omega\mathbf{H}\|_F + \frac{1}{2}\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_F + \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_F,
\end{aligned}$$

which implies that $\frac{\lambda}{4}\|\mathcal{P}_\Omega\mathbf{H}\|_F \le \frac{\lambda}{4}\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_F + \frac{\lambda}{2}\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_F \le \frac{\lambda}{4}\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_1 + \frac{\lambda}{2}\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_*$. Therefore,

$$\begin{aligned}
\|\mathbf{X}^* + \mathbf{H}\|_{r*} + \lambda\|\mathbf{S}^* - \mathbf{H}\|_1 &\ge \|\mathbf{X}^*\|_{r*} + \lambda\|\mathbf{S}^*\|_1 + \frac{\sigma_r(\mathbf{X}^*) - \lambda}{2}\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_* + \frac{\lambda}{4}\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_1 \\
&\ge \|\mathbf{X}^* + \mathbf{H}\|_{r*} + \lambda\|\mathbf{S}^* - \mathbf{H}\|_1 + \frac{\sigma_r(\mathbf{X}^*) - \lambda}{2}\|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{H}\|_* + \frac{\lambda}{4}\|\mathcal{P}_{\Omega^\perp}\mathbf{H}\|_1,
\end{aligned}$$

where the second inequality holds because $(\mathbf{X}^* + \mathbf{H}, \mathbf{S}^* - \mathbf{H})$ is optimal. Thus $\mathbf{H} \in \mathcal{T} \cap \Omega$. Note that $\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| < 1$ implies $\mathcal{T} \cap \Omega = \{0\}$. This completes the proof. ■

According to Lemma 28, to show the exact recoverability of problem (39), it is sufficient to find an appropriate $\mathbf{W}$ for which

$$\begin{cases}
\mathbf{W} \in \mathcal{T}^\perp, \\
\|\mathbf{W}\| \le \frac{\sigma_r(\mathbf{X}^*)}{2}, \\
\|\mathcal{P}_\Omega(\mathbf{X}^* + \mathbf{W} - \lambda\mathsf{sign}(S^*))\|_F \le \frac{\lambda}{4}, \\
\|\mathcal{P}_{\Omega^\perp}(\mathbf{X}^* + \mathbf{W})\|_\infty \le \frac{\lambda}{2}.
\end{cases} \tag{40}$$

F.2. Dual Certification by Least Squares and the Golfing Scheme

The remainder of the proof is to construct $\mathbf{W}$ such that the dual condition (40) holds true. Before introducing our construction, we assume $\Omega \sim \mathsf{Ber}(p)$, or equivalently $\Omega^\perp \sim \mathsf{Ber}(1-p)$, where p is allowed be as large as an absolute constant. Note that $\Omega^\perp$ has the same distribution as that of $\Omega_1 \cup \Omega_2 \cup ... \cup \Omega_{j_0}$, where the Ω_j's are drawn independently with replacement from $\mathsf{Ber}(q)$, $j_0 = \lceil \log n_{(1)} \rceil$, and q obeys $p = (1-q)^{j_0}$ ($q = \Omega(1/\log n_{(1)})$ implies $p = \mathcal{O}(1)$). We construct $\mathbf{W}$ based on such a distribution.

Our construction separates $\mathbf{W}$ into two terms: $\mathbf{W} = \mathbf{W}^L + \mathbf{W}^S$. To construct $\mathbf{W}^L$, we apply the golfing scheme introduced in (Gross, 2011; Recht, 2011). Specifically, $\mathbf{W}^L$ is constructed by an inductive procedure:

$$\begin{aligned}
\mathbf{Y}_j = \mathbf{Y}_{j-1} + q^{-1}\mathcal{P}_{\Omega_j}\mathcal{P}_\mathcal{T}(\mathbf{X}^* - \mathbf{Y}_{j-1}),\ \mathbf{Y}_0 = \mathbf{0}, \\
\mathbf{W}^L = \mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}_{j_0}.
\end{aligned} \tag{41}$$

To construct $\mathbf{W}^S$, we apply the method of least squares by Candès et al. (2011), which is

$$\mathbf{W}^S = \lambda\mathcal{P}_{\mathcal{T}^\perp}\sum_{k\ge 0}(\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^k\mathsf{sign}(\mathbf{S}^*). \tag{42}$$

Note that $\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| \le 1/2$. Thus $\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega\| \le 1/4$ and the Neumann series in (42) is well-defined. Observe that $\mathcal{P}_\Omega\mathbf{W}^S = \lambda(\mathcal{P}_\Omega - \mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)(\mathcal{P}_\Omega - \mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^{-1}\mathsf{sign}(\mathbf{S}^*) = \lambda\mathsf{sign}(\mathbf{S}^*)$. So to prove the dual condition (40), it suffices to show that

$$\begin{aligned}
&\text{(a)} \quad \|\mathbf{W}^L\| \le \frac{\sigma_r(\mathbf{X}^*)}{4}, \\
&\text{(b)} \quad \|\mathcal{P}_\Omega(\mathbf{X}^* + \mathbf{W}^L)\|_F \le \frac{\lambda}{4}, \\
&\text{(c)} \quad \|\mathcal{P}_{\Omega^\perp}(\mathbf{X}^* + \mathbf{W}^L)\|_\infty \le \frac{\lambda}{4},
\end{aligned} \tag{43}$$

and

$$\begin{aligned}
&\text{(d)} \quad \|\mathbf{W}^S\| \le \frac{\sigma_r(\mathbf{X}^*)}{4}, \\
&\text{(e)} \quad \|\mathcal{P}_{\Omega^\perp}\mathbf{W}^S\|_\infty \le \frac{\lambda}{4}.
\end{aligned} \tag{44}$$

F.3. Proof of Dual Conditions

Since we have constructed the dual certificate $\mathbf{W}$, the remainder is to show that $\mathbf{W}$ obeys dual conditions (43) and (44) with high probability. We have the following.

Lemma 29 *Assume $\Omega_j \sim \mathsf{Ber}(q)$, $j = 1, 2, ..., j_0$, and $j_0 = 2\lceil \log n_{(1)} \rceil$. Then under the other assumptions of Theorem 9, $\mathbf{W}^L$ given by* (41) *obeys dual condition* (43).

Proof Let $\mathbf{Z}_j = \mathcal{P}_{\mathcal{T}}(\mathbf{X}^* - \mathbf{Y}_j) \in \mathcal{T}$. Then we have

$$\mathbf{Z}_j = \mathcal{P}_{\mathcal{T}}\mathbf{Z}_{j-1} - q^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega_j}\mathcal{P}_{\mathcal{T}}\mathbf{Z}_{j-1} = (\mathcal{P}_{\mathcal{T}} - q^{-1}\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega_j}\mathcal{P}_{\mathcal{T}})\mathbf{Z}_{j-1},$$

and $\mathbf{Y}_j = \sum_{k=1}^{j} q^{-1}\mathcal{P}_{\Omega_k}\mathbf{Z}_{k-1} \in \Omega^{\perp}$. We set $q = \Omega(\epsilon^{-2}\mu r \log n_{(1)}/n_{(2)})$.

Proof of (a). It holds that

$$\begin{aligned}
\|\mathbf{W}^L\| = \|\mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}_{j_0}\| &\le \sum_{k=1}^{j_0} \|q^{-1}\mathcal{P}_{\mathcal{T}^\perp}\mathcal{P}_{\Omega_k}\mathbf{Z}_{k-1}\| \\
&= \sum_{k=1}^{j_0} \|\mathcal{P}_{\mathcal{T}^\perp}(q^{-1}\mathcal{P}_{\Omega_k}\mathbf{Z}_{k-1} - \mathbf{Z}_{k-1})\| \\
&\le \sum_{k=1}^{j_0} \|q^{-1}\mathcal{P}_{\Omega_k}\mathbf{Z}_{k-1} - \mathbf{Z}_{k-1}\| \\
&\le C_0' \left(\frac{\log n_{(1)}}{q} \sum_{k=1}^{j_0} \|\mathbf{Z}_{k-1}\|_\infty + \sqrt{\frac{\log n_{(1)}}{q}} \sum_{k=1}^{j_0} \|\mathbf{Z}_{k-1}\|_{\infty,2} \right). \quad \text{(by Lemma 15)}
\end{aligned}$$

We note that by Lemma 16 and Lemma 17, respectively,

$$\begin{aligned}
\|\mathbf{Z}_{k-1}\|_\infty &\le \left(\frac{1}{2}\right)^{k-1} \|\mathbf{Z}_0\|_\infty, \\
\|\mathbf{Z}_{k-1}\|_{\infty,2} &\le \frac{1}{2}\sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{Z}_{k-2}\|_\infty + \frac{1}{2}\|\mathbf{Z}_{k-2}\|_{\infty,2}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\mathbf{Z}_{k-1}\|_{\infty,2} &\le \left(\frac{1}{2}\right)^{k-1} \sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{Z}_0\|_\infty + \frac{1}{2}\|\mathbf{Z}_{k-2}\|_{\infty,2} \\
&\le (k-1)\left(\frac{1}{2}\right)^{k-1} \sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{Z}_0\|_\infty + \left(\frac{1}{2}\right)^{k-1} \|\mathbf{Z}_0\|_{\infty,2},
\end{aligned}$$

and so we have

$$\begin{aligned}
\|\mathbf{W}^L\| &\le C_0' \Bigg[\frac{\log n_{(1)}}{q} \sum_{k=1}^{j_0} \left(\frac{1}{2}\right)^{k-1} \|\mathbf{Z}_0\|_\infty \\
&\quad + \sqrt{\frac{\log n_{(1)}}{q}} \sum_{k=1}^{j_0} \left((k-1)\left(\frac{1}{2}\right)^{k-1} \sqrt{\frac{n_{(1)}}{\mu r}}\|\mathbf{Z}_0\|_\infty + \left(\frac{1}{2}\right)^{k-1} \|\mathbf{Z}_0\|_{\infty,2} \right) \Bigg] \\
&\le 2C_0' \left[\frac{\log n_{(1)}}{q}\|\mathbf{X}^*\|_\infty + \sqrt{\frac{n_{(1)}\log n_{(1)}}{q\mu r}}\|\mathbf{X}^*\|_\infty + \sqrt{\frac{\log n_{(1)}}{q}}\|\mathbf{X}^*\|_{\infty,2} \right] \\
&\le 2C_0 \left[\frac{n_{(2)}}{\mu r}\|\mathbf{X}^*\|_\infty + \frac{\sqrt{n_1 n_2}}{\mu r}\|\mathbf{X}^*\|_\infty + \sqrt{\frac{n_{(2)}}{\mu r}}\|\mathbf{X}^*\|_{\infty,2} \right] \quad (\text{since } q = \Omega(\mu r \log n_{(1)})/n_{(2)}) \\
&\le \frac{\sigma_r(\mathbf{X}^*)}{4}, \quad \text{(by incoherence (20))}
\end{aligned}$$

where we have used the fact that

$$\|\mathbf{X}^*\|_{\infty,2} \le \sqrt{n_{(1)}}\|\mathbf{X}^*\|_\infty \le \sqrt{\frac{\mu r}{n_{(2)}}}\sigma_r(\mathbf{X}^*).$$

Proof of (b). Because $\mathbf{Y}_{j_0} \in \Omega^\perp$, we have $\mathcal{P}_\Omega(\mathbf{X}^* + \mathcal{P}_{\mathcal{T}^\perp}\mathbf{Y}_{j_0}) = \mathcal{P}_\Omega(\mathbf{X}^* - \mathcal{P}_{\mathcal{T}}\mathbf{Y}_{j_0}) = \mathcal{P}_\Omega\mathbf{Z}_{j_0}$. It then follows from Theorem 13 that

$$\begin{aligned}
\|\mathbf{Z}_{j_0}\|_F &\le t^{j_0}\|\mathbf{X}^*\|_F \\
&\le t^{j_0}\sqrt{n_1 n_2}\|\mathbf{X}^*\|_\infty \\
&\le t^{j_0}\sqrt{n_1 n_2}\sqrt{\frac{\mu r}{n_1 n_2}}\sigma_r(\mathbf{X}^*) \\
&\le \frac{\lambda}{8}. \quad (t^{j_0} \le e^{-2\log n_{(1)}} \le n_{(1)}^{-2})
\end{aligned}$$

Proof of (c). By definition, $\mathbf{X}^* + \mathbf{W}^L = \mathbf{Z}_{j_0} + \mathbf{Y}_{j_0}$. Since we have shown $\|\mathbf{Z}_{j_0}\|_F \le \lambda/8$, it suffices to prove $\|\mathbf{Y}_{j_0}\|_\infty \le \lambda/8$. We have

$$\begin{aligned}
\|\mathbf{Y}_{j_0}\|_\infty &\le q^{-1}\sum_{k=1}^{j_0}\|\mathcal{P}_{\Omega_k}\mathbf{Z}_{k-1}\|_\infty \\
&\le q^{-1}\sum_{k=1}^{j_0}\epsilon^{k-1}\|\mathbf{X}^*\|_\infty \quad \text{(by Lemma 16)} \\
&\le \frac{n_{(2)}\epsilon^2}{C_0\mu r \log n_{(1)}}\sqrt{\frac{\mu r}{n_1 n_2}}\sigma_r(\mathbf{X}^*) \quad \text{(by incoherence (20))} \\
&\le \frac{\lambda}{8},
\end{aligned}$$

if we choose $\epsilon = C\left(\frac{\mu r(\log n_{(1)})^2}{n_{(2)}}\right)^{1/4}$ for an absolute constant C. This can be true once the constant ρ_r is sufficiently small. ■

We now prove that $\mathbf{W}^S$ given by (42) obeys dual condition (44).

Lemma 30 *Assume $\Omega \sim \mathsf{Ber}(p)$. Then under the other assumptions of Theorem 9, $\mathbf{W}^S$ given by* (42) *obeys dual condition* (44).

Proof According to the standard de-randomization argument (Candès et al., 2011), it is equivalent to studying the case when the signs δ_{ij} of $\mathbf{S}^*_{ij}$ are independently distributed as

$$\delta_{ij} = \begin{cases} 1, & \text{w.p. } p/2, \\ 0, & \text{w.p. } 1-p, \\ -1, & \text{w.p. } p/2. \end{cases}$$

Proof of (d). Recall that

$$\begin{aligned}\mathbf{W}^S &= \lambda \mathcal{P}_{\mathcal{T}^\perp} \sum_{k\geq 0} (\mathcal{P}_\Omega \mathcal{P}_\mathcal{T} \mathcal{P}_\Omega)^k \mathsf{sign}(\mathbf{S}^*) \\ &= \lambda \mathcal{P}_{\mathcal{T}^\perp} \mathsf{sign}(\mathbf{S}^*) + \lambda \mathcal{P}_{\mathcal{T}^\perp} \sum_{k\geq 1} (\mathcal{P}_\Omega \mathcal{P}_\mathcal{T} \mathcal{P}_\Omega)^k \mathsf{sign}(\mathbf{S}^*).\end{aligned}$$

To bound the first term, we have $\|\mathsf{sign}(\mathbf{S}^*)\| \leq 4\sqrt{n_{(1)}p}$ as shown in (Vershynin, 2010). So $\|\lambda \mathcal{P}_{\mathcal{T}^\perp} \mathsf{sign}(\mathbf{S}^*)\| \leq \lambda \|\mathsf{sign}(\mathbf{S}^*)\| \leq 4\sqrt{p}\sigma_r(\mathbf{X}^*) \leq \sigma_r(\mathbf{X}^*)/8$.

We now bound the second term. Let $\mathcal{G} = \sum_{k\geq 1} (\mathcal{P}_\Omega \mathcal{P}_\mathcal{T} \mathcal{P}_\Omega)^k$, which is self-adjoint, and denote by N_{n_1} and N_{n_2} the $\frac{1}{2}$-nets of $\mathbb{S}^{n_1-1}$ and $\mathbb{S}^{n_1-1}$ of sizes at most 6^{n_1} and 6^{n_2}, respectively (Ledoux, 2005). Vershynin (2010) has shown that

$$\begin{aligned}\|\mathcal{G}(\mathsf{sign}(\mathbf{S}^*))\| &= \sup_{\mathbf{x}\in\mathbb{S}^{n_2-1}, \mathbf{y}\in\mathbb{S}^{n_1-1}} \langle \mathcal{G}(\mathbf{y}\mathbf{x}^T), \mathsf{sign}(\mathbf{S}^*)\rangle \\ &\leq 4 \sup_{\mathbf{x}\in N_{n_2}, \mathbf{y}\in N_{n_1}} \langle \mathcal{G}(\mathbf{y}\mathbf{x}^T), \mathsf{sign}(\mathbf{S}^*)\rangle.\end{aligned}$$

Consider the random variable $X(\mathbf{x},\mathbf{y}) = \langle \mathcal{G}(\mathbf{y}\mathbf{x}^T), \mathsf{sign}(\mathbf{S}^*)\rangle$ which has zero expectation. By Hoeffding's inequality, we have

$$\Pr(|X(\mathbf{x},\mathbf{y})| > t|\Omega) \leq 2\exp\left(-\frac{2t^2}{\|\mathcal{G}(\mathbf{x}\mathbf{y}^T)\|_F^2}\right) \leq 2\exp\left(-\frac{2t^2}{\|\mathcal{G}\|^2}\right).$$

Therefore, by a union bound,

$$\Pr(\|\mathcal{G}(\mathsf{sign}(\mathbf{S}^*))\| > t|\Omega) \leq 2\times 6^{n_1+n_2}\exp\left(-\frac{t^2}{8\|\mathcal{G}\|^2}\right).$$

Note that conditioned on the event $\{\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| \leq \sigma\}$, we have $\|\mathcal{G}\| = \left\|\sum_{k\geq 1}(\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^k\right\| \leq \frac{\sigma^2}{1-\sigma^2}$. So

$$\begin{aligned}&\Pr(\lambda\|\mathcal{G}(\mathsf{sign}(\mathbf{S}^*))\| > t) \\ &\leq 2\times 6^{n_1+n_2}\exp\left(-\frac{t^2}{8\lambda^2}\left(\frac{1-\sigma^2}{\sigma^2}\right)^2\right)\Pr(\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| \leq \sigma) + \Pr(\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| > \sigma).\end{aligned}$$

The following lemma guarantees that event $\{\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| \leq \sigma\}$ holds with high probability for a very small absolute constant σ.

Lemma 31 (Candès et al. (2011), Corollary 2.7) *Suppose that $\Omega \sim \mathsf{Ber}(p)$ and incoherence* (5) *holds. Then with probability at least $1 - n_{(1)}^{-10}$, $\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\|^2 \leq p + \epsilon$, provided that $1 - p \geq C_0\epsilon^{-2}\mu r \log n_{(1)}/n_{(2)}$ for an absolute constant C_0.*

Setting $t = \frac{\sigma_r(\mathbf{X}^*)}{8}$, this completes the proof of (d). ■

Proof of (e). Recall that $\mathbf{W}^S = \lambda\mathcal{P}_{\mathcal{T}^\perp}\sum_{k\geq 0}(\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^k\mathsf{sign}(\mathbf{S}^*)$ and so

$$\begin{aligned}\mathcal{P}_{\Omega^\perp}\mathbf{W}^S &= \lambda\mathcal{P}_{\Omega^\perp}(\mathcal{I} - \mathcal{P}_\mathcal{T})\sum_{k\geq 0}(\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^k\mathsf{sign}(\mathbf{S}^*) \\ &= -\lambda\mathcal{P}_{\Omega^\perp}\mathcal{P}_\mathcal{T}\sum_{k\geq 0}(\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^k\mathsf{sign}(\mathbf{S}^*).\end{aligned}$$

Then for any $(i,j) \in \Omega^\perp$, we have

$$\mathbf{W}^S_{ij} = \langle \mathbf{W}^S, \mathbf{e}_i\mathbf{e}_j^T\rangle = \left\langle \lambda\mathsf{sign}(\mathbf{S}^*), -\sum_{k\geq 0}(\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^k\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}(\mathbf{e}_i\mathbf{e}_j^T)\right\rangle.$$

Let $X(i,j) = -\sum_{k\geq 0}(\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\mathcal{P}_\Omega)^k\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}(\mathbf{e}_i\mathbf{e}_j^T)$. By Hoeffding's inequality and a union bound,

$$\Pr\left(\sup_{ij}|\mathbf{W}^S_{ij}| > t|\Omega\right) \leq 2\sum_{ij}\exp\left(-\frac{2t^2}{\lambda^2\|X(i,j)\|_F^2}\right).$$

We note that conditioned on the event $\{\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| \leq \sigma\}$, for any $(i,j) \in \Omega^\perp$,

$$\begin{aligned}\|X(i,j)\|_F &\leq \frac{1}{1-\sigma^2}\sigma\|\mathcal{P}_\mathcal{T}(\mathbf{e}_i\mathbf{e}_j^T)\|_F \\ &\leq \frac{1}{1-\sigma^2}\sigma\sqrt{1 - \|\mathcal{P}_{\mathcal{T}^\perp}(\mathbf{e}_i\mathbf{e}_j^T)\|_F^2} \\ &= \frac{1}{1-\sigma^2}\sigma\sqrt{1 - \|(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{e}_i\|_2^2\|(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{e}_j\|_2^2} \\ &\leq \frac{1}{1-\sigma^2}\sigma\sqrt{1 - \left(1 - \frac{\mu r}{n_1}\right)\left(1 - \frac{\mu r}{n_2}\right)} \\ &\leq \frac{1}{1-\sigma^2}\sigma\sqrt{\frac{\mu r}{n_1} + \frac{\mu r}{n_2}}.\end{aligned}$$

Then unconditionally,

$$\begin{aligned}\Pr\left(\sup_{ij}|\mathbf{W}^S_{ij}| > t\right) \leq& 2n_1n_2\exp\left(-\frac{2t^2}{\lambda^2}\frac{(1-\sigma^2)^2 n_1 n_2}{\sigma^2\mu r(n_1+n_2)}\right)\Pr(\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| \leq \sigma) \\ &+ \Pr(\|\mathcal{P}_\Omega\mathcal{P}_\mathcal{T}\| > \sigma).\end{aligned}$$

By Lemma 31 and setting $t = \lambda/4$, the proof of (e) is completed.

G. Proof of Theorem 10

Our computational lower bound for problem (**P**) (and (**P'**)) assumes the average-case hardness of SAT, which has been used in (Razenshteyn et al., 2016; Song et al., 2019).

Conjecture 32 (Random 4-SAT, Razenshteyn et al. (2016)) *Let $c > \ln 2$ be a constant. Consider a random 4-SAT formula on n variables in which each clause has 4 literals, and in which each of the $16n^4$ clauses is picked independently with probability c/n^3. Then any algorithm which always outputs 1 when the random formula is satisfiable, and outputs 0 with probability at least $1/2$ when the random formula is unsatisfiable, must run in $2^{c'n}$ time on some input, where $c' > 0$ is an absolute constant.*

Based on Conjecture 32, we have the following computational lower bound for problem (**P**) (and (**P'**)). We show that problem (**P**) (and (**P'**)) is in general hard for deterministic algorithms. If we additionally assume BPP = P, then the same conclusion holds for randomized algorithms with high probability.

Theorem 10 (Computational Lower Bound. Restated)**.** *Assume Conjecture 32. Then there exists an absolute constant $\epsilon_0 > 0$ for which any algorithm that achieves $(1+\epsilon)$OPT in objective function value for problem* (**P**) *(and* (**P'**)*) with $\epsilon \le \epsilon_0$, and with constant probability, requires $2^{\Omega(n_1+n_2)}$ time, where* OPT *is the optimum. If in addition,* BPP = P*, then the same conclusion holds for randomized algorithms succeeding with probability at least $2/3$.*

Proof Theorem 10 is proved by using the hypothesis that random 4-SAT is hard to show hardness of the Maximum Edge Biclique problem for deterministic algorithms. The same construction is used for both problem (**P**) and problem (**P'**), so we only present the proof for the former in the following.

Definition 33 (Maximum Edge Biclique) *The problem is*

Input: An n-by-n bipartite graph G.

Output: A k_1-by-k_2 complete bipartite subgraph of G, such that $k_1 \cdot k_2$ is maximized.

Goerdt and Lanka (2004) showed that under the random 4-SAT assumption there exist two constants $1 > \epsilon_1 > \epsilon_2 > 0$ such that no efficient deterministic algorithm is able to distinguish between bipartite graphs $G(U, V, E)$ with $|U| = |V| = n$ which have a clique of size $\ge (n/16)^2(1+\epsilon_1)$ and those in which all bipartite cliques are of size $\le (n/16)^2(1+\epsilon_2)$. The reduction uses a bipartite graph G with at least tn^2 edges with large probability, for a constant t.

Given a bipartite graph $G(U, V, E)$, define $H(\cdot)$ as follows. Define the matrix $\mathbf{Y}$ and $\mathbf{W}$: $\mathbf{Y}_{ij} = 1$ if edge $(U_i, V_j) \in E$, $\mathbf{Y}_{ij} = 0$ if edge $(U_i, V_j) \notin E$; $\mathbf{W}_{ij} = 1$ if edge $(U_i, V_j) \in E$, and $\mathbf{W}_{ij} = \mathrm{poly}(n)$ if edge $(U_i, V_j) \notin E$. Choose a large enough constant $\beta > 0$ and let $H(\mathbf{AB}) = \beta \sum_{ij} \mathbf{W}_{ij}^2 (\mathbf{Y}_{ij} - (\mathbf{AB})_{ij})^2$. Now, if there exists a biclique in G with at least $(n/16)^2(1+\epsilon_1)$ edges, then the number of remaining edges is at most $tn^2 - (n/16)^2(1+\epsilon_1)$, and so the solution to $\min H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2$ has cost at most $\beta[tn^2 - (n/16)^2(1+\epsilon_1)] + n^2$. On the other hand, if there does not exist a biclique that has more than $(n/16)^2(1+\epsilon_2)$ edges, then the number of remaining edges is at least $(n/16)^2(1+\epsilon_2)$, and so any solution to $\min H(\mathbf{AB}) + \frac{1}{2}\|\mathbf{AB}\|_F^2$ has cost at least $\beta[tn^2 - (n/16)^2(1+\epsilon_2)]$. Choose β large enough so that $\beta[tn^2 - (n/16)^2(1+\epsilon_2)] > \beta[tn^2 - (n/16)^2(1+\epsilon_1)] + n^2$. This combined with the result in (Goerdt and Lanka, 2004) completes the proof for deterministic algorithms.

To rule out randomized algorithms running in time $2^{\alpha(n_1+n_2)}$ for some function α of n_1, n_2 for which $\alpha = o(1)$, observe that we can define a new problem which is the same as problem (**P**) except the input description of H is padded with a string of 1's of length $2^{(\alpha/2)(n_1+n_2)}$. This string is irrelevant for solving problem (**P**) but changes the input size to $N = \mathrm{poly}(n_1, n_2) + 2^{(\alpha/2)(n_1+n_2)}$. By the argument in the previous paragraph, any deterministic algorithm still requires $2^{\Omega(n)} = N^{\omega(1)}$ time to solve this problem, which is super-polynomial in the new input size N. However, if a randomized algorithm can solve it in $2^{\alpha(n_1+n_2)}$ time, then it runs in $\mathrm{poly}(N)$ time. This contradicts the assumption that BPP = P. This completes the proof. ■

H. Dual and Bi-Dual Problems

In this section, we will present the dual and bi-dual problems of r^* minimization and nuclear norm minimization, respectively.

H.1. r^* Minimization

We derive the dual and bi-dual problems of non-convex program (**P**). According to (7), the primal problem (**P**) is equivalent to

$$\min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} \frac{1}{2}\| - \mathbf{\Lambda} - \mathbf{AB}\|_F^2 - \frac{1}{2}\|\mathbf{\Lambda}\|_F^2 - H^*(\mathbf{\Lambda}).$$

Therefore, the dual problem is given by

$$\begin{aligned}
&\max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\| - \mathbf{\Lambda} - \mathbf{AB}\|_F^2 - \frac{1}{2}\|\mathbf{\Lambda}\|_F^2 - H^*(\mathbf{\Lambda}) \\
&= \max_{\mathbf{\Lambda}} \frac{1}{2} \sum_{i=r+1}^{n_{(2)}} \sigma_i^2(-\mathbf{\Lambda}) - \frac{1}{2}\|\mathbf{\Lambda}\|_F^2 - H^*(\mathbf{\Lambda}) \\
&= \max_{\mathbf{\Lambda}} -\frac{1}{2}\|\mathbf{\Lambda}\|_r^2 - H^*(\mathbf{\Lambda}), \qquad (\mathbf{D1})
\end{aligned}$$

where $\|\mathbf{\Lambda}\|_r^2 = \sum_{i=1}^r \sigma_i^2(\mathbf{\Lambda})$. The bi-dual problem is derived by

$$\begin{aligned}
&\min_{\mathbf{M}} \max_{\mathbf{\Lambda},\mathbf{\Lambda}'} -\frac{1}{2}\|\mathbf{\Lambda}\|_r^2 - H^*(\mathbf{\Lambda}') + \langle \mathbf{M}, \mathbf{\Lambda}' - \mathbf{\Lambda}\rangle \\
&= \min_{\mathbf{M}} \max_{-\mathbf{\Lambda}} \left[\langle \mathbf{M}, -\mathbf{\Lambda}\rangle - \frac{1}{2}\| - \mathbf{\Lambda}\|_r^2\right] + \max_{\mathbf{\Lambda}'} \left[\langle \mathbf{M}, \mathbf{\Lambda}'\rangle - H^*(\mathbf{\Lambda}')\right] \\
&= \min_{\mathbf{M}} \|\mathbf{M}\|_{r*} + H(\mathbf{M}), \qquad (\mathbf{D2})
\end{aligned}$$

where $\|\mathbf{M}\|_{r*} = \max_{\mathbf{X}} \langle \mathbf{M}, \mathbf{X}\rangle - \frac{1}{2}\|\mathbf{X}\|_r^2$ is a convex function, and the argument $H(\mathbf{M}) = \max_{\mathbf{\Lambda}'} [\langle \mathbf{M}, \mathbf{\Lambda}'\rangle - H^*(\mathbf{\Lambda}')]$ holds by the definition of conjugate function.

H.2. Nuclear Norm Minimization

In this section, we derive the dual and bi-dual problems of non-convex program (34). First note that

$$\begin{aligned}
&\min_{\mathbf{A},\mathbf{B}} \|\mathbf{AB}\|_* + H(\mathbf{AB}) \\
&= \min_{\mathbf{A},\mathbf{B}} \|\mathbf{AB}\|_* + H^{**}(\mathbf{AB}) \\
&= \min_{\mathbf{A},\mathbf{B}} \max_{\mathbf{\Lambda}} \|\mathbf{AB}\|_* + \langle \mathbf{\Lambda}, \mathbf{AB}\rangle - H^*(\mathbf{\Lambda}).
\end{aligned}$$

Therefore, the dual problem is given by

$$\begin{aligned}
&\max_{\mathbf{\Lambda}} \min_{\mathbf{A},\mathbf{B}} \|\mathbf{AB}\|_* + \langle \mathbf{\Lambda}, \mathbf{AB}\rangle - H^*(\mathbf{\Lambda}) \\
&= \max_{\|\mathbf{\Lambda}\|\le 1} -H^*(\mathbf{\Lambda}), \qquad \textbf{(D1'')}
\end{aligned}$$

where $\|\mathbf{\Lambda}\| \le 1$ because otherwise, $\min_{\mathbf{A},\mathbf{B}} \|\mathbf{AB}\|_* + \langle \mathbf{\Lambda}, \mathbf{AB}\rangle - H^*(\mathbf{\Lambda}) = -\infty$ while we would like to maximize it over all $\mathbf{\Lambda}$.

The bi-dual problem is derived by

$$\begin{aligned}
&\min_{\mathbf{M}} \max_{\|\mathbf{\Lambda}\|\le 1, \mathbf{\Lambda}'} -H^*(\mathbf{\Lambda}') + \langle \mathbf{M}, \mathbf{\Lambda}' - \mathbf{\Lambda}\rangle \\
&= \min_{\mathbf{M}} \max_{\|-\mathbf{\Lambda}\|\le 1} \langle \mathbf{M}, -\mathbf{\Lambda}\rangle + \max_{\mathbf{\Lambda}'} \left[\langle \mathbf{M}, \mathbf{\Lambda}'\rangle - H^*(\mathbf{\Lambda}')\right] \\
&= \min_{\mathbf{M}} \|\mathbf{M}\|_* + H(\mathbf{M}), \qquad \textbf{(D2'')}
\end{aligned}$$

where $H(\mathbf{M}) = \max_{\mathbf{\Lambda}'} \left[\langle \mathbf{M}, \mathbf{\Lambda}'\rangle - H^*(\mathbf{\Lambda}')\right]$ holds by the definition of conjugate function, and $\|\mathbf{M}\|_* = \max_{\|-\mathbf{\Lambda}\|\le 1} \langle \mathbf{M}, -\mathbf{\Lambda}\rangle$ because $\|\cdot\|_*$ and $\|\cdot\|$ are a pair of dual norms.

I. Equivalence of Bernoulli and Uniform Models

We begin by arguing that a recovery result under the Bernoulli model with some probability automatically implies a corresponding result for the uniform model with at least the same probability. The argument follows Section 7.1 of Candès et al. (2011). For completeness, we provide the proof here.

Denote by $\Pr_{\mathsf{Unif}(m)}$ and $\Pr_{\mathsf{Ber}(p)}$ probabilities calculated under the uniform and Bernoulli models and let "Success" be the event that the algorithm succeeds. We have

$$\begin{aligned}
\Pr_{\mathsf{Ber}(p)}(\text{Success}) &= \sum_{k=0}^{n_1 n_2} \Pr_{\mathsf{Ber}(p)}(\text{Success} \mid |\Omega| = k) \Pr_{\mathsf{Ber}(p)}(|\Omega| = k) \\
&\le \sum_{k=0}^{m} \Pr_{\mathsf{Unif}(k)}(\text{Success} \mid |\Omega| = k) \Pr_{\mathsf{Ber}(p)}(|\Omega| = k) + \sum_{k=m+1}^{n_1 n_2} \Pr_{\mathsf{Ber}(p)}(|\Omega| = k) \\
&\le \Pr_{\mathsf{Unif}(m)}(\text{Success}) + \Pr_{\mathsf{Ber}(p)}(|\Omega| > m),
\end{aligned}$$

where we have used the fact that for $k \le m$, $\Pr_{\mathsf{Unif}(k)}(\text{Success}) \le \Pr_{\mathsf{Unif}(m)}(\text{Success})$, and that the conditional distribution of $|\Omega|$ is uniform. Thus

$$\Pr_{\mathsf{Unif}(m)}(\text{Success}) \ge \Pr_{\mathsf{Ber}(p)}(\text{Success}) - \Pr_{\mathsf{Ber}(p)}(|\Omega| > m).$$

Take $p = m/(n_1 n_2) - \epsilon$, where $\epsilon > 0$. The conclusion follows from $\Pr_{\mathsf{Ber}(p)}(|\Omega| > m) \le e^{-\frac{\epsilon^2 n_1 n_2}{2p}}$.

References

Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, pages 1171–1197, 2012.

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima for nonconvex optimization in linear time. In *ACM Symposium on the Theory of Computing*, pages 1195–1199, 2017.

Ali Ahmed, Benjamin Recht, and Justin Romberg. Blind deconvolution using convex programming. *IEEE Transactions on Information Theory*, 60(3):1711–1732, 2014.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Christoph Ambühl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM Journal on Computing*, 40(2): 567–596, 2011.

Anima Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. *Annual Conference on Learning Theory*, 2016.

Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1): 2773–2832, 2014.

Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization–provably. In *ACM Symposium on Theory of Computing*, pages 145–162, 2012.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. More algorithms for provable dictionary learning. *arXiv preprint arXiv:1401.0579*, 2014.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Annual Conference on Learning Theory*, pages 152–192, 2016.

Francis Bach, Julien Mairal, and Jean Ponce. Convex sparse matrix factorizations. *arXiv preprint arXiv:0812.1869*, 2008.

Maria-Florina Balcan and Hongyang Zhang. Noise-tolerant life-long matrix completion via adaptive sampling. In *Advances in Neural Information Processing Systems*, pages 2955–2963, 2016.

Maria-Florina Balcan, Yingyu Liang, David P. Woodruff, and Hongyang Zhang. Matrix completion and related problems via strong duality. In *Innovations in Theoretical Computer Science*, volume 94, 2018.

Maria-Florina Balcan, Yi Li, David P Woodruff, and Hongyang Zhang. Testing matrix rank, optimally. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 727–746, 2019.

Amir Beck and Yonina C Eldar. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM Journal on Optimization*, 17(3):844–860, 2006.

Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed analysis of tensor decompositions. In *ACM Symposium on Theory of Computing*, pages 594–603, 2014.

Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. Dropping convexity for faster semi-definite optimization. In *Annual Conference on Learning Theory*, pages 530–582, 2016a.

Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881, 2016b.

Thierry Bouwmans, Sajid Javed, Hongyang Zhang, Zhouchen Lin, and Ricardo Otazo. On the applications of robust pca in image and video processing. *Proceedings of the IEEE*, 106(8): 1427–1457, 2018.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Emmanuel J. Candès and Benjamin Recht. Simple bounds for recovering low-complexity models. *Mathematical Programming*, pages 1–13, 2013.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

Minming Chen, Zhouchen Lin, Yi Ma, and Leqin Wu. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Coordinated Science Laboratory Report no. UILU-ENG-09-2215*, 2009.

Yudong Chen. Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923, 2015.

Yudong Chen and Martin J. Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *International Conference on Artificial Intelligence and Statistics*, 2015.

Uriel Feige. Relations between average case complexity and approximation complexity. In *IEEE Conference on Computational Complexity*, page 5, 2002.

Rina Foygel and Nathan Srebro. Concentration-based guarantees for low-rank matrix reconstruction. In *Annual Conference on Learning Theory*, pages 315–340, 2011.

David Gamarnik, Quan Li, and Hongyi Zhang. Matrix completion from $o(n)$ samples in linear time. In *Annual Conference on Learning Theory*, volume 65, pages 940–947, 2017.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points – online stochastic gradient for tensor decomposition. In *Annual Conference on Learning Theory*, pages 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Rong Ge, Chi Jin, and Zheng Yi. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *ICML*, 2017.

Andreas Goerdt and André Lanka. An approximation hardness result for bipartite clique. In *Electronic Colloquium on Computational Complexity, Report*, volume 48, 2004.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

Christian Grussler, Anders Rantzer, and Pontus Giselsson. Low-rank optimization with convex constraints. *arXiv preprint arXiv:1606.01793*, 2016.

Quanquan Gu, Zhaoran Wang, and Han Liu. Low-rank and sparse structure pursuit via alternating minimization. In *International Conference on Artificial Intelligence and Statistics*, pages 600–609, 2016.

Benjamin Haeffele, Eric Young, and Rene Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning*, pages 2007–2015, 2014.

Benjamin D Haeffele and René Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*, 2015.

Moritz Hardt. Understanding alternating minimization for matrix completion. In *IEEE Symposium on Foundations of Computer Science*, pages 651–660, 2014.

Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. *COLT*, 2013.

Moritz Hardt, Raghu Meka, Prasad Raghavendra, and Benjamin Weitz. Computational limits for matrix completion. In *Annual Conference on Learning Theory*, pages 703–725, 2014.

Bingsheng He and Xiaoming Yuan. On the $O(1/n)$ convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

Russell Impagliazzo and Avi Wigderson. *P = BPP* if *E* requires exponential circuits: Derandomizing the XOR lemma. In *ACM Symposium on the Theory of Computing*, pages 220–229, 1997.

Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *ACM Symposium on Theory of Computing*, pages 665–674, 2013.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. *ICML*, 2017.

Michel Journée, Francis Bach, P-A Absil, and Rodolphe Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

Kenji Kawaguchi. Deep learning without poor local minima. *arXiv preprint arXiv:1605.07110*, 2016.

Raghunandan H Keshavan. *Efficient algorithms for collaborative filtering*. PhD thesis, Stanford University, 2012.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010a.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010b.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Society, 2005.

Yuanzhi Li, Yingyu Liang, and Andrej Risteski. Recovery guarantee of weighted low-rank approximation via alternating minimization. In *International Conference on Machine Learning*, pages 2358–2367, 2016.

Zhouchen Lin and Hongyang Zhang. *Low Rank Models for Visual Analysis: Theories, Algorithms and Applications*. Elsevier, 2017.

Ankur Moitra. An almost optimal algorithm for computing nonnegative rank. *SIAM Journal on Computing*, 45(1):156–173, 2016.

Sahand Negahban and Martin J Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697, 2012.

Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

Michael L Overton and Robert S Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 13(1):41–45, 1992.

Dohyung Park, Anastasios Kyrillidis, Constantine Carmanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. In *International Conference on Artificial Intelligence and Statistics*, volume 54, pages 65–74, 2017.

Ilya Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In *ACM Symposium on Theory of Computing*, pages 250–263, 2016.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

Yuan Shen, Zaiwen Wen, and Yin Zhang. Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods and Software*, 29(2): 239–263, 2014.

Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In *Annual Symposium on the Theory of Computing*, 2017.

Zhao Song, David P Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 2772–2789, 2019.

Nathan Srebro. *Learning with matrix factorizations*. PhD thesis, Massachusetts Institute of Technology, 2004.

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*, pages 545–560, 2005.

Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. In *IEEE International Symposium on Information Theory*, pages 2379–2383, 2016.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere II: Recovery by Riemannian trust-region method. *IEEE Transactions on Information Theory*, 63(2):885–914, 2017a.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, 2017b.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via nonconvex factorization. In *IEEE Symposium on Foundations of Computer Science*, pages 270–289, 2015.

Gongguo Tang, Badri Narayan Bhaskar, Parikshit Shah, and Benjamin Recht. Compressed sensing off the grid. *IEEE Transactions on Information Theory*, 59(11):7465–7490, 2013.

Stephen Tu, Ross Boczar, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank solutions of linear matrix equations via procrustes flow. *ICML*, 2016.

Madeleine Udell, Corinne Horn, Reza Zadeh, Stephen Boyd, et al. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, 2016.

Roman Vershynin. Lectures in geometric functional analysis. pages 1–76, 2009.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint: 1011.3027*, 2010.

Roman Vershynin. Estimation in high dimensions: A geometric perspective. In *Sampling theory, a renaissance*, pages 3–66. Springer, 2015.

Yu-Xiang Wang and Huan Xu. Stability of matrix factorization for collaborative filtering. In *International Conference on Machine Learning*, pages 417–424, 2012.

Zaiwen Wen, Wotao Yin, and Yin Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in neural information processing systems*, pages 4152–4160, 2016.

Hongyang Zhang, Zhouchen Lin, and Chao Zhang. A counterexample for the validity of using nuclear norm as a convex surrogate of rank. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, volume 8189, pages 226–241, 2013.

Hongyang Zhang, Zhouchen Lin, Chao Zhang, and Junbin Gao. Robust latent low rank representation for subspace clustering. *Neurocomputing*, 145:369–373, 2014.

Hongyang Zhang, Zhouchen Lin, Chao Zhang, and Edward Chang. Exact recoverability of robust PCA via outlier pursuit with tight recovery bounds. In *AAAI Conference on Artificial Intelligence*, pages 3143–3149, 2015a.

Hongyang Zhang, Zhouchen Lin, Chao Zhang, and Junbin Gao. Relations among some low rank subspace recovery models. *Neural Computation*, 27:1915–1950, 2015b.

Hongyang Zhang, Zhouchen Lin, and Chao Zhang. Completing low-rank matrices with corrupted samples from few coefficients in general basis. *IEEE Transactions on Information Theory*, 62(8): 4748–4768, 2016.

Hongyang Zhang, Junru Shao, and Ruslan Salakhutdinov. Deep neural networks with multi-branch architectures are intrinsically less non-convex. In *International Conference on Artificial Intelligence and Statistics*, pages 1099–1109, 2019.

Xiao Zhang, Lingxiao Wang, and Quanquan Gu. A nonconvex free lunch for low-rank plus sparse matrix recovery. *arXiv preprint arXiv:1702.06525*, 2017.

Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. In *Advances in Neural Information Processing Systems*, pages 559–567, 2015.

Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems*, pages 109–117, 2015.

Qinqing Zheng and John Lafferty. Convergence analysis for rectangular matrix completion using Burer-Monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.

Journal of Machine Learning Research 20 (2019) 1-68 Submitted 10/17; Revised 5/19; Published 6/19

Regularization via Mass Transportation

Soroosh Shafieezadeh-Abadeh SOROOSH.SHAFIEE@EPFL.CH
Daniel Kuhn DANIEL.KUHN@EPFL.CH
Risk Analytics and Optimization Chair,
EPFL, Switzerland

Peyman Mohajerin Esfahani P.MOHAJERINESFAHANI@TUDELFT.NL
Delft Center for Systems and Control
TU Delft, The Netherlands

Editor: Koji Tsuda

Abstract

The goal of regression and classification methods in supervised learning is to minimize the empirical risk, that is, the expectation of some loss function quantifying the prediction error under the empirical distribution. When facing scarce training data, overfitting is typically mitigated by adding regularization terms to the objective that penalize hypothesis complexity. In this paper we introduce new regularization techniques using ideas from distributionally robust optimization, and we give new probabilistic interpretations to existing techniques. Specifically, we propose to minimize the worst-case expected loss, where the worst case is taken over the ball of all (continuous or discrete) distributions that have a bounded transportation distance from the (discrete) empirical distribution. By choosing the radius of this ball judiciously, we can guarantee that the worst-case expected loss provides an upper confidence bound on the loss on test data, thus offering new generalization bounds. We prove that the resulting regularized learning problems are tractable and can be tractably kernelized for many popular loss functions. The proposed approach to regluarization is also extended to neural networks. We validate our theoretical out-of-sample guarantees through simulated and empirical experiments.

Keywords: Distributionally robust optimization, optimal transport, Wasserstein distance, robust optimization, regularization

1. Introduction

The fields of machine learning and optimization are closely intertwined. On the one hand, optimization algorithms are routinely used for the solution of classical machine learning problems. Conversely, recent advances in optimization under uncertainty have inspired many new machine learning models.

From a conceptual point of view, many statistical learning tasks give naturally rise to stochastic optimization problems. Indeed, they aim to find an estimator from within a prescribed hypothesis space that minimizes the expected value of some loss function. The loss function quantifies the estimator's ability to correctly predict random outputs (*i.e.*, dependent variables or labels) from random inputs (*i.e.*, independent variables or features). Unfortunately, such stochastic optimization problems cannot be solved exactly because the input-output distribution, which is needed to evaluate the expected loss in the objective

©2019 Soroosh Shafieezadeh-Abadeh and Daniel Kuhn and Peyman Mohajerin Esfahani.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-633.html.

function, is not accessible and only indirectly observable through finitely many training samples. Approximating the expected loss with the empirical loss, that is, the average loss across all training samples, yields fragile estimators that are sensitive to perturbations in the data and suffer from overfitting.

Regularization is the standard remedy to combat overfitting. Regularized learning models minimize the sum of the empirical loss and a penalty for hypothesis complexity, which is typically chosen as a norm of the hypothesis. There is ample empirical evidence that regularization reduces a model's generalization error. Statistical learning theory reasons that regularization implicitly restricts the hypothesis space, thereby controlling the gap between the training error and the testing error, see, *e.g.*, Bartlett and Mendelson (2002). However, alternative explanations for the practical success of regularization are possible. In particular, ideas from modern robust optimization (Ben-Tal et al. (2009)) recently led to a fresh perspective on regularization.

Robust regression and classification models seek estimators that are immunized against adversarial perturbations in the training data. They have received considerable attention since the seminal treatise on robust least-squares regression by El Ghaoui and Lebret (1997), who seem to be the first authors to discover an intimate connection between robustification and regularization. Specifically, they show that minimizing the worst-case residual error with respect to all perturbations in a Frobenius norm-uncertainty set is equivalent to a Tikhonov regularization procedure. Xu et al. (2010) disclose a similar equivalence between robust least-squares regression with a feature-wise independent uncertainty set and the celebrated Lasso (least absolute shrinkage and selection operator) algorithm. Leveraging this new robustness interpretation, they extend Lasso to a wider class of regularization schemes tailored to regression problems with disturbances that are coupled across features. In the context of classification, Xu et al. (2009) provide a linkage between robustification over non-box-typed uncertainty sets and the standard regularization scheme of support vector machines. A comprehensive characterization of the conditions under which robustification and regularization are equivalent has recently been compiled by Bertsimas and Copenhaver (2017).

New learning models have also been inspired by recent advances in the emerging field of *distributionally* robust optimization, which bridges the gap between the conservatism of robust optimization and the specificity of stochastic programming. Distributionally robust optimization seeks to minimize a worst-case expected loss, where the worst case is taken with respect to all distributions in an ambiguity set, that is, a family of distributions consistent with the given prior information on the uncertainty, see, *e.g.*, Calafiore and El Ghaoui (2006), Delage and Ye (2010), Goh and Sim (2010), Wiesemann et al. (2014) and the references therein. Ambiguity sets are often characterized through generalized moment conditions. For instance, Lanckriet et al. (2002a) propose a distributionally robust minimax probability machine for binary classification, where both classes are encoded by the first and second moments of their features, and the goal is to find a linear classifier that minimizes the worst-case misclassification error in view of all possible input distributions consistent with the given moment information. By construction, this approach forces the worst-case accuracies of both classes to be equal. Huang et al. (2004) propose a generalization of the minimax probability machine that allows for uneven worst-case classification accuracies. Lanckriet et al. (2002b) extend the minimax probability machine to account for estimation errors in the mean vectors and covariance matrices. Strohmann and Grudic (2003) and Bhattacharyya (2004) develop

minimax probability machines for regression and feature selection, respectively. Shivaswamy et al. (2006) study linear classification problems trained with incomplete and noisy features, where each training sample is modeled by an ambiguous distribution with known first and second-order moments. The authors propose to address such classification problems with a distributionally robust soft margin support vector machine and then prove that it is equivalent to a classical robust support vector machine with a feature-wise uncertainty set. Farnia and Tse (2016) investigate distributionally robust learning models with moment ambiguity sets that restrict the marginal of the features to the empirical marginal. The authors highlight similarities and differences to classical regression models.

Ambiguity sets containing all distributions that share certain low-order moments are computationally attractive but fail to converge to a singleton when the number N of training samples tends to infinity. Thus, they preclude any asymptotic consistency results. A possible remedy is to design spherical ambiguity sets with respect to some probability distance functions and to drive their radii to zero as N grows. Examples include the ϕ-divergence ambiguity sets proposed by Ben-Tal et al. (2013) or the Wasserstein ambiguity sets studied by Mohajerin Esfahani and Kuhn (2018) and Zhao and Guan (2018). Blanchet and Murthy (2019) and Gao and Kleywegt (2016) consider generalized Wasserstein ambiguity sets defined over Polish spaces.

In this paper we investigate distributionally robust learning models with Wasserstein ambiguity sets. The Wasserstein distance between two distributions is defined as the minimum cost of transporting one distribution to the other, where the cost of moving a unit point mass is determined by the ground metric on the space of uncertainty realizations. In computer science the Wasserstein distance is therefore sometimes aptly termed the 'earth mover's distance' (Rubner et al. (2000)). Following Mohajerin Esfahani and Kuhn (2018), we define Wasserstein ambiguity sets as balls with respect to the Wasserstein distance that are centered at the empirical distribution on the training samples. These ambiguity sets contain all (continuous or discrete) distributions that can be converted to the (discrete) empirical distribution at bounded transportation cost.

Wasserstein distances are widely used in machine learning to compare histograms. For example, Rubner et al. (2000) use the Wasserstein distance as a metric for image retrieval with a focus on applications to color and texture. Cuturi (2013) and Benamou et al. (2015) propose fast iterative algorithms to compute a regularized Wasserstein distance between two high-dimensional discrete distributions for image classification tasks. Moreover, Cuturi and Doucet (2014) develop first-order algorithms to compute the Wasserstein barycenter between several empirical probability distributions, which has applications in clustering. Arjovsky et al. (2017) utilize the Wasserstein distance to measure the distance between the data distribution and the model distribution in generative adversarial networks. Furthermore, Frogner et al. (2015) propose a learning algorithm based on the Wasserstein distance to predict multi-label outputs.

Distributionally robust optimization models with Wasserstein ambiguity sets were introduced to the realm of supervised learning by Shafieezadeh-Abadeh et al. (2015), who show that distributionally robust logistic regression problems admit a tractable reformulation and encapsulate the classical as well as the popular regularized logistic regression problems as special cases. When the Wasserstein ball is restricted to distributions on a compact set, the problem becomes intractable but can still be addressed with an efficient decomposition

algorithm due to Luo and Mehrotra (2019). Support vector machine models with distributionally robust chance constraints over Wasserstein ambiguity sets are studied by Lee and Mehrotra (2015). These models are equivalent to hard semi-infinite programs and can be solved approximately with a cutting plane algorithm.

Wasserstein ambiguity sets are popular for their attractive statistical properties. For example, Fournier and Guillin (2015) prove that the empirical distribution on N training samples converges in Wasserstein distance to the true distribution at rate $\mathcal{O}(N^{-1/(n+1)})$, where n denotes the feature dimension. This implies that properly scaled Wasserstein balls constitute natural confidence regions for the data-generating distribution. The worst-case expected prediction loss over all distributions in a Wasserstein ball thus provides an upper confidence bound on the expected loss under the unknown true distribution; see Mohajerin Esfahani and Kuhn (2018). Blanchet et al. (2016) show, however, that radii of the order $\mathcal{O}(N^{-1/2})$ are asymptotically optimal even though the corresponding Wasserstein balls are too small to contain the true distribution with constant confidence. For Wasserstein distances of type two (where the transportation cost equals the squared ground metric) Blanchet et al. (2017) develop a systematic methodology for selecting the ground metric. Generalization bounds for the *worst-case* prediction loss with respect to a Wasserstein ball centered at the *true* distribution are derived by Lee and Raginsky (2018) in order to address emerging challenges in domain adaptation problems, where the distributions of the training and test samples can differ.

This paper extends the results by Shafieezadeh-Abadeh et al. (2015) on distributionally robust logistic regression along several dimensions. Our main contributions can be summarized as follows:

- **Tractability:** We propose data-driven distributionally robust regression and classification models that hedge against all input-output distributions in a Wasserstein ball. We demonstrate that the emerging semi-infinite optimization problems admit equivalent reformulations as tractable convex programs for many commonly used loss functions and for spaces of linear hypotheses. We also show that lifted variants of these new learning models are kernelizable and thus offer an efficient procedure for optimizing over all nonlinear hypotheses in a reproducible kernel Hilbert space. Finally, we study distributionally robust learning models over families of feed-forward neural networks. We show that these models can be approximated by regularized empirical loss minimization problems with a convex regularization term and can be addressed with a stochastic proximal gradient descent algorithm.

- **Probabilistic Interpretation of Existing Regularization Techniques:** We show that the classical regularized learning models emerge as special cases of our framework when the cost of moving probability mass along the output space tends to infinity. In this case, the regularization function and its regularization weight are determined by the transportation cost on the input space and the radius of the Wasserstein ball underlying the distributionally robust optimization model, respectively.

- **Generalization Bounds:** We demonstrate that the proposed distributionally robust learning models enjoy new generalization bounds that can be obtained under minimal assumptions. In particular, they do not rely on any notions of hypothesis complexity

and may therefore even extend to hypothesis spaces with infinite VC-dimensions. A naïve generalization bound is obtained by leveraging modern measure concentration results, which imply that Wasserstein balls constitute confidence sets for the unknown data-generating distribution. Unfortunately, this generalization bound suffers from a curse of dimensionality and converges slowly for high input dimensions. By imposing bounds on the hypothesis space, however, we can derive an improved generalization bound, which essentially follows a dimension-independent square root law reminiscent of the central limit theorem.

- **Relation to Robust Optimization:** In classical robust regression and classification the training samples are viewed as uncertain variables that range over a joint uncertainty set, and the best hypothesis is found by minimizing the worst-case loss over this set. We prove that the classical robust and new distributionally robust learning models are equivalent if the data satisfies a dispersion condition (for regression) or a separability condition (for classification). While there is no efficient algorithm for solving the robust learning models in the absence of this condition, the distributionally robust models are efficiently solvable irrespective of the underlying training datasets.

- **Confidence Intervals for Error and Risk:** Using distributionally robust optimization techniques based on the Wasserstein ball, we develop two tractable linear programs whose optimal values provide a confidence interval for the absolute prediction error of any fixed regressor or the misclassification risk of any fixed classifier.

- **Worst-Case Distributions:** We formulate tractable convex programs that enable us to efficiently compute a worst-case distribution in the Wasserstein ball for any fixed hypothesis. This worst-case distribution can be useful for stress tests or contamination experiments.

The rest of the paper develops as follows. Section 2 introduces our new distributionally robust learning models. Section 3 provides finite convex reformulations for learning problems over linear and nonlinear hypothesis spaces and describes efficient procedures for constructing worst-case distributions. Moreover, it compares the new distributionally robust method against existing robust optimization and regularization approaches. Section 4 develops new generalization bounds, while Section 5 addresses error and risk estimation. Numerical experiments are reported in Section 6. All proofs are relegated to the appendix.

1.1. Notation

Throughout this paper, we adopt the conventions of extended arithmetics, whereby $\infty \cdot 0 = 0 \cdot \infty = 0/0 = 0$ and $\infty - \infty = -\infty + \infty = 1/0 = \infty$. The inner product of two vectors $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ is denoted by $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$, and for any norm $\|\cdot\|$ on $\mathbb{R}^n$, we use $\|\cdot\|_*$ to denote its dual norm defined through $\|\boldsymbol{x}\|_* = \sup\ \{\langle \boldsymbol{x}, \boldsymbol{x}' \rangle : \|\boldsymbol{x}'\| \leq 1\}$. The conjugate of an extended real-valued function $f(\boldsymbol{x})$ on $\mathbb{R}^n$ is defined as $f^*(\boldsymbol{x}) = \sup_{\boldsymbol{x}'} \langle \boldsymbol{x}, \boldsymbol{x}' \rangle - f(\boldsymbol{x}')$. The indicator function of a set $\mathbb{X} \subseteq \mathbb{R}^n$ is defined as $\delta_{\mathbb{X}}(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in \mathbb{X}; = \infty$ otherwise. Its conjugate $S_{\mathbb{X}}(\boldsymbol{x}) = \sup\{\langle \boldsymbol{x}', \boldsymbol{x} \rangle : \boldsymbol{x}' \in \mathbb{X}\}$ is termed the support function of $\mathbb{X}$. The characteristic function of $\mathbb{X}$ is defined through $\mathbb{1}_{\mathbb{X}}(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in \mathbb{X}; = 0$ otherwise. For a proper cone $\mathcal{C} \subseteq \mathbb{R}^n$ the relation $\boldsymbol{x} \succeq_{\mathcal{C}} \boldsymbol{x}'$ indicates that $\boldsymbol{x} - \boldsymbol{x}' \in \mathcal{C}$. The cone dual to $\mathcal{C}$ is defined as

$\mathcal{C}^* = \{\boldsymbol{x}' : \langle \boldsymbol{x}', \boldsymbol{x} \rangle \geq 0\, \forall \boldsymbol{x} \in \mathcal{C}\}$. The Lipschitz modulus of a function $L : \mathbb{X} \to \mathbb{R}$ is denoted by $\mathrm{lip}(L) = \sup_{\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{X}} \{|L(\boldsymbol{x}) - L(\boldsymbol{x}')| / \|\boldsymbol{x} - \boldsymbol{x}'\| : \boldsymbol{x} \neq \boldsymbol{x}'\}$. If $\mathbb{P}$ is a distribution on a set Ξ, then $\mathbb{P}^N$ denotes the N-fold product of $\mathbb{P}$ on the Cartesian product Ξ^N. For $N \in \mathbb{N}$ we define $[N] = \{1, \ldots, N\}$. A list of commonly used notations is provided in the following table.

$\mathbb{X}$	input space	$\mathbb{Y}$	output space
ℓ	loss function	L	univariate loss function
$\mathbb{H}$	hypothesis space	$\boldsymbol{\mathcal{K}}$	kernel matrix
f^*	conjugate of f	$\mathrm{lip}(f)$	Lipschitz modulus of f
$\mathcal{C}^*$	dual cone of $\mathcal{C}$	$\|\cdot\|_*$	dual norm of $\|\cdot\|$
$S_{\mathbb{X}}$	support function of $\mathbb{X}$	$\delta_{\mathbb{X}}$	indicator function of $\mathbb{X}$
$\mathbb{1}_{\mathbb{X}}$	characteristic function of $\mathbb{X}$	$[N]$	$\{1, \ldots, N\}$

2. Problem Statement

We first introduce the basic terminology and then describe our new perspective on regularization.

2.1. Classical Statistical Learning

The goal of supervised learning is to infer an unknown target function $f : \mathbb{X} \to \mathbb{Y}$ from limited data. The target function maps any input $\boldsymbol{x} \in \mathbb{X}$ (*e.g.*, information on the frequency of certain keywords in an email) to some output $y \in \mathbb{Y}$ (*e.g.*, a label $+1$ (-1) if the email is likely (unlikely) to be a spam message). If the true target function was accessible, it could be used as a means to reliably predict outputs from inputs (*e.g.*, it could be used to recognize spam messages in an automated fashion). In a supervised learning framework, however, one has only access to finitely many input-output examples $(\widehat{\boldsymbol{x}}_i, \widehat{y}_i)$ for $i = 1, \ldots, N$ (*e.g.*, a database of emails which have been classified by a human as legitimate or as spam messages). We will henceforth refer to these examples as the *training data* or the *in-sample data*. It is assumed that the training samples are mutually independent and follow an unknown distribution $\mathbb{P}$ on $\mathbb{X} \times \mathbb{Y}$.

The supervised learning problems are commonly subdivided into *regression problems*, where the output y is continuous and $\mathbb{Y} = \mathbb{R}$, and *classification problems*, where y is categorical and $\mathbb{Y} = \{+1, -1\}$. As the space of all functions from $\mathbb{X}$ to $\mathbb{Y}$ is typically vast, it may be very difficult to learn the target function from finitely many training samples. Thus, it is convenient to restrict the search space to a structured family of candidate functions $\mathbb{H} \subseteq \mathbb{R}^{\mathbb{X}}$ such as the space of all linear functions, some reproducible kernel Hilbert space or the family of all feed-forward neural networks with a fixed number of layers. We henceforth refer to each candidate function $h \in \mathbb{H}$ as a *hypothesis* and to $\mathbb{H}$ as the *hypothesis space*.

A *learning algorithm* is a method for finding a hypothesis $h \in \mathbb{H}$ that faithfully replicates the unknown target function f. Specifically, in regression we seek to approximate f with a hypothesis h, and in classification we seek to approximate f with a *thresholded* hypothesis $\mathrm{sign}(h)$. Many learning algorithms achieve this goal by minimizing the in-sample error, that is, the empirical average of a loss function $\ell : \mathbb{R} \times \mathbb{Y} \to \mathbb{R}_+$ that estimates the mismatch

between the output predicted by $h(\boldsymbol{x})$ and the actual output y for a particular input-output pair $(\boldsymbol{x}, y)$. Any such algorithm solves a minimization problem of the form

$$\inf_{h\in\mathbb{H}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \ell(h(\widehat{\boldsymbol{x}}_i), \widehat{y}_i) = \mathbb{E}^{\widehat{\mathbb{P}}_N} \left[\ell(h(\boldsymbol{x}), y) \right] \right\}, \tag{1}$$

where $\widehat{\mathbb{P}}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{(\widehat{\boldsymbol{x}}_i, \widehat{y}_i)}$ denotes the *empirical distribution*, that is, the uniform distribution on the training data. For different choices of the the loss function ℓ, the generic supervised learning problem (1) reduces to different popular regression and classification problems from the literature.

Examples of Regression Models

For ease of exposition, we focus here on learning models with $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{Y} \subseteq \mathbb{R}$, where $\mathbb{H}$ is set to the space of all linear hypotheses $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ with $\boldsymbol{w} \in \mathbb{R}^n$. Thus, there is a one-to-one correspondence between hypotheses and weight vectors $\boldsymbol{w}$. Moreover, we focus on loss functions of the form $\ell(h(\boldsymbol{x}), y) = L(h(\boldsymbol{x}) - y) = L(\langle \boldsymbol{w}, \boldsymbol{x} \rangle) - y)$ that are generated by a univariate loss function L.

1. A rich class of *robust regression* problems is obtained from (1) if ℓ is generated by the *Huber loss function* with robustness parameter $\delta > 0$, which is defined as $L(z) = \frac{1}{2}z^2$ if $|z| \leq \delta$; $= \delta(|z| - \frac{1}{2}\delta)$ otherwise. Note that the Huber loss function is both convex and smooth and reduces to the squared loss $L(z) = \frac{1}{2}z^2$ for $\delta \uparrow \infty$, which is routinely used in ordinary least squares regression. Problem (1) with squared loss seeks a hypothesis $\boldsymbol{w}$ under which $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ approximates the mean of y conditional on $\boldsymbol{x}$. The Huber loss function for finite δ favors similar hypotheses but is less sensitive to outliers.

2. The *support vector regression* problem (Smola and Schölkopf, 2004) emerges as a special case of (1) if ℓ is generated by the *ϵ-insensitive loss function* $L(z) = \max\{0, |z| - \epsilon\}$ with $\epsilon \geq 0$. In this setting, a training sample $(\widehat{\boldsymbol{x}}_i, \widehat{y}_i)$ is penalized in (1) only if the output $\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle$ predicted by hypothesis $\boldsymbol{w}$ differs from the true output $\widehat{y}_i$ by more than ϵ. Support vector regression thus seeks hypotheses $\boldsymbol{w}$ under which all training samples reside within a slab of width 2ϵ centered around the hyperplane $\{(\boldsymbol{x}, y) : \langle \boldsymbol{w}, \boldsymbol{x} \rangle = y\}$.

3. The *quantile regression* problem (Koenker, 2005) is obtained from (1) if ℓ is generated by the *pinball loss function* $L(z) = \max\{-\tau z, (1-\tau)z\}$ defined for $\tau \in [0, 1]$. Quantile regression seeks hypotheses that approximate the $\tau \times 100\%$-quantile of the output conditional on the input. More precisely, it seeks hypotheses $\boldsymbol{w}$ for which $\tau \times 100\%$ of all training samples lie in the halfspace $\{(\boldsymbol{x}, y) : \langle \boldsymbol{w}, \boldsymbol{x} \rangle \geq y\}$.

Examples of Classification Models

We focus here on linear learning models with $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{Y} = \{+1, -1\}$, where $\mathbb{H}$ is again identified with the space of all linear hypotheses $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ with $\boldsymbol{w} \in \mathbb{R}^n$. Moreover, we focus on loss functions of the form $\ell(\boldsymbol{x}, y) = L(yh(\boldsymbol{x})) = L(y\langle \boldsymbol{w}, \boldsymbol{x} \rangle)$ generated by a univariate loss function L.

1. The *support vector machine* problem (Cortes and Vapnik, 1995) is obtained from (1) if ℓ is generated by the *hinge loss function* $L(z) = \max\{0, 1 - z\}$, which is large if z is small. Thus, a training sample $(\widehat{\boldsymbol{x}}_i, \widehat{y}_i)$ is penalized in (1) if the output $\text{sign}(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle)$ predicted by hypothesis $\boldsymbol{w}$ and the true output $\widehat{y}_i$ have opposite signs. More precisely, support vector machines seek hypotheses $\boldsymbol{w}$ under which the inputs of all training samples with output $+1$ reside in the halfspace $\{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle \geq 1\}$, while the inputs of training samples with output -1 are confined to $\{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle \leq -1\}$.

2. An alternative support vector machine problem is obtained from (1) if ℓ is generated by the *smooth hinge loss function*, which is defined as $L(z) = \frac{1}{2} - z$ if $z \leq 0$; $= \frac{1}{2}(1-z)^2$ if $0 < z < 1$; $= 0$ otherwise. The smooth hinge loss inherits many properties of the ordinary hinge loss but has a continuous derivative. Thus, it may be amenable to faster optimization algorithms.

3. The *logistic regression* problem (Hosmer et al., 2013) emerges as a special case of (1) if ℓ is generated by the *logloss function* $L(z) = \log(1 + e^{-z})$, which is large if z is small—similar to the hinge loss function. In this case the objective function of (1) can be viewed as the log-likelihood function corresponding to the logistic model $\mathbb{P}(y = 1|\boldsymbol{x}) = [1 + \exp(-\langle \boldsymbol{w}, \boldsymbol{x} \rangle)]^{-1}$ for the conditional probability of $y = 1$ given $\boldsymbol{x}$. Thus, logistic regression allows us to learn the conditional distribution of y given $\boldsymbol{x}$.

Remark 1 (Convex approximation) *Note that the hinge loss and the logloss functions represent convex approximations for the (discontinuous)* one-zero loss *defined through* $L(z) = 1$ *if* $z \leq 0$*;* $= 0$ *otherwise.*

In practice there may be many hypotheses that are compatible with the given training data and thus achieve a small empirical loss in (1). Any such hypothesis would accurately predict outputs from inputs *within the training dataset* (Defourny, 2010). However, due to overfitting, these hypotheses might constitute poor predictors *beyond the training dataset*, that is, on inputs that have not yet been recorded in the database. Mathematically, even if the in-sample error $\mathbb{E}^{\widehat{\mathbb{P}}_N}[\ell(\langle \boldsymbol{w}, \boldsymbol{x} \rangle, y)]$ of a given hypothesis $\boldsymbol{w}$ is small, the out-of-sample error $\mathbb{E}^{\mathbb{P}}[\ell(\langle \boldsymbol{w}, \boldsymbol{x} \rangle, y)]$ with respect to the unknown true input-output distribution $\mathbb{P}$ may be large.

Regularization is the standard remedy to combat overfitting. Instead of naïvely minimizing the in-sample error as is done in (1), it may thus be advisable to solve the regularized learning problem

$$\inf_{\boldsymbol{w}} \; \mathbb{E}^{\widehat{\mathbb{P}}_N} \left[\ell(\langle \boldsymbol{w}, \boldsymbol{x} \rangle, y) \right] + c\, \Omega(\boldsymbol{w}), \tag{2}$$

which minimizes the sum of the emiprial average loss and a penalty for hpothesis complexity, which consists of a regularization function $\Omega(\boldsymbol{w})$ and its associated regularization weight c. Tikhonov regularization (Tikhonov et al., 1977), for example, corresponds to the choice $\Omega(\boldsymbol{w}) = \|\boldsymbol{\Gamma}\boldsymbol{w}\|_2^2$ for some Tikhonov matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$. Setting $\boldsymbol{\Gamma}$ to the identity matrix gives rise to standard L_2-regularization. Similarly, Lasso (least absolute shrinkage and selection operator) regularization or L_1-regularization (Tibshirani, 1996) is obtained by setting $\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$. Lasso regularization has gained popularity because it favors parsimonious interpretable hypotheses.

Most popular regualization methods admit probabilistic interpretations. However, these interpretations typically rely on prior distributional assumptions that remain to some extent arbitrary (*e.g.*, L_2- and L_1-regularization can be justified if $\boldsymbol{w}$ is governed by a Gaussian or Laplacian prior distribution, respectively (Tibshirani, 1996)). Thus, in spite of their many desirable theoretical properties, there is a consensus that "*most of the (regularization) methods used successfully in practice are heuristic methods*" (Abu-Mostafa et al., 2012).

2.2. A New Perspective on Regularization

When *linear* hypotheses are used, problem (1) minimizes the in-sample error $\mathbb{E}^{\widehat{\mathbb{P}}_N}[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)]$. However, a hypothesis $\boldsymbol{w}$ enjoying a low in-sample error may still suffer from a high out-of-sample error $\mathbb{E}^{\mathbb{P}}[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)]$ due to overfitting. This is unfortunate as we seek hypotheses that offer high prediction accuracy on *future* data, meaning that the out-of-sample error is the actual quantity of interest. An ideal learning model would therefore minimize the out-of-sample error. This is impossible, however, for the following reasons:

- The true input-output distribution $\mathbb{P}$ is unknown and only indirectly observable through the N training samples. Thus, we lack essential information to compute the out-of-sample error.
- Even if the distribution $\mathbb{P}$ was known, computing the out-of-sample error would typically be hard due to the intractability of high-dimensional integration; see, *e.g.*, (Hanasusanto et al., 2016, Corollary 1).

The regularized loss $\mathbb{E}^{\widehat{\mathbb{P}}_N}[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)] + c\,\Omega(\boldsymbol{w})$ used in (2), which consists of the in-sample error and an overfitting penalty, can be viewed as an in-sample estimate of the out-of-sample error. Yet, problem (2) remains difficult to justify rigorously. Therefore, we advocate here a more principled approach to regularization. Specifically, we propose to take into account the expected loss of hypothesis $\boldsymbol{w}$ under *every* distribution $\mathbb{Q}$ that is close to the empirical distribution $\widehat{\mathbb{P}}_N$, that is, every $\mathbb{Q}$ that could have generated the training data with high confidence. To this end, we first introduce a distance measure for distributions. For ease of notation, we henceforth denote the input-output pair $(\boldsymbol{x}, y)$ by $\boldsymbol{\xi}$, and we set $\Xi = \mathbb{X} \times \mathbb{Y}$.

Definition 2 (Wasserstein metric) *The Wasserstein distance between two distributions $\mathbb{Q}$ and $\mathbb{Q}'$ supported on Ξ is defined as*

$$W(\mathbb{Q}, \mathbb{Q}') := \inf_{\Pi} \left\{ \int_{\Xi^2} d(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \Pi(\mathrm{d}\boldsymbol{\xi}, \mathrm{d}\boldsymbol{\xi}') \;:\; \begin{array}{l} \Pi \text{ is a joint distribution of } \boldsymbol{\xi} \text{ and } \boldsymbol{\xi}' \\ \text{with marginals } \mathbb{Q} \text{ and } \mathbb{Q}', \text{ respectively} \end{array} \right\},$$

where d is a metric on Ξ.

By definition, $W(\mathbb{Q}, \mathbb{Q}')$ represents the solution of an infinite-dimensional transportation problem, that is, it corresponds to the minimal cost for moving the distribution $\mathbb{Q}$ to $\mathbb{Q}'$, where the cost for moving a unit probability mass from $\boldsymbol{\xi}$ to $\boldsymbol{\xi}'$ is given by the transportation distance $d(\boldsymbol{\xi}, \boldsymbol{\xi}')$. Due to this interpretation, the metric d is often referred to as the transportation cost (Villani, 2008) or ground metric (Cuturi and Avis, 2014), while the Wasserstein metric is sometimes termed the mass transportation distance or earth mover's distance (Rubner et al., 2000).

Consider now the Wasserstein ball of radius $\rho \geq 0$ around the empirical distribution $\widehat{\mathbb{P}}_N$,

$$\mathbb{B}_\rho(\widehat{\mathbb{P}}_N) = \big\{ \mathbb{Q} \,:\, \mathbb{Q}(\Xi) = 1,\; W(\mathbb{Q}, \widehat{\mathbb{P}}_N) \leq \rho \big\}, \tag{3}$$

which contains all input-output distributions $\mathbb{Q}$ supported on Ξ whose Wasserstein distance from $\widehat{\mathbb{P}}_N$ does not exceed ρ. This means that $\mathbb{Q}$ can be transported to $\widehat{\mathbb{P}}_N$ (or vice versa) at a cost of at most ρ. The hope is that a large enough Wasserstein ball will contain distributions that are representative of the unknown true input-output distribution $\mathbb{P}$, such that the worst-case expectation $\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)]$ can serve as an upper confidence bound on the out-of-sample error $\mathbb{E}^{\mathbb{P}}[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)]$. This motivates us to introduce a new regularized learning model, which minimizes precisely this worst-case expectation.

$$\inf_{\boldsymbol{w}} \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] \tag{4}$$

Problem (4) represents a distributionally robust convex program of the type considered in (Mohajerin Esfahani and Kuhn, 2018). Note that if $\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)$ is convex in $\boldsymbol{w}$ for every fixed $(\boldsymbol{x}, y)$, *i.e.*, if ℓ is convex in its first argument, then the objective function of (4) is convex because convexity is preserved under integration and maximization. Note also that if ρ is set to zero, then (4) collapses to the unregularized in-sample error minimization problem (1).

Remark 3 (Support information) *The uncertainty set Ξ captures prior information on the range of the inputs and outputs. In image processing, for example, pixel intensities range over a known interval. Similarly, in diagnostic medicine, physiological parameters such as blood glucose or cholesterol concentrations are restricted to be non-negative. Sometimes it is also useful to construct Ξ as a confidence set that covers the support of $\mathbb{P}$ with a prescribed probability. Such confidence sets are often constructed as ellipsoids, as intersections of different norm balls (Ben-Tal et al., 2009; Delage and Ye, 2010) or as sublevel sets of kernel expansions (Schölkopf et al., 2001).*

In the remainder we establish that the distributionally robust learning problem (4) has several desirable properties. (i) Problem (4) is computationally tractable under standard assumptions about the loss function ℓ, the input-output space Ξ and the transportation metric d. For specific choices of d it even reduces to a regularized learning problem of the form (2). (ii) For all univariate loss functions reviewed in Section 2.1, a tight conservative approximation of (4) is kernelizable, that is, it can be solved implicitly over high-dimensional spaces of nonlinear hypotheses at the same computational cost required for linear hypothesis spaces. (iii) Leveraging modern measure concentration results, the optimal value of (4) can be shown to provide an upper confidence bound on the out-of-sample error. This obviates the need to mobilize the full machinery of VC theory and, in particular, to estimate the VC dimension of the hypothesis space in order to establish generalization bounds. (iv) If the number of training samples tends to infinity while the Wasserstein ball shrinks at an appropriate rate, then problem (4) asymptotically recovers the *ex post* optimal hypothesis that attains the minimal out-of-sample error.

3. Tractable Reformulations

In this section we demonstrate that the distributionally robust learning problem (4) over linear hypotheses is amenable to efficient computational solution procedures. We also discuss generalizations to nonlinear hypothesis classes such as reproducing kernel Hilbert spaces and families of feed-forward neural networks.

3.1. Distributionally Robust Linear Regression

Throughout this section we focus on linear regression problems, where $\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y) = L(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y)$ for some convex univariate loss function L. We also assume that $\mathbb{X}$ and $\mathbb{Y}$ are both convex and closed and that the transportation metric d is induced by a norm $\|\cdot\|$ on the input-output space $\mathbb{R}^{n+1}$. In this setting, the distributionally robust regression problem (4) admits an equivalent reformulation as a finite convex optimization problem if either (i) the univariate loss function L is piecewise affine or (ii) $\Xi = \mathbb{R}^{n+1}$ and L is Lipschitz continuous (but not necessarily piecewise affine).

Theorem 4 (Distributionally robust linear regression) *The following statements hold.*

(i) If $L(z) = \max_{j\leq J}\{a_j z + b_j\}$, then (4) is equivalent to

$$\left\{\begin{array}{lll} \inf\limits_{\substack{\boldsymbol{w},\lambda,s_i\\ \boldsymbol{p}_{ij},u_{ij}}} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & S_\Xi(a_j\boldsymbol{w} - \boldsymbol{p}_{ij}, -a_j - u_{ij}) + b_j + \langle \boldsymbol{p}_{ij}, \widehat{\boldsymbol{x}}_i\rangle + u_{ij}\widehat{y}_i \leq s_i & i\in[N], j\in[J] \\ & \|(\boldsymbol{p}_{ij}, u_{ij})\|_* \leq \lambda & i\in[N], j\in[J], \end{array}\right. \tag{5}$$

where S_Ξ denotes the support function of Ξ.

(ii) If $\Xi = \mathbb{R}^{n+1}$ and $L(z)$ is Lipschitz continuous, then (4) is equivalent to

$$\inf_{\boldsymbol{w}} \ \frac{1}{N}\sum_{i=1}^N L(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i) + \rho\,\mathrm{lip}(L)\,\|(\boldsymbol{w}, -1)\|_*. \tag{6}$$

In the following, we exemplify Theorem 4 for the Huber, ϵ-insensitive and pinball loss functions under the assumption that the uncertainty set Ξ admits the conic representation

$$\Xi = \{(\boldsymbol{x}, y) \in \mathbb{R}^{n+1} : \boldsymbol{C}_1\boldsymbol{x} + \boldsymbol{c}_2 y \preceq_{\mathcal{C}} \boldsymbol{d}\} \tag{7}$$

for some matrix $\boldsymbol{C}_1$, vectors $\boldsymbol{c}_2$ and $\boldsymbol{d}$ and proper convex cone $\mathcal{C}$ of appropriate dimensions. We also assume that Ξ admits a Slater point $(\boldsymbol{x}_{\mathrm{S}}, y_{\mathrm{S}}) \in \mathbb{R}^{n+1}$ with $\boldsymbol{C}_1\boldsymbol{x}_{\mathrm{S}} + \boldsymbol{c}_2 y_{\mathrm{S}} \prec_{\mathcal{C}} \boldsymbol{d}$.

Corollary 5 (Robust regression) *If L represents the Huber loss function with threshold $\delta \geq 0$ and $\Xi = \mathbb{R}^{n+1}$, then (4) is equivalent to*

$$\inf_{\boldsymbol{w}, z_i} \ \frac{1}{N}\sum_{i=1}^N \frac{1}{2}z_i^2 + \delta|\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i - z_i| + \rho\,\delta\,\|(\boldsymbol{w}, -1)\|_*. \tag{8}$$

Corollary 6 (Support vector regression) *If L represents the ϵ-insensitive loss function for some $\epsilon \geq 0$ and Ξ is of the form (7), then (4) is equivalent to*

$$\left\{\begin{array}{lll} \inf\limits_{\substack{\boldsymbol{w},\lambda,s_i\\ \boldsymbol{p}_i^+,\boldsymbol{p}_i^-}} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & \widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \epsilon + \langle \boldsymbol{p}_i^+, \boldsymbol{d} - \boldsymbol{C}_1\widehat{\boldsymbol{x}}_i - \boldsymbol{c}_2\widehat{y}_i\rangle \leq s_i & i\in[N] \\ & \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i - \epsilon + \langle \boldsymbol{p}_i^-, \boldsymbol{d} - \boldsymbol{C}_1\widehat{\boldsymbol{x}}_i - \boldsymbol{c}_2\widehat{y}_i\rangle \leq s_i & i\in[N] \\ & \left\|\left(\boldsymbol{C}_1^\top\boldsymbol{p}_i^+ + \boldsymbol{w}, \boldsymbol{c}_2^\top\boldsymbol{p}_i^+ - 1\right)\right\|_* \leq \lambda & i\in[N] \\ & \left\|\left(\boldsymbol{C}_1^\top\boldsymbol{p}_i^- - \boldsymbol{w}, \boldsymbol{c}_2^\top\boldsymbol{p}_i^- + 1\right)\right\|_* \leq \lambda & i\in[N] \\ & \boldsymbol{p}_i^+, \boldsymbol{p}_i^- \in \mathcal{C}^* & i\in[N] \\ & s_i \geq 0 & i\in[N]. \end{array}\right. \tag{9}$$

Corollary 7 (Quantile regression) *If L represents the pinball loss function for some $\tau \in [0,1]$ and Ξ is of the form (7), then (4) is equivalent to*

$$\left\{\begin{array}{lll} \inf\limits_{\substack{\boldsymbol{w},\lambda,s_i\\ \boldsymbol{p}_i^+,\boldsymbol{p}_i^-}} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & \tau\left(\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle\right) + \langle \boldsymbol{p}_i^+, \boldsymbol{d} - \boldsymbol{C}_1\widehat{\boldsymbol{x}}_i - \boldsymbol{c}_2\widehat{y}_i\rangle \leq s_i & i\in[N] \\ & (1-\tau)\left(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i\right) + \langle \boldsymbol{p}_i^-, \boldsymbol{d} - \boldsymbol{C}_1\widehat{\boldsymbol{x}}_i - \boldsymbol{c}_2\widehat{y}_i\rangle \leq s_i & i\in[N] \\ & \left\|\left(\boldsymbol{C}_1^\top\boldsymbol{p}_i^+ + \tau\boldsymbol{w}, \boldsymbol{c}_2^\top\boldsymbol{p}_i^+ - \tau\right)\right\|_* \leq \lambda & i\in[N] \\ & \left\|\left(\boldsymbol{C}_1^\top\boldsymbol{p}_i^- - (1-\tau)\boldsymbol{w}, \boldsymbol{c}_2^\top\boldsymbol{p}_i^- + 1 - \tau\right)\right\|_* \leq \lambda & i\in[N] \\ & \boldsymbol{p}_i^+, \boldsymbol{p}_i^- \in \mathcal{C}^* & i\in[N] \\ & s_i \geq 0 & i\in[N]. \end{array}\right. \tag{10}$$

Remark 8 (Relation to classical regularization) *Assume now that the mass transportation costs are additively separable with respect to inputs and outputs, that is,*

$$d((\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2)) = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| + \kappa|y_1 - y_2| \tag{11}$$

for some $\kappa > 0$.[1] *Note that κ captures the costs of moving probability mass along the output space. For $\kappa = \infty$ all distributions in the Wasserstein ball $\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ are thus obtained by reshaping $\widehat{\mathbb{P}}_N$ only along the input space. It is easy to verify that for $\kappa = \infty$ and $\Xi = \mathbb{R}^{n+1}$ the learning models portrayed in Corollaries 5-7 all simplify to*

$$\inf_{\boldsymbol{w}} \ \frac{1}{N}\sum_{i=1}^N L(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i) + c\|\boldsymbol{w}\|_*, \tag{12}$$

where $c = \rho\delta$ for robust regression with Huber loss, $c = \rho$ for support vector regression with ϵ-insensitive loss and $c = \max\{\tau, 1-\tau\}\,\rho$ for quantile regression with pinball loss. Thus, (12) is easily identified as an instance of the classical regularized learning problem (2), where the dual norm term $\|\boldsymbol{w}\|_*$ plays the role of the regularization function, while c represents

1. By slight abuse of notation, the symbol $\|\cdot\|$ now denotes a norm on $\mathbb{R}^n$.

the usual regularization weight. By definition of the dual norm, the penalty $\|\boldsymbol{w}\|_*$ assigned to a hypothesis $\boldsymbol{w}$ is maximal (minimal) if the cost of moving probability mass along $\boldsymbol{w}$ is minimal (maximal). We emphasize that if $\kappa = \infty$, then the marginal distribution of y corresponding to every $\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ *coincides with the empirical distribution $\frac{1}{N}\sum_{i=1}^N \delta_{\widehat{y}_i}$. Thus, classical regularization methods, which correspond to $\kappa = \infty$, are explained by a counterintuitive probabilistic model, which pretends that any training sample must have an output that has already been recordeded in the training dataset. In other words, classical regularization implicitly assumes that there is no uncertainty in the outputs. More intuitively appealing regularization schemes are obtained for finite values of κ.*

To establish a connection between distributionally robust and classical robust regression as discussed in (El Ghaoui and Lebret, 1997; Xu et al., 2010), we further investigate the worst-case expected loss of a fixed linear hypothesis $\boldsymbol{w}$.

$$\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}[L(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y)] \tag{13}$$

Theorem 9 (Extremal distributions in linear regression) *The following statements hold.*

(i) If $L(z) = \max_{j\leq J}\{a_j z + b_j\}$, then the worst-case expectation (13) *coincides with*

$$\left\{\begin{array}{lll} \sup\limits_{\alpha_{ij},\boldsymbol{q}_{ij},v_{ij}} & \frac{1}{N}\sum\limits_{i=1}^N\sum\limits_{j=1}^J \alpha_{ij}\big(a_j(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i) + b_j\big) + a_j(\langle \boldsymbol{w}, \boldsymbol{q}_{ij}\rangle - v_{ij}) & \\ \text{s.t.} & \frac{1}{N}\sum\limits_{i=1}^N\sum\limits_{j=1}^J \|(\boldsymbol{q}_{ij}, v_{ij})\| \leq \rho & \\ & \sum\limits_{j=1}^J \alpha_{ij} = 1 & i\in[N] \\ & (\widehat{\boldsymbol{x}}_i + \boldsymbol{q}_{ij}/\alpha_{ij}, \widehat{y}_i + v_{ij}/\alpha_{ij}) \in \Xi & i\in[N], j\in[J] \\ & \alpha_{ij} \geq 0 & i\in[N], j\in[J] \end{array}\right. \tag{14}$$

for any fixed hypothesis $\boldsymbol{w}$. Moreover, if $(\alpha^\star_{ij}, \boldsymbol{q}^\star_{ij}, v^\star_{ij})$ maximizes (14)*, then the discrete distribution*

$$\mathbb{Q}^* = \frac{1}{N}\sum_{i=1}^N\sum_{j=1}^J \alpha^\star_{ij}\,\delta_{(\widehat{\boldsymbol{x}}_i + \boldsymbol{q}^\star_{ij}/\alpha^\star_{ij},\, \widehat{y}_i + v^\star_{ij}/\alpha^\star_{ij})},$$

represents a maximizer for (13)*.*

(ii) If $\Xi = \mathbb{R}^{n+1}$ and $L(z)$ is Lipschitz continuous, then the discrete distributions

$$\mathbb{Q}_\gamma = \frac{1}{N}\sum_{i=2}^N \delta_{(\widehat{\boldsymbol{x}}_i, \widehat{y}_i)} + \frac{1-\gamma}{N}\delta_{(\widehat{\boldsymbol{x}}_1, \widehat{y}_1)} + \frac{\gamma}{N}\delta_{(\widehat{\boldsymbol{x}}_1 + \frac{\rho N}{\gamma}\boldsymbol{x}^\star, \widehat{y}_1 + \frac{\rho N}{\gamma}y^\star)} \quad \text{for} \quad \gamma\in(0,1],$$

where $(\boldsymbol{x}^\star, y^\star)$ solves $\max_{\|(\boldsymbol{x},y)\|\leq 1}\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y$, are feasible and asymptotically optimal in (13) *for $\gamma \downarrow 0$.*

Recall that $0/0 = 0$ and $1/0 = \infty$ by our conventions of extended arithmetic. Thus, any solution feasible in (14) with $\alpha_{ij} = 0$ must satisfy $\boldsymbol{q}_{ij} = \boldsymbol{0}$ and $v_{ij} = 0$ because otherwise $(\widehat{\boldsymbol{x}}_i + \boldsymbol{q}_{ij}/\alpha_{ij}, \widehat{y}_i + v_{ij}/\alpha_{ij}) \notin \Xi$.

Theorem 9 shows how one can use convex optimization to construct a sequence of distributions that are asymptotically optimal in (13). Next, we argue that the worst-case expected cost (13) is equivalent to a (robust) worst-case cost over a suitably defined uncertainty set if the following assumption holds.

Assumption 10 (Minimal dispersion) *For every $\boldsymbol{w} \in \mathbb{R}^n$ there is a training sample $(\widehat{\boldsymbol{x}}_k, \widehat{y}_k)$ for some $k \leq N$ such that the derivative L' exists at $\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k\rangle - \widehat{y}_k$ and satisfies $|L'(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k\rangle - \widehat{y}_k)| = \mathrm{lip}(L)$.*

Remark 11 (Minimal dispersion) *Assumption 10 is reminiscent of the non-separability condition in (Xu et al., 2009, Theorem 3), which is necessary to prove the equivalence of robust and regularized support vector machines. In the regression context studied here, Assumption 10 ensures that, for every $\boldsymbol{w}$, there exists a training sample that activates the largest absolute slope of L.*

For instance, in support vector regression, it means that for every $\boldsymbol{w}$ there exists a data point outside of the slab of width $2\epsilon/\|(\boldsymbol{w}, -1)\|_2$ centered around the hyperplane $H_{\boldsymbol{w}} = \{(\boldsymbol{x}, y) \in \mathbb{R}^n\times\mathbb{R} : \langle \boldsymbol{w}, \boldsymbol{x}\rangle - y = 0\}$ (i.e., the empirical ϵ-insensitive loss is not zero). Similarly, in robust regression with the Huber loss function, Assumption 10 stipulates that for every $\boldsymbol{w}$ there exists a data point outside of the slab of width $2\delta/\|(\boldsymbol{w}, -1)\|_2$ centered around $H_{\boldsymbol{w}}$. However, quantile regression with $\tau \neq 0.5$ fails to satisfy Assumption 10. Indeed, for any training dataset there always exists some $\boldsymbol{w}$ such that all data points reside on the side of $H_{\boldsymbol{w}}$ where the pinball loss function is less steep.

Theorem 12 (Robust regression) *If $\Xi = \mathbb{R}^{n+1}$ and the loss function $L(z)$ is Lipschitz continuous, then the worst-case expected loss* (13) *provides an upper bound on the (robust) worst-case loss*

$$\left\{\begin{array}{lll} \sup\limits_{\Delta\boldsymbol{x}_i, \Delta y_i} & \frac{1}{N}\sum\limits_{i=1}^N [L(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i + \Delta\boldsymbol{x}_i\rangle - \widehat{y}_i - \Delta y_i)] & \\ \text{s.t.} & \frac{1}{N}\sum\limits_{i=1}^N \|(\Delta\boldsymbol{x}_i, \Delta y_i)\| \leq \rho. & \end{array}\right. \tag{15}$$

Moreover, if Assumption 10 holds, then (13) *and* (15) *are equal.*

Remark 13 (Tractability of robust regression) *Assume that $\Xi = \mathbb{R}^{n+1}$, while L and $\|\cdot\|$ both admit a tractable conic representation. By Theorem 4, the worst-case* expected *loss* (13) *can then be computed in polynomial time by solving a tractable convex program. Theorem 12 thus implies that the worst-case loss* (28) *can also be computed in polynomial time if Assumption 10 holds. To our best knowledge, there exists no generic efficient method for computing* (28) *if Assumption 10 fails to hold and L is* not *piecewise affine. This reinforces our belief that a distributionally robust approach to regression is more natural.*

3.2. Distributionally Robust Linear Classification

Throughout this section we focus on linear classification problems, where $\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y) = L(y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)$ for some convex univariate loss function L. We also assume that $\mathbb{X}$ is both convex and closed and that $\mathbb{Y} = \{+1, -1\}$. Moreover, we assume that the transportation metric d is defined via

$$d((\boldsymbol{x}, y), (\boldsymbol{x}', y')) = \|\boldsymbol{x} - \boldsymbol{x}'\| + \kappa \mathbb{1}_{\{y \neq y'\}}, \tag{16}$$

where $\|\cdot\|$ represents a norm on the input space $\mathbb{R}^n$, and $\kappa > 0$ quantifies the cost of switching a label. In this setting, the distributionally robust classification problem (4) admits an equivalent reformulation as a finite convex optimization problem if either (*i*) the univariate loss function L is piecewise affine or (*ii*) $\mathbb{X} = \mathbb{R}^n$ and L is Lipschitz continuous (but not necessarily piecewise affine).

Theorem 14 (Distributionally robust linear classification) *The following statements hold.*

(i) If $L(z) = \max_{j \in J}\{a_j z + b_j\}$, then (4) is equivalent to

$$\left\{\begin{array}{lll} \inf\limits_{\substack{\boldsymbol{w},\lambda,s_i \\ \boldsymbol{p}_{ij}^+,\boldsymbol{p}_{ij}^-}} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & S_{\mathbb{X}}(a_j \widehat{y}_i \boldsymbol{w} - \boldsymbol{p}_{ij}^+) + b_j + \langle \boldsymbol{p}_{ij}^+, \widehat{\boldsymbol{x}}_i\rangle \leq s_i & i \in [N], j \in [J] \\ & S_{\mathbb{X}}(-a_j \widehat{y}_i \boldsymbol{w} - \boldsymbol{p}_{ij}^-) + b_j + \langle \boldsymbol{p}_{ij}^-, \widehat{\boldsymbol{x}}_i\rangle - \kappa\lambda \leq s_i & i \in [N], j \in [J] \\ & \|\boldsymbol{p}_{ij}^+\|_* \leq \lambda, \|\boldsymbol{p}_{ij}^-\|_* \leq \lambda & i \in [N], j \in [J], \end{array}\right. \tag{17}$$

where $S_{\mathbb{X}}$ denotes the support function of $\mathbb{X}$.

(ii) If $\mathbb{X} = \mathbb{R}^n$ and L is Lipschitz continuous, then (4) is equivalent to

$$\left\{\begin{array}{lll} \inf\limits_{\boldsymbol{w},\lambda,s_i} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ s.t. & L(\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle) \leq s_i & i \in [N] \\ & L(-\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle) - \kappa\lambda \leq s_i & i \in [N] \\ & \operatorname{lip}(L)\|\boldsymbol{w}\|_* \leq \lambda. & \end{array}\right. \tag{18}$$

In the following, we exemplify Theorem 14 for the hinge loss, logloss and smoothed hinge loss functions under the assumption that the input space $\mathbb{X}$ admits the conic representation

$$\mathbb{X} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{C}\boldsymbol{x} \preceq_{\mathcal{C}} \boldsymbol{d}\} \tag{19}$$

for some matrix $\boldsymbol{C}$, vector $\boldsymbol{d}$ and proper convex cone $\mathcal{C}$ of appropriate dimensions. We also assume that $\mathbb{X}$ admits a Slater point $\boldsymbol{x}_{\mathrm{S}} \in \mathbb{R}^n$ with $\boldsymbol{C}\boldsymbol{x}_{\mathrm{S}} \prec_{\mathcal{C}} \boldsymbol{d}$.

Corollary 15 (Support vector machine) *If L represents the hinge loss function and $\mathbb{X}$ is of the form (19), then (4) is equivalent to*

$$\left\{\begin{array}{lll} \inf\limits_{\substack{\boldsymbol{w},\lambda \\ s_i,\boldsymbol{p}_i^+,\boldsymbol{p}_i^-}} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & 1 - \widehat{y}_i\left(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle\right) + \langle \boldsymbol{p}_i^+, \boldsymbol{d} - \boldsymbol{C}\widehat{\boldsymbol{x}}_i\rangle \leq s_i & i \in [N] \\ & 1 + \widehat{y}_i\left(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle\right) + \langle \boldsymbol{p}_i^-, \boldsymbol{d} - \boldsymbol{C}\widehat{\boldsymbol{x}}_i\rangle - \kappa\lambda \leq s_i & i \in [N] \\ & \|\boldsymbol{C}^\top \boldsymbol{p}_i^+ + \widehat{y}_i \boldsymbol{w}\|_* \leq \lambda, \quad \|\boldsymbol{C}^\top \boldsymbol{p}_i^- - \widehat{y}_i \boldsymbol{w}\|_* \leq \lambda & i \in [N] \\ & s_i \geq 0, \quad \boldsymbol{p}_i^+, \boldsymbol{p}_i^- \in \mathcal{C}^* & i \in [N]. \end{array}\right. \tag{20}$$

Corollary 16 (Support vector machine with smooth hinge loss) *If L represents the smooth hinge loss function and $\mathbb{X} = \mathbb{R}^n$, then (4) is equivalent to*

$$\left\{\begin{array}{lll} \min\limits_{\substack{\boldsymbol{w},\lambda,s_i \\ z_i^+,z_i^-,t_i^+,t_i^-}} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ s.t. & \frac{1}{2}(z_i^+ - \widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle)^2 + t_i^+ \leq s_i & i \in [N] \\ & \frac{1}{2}(z_i^- + \widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle)^2 + t_i^- - \kappa\lambda \leq s_i & i \in [N] \\ & 1 - z_i^+ \leq t_i^+, \quad 1 - z_i^- \leq t_i^- & i \in [N] \\ & t_i^+, t_i^- \geq 0 & i \in [N] \\ & \|\boldsymbol{w}\|_* \leq \lambda. & \end{array}\right. \tag{21}$$

Corollary 17 (Logistic regression) *If L represents the logloss function and $\mathbb{X} = \mathbb{R}^n$, then (4) is equivalent to*

$$\left\{\begin{array}{lll} \min\limits_{\boldsymbol{w},\lambda,s_i} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ s.t. & \log\Big(1 + \exp\big(-\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle\big)\Big) \leq s_i & i \in [N] \\ & \log\Big(1 + \exp\big(\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle\big)\Big) - \kappa\lambda \leq s_i & i \in [N] \\ & \|\boldsymbol{w}\|_* \leq \lambda. & \end{array}\right. \tag{22}$$

Remark 18 (Relation to classical regularization) *If $\mathbb{X} = \mathbb{R}^n$ and the weight parameter κ in the transportation metric (16) is set to infinity, then the learning problems portrayed in Corollaries 15–17 all simplify to*

$$\inf_{\boldsymbol{w}} \ \frac{1}{N}\sum_{i=1}^N L(\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle) + \rho\|\boldsymbol{w}\|_*. \tag{23}$$

Thus, in analogy to the case of regression, (23) reduces to an instance of the classical regularized learning problem (2), where the dual norm term $\|\boldsymbol{w}\|_*$ plays the role of the regularization function, while the Wasserstein radius ρ represents the usual regularization weight. Note that if $\kappa = \infty$, then mass transportation along the output space is infinitely expensive, that is, any distribution $\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ can smear out the training samples along

$\mathbb{X}$, but it cannot flip outputs from $+1$ to -1 or vice versa. Thus, classical regularization schemes, which are recovered for $\kappa = \infty$, implicitly assume that output measurements are exact. As this belief is not tenable in most applications, an approach with $\kappa < \infty$ may be more satisfying. We remark that alternative approaches for learning with noisy labels have previously been studied by Lawrence and Schölkopf (2001), Natarajan et al. (2013), and Yang et al. (2012).

Remark 19 (Relation to Tikhonov regularization) *The learning problem*

$$\inf_{\boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} L(\widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle) + c \|\boldsymbol{w}\|_2^2 \tag{24}$$

with Tikhonov regularizer enjoys wide popularity. If L represents the hinge loss, for example, then (24) reduces to the celebrated soft margin support vector machine problem. However, the Tikhonov regularizer appearing in (24) is not explained by a distributionally robust learning problem of the form (4). It is known, however, that (23) with $\|\cdot\|_* = \|\cdot\|_2$ and (24) are equivalent in the sense that for every $\rho \geq 0$ there exists $c \geq 0$ such that the solution of (23) also solves (24) and vice versa (Xu et al., 2009, Corollary 6).

To establish a connection between distributionally robust and classical robust classification as discussed in (Xu et al., 2009), we further investigate the worst-case expected loss of a fixed linear hypothesis $\boldsymbol{w}$.

$$\sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}[L(y \langle \boldsymbol{w}, \boldsymbol{x} \rangle)] \tag{25}$$

Theorem 20 (Extremal distributions in linear classification) *The following statements hold.*

(i) If $L(z) = \max_{j \in J} \{a_j z + b_j\}$, then the worst-case expectation (25) coincides with

$$\left\{ \begin{array}{lll} \sup\limits_{\substack{\alpha_{ij}^+, \alpha_{ij}^- \\ \boldsymbol{q}_{ij}^+, \boldsymbol{q}_{ij}^-}} & \frac{1}{N} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{J} (\alpha_{ij}^+ - \alpha_{ij}^-) a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle + a_j \widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{q}_{ij}^+ - \boldsymbol{q}_{ij}^- \rangle + \sum\limits_{j=1}^{J} b_j & \\ \text{s.t.} & \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{J} \|\boldsymbol{q}_{ij}^+\| + \|\boldsymbol{q}_{ij}^-\| + \kappa \alpha_{ij}^- \leq N \rho & \\ & \sum\limits_{j=1}^{J} \alpha_{ij}^+ + \alpha_{ij}^- = 1 & i \in [N] \\ & \widehat{\boldsymbol{x}}_i + \boldsymbol{q}_{ij}^+ / \alpha_{ij}^+ \in \mathbb{X}, \quad \widehat{\boldsymbol{x}}_i + \boldsymbol{q}_{ij}^- / \alpha_{ij}^- \in \mathbb{X} & i \in [N], j \in [J] \\ & \alpha_{ij}^+, \alpha_{ij}^- \geq 0 & i \in [N], j \in [J] \end{array} \right. \tag{26}$$

for any fixed $\boldsymbol{w}$. Also, if $(\alpha_{ij}^{+\star}, \alpha_{ij}^{-\star}, \boldsymbol{q}_{ij}^{+\star}, \boldsymbol{q}_{ij}^{-\star})$ maximizes (26), then the discrete distribution

$$\mathbb{Q} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{J} \alpha_{ij}^{+\star} \, \delta_{(\widehat{\boldsymbol{x}}_i - \boldsymbol{q}_{ij}^{+\star} / \alpha_{ij}^{+\star}, \, \widehat{y}_i)} + \alpha_{ij}^{-\star} \, \delta_{(\widehat{\boldsymbol{x}}_i - \boldsymbol{q}_{ij}^{-\star} / \alpha_{ij}^{-\star}, \, -\widehat{y}_i)}$$

represents a maximizer for (25).

(ii) If $\mathbb{X} = \mathbb{R}^n$, then the worst-case expectation (25) coincides with the optimal value of

$$\left\{ \begin{array}{lll} \sup\limits_{\alpha_i, \theta} & \operatorname{lip}(L) \|(\boldsymbol{w})\|_* \theta + \frac{1}{N} \sum\limits_{i=1}^{N} (1 - \alpha_i) L(\widehat{y}_i \langle \boldsymbol{w}, \widehat{x}_i \rangle) + \alpha_i L(-\widehat{y}_i \langle \boldsymbol{w}, \widehat{x}_i \rangle) & \\ \text{s.t.} & \theta + \frac{\kappa}{N} \sum\limits_{i=1}^{N} \alpha_i = \rho - \gamma & \\ & 0 \leq \alpha_i \leq 1 & i \in [N] \\ & \theta \geq 0 & \end{array} \right. \tag{27}$$

for $\gamma = 0$. Moreover, if $(\alpha_i^\star(\gamma), \theta^\star(\gamma))$ maximizes (27) for $\gamma > 0$, $\eta(\gamma) = \gamma / (\theta^\star(\gamma) + \kappa - \rho + \gamma + 1)$ and $\boldsymbol{x}^\star$ solves $\max_{\boldsymbol{x}} \{\langle \boldsymbol{w}, \boldsymbol{x} \rangle : \|\boldsymbol{x}\| \leq 1\}$, then the discrete distributions

$$\begin{aligned} \mathbb{Q}_\gamma = & \frac{1}{N} \sum_{i=2}^{N} (1 - \alpha_i^\star(\gamma)) \, \delta_{(\widehat{\boldsymbol{x}}_i, \, \widehat{y}_i)} + \alpha_i^\star(\gamma) \, \delta_{(\widehat{\boldsymbol{x}}_i, \, -\widehat{y}_i)} + \frac{\eta(\gamma)}{N} \, \delta_{(\widehat{\boldsymbol{x}}_1 + \frac{\theta^\star(\gamma) N}{\eta(\gamma)} \boldsymbol{x}^\star, \, \widehat{y}_1)} \\ & + \frac{1 - \eta(\gamma)}{N} \Big[(1 - \alpha_1^\star(\gamma)) \, \delta_{(\widehat{\boldsymbol{x}}_1, \, \widehat{y}_1)} + \alpha_1^\star(\gamma) \, \delta_{(\widehat{\boldsymbol{x}}_1, \, -\widehat{y}_1)} \Big] \end{aligned}$$

for $\gamma \in [0, \min\{\rho, 1\}]$ are feasible and asymptotically optimal in (25) for $\gamma \downarrow 0$.

Theorem 20 shows how one can use convex optimization to construct a sequence of distributions that are asymptotically optimal in (25). Next, we show that the worst-case expected cost (25) is equivalent to a (robust) worst-case cost over a suitably defined uncertainty set if the following assumption holds.

Assumption 21 (Non-separability) *For every $\boldsymbol{w} \in \mathbb{R}^n$ there is a training sample $(\widehat{\boldsymbol{x}}_k, \widehat{y}_k)$ for some $k \leq N$ such that the derivative L' exists at $\widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle$ and satisfies $|L'(\widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle)| = \operatorname{lip}(L)$.*

Remark 22 (Non-separability) *Assumption 21 generalizes the non-separability condition in (Xu et al., 2009, Theorem 3) for the classical and smooth hinge loss functions to more general Lipschitz continuous losses. Note that, in the case of the hinge loss, Assumption 21 effectively stipulates that for any $\boldsymbol{w}$ there exists a training sample $(\widehat{\boldsymbol{x}}_k, \widehat{y}_k)$ with $\widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle < 1$, implying that the dataset cannot be perfectly separated by any linear hypothesis $\boldsymbol{w}$. An equivalent requirement is that the empirical hinge loss is nonzero for every $\boldsymbol{w}$. Similarly, in the case of the smooth hinge loss, Assumption 21 ensures that for any $\boldsymbol{w}$ there is a training sample with $\widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle < 0$, which implies again that the dataset admits no perfect linear separation. Note, however, that the logloss fails to satisfy Assumption 21 as its steepest slope is attained at infinity.*

Theorem 23 (Robust classification) *Suppose that $\mathbb{X} = \mathbb{R}^n$, the loss function L is Lipschitz continuous and the cost of flipping a label in the transportation metric (16) is set to $\kappa = \infty$. Then, the worst-case expected loss (25) provides an upper bound on the (robust) worst-case loss*

$$\left\{ \begin{array}{ll} \sup\limits_{\Delta \boldsymbol{x}_i} & \frac{1}{N} \sum\limits_{i=1}^{N} L\left(\widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i + \Delta \boldsymbol{x}_i \rangle\right) \\ \text{s.t.} & \frac{1}{N} \sum\limits_{i=1}^{N} \|\Delta \boldsymbol{x}_i\| \leq \rho. \end{array} \right. \tag{28}$$

Moreover, if Assumption 21 holds, then (25) *and* (28) *are equal.*

Remark 24 (Tractability of robust classification) *Assume that $\mathbb{X} = \mathbb{R}^n$, while L and $\|\cdot\|$ both admit a tractable conic representation. By Theorem 14, the worst-case expected loss* (25) *can then be computed in polynomial time by solving a tractable convex program. Theorem 23 thus implies that the worst-case loss* (28) *can also be computed in polynomial time if Assumption 21 holds. This confirms Proposition 4 in (Xu et al., 2009). No efficient method for computing* (28) *is know if Assumption 21 fails to hold.*

3.3. Nonlinear Hypotheses: Reproducing Kernel Hilbert Spaces

We now generalize the learning models from Sections 3.1 and 3.2 to nonlinear hypotheses that range over a *reproducing kernel Hilbert space* (RKHS) $\mathbb{H} \subseteq \mathbb{R}^{\mathbb{X}}$ with inner product $\langle \cdot, \cdot \rangle_{\mathbb{H}}$. By definition, $\mathbb{H}$ thus constitutes a complete metric space with respect to the norm $\|\cdot\|_{\mathbb{H}}$ induced by the inner product, and the point evaluation $h \mapsto h(\boldsymbol{x})$ of the functions $h \in \mathbb{H}$ represents a continuous linear functional on $\mathbb{H}$ for any fixed $\boldsymbol{x} \in \mathbb{X}$. The Riesz representation theorem then implies that for every $\boldsymbol{x} \in \mathbb{X}$ there exists a unique function $\Phi(\boldsymbol{x}) \in \mathbb{H}$ such that $h(\boldsymbol{x}) = \langle h, \Phi(\boldsymbol{x}) \rangle_{\mathbb{H}}$ for all $h \in \mathbb{H}$. We henceforth refer to $\Phi : \mathbb{X} \to \mathbb{H}$ as the *feature map* and to $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_+$ with $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle_{\mathbb{H}}$ as the *kernel function*. By construction, the kernel function is symmetric and positive definite, that is, the *kernel matrix* $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N \times N}$ defined through $\boldsymbol{\mathcal{K}}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive definite for all $N \in \mathbb{N}$ and $\{\boldsymbol{x}_i\}_{i \leq N} \subseteq \mathbb{X}$.

By the Moore-Aronszajn theorem, any symmetric and positive definite kernel function k on $\mathbb{X}$ induces a unique RKHS $\mathbb{H} \subseteq \mathbb{R}^{\mathbb{X}}$, which can be represented as

$$\mathbb{H} = \left\{ h \in \mathbb{R}^{\mathbb{X}} : \exists \beta_i \in \mathbb{R},\ \boldsymbol{x}_i \in \mathbb{X}\ \forall i \in \mathbb{N} \text{ with } h(\boldsymbol{x}) = \sum_{i=1}^{\infty} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}) \text{ and } \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}_j) \beta_j < \infty \right\},$$

where the inner product of two arbitrary functions $h_1, h_2 \in \mathbb{H}$ with $h_1(\boldsymbol{x}) = \sum_{i=1}^{\infty} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x})$ and $h_2(\boldsymbol{x}) = \sum_{j=1}^{\infty} \beta'_j k(\boldsymbol{x}'_j, \boldsymbol{x})$ is defined as $\langle h_1, h_2 \rangle_{\mathbb{H}} = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}'_j) \beta'_j$. One may now use the kernel function to define the feature map Φ through $[\Phi(\boldsymbol{x}')](\boldsymbol{x}) = k(\boldsymbol{x}', \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{X}$. This choice is admissible because it respects the consistency condition $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle_{\mathbb{H}} = k(\boldsymbol{x}, \boldsymbol{x}')$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{H}$, and because it implies the desired *reproducing property* $\langle f, \Phi(\boldsymbol{x}') \rangle_{\mathbb{H}} = \sum_{i=1}^{\infty} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}') = f(\boldsymbol{x}')$ for all $f \in \mathbb{H}$ and $\boldsymbol{x}' \in \mathbb{X}$.

In summary, given a symmetric and positive definite kernel function k, there exists an associated RKHS $\mathbb{H}$ and a feature map Φ with the reproducing property. As we will see below, however, to optimize over nonlinear hypotheses in $\mathbb{H}$, knowledge of k is sufficient, and there is no need to construct $\mathbb{H}$ and Φ explicitly.

Assume now that we are given any symmetric and positive definite kernel function k, and construct a distributionally robust learning problem over all *nonlinear* hypotheses in the corresponding RKHS $\mathbb{H}$ via

$$\widehat{J}(\rho) = \inf_{h \in \mathbb{H}} \sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}} \left[\ell(h(\boldsymbol{x}), y) \right], \tag{29}$$

where the transportation metric is given by the Euclidean norm on $\mathbb{X} \times \mathbb{Y}$ (for regression problems) or the separable metric (16) with the Euclidean norm on $\mathbb{X}$ (for classification problems). While problem (29) is hard to solve in general due to the nonlinearity of the hypotheses $h \in \mathbb{H}$, it is easy to solve a lifted learning problem where the inputs $\boldsymbol{x} \in \mathbb{X}$ are replaced with features $\boldsymbol{x}_{\mathbb{H}} \in \mathbb{H}$, while each *nonlinear* hypothesis $h \in \mathbb{H}$ over the input space $\mathbb{X}$ is identified with a *linear* hypothesis $h_{\mathbb{H}} \in \mathbb{H}$ over the feature space $\mathbb{H}$ through the identity $h_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}}) = \langle h, \boldsymbol{x}_{\mathbb{H}} \rangle_{\mathbb{H}}$. Thus, the lifted learning problem can be represented as

$$\widehat{J}_{\mathbb{H}}(\rho) = \inf_{h \in \mathbb{H}} \sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N^{\mathbb{H}})} \mathbb{E}^{\mathbb{Q}} \left[\ell(\langle h, \boldsymbol{x}_{\mathbb{H}} \rangle_{\mathbb{H}}, y) \right], \tag{30}$$

where $\widehat{\mathbb{P}}_N^{\mathbb{H}} = 1/N \sum_{i=1}^{N} \delta_{(\Phi(\widehat{\boldsymbol{x}}_i), \widehat{y}_i)}$ on $\mathbb{H} \times \mathbb{Y}$ denotes the pushforward measure of the emprical distribution $\widehat{\mathbb{P}}_N$ under the feature map Φ induced by k, while $\mathbb{B}_\rho(\widehat{\mathbb{P}}_N^{\mathbb{H}})$ constitutes the Wasserstein ball of radius ρ around $\widehat{\mathbb{P}}_N^{\mathbb{H}}$ corresponding to the transportation metric

$$d_{\mathbb{H}}\left((\boldsymbol{x}_{\mathbb{H}}, y), (\boldsymbol{x}'_{\mathbb{H}}, y') \right) = \begin{cases} \sqrt{\|\boldsymbol{x}_{\mathbb{H}} - \boldsymbol{x}'_{\mathbb{H}}\|_{\mathbb{H}}^2 + (y - y')^2} & \text{for regression problems,} \\ \|\boldsymbol{x}_{\mathbb{H}} - \boldsymbol{x}'_{\mathbb{H}}\|_{\mathbb{H}} + \kappa \mathbb{1}_{\{y \neq y'\}} & \text{for classification problems.} \end{cases}$$

Even though $\widehat{\mathbb{P}}_N^{\mathbb{H}}$ constitutes the pushforward measure of $\widehat{\mathbb{P}}_N$ under Φ, not every distribution $\mathbb{Q}^{\mathbb{H}} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N^{\mathbb{H}})$ can be obtained as the pushforward measure of some $\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$. Thus, we should not expect (29) to be equivalent to (30). Instead, one can show that under a judicious transformation of the Wasserstein radius, (30) provides an upper bound on (29) whenever the kernel function satisfies a calmness condition.

Assumption 25 (Calmness of the kernel) *The kernel function k is calm from above, that is, there exist a concave smooth growth function $g : \mathbb{R}_+ \to \mathbb{R}_+$ with $g(0) = 0$ and $g'(z) \geq 1$ for all $z \in \mathbb{R}_+$ such that*

$$\sqrt{k(\boldsymbol{x}_1, \boldsymbol{x}_1) - 2k(\boldsymbol{x}_1, \boldsymbol{x}_2) + k(\boldsymbol{x}_2, \boldsymbol{x}_2)} \leq g(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2) \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{X}.$$

The calmness condition is non-restrictive. In fact, it is satisfied by most commonly used kernels.

Example 1 (Growth Functions for Popular Kernels) *For most commonly used kernels k on $\mathbb{X} \subseteq \mathbb{R}^n$, we can construct an explicit growth function g that certifies the calmness of k in the sense of Assumption 25. This construction typically relies on elementary estimates. Derivations are omitted for brevity.*

1. ***Linear kernel:*** *For $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle$, we may set $g(z) = z$.*

2. ***Gaussian kernel:*** *For $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\gamma \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}$ with $\gamma > 0$, we may set $g(z) = \max\{\sqrt{2\gamma}, 1\} z$.*

3. ***Laplacian kernel:*** *For $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\gamma \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_1}$ with $\gamma > 0$, we may set $g(z) = \sqrt{2\gamma z \sqrt{n}}$ if $0 \leq z \leq \gamma \sqrt{n}/2$ and $g(z) = z + \gamma \sqrt{n}/2$ otherwise.*

4. ***Polynomial kernel:*** *The kernel* $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\gamma\langle \boldsymbol{x}_1, \boldsymbol{x}_2\rangle + 1)^d$ *with* $\gamma > 0$ *and* $d \in \mathbb{N}$ *fails to satisfy the calmness condition if* $\mathbb{X}$ *is unbounded and* $d > 1$*, in which case* $\sqrt{k(\boldsymbol{x}_1, \boldsymbol{x}_1) - 2k(\boldsymbol{x}_1, \boldsymbol{x}_2) + k(\boldsymbol{x}_2, \boldsymbol{x}_2)}$ *grows superlinearly. If* $\mathbb{X} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 \leq R\}$ *for some* $R > 0$*, however, the polynomial kernel is calm with respect to the growth function*

$$g(z) = \begin{cases} \max\{\frac{1}{2R}\sqrt{2(\gamma R^2 + 1)^d}, 1\}z & \text{if } d \text{ is even,} \\ \max\{\frac{1}{2R}\sqrt{2(\gamma R^2 + 1)^d - 2(1 - \gamma R^2)^d}, 1\}z & \text{if } d \text{ is odd.} \end{cases}$$

Theorem 26 (Lifted learning problems) *If Assumption 25 holds for some growth function* g*, then the following statements hold for all Wasserstein radii* $\rho \geq 0$*.*

(i) For regression problems we have $\widehat{J}(\rho) \leq \widehat{J}_{\mathbb{H}}(\sqrt{2}g(\rho))$*.*

(ii) For classification problems we have $\widehat{J}(\rho) \leq \widehat{J}_{\mathbb{H}}(g(\rho))$*.*

We now argue that the lifted learning problem (30) can be solved efficiently by leveraging the following representer theorem, which generalizes (Schölkopf and Smola, 2001, Theorem 4.2) to non-separable loss functions.

Theorem 27 (Representer theorem) *Assume that we are given a symmetric positive definite kernel* k *on* $\mathbb{X}$ *with corresponding RKHS* $\mathbb{H}$*, a set of training samples* $(\widehat{\boldsymbol{x}}_i, \widehat{y}_i) \in \mathbb{X} \times \mathbb{Y}$*,* $i \leq N$*, and an arbitrary loss function* $f : (\mathbb{X} \times \mathbb{Y} \times \mathbb{R})^N \times \mathbb{R}_+ \to \mathbb{R}$ *that is non-decreasing in its last argument. Then, there exist* $\beta_i \in \mathbb{R}$*,* $i \leq N$*, such that the learning problem*

$$\min_{h \in \mathbb{H}} \; f((\widehat{\boldsymbol{x}}_1, \widehat{y}_1, h(\widehat{\boldsymbol{x}}_1)), \ldots, (\widehat{\boldsymbol{x}}_N, \widehat{y}_N, h(\widehat{\boldsymbol{x}}_N)), \|h\|_{\mathbb{H}}) \tag{31}$$

is solved by a hypothesis $h^\star \in \mathbb{H}$ *representable as* $h^\star(\boldsymbol{x}) = \sum_{i=1}^N \beta_i k(\boldsymbol{x}, \widehat{\boldsymbol{x}}_i)$*.*

The subsequent results involve the Kernel matrix $\boldsymbol{\mathcal{K}} = [\boldsymbol{\mathcal{K}}_{ij}]$ defined through $\boldsymbol{\mathcal{K}}_{ij} = k(\widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{x}}_j)$, $i, j \leq N$. The following theorems demonstrate that the lifted learning problem (30) admits a kernel representation.

Theorem 28 (Kernelized distributionally robust regression) *Suppose that* $\mathbb{X} = \mathbb{R}^n$*,* $\mathbb{Y} = \mathbb{R}$ *and* k *is a symmetric positive definite kernel on* $\mathbb{X}$ *with associated RKHS* $\mathbb{H}$*. If* ℓ *is generated by a convex and Lipschitz continuous loss function* L*, that is,* $\ell(h(\boldsymbol{x}), y) = L(h(\boldsymbol{x}) - y)$*, then* (30) *is equivalent to*

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^N} \frac{1}{N}\sum_{i=1}^N L\Big(\sum_{j=1}^N \boldsymbol{\mathcal{K}}_{ij}\beta_j - \widehat{y}_i\Big) + \rho\,\mathrm{lip}(L)\|(\boldsymbol{\mathcal{K}}^{\frac{1}{2}}\boldsymbol{\beta}, 1)\|_2,$$

and for any of its minimizers $\boldsymbol{\beta}^\star$ *the hypothesis* $h^\star(\boldsymbol{x}) = \sum_{i=1}^N \beta_i^\star k(\boldsymbol{x}, \widehat{\boldsymbol{x}}_i)$ *is optimal in* (30)*.*

Theorem 29 (Kernelized distributionally robust classification) *Suppose that* $\mathbb{X} = \mathbb{R}^n$*,* $\mathbb{Y} = \{+1, -1\}$ *and* k *is a symmetric positive definite kernel on* $\mathbb{X} = \mathbb{R}^n$ *with associated RKHS* $\mathbb{H}$*. If* ℓ *is generated by a convex and Lipschitz continuous loss function* L*, that is,* $\ell(h(\boldsymbol{x}), y) = L(yh(\boldsymbol{x}))$*, then* (30) *is equivalent to*

$$\begin{cases} \min\limits_{\beta_i, \lambda, s_i} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i \\ \text{s.t.} & L\Big(\displaystyle\sum_{j=1}^N \widehat{y}_i \boldsymbol{\mathcal{K}}_{ij}\beta_j\Big) \leq s_i \qquad i \in [N] \\ & L\Big(-\displaystyle\sum_{j=1}^N \widehat{y}_i \boldsymbol{\mathcal{K}}_{ij}\beta_j\Big) - \kappa\lambda \leq s_i \qquad i \in [N] \\ & \mathrm{lip}(L)\|\boldsymbol{\mathcal{K}}^{\frac{1}{2}}\boldsymbol{\beta}\|_2 \leq \lambda, \end{cases} \tag{32}$$

and for any of its minimizers $\boldsymbol{\beta}^\star$ *the hypothesis* $h^\star(\boldsymbol{x}) = \sum_{i=1}^N \beta_i^\star k(\boldsymbol{x}, \widehat{\boldsymbol{x}}_i)$ *is optimal in* (30)*.*

Theorems 28 and 29 show that the lifted learning problem (30) can be solved with similar computational effort as problem (4), that is, optimizing over a possibly infinite-dimensional RKHS of nonlinear hypotheses is not substantially harder than optimizing over the space of linear hypotheses.

Remark 30 (Kernelization in robust regression and classification) *Recall from Theorem 12 that distributionally robust and classical robust linear regression are equivalent if* $\Xi = \mathbb{R}^{n+1}$ *and the training samples are sufficiently dispersed in the sense of Assumption 10. Similarly, Theorem 23 implies that distributionally robust and classical robust linear classification are equivalent if* $\kappa = \infty$ *and the training samples are non-separable in the sense of Assumption 21. One can show that Theorems 12 and 23 naturally extend to nonlinear regression and classification models over an RKHS induced by some symmetric and positive definite kernel. Specifically, one can show that some* lifted *robust learning problem is equivalent to the* lifted *distributionally robust learning problem* (30) *whenever the lifted training samples* $(\boldsymbol{\Phi}(\widehat{\boldsymbol{x}}_1), \widehat{y}_1), \cdots, (\boldsymbol{\Phi}(\widehat{\boldsymbol{x}}_N), \widehat{y}_N)$ *satisfy Assumption 10 (for regression) or 21 (for classification). Theorems 28 and 29 thus imply that the lifted robust regression and classification problems can be solved efficiently under mild regularity conditions whenever Assumptions 10 and 21 hold, respectively. Unfortunately, these conditions are often violated for popular kernels. For example, the lifted samples are always linearly separable under the Gaussian kernel (Xu et al., 2009, p. 1496). In this case, the lifted robust classification problem can never be reduced to an efficiently solvable lifted distributionally robust classification problem of the form* (30)*. In fact, no efficient method for solving the lifted robust classification problem seems to be known. In contrast, the lifted* distributionally *robust learning problems are always efficiently solvable under standard regularity conditions.*

3.4. Nonlinear Hypotheses: Neural Networks[2]

Families of *neural networks* represent particularly expressive classes of nonlinear hypotheses. In the following, we characterize a family $\mathbb{H}$ of neural networks with $M \in \mathbb{N}$ layers through M continuous activation functions $\sigma_m : \mathbb{R}^{n_{m+1}} \to \mathbb{R}^{n_{m+1}}$ and M weight matrices $\boldsymbol{W}_m \in$

2. We are grateful to an anonymous referee for encouraging us to write this section.

$\mathbb{R}^{n_{m+1}\times n_m}$, $m \in [M]$. The weight matrices can encode *fully connected* or *convolutional* layers, for example. If $n_1 = n$ and $n_{M+1} = 1$, then we may set

$$\mathbb{H} = \Big\{ h \in \mathbb{R}^{\mathbb{X}} : \exists \boldsymbol{W}_m \in \mathbb{R}^{n_{m+1}\times n_m},\ m \in [M],\ h(\boldsymbol{x}) = \sigma_M\Big(\boldsymbol{W}_M \cdots \sigma_2\big(\boldsymbol{W}_2\sigma_1(\boldsymbol{W}_1\boldsymbol{x})\big)\cdots\Big)\Big\}.$$

Each hypothesis $h \in \mathbb{H}$ constitutes a neural network and is uniquely determined by the collection of all weight matrices $\boldsymbol{W}_{[M]} := (\boldsymbol{W}_1, \ldots, \boldsymbol{W}_M)$. In order to emphasize the dependence on $\boldsymbol{W}_{[M]}$, we will sometimes use $h(\boldsymbol{x}; \boldsymbol{W}_{[M]})$ to denote the hypotheses in $\mathbb{H}$. Setting $\boldsymbol{x}_1 = \boldsymbol{x}$, the features of the neural network are defined recursively through $\boldsymbol{x}_{m+1} = \sigma_m(\boldsymbol{z}_m)$, where $\boldsymbol{z}_m = \boldsymbol{W}_m\boldsymbol{x}_m$, $m \in [M]$. The features $\boldsymbol{x}_m$, $m = 2, \ldots, M$, correspond to the *hidden* layers of the neural network, while $\boldsymbol{x}_{M+1}$ determines its output.

Example 2 (Activation functions) *The following activation functions are most widely used.*

1. ***Hyperbolic tangent:*** $[\sigma_m(\boldsymbol{z}_m)]_i = (\exp(2[\boldsymbol{z}_m]_i) - 1)/(\exp(2[\boldsymbol{z}_m]_i) + 1)$
2. ***Sigmoid:*** $[\sigma_m(\boldsymbol{z}_m)]_i = 1/(1 + \exp(-[\boldsymbol{z}_m]_i))$
3. ***Softmax:*** $[\sigma_m(\boldsymbol{z}_m)]_i = \exp([\boldsymbol{z}_m]_i)/\sum_{j=1}^{n_{m+1}} \exp([\boldsymbol{z}_m]_j)$
4. ***Rectified linear unit (ReLU):*** $[\sigma_m(\boldsymbol{z}_m)]_i = \max\{0, [\boldsymbol{z}_m]_i\}$
5. ***Exponential linear unit (ELU):*** $[\sigma_m(\boldsymbol{z}_m)]_i = \max\{0, [\boldsymbol{z}_m]_i\} + \min\{0, \alpha(\exp([\boldsymbol{z}_m]_i) - 1)\}$

The distributionally robust learning model over the hypothesis class $\mathbb{H}$ can now be represented as

$$\inf_{h\in\mathbb{H}} \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell\big(h(\boldsymbol{x}), y\big)\right] = \inf_{\boldsymbol{W}_{[M]}} \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell\big(h(\boldsymbol{x}; \boldsymbol{W}_{[M]}), y\big)\right], \tag{33}$$

where we use the transportation metrics (11) and (16) for regression and classification problems, respectively. Moreover, we adopt the standard convention that $\ell(h(\boldsymbol{x}), y) = L(h(\boldsymbol{x}) - y)$ for regression problems and $\ell(h(\boldsymbol{x}, y)) = L(yh(\boldsymbol{x}))$ for classification problems, where L is a convex and Lipschitz continuous univariate loss function. In the following we equip each feature space $\mathbb{R}^{n_m}$ with a norm $\|\cdot\|$, $m \in [M+1]$. By slight abuse of notation, we use the same symbol for all norms even though the norms on different feature spaces may differ. Using the norm on $\mathbb{R}^{n_{m+1}}$, we define the Lipschitz modulus of σ_m as

$$\operatorname{lip}(\sigma_m) := \sup_{\boldsymbol{z},\boldsymbol{z}'\in\mathbb{R}^{n_{m+1}}} \left\{ \frac{\|\sigma(\boldsymbol{z}) - \sigma(\boldsymbol{z}')\|}{\|\boldsymbol{z} - \boldsymbol{z}'\|} : \boldsymbol{z} \neq \boldsymbol{z}' \right\}.$$

We are now ready to state the main result of this section, which provides a conservative upper bound on the distributionally robust learning model (33).

Theorem 31 (Distributionally robust learning with neural networks) *The distributionally robust learning model* (33) *is bounded above by the regularized empirical loss minimization problem*

$$\inf_{\boldsymbol{W}_{[M]}} \frac{1}{N}\sum_{i=1}^{N} \ell(h(\widehat{\boldsymbol{x}}_i; \boldsymbol{W}_{[M]}), \widehat{y}_i) + \rho \operatorname{lip}(L) \max\left\{ \prod_{m=1}^{M} \operatorname{lip}(\sigma_m)\|\boldsymbol{W}_m\|, \frac{c}{\kappa} \right\}, \tag{34}$$

where $c = 1$ for regression problems and $c = \max\{1, 2\sup_{h\in\mathbb{H},\boldsymbol{x}\in\mathbb{X}} |h(\boldsymbol{x})|\}$ for classification problems. Moreover, $\|\boldsymbol{W}_m\| = \sup_{\|\boldsymbol{x}_m\|=1} \|\boldsymbol{W}_m\boldsymbol{x}_m\|$ is the operator norm induced by the norms on $\mathbb{R}^{n_m}$ and $\mathbb{R}^{n_{m+1}}$.

Remark 32 (Uniform upper bound on all neural networks) *For classification problems the constant c in* (34) *represents a uniform upper bound on all neural networks and may be difficult to evaluate in general. It is easy to estimate c, however, if the last activation function is itself bounded such as the softmax function, which yields a probability distribution over the output space. In this case one may simply set $c = 2$.*

The product term $\prod_{m=1}^{M} \operatorname{lip}(\sigma_m)\|\boldsymbol{W}_m\|$ in (34) represents an upper bound on the Lipschitz modulus of $h(\boldsymbol{x}; \boldsymbol{W}_{[M]})$. We emphasize that computing the exact Lipschitz modulus of a neural network is NP-hard even if there are only two layers and all activation functions are of the ReLU type (Scaman and Virmaux, 2018, Theorem 2). In contrast, the upper bound at hand is easy to compute as all activation functions listed in Example 2 have Lipschitz modulus 1 with respect to the Euclidean norms on the domain and range spaces (Gouk et al., 2018; Wiatowski et al., 2016). For more details on how to estimate the Lipschitz moduli of neural networks we refer to (Gouk et al., 2018; Miyato et al., 2018; Neyshabur et al., 2018; Szegedy et al., 2013).

Note that even though (34) represents a finite-dimensional optimization problem over the weight matrices of the neural network, both the empirical prediction loss as well as the regularization term are non-convex in $\boldsymbol{W}_{[M]}$, which complicates numerical solution. If $\kappa = \infty$, however, one can derive an alternative upper bound on the distributionally robust learning model (33) with a *convex* regularization term.

Corollary 33 (Convex regularization term) *If $\kappa = \infty$, then there is $\overline{\rho} \geq 0$ such that the distributionally robust learning model* (33) *is bounded above by the regularized empirical loss minimization problem*

$$\inf_{\boldsymbol{W}_{[M]}} \frac{1}{N}\sum_{i=1}^{N} \ell(h(\widehat{\boldsymbol{x}}_i; \boldsymbol{W}_{[M]}), \widehat{y}_i) + \overline{\rho} \sum_{m=1}^{M} \|\boldsymbol{W}_m\|. \tag{35}$$

As the empirical prediction loss remains non-convex, it is expedient to address problem (35) with local optimization methods such as stochastic gradient descent algorithms. For a comprehensive review of first- and the second-order stochastic gradient algorithms we refer to (Agarwal et al., 2017) and the references therein. In the numerical experiments we will use a stochastic proximal gradient descent algorithm that exploits the convexity of the regularization term and generates iterates $\boldsymbol{W}_{[M]}^k$ for $k \in \mathbb{N}$ according to the update rule

$$\boldsymbol{W}_m^{k+1} = \operatorname{prox}_{\eta_k\overline{\rho}\|\boldsymbol{W}_m\|}\Big(\boldsymbol{W}_m^k - \eta_k \nabla_{\boldsymbol{W}_m} \ell(h(\widehat{\boldsymbol{x}}_{i_k}; \boldsymbol{W}_{[M]}^k), \widehat{y}_{i_k})\Big) \quad \forall m \in [M],$$

where $\eta_k > 0$ is a given step size and i_k is drawn randomly from the index set $[N]$, see, *e.g.*, Nitanda (2014). Here, the proximal operator associated with a convex function $\varphi : \mathbb{R}^{n_{m+1}\times n_m} \to \mathbb{R}$ is defined through

$$\operatorname{prox}_\varphi(\boldsymbol{W}_m) := \arg\min_{\boldsymbol{W}'_m} \ \varphi(\boldsymbol{W}'_m) + \frac{1}{2}\|\boldsymbol{W}'_m - \boldsymbol{W}_m\|_F^2,$$

where $\|\cdot\|_F$ stands for the Frobenius norm. The algorithm is stopped as soon as the improvement of the objective value falls below a prescribed threshold. As the empirical prediction loss is non-convex and potentially non-smooth, the algorithm fails to offer any strong performance guarantees. For the scalability of the algorithm, however, it is essential that the proximal operator can be evaluated efficiently.

Example 3 (Proximal operator) *Suppose that all feature spaces $\mathbb{R}^{n_m}$ are equipped with the p-norm for some $p \in \{1, 2, \infty\}$, which implies that all parameter spaces $\mathbb{R}^{n_{m+1}\times n_m}$ are equipped with the corresponding matrix p-norm. In this case the proximal operator of $\varphi(\boldsymbol{W}_m) = \eta\|\boldsymbol{W}_m\|_p$ for some fixed $\eta > 0$ can be evaluated highly efficiently.*

1. ***MACS** ($p = 1$)**:** The matrix 1-norm returns the maximum absolute column sum (MACS). Evaluating the proximal operator of $\varphi(\boldsymbol{W}_m) = \eta\|\boldsymbol{W}_m\|_1$ amounts to solving the minimization problem*

$$\operatorname{prox}_\varphi(\boldsymbol{W}_m) = \left\{ \begin{array}{ll} \min\limits_{\boldsymbol{W}'_m, u} & \eta u + \sum\limits_{i=1}^{n_m} \|[\boldsymbol{W}'_m]_{:,i} - [\boldsymbol{W}_m]_{:,i}\|_2^2 \\ \text{s.t.} & \|[\boldsymbol{W}'_m]_{:,i}\|_1 \le u \quad i \in [n_m], \end{array} \right.$$

where $[\boldsymbol{W}_m]_{:,i}$ and $[\boldsymbol{W}'_m]_{:,i}$ represent the i-th columns of $\boldsymbol{W}_m$ and $\boldsymbol{W}'_m$, respectively. For any fixed u, the above problem decomposes into n_m projections of the vectors $[\boldsymbol{W}_m]_{:,i}$, $i \in [n_m]$, to the ℓ_1-ball of radius u centered at the origin. Each of these projections can be computed via an efficient sorting algorithm proposed in (Duchi et al., 2008). Next, we can use any line search method such as the golden-section search algorithm to optimize over u, thereby solving the full proximal problem.

2. ***Spectral** ($p = 2$)**:** The matrix 2-norm coincides with the spectral norm, which returns the maximum singular value. In this case, the proximal problem for $\varphi(\boldsymbol{W}_m) = \eta\|\boldsymbol{W}_m\|_2$ can be solved analytically via singular value thresholding (Cai et al., 2010, Theorem 2.1), that is, given the singular value decomposition $\boldsymbol{W}_m = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ with $\boldsymbol{U} \in \mathbb{R}^{n_{m+1}\times n_{m+1}}$ orthogonal, $\boldsymbol{S} \in \mathbb{R}_+^{n_{m+1}\times n_m}$ diagonal and $\boldsymbol{V} \in \mathbb{R}^{n_m\times n_m}$ orthogonal, the proximal operator satisfies*

$$\operatorname{prox}_\varphi(\boldsymbol{W}_m) = \operatorname{prox}_\varphi(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top) = \boldsymbol{U}\widetilde{\boldsymbol{S}}\boldsymbol{V}^\top, \qquad \text{where} \quad \widetilde{\boldsymbol{S}}_{ij} = \max\{\boldsymbol{S}_{ij} - \eta, 0\}\,.$$

The singular value decomposition can be accelerated using a randomized algorithm proposed in (Halko et al., 2011).

3. ***MARS** ($p = \infty$)**:** The matrix ∞-norm returns the maximum absolute row sum (MARS) and thus satisfies $\|\boldsymbol{W}_m\|_\infty = \|\boldsymbol{W}_m^\top\|_1$. Therefore, one can use the iterative scheme developed for MACS to compute the proximal operator of $\varphi(\boldsymbol{W}_m) = \eta\|\boldsymbol{W}_m\|_\infty$ by simply transposing the weight matrix $\boldsymbol{W}_m$.*

The convergence behavior of the stochastic proximal gradient descent algorithm can be further improved by including a momentum term inside the proximal operator, see, *e.g.*, Loizou and Richtárik (2017).

4. Generalization Bounds

Generalization bounds constitute upper confidence bounds on the out-of-sample error. Traditionally, generalization bounds are derived by controlling the complexity of the hypothesis space, which is typically quantified in terms of its VC-dimension or via covering numbers or Rademacher averages (Shalev-Shwartz and Ben-David, 2014). Strengthened generalization bounds for large margin classifiers can be obtained by improving the estimates of the VC-dimension and the Rademacher average (Shivaswamy and Jebara, 2007, 2010). We will now demonstrate that distributionally robust learning models of the type (4) or (30) enjoy simple new generalization bounds that can be obtained under minimal assumptions. In particular, they do not rely on any notions of hypothesis complexity and may therefore even extend to hypothesis spaces with infinite VC-dimensions. Our approach is reminiscent of the generalization theory for robust support vector machines portrayed in (Xu et al., 2009), which also replaces measures of hypothesis complexity with robustness properties. However, we derive explicit finite sample guarantees, while (Xu et al., 2009) establishes asymptotic consistency results. Moreover, we relax some technical conditions used in (Xu et al., 2009) such as the compactness of the input space $\mathbb{X}$.

The key enabling mechanism of our analysis is a measure concentration property of the Wasserstein metric, which holds whenever the unknown data-generating distribution has exponentially decaying tails.

Assumption 34 (Light-tailed distribution) *There exist constants $a > 1$ and $A > 0$ and a reference point $\boldsymbol{\xi}' \in \mathbb{R}^{n+1}$ such that $\mathbb{E}^{\mathbb{P}}[\exp\left(d(\boldsymbol{\xi}, \boldsymbol{\xi}')\right)^a)] \le A$, where d denotes the usual mass transportation cost.*

Theorem 35 (Measure concentration (Fournier and Guillin, 2015, Theorem 2)) *If Assumption 34 holds, then we have*

$$\mathbb{P}^N\Big\{W(\mathbb{P}, \widehat{\mathbb{P}}_N) \ge \rho\Big\} \le \left\{ \begin{array}{ll} c_1 \exp\left(-c_2 N \rho^{\max\{n+1,2\}}\right) & \text{if } \rho \le 1, \\ c_1 \exp\left(-c_2 N \rho^a\right) & \text{if } \rho > 1, \end{array} \right. \tag{36}$$

for all $N \ge 1$, $n \ne 1$, and $\rho > 0$, where the constants $c_1, c_2 > 0$ depend only on a, A, d and n.[3]

Theorem 35 asserts that the empirical distribution $\widehat{\mathbb{P}}_N$ converges exponentially fast to the unknown data-generating distribution $\mathbb{P}$, in probability with respect to the Wasserstein metric, as the sample size N tends to infinity. We can now derive simple generalization bounds by increasing the Wasserstein radius ρ until the violation probability on the of the right hand side of (36) drops below a prescribed significance level $\eta \in (0, 1]$. Specifically,

3. A similar but slightly more complicated inequality also holds for the special case $n = 1$; see (Fournier and Guillin, 2015, Theorem 2) for details.

Theorem 36 borrowed from (Mohajerin Esfahani and Kuhn, 2018, Theorem 3.5) implies that $\mathbb{P}^N\{\mathbb{P} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)\} \geq 1-\eta$ for any $\rho \geq \rho_N(\eta)$, where

$$\rho_N(\eta) = \begin{cases} \left(\dfrac{\log(c_1/\eta)}{c_2 N}\right)^{\frac{1}{\max\{n+1,2\}}} & \text{if } N \geq \frac{\log(c_1/\eta)}{c_2}, \\ \left(\dfrac{\log(c_1/\eta)}{c_2 N}\right)^{\frac{1}{a}} & \text{if } N < \frac{\log(c_1/\eta)}{c_2}. \end{cases} \tag{37}$$

Theorem 36 (Basic generalization bound) *If Assumption 34 holds, then*

$$\mathbb{P}^N\left\{\mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] \leq \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] \;\; \forall \boldsymbol{w} \in \mathbb{R}^n\right\} \geq 1-\eta \tag{38}$$

for any $N \geq 1$, $n \neq 1$, $\eta \in (0,1]$ and $\rho \geq \rho_N(\eta)$.

Remark 37 (Discussion of basic generalization bound) *The following comments are in order.*

I. ***Performance guarantees for optimal hypotheses:*** *If $\widehat{J}(\rho)$ denotes the minimum and $\widehat{\boldsymbol{w}}$ a minimizer of the distributionally robust learning problem* (4)*, then Theorem 36 implies that*

$$\mathbb{P}^N\left\{\mathbb{E}^{\mathbb{P}}\left[\ell(\langle \widehat{\boldsymbol{w}}, \boldsymbol{x}\rangle, y)\right] \leq \widehat{J}(\rho)\right\} \geq 1-\eta$$

for any $N \geq 1$, $n > 1$, $\eta \in (0,1]$ and $\rho \geq \rho_N(\eta)$.

II. ***Light-tail assumption:*** *Assumption 34 is restrictive but unavoidable for any measure concentration result of the type described in Theorem 35. It is automatically satisfied if the input-output pair has bounded support or is known to follow a Gaussian or exponential distribution, for instance.*

III. ***Asymptotic consistency:*** *It is clear from* (37) *that for any fixed $\eta \in (0,1]$, the radius $\rho_N(\eta)$ tends to 0 as N increases. Moreover, Theorem 3.6 in (Mohajerin Esfahani and Kuhn, 2018) implies that if η_N converges to 0 at a carefully chosen rate (*e.g.*, $\eta_N = \exp(-\sqrt{N})$), then the solution of the distributionally robust learning problem* (4) *with Wasserstein radius $\rho = \rho_N(\eta_N)$ converges almost surely to the solution of the ideal learning problem that minimizes the out-of-sample error under the unknown true distribution $\mathbb{P}$.*

IV. ***Curse of dimensionality:*** *The Wasserstein radius* (37) *has two decay regimes. For small N, $\rho_N(\eta)$ decays as $N^{-\frac{1}{a}}$, and for large N it is proportional to $N^{-\frac{1}{n+1}}$. We thus face a curse of dimensionality for large sample sizes. In order to half the Wasserstein radius, one has to increase N by a factor of 2^n. This curse of dimensionality is fundamental,* i.e.*, the dependence of the measure concentration result in Theorem 35 on the input dimension n cannot be improved for generic distributions $\mathbb{P}$; see (Weed and Bach, 2019) or (Fournier and Guillin, 2015, Section 1.3). Improvements are only possible in special cases,* e.g.*, if $\mathbb{P}$ is finitely supported.*

V. ***Extension to nonlinear hypotheses:*** *Theorem 36 directly extends to any distributionally robust learning problem over an RKHS $\mathbb{H}$ induced by some symmetric and positive definite kernel function k. Specifically, if k is calm in the sense of Assumption 25 with growth function g, then we have*

$$\mathbb{P}^N\left\{\mathbb{E}^{\mathbb{P}}[\ell(h(\boldsymbol{x}), y)] \leq \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N^{\mathbb{H}})} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle h, \boldsymbol{x}_{\mathbb{H}}\rangle_{\mathbb{H}}, y)\right] \;\; \forall h \in \mathbb{H}\right\} \geq 1-\eta \tag{39}$$

for any $N \geq 1$, $n \neq 1$, $\eta \in (0,1]$ and $\rho \geq cg(\rho_N(\eta))$, where $c = \sqrt{2}$ for regression problems and $c = 1$ for classification problems. To see this, note that the inclusion $\mathbb{P} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ implies

$$\mathbb{E}^{\mathbb{P}}[\ell(h(\boldsymbol{x}), y)] \leq \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(h(\boldsymbol{x}), y)\right] \leq \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N^{\mathbb{H}})} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle h, \boldsymbol{x}_{\mathbb{H}}\rangle_{\mathbb{H}}, y)\right] \quad \forall h \in \mathbb{H}, \tag{40}$$

where the second inequality follows from the proof of Theorem 26. The generalization bound (39) *thus holds because $\mathbb{P}^N\{\mathbb{P} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)\} \geq 1-\eta$ for any $\rho \geq \rho_N(\eta)$. Note that the rightmost term in* (40) *can be computed for any finitely generated hypothesis $h \in \mathbb{H}$ representable as $h(\boldsymbol{x}) = \sum_{i=1}^N \beta_i k(\boldsymbol{x}, \widehat{\boldsymbol{x}}_i)$, which follows from Theorems 28 and 29, while the middle term is hard to compute. We emphasize that the generalization bound* (39) *does not rely on any notion of hypothesis complexity and remains valid even if $\mathbb{H}$ has infinite VC-dimension (*e.g.*, if $\mathbb{H}$ is generated by the Gaussian kernel).*

Theorem 35 provides a confidence set for the unknown probability distribution $\mathbb{P}$, and Theorem 36 uses this confidence set to construct a uniform generalization bound on the prediction error under $\mathbb{P}$. The radius of the confidence set for $\mathbb{P}$ decreases slowly due to a curse of dimensionality, but the decay rate is essentially optimal. This does not imply that the decay rate of the generalization bound (38) is optimal, too. In fact, the worst-case expected error over a Wasserstein ball of radius ρ can be a $(1-\eta)$-confidence bound on the expected error under $\mathbb{P}$ even if the Wasserstein ball fails to contain $\mathbb{P}$ with confidence $1-\eta$. Thus, the measure concentration result of Theorem 35 is too powerful for our purposes and leads to an over-conservative generalization bound. Below we will show that the curse of dimensionality in the generalization bound (38) can be broken if we impose the following restriction on the hypothesis space.

Assumption 38 (Hypothesis space) *The space of admissible hypotheses in* (4) *is restricted to $\mathbb{W} \subseteq \mathbb{R}^n$. There exists $\underline{\Omega} > 0$ with $\inf_{\boldsymbol{w}\in\mathbb{W}} \|(\boldsymbol{w}, -1)\|_* \geq \underline{\Omega}$ if* (4) *is a regression problem and $\inf_{\boldsymbol{w}\in\mathbb{W}} \|\boldsymbol{w}\|_* \geq \underline{\Omega}$ if* (4) *is a classification problem. Similarly, there exists $\overline{\Omega} \geq 0$ with $\sup_{\boldsymbol{w},\boldsymbol{w}'\in\mathbb{W}} \|\boldsymbol{w}-\boldsymbol{w}'\|_\infty \leq \overline{\Omega}$.*

Theorem 39 (Improved generalization bound) *Suppose that Assumptions 34 and 38 hold, and the function L is Lipschitz continuous. Moreover, assume that $\Xi = \mathbb{R}^{n+1}$ and $M_n = \max_{i\leq n} \|\boldsymbol{e}_i^{n+1}\|_*$ if* (4) *is a regression problem, while $\Xi = \mathbb{R}^n \times \{-1,1\}$ and $M_n = \max_{i\leq n} \|\boldsymbol{e}_i^n\|_*$ if* (4) *is a classification problem, where $\boldsymbol{e}_i^n$ is the i-th standard basis vector in $\mathbb{R}^n$. Then, there exist constants $c_3 \geq 1$, $c_4 > 0$ depending only on*

the light tail constants a and A such that the generalization bound (38) *holds for any* $N \geq \max\left\{(16n/c_4)^2, 16\log(c_3/\eta)/c_4\right\}$, $\eta \in (0,1]$ *and* $\rho \geq \rho'_N(\eta)$, *where*

$$\rho'_N(\eta) = \frac{2\overline{\Omega}}{\sqrt{N}\underline{\Omega}}\left[M_n n A + \sqrt{\frac{n\log(\sqrt{N}) + \log(c_3/\eta)}{c_4}}\right].$$

The improved generalization bound from Theorem 38 does not suffer from a curse of dimensionality. In fact, in order to half the Wasserstein radius $\rho'_N(\eta)$, it suffices to increase the sample size N by a factor of 4, irrespective of the input dimension n.

Remark 40 (Discussion of improved generalization bound) *The following comments are in order.*

I. ***Bounds on hypothesis space:*** *Assumption 38 imposes upper and lower bounds on $\mathbb{W}$. The upper bound enables us to control the difference between the empirical and the true expected loss uniformly across all admissible hypotheses. This bound is less restrictive than the uniform bound on the loss function used to derive Rademacher generalization bounds (see,* e.g.*, (Shalev-Shwartz and Ben-David, 2014, Theorem 26.4)), which essentially imposes upper bounds both on the hypotheses and the input-output pairs. The lower bound in Assumption 38 is restrictive for classification problems but trivially holds for regression problems because $\|(\boldsymbol{w},-1)\|_*$ is uniformly bounded away from zero for any (dual) norm on $\mathbb{R}^{n+1}$.*

II. ***Breaking the curse of dimensionality:*** *By leveraging Assumption 38, Theorem 39 reduces the critical Wasserstein radius in the generalization bound* (38) *from $\rho_N(\eta) \propto \mathcal{O}([\log(\eta^{-1})/N]^{1/(n+1)})$, which suffers from a curse of dimensionality, to $\rho'_N(\eta) \propto \mathcal{O}([\log(\eta^{-1}) + n\log(N))/N]^{1/2})$, which essentially follows a square root law reminiscent of the central limit theorem.*

5. Error and Risk Estimation

Once a hypothesis $h(\boldsymbol{x})$ has been chosen, it is instructive to derive pessimistic *and* optimistic estimates of its out-of-sample prediction error (in the case of regression) or its out-of-sample risk (in the case of classification). We will argue below that the distributionally robust optimization techniques developed in this paper also offer new perspectives on error and risk estimation. For ease of exposition, we ignore any support constraints, that is, we set $\mathbb{X} = \mathbb{R}^n$ and $\mathbb{Y} = \mathbb{R}$ (for regression) or $\mathbb{X} = \mathbb{R}^n$ and $\mathbb{Y} = \{+1, -1\}$ (for classification). Moreover, we focus on linear hypotheses of the form $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x}\rangle$. Note, however, that all results extend directly to conic representable support sets and to nonlinear hypotheses.

In the context of regression, we aim to estimate the prediction error defined as $\mathcal{E}(\boldsymbol{w}) = \mathbb{E}^{\mathbb{P}}\left[|y - \langle \boldsymbol{w}, \boldsymbol{x}\rangle|\right]$ or, more precisely, the mean absolute prediction error under the unknown data-generating distribution $\mathbb{P}$. As usual, we assume that the transportation metric d is induced by a norm $\|\cdot\|$ on the input-output space $\mathbb{R}^{n+1}$.

Theorem 41 (Error bounds in linear regression) *The prediction error admits the following estimates.*

(i) The worst-case error $\mathcal{E}_{\max}(\boldsymbol{w}) = \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[|y - \langle \boldsymbol{w}, \boldsymbol{x}\rangle|\right]$ is given by

$$\mathcal{E}_{\max}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} |\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle| + \rho\|(\boldsymbol{w},-1)\|_*. \tag{41a}$$

[(ii)] The best-case error $\mathcal{E}_{\min}(\boldsymbol{w}) = \inf_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[|y - \langle \boldsymbol{w}, \boldsymbol{x}\rangle|\right]$ is given by

$$\mathcal{E}_{\min}(\boldsymbol{w}) = \max\left\{\frac{1}{N}\sum_{i=1}^{N} |\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle| - \rho\|(\boldsymbol{w},-1)\|_*, 0\right\} \tag{41b}$$

In the context of classification, we aim to quantify the risk $\mathcal{R}(\boldsymbol{w}) = \mathbb{P}\left[y \neq \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\right]$, that is, the misclassification probability under the unknown true distribution $\mathbb{P}$. Note that the risk can equivalently be defined as the expectation of a characteristic function, that is, $\mathcal{R}(\boldsymbol{w}) = \mathbb{E}^{\mathbb{P}}[\mathbb{1}_{\{y\neq\langle \boldsymbol{w},\boldsymbol{x}\rangle\}}]$. As usual, we assume that the transportation metric d is of the form (16), where $\kappa \geq 0$ is the cost of flipping a label.

Theorem 42 (Risk bounds in linear classification) *The risk admits the following estimates.*

(i) The worst-case risk $\mathcal{R}_{\max}(\boldsymbol{w}) = \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}[\mathbb{1}_{\{y\neq\langle \boldsymbol{w},\boldsymbol{x}\rangle\}}]$ is given by

$$\mathcal{R}_{\max}(\boldsymbol{w}) = \left\{\begin{array}{lll} \min\limits_{\substack{\lambda, s_i \\ r_i, t_i}} & \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} s_i & \\ \text{s.t.} & 1 - r_i\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle \leq s_i & i \in [N] \\ & 1 + t_i\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \lambda\kappa \leq s_i & i \in [N] \\ & r_i\|\boldsymbol{w}\|_* \leq \lambda,\ t_i\|\boldsymbol{w}\|_* \leq \lambda & i \in [N] \\ & r_i, t_i, s_i \geq 0 & i \in [N]. \end{array}\right. \tag{42a}$$

(ii) The best-case risk $\mathcal{R}_{\min}(\boldsymbol{w}) = \inf_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}[\mathbb{1}_{\{y\neq\langle \boldsymbol{w},\boldsymbol{x}\rangle\}}]$ is given by

$$\mathcal{R}_{\min}(\boldsymbol{w}) = 1 - \left\{\begin{array}{lll} \min\limits_{\substack{\lambda, s_i \\ r_i, t_i}} & \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} s_i & \\ \text{s.t.} & 1 + r_i\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle \leq s_i & i \in [N] \\ & 1 - t_i\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \lambda\kappa \leq s_i & i \in [N] \\ & r_i\|\boldsymbol{w}\|_* \leq \lambda,\ t_i\|\boldsymbol{w}\|_* \leq \lambda & i \in [N] \\ & r_i, t_i, s_i \geq 0 & i \in [N]. \end{array}\right. \tag{42b}$$

We emphasize that, as the hypothesis $\boldsymbol{w}$ is fixed, the error and risk estimation problems (41) and (42) constitute tractable linear programs that can be solved highly efficiently.

Remark 43 (Confidence intervals for error and risk) *If the Wasserstein radius is set to $\rho_N(\eta/2)$ defined in* (37)*, where $\eta \in (0,1]$ is a prescribed significance level, then Theorem 36 implies that $\mathcal{E}(\widehat{\boldsymbol{w}}) \in [\mathcal{E}_{\min}(\widehat{\boldsymbol{w}}), \mathcal{E}_{\max}(\widehat{\boldsymbol{w}})]$ and $\mathcal{R}(\widehat{\boldsymbol{w}}) \in [\mathcal{R}_{\min}(\widehat{\boldsymbol{w}}), \mathcal{R}_{\max}(\widehat{\boldsymbol{w}})]$ with confidence $1-\eta$ for any $\widehat{\boldsymbol{w}} \in \mathbb{R}^n$ that may even depend on the training data. Theorem 41 implies that the confidence interval for the true error $\mathcal{E}(\widehat{\boldsymbol{w}})$ can be calculated analytically from* (41)*, while Theorem 42 implies that the confidence interval for the true risk $\mathcal{R}(\widehat{\boldsymbol{w}})$ can be computed efficiently by solving the tractable linear programs* (42)*.*

Remark 44 (Extension to nonlinear hypotheses) *By using the tools of Section 3.3, Theorems 41 and 42 generalize immediately to nonlinear hypotheses that range over a RKHS. Specifically, we can formulate lifted error and risk estimation problems where the inputs $\boldsymbol{x} \in \mathbb{X}$ are replaced with features $\boldsymbol{x}_{\mathbb{H}} \in \mathbb{H}$, while each* nonlinear *hypothesis $h \in \mathbb{H}$ over the input space $\mathbb{X}$ is identified with a* linear *hypothesis $h_{\mathbb{H}} \in \mathbb{H}$ over the feature space $\mathbb{H}$ through the identity $h_{\mathbb{H}}(\boldsymbol{x}_{\mathbb{H}}) = \langle h, \boldsymbol{x}_{\mathbb{H}} \rangle_{\mathbb{H}}$. Tractability is again facilitated by Theorem 27, which allows us to focus on finitely parameterized hypotheses of the form $h(\boldsymbol{x}) = \sum_{i=1}^N \beta_i k(\boldsymbol{x}, \widehat{\boldsymbol{x}}_i)$.*

6. Numerical Results

We showcase the power of regularization via mass transportation in various applications based on standard datasets from the literature. All optimization problems are implemented in Python and solved with Gurobi 7.5.1 All experiments are run on an Intel XEON CPU (3.40GHz), and the corresponding codes are made publicly available at https://github.com/sorooshafiee/Regularization-via-Transportation.

6.1. Regularization with Pre-selected Parameters

We first assess how the out-of-sample performance of a distributionally robust support vector machine (DRSVM) is impacted by the choice of the Wasserstein radius ρ, the cost κ of flipping a label, and the kernel function k. To this end, we solve three binary classification problems from the MNIST database (LeCun et al., 1998) targeted at distinguishing pairs of similar handwritten digits (1-vs-7, 3-vs-8, 4-vs-9). In the first experiment we optimize over linear hypotheses and use the separable transporation metric (16) involving the ∞-norm on the input space. All results are averaged over 100 independent trials. In each trial, we randomly select 500 images to train the DRSVM model (20) and use the remaining images for testing. The correct classification rate (CCR) on the test data, averaged across all 100 trials, is visualized in Figure 1 as a function of the Wasserstein radius ρ for each $\kappa \in \{0.1, 0.25, 0.5, 0.75, \infty\}$. The best out-of-sample CCR is obtained for $\kappa = 0.25$ uniformly across all Wasserstein radii, and performance deteriorates significantly when κ is reduced or increased. Recall from Remark 18 that, as κ tends to infinity, the DRSVM reduces to the classical regularized support vector machine (RSVM) with 1-norm regularizer. Thus, the results of Figure 1 indicate that regularization via mass transportation may be preferable to classical regularization in terms of the maximum achievable out-of-sample CCR. More specifically, we observe that the out-of-sample CCR of the best DRSVM ($\kappa = 0.25$) displays a slightly higher and significantly wider plateau around the optimal regularization parameter ρ than the classical RSVM ($\kappa = \infty$). This suggests that the regularization parameter in DRSVMs may be easier to calibrate from data than in RSVMs, a conjecture that will be put

Table 1: Average out-of-sample CCR scores of the DRSVM with learned parameters.

	Polynomial		Laplacian		Gaussian	
	RSVM	DRSVM	RSVM	DRSVM	RSVM	DRSVM
1-vs-7	98.9 ± 0.2	99.1 ± 0.2	98.3 ± 0.5	98.5 ± 0.4	99.1 ± 0.2	99.2 ± 0.2
3-vs-8	95.2 ± 0.4	97.0 ± 0.4	96.5 ± 0.4	96.8 ± 0.4	97.0 ± 0.3	97.2 ± 0.3
4-vs-9	95.0 ± 0.4	96.5 ± 0.4	95.8 ± 0.6	96.0 ± 0.6	96.8 ± 0.4	96.9 ± 0.4

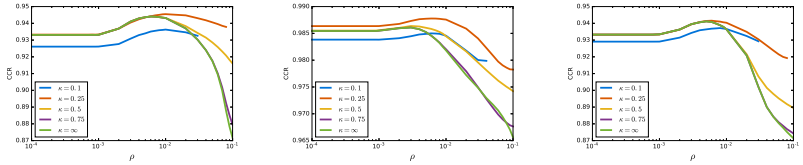
to scrutiny in Section 6.2. Finally, Figure 1 reveals that the standard (unregularized) support vector machine (SVM), which can be viewed as a special case of the DRSVM with $\rho = 0$, is dominated by the RSVMs and DRSVMs across a wide range of regularization parameters.[4] Note that the SVM problem (20) with $\rho = 0$ reduces to a linear program and may thus suffer from multiple optimal solutions. This explains why the limiting out-of-sample CCR for $\rho \downarrow 0$ changes with κ.

6.2. Regularization with Learned Parameters

It is easy to read off the best regularization parameters ρ and κ from the charts in Figure 1. As these charts are constructed from more than 12,000 test samples, however, they are not accessible in the training phase. In practice, ρ and κ must be calibrated from the training data alone. This motivates us to revisit the three classification problems from Section 6.1 using a fully data-driven procedure, where all free model parameters are calibrated via 5-fold cross validation; see, *e.g.*, (Abu-Mostafa et al., 2012, § 4.3.3). Moreover, to evaluate the benefits of kernelization, we now solve a generalized DRSVM model of the form (32), which implicitly optimizes over all nonlinear hypotheses in some RKHS. As explained in Section 3.3, kernelization necessitates the use of the separable transportation metric (16) with the Euclidean norm on the input space.

All free parameters of the resulting DRSVM model are restricted to finite search grids in order to ease the computational burden of cross validation. Specifically, we select the Wasserstein radius ρ from within $\{b \cdot 10^e : b \in \{1,5\}, e \in \{1,2,3,4\}\}$ and the label flipping cost κ from within $\{0.1, 0.25, 0.5, 0.75, \infty\}$. Moreover, we select the degree d of the polynomial kernel from within $\{1,2,3,4,5\}$ and the peakedness parameter γ of the Laplacian and Gaussian kernels from within $\{\frac{1}{100}, \frac{1}{81}, \frac{1}{64}, \frac{1}{49}, \frac{1}{36}, \frac{1}{25}\}$. Otherwise, we use the same experimental setup as in Section 6.1. Table 1 reports the averages and standard deviations of the CCR scores on the test data based on 100 independent trials. We observe that the DRSVM (ρ, κ, d, and γ learned by cross validation) outperforms the RSVM (ρ, d and γ learned by cross validation, $\kappa = \infty$) consistently across all tested kernel functions (Polynomial, Laplacian, Gaussian). Note that the DRSVM with polynomial kernel subsumes the non-kernelized DRSVM (20) as a special case because the polynomial kernel with $d = 1$ coincides with the linear kernel.

4. By slight abuse of notation, we use the acronym 'SVM' to refer to the unregularized empirical hinge loss minimization problem even though the traditional formulations of the support vector machine involve a Tikhonov regularization term.



(a) 1-vs-7 (b) 3-vs-8 (c) 4-vs-9

Figure 1: Average out-of-sample CCR scores of the DRSVM with pre-selected parameters.

In the third experiment, we assess the out-of-sample performance of the DRSVM (20) for different transportation metrics on 10 standard datasets from the UCI repository (Bache and Lichman, 2013). Specifically, we use different variants of the separable transportation metric (16), where distances in the input space are measured via a p-norm with $p \in \{1, 2, \infty\}$. We focus exclusively on linear hypotheses because the kernelization techniques described in Section 3.3 are only available for $p = 2$. The DRSVM is compared against the standard (unregularized) SVM and the RSVM with q-norm regularizer ($\frac{1}{p} + \frac{1}{q} = 1$). All results are averaged across 100 independent trials. In each trial, we randomly select 75% of the data for training and the remaining 25% for testing. The training dataset is first standardized to zero mean and unit variance along each coordinate axis. The Wasserstein radius ρ and the label flipping cost κ in the DRSVM as well as the regularization weight ρ in the RSVM are estimated via stratified 5-fold cross validation.

Classifier performance is now quantified in terms of the *receiver operating characteristic* (ROC) curve, which plots the true positive rate (percentage of correctly classified test samples with true label $y = 1$) against the false positive rate (percentage of *in*correctly classified test samples with true label $y = -1$) by sweeping the discrimination threshold. Specifically, we use the *area under the ROC curve* (AUC) as a measure of classifier performance. AUC does not bias on the size of the test data and is a more appropriate performance measure than CCR in the presence of an unbalanced label distribution in the training data. We emphasize that most of the considered datasets are indeed imbalanced, and thus a high CCR score would not necessarily provide evidence of superior classifier performance. The averages and standard deviations of the AUC scores based on 100 trials are reported in Table 2. The results suggest that the DRSVM outperforms the RSVM in terms of AUC for all norms by about the same amount by which the RSVM outperforms the classical hinge loss minimization, consistently across all datasets.

6.3. Multi-Label Classification

The aim of object recognition is to discover instances of particular object classes in digital images. We now describe an object recognition experiment based on the PASCAL VOC 2007 dataset Everingham et al. (2010) consisting of 9,963 images, which are pre-partitioned into 25% for training, 25% for validation and 50% for testing. Each image is annotated with 20 binary labels corresponding to 20 given object categories (the n-th label is set to

Table 2: Average out-of-sample AUC scores of the SVM, RSVM and DRSVM.

		$p = \infty$ / $q = 1$		$p = 2$ / $q = 2$		$p = 1$ / $q = \infty$	
	SVM	RSVM	DRSVM	RSVM	DRSVM	RSVM	DRSVM
Australian	91.6 ± 3.0	91.5 ± 3.2	92.0 ± 2.5	92.0 ± 2.2	92.3 ± 2.0	91.9 ± 2.8	92.2 ± 2.4
Blood transfusion	73.7 ± 3.8	73.8 ± 3.8	75.5 ± 3.8	74.9 ± 3.5	75.5 ± 3.7	75.4 ± 3.4	75.4 ± 3.7
Climate model	93.8 ± 3.9	94.4 ± 4.0	94.3 ± 4.0	94.3 ± 3.8	94.0 ± 4.0	93.6 ± 3.9	93.9 ± 4.0
Cylinder	72.0 ± 3.7	71.2 ± 4.0	72.1 ± 4.1	71.3 ± 4.0	71.8 ± 3.8	71.5 ± 3.8	72.2 ± 3.7
Heart	90.4 ± 2.7	90.1 ± 2.7	90.3 ± 2.7	90.6 ± 2.6	90.9 ± 2.5	90.5 ± 2.6	90.7 ± 2.6
Ionosphere	85.0 ± 4.9	89.7 ± 4.5	89.2 ± 4.3	90.4 ± 3.7	89.9 ± 3.9	86.0 ± 4.9	87.2 ± 4.8
Liver disorders	60.5 ± 0.0	61.1 ± 0.7	61.7 ± 0.7	61.2 ± 0.3	61.7 ± 0.5	61.1 ± 0.4	61.8 ± 0.5
QSAR	90.5 ± 1.5	90.5 ± 1.6	91.0 ± 1.6	90.5 ± 1.5	91.2 ± 1.5	90.6 ± 1.5	91.1 ± 1.6
Splice	92.1 ± 0.0	93.0 ± 0.4	93.1 ± 0.1	92.5 ± 0.1	92.6 ± 0.1	92.0 ± 0.1	92.5 ± 0.1
Thoracic surgery	61.7 ± 7.1	61.5 ± 6.5	64.6 ± 6.6	64.4 ± 6.4	64.3 ± 7.0	64.0 ± 6.3	64.6 ± 6.3

$+1$ if the image contains the n-th object and to -1 otherwise). A multi-label classifier is a function that predicts all labels of an unlabelled input image. The ability of a classifier to detect objects belonging to any fixed category is measured by the *average precision* (AP), which is defined in Everingham et al. (2010) as (a proxy for) the area under the classifier's precision-recall curve. The overall performance of a classifier is quantified by the *mean average precision* (mAP), that is, the arithmetic mean of the AP scores across all object categories.

In the first scenario, we train a separate binary RSVM and DRSVM classifier for each of the 20 object categories. This classifier predicts whether an object of the respective category appears in the input image. At the beginning we preprocess the entire dataset by resizing each image to 256×256 pixels and extracting the central patch of 244×244 pixels. As shown in (Chatfield et al., 2014; Donahue et al., 2014; Zeiler and Fergus, 2014), the features generated by the penultimate layer of a deep convolutional neural network trained on a large image dataset provide a powerful image descriptor. Using the ALEXNET neural network trained on the ImageNet dataset (Krizhevsky et al., 2012), we can thus compress each (preprocessed) image of the PASCAL VOC 2007 dataset into 1,000 meaningful features. We normalize these feature vectors to lie on the unit sphere. When training the RSVM and DRSVM classifiers, we can thus work with these feature vectors instead of the corresponding images. Moreover, we restrict attention to linear hypotheses and assume that transportation distances in the input-output space are measured by the separable metric (16) with the Euclidean norm on the input space. We tune the Wasserstein radius $\rho \in \{b \cdot 10^e : b \in \{1, \ldots, 9\}, e \in \{-2, -3, -4\}\}$ and the label flipping cost $\kappa \in \{0.1, 0.2, \ldots, 1, \infty\}$ via the holdout method using the validation data. As usual, we fix $\kappa = \infty$ for RSVM. Table 3 reports the AP scores of the RSVM and DRSVM models for each object category. The ensemble of all 20 binary RSVM or DRSVM classifiers, respectively, can be viewed as a naïve multi-label classifier that predicts all labels of an image. As DRSVM outperforms RSVM on an object-by-object basis, it also wins in terms of mAP.

In the second scenario, we construct a proper multi-label classifier by fine-tuning the last layer of the pre-trained ALEXNET network. To this end, we replace the original M-th layer of the network with a new fully connected layer characterized by a parameter matrix $\boldsymbol{W}_M \in \mathbb{R}^{20 \times 1000}$, and we set σ_M to the Sigmoid activation function. The resulting classifer outputs for each of the 20 object categories a probability that an object from the respective

category appears in the input image. The quality of a classifier (which is encoded by $\boldsymbol{W}_M$) is measured by the cross-entropy loss function, which naturally generalizes the logloss to multiple labels. The resulting empirical loss minimization problem is enhanced with a regularization term proportional to $\|\boldsymbol{W}_M\|_{1,1}$ (Lasso), $\|\boldsymbol{W}_M\|_F^2$ (Tikhonov), $\|\boldsymbol{W}_M\|_1$ (MACS), $\|\boldsymbol{W}_M\|_2$ (Spectral) or $\|\boldsymbol{W}_M\|_\infty$ (MARS). By using similar arguments as in Section 3.4, one can show that the empirical cross-entropy with MACS, Spectral or MARS regularization term overestimates the worst-case expected cross-entropy over all distributions of $(\boldsymbol{x}_M, \boldsymbol{x}_{M+1})$ in a Wasserstein ball provided that the transportation cost is given by

$$d((\boldsymbol{x}_M, \boldsymbol{x}_{M+1}), (\boldsymbol{x}'_M, \boldsymbol{x}'_{M+1})) = \|\boldsymbol{x}_M - \boldsymbol{x}'_M\|_p + \kappa \mathbb{1}_{\{\boldsymbol{x}_{M+1} \neq \boldsymbol{x}'_{M+1}\}}$$

for $\kappa = \infty$, whenever $p = 1$, $p = 2$ or $p = \infty$, respectively. Thus, the MACS, Spectral and MARS regularization terms admit a distributionally robust interpretation.

We use the stochastic proximal gradient descent algorithm of Section 3.4 to tune $\boldsymbol{W}_M$, including an additional momentum term with weight 0.9. As in (Krizhevsky et al., 2012), we split the training phase into 100 epochs, each corresponding to a complete pass through the training dataset in a random order. As the ALEXNET requires input images of size 244×244, in each iteration we extract a random patch of 244×244 pixels from the current image and flip it horizontally at random. This procedure effectively augments the training dataset. The initial step size is set to 10^{-3} and then reduced by a factor of 10 after every 7 epochs. The algorithm terminates after 100 epochs. We preprocess the images in the validation and test datasets as in Scenario 1 and tune the regularization weights via the holdout method using the validation data. Table 3 reports the AP and mAP scores of the different classifiers that were tested. These results suggest that fine-tuning the last layer of a pre-trained neural network may improve classifier performance. We observe that the spectral norm regularizer, which has a distributionally robust interpretation, consistently outperforms almost all other methods. For further details on the experimental setup (such as the exact search grids for all hyperparameters) we refer to the code publicized on Github.

6.4. Generalization Bounds

The next experiment estimates the scaling behavior of the smallest Wasserstein radius that verifies the generalization bound (38) for the synthetic `threenorm` classification problem (Breiman, 1996). The experiment involves 1,000 simulation trials. In each trial we generate N training samples for some $N \in \{10, \ldots, 90\} \cup \{100, \ldots, 1{,}000\}$ as well as 10^5 test samples. Each sample $(\boldsymbol{x}, y) \in \mathbb{R}^{20} \times \{-1, 1\}$ is constructed as follows. The label y is drawn uniformly from $\{-1, 1\}$. If $y = -1$, then $\boldsymbol{x}$ is drawn from a standard multivariate normal distribution shifted by $(c, \ldots, c)$ or $(-c, \ldots, -c)$ with equal probabilities, where $c = 2/\sqrt{20}$. If $\widehat{y} = 1$, on the other hand, then $\boldsymbol{x}$ is drawn from a standard multivariate normal distribution shifted by $(c, -c, +c, \ldots, -c)$.

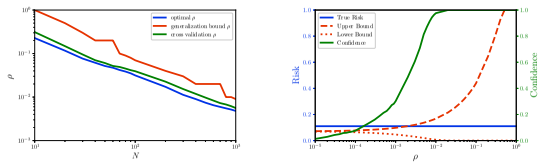
We now describe three different approaches to choose the Wasserstein radius ρ in the DRSVM (20) with transportation cost (16), where $\kappa = \infty$ and $\|\cdot\|$ represents the ∞-norm on the input space. Throughout the experiment we use $P = \{b \cdot 10^{-e} : b \in \{1, \ldots, 10\}, e \in \{1, \ldots, 5\}\}$ as the search space for ρ. Approach 1 ('cross validation') calibrates the Wasserstein radius as before via 5-fold cross validation based solely on the N training samples. This approach reflects what would typically be done in practice.

Table 3: AP scores of different multi-label classifiers.

	Scenario 1		Scenario 2				
	RSVM	DRSVM	Lasso	Tikhonov	MACS	Spectral	MARS
	84.80	84.85	85.50	84.26	84.35	83.89	85.53
	78.54	78.49	76.18	76.55	76.37	75.67	76.22
	82.19	82.19	83.37	83.11	83.52	84.19	83.08
	79.55	79.56	77.65	78.04	77.70	78.92	77.53
	37.52	37.98	38.10	39.73	39.53	38.75	38.10
	72.05	72.03	70.05	69.43	69.17	70.28	69.77
	83.08	83.12	83.10	83.67	83.50	82.92	83.17
	80.60	80.56	79.87	79.86	80.07	79.85	79.82
	54.64	54.64	54.40	54.76	53.94	54.72	54.55
	47.82	53.13	52.06	52.10	51.62	55.54	51.19
	54.26	58.88	63.41	65.23	65.15	66.79	62.95
	75.81	75.81	76.92	77.39	77.26	76.54	76.97
	82.74	82.72	82.17	81.89	81.6	80.81	81.9
	72.69	72.88	74.70	75.41	74.76	76.48	74.31
	90.36	90.36	90.07	90.30	90.22	90.35	90.09
	50.22	51.90	50.20	50.18	50.27	51.39	50.20
	60.75	63.57	71.78	71.3	70.39	71.64	71.40
	56.85	56.98	52.15	54.36	54.65	55.12	51.94
	85.09	85.03	84.89	84.55	84.41	85.43	84.96
	69.02	69.08	64.73	65.85	65.63	64.26	64.48
mAP	69.92	70.69	70.56	70.90	70.71	71.20	70.40

Approaches 2 and 3 both solve (20) based on the empirical distribution induced by the N training samples and select the Wasserstein radius using the 10^5 test samples. Specifically, approach 2 ('optimal') chooses the Wasserstain radius that leads to the lowest test error, while approach 3 ('generalization bound') selects the smallest Wasserstein radius for which the optimal value of (20) exceeds the expected loss on the test samples in at least 95% of all trials, that is, it approximates the smallest Wasserstein radius that verifies the generalization bound (38) for $\eta = 5\%$. As the test samples are not available in the training phase, the last two approaches are not implementable in practice, and we merely study them to gain insights. Figure 2(a) visualizes all resulting Wasserstein radii as a function of N. Note that the radii obtained with the first two approaches are uncertain as they depend on a particular choice of the training samples. Figure 2(a) thus only shows their averages across all simulation trials. In contrast, the radii obtained with the third approach depend on the training sample sets of all 1,000 trials and are thus essentially deterministic.

We observe that the Wasserstein radii of all three approaches decay approximately as $1/\sqrt{N}$, which is in line with the theoretical generalization bound of Theorem 39. We expect



(a) Dependence of the Wasserstein radius on the number of training samples

(b) Confidence bounds on the risk

Figure 2: Results of the `threenorm` classification problem.

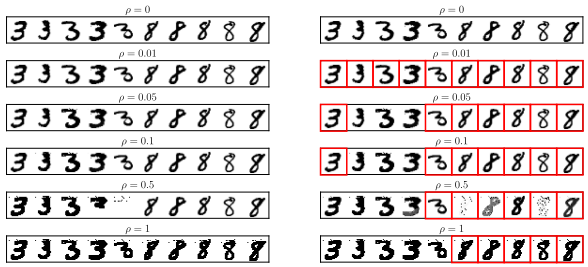
this decay rate to be optimal because any faster decay would be in conflict with the central limit theorem. Note also that our results empirically confirm Theorem 39 even though we did not impose any restrictions on $\mathbb{W}$ as dictated by Assumption 38. This suggests that Theorem 39 might remain valid under weaker conditions.

In the experiment underlying Figure 2(b), we first fix $\widehat{\boldsymbol{w}}$ to an optimal solution of (20) for $\rho = 0.1$ and $N = 100$. Figure 2(b) shows the true risk $\mathcal{R}(\widehat{\boldsymbol{w}})$ and its confidence bounds given by Theorem 42. As expected, for $\rho = 0$ the upper and lower bounds coincide with the empirical risk on the training data, which is a lower bound for the true risk on the test data due to over-fitting effects. As ρ increases, the confidence interval between the bounds widens and eventually covers the true risk. For instance, at $\rho \approx 0.009$ the confidence interval is given by $[0.008, 0.162]$ and contains the true risk with probability $1 - \eta = 95\%$.

6.5. Worst-Case Distributions

Consider again the 3-vs-8 classification problems from the MNIST database (LeCun et al., 1998) and fix $\boldsymbol{w}^\star$ to an optimal solution of the empirical hinge loss minimization problem. The goal of the last experiment is to evaluate the *worst-case* hinge loss of $\boldsymbol{w}^\star$ for different Wasserstein radii $\rho \in \{0, 0.01, 0.05, 0.1, 0.5, 1\}$ and label flipping costs $\kappa \in \{0, \infty\}$ and to investigate the corresponding worst-case distributions, which are computable by virtue of Theorem 20(i). As each input constitutes a vector of pixels intensities between zero and one, we impose support constraints of the form $\boldsymbol{C}\boldsymbol{x} \leq \boldsymbol{d}$ with $\boldsymbol{C} = [\boldsymbol{I}, -\boldsymbol{I}]^\top$ and $\boldsymbol{d} = [\boldsymbol{1}^\top, \boldsymbol{0}^\top]^\top$.

For illustrative purposes we only use the $N = 10$ first datapoints in the MNIST dataset as training samples. Each training sample $\widehat{\boldsymbol{x}}_i$ corresponds to four discretization points ($\widehat{\boldsymbol{x}}_i + \boldsymbol{q}^{+\star}_{ij}/\alpha^{+\star}_{ij}$ and $\widehat{\boldsymbol{x}}_i + \boldsymbol{q}^{-\star}_{ij}/\alpha^{-\star}_{ij}$ for $j = 1, 2$) in the worst-case distribution obtained from (26). We observe that for every i exactly one out of these four points has probability $\frac{1}{N}$, while all others have probability 0. Figure 3 depicts only those 10 discretization points that have nonzero probability for a fixed ρ and κ. As expected, the perturbations of the training samples are more severe for larger Wasserstein radii. For $\kappa = \infty$ these scenarios must have the same labels as the corresponding training samples. For $\kappa = 0$, on the other hand, the labels can be flipped at no cost (flipped labels are indicated by red frames). Each scenario group shown in Figure 3 can thus be viewed as a worst-case training dataset for the corresponding Wasserstein radius and label flipping cost.



(a) $\kappa = \infty$

(b) $\kappa = 0$

Figure 3: Discretization points (input images) of the worst-case distribution for different ρ and κ. Red frames indicate that the corresponding labels are flipped under the worst-case distribution.

Acknowledgments

We gratefully acknoweldge financial support from the Swiss National Science Foundation under grants BSCGI0_157733 and P2EZP2_165264.

Appendix: Proofs

A.1. Proofs of Section 3

The proof of Theorem 4 requires three preparatory Lemmas. The first lemma is adapted from (Mohajerin Esfahani and Kuhn, 2018) and asserts that the worst-case expectation over a Wasserstein ball can be re-expressed as a classical robust optimization problem.

Lemma 45 (Robust reformulation) *Set $\widehat{\boldsymbol{\xi}}_i = (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)$ for all $i \leq N$. For any measurable integrand $I(\boldsymbol{\xi})$ that is bounded above by a Lipschitz continuous function we have*

$$\sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[I(\boldsymbol{\xi})\right] = \inf_{\lambda \geq 0} \; \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} \sup_{\boldsymbol{\xi} \in \Xi} \; I(\boldsymbol{\xi}) - \lambda d(\boldsymbol{\xi}, \widehat{\boldsymbol{\xi}}_i). \tag{A.1}$$

Proof By the definition of the Wasserstein ball we have

$$\sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[I(\boldsymbol{\xi})\right] = \left\{ \begin{array}{ll} \sup\limits_{\Pi} & \displaystyle\int_{\Xi^2} I(\boldsymbol{\xi})\, \Pi(\mathrm{d}\boldsymbol{\xi}, \mathrm{d}\boldsymbol{\xi}') \\ \text{s.t.} & \Pi \text{ is a joint distribution of } \boldsymbol{\xi} \\ & \text{and } \boldsymbol{\xi}' \text{ with marginals } \mathbb{Q} \text{ and } \widehat{\mathbb{P}}_N \\ & \displaystyle\int_{\Xi^2} d\left(\boldsymbol{\xi}, \boldsymbol{\xi}'\right)\, \Pi(\mathrm{d}\boldsymbol{\xi}, \mathrm{d}\boldsymbol{\xi}') \leq \rho \end{array} \right.$$

$$= \left\{ \begin{array}{ll} \sup\limits_{\mathbb{Q}^i} & \dfrac{1}{N} \displaystyle\sum_{i=1}^N \int_\Xi I(\boldsymbol{\xi})\, \mathbb{Q}^i(\mathrm{d}\boldsymbol{\xi}) \\ \text{s.t.} & \displaystyle\int_\Xi \mathbb{Q}^i(\mathrm{d}\boldsymbol{\xi}) = 1 \quad i \in [N] \\ & \dfrac{1}{N} \displaystyle\sum_{i=1}^N \int_\Xi d(\boldsymbol{\xi}, \widehat{\boldsymbol{\xi}}_i)\, \mathbb{Q}^i(\mathrm{d}\boldsymbol{\xi}) \le \rho. \end{array} \right.$$

Note that the integral of $I(\boldsymbol{\xi})$ exists under every $\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ because $I(\boldsymbol{\xi})$ admits a Lipschitz continuous majorant. The last equality in the above expression holds because the marginal distribution of $\boldsymbol{\xi}'$ is the uniform distribution on the training samples, which implies that Π is completely determined by the conditional distributions $\mathbb{Q}^i$ of $\boldsymbol{\xi}$ given $\boldsymbol{\xi}' = \widehat{\boldsymbol{\xi}}_i$, that is, $\Pi(\mathrm{d}\boldsymbol{\xi}, \mathrm{d}\boldsymbol{\xi}') = \frac{1}{N}\sum_{i=1}^N \delta_{\widehat{\boldsymbol{\xi}}_i}(\mathrm{d}\boldsymbol{\xi}')\mathbb{Q}^i(\mathrm{d}\boldsymbol{\xi})$. The resulting generalized moment problem over the normalized measures $\mathbb{Q}^i$ admits the semi-infinite dual

$$\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[I(\boldsymbol{\xi})\right] = \left\{ \begin{array}{ll} \inf\limits_{\lambda, s_i} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i \\ \text{s.t.} & \sup\limits_{\boldsymbol{\xi}\in\Xi} \; I(\boldsymbol{\xi}) - \lambda d(\boldsymbol{\xi}, \widehat{\boldsymbol{\xi}}_i) \le s_i \qquad i \in [N] \\ & \lambda \ge 0. \end{array} \right.$$

Strong duality holds for any $\rho > 0$ due to (Shapiro, 2001, Proposition 3.4). The claim then follows by eliminating s_i. ■

Lemma 46 *For any $a \in \mathbb{R}$, $\boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}} \in \mathbb{R}^d$, $\gamma \in \mathbb{R}_+$ and $\boldsymbol{\zeta} \in \mathcal{Z}$, where $\mathcal{Z} \subseteq \mathbb{R}^d$ is a closed convex set, we have*

$$\sup_{\boldsymbol{\zeta}\in\mathcal{Z}} \; a\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\| = \left\{ \begin{array}{ll} \inf\limits_{\boldsymbol{p}} & S_{\mathcal{Z}}(a\boldsymbol{\beta} - \boldsymbol{p}) + \langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle \\ \text{s.t.} & \|\boldsymbol{p}\|_* \le \gamma. \end{array} \right.$$

Proof We have

$$\begin{aligned} \sup_{\boldsymbol{\zeta}\in\mathcal{Z}} \; a\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\| &= \sup_{\boldsymbol{\zeta}\in\mathcal{Z}} \; \inf_{\|\boldsymbol{p}\|_*\le\gamma} \; a\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle - \langle \boldsymbol{p}, \boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\rangle \\ &= \inf_{\|\boldsymbol{p}\|_*\le\gamma} \; \sup_{\boldsymbol{\zeta}\in\mathcal{Z}} \; a\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle - \langle \boldsymbol{p}, \boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\rangle \\ &= \inf_{\|\boldsymbol{p}\|_*\le\gamma} \; \sup_{\boldsymbol{\zeta}\in\mathbb{R}^d} \; \langle a\boldsymbol{\beta} - \boldsymbol{p}, \boldsymbol{\zeta}\rangle - \delta_{\mathcal{Z}}(\boldsymbol{\zeta}) + \langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle \\ &= \inf_{\|\boldsymbol{p}\|_*\le\gamma} \; S_{\mathcal{Z}}(a\boldsymbol{\beta} - \boldsymbol{p}) + \langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle, \end{aligned}$$

where the first equality follows from the definition of the dual norm, the second equality holds due to the minimax theorem (Bertsekas, 2009, Proposition 5.5.4), and the last equality holds because the support function $S_{\mathcal{Z}}$ is the conjugate of the indicator function $\delta_{\mathcal{Z}}$. Thus, the claim follows. ■

Lemma 47 *If $L(z)$ is a convex and Lipschitz continuous loss function, $\boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}} \in \mathbb{R}^d$ and $\gamma > 0$, then*

$$\sup_{\boldsymbol{\zeta}\in\mathbb{R}^d} \; L(\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle) - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\| = \left\{ \begin{array}{ll} L(\langle \boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}\rangle) & \text{if } \operatorname{lip}(L)\|\boldsymbol{\beta}\|_* \le \gamma \\ +\infty & \text{otherwise.} \end{array} \right.$$

Proof Note that $L(\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle) - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\|$ constitutes a difference of convex functions and may thus be neither convex nor concave in $\boldsymbol{\zeta}$. In order to maximize this function, we re-write $I(\boldsymbol{\zeta}) = L(\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle)$ as an upper envelope of infinitely many affine functions. To this end, we express the conjugate of $I(\boldsymbol{\zeta})$ as

$$I^*(\boldsymbol{z}) = \sup_{\boldsymbol{\zeta}} \langle \boldsymbol{z}, \boldsymbol{\zeta}\rangle - L(\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle) = \sup_{t,\boldsymbol{\zeta}} \left\{ \langle \boldsymbol{z}, \boldsymbol{\zeta}\rangle - L(t) : t = \langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle \right\} = \inf_{\theta} \left\{ L^*(\theta) : \theta\boldsymbol{\beta} = \boldsymbol{z} \right\},$$

where the last equality follows from strong Lagrangian duality, which holds because Slater's constraint qualification is trivially satisfied in the absence of inequality constraints (Bertsekas, 2009, Proposition 5.3.1). Defining $\Theta = \{\theta \in \mathbb{R} : L^*(\theta) < \infty\}$ as the effective domain of $L^*(\theta)$, we may then replace $\theta \in \mathbb{R}$ with $\theta \in \Theta$ in the last expression. As $I(\boldsymbol{\zeta})$ is convex and continuous, it coincides with its bi-conjugate, that is,

$$I(\boldsymbol{\zeta}) = I^{**}(\boldsymbol{\zeta}) = \sup_{\boldsymbol{z}} \; \langle \boldsymbol{z}, \boldsymbol{\zeta}\rangle - I^*(\boldsymbol{z}) = \sup_{\theta\in\Theta} \; \langle \theta\boldsymbol{\beta}, \boldsymbol{\zeta}\rangle - L^*(\theta).$$

In other words, we have represented $I(\boldsymbol{\zeta})$ as the upper envelope of infinitely many linear functions. Using this representation, we obtain

$$\begin{aligned} \sup_{\boldsymbol{\zeta}} \; I(\boldsymbol{\zeta}) - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\| &= \sup_{\boldsymbol{\zeta}} \; I^{**}(\boldsymbol{\zeta}) - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\| \\ &= \sup_{\boldsymbol{\zeta}} \; \sup_{\theta\in\Theta} \; \langle \theta\boldsymbol{\beta}, \boldsymbol{\zeta}\rangle - L^*(\theta) - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\| \\ &= \sup_{\theta\in\Theta} \; \sup_{\boldsymbol{\zeta}} \; \inf_{\|\boldsymbol{p}\|_*\le\gamma} \; \theta\langle \boldsymbol{\beta}, \boldsymbol{\zeta}\rangle - L^*(\theta) - \langle \boldsymbol{p}, \boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\rangle \\ &= \sup_{\theta\in\Theta} \; \inf_{\|\boldsymbol{p}\|_*\le\gamma} \; \sup_{\boldsymbol{\zeta}} \; \langle \theta\boldsymbol{\beta} - \boldsymbol{p}, \boldsymbol{\zeta}\rangle - L^*(\theta) + \langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle, \end{aligned}$$

where the last equality holds due to (Bertsekas, 2009, Proposition 5.5.4). Evaluating the maximization over $\boldsymbol{\zeta}$ yields

$$\begin{aligned} \sup_{\boldsymbol{\zeta}} \; I(\boldsymbol{\zeta}) - \gamma\|\boldsymbol{\zeta} - \widehat{\boldsymbol{\zeta}}\| &= \sup_{\theta\in\Theta} \; \inf_{\|\boldsymbol{p}\|_*\le\gamma} \left\{ \begin{array}{ll} \langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle - L^*(\theta) & \text{if } \boldsymbol{p} = \theta\boldsymbol{\beta} \\ +\infty & \text{otherwise} \end{array} \right. \\ &= \sup_{\theta\in\Theta} \left\{ \begin{array}{ll} \langle \theta\boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}\rangle - L^*(\theta) & \text{if } \|\theta\boldsymbol{\beta}\|_* \le \gamma \\ +\infty & \text{otherwise} \end{array} \right. \\ &= \left\{ \begin{array}{ll} \sup\limits_{\theta\in\Theta} \; \theta\langle \boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}\rangle - L^*(\theta) & \text{if } \sup\limits_{\theta\in\Theta} \|\theta\boldsymbol{\beta}\|_* \le \gamma \\ +\infty & \text{otherwise} \end{array} \right. \\ &= \left\{ \begin{array}{ll} L(\langle \boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}\rangle) & \text{if } \sup\limits_{\theta\in\Theta} |\theta| \cdot \|\boldsymbol{\beta}\|_* \le \gamma \\ +\infty & \text{otherwise.} \end{array} \right. \end{aligned}$$

Thus, the claim follows by noting that $\sup_\theta\{|\theta| : L^*(\theta) < \infty\}$ represents the Lipschitz modulus of L. ■

Proof of Theorem 4 To prove assertion (i), we apply Lemma 45 to the integrand $f(\boldsymbol{x}, y) = L(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y)$ with $L(z) = \max_{j\leq J}\{a_j z + b_j\}$ to obtain

$$\begin{aligned}
&\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] \\
&\quad = \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{(\boldsymbol{x},y)\in\Xi} L(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y) - \lambda\|(\boldsymbol{x}, y) - (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\| \\
&\quad = \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \max_{j\in[J]} \sup_{(\boldsymbol{x},y)\in\Xi} a_j(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y) + b_j - \lambda\|(\boldsymbol{x}, y) - (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\| \\
&\quad = \begin{cases} \inf\limits_{\lambda, \boldsymbol{p}_{ij}, u_{ij}} & \lambda\rho + \sum\limits_{i=1}^N \max\limits_{j\leq J} S_\Xi(a_j\boldsymbol{w} - \boldsymbol{p}_{ij}, -a_j - u_{ij}) + \langle \boldsymbol{p}_{ij}, \widehat{\boldsymbol{x}}_i\rangle + u_{ij}\widehat{y}_i + b_j \\ \text{s.t.} & \|(\boldsymbol{p}_{ij}, u_{ij})\|_* \leq \lambda \qquad i\in[N], j\in[J], \end{cases}
\end{aligned}$$

where the last equality follows from Lemma 46. The claim now follows by introducing auxiliary epigraphical variables s_i for the max-terms in the objective function and by including $\boldsymbol{w}$ as a decision variable.

To prove assertion (ii), we apply Lemma 45 to the integrand $f(\boldsymbol{x}, y) = L(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y)$, where L is a Lipschitz continuous convex loss function. Thus we find

$$\begin{aligned}
\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell_{\boldsymbol{w}}(\boldsymbol{x}, y)\right] &= \inf_{\lambda\geq 0}\ \lambda\rho + \sum_{i=1}^N \sup_{\boldsymbol{x},y} L(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - y) - \lambda\|(\boldsymbol{x}, y) - (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\| \\
&= \begin{cases} \inf\limits_{\lambda} & \lambda\rho + \frac{1}{N}\sum\limits_{i=1}^N L(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i) \\ \text{s.t.} & \operatorname{lip}(L)|\theta|\cdot\|(\boldsymbol{w}, -1)\|_* \leq \lambda, \end{cases}
\end{aligned}$$

where the last equality uses Lemma 47. Next, we eliminate λ and include $\boldsymbol{w}$ as a decision variable. ■

Proof of Corollary 5 Note that the Huber loss function $L(z)$ coincides with the inf-convolution of $\frac{1}{2}z^2$ and $\delta|z|$ and can thus be expressed as $L(z) = \min_{z_1}\ \frac{1}{2}z_1^2 + \delta|z - z_1|$. Moreover, the Lipschitz modulus of the Huber loss function is δ. The rest of the proof follows from Theorem 4(ii). ■

Proof of Corollary 6 Notice that the ϵ-insensitive loss function is a piecewise linear function with $J = 3$ pieces, see Section 2.1. By strong conic duality, the support function of $\Xi = \{(\boldsymbol{x}, y) \in \mathbb{R}^{n+1} : \boldsymbol{C}_1\boldsymbol{x} + \boldsymbol{c}_2 y \preceq_{\mathcal{C}} \boldsymbol{d}\}$ can be re-expressed as

$$S_\Xi(\boldsymbol{z}_1, z_2) = \sup_{\boldsymbol{x},y}\{\langle \boldsymbol{z}_1, \boldsymbol{x}\rangle + z_2 y : \boldsymbol{C}_1\boldsymbol{x} + \boldsymbol{c}_2 y \preceq_{\mathcal{C}} \boldsymbol{d}\} = \inf_{\boldsymbol{q}\in\mathcal{C}^*}\left\{\langle \boldsymbol{q}, \boldsymbol{d}\rangle : \boldsymbol{C}_1^\top\boldsymbol{q} = \boldsymbol{z}_1,\ \boldsymbol{c}_2^\top\boldsymbol{q} = z_2\right\}.$$

Strong duality holds because Ξ admits a Slater point. The rest of proof follows from Theorem 4(i). ■

Proof of Corollary 7 The pinball loss function is a piecewise linear function with $J = 2$ pieces, see Section 2.1. The rest of proof follows from the dual representation of the support function $S_\Xi(\boldsymbol{z}_1, z_2)$, which is known from the proof of Corollary 6, and from Theorem 4(i). ■

The proof of Theorem 9 is based on the following preparatory lemma.

Lemma 48 *If $\mathcal{Z} \subseteq \mathbb{R}^d$ is a non-empty convex closed set, $\widehat{\boldsymbol{\zeta}} \in \mathcal{Z}$, $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\alpha, \gamma \geq 0$, then we have*

$$\inf_{\boldsymbol{p}}\ \alpha S_{\mathcal{Z}}(\boldsymbol{\beta} - \boldsymbol{p}) + \alpha\langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle + \gamma\|\boldsymbol{p}\|_* = \begin{cases} \sup\limits_{\|\boldsymbol{q}\|\leq\gamma} & \alpha\langle \boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}\rangle + \langle \boldsymbol{\beta}, \boldsymbol{q}\rangle \\ \text{s.t.} & \widehat{\boldsymbol{\zeta}} + \boldsymbol{q}/\alpha \in \mathcal{Z}. \end{cases}$$

Proof If $\alpha = 0$, then the optimal values of both opimization problems vanish due to our conventions of extended arithmetic, and thus the claim trivially holds. If $\alpha > 0$, however, we have

$$\begin{aligned}
\inf_{\boldsymbol{p}}\ \alpha S_{\mathcal{Z}}(\boldsymbol{\beta} - \boldsymbol{p}) + \alpha\langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle + \gamma\|\boldsymbol{p}\|_* &= \inf_{\boldsymbol{p}} \sup_{\|\boldsymbol{q}\|\leq\gamma}\ \alpha S_{\mathcal{Z}}(\boldsymbol{\beta} - \boldsymbol{p}) + \alpha\langle \boldsymbol{p}, \widehat{\boldsymbol{\zeta}}\rangle + \langle \boldsymbol{p}, \boldsymbol{q}\rangle \\
&= \sup_{\|\boldsymbol{q}\|\leq\gamma} \inf_{\boldsymbol{p}}\ \alpha S_{\mathcal{Z}}(\boldsymbol{\beta} - \boldsymbol{p}) + \langle \boldsymbol{p}, \alpha\widehat{\boldsymbol{\zeta}} + \boldsymbol{q}\rangle \\
&= \sup_{\|\boldsymbol{q}\|\leq\gamma} \inf_{\boldsymbol{z}}\ \alpha S_{\mathcal{Z}}(\boldsymbol{z}) + \langle \boldsymbol{\beta} - \boldsymbol{z}, \alpha\widehat{\boldsymbol{\zeta}} + \boldsymbol{q}\rangle \\
&= \sup_{\|\boldsymbol{q}\|\leq\gamma}\ \alpha\langle \boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}\rangle + \langle \boldsymbol{\beta}, \boldsymbol{q}\rangle - \alpha\big(\sup_{\boldsymbol{z}}\langle \boldsymbol{z}, \widehat{\boldsymbol{\zeta}} + \boldsymbol{q}/\alpha\rangle - S_{\mathcal{Z}}(\boldsymbol{z})\big) \\
&= \sup_{\|\boldsymbol{q}\|\leq\gamma}\ \alpha\langle \boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}\rangle + \langle \boldsymbol{\beta}, \boldsymbol{q}\rangle - \alpha\delta_{\mathcal{Z}}(\widehat{\boldsymbol{\zeta}} + \boldsymbol{q}/\alpha),
\end{aligned}$$

where the first equality follows from the definition of the dual norm, the second equality exploits (Bertsekas, 2009, Proposition 5.5.4), and the last equality holds because, for any convex closed set, the indicator function is the conjugate of the support function. ■

Proof of Theorem 9 We first prove assertion (i). By Theorem 4(i), the worst-case expectation problem (13) constitutes a restriction of (5) where $\boldsymbol{w}$ is fixed, and thus it coincides with the minimax problem

$$\begin{aligned}
\inf_{\substack{\lambda, s_i \\ \boldsymbol{p}_{ij}, u_{ij}}} \sup_{\alpha_{ij}\geq 0, \gamma_{ij}\geq 0}\ & \lambda\rho + \frac{1}{N}\sum_{i=1}^N s_i + \sum_{i=1}^N\sum_{j=1}^J \gamma_{ij}\big(\|(\boldsymbol{p}_{ij}, u_{ij})\|_* - \lambda\big) \\
& + \sum_{i=1}^N\sum_{j=1}^J \alpha_{ij}\big(S_\Xi(-a_j\boldsymbol{w} - \boldsymbol{p}_{ij}, a_j - u_{ij}) + b_j + \langle \boldsymbol{p}_{ij}, \widehat{\boldsymbol{x}}_i\rangle + u_{ij}\widehat{y}_i - s_i\big).
\end{aligned}$$

The minimization and the maximization may be interchanged by strong duality, which holds because the convex program (5) satisfies Slater's constraint qualification for every fixed

$\boldsymbol{w}$ (Bertsekas, 2009, Proposition 5.3.1). Indeed, note that S_Ξ is proper, convex and lower semi-continuous and appears in constraints that are always satisfiable because they involve a free decision variable. Thus, the above minimax problem is equivalent to

$$\left\{\begin{array}{lll} \sup\limits_{\alpha_{ij},\gamma_{ij}} & \inf\limits_{\boldsymbol{p}_{ij},u_{ij}} & \sum\limits_{i=1}^N\sum\limits_{j=1}^J \alpha_{ij}\big(S_\Xi(a_j\boldsymbol{w}-\boldsymbol{p}_{ij}, a_j-u_{ij})+b_j+\langle \boldsymbol{p}_{ij},\widehat{\boldsymbol{x}}_i\rangle+u_{ij}\widehat{y}_i\big) \\ & & +\sum\limits_{i=1}^N\sum\limits_{j=1}^J \gamma_{ij}\|\boldsymbol{p}_{ij}\|_* \\ \text{s.t.} & \sum\limits_{i=1}^N\sum\limits_{j=1}^J \gamma_{ij}=\rho & \\ & \sum\limits_{j=1}^J \alpha_{ij}=\dfrac{1}{N} & i\in[N] \\ & \alpha_{ij},\gamma_{ij}\geq 0 & i\in[N], j\in[J]. \end{array}\right.$$

By Lemma 48, which applies because $(\widehat{\boldsymbol{x}}_i,\widehat{y}_i)\in\Xi$ for all $i\leq N$, the above dual problem simplifies to

$$\left\{\begin{array}{lll} \sup\limits_{\substack{\alpha_{ij},\gamma_{ij}\\ \boldsymbol{q}_{ij},v_{ij}}} & \sum\limits_{i=1}^N\sum\limits_{j=1}^J \alpha_{ij}\big(a_j(\langle\boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle-\widehat{y}_i)+b_j\big)+a_j(\langle\boldsymbol{w},\boldsymbol{q}_{ij}\rangle-v_{ij}) & \\ \text{s.t.} & \sum\limits_{i=1}^N\sum\limits_{j=1}^J \gamma_{ij}=\rho & \\ & \sum\limits_{j=1}^J \alpha_{ij}=\dfrac{1}{N} & i\in[N] \\ & \|(\boldsymbol{q}_{ij},v_{ij})\|\leq\gamma_{ij} & i\in[N], j\in[J] \\ & (\widehat{\boldsymbol{x}}_i-\boldsymbol{q}_{ij}/\alpha_{ij},\widehat{y}_i-v_{ij}/\alpha_{ij})\in\Xi & i\in[N], j\in[J] \\ & \alpha_{ij},\gamma_{ij}\geq 0 & i\in[N], j\in[J]. \end{array}\right.$$

Problem (14) is now obtained by eliminating the variables γ_{ij} and by substituting α_{ij}, $\boldsymbol{q}_{ij}$, and v_{ij} with α_{ij}/N, $\boldsymbol{q}_{ij}/N$, and v_{ij}/N, respectively.

As for assertion (ii), we first show that the discrete distribution $\mathbb{Q}_\gamma$ belongs to the Wasserstein ball $\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ for all $\gamma\in(0,1]$. Indeed, the Wasserstein distance between $\mathbb{Q}_\gamma$ and $\widehat{\mathbb{P}}_N$ amounts to

$$d(\mathbb{Q}_\gamma,\widehat{\mathbb{P}}_N)\leq\frac{\gamma}{N}\left\|\left(\widehat{\boldsymbol{x}}_1+\frac{\rho N}{\gamma}\boldsymbol{x}^\star,\widehat{y}_1+\frac{\rho N}{\gamma}y^\star\right)-(\widehat{\boldsymbol{x}}_1,\widehat{y}_1)\right\|=\rho\|(\boldsymbol{x}^\star,y^\star)\|\leq\rho,$$

where the first inequality holds because the Wasserstein distance coincides with the optimal mass transportation cost, and the last inequality holds because the norm of $(\boldsymbol{x}^\star,y^\star)$ is at most 1 by construction. Thus, $\mathbb{Q}_\gamma\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ for all $\gamma\in(0,1]$. Denoting the optimal value

of (13) by $J^\star(\boldsymbol{w})$ and using $\ell_{\boldsymbol{w}}(\boldsymbol{x},y)$ as a shorthand for $L(\langle\boldsymbol{w},\boldsymbol{x}\rangle-y)$, we find

$$\begin{aligned} J^\star(\boldsymbol{w}) &\geq \mathbb{E}^{\mathbb{Q}_\gamma}[\ell_{\boldsymbol{w}}(\boldsymbol{x},y)] \\ &= \tfrac{1}{N}\sum_{i=1}^N \ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_i,\widehat{y}_i)-\tfrac{\gamma}{N}\ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_1,\widehat{y}_1)+\tfrac{\gamma}{N}\ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_1+\tfrac{\rho N}{\gamma}\boldsymbol{x}^\star,\widehat{y}_1+\tfrac{\rho N}{\gamma}y^\star) \\ &\geq \tfrac{1}{N}\sum_{i=1}^N \ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_i,\widehat{y}_i)-\tfrac{\gamma}{N}\ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_1,\widehat{y}_1)+\tfrac{\gamma}{N}\big(\langle(\boldsymbol{x},y),(\widehat{\boldsymbol{x}}_1+\tfrac{\rho N}{\gamma}\boldsymbol{x}^\star,\widehat{y}_1+\tfrac{\rho N}{\gamma}y^\star)\rangle \\ &\qquad -\ell^*_{\boldsymbol{w}}(\boldsymbol{x},y)\big) \qquad \forall(\boldsymbol{x},y)\in\mathbb{R}^{n+1}, \end{aligned}$$

where the last estimate follows from Fenchel's inequality. Setting $(\boldsymbol{x},y)=\mathrm{lip}(L)(\boldsymbol{w},-1)$ we thus have

$$\begin{aligned} J^\star(\boldsymbol{w}) &\geq \lim_{\gamma\to 0^+}\frac{1}{N}\sum_{i=1}^N \ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_i,\widehat{y}_i)-\frac{\gamma}{N}\ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_1,\widehat{y}_1)+\frac{\gamma}{N}\,\mathrm{lip}(L)(\langle\boldsymbol{w},\widehat{\boldsymbol{x}}_1\rangle-\widehat{y}_1) \\ &\qquad +\rho\,\mathrm{lip}(L)\|(\boldsymbol{w},-1)\|_*-\frac{\gamma}{N}\ell^*_{\boldsymbol{w}}(\mathrm{lip}(L)(\boldsymbol{w},-1)) \\ &= \frac{1}{N}\sum_{i=1}^N \ell_{\boldsymbol{w}}(\widehat{\boldsymbol{x}}_i,\widehat{y}_i)+\rho\,\mathrm{lip}(L)\|(\boldsymbol{w},-1)\|_*=J^\star(\boldsymbol{w}), \end{aligned}$$

where the equality follows from Theorem 4(ii). The above reasoning implies that

$$\lim_{\gamma\to 0^+}\mathbb{E}^{\mathbb{Q}_\gamma}[\ell_{\boldsymbol{w}}(\boldsymbol{x},y)]=J^\star(\boldsymbol{w}),$$

and thus the claim follows. ■

Proof of Theorem 12 Assume first that the loss function L is convex piecewise linear, that is, $L(z)=\max_{j\in J}\{a_jz+b_j\}$. As $\Xi\in\mathbb{R}^{n+1}$, Theorem 9(i) implies that the worst-case expectation (13) is given by

$$\left\{\begin{array}{lll} \sup\limits_{\alpha_{ij},\boldsymbol{q}_{ij},v_{ij}} & \dfrac{1}{N}\sum\limits_{i=1}^N\sum\limits_{j=1}^J \alpha_{ij}\Big[a_j(\langle\boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle-\widehat{y}_i)+b_j\Big]+a_j(\langle\boldsymbol{w},\boldsymbol{q}_{ij}\rangle-v_{ij}) & \\ \text{s.t.} & \dfrac{1}{N}\sum\limits_{i=1}^N\sum\limits_{j=1}^J \|(\boldsymbol{q}_{ij},v_{ij})\|\leq\rho & \\ & \sum\limits_{j=1}^J \alpha_{ij}=1 & i\in[N] \\ & \alpha_{ij}\geq 0 & i\in[N], j\in[J] \end{array}\right.$$

$$\geq\left\{\begin{array}{lll} \sup\limits_{\alpha_{ij},\Delta\boldsymbol{x}_{ij},\Delta y_{ij}} & \dfrac{1}{N}\sum\limits_{i=1}^N\sum\limits_{j=1}^J \alpha_{ij}\left[a_j(\langle\boldsymbol{w},\widehat{\boldsymbol{x}}_i+\Delta\boldsymbol{x}_{ij}\rangle-\widehat{y}_i-\Delta y_{ij})+b_j\right] & \\ \text{s.t.} & \dfrac{1}{N}\sum\limits_{i=1}^N\sum\limits_{j=1}^J \alpha_{ij}\|(\Delta\boldsymbol{x}_{ij},\Delta y_{ij})\|\leq\rho & \\ & \sum\limits_{j=1}^J \alpha_{ij}=1 & i\in[N] \\ & \alpha_{ij}\geq 0 & i\in[N], j\in[J] \end{array}\right.$$

$$\geq \left\{ \begin{array}{lll} \sup\limits_{\alpha_{ij},\Delta\boldsymbol{x}_i,\Delta y_i} & \frac{1}{N}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{J}\alpha_{ij}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i+\Delta\boldsymbol{x}_i\rangle-\widehat{y}_i-\Delta y_i)+b_j\right] & \\ \text{s.t.} & \frac{1}{N}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{J}\alpha_{ij}\|(\Delta\boldsymbol{x}_i,\Delta y_i)\|\leq\rho & \\ & \sum\limits_{j=1}^{J}\alpha_{ij}=1 & i\in[N] \\ & \alpha_{ij}\geq 0 & i\in[N],j\in[J]. \end{array}\right.$$

The first inequality holds because for any feasible solution $\{\alpha_{ij},\Delta\boldsymbol{x}_{ij},\Delta y_{ij}\}$ to the second optimization problem, the solution $\{\alpha_{ij},\boldsymbol{q}_{ij},v_{ij}\}$ with $\boldsymbol{q}_{ij}=\alpha_{ij}\Delta\boldsymbol{x}_{ij}$ and $v_{ij}=\alpha_{ij}\Delta y_{ij}$ is feasible in the first problem and attains the same objective value (conversely, note that the first problem admits feasible solutions with $\alpha_{ij}=0$ and $\boldsymbol{q}_{ij}\neq\boldsymbol{0}$ that have no counterpart in the second problem). The second inequality follows from the restriction that $\Delta\boldsymbol{x}_{ij}$ and Δy_{ij} must be independent of j. It is easy to verify that the last optimization problem in the above expression is equivalent to (15) because $(\alpha_{i1},\ldots,\alpha_{iJ})$ ranges over a simplex for every $i\leq N$, and thus (15) provides a lower bound on (13).

Suppose now that Assumption 10 holds, and note that the worst-case loss (15) can be expressed as

$$\left\{ \begin{array}{ll} \sup\limits_{\Delta\boldsymbol{x}_i,\Delta y_i} & \frac{1}{N}\sum\limits_{i=1}^{N}\max\limits_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i+\Delta\boldsymbol{x}_i\rangle-\widehat{y}_i-\Delta y_i)+b_j\right] \\ \text{s.t.} & \frac{1}{N}\sum\limits_{i=1}^{N}\|(\Delta\boldsymbol{x}_i,\Delta y_i)\|\leq\rho \end{array}\right.$$

$$\geq \left\{ \begin{array}{ll} \sup\limits_{\Delta\boldsymbol{x},\Delta y} & \frac{1}{N}\sum\limits_{i\neq k}\max\limits_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle-\widehat{y}_i)+b_j\right]+\frac{1}{N}\max\limits_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_k+\Delta\boldsymbol{x}\rangle-\widehat{y}_k-\Delta y)+b_j\right] \\ \text{s.t.} & \frac{1}{N}\|(\Delta\boldsymbol{x},\Delta y)\|\leq\rho. \end{array}\right.$$

The above inequality follows from setting $\Delta\boldsymbol{x}_i=0$ and $\Delta y_i=0$ for all $i\neq k$, where $(\widehat{\boldsymbol{x}}_k,\widehat{y}_k)$ is a training sample satisfying $|L'(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_k\rangle-\widehat{y}_k)|=\mathrm{lip}(L)$, which exists due to Assumption 10. The last expression equals

$$=\frac{1}{N}\sum_{i\neq k}\max_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle-\widehat{y}_i)+b_j\right]+\max_{j\leq J}\left\{ \begin{array}{ll} \sup\limits_{\Delta\boldsymbol{x},\Delta y} & \frac{1}{N}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_k\rangle-\widehat{y}_k)+b_j\right] \\ & +\frac{1}{N}\left[a_j(\langle \boldsymbol{w},\Delta\boldsymbol{x}\rangle-\Delta y)\right] \\ \text{s.t.} & \frac{1}{N}\|(\Delta\boldsymbol{x},\Delta y)\|\leq\rho \end{array}\right.$$

$$=\frac{1}{N}\sum_{i\neq k}\max_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle-\widehat{y}_i)+b_j\right]+\frac{1}{N}\max_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_k\rangle-\widehat{y}_k)+b_j+\rho N\|(\boldsymbol{w},-1)\|_*|a_j|\right]$$

$$=\frac{1}{N}\sum_{i\neq k}\max_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle-\widehat{y}_i)+b_j\right]+\frac{1}{N}\max_{j\leq J}\left[a_j(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_k\rangle-\widehat{y}_k)+b_j\right]+\max_{j\leq J}\rho\|(\boldsymbol{w},-1)\|_*|a_j|$$

$$=\frac{1}{N}\sum_{i=1}^{N}L(\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle-\widehat{y}_i)+\max_{j\leq J}\rho\|(\boldsymbol{w},-1)\|_*|a_j|,$$

where the penultimate equality holds because $\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_k\rangle-\widehat{y}_k$ resides on the steepest linear piece of the loss function L by virtue of Assumption 10. The claim then follows from Theorem 4(ii) because $\mathrm{lip}(L)=\max_{j\leq J}|a_j|$. Note that generic convex Lipschitz continuous loss functions can be uniformly approximated as closely as desired with convex piecewise linear functions. Thus, the above argument extends directly to generic convex Lipschitz continuous loss functions. Details are omitted for brevity. ■

Proof of Theorem 14 To prove assertion (i), we apply Lemma 45 to problem (4) with the transportation distance (16), where $\boldsymbol{\xi}=(\boldsymbol{x},y)$ and $\Xi=\mathbb{X}\times\{-1,+1\}$. Thus, we obtain

$$\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)}\mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)\right]$$

$$=\left\{ \begin{array}{lll} \inf\limits_{\lambda,s_i} & \lambda\rho+\frac{1}{N}\sum\limits_{i=1}^{N}s_i & \\ \text{s.t.} & \sup\limits_{(\boldsymbol{x},y)\in\Xi}L(y\langle \boldsymbol{w},\boldsymbol{x}\rangle)-\lambda\big(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|+\frac{\kappa}{2}|y-\widehat{y}_i|\big)\leq s_i & i\in[N] \\ & \lambda\geq 0. & \end{array}\right.$$

$$=\left\{ \begin{array}{lll} \inf\limits_{\lambda,s_i} & \lambda\rho+\frac{1}{N}\sum\limits_{i=1}^{N}s_i & \\ \text{s.t.} & \sup\limits_{\boldsymbol{x}\in\mathbb{X}}L(\widehat{y}_i\langle \boldsymbol{w},\boldsymbol{x}\rangle)-\lambda\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|\leq s_i & i\in[N] \\ & \sup\limits_{\boldsymbol{x}\in\mathbb{X}}L(-\widehat{y}_i\langle \boldsymbol{w},\boldsymbol{x}\rangle)-\lambda\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|-\kappa\lambda\leq s_i & i\in[N] \\ & \lambda\geq 0 & \end{array}\right.$$

$$=\left\{ \begin{array}{lll} \inf\limits_{\lambda,s_i} & \lambda\rho+\frac{1}{N}\sum\limits_{i=1}^{N}s_i & \\ \text{s.t.} & \sup\limits_{\boldsymbol{x}\in\mathbb{X}}\ a_j\widehat{y}_i\langle \boldsymbol{w},\boldsymbol{x}\rangle+b_j-\lambda\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|\leq s_i & i\in[N],j\in[J] \\ & \sup\limits_{\boldsymbol{x}\in\mathbb{X}}\ -a_j\widehat{y}_i\langle \boldsymbol{w},\boldsymbol{x}\rangle+b_j-\lambda\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|-\kappa\lambda\leq s_i & i\in[N],j\in[J] \\ & \lambda\geq 0, & \end{array}\right.$$

where the second equality holds because, for every i, y can be either equal to $\widehat{y}_i$ or to $-\widehat{y}_i$. Reformulating the constraints using Lemma 46 and including $\boldsymbol{w}$ as a decision variable then yields (17).

When $\mathbb{X} = \mathbb{R}^n$ and L is Lipschitz continuous, we can use similar arguments as above to prove that

$$\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle, y)\right] = \left\{\begin{array}{lll} \inf\limits_{\lambda,s_i} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & \sup\limits_{\boldsymbol{x}\in\mathbb{R}^n} \; L(\widehat{y}_i\langle \boldsymbol{w},\boldsymbol{x}\rangle) - \lambda\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| \le s_i & i\in[N] \\ & \sup\limits_{\boldsymbol{x}\in\mathbb{R}^n} \; L(-\widehat{y}_i\langle \boldsymbol{w},\boldsymbol{x}\rangle) - \lambda\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| - \kappa\lambda \le s_i & i\in[N] \\ & \lambda \ge 0. & \end{array}\right.$$

Applying Lemma 47 to the subordinate maximization problems in the constraints yields

$$\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle, y)\right] = \left\{\begin{array}{lll} \inf\limits_{\lambda,s_i} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & L(\widehat{y}_i\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle) \le s_i & i\in[N], j\in[J] \\ & L(-\widehat{y}_i\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle) - \kappa\lambda \le s_i & i\in[N], j\in[J] \\ & \sup\limits_{\theta\in\Theta} \; |\theta|\cdot\|\boldsymbol{w}\|_* \le \lambda. & \end{array}\right.$$

Thus, assertion (ii) follows by recalling that $\mathrm{lip}(L) = \sup_\theta\{|\theta| : L^*(\theta) < \infty\} = \sup_{\theta\in\Theta}|\theta|$ and by including $\boldsymbol{w}$ as a decision variable. ■

Proof of Corollary 15 Notice that the hinge loss function is piecewise linear with $J = 2$ pieces, see Section 2.1. Moreover, by strong conic duality the support function of $\mathbb{X}$ can be re-expressed as

$$S_{\mathbb{X}}(\boldsymbol{z}) = \sup_{\boldsymbol{x}}\left\{\langle \boldsymbol{z},\boldsymbol{x}\rangle : \boldsymbol{C}\boldsymbol{x} \preceq_{\mathcal{C}} \boldsymbol{d}\right\} = \inf_{\boldsymbol{q}\in\mathcal{C}^*}\left\{\langle \boldsymbol{q},\boldsymbol{d}\rangle : \boldsymbol{C}^\top\boldsymbol{q} = \boldsymbol{z}\right\}.$$

Strong duality holds because $\mathbb{X}$ admits a Slater point. The proof thus follows from Theorem 14(i). ■

Proof of Corollary 16 The smooth hinge loss $L(z)$ coincides with the inf-convolution of $\frac{1}{2}z^2$ and $\max\{0, 1-z\}$ and can thus be expressed as $L(z) = \min_{z_1}\frac{1}{2}z_1^2 + \max\{0, 1-z-z_1\}$. Moreover, the Lipschitz modulus of the smooth hinge loss function is 1. The proof thus follows from Theorem 14(ii). ■

Proof of Corollary 17 Notice that the logloss function is convex and has Lipschitz modulus 1, see Section 2.1. The rest of proof follows from Theorem 14(ii). For more details see (Shafieezadeh-Abadeh et al., 2015). ■

Proof of Theorem 20 We first prove assertion (i). By Theorem 14(i), the worst-case expectation problem (25) constitutes a restriction of (17) where $\boldsymbol{w}$ is fixed, and thus it is

equivalent to the minimax problem

$$\begin{aligned} \inf_{\substack{\lambda,s_i\\ \boldsymbol{p}_{ij}^+,\boldsymbol{p}_{ij}^-}} \sup_{\substack{\alpha_{ij}^+\ge0,\gamma_{ij}^+\ge0\\ \alpha_{ij}^-\ge0,\gamma_{ij}^-\ge0}} \; & \lambda\rho + \frac{1}{N}\sum_{i=1}^N s_i + \sum_{i=1}^N\sum_{j=1}^J \alpha_{ij}^+\big(S_{\mathbb{X}}(a_j\widehat{y}_i\boldsymbol{w} - \boldsymbol{p}_{ij}^+) + b_j + \langle \boldsymbol{p}_{ij}^+, \widehat{\boldsymbol{x}}_i\rangle - s_i\big) \\ & + \sum_{i=1}^N\sum_{j=1}^J \alpha_{ij}^-\big(S_{\mathbb{X}}(-a_j\widehat{y}_i\boldsymbol{w} - \boldsymbol{p}_{ij}^-) + b_j + \langle \boldsymbol{p}_{ij}^-, \widehat{\boldsymbol{x}}_i\rangle - \kappa\lambda - s_i\big) \\ & + \sum_{i=1}^N\sum_{j=1}^J \gamma_{ij}^+\big(\|\boldsymbol{p}_{ij}^+\|_* - \lambda\big) + \sum_{i=1}^N\sum_{j=1}^J \gamma_{ij}^-\big(\|\boldsymbol{p}_{ij}^-\|_* - \lambda\big). \end{aligned}$$

The minimization and the maximization may be interchanged by strong Lagrangian duality, which holds because the convex program (17) satisfies Slater's constraint qualification for any fixed $\boldsymbol{w}$ (Bertsekas, 2009, Proposition 5.3.1). Indeed, note that $S_{\mathbb{X}}$ is proper, convex and lower semi-continuous and appears in constraints that are always satisfiable because they involve a free decision variable. Thus, problem (25) is equivalent to

$$\left\{\begin{array}{lll} \sup\limits_{\substack{\alpha_{ij}^+,\gamma_{ij}^+\\ \alpha_{ij}^-,\gamma_{ij}^-}} & \inf\limits_{\boldsymbol{p}_{ij}^+,\boldsymbol{p}_{ij}^-} \displaystyle\sum_{i=1}^N\sum_{j=1}^J \alpha_{ij}^+\big(S_{\mathbb{X}}(a_j\widehat{y}_i\boldsymbol{w} - \boldsymbol{p}_{ij}^+) + b_j + \langle \boldsymbol{p}_{ij}^+, \widehat{\boldsymbol{x}}_i\rangle\big) + \gamma_{ij}^+\|\boldsymbol{p}_{ij}^+\|_* & \\ & \quad + \displaystyle\sum_{i=1}^N\sum_{j=1}^J \alpha_{ij}^-\big(S_{\mathbb{X}}(-a_j\widehat{y}_i\boldsymbol{w} - \boldsymbol{p}_{ij}^-) + b_j + \langle \boldsymbol{p}_{ij}^-, \widehat{\boldsymbol{x}}_i\rangle\big) + \gamma_{ij}^-\|\boldsymbol{p}_{ij}^-\|_* & \\ \text{s.t.} & \displaystyle\sum_{i=1}^N\sum_{j=1}^J \gamma_{ij}^+ + \gamma_{ij}^- + \kappa\alpha_{ij}^- = \rho & \\ & \displaystyle\sum_{j=1}^J \alpha_{ij}^+ + \alpha_{ij}^- = \frac{1}{N} & i\in[N] \\ & \alpha_{ij}^+, \alpha_{ij}^-, \gamma_{ij}^+, \gamma_{ij}^- \ge 0 & i\in[N], j\in[J]. \end{array}\right.$$

By Lemma 48, the above dual problem simplifies to

$$\left\{\begin{array}{lll} \sup\limits_{\substack{\alpha_{ij}^+,\gamma_{ij}^+,\boldsymbol{q}_{ij}^+\\ \alpha_{ij}^-,\gamma_{ij}^-,\boldsymbol{q}_{ij}^-}} & \displaystyle\sum_{i=1}^N\sum_{j=1}^J (\alpha_{ij}^+ - \alpha_{ij}^-)a_j\widehat{y}_i\langle \boldsymbol{w},\widehat{\boldsymbol{x}}_i\rangle + a_j\widehat{y}_i\langle \boldsymbol{w}, \boldsymbol{q}_{ij}^+ - \boldsymbol{q}_{ij}^-\rangle + \sum_{j=1}^J b_j & \\ \text{s.t.} & \displaystyle\sum_{i=1}^N\sum_{j=1}^J \gamma_{ij}^+ + \gamma_{ij}^- + \kappa\alpha_{ij}^- = \rho & \\ & \displaystyle\sum_{j=1}^J \alpha_{ij}^+ + \alpha_{ij}^- = \frac{1}{N} & i\in[N] \\ & \|\boldsymbol{q}_{ij}^+\| \le \gamma_{ij}^+, \quad \|\boldsymbol{q}_{ij}^-\| \le \gamma_{ij}^- & i\in[N], j\in[J] \\ & \widehat{\boldsymbol{x}}_i + \boldsymbol{q}_{ij}^+/\alpha_{ij}^+ \in \mathbb{X}, \quad \widehat{\boldsymbol{x}}_i + \boldsymbol{q}_{ij}^-/\alpha_{ij}^- \in \mathbb{X} & i\in[N], j\in[J] \\ & \alpha_{ij}^+, \alpha_{ij}^-, \gamma_{ij}^+, \gamma_{ij}^- \ge 0 & i\in[N], j\in[J]. \end{array}\right.$$

Problem (26) is now obtained by eliminating γ_{ij}^+ and γ_{ij}^- and by substituting α_{ij}^+, α_{ij}^-, $\boldsymbol{q}_{ij}^+$, and $\boldsymbol{q}_{ij}^-$ with α_{ij}^+/N, α_{ij}^-/N, $\boldsymbol{q}_{ij}^+/N$, and $\boldsymbol{q}_{ij}^-/N$, respectively.

As for assertion (ii), we use L_i^+ and L_i^- to abbreviate $L(\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle)$ and $L(-\widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle)$, respectively. By Theorem 14(ii), we have

$$\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}[L(y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)] = \left\{\begin{array}{lll} \inf\limits_{\boldsymbol{w},\lambda,s_i} & \lambda\rho + \dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & L_i^+ \le s_i & i\in[N] \\ & L_i^- - \kappa\lambda \le s_i & i\in[N] \\ & \operatorname{lip}(L)\|(\boldsymbol{w})\|_* \le \lambda & \end{array}\right.$$

$$= \left\{\begin{array}{lll} \sup\limits_{\alpha_i^+,\alpha_i^-,\theta} & \operatorname{lip}(L)\|(\boldsymbol{w})\|_*\theta + \displaystyle\sum_{i=1}^N \alpha_i^+ L_i^+ + \alpha_i^- L_i^- & \\ \text{s.t.} & \alpha_i^+ + \alpha_i^- = \frac{1}{N} & i\in[N] \\ & \theta + \kappa\displaystyle\sum_{i=1}^N \alpha_i^- = \rho & \\ & \alpha_i^+ \ge 0,\ \alpha_i^- \ge 0 & i\in[N] \\ & \theta \ge 0, & \end{array}\right.$$

where the second equality follows from strong linear programming duality, which holds because the primal problem is manifestly feasible. Eliminating the first constraint and replacing α_i^- with α_i/N and α_i^+ with $(1-\alpha_i)/N$ allows us to reformulate the dual linear program as

$$\left\{\begin{array}{lll} \sup\limits_{\alpha_i,\theta} & \operatorname{lip}(L)(\boldsymbol{w})\|_*\theta + \dfrac{1}{N}\displaystyle\sum_{i=1}^N (1-\alpha_i)L_i^+ + \alpha_i L_i^- & \\ \text{s.t.} & \theta + \dfrac{\kappa}{N}\displaystyle\sum_{i=1}^N \alpha_i = \rho & \\ & 0 \le \alpha_i \le 1 & i\in[N] \\ & \theta \ge 0. & \end{array}\right.$$

Thus, the worst-case expectation (25) coincides with the optimal value of (27) for $\gamma = 0$. Next let $(\alpha_i^\star(\gamma), \theta^\star(\gamma))$ be an optimal solution of (27) for $\gamma \ge 0$, and define $\mathbb{Q}_\gamma$ as in the theorem statement. Note that (27) is infeasible for $\gamma > \rho$. Moreover, note that the atoms of $\mathbb{Q}_\gamma$ have non-negative probabilities if $\eta(\gamma) \in [0,1]$, which holds whenever $\gamma \in [0,1]$. We thus focus on parameter values $\gamma \in [0, \min\{\rho, 1\}]$. By construction, the Wasserstein distance between $\mathbb{Q}_\gamma$ and the empirical distribution satisfies

$$\begin{aligned} d(\mathbb{Q}_\gamma, \widehat{\mathbb{P}}_N) &\le \tfrac{\kappa}{N}\sum_{i=1}^N \alpha_i^\star(\gamma) - \tfrac{\eta(\gamma)\kappa}{N}\alpha_1^\star(\gamma) + \tfrac{\eta(\gamma)}{N} d\big((\widehat{\boldsymbol{x}}_1 + \tfrac{\theta^\star(\gamma)N}{\eta(\gamma)}\boldsymbol{x}^\star, \widehat{y}_1) - (\widehat{\boldsymbol{x}}_1, \widehat{y}_1)\big) \\ &\le \rho - \gamma - \theta^\star(\gamma) + \theta^\star(\gamma)\|\boldsymbol{x}^\star\| \le \rho, \end{aligned}$$

where the first inequality holds because the Wasserstein distance is defined as the minimum cost of moving $\mathbb{Q}_\gamma$ to $\widehat{\mathbb{P}}_N$, the second inequality follows from the feasibility of $(\alpha_i^\star(\gamma), \theta^\star(\gamma))$ in (27) and the non-negativity of $\eta(\gamma)$, and the last inequality holds because $\|\boldsymbol{x}^\star\| \le 1$,

$\theta^\star(\gamma) \ge 0$ and $\gamma \ge 0$. Thus, $\mathbb{Q}_\gamma \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)$ for all $\gamma \in [0, \min\{\rho, 1\}]$. Denoting the optimal value of (25) by $J^\star(\boldsymbol{w})$, we find

$$\begin{aligned} J^\star(\boldsymbol{w}) &\ge \mathbb{E}^{\mathbb{Q}_\gamma}[L(y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)] \\ &= \tfrac{1}{N}\big(\sum_{i=1}^N (1-\alpha_i^\star(\gamma))L_i^+ + \alpha_i^\star(\gamma)L_i^-\big) - \tfrac{\eta(\gamma)}{N}\big((1-\alpha_1^\star(\gamma))L_1^+ + \alpha_1^\star(\gamma)L_1^-\big) \\ &\quad + \tfrac{\eta(\gamma)}{N} L\big(\widehat{y}_1\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_1 + \tfrac{\theta^\star(\gamma)N}{\eta(\gamma)}\boldsymbol{x}^\star\rangle\big) \\ &\ge \tfrac{1}{N}\big(\sum_{i=1}^N (1-\alpha_i^\star(\gamma))L_i^+ + \alpha_i^\star(\gamma)L_i^-\big) - \tfrac{\eta(\gamma)}{N}\big((1-\alpha_1^\star(\gamma))L_1^+ + \alpha_1^\star(\gamma)L_1^-\big) \\ &\quad + \tfrac{\eta(\gamma)}{N}\Big(\langle \boldsymbol{x}, \widehat{\boldsymbol{x}}_1 + \tfrac{\theta^*(\gamma)N}{\eta(\gamma)}\boldsymbol{x}^\star\rangle - L^*(\widehat{y}_1\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\Big) \qquad \forall \boldsymbol{x} \in \mathbb{R}^n, \end{aligned}$$

where the last estimate follows from Fenchel's inequality. Setting $\boldsymbol{x} = \operatorname{lip}(L)\boldsymbol{w}$ and driving γ to zero yields

$$\begin{aligned} J^\star(\boldsymbol{w}) &\ge \lim_{\gamma\to 0^+} \tfrac{1}{N}\Big(\sum_{i=1}^N (1-\alpha_i^\star(\gamma))L_i^+ + \alpha_i^\star(\gamma)L_i^-\Big) - \tfrac{\eta(\gamma)}{N}\big((1-\alpha_1^\star(\gamma))L_1^+ + \alpha_1^\star(\gamma)L_1^-\big) \\ &\quad + \tfrac{\eta(\gamma)}{N}\big(\operatorname{lip}(L)\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_1\rangle - L^*(\widehat{y}_1 \operatorname{lip}(L)\langle \boldsymbol{w}, \boldsymbol{w}\rangle)\big) + \operatorname{lip}(L)\|\boldsymbol{w}\|_* \cdot \theta^\star(\gamma) \\ &= \lim_{\gamma\to 0^+} \tfrac{1}{N}\big(\sum_{i=1}^N (1-\alpha_i^\star(\gamma))L_i^+ + \alpha_i^\star(\gamma)L_i^-\big) + \operatorname{lip}(L)\|\boldsymbol{w}\|_* \cdot \theta^\star(\gamma) = J^\star(\boldsymbol{w}), \end{aligned}$$

where the first equality follows from the observation that $\eta(\gamma) \in [0, \gamma]$, which implies that $\eta(\gamma)$ converges to zero as γ tends to zero. The second equality holds because the optimal value of (27) is concave and non-increasing and—*a fortiori*—continuous in $\gamma \in [0, \min\{\rho, 1\}]$ and because $J^\star(\boldsymbol{w})$ coincides with the optimal value of (27) when $\gamma = 0$. The above reasoning implies that $\lim_{\gamma\to 0^+} \mathbb{E}^{\mathbb{Q}_\gamma}[L(y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)] = J^\star(\boldsymbol{w})$, and thus the claim follows. ■

Proof of Theorem 23 Assume first that the loss function L is convex piecewise linear, that is, $L(z) = \max_{j\in J}\{a_j z + b_j\}$. As $\mathbb{X} = \mathbb{R}^n$ and $\kappa = \infty$, Theorem 20(i) implies that (25) can be expressed as

$$\left\{\begin{array}{lll} \sup\limits_{\alpha_{ij}^+, \boldsymbol{q}_{ij}^+} & \dfrac{1}{N}\displaystyle\sum_{i=1}^N\sum_{j=1}^J \alpha_{ij}^+ a_j \widehat{y}_i\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle + a_j\widehat{y}_i\langle \boldsymbol{w}, \boldsymbol{q}_{ij}^+\rangle + \displaystyle\sum_{j=1}^J b_j & \\ \text{s.t.} & \displaystyle\sum_{i=1}^N\sum_{j=1}^J \|\boldsymbol{q}_{ij}^+\| \le N\rho & \\ & \displaystyle\sum_{j=1}^J \alpha_{ij}^+ = 1 & i\in[N] \\ & \alpha_{ij}^+ \ge 0 & i\in[N], j\in[J] \end{array}\right.$$

$$\geq \left\{ \begin{array}{lll} \sup\limits_{\alpha_{ij}, \Delta \boldsymbol{x}_{ij}} & \frac{1}{N} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{J} \alpha_{ij} a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i + \Delta \boldsymbol{x}_{ij} \rangle + \sum\limits_{j=1}^{J} b_j & \\ \text{s.t.} & \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{J} \alpha_{ij} \|\Delta \boldsymbol{x}_{ij}\| \leq N\rho & \\ & \sum\limits_{j=1}^{J} \alpha_{ij} = 1 & i \in [N] \\ & \alpha_{ij} \geq 0 & i \in [N], j \in [J] \end{array} \right.$$

$$\geq \left\{ \begin{array}{lll} \sup\limits_{\alpha_{ij}, \Delta \boldsymbol{x}_{i}} & \frac{1}{N} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{J} \alpha_{ij} a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i + \Delta \boldsymbol{x}_{i} \rangle + \sum\limits_{j=1}^{J} b_j & \\ \text{s.t.} & \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{J} \alpha_{ij} \|\Delta \boldsymbol{x}_{i}\| \leq N\rho & \\ & \sum\limits_{j=1}^{J} \alpha_{ij} = 1 & i \in [N] \\ & \alpha_{ij} \geq 0 & i \in [N], j \in [J]. \end{array} \right.$$

The first optimization problem constitutes a special case of (26). Indeed, as $\kappa = \infty$, the first constraint in (26) implies that $\alpha_{ij}^- = 0$, which in turn implies via the fourth constraint and our conventions of extended arithmetics that $\boldsymbol{q}_{ij}^- = \boldsymbol{0}$. The first inequality in the above expression holds because for any feasible solution $\{\alpha_{ij}, \Delta \boldsymbol{x}_{ij}\}$ to the second problem, the solution $\{\alpha_{ij}^+, \boldsymbol{q}_{ij}^+\}$ with $\boldsymbol{q}_{ij}^+ = \alpha_{ij}^+ \Delta \boldsymbol{x}_{ij}$ and $\alpha_{ij} = \alpha_{ij}^+$ for all $i \leq N$ and $j \leq J$ is feasible in the first problem and attains the same objective value. The second inequality in the above expression follows from the restriction that $\Delta \boldsymbol{x}_{ij}$ must be independent of j. It is easy to verify that the last optimization problem is equivalent to (28) because $(\alpha_{i1}, \ldots, \alpha_{iJ})$ ranges over a simplex for every $i \leq N$, and thus (28) provides a lower bound on (25).

Suppose now that Assumption 21 holds, and note that (28) can be expressed as

$$\left\{ \begin{array}{ll} \sup\limits_{\Delta \boldsymbol{x}_i, \Delta y_i} & \frac{1}{N} \sum\limits_{i=1}^{N} \max\limits_{j \leq J} \left[a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i + \Delta \boldsymbol{x}_i \rangle + b_j \right] \\ \text{s.t.} & \frac{1}{N} \sum\limits_{i=1}^{N} \|\Delta \boldsymbol{x}_i\| \leq \rho \end{array} \right.$$

$$\geq \left\{ \begin{array}{ll} \sup\limits_{\Delta \boldsymbol{x}, \Delta y} & \frac{1}{N} \sum\limits_{i \neq k} \max\limits_{j \leq J} \left[a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle + b_j \right] + \frac{1}{N} \max\limits_{j \leq J} \left[a_j \widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k + \Delta \boldsymbol{x} \rangle + b_j \right] \\ \text{s.t.} & \frac{1}{N} \|\Delta \boldsymbol{x}\| \leq \rho. \end{array} \right.$$

The above inequality follows from setting $\Delta \boldsymbol{x}_i = 0$ and $\Delta y_i = 0$ for all $i \neq k$, where $(\widehat{\boldsymbol{x}}_k, \widehat{y}_k)$ is a training sample satisfying $|L'(\widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle)| = \operatorname{lip}(L)$, which exists due to Assumption 21. The last expression equals

$$\frac{1}{N} \sum_{i \neq k} \max_{j \leq J} \left[a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle + b_j \right] + \max_{j \leq J} \left\{ \begin{array}{ll} \sup\limits_{\Delta \boldsymbol{x}, \Delta y} & \frac{1}{N} \left[a_j \widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle + b_j \right] + \frac{1}{N} \left[a_j \widehat{y}_k \langle \boldsymbol{w}, \Delta \boldsymbol{x} \rangle \right] \\ \text{s.t.} & \frac{1}{N} \|(\Delta \boldsymbol{x}, \Delta y)\| \leq \rho \end{array} \right.$$

$$\begin{aligned} &= \frac{1}{N} \sum_{i \neq k} \max_{j \leq J} \left[a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle + b_j \right] + \frac{1}{N} \max_{j \leq J} \left[a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle + b_j + \rho N \|\boldsymbol{w}\|_* |a_j| \right] \\ &= \frac{1}{N} \sum_{i \neq k} \max_{j \leq J} \left[a_j \widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle + b_j \right] + \frac{1}{N} \max_{j \leq J} \left[a_j \widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle + b_j \right] + \max_{j \leq J} \rho \|\boldsymbol{w}\|_* |a_j| \\ &= \frac{1}{N} \sum_{i=1}^{N} L(\widehat{y}_i \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i \rangle) + \max_{j \leq J} \rho \|\boldsymbol{w}\|_* |a_j|, \end{aligned}$$

where the penultimate equality holds because $\widehat{y}_k \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_k \rangle$ resides on the steepest linear piece of the loss function L by virtue of Assumption 21. The claim then follows from Theorem 14(ii) because $\operatorname{lip}(L) = \max_{j \leq J} |a_j|$. Note that generic convex Lipschitz continuous loss functions can be uniformly approximated as closely as desired with convex piecewise linear functions. Thus, the above arguments extend directly to generic convex Lipschitz continuous loss functions. Details are omitted for brevity. ■

Proof of Theorem 26 By the definition of the feature map Φ corresponding to the kernel k, we have

$$\begin{aligned} \|\Phi(\boldsymbol{x}_1) - \Phi(\boldsymbol{x}_2)\|_{\mathbb{H}} &= \sqrt{\langle \Phi(\boldsymbol{x}_1), \Phi(\boldsymbol{x}_1) \rangle_{\mathbb{H}} - 2 \langle \Phi(\boldsymbol{x}_1), \Phi(\boldsymbol{x}_2) \rangle_{\mathbb{H}} + \langle \Phi(\boldsymbol{x}_2), \Phi(\boldsymbol{x}_2) \rangle_{\mathbb{H}}} \\ &= \sqrt{k(\boldsymbol{x}_1, \boldsymbol{x}_1) - 2k(\boldsymbol{x}_1, \boldsymbol{x}_2) + k(\boldsymbol{x}_2, \boldsymbol{x}_2)} \;\leq\; g(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2) \end{aligned} \tag{A.3}$$

for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{X}$, where the inequality follows from Assumption 25. As for assertion (i), we may use similar argument as in the proof of Lemma 45 to reformulate the worst-case expectation in (29) as

$$\sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}} \left[\ell(h(\boldsymbol{x}), y) \right] \tag{A.4}$$

$$= \left\{ \begin{array}{ll} \sup\limits_{\mathbb{Q}^i} & \frac{1}{N} \sum\limits_{i=1}^{N} \int_{\mathbb{X} \times \mathbb{Y}} \ell(h(\boldsymbol{x}), y) \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) \\ \text{s.t.} & \frac{1}{N} \sum\limits_{i=1}^{N} \int_{\mathbb{X} \times \mathbb{Y}} d\big((\boldsymbol{x}, y), (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\big) \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) \leq \rho \\ & \int_{\mathbb{X} \times \mathbb{Y}} \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) = 1 \qquad i \in [N] \end{array} \right.$$

$$= \left\{ \begin{array}{ll} \sup\limits_{\mathbb{Q}^i} & \frac{1}{N} \sum\limits_{i=1}^{N} \int_{\mathbb{X} \times \mathbb{Y}} \ell(h(\boldsymbol{x}), y) \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) \\ \text{s.t.} & g\left(\frac{1}{N} \sum\limits_{i=1}^{N} \int_{\mathbb{X} \times \mathbb{Y}} \sqrt{2} d\big((\boldsymbol{x}, y), (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\big) \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) \right) \leq g(\sqrt{2}\rho) \\ & \int_{\mathbb{X} \times \mathbb{Y}} \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) = 1 \qquad i \in [N] \end{array} \right.$$

$$\leq \left\{ \begin{array}{ll} \sup\limits_{\mathbb{Q}^i} & \frac{1}{N} \sum\limits_{i=1}^{N} \int_{\mathbb{X} \times \mathbb{Y}} \ell(h(\boldsymbol{x}), y) \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) \\ \text{s.t.} & \frac{1}{N} \sum\limits_{i=1}^{N} \int_{\mathbb{X} \times \mathbb{Y}} g\left(\sqrt{2} d\big((\boldsymbol{x}, y), (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\big) \right) \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) \leq g(\sqrt{2}\rho) \\ & \int_{\mathbb{X} \times \mathbb{Y}} \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}, \mathrm{d}y) = 1 \qquad i \in [N], \end{array} \right. \tag{A.5}$$

where the second equality holds because g is strictly monotonically increasing, and the inequality follows from Jensen inequality, which applies because g is concave. By the definition of the transportation metric on $\mathbb{H} \times \mathbb{Y}$ for regression problems, we then find

$$\begin{aligned} g\Big(\sqrt{2}d\big((\boldsymbol{x},y),(\widehat{\boldsymbol{x}}_i,\widehat{y}_i)\big)\Big) &= g\Big(\sqrt{2\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|_2^2+2(y-\widehat{y}_i)^2}\Big) \\ &\geq g(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|_2+|y-\widehat{y}_i|) \\ &\geq g(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|_2)+|y-\widehat{y}_i| \\ &\geq \|\Phi(\boldsymbol{x})-\Phi(\widehat{\boldsymbol{x}}_i)\|_{\mathbb{H}}+|y-\widehat{y}_i| \\ &\geq \sqrt{\|\Phi(\boldsymbol{x})-\Phi(\widehat{\boldsymbol{x}}_i)\|_{\mathbb{H}}^2+(y-\widehat{y}_i)^2} \\ &= d_{\mathbb{H}}\big((\Phi(\boldsymbol{x}),y),(\Phi(\widehat{\boldsymbol{x}}_i),\widehat{y}_i)\big), \end{aligned} \tag{A.6}$$

where the first inequality holds because $2a^2+2b^2 \geq (a+b)^2$ for all $a,b \geq 0$ and because g is strictly monotonically increasing, the second inequality exploits the assumption that $g'(z) \geq 1$ for all $z \geq 0$, the third inequality follows form (A.3), and the last equality holds because $a^2+b^2 \leq (a+b)^2$ for all $a,b \geq 0$. Substituting the above estimate into (A.5) and using the reproducing property $h(\boldsymbol{x}) = \langle h, \Phi(\boldsymbol{x})\rangle_{\mathbb{H}}$ yields

$$\begin{aligned} &\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(h(\boldsymbol{x}),y)\right] \\ &\leq \left\{\begin{array}{ll} \sup\limits_{\mathbb{Q}^i} & \dfrac{1}{N}\displaystyle\sum_{i=1}^N \int_{\mathbb{X}\times\mathbb{Y}} \ell(\langle h,\Phi(\boldsymbol{x})\rangle_{\mathbb{H}},y)\mathbb{Q}^i(\mathrm{d}\boldsymbol{x},\mathrm{d}y) \\ \text{s.t.} & \dfrac{1}{N}\displaystyle\sum_{i=1}^N \int_{\mathbb{X}\times\mathbb{Y}} d_{\mathbb{H}}\Big(\big(\Phi(\boldsymbol{x}),y\big),\big(\Phi(\widehat{\boldsymbol{x}}_i),\widehat{y}_i\big)\Big)\mathbb{Q}^i(\mathrm{d}\boldsymbol{x},\mathrm{d}y) \leq g(\sqrt{2}\rho) \\ & \int_{\mathbb{X}\times\mathbb{Y}} \mathbb{Q}^i(\mathrm{d}\boldsymbol{x},\mathrm{d}y) = 1 \qquad i\in[N] \end{array}\right. \\ &\leq \left\{\begin{array}{ll} \sup\limits_{\mathbb{Q}^i} & \dfrac{1}{N}\displaystyle\sum_{i=1}^N \int_{\mathbb{H}\times\mathbb{Y}} \ell(\boldsymbol{x}_{\mathbb{H}},y)\mathbb{Q}^i(\mathrm{d}\boldsymbol{x}_{\mathbb{H}},\mathrm{d}y) \\ \text{s.t.} & \dfrac{1}{N}\displaystyle\sum_{i=1}^N \int_{\mathbb{H}\times\mathbb{Y}} d_{\mathbb{H}}\Big(\big(\boldsymbol{x}_{\mathbb{H}},y\big),\big(\Phi(\widehat{\boldsymbol{x}}_i),\widehat{y}_i\big)\Big)\mathbb{Q}^i(\mathrm{d}\boldsymbol{x}_{\mathbb{H}},\mathrm{d}y) \leq g(\sqrt{2}\rho) \\ & \int_{\mathbb{H}\times\mathbb{Y}} \mathbb{Q}^i(\mathrm{d}\boldsymbol{x}_{\mathbb{H}},\mathrm{d}y) = 1 \qquad i\in[N] \end{array}\right. \\ &= \sup_{\mathbb{Q}\in\mathbb{B}_{g(\sqrt{2}\rho)}(\widehat{\mathbb{P}}_N^{\mathbb{H}})} \mathbb{E}^{\mathbb{Q}}\left[\ell(\boldsymbol{x}_{\mathbb{H}},y)\right], \end{aligned}$$

where the second inequality follows from relaxing the implicit condition that the random variable $\boldsymbol{x}_{\mathbb{H}}$ must be supported on $\{\Phi(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{X}\} \subseteq \mathbb{H}$. This completes the proof of assertion (i) for regression problems.

The proof of assertion (ii) parallels that of assertion (i) with obvious modifications. Due to the different transportation metric for classification problems, however, the estimate (A.6) changes to

$$\begin{aligned} g\left(d\big((\boldsymbol{x},y),(\widehat{\boldsymbol{x}}_i,\widehat{y}_i)\big)\right) &= g\left(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|_2+\kappa|y-\widehat{y}_i|\right) \\ &\geq g\left(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|_2\right)+\kappa|y-\widehat{y}_i| \end{aligned}$$

$$\geq \|\Phi(\boldsymbol{x})-\Phi(\widehat{\boldsymbol{x}}_i)\|_{\mathbb{H}}+\kappa|y-\widehat{y}_i| = d_{\mathbb{H}}\big((\Phi(\boldsymbol{x}),y),(\Phi(\widehat{\boldsymbol{x}}_i),\widehat{y}_i)\big),$$

where the first inequality exploits the assumption that $g'(z) \geq 1$ for all $z \geq 0$, and the second inequality follows form (A.3). Further details are omitted for brevity. ■

Proof of Theorem 27 The proof follows immediately from (Schölkopf and Smola, 2001, Theorem 4.2), which applies to loss functions representable as a sum of an empirical loss depending on $(\widehat{\boldsymbol{x}}_i,\widehat{y}_i,h(\widehat{\boldsymbol{x}}_i))$, $i \leq N$, and a regularization term that is strictly monotonically increasing in $\|h\|_{\mathbb{H}}$. However, the additive separability is not needed for the proof. We remark that the optimal solution of (31) is unique if f is striclty increasing in $\|h\|_{\mathbb{H}}$. If f is only non-decreasing in $\|h\|_{\mathbb{H}}$, on the other hand, uniqueness may be lost. Details are omitted for brevity. ■

Proof of Theorem 28 Using similar arguments as in the proof of Theorem 4(ii) and observing that any Hilbert norm is self-dual, one can show that

$$\inf_{h\in\mathbb{H}} \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N^{\mathbb{H}})} \mathbb{E}^{\mathbb{Q}}\left[L(\langle h,\boldsymbol{x}_{\mathbb{H}}\rangle_{\mathbb{H}}-y)\right] = \min_{h\in\mathbb{H}} \frac{1}{N}\sum_{i=1}^N L(h(\widehat{\boldsymbol{x}}_i)-\widehat{y}_i)+\rho\,\mathrm{lip}(L)\sqrt{\|h\|_{\mathbb{H}}^2+1}.$$

By the representer theorem, which applies because the objective function of the above optimization problem is non-decreasing in $\|h\|_{\mathbb{H}}$, we may restrict the feasible set from $\mathbb{H}$ to the subset of all linearly parametrized hypotheses of the form $h(\boldsymbol{x}) = \sum_{j=1}^N \beta_j k(\boldsymbol{x},\widehat{\boldsymbol{x}}_j)$ for some $\boldsymbol{\beta} \in \mathbb{R}^N$ without sacrificing optimality. The claim then follows by observing that $h(\widehat{\boldsymbol{x}}_i) = \sum_{j=1}^N \boldsymbol{\mathcal{K}}_{ij}\beta_j$ and $\|h\|_2^2 = \langle \boldsymbol{\beta}, \boldsymbol{\mathcal{K}}\boldsymbol{\beta}\rangle$. ■

Proof of Theorem 29 Using similar arguments as in the proof of Theorem 14(ii) and observing that any Hilbert norm is self-dual, one can show that

$$\inf_{h\in\mathbb{H}} \sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N^{\mathbb{H}})} \mathbb{E}^{\mathbb{Q}}\left[L(y\langle h,\boldsymbol{x}_{\mathbb{H}}\rangle_{\mathbb{H}})\right] = \left\{\begin{array}{lll} \min\limits_{h,\lambda,s_i} & \lambda\rho+\dfrac{1}{N}\displaystyle\sum_{i=1}^N s_i & \\ \text{s.t.} & L(\widehat{y}_i h(\widehat{\boldsymbol{x}}_i)) \leq s_i & i\in[N] \\ & L(-\widehat{y}_i h(\widehat{\boldsymbol{x}}_i))-\kappa\lambda \leq s_i & i\in[N] \\ & \mathrm{lip}(L)\|h\|_{\mathbb{H}} \leq \lambda, & \end{array}\right.$$

see (Gao and Kleywegt, 2016, Theorem 1) for a full proof. By the representer theorem, which applies because the loss function

$$\begin{aligned} &f((\widehat{\boldsymbol{x}}_1,\widehat{y}_1,h(\widehat{\boldsymbol{x}}_1)),\ldots,(\widehat{\boldsymbol{x}}_N,\widehat{y}_N,h(\widehat{\boldsymbol{x}}_N)),\|h\|_{\mathbb{H}}) \\ &= \left\{\begin{array}{ll} \min\limits_{\lambda} & \lambda\rho+\dfrac{1}{N}\displaystyle\sum_{i=1}^N \max\{L(\widehat{y}_i h(\widehat{\boldsymbol{x}}_i)),L(-\widehat{y}_i h(\widehat{\boldsymbol{x}}_i))-\kappa\lambda\} \\ \text{s.t.} & \lambda \geq \mathrm{lip}(L)\|h\|_{\mathbb{H}} \end{array}\right. \end{aligned}$$

is non-decreasing in $\|h\|_{\mathbb{H}}$, we may restrict attention to all linearly parametrized hypotheses of the form $h(\boldsymbol{x}) = \sum_{j=1}^N \beta_j k(\boldsymbol{x},\widehat{\boldsymbol{x}}_j)$ for some $\boldsymbol{\beta} \in \mathbb{R}^N$ without sacrificing optimality. Thus,

the claim follows. ■

Proof of Theorem 31 For regression problems, the worst-case expected prediction loss satisfies

$$
\begin{aligned}
&\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(h(\boldsymbol{x}),y)\right] \\
&= \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ L(h(\boldsymbol{x})-y) - \lambda(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa|y-\widehat{y}_i|) \\
&\leq \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ L(h(\widehat{\boldsymbol{x}}_i)-\widehat{y}_i) + \mathrm{lip}(L)(|h(\boldsymbol{x})-y-h(\widehat{\boldsymbol{x}}_i)+\widehat{y}_i|) \\
&\qquad\qquad - \lambda(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa|y-\widehat{y}_i|) \\
&\leq \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ L(h(\widehat{\boldsymbol{x}}_i)-\widehat{y}_i) + \mathrm{lip}(L)\,\mathrm{lip}(h)\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \mathrm{lip}(L)|y-\widehat{y}_i| \\
&\qquad\qquad - \lambda(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa|y-\widehat{y}_i|) \\
&\leq \frac{1}{N}\sum_{i=1}^N \ell(h(\widehat{\boldsymbol{x}}_i),\widehat{y}_i) + \rho\,\mathrm{lip}(L)\max\left\{\mathrm{lip}(h), 1/\kappa\right\},
\end{aligned}
$$

where the equality holds due to Lemma 45, while the first and the second inequalities follow from the Lipschitz continuity of L and h, respectively. The last inequality holds by setting $\lambda = \mathrm{lip}(L)\max\{\mathrm{lip}(h), 1/\kappa\}$. Note that $\mathrm{lip}(\varphi(\psi)) \leq \mathrm{lip}(\varphi)\,\mathrm{lip}(\psi)$ for any functions φ and ψ defined on appropriate normed spaces; see for example (Rockafellar and Wets, 2009, Exercise 9.8). Thus, the Lipschitz modulus of h can be estimated in terms of the Lipschitz moduli of the activation functions σ_m and the operator norms of the weight matrices $\boldsymbol{W}_M$, which coincide with the Lipschitz moduli of the corresponding linear mappings. Formally, we have $\mathrm{lip}(h) \leq \prod_{m=1}^M \mathrm{lip}(\sigma_m)\|\boldsymbol{W}_m\|$. Substituting this estimate into the last line of the above display equation yields the postulated upper bound on the worst-case expectation for regression problems.

Similarly, for classification problems we have

$$
\begin{aligned}
&\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(h(\boldsymbol{x}),y)\right] \\
&= \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ L(yh(\boldsymbol{x})) - \lambda(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa\mathbb{1}_{\{y\neq\widehat{y}_i\}}) \\
&\leq \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ L(\widehat{y}_i h(\widehat{\boldsymbol{x}}_i)) + \mathrm{lip}(L)(|yh(\boldsymbol{x}) - \widehat{y}_i h(\widehat{\boldsymbol{x}}_i)|) - \lambda(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa\mathbb{1}_{\{y\neq\widehat{y}_i\}}) \\
&\leq \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ L(\widehat{y}_i h(\widehat{\boldsymbol{x}}_i)) + \mathrm{lip}(L)\,\mathrm{lip}(h)\,\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\|\mathbb{1}_{\{y=\widehat{y}_i\}} \\
&\qquad\qquad + \mathrm{lip}(L)\,|h(\boldsymbol{x}) + h(\widehat{\boldsymbol{x}}_i)|\,\mathbb{1}_{\{y\neq\widehat{y}_i\}} - \lambda(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa\mathbb{1}_{\{y\neq\widehat{y}_i\}}) \\
&\leq \inf_{\lambda\geq 0}\ \lambda\rho + \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ L(\widehat{y}_i h(\widehat{\boldsymbol{x}}_i)) - \lambda(\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa\mathbb{1}_{\{y\neq\widehat{y}_i\}}) \\
&\qquad\qquad + \mathrm{lip}(L)\ \max\left\{2c/\kappa, \mathrm{lip}(h)\right\}\ (\|\boldsymbol{x}-\widehat{\boldsymbol{x}}_i\| + \kappa\mathbb{1}_{\{y\neq\widehat{y}_i\}}),
\end{aligned}
$$

where $c = \sup_{h\in\mathbb{H},\boldsymbol{x}\in\mathbb{X}} |h(\boldsymbol{x})|$ is defined as in the claim. Setting $\lambda = \mathrm{lip}(L)\max\{1/\kappa, 2c/\kappa, \mathrm{lip}(h)\}$ and using the estimate $\mathrm{lip}(h) \leq \prod_{m=1}^M \mathrm{lip}(\sigma_m)\|\boldsymbol{W}_m\|$ yields the postulated upper bound on the worst-case expectation for classification problems. ■

Proof of Corollary 33 Set $\bar{\sigma} = \prod_{m\in[M]} \mathrm{lip}(\sigma_m)$. As $\kappa = \infty$, Theorem 31 implies that the worst-case expected prediction loss, both for classification and regression problems, satisfies

$$
\begin{aligned}
\sup_{\mathbb{Q}\in\mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(h(\boldsymbol{x}),y)\right] &\leq \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ \ell(h(\widehat{\boldsymbol{x}}_i),\widehat{y}_i) + \rho\,\bar{\sigma}\ \mathrm{lip}(L)\prod_{m=1}^M \|\boldsymbol{W}_m\| \\
&\leq \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ \ell(h(\widehat{\boldsymbol{x}}_i),\widehat{y}_i) + \rho\,\bar{\sigma}\ \mathrm{lip}(L)\left(\sum_{m=1}^M \frac{\|\boldsymbol{W}_m\|}{M}\right)^M,
\end{aligned}
$$

where the second inequality follows from the inequality of arithmetic and geometric means. By (Everett III, 1963, Theorem 1), if $\boldsymbol{W}^\star_{[M]}$ is a minimizer of the optimization problem

$$
\min_{\boldsymbol{W}_{[M]}}\ \frac{1}{N}\sum_{i=1}^N \sup_{\substack{\boldsymbol{x}\in\mathbb{X}\\ y\in\mathbb{Y}}}\ \ell(h(\widehat{\boldsymbol{x}}_i),\widehat{y}_i) + \rho\,\bar{\sigma}\,\mathrm{lip}(\ell)\left(\sum_{k=1}^M \frac{\|\boldsymbol{W}_k\|}{M}\right)^M,
$$

then the same $\boldsymbol{W}^\star_{[M]}$ also minimizes the constrained optimization problem

$$
\left\{
\begin{array}{ll}
\inf\limits_{\boldsymbol{W}_{[M]}} & \dfrac{1}{N}\displaystyle\sum_{i=1}^N \ell(h(\widehat{\boldsymbol{x}}_i),\widehat{y}_i) \\
\text{s.t.} & \left(\displaystyle\sum_{m=1}^M \frac{\|\boldsymbol{W}_m\|}{M}\right)^M \leq \left(\frac{\theta}{M}\right)^M,
\end{array}
\right.
$$

for $\theta = \sum_{m=1}^M \|\boldsymbol{W}^\star_m\|$. Notice that the constraint in the above optimization problem can be simplified to $\sum_{m=1}^M \|\boldsymbol{W}_m\| \leq \theta$. Hence, there exists a Lagrange multiplier $\bar{\rho}$ for the simplified constraint such that $\boldsymbol{W}^\star_{[M]}$ is a minimizer of the penalized problem

$$
\inf_{\boldsymbol{W}_{[M]}}\ \frac{1}{N}\sum_{i=1}^N \ell(h(\widehat{\boldsymbol{x}}_i),\widehat{y}_i) + \bar{\rho}\sum_{m=1}^M \|\boldsymbol{W}_m\|.
$$

This observation completes the proof. ■

A.2. Proofs of Section 4

The proof of Theorem 39 relies on the following preparatory lemma, which basically asserts that the sample average of a linearly growing function of $\boldsymbol{\xi}$ has sub-Gaussian tails.

Lemma 49 (Sub-Gaussian tails) *If Assumption 34 holds, then there exist constants $c_3 \geq 1$ and $c_4 > 0$ that depend only on the light tail constants a and A of $\mathbb{P}$ such that*

$$\mathbb{P}^N\left\{\left|\mathbb{E}^{\mathbb{P}}[f(\boldsymbol{\xi})] - \mathbb{E}^{\widehat{\mathbb{P}}_N}[f(\boldsymbol{\xi})]\right| \geq \delta\right\} \leq c_3 \exp(-c_4 N\delta^2)$$

for any $N \in \mathbb{N}$, $\delta \in [0,1]$ and function $f : \Xi \to \mathbb{R}$ with $|f(\boldsymbol{\xi}) - f(\boldsymbol{\xi}')| \leq d(\boldsymbol{\xi}, \boldsymbol{\xi}')$ for all $\boldsymbol{\xi} \in \Xi$ and some reference point $\boldsymbol{\xi}' \in \Xi$.

Proof Assume that $f : \Xi \to \mathbb{R}$ is a linear growth function with $|f(\boldsymbol{\xi}) - f(\boldsymbol{\xi}')| \leq d(\boldsymbol{\xi}, \boldsymbol{\xi}')$ for all $\boldsymbol{\xi} \in \Xi$ and some reference point $\boldsymbol{\xi}' \in \Xi$. Set $\xi_f = f(\boldsymbol{\xi})$ and $\xi'_f = f(\boldsymbol{\xi}')$. Thus, the distribution of the scalar random variable ξ_f is given by the pushforward measure $f_*(\mathbb{P})$ of $\mathbb{P}$. By construction, we have

$$\mathbb{E}^{f_*(\mathbb{P})}\left[\exp\left(|\xi_f - \xi'_f|^a\right)\right] = \mathbb{E}^{\mathbb{P}}\left[\exp\left(|f(\boldsymbol{\xi}) - f(\boldsymbol{\xi}')|^a\right)\right] \leq \mathbb{E}^{\mathbb{P}}\left[\exp\left(d(\boldsymbol{\xi}, \boldsymbol{\xi}')^a\right)\right] \leq A,$$

where the first inequality follows form the growth condition of f, while the second inequality holds because $\mathbb{P}$ satisfies Assumption 34. Hence, the distribution $f_*(\mathbb{P})$ satisfies Assumption 34 for $n = 0$ when distances on $\mathbb{R}$ are measured by the absolute value, and it inherits the light-tail constants a and A from $\mathbb{P}$. By using Theorem 35 for $n = 0$, we may thus conclude that there exist constants $c_3, c_4 > 0$ with

$$\mathbb{P}^N\left\{W(f_*(\mathbb{P}), f_*(\widehat{\mathbb{P}}_N)) \geq \delta\right\} \leq c_3 \exp\left(-c_4 N\delta^2\right) \quad \forall \delta \in [0,1],$$

where $f_*(\widehat{\mathbb{P}}_N)$ represents the empirical distribution of ξ_f, which coincides with the pushforward measure of $\widehat{\mathbb{P}}_N$ under f. By slight abuse of notation, W stands here for the Wasserstein distance between *univariate* distributions, where the absolute value is used as the ground metric. Note that the above univariate measure concentration result holds for any linear growth function f with asymptotic growth rate ≤ 1. We emphasize that $c_3 \geq 1$ because otherwise the above estimate would fail to hold for $\delta = 0$.

By construction of the Wasserstein distance in Definition 2, we have $W(f_*(\mathbb{P}), f_*(\widehat{\mathbb{P}}_N)) < \delta$ if and only if the scalar random variables ξ_f and ξ'_f admit a joint distribution Π with $\mathbb{E}^{\Pi}[|\xi_f - \xi'_f|] < \delta$ under which ξ_f and ξ'_f have marginals $f_*(\mathbb{P})$ and $f_*(\widehat{\mathbb{P}}_N)$, respectively. The inequality $W(f_*(\mathbb{P}), f_*(\widehat{\mathbb{P}}_N)) < \delta$ thus implies

$$\left|\mathbb{E}^{\mathbb{P}}[f(\boldsymbol{\xi})] - \mathbb{E}^{\widehat{\mathbb{P}}_N}[f(\boldsymbol{\xi}')]\right| = \left|\mathbb{E}^{f_*(\mathbb{P})}[\xi_f] - \mathbb{E}^{f_*(\widehat{\mathbb{P}}_N)}[\xi'_f]\right| \leq \left|\mathbb{E}^{\Pi}[\xi_f - \xi'_f]\right| = \mathbb{E}^{\Pi}\left[|\xi_f - \xi'_f|\right] < \delta.$$

By contraposition, we then obtain the implication

$$\left|\mathbb{E}^{\mathbb{P}}[f(\boldsymbol{\xi})] - \mathbb{E}^{\widehat{\mathbb{P}}_N}[f(\boldsymbol{\xi}')]\right| \geq \delta \qquad \Longrightarrow \qquad W(f_*(\mathbb{P}), f_*(\widehat{\mathbb{P}}_N)) \geq \delta,$$

which leads to the desired inequality

$$\mathbb{P}^N\left\{\left|\mathbb{E}^{\mathbb{P}}[f(\boldsymbol{\xi})] - \mathbb{E}^{\widehat{\mathbb{P}}_N}[f(\boldsymbol{\xi})]\right| \geq \delta\right\} \leq \mathbb{P}^N\left\{W(f_*(\mathbb{P}), f_*(\widehat{\mathbb{P}}_N)) \geq \delta\right\} \leq c_3 \exp\left(-c_4 N\delta^2\right).$$

The last inequality holds for all $\delta \in [0,1]$ and $N \in \mathbb{N}$, irrespective of the linear growth function f. ■

Remark 50 (Hoeffding's inequality) *If it is known that $\mathbb{P}\{\underline{f} \leq f(\boldsymbol{\xi}) \leq \overline{f}\} = 1$, then Lemma 49 reduces to Hoeffding's inequality (Boucheron et al., 2013, Theorem 2.8), in which case we may set $c_3 = 2$ and $c_4 = 2/(\overline{f} - \underline{f})^2$.*

Proof of Theorem 39 To avoid cumbersome case distinctions, we prove the theorem only in the case when (4) is a classification problem. Thus, we assume that $\Xi = \mathbb{R}^n \times \{-1, 1\}$ and that the transportation cost is of the form (16), where $\|\cdot\|$ denotes a norm on the input space $\mathbb{R}^n$. The proof for regression problems is similar and only requires minor modifications. It will be omitted for brevity.

From Theorem 14(ii) we know that

$$\sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] = \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] + \rho \operatorname{lip}(L)\Omega(\boldsymbol{w}) \quad \forall \boldsymbol{w} \in \mathbb{W},$$

where $\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_*$ can be viewed as a regularization function. For every $\rho \geq \rho'_N(\eta)$ we thus have

$$\begin{aligned}
&\mathbb{P}^N\left\{\mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] \leq \sup_{\mathbb{Q} \in \mathbb{B}_\rho(\widehat{\mathbb{P}}_N)} \mathbb{E}^{\mathbb{Q}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] \ \forall \boldsymbol{w} \in \mathbb{W}\right\} \\
\geq\ & \mathbb{P}^N\left\{0 \leq \min_{\boldsymbol{w} \in \mathbb{W}} \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] + \rho'_N(\eta)\operatorname{lip}(L)\Omega(\boldsymbol{w}) - \mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right]\right\} \\
=\ & 1 - \mathbb{P}^N\left\{\min_{\boldsymbol{w} \in \mathbb{W}} \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] + \rho'_N(\eta)\operatorname{lip}(L)\Omega(\boldsymbol{w}) - \mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] < 0\right\}. \qquad \text{(A.7)}
\end{aligned}$$

Observe that $\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)$ is Lipschitz continuous in $\boldsymbol{w}$ for every fixed $\boldsymbol{x}$ and y because the underlying univariate loss function L is Lipschitz continuous by assumption. Specifically, we have

$$|\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y) - \ell(\langle \boldsymbol{w}', \boldsymbol{x}\rangle, y)| \leq \operatorname{lip}(L)|\langle \boldsymbol{w} - \boldsymbol{w}', \boldsymbol{x}\rangle| \leq \operatorname{lip}(L)\|\boldsymbol{w} - \boldsymbol{w}'\|_\infty \|\boldsymbol{x}\|_1 \quad \forall \boldsymbol{w}, \boldsymbol{w}' \in \mathbb{W}.$$

For any $\Delta > 0$ there exists a finite set $\mathbb{W}_\Delta \subseteq \mathbb{W}$ with $\Delta = \sup_{\boldsymbol{w} \in \mathbb{W}} \inf_{\boldsymbol{w}' \in \mathbb{W}_\Delta} \|\boldsymbol{w} - \boldsymbol{w}'\|_\infty$ and whose cardinality satisfies $|\mathbb{W}_\Delta| < (\overline{\Omega}/\Delta - 1)^n < (\overline{\Omega}/\Delta)^n - 1$ where the second inequality holds because $\Delta \leq \overline{\Omega}$ by construction. In the following we set $\Delta = \overline{\Omega}/\sqrt{N}$.

As $\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)$ is Lipschitz continuous in $\boldsymbol{w}$, for every $\boldsymbol{w} \in \mathbb{W}$ there is $\boldsymbol{w}' \in \mathbb{W}_\Delta$ with

$$|\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)| \geq \left|\ell(\langle \boldsymbol{w}', \boldsymbol{x}\rangle, y)\right| - \operatorname{lip}(L)\Delta\|\boldsymbol{x}\|_1.$$

Applying this estimate twice and recalling the assumption that $\Omega(\boldsymbol{w}) \geq \underline{\Omega}$ for all $\boldsymbol{w} \in \mathbb{W}$, we may thus conclude that the probability in (A.7) is smaller or equal to

$$\mathbb{P}^N\left\{\min_{\boldsymbol{w} \in \mathbb{W}} \ \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] - \mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y)\right] + \rho'_N(\eta)\operatorname{lip}(L)\underline{\Omega} < 0\right\}$$

$$\leq \mathbb{P}^N\Big\{ \min_{\boldsymbol{w}\in\mathbb{W}_\Delta} \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)\right] - \mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)\right] - \operatorname{lip}(L)\Delta\left(\mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\|\boldsymbol{x}\|_1\right] + \mathbb{E}^{\mathbb{P}}\left[\|\boldsymbol{x}\|_1\right]\right) < -\rho_N'(\eta)\operatorname{lip}(L)\underline{\Omega}\Big\}$$

$$\leq \mathbb{P}^N\Big\{ \min_{\boldsymbol{w}\in\mathbb{W}_\Delta} \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)\right] - \mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)\right] - \operatorname{lip}(L)\Delta\left(\mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\|\boldsymbol{x}\|_1\right] - \mathbb{E}^{\mathbb{P}}\left[\|\boldsymbol{x}\|_1\right]\right) < 2\operatorname{lip}(L)\Delta M_n n A - \rho_N'(\eta)\operatorname{lip}(L)\underline{\Omega}\Big\}. \tag{A.8}$$

The second inequality in the above expression follows from the estimate

$$M_n = \max_{i\leq n}\|\boldsymbol{e}_i^n\|_* = \max_{i\leq n}\sup_{\|\boldsymbol{x}\|\leq 1}\langle \boldsymbol{e}_i^n,\boldsymbol{x}\rangle = \max_{i\leq n}\sup_{\|\boldsymbol{x}\|\leq 1}|x_i| = \sup_{\|\boldsymbol{x}\|\leq 1}\|\boldsymbol{x}\|_\infty \geq \sup_{\boldsymbol{x}\in\mathbb{R}^n}\frac{\|\boldsymbol{x}\|_1}{n\|\boldsymbol{x}\|},$$

which can be paraphrased as $\|\boldsymbol{x}\|_1 \leq M_n n\|\boldsymbol{x}\|$ for every $\boldsymbol{x}\in\mathbb{R}^n$ and thus implies

$$\mathbb{E}^{\mathbb{P}}[\|\boldsymbol{x}\|_1] \leq M_n n\mathbb{E}^{\mathbb{P}}[\|\boldsymbol{x}\|] \leq M_n n\mathbb{E}^{\mathbb{P}}[\exp(\|\boldsymbol{x}\|^a)] \leq M_n n\mathbb{E}^{\mathbb{P}}[\exp\left(d(\boldsymbol{\xi},\boldsymbol{\xi}')\right)^a)] \leq M_n n A.$$

Next, we introduce an auxiliary parameter

$$\delta = \frac{\rho_N'(\eta)\underline{\Omega} - M_n A}{\Delta + \overline{\Omega}} = \frac{2\sqrt{n\log(\sqrt{N})/c_4 + \log(c_3/\eta)/c_4}}{1+\sqrt{N}},$$

where the second equality follows from the definition of $\rho_N'(\eta)$ in the theorem statement and the convention that $\Delta = \overline{\Omega}/\sqrt{N}$. One can prove that $\delta\in[0,1]$. Indeed, the nonnegativity of δ is immediate because $c_3\geq 1$ and $c_4>0$. Moreover, we find

$$\delta \leq \frac{2\sqrt{n\log(\sqrt{N})/c_4}}{1+\sqrt{N}} + \frac{2\sqrt{\log(c_3/\eta)/c_4}}{1+\sqrt{N}} \leq 1,$$

where the first inequality follows from the observation that $\sqrt{x_1+x_2}\leq\sqrt{x_1}+\sqrt{x_2}$ for all $x_1,x_2\geq 0$, while the second inequality holds because $\log(\sqrt{N})\leq\sqrt{N}$ for all $N\in\mathbb{N}$ and $N\geq\max\left\{(16n/c_4)^2, 16\log(c_3/\eta)/c_4\right\}$, which implies that both fractions in the middle of the above expression are smaller ore qual to $\frac{1}{2}$.

Multiplying the definition of δ with $-\operatorname{lip}(L)(\Delta+\overline{\Omega})$ yields the identity

$$-\operatorname{lip}(L)\Delta\delta - \operatorname{lip}(L)\overline{\Omega}\delta = 2\operatorname{lip}(L)\Delta M_n n A - \operatorname{lip}(L)\rho_N(\eta)\underline{\Omega},$$

and thus the probability (A.8) can be bounded above by

$$\mathbb{P}^N\Big\{ \min_{\boldsymbol{w}\in\mathbb{W}_\Delta} \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)\right] - \mathbb{E}^{\mathbb{P}}\left[\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)\right] \leq -\operatorname{lip}(L)\overline{\Omega}\delta \quad\text{or}\quad -\operatorname{lip}(L)\Delta\left(\mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\|\boldsymbol{x}\|\right] - \mathbb{E}^{\mathbb{P}}\left[\|\boldsymbol{x}\|\right]\right) \leq -\operatorname{lip}(L)\Delta\delta\Big\}$$

$$\leq \mathbb{P}^N\Big\{ \min_{\boldsymbol{w}\in\mathbb{W}_\Delta} \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\frac{\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)}{\operatorname{lip}(L)\overline{\Omega}}\right] - \mathbb{E}^{\mathbb{P}}\left[\frac{\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)}{\operatorname{lip}(L)\overline{\Omega}}\right] \leq -\delta\Big\} \tag{A.9a}$$

$$+\mathbb{P}^N\Big\{\left(\mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\|\boldsymbol{x}\|\right] - \mathbb{E}^{\mathbb{P}}\left[\|\boldsymbol{x}\|\right]\right) \geq \delta\Big\}, \tag{A.9b}$$

where the inequality follows from the subadditivity of probability measures.

For any fixed $\boldsymbol{w}\in\mathbb{W}$ one can show that the function $f(\boldsymbol{\xi}) = \ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)/(\operatorname{lip}(L)\overline{\Omega})$ with $\boldsymbol{\xi}=(\boldsymbol{x},y)$ satisfies the linear growth condition $|f(\boldsymbol{\xi})-f(\boldsymbol{\xi}')|\leq d(\boldsymbol{\xi},\boldsymbol{\xi}')$ for all $\boldsymbol{\xi}\in\Xi$ if $\boldsymbol{\xi}'=(\boldsymbol{0},1)$. Details are omitted for brevity. By the subadditivity of probability measures, the probability (A.9a) is thus smaller or equal to

$$\sum_{\boldsymbol{w}\in\mathbb{W}_\Delta}\mathbb{P}^N\Big\{\Big|\mathbb{E}^{\widehat{\mathbb{P}}_N}\Big[\frac{\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)}{\operatorname{lip}(L)\overline{\Omega}}\Big] - \mathbb{E}^{\mathbb{P}}\Big[\frac{\ell(\langle \boldsymbol{w},\boldsymbol{x}\rangle,y)}{\operatorname{lip}(L)\overline{\Omega}}\Big]\Big| > \delta\Big\} \leq |\mathbb{W}_\Delta| c_3\exp(-c_4 N\delta^2),$$

where the inequality follows from Lemma 49, which applies because $\delta\in[0,1]$. Moreover, the function $f(\boldsymbol{\xi}) = \|\boldsymbol{x}\|$ with $\boldsymbol{\xi}=(\boldsymbol{x},y)$ trivially satisfies the linear growth condition $|f(\boldsymbol{\xi})-f(\boldsymbol{\xi}')|\leq d(\boldsymbol{\xi},\boldsymbol{\xi}')$ for all $\boldsymbol{\xi}\in\Xi$ if $\boldsymbol{\xi}'=(\boldsymbol{0},1)$. By Lemma 49, the probability (A.9b) is thus smaller or equal to

$$\mathbb{P}^N\Big\{\Big|\mathbb{E}^{\mathbb{P}}\left[\|\boldsymbol{x}\|\right] - \mathbb{E}^{\widehat{\mathbb{P}}_N}\left[\|\boldsymbol{x}\|\right]\Big| > \delta\Big\} \leq c_3\exp(-c_4 N\delta^2).$$

By combining the above estimates, we may conclude that the probability in (A.7) does not exceed

$$\begin{aligned}
&(|\mathbb{W}_\Delta|+1)\,c_3\exp\big(-c_4 N\delta^2\big)\\
&\leq \big(\overline{\Omega}/\Delta\big)^n c_3\exp\left(-c_4 N\left(\frac{\rho_N'(\eta)\underline{\Omega} - 2\Delta M_n n A}{\Delta+\overline{\Omega}}\right)^2\right)\\
&= N^{\frac{n}{2}} c_3\exp\left(-c_4 N\left(\frac{\rho_N'(\eta)\underline{\Omega}\sqrt{N} - 2\overline{\Omega}M_n n A}{\overline{\Omega}(\sqrt{N}+1)}\right)^2\right)\\
&\leq N^{\frac{n}{2}} c_3\exp\left(-c_4 N\left(\frac{2\overline{\Omega}\sqrt{n\log(\sqrt{N})/c_4 + \log(c_3/\eta)/c_4}}{2\overline{\Omega}\sqrt{N}}\right)^2\right)\\
&= N^{\frac{n}{2}} c_3\exp\Big(-n\log(\sqrt{N}) - \log(c_3/\eta)\Big) = \eta,
\end{aligned}$$

where the first inequality follows from the definition of δ and the assumption that $|\mathbb{W}_\Delta| < (\overline{\Omega}/\Delta)^n - 1$, the first equality holds because $\overline{\Omega}/\Delta = \sqrt{N}$, and the second inequality holds due to the definition of $\rho_N'(\eta)$. In summary, we have shown that the probability in (A.7) is at most η, and thus the claim follows. ■

A.3. Proofs of Section 5

Proof of Theorem 41 As for assertion (i), note that the absolute value function coincides with the ϵ-insensitive loss for $\epsilon=0$. Thus, (41a) follows immediately from Corollary 6 by

fixing $\boldsymbol{w}$ and by setting $\epsilon = 0$ and $\Xi = \mathbb{R}^{n+1}$. As for assertion (ii), similar arguments as in the proof of Lemma 45 show that

$$\mathcal{E}_{\min}(\boldsymbol{w}) = \sup_{\lambda \geq 0} \; -\lambda\rho + \frac{1}{N}\sum_{i=1}^{N} \inf_{\boldsymbol{x},y} \; |y - \langle \boldsymbol{w}, \boldsymbol{x}\rangle| + \lambda \|(\boldsymbol{x}, y) - (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\|. \tag{A.10}$$

The subordinate minimization problem in the first constraint of (A.10) is equivalent to

$$\begin{aligned}
&\inf_{\boldsymbol{x},y} \; |y - \langle \boldsymbol{w}, \boldsymbol{x}\rangle| + \lambda \|(\boldsymbol{x}, y) - (\widehat{\boldsymbol{x}}_i, \widehat{y}_i)\| \\
&= \inf_{\boldsymbol{x},y} \sup_{\|(\boldsymbol{q}_i, v_i)\|_* \leq \lambda} |y - \langle \boldsymbol{w}, \boldsymbol{x}\rangle| + \langle \boldsymbol{q}_i, \boldsymbol{x} - \widehat{\boldsymbol{x}}_i\rangle + v_i(y_i - \widehat{y}_i) \\
&= \sup_{\|(\boldsymbol{q}_i, v_i)\|_* \leq \lambda} \left\{ \begin{array}{ll} \inf\limits_{\boldsymbol{x},y,z} & z + \langle \boldsymbol{q}_i, \boldsymbol{x} - \widehat{\boldsymbol{x}}_i\rangle + v_i(y_i - \widehat{y}_i) \\ \text{s.t.} & z \geq y - \langle \boldsymbol{w}, \boldsymbol{x}\rangle, \; z \geq \langle \boldsymbol{w}, \boldsymbol{x}\rangle - y \end{array} \right. \\
&= \left\{ \begin{array}{ll} \sup\limits_{\boldsymbol{q}_i, v_i, r_i, t_i} & -\langle \boldsymbol{q}_i, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i v_i \\ \text{s.t.} & t_i + r_i = 1, \; t_i - r_i = v_i \\ & (r_i - t_i)\boldsymbol{w} = \boldsymbol{q}_i, \; \|(\boldsymbol{q}_i, v_i)\|_* \leq \lambda \\ & r_i, t_i \geq 0 \end{array} \right. \\
&= \left\{ \begin{array}{ll} \sup\limits_{v_i} & v_i(\langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle - \widehat{y}_i) \\ \text{s.t.} & -1 \leq v_i \leq 1 \\ & v_i \|(\boldsymbol{w}, -1)\|_* \leq \lambda, \; -v_i \|(\boldsymbol{w}, -1)\|_* \leq \lambda, \end{array} \right.
\end{aligned}$$

where the second equality holds due to Proposition 5.5.4 in (Bertsekas, 2009), and the third equality holds due to strong linear programming duality. By substituting the last optimization problem into (A.10) and replacing v_i with $-v_i$, we have

$$\begin{aligned}
\mathcal{E}_{\min}(\boldsymbol{w}) &= \left\{ \begin{array}{lll} \sup\limits_{v_i, \lambda} & -\lambda\rho + \frac{1}{N}\sum_{i=1}^{N} v_i(\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle) & \\ \text{s.t.} & -1 \leq v_i \leq 1 & i \in [N] \\ & v_i \|(\boldsymbol{w}, -1)\|_* \leq \lambda & i \in [N] \\ & -v_i \|(\boldsymbol{w}, -1)\|_* \leq \lambda & i \in [N] \end{array} \right. \\
&= \left\{ \begin{array}{lll} \sup\limits_{v_i, \lambda} & -\lambda\rho + \frac{1}{N}\sum_{i=1}^{N} v_i |\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle| & \\ \text{s.t.} & 0 \leq v_i \leq 1 & i \in [N] \\ & v_i \|(\boldsymbol{w}, -1)\|_* \leq \lambda & i \in [N] \end{array} \right. \\
&= \left\{ \begin{array}{ll} \sup\limits_{v, \lambda} & -\lambda\rho + \frac{1}{N}\sum_{i=1}^{N} v |\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle| \\ \text{s.t.} & 0 \leq v \leq 1 \\ & v \|(\boldsymbol{w}, -1)\|_* \leq \lambda \end{array} \right. \\
&= \left\{ \begin{array}{ll} \sup\limits_{v} & v \left(\frac{1}{N}\sum_{i=1}^{N} |\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle| - \rho \|(\boldsymbol{w}, -1)\|_* \right) \\ \text{s.t.} & 0 \leq v \leq 1 \end{array} \right.
\end{aligned}$$

$$= \max\left\{ \frac{1}{N}\sum_{i=1}^{N} |\widehat{y}_i - \langle \boldsymbol{w}, \widehat{\boldsymbol{x}}_i\rangle| - \rho \|(\boldsymbol{w}, -1)\|_*, 0 \right\}.$$

Thus, the claim follows. ■

Proof of Theorem 42 As for assertion (i), similar arguments as in the proof of Lemma 45 show that

$$\mathcal{R}_{\max}(\boldsymbol{w}) = \left\{ \begin{array}{lll} \inf\limits_{\lambda, s_i} & \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} s_i & \\ \text{s.t.} & \sup\limits_{\boldsymbol{x}} \mathbb{1}_{\{\widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x}\rangle \leq 0\}} - \lambda \|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}\| \leq s_i & \forall i \leq N \\ & \sup\limits_{\boldsymbol{x}} \mathbb{1}_{\{-\widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x}\rangle \leq 0\}} - \lambda \|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}\| - \kappa\lambda \leq s_i & \forall i \leq N \\ & \lambda \geq 0. & \end{array} \right. \tag{A.11a}$$

Next, observe that the indicator functions in (A.11a) can be represented as pointwise maxima of extended real-valued concave functions of the form $\mathbb{1}_{\{\widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x}\rangle \leq 0\}} = \max\{I_1(\boldsymbol{x}), 0\}$ and $\mathbb{1}_{\{\widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x}\rangle \geq 0\}} = \max\{I_2(\boldsymbol{x}), 0\}$, respectively, where

$$I_1(\boldsymbol{x}) = \begin{cases} 1 & \widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x}\rangle \leq 0, \\ -\infty & \text{otherwise}, \end{cases} \quad \text{and} \quad I_2(\boldsymbol{x}) = \begin{cases} 1 & \widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x}\rangle \geq 0, \\ -\infty & \text{otherwise}. \end{cases}$$

This allows us to reformulate (A.11a) as

$$\left\{ \begin{array}{lll} \inf\limits_{\lambda, s_i} & \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} s_i & \\ \text{s.t.} & \sup\limits_{\boldsymbol{x} \in \mathbb{R}^n} I_1(\boldsymbol{x}) - \lambda \|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}\| \leq s_i & \forall i \leq N \\ & \sup\limits_{\boldsymbol{x}} 0 - \lambda \|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}\| \leq s_i & \forall i \leq N \\ & \sup\limits_{\boldsymbol{x}} I_2(\boldsymbol{x}) - \lambda \|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}\| - \kappa\lambda \leq s_i & i \in [N] \\ & \sup\limits_{\boldsymbol{x}} 0 - \lambda \|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}\| - \kappa\lambda \leq s_i & \forall i \leq N \\ & \lambda \geq 0. & \end{array} \right.$$

Using the definition of the dual norm and applying the duality theorem (Bertsekas, 2009, Proposition 5.5.4), we find

$$\mathcal{R}_{\max}(\boldsymbol{w}) = \left\{ \begin{array}{lll} \inf\limits_{\lambda, s_i, \boldsymbol{p}_i, \boldsymbol{q}_i} & \lambda\rho + \frac{1}{N}\sum_{i=1}^{N} s_i & \\ \text{s.t.} & \sup\limits_{\boldsymbol{x}} I_1(\boldsymbol{x}) + \langle \boldsymbol{p}_i, \boldsymbol{x}\rangle - \langle \boldsymbol{p}_i, \widehat{\boldsymbol{x}}_i\rangle \leq s_i & \forall i \leq N \\ & \sup\limits_{\boldsymbol{x}} I_2(\boldsymbol{x}) + \langle \boldsymbol{q}_i, \boldsymbol{x}\rangle - \langle \boldsymbol{q}_i, \widehat{\boldsymbol{x}}_i\rangle - \kappa\lambda \leq s_i & \forall i \leq N \\ & s_i \geq 0, \|\boldsymbol{p}_i\|_* \leq \lambda, \|\boldsymbol{q}_i\|_* \leq \lambda & i \in [N]. \end{array} \right. \tag{A.11b}$$

Moreover, by strong linear programming duality we have

$$\sup_{\boldsymbol{x}} \; I_1(\boldsymbol{x}) + \langle \boldsymbol{p}_i, \boldsymbol{x}\rangle = \sup_{\boldsymbol{x}} \{1 + \langle \boldsymbol{p}_i, \boldsymbol{x}\rangle : \widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x}\rangle \leq 0\} = \inf_{r_i \geq 0} \{1 : \widehat{y}_i r_i \boldsymbol{w} = \boldsymbol{p}_i\} \tag{A.11c}$$

and

$$\sup_{\boldsymbol{x}} \; I_2(\boldsymbol{x}) + \langle \boldsymbol{q}_i, \boldsymbol{x} \rangle = \sup_{\boldsymbol{x}} \left\{ 1 + \langle \boldsymbol{q}_i, \boldsymbol{x} \rangle : \widehat{y}_i \langle \boldsymbol{w}, \boldsymbol{x} \rangle \geq 0 \right\} = \inf_{t_i \geq 0} \left\{ 1 : \widehat{y}_i t_i \boldsymbol{w} = \boldsymbol{q}_i \right\}. \tag{A.11d}$$

Substituting (A.11c) and (A.11d) into (A.11b) yields (42a). The expression (42b) for the best-case risk can be proved in a similar fashion. Details are omitted for brevity. ■

References

Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from Data*. AMLBook, 2012.

N. Agarwal, B. Bullins, and E. Hazan. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18:4148–4187, 2017.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`. Available from `http://archive.ics.uci.edu/ml`.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2): A1111–A1138, 2015.

D. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

D. Bertsimas and M. S. Copenhaver. Characterization of the equivalence of robustification and regularization in linear and matrix regression. *European Journal of Operational Research*, 2017.

C. Bhattacharyya. Second order cone programming formulations for feature selection. *Journal of Machine Learning Research*, 5:1417–1433, 2004.

J. Blanchet and K. Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 2019.

J. Blanchet, Y. Kang, and K. Murthy. Robust Wasserstein profile inference and applications to machine learning. *arXiv preprint arXiv:1610.05627*, 2016.

J. Blanchet, Y. Kang, F. Zhang, and K. Murthy. Data-driven optimal transport cost selection for distributionally robust optimization. *arXiv preprint arXiv:1705.07152*, 2017.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

L. Breiman. Bias, variance, and arcing classifiers. Technical report, University of California, Berkeley, 1996.

J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

G. C. Calafiore and L. El Ghaoui. On distributionally robust chance-constrained linear programs. *Journal of Optimization Theory and Applications*, 130(1):1–22, 2006.

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

M. Cuturi and D. Avis. Ground metric learning. *Journal of Machine Learning Research*, 15: 533–564, 2014.

M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

B. Defourny. *Machine Learning Solution Methods for Multistage Stochastic Programming*. PhD thesis, Institut Montefiore, Université de Liege, 2010.

E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, pages 647–655, 2014.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *International Conference on Machine Learning*, pages 272–279, 2008.

L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

H. Everett III. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2): 303–338, 2010.

F. Farnia and D. Tse. A minimax approach to supervised learning. In *Advances in Neural Information Processing Systems*, pages 4240–4248, 2016.

N. Fournier and A. Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.

C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio. Learning with a Wasserstein loss. In *Advances in Neural Information Processing Systems*, pages 2053–2061, 2015.

R. Gao and A. Kleywegt. Distributionally robust stochastic optimization with Wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4):902–917, 2010.

H. Gouk, E. Frank, B. Pfahringer, and M. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011.

G. A. Hanasusanto, D. Kuhn, and W. Wiesemann. A comment on computational complexity of stochastic programming problems. *Mathematical Programming A*, 159(1-2):557–569, 2016.

D. Hosmer, S. Lemeshow, and R. X. Sturdivant. *Applied Logistic Regression*. John Wiley & Sons, 2013.

K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan. The minimum error minimax probability machine. *Journal of Machine Learning Research*, 5:1253–1286, 2004.

R. Koenker. *Quantile Regression*. Cambridge University Press, 2005.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

G. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M. I. Jordan. Minimax probability machine. In *Advances in Neural Information Processing Systems*, pages 801–807, 2002a.

G. R. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002b.

N. Lawrence and B. Schölkopf. Estimating a kernel Fisher discriminant in the presence of label noise. In *International Conference on Machine Learning*, pages 306–306, 2001.

Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998. Available from `http://yann.lecun.com/exdb/mnist`.

C. Lee and S. Mehrotra. A distributionally-robust approach for finding support vector machines. *Available from Optimization Online*, 2015.

J. Lee and M. Raginsky. Minimax statistical learning with Wasserstein distances. In *Advances in Neural Information Processing Systems*, pages 2687–2696, 2018.

N. Loizou and P. Richtárik. Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods. *arXiv preprint arXiv:1712.09677*, 2017.

F. Luo and S. Mehrotra. Decomposition algorithm for distributionally robust optimization using Wasserstein metric with an application to a class of regression models. *European Journal of Operational Research*, 278(1):20–35, 2019.

T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

P. Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming A*, 171(1-2):115–166, 2018.

N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013.

B. Neyshabur, S. Bhojanapalli, and N. Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems*, pages 1574–1582, 2014.

R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 2009.

Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

K. Scaman and A. Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pages 3835–3844, 2018.

B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

S. Shafieezadeh-Abadeh, P. M. Esfahani, and D. Kuhn. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, pages 1576–1584, 2015.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

A. Shapiro. On duality theory of conic linear problems. In M. Á. Goberna and M. A. López, editors, *Semi-Infinite Programming*, pages 135–165. Kluwer Academic Publishers, 2001.

P. K. Shivaswamy and T. Jebara. Ellipsoidal kernel machines. In *Artificial Intelligence and Statistics*, pages 484–491, 2007.

P. K. Shivaswamy and T. Jebara. Maximum relative margin and data-dependent regularization. *Journal of Machine Learning Research*, 11:747–788, 2010.

P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314, 2006.

A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

T. Strohmann and G. Z. Grudic. A formulation for minimax probability machine regression. In *Advances in Neural Information Processing Systems*, pages 785–792, 2003.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B*, 58(1):267–288, 1996.

A. N. Tikhonov, V. I. Arsenin, and F. John. *Solutions of Ill-Posed Problems*. Winston, 1977.

C. Villani. *Optimal Transport: Old and New*. Springer, 2008.

J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *To appear in Bernoulli*, 2019.

T. Wiatowski, M. Tschannen, A. Stanic, P. Grohs, and H. Bölcskei. Discrete deep feature extraction: A theory and new architectures. In *International Conference on Machine Learning*, pages 2149–2158, 2016.

W. Wiesemann, D. Kuhn, and M. Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10:1485–1510, 2009.

H. Xu, C. Caramanis, and S. Mannor. Robust regression and Lasso. *IEEE Transactions on Information Theory*, 56(7):3561–3574, 2010.

T. Yang, M. Mahdavi, R. Jin, L. Zhang, and Y. Zhou. Multiple kernel learning from noisy labels by stochastic programming. In *International Conference on Machine Learning*, pages 123–130, 2012.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014.

C. Zhao and Y. Guan. Data-driven risk-averse stochastic optimization with Wasserstein metric. *Operations Research Letters*, 46(2):262–267, 2018.

Journal of Machine Learning Research 20 (2019) 1-34 Submitted 1/18; Revised 6/19; Published 6/19

Complete Search for Feature Selection in Decision Trees

Salvatore Ruggieri SALVATORE.RUGGIERI@UNIPI.IT
Department of Computer Science
University of Pisa
Largo B. Pontecorvo 3, 56127, Pisa, Italy

Editor: Inderjit Dhillon

Abstract

The search space for the feature selection problem in decision tree learning is the lattice of subsets of the available features. We design an exact enumeration procedure of the subsets of features that lead to *all and only the* distinct decision trees built by a greedy top-down decision tree induction algorithm. The procedure stores, in the worst case, a number of trees linear in the number of features. By exploiting a further pruning of the search space, we design a complete procedure for finding δ-acceptable feature subsets, which depart by at most δ from the best estimated error over any feature subset. Feature subsets with the best estimated error are called *best feature subsets*. Our results apply to any error estimator function, but experiments are mainly conducted under the wrapper model, in which the misclassification error over a search set is used as an estimator. The approach is also adapted to the design of a computational optimization of the sequential backward elimination heuristic, extending its applicability to large dimensional datasets. The procedures of this paper are implemented in a multi-core data parallel C++ system. We investigate experimentally the properties and limitations of the procedures on a collection of 20 benchmark datasets, showing that oversearching increases both overfitting and instability.

Keywords: Feature Selection, Decision Trees, Wrapper models, Complete Search

1. Introduction

Feature selection is essential for optimizing the accuracy of classifiers, for reducing the data collection effort, for enhancing model interpretability, and for speeding up prediction time (Guyon et al., 2006b). In this paper, we will consider decision tree classifiers $DT(S)$ built on a subset S of available features. Our results will hold for any top-down tree induction algorithm that greedily selects a split attribute at every node by maximizing a quality measure. The well-known C4.5 (Quinlan, 1993) and CART systems (Breiman et al., 1984) belong to this class of algorithms. Although more advanced learning models achieve better predictive performance (Caruana and Niculescu-Mizil, 2006; Delgado et al., 2014), decision trees are worth investigation, because they are building-blocks of the advanced models, e.g., random forests, or because they represent a good trade-off between accuracy and interpretability (Guidotti et al., 2018; Huysmans et al., 2011).

In this paper, we will consider the search for a *best feature subset* S, i.e., such that the estimated error $err(DT(S))$ on $DT(S)$ is minimum among all possible feature subsets. The size of the lattice of feature subsets is exponential in the number of available features. The complete search of the lattice is known to be an NP-hard problem (Amaldi and Kann, 1998). For this reason, heuristic searches are typically adopted in practice. For instance, the sequential backward elimination (**SBE**) heuristic starts with all features and repeatedly eliminates one feature at a time while error estimation does not increase. However, complete strategies do not have to be exhaustive. In particular, feature subsets that lead to duplicate decision trees can be pruned from the search space. A naïve approach that stores all distinct trees found during the search is, however, unfeasible, since there may be an exponential number of such trees. *Our first contribution* is a non-trivial enumeration algorithm **DTdistinct** of all distinct decision trees built using subsets of the available features. The procedure requires the storage of a linear number of decision trees in the worst case. The starting point is a recursive procedure for the visit of the lattice of all subsets of features. The key idea is that a subset of features is denoted by the union $R \cup S$ of two sets, where elements in R must *necessarily* be used as split attributes, and elements in S may be used or not. Pruning of the search space is driven by the observation that if a feature $a \in S$ is not used as split attribute by a decision tree built on $R \cup S$, then the feature subset $R \cup S \setminus \{a\}$ leads to the same decision tree. Duplicate decision trees that still pass such a (necessary but not sufficient) pruning condition can be identified through a test on whether or not they use all features in R. An intruiguing contribution of this paper consists in a specific order of visit of the search space, for which a negligible fraction of trees actually built are duplicates.

Enumeration of distinct decision trees can be used for finding the best feature subsets with reference to an error estimation function. Our results will hold for any error estimation function $err(DT(S))$. In experiments, we mainly adhere to the wrapper model (John et al., 1994; Kohavi and John, 1997), and consider the misclassification error on a search set that is not used for building the decision tree. The wrapper model for feature selection has shown superior performance in many contexts (Doak, 1992; Bolón-Canedo et al., 2013). We introduce the notion of a δ-acceptable feature subset, which leads to a decision tree with an estimated error that departs by at most δ from the minimum estimated error over any feature subset. *Our second contribution* is a complete search procedure $\mathbf{DTaccept}_\delta$ of δ-acceptable and best (for $\delta = 0$) feature subsets. The search builds on the enumeration of distinct decision trees. It relies on a key pruning condition that is a conservative extension of the condition above. If, for a feature $a \in S$, we have that $err(DT(R \cup S)) \leq \delta + err(DT(R \cup S \setminus \{a\}))$, then $R \cup S \setminus \{a\}$ can be pruned from the search with the guarantee of only missing decision trees whose error is at most δ from the best error of visited trees. Hence, visited feature subsets include acceptable ones.

Coupled with the tremendous computational optimization and multi-core parallelization of greedy decision tree induction algorithms, our approach makes it possible to increase the limit of practical applicability of theoretically hard complete searches. We show experimentally that the best feature subsets can be found in reasonable time for up to 60 features for small-sized datasets. Beyond such a limit, even heuristic approaches may require a large amount of time. We devise a white-box implementation **DTsbe** of **SBE**, specific for greedy decision tree algorithms, that exploits some of the pruning and computational optimization ideas. *Our third contribution* consists of a white-box optimization of **SBE** which extends its applicability to large dimensional datasets, and which exhibits, for medium and low dimensional datasets, a computational speedup of up to 100×.

©2019 Salvatore Ruggieri.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-035.html.

Both **DTaccept**$_\delta$ and **DTsbe** are implemented in a multi-core data parallel C++ system, which is made publicly available. We report experiments on 20 benchmark datasets of small-to-large dimensionality. Results confirm previous studies that oversearching increases overfitting. In addition, they also highlight that oversearching increases instability, namely variability of the subset of selected features due to perturbation of the training set. Moreover, we show that sequential backward elimination can improve the generalization error of random forests for medium to large dimensional datasets. Such an experiment is made possible only thanks to the computational speedup of **DTsbe** over **SBE**.

This paper is organized as follows. First, we recall related work in Section 2. The visit of the lattice of feature subsets is based on a generalization of binary counting enumeration of subsets devised in Section 3. Next, Section 4 introduces a procedure for the enumeration of distinct decision trees as a pruning of the feature subset lattice. Complete search of best and acceptable feature subset is then presented in Section 5. Optimization of the sequential backward elimination heuristic is discussed in Section 6. Experimental results are presented in Section 7, with additional tables reported in Appendix A. Finally, we summarize the contribution of the paper in the conclusions.

2. Related Work

Blum and Langley (1997); Dash and Liu (1997); Guyon and Elisseeff (2003); Liu and Yu (2005); Bolón-Canedo et al. (2013) provide a categorization of approaches of feature subset selection along the orthogonal axes of the evaluation criteria, the search strategies, and the machine learning tasks. Common evaluation criteria include filter models, embedded approaches, and wrapper approaches. *Filters* are pre-processing algorithms that select a subset of features by looking at the data distribution, independently from the induction algorithm (Cover, 1977). *Embedded* approaches perform feature selection in the process of training and are specific to the learning algorithm (Lal et al., 2006). *Wrappers* approaches optimize induction algorithm performances as part of feature selection (Kohavi and John, 1997). In particular, training data is split into a building set and a search set, and the space of feature subsets is explored. For each feature subset considered, the building set is used to train a classifier, which is then evaluated on the search set. Search space exploration strategies include (Doak, 1992): *hill-climbing* search (forward selection, backward elimination, bidirectional selection, beam search, genetic search), *random* search (random start hill-climbing, simulated annealing, Las Vegas), and *complete* search. The aim of complete search is to find a feature subset that optimizes an evaluation metric. Typical objectives include minimizing the size of the feature subset provided that the classifier built from it has an accuracy greater or equal to a given threshold (*dimensionality reduction*), or minimizing the empirical misclassification error of the classifier on the search set (*performance maximization*). Finally, feature subset selection has been considered for classification, regression, and clustering tasks. Machine learning models and algorithms can be either treated as *black-boxes* or, instead, feature selection methods can be specific to the model and/or algorithm at hand (*white-box*). White-box approaches are less general, but can exploit assumptions on the model or algorithm to direct and speed up the search. For instance, the best k-subset problem for linear regression (Miller, 2002) smoothly generalizes the linear regression problem to find out the subset of up to k features that best predict an independent variable.

Only complete space exploration can provide the guarantee of finding best feature subsets with respect to a given error estimation function. Several estimators have been proposed in the literature, including: the empirical misclassification error on the training set or in the search dataset; estimators adopted for tree simplification (Breslow and Aha, 1997; Esposito et al., 1997); bootstrap and cross-validation (Kohavi, 1995; Stone, 1997); and the recent *jeff* method (Fan, 2016), which is specific to decision tree models. Heuristic search approaches can lead to results arbitrarily worse than the best feature subset (Murthy, 1998). Complete search is known to be NP-hard (Amaldi and Kann, 1998). However, complete strategies do not need to be exhaustive in order to find a best feature subset. For instance, filter models can rely on monotonic evaluation metrics to support Branch & Bound search (Liu et al., 1998). Regarding wrapper approaches,the empirical misclassification error lacks the monotonicity property that would allow for pruning the search space in a complete search. Approximate Monotonicity with Branch & Bound (AMB&B) (Foroutan and Sklansky, 1987) tries and tackles this limitation, but it provides no formal guarantee that a best feature subset is found. Another form of search space pruning in wrapper approaches for decision trees has been pointed out by Caruana and Freitag (1994), who examine five hillclimbing procedures. They adopt a caching approach to prevent re-building duplicate decision trees. The basic property they observe is reported in a generalized form in this paper as Remark 6. While caching improves on the efficiency of a limited search, in the case of a complete search, it requires an exponential number of decision trees to be stored in cache, while our approach requires a linear number of them. We will also observe that Remark 6 may still leave duplicate trees in the search space, i.e., it is a necessary but not sufficient condition for enumerating distinct decision trees, while we will provide an exact enumeration and, in addition, a further pruning of trees that cannot lead to best/acceptable feature subsets.

A problem related to the focus of this paper regards the construction of optimal decision trees using non-greedy algorithms. In such a problem, the structure of a decision tree and the split attributes are determined at once as a global optimization problem. Bertsimas and Dunn (2017) and Menickelly et al. (2016); Verwer and Zhang (2017) formulate tree induction as a mixed-integer optimization problem and as an integer programming problem respectively. The optimization function is the misclassification error on the training set, possibly regularized with respect to decision tree size. Other approaches, e.g. Narodytska et al. (2018), encode tree construction as a constraint solving problem, with the aim of minimizing tree size. The search space of the optimal decision tree problem is larger than in the best feature subset problem. The former is exponential in the product of the number of features and the maximal tree depth, while the latter is exponential only in the number of features. An optimal decision tree may not be produceable by a fixed greedy algorithm, for any feature subset.

This paper significantly extends the preliminary results that appeared in Ruggieri (2017) in several directions. First, it improves on the enumeration procedure of distinct decision trees. The new ordering of visits in the search space has a clear theoretical justification, and an overhead (duplicated trees built) close to zero for all experimental datasets. Second, the paper introduces the notion of δ-acceptable feature subsets, which depart from best feature

subsets by at most δ in estimated error, and a novel algorithm that further prunes the enumeration of distinct decision trees to find out δ-acceptable feature subsets. Moreover, our approach applies to any error estimation function. Third, the experimental section (and an appendix with additional tables) now includes a comprehensive set of results on a larger collection of benchmark datasets. Fourth, the implementation of all proposed algorithms is now multi-core parallel, and it is publicly available. It reaches run-time efficiency improvements of up to 7× on an 8-core computer.

3. Enumerating Subsets

Let $S = \{a_1, \ldots, a_n\}$ be a set of n elements, with $n \geq 0$. The powerset of S is the set of its subsets: $Pow(S) = \{S' \mid S' \subseteq S\}$. There are 2^n subsets of S, and, for $0 \leq k \leq n$, there are $\binom{n}{k}$ subsets of size k. Figure 1 (left) shows the lattice (w.r.t. set inclusion) of subsets for $n = 3$. The order of visit of the lattice, or, equivalently, the order of enumeration of elements in $Pow(S)$, can be of primary importance for problems that explore the lattice as a search space. Well-known algorithms for subset generation produce lexicographic ordering, Grey code ordering, or binary counting ordering (Skiena, 2008). Binary counting maps each subset into a binary number with n bits by setting the ith bit to 1 iff a_i belongs to the subset, and generating subsets by counting from 0 to $2^n - 1$. Subsets for $n = 3$ are generated as $\{\}$, $\{a_3\}$, $\{a_2\}$, $\{a_2, a_3\}$, $\{a_1\}$, $\{a_1, a_3\}$, $\{a_1, a_2\}$, $\{a_1, a_2, a_3\}$. In this section, we introduce a recursive algorithm for a generalization of reverse binary counting (namely, counting from $2^n - 1$ down to 0) that will be the building block for solving the problem of generating distinct decision trees. Let us start by introducing the notation $R \bowtie P = \cup_{S' \in P}\{R \cup S'\}$ to denote sets obtained by the union of R with elements of P. In particular:

$$R \bowtie Pow(S) = \cup_{S' \subseteq S}\{R \cup S'\}$$

consists of the subsets of $R \cup S$ that *necessarily* include R. This generalization of powersets will be crucial later on when we have to distinguish predictive attributes that *must* be used in a decision tree from those that *may* be used. A key observation of binary counting is that subsets can be partitioned between those including the value a_1 and those not including it. For example, $Pow(\{a_1, a_2, a_3\}) = (\{a_1\} \bowtie Pow(\{a_2, a_3\})) \cup (\emptyset \bowtie Pow(\{a_2, a_3\}))$. We can iterate the observation for the leftmost occurrence of a_2 and obtain:

$$Pow(\{a_1, a_2, a_3\}) = (\{a_1, a_2\} \bowtie Pow(\{a_3\})) \cup (\{a_1\} \bowtie Pow(\{a_3\})) \cup \ (\emptyset \bowtie Pow(\{a_2, a_3\})).$$

By iterating again for the leftmost occurrence of a_3, we conclude:

$$\begin{aligned} Pow(\{a_1, a_2, a_3\}) &= (\{a_1, a_2, a_3\} \bowtie Pow(\emptyset)) \cup (\{a_1, a_2\} \bowtie Pow(\emptyset)) \cup \\ &\quad (\{a_1\} \bowtie Pow(\{a_3\})) \cup (\emptyset \bowtie Pow(\{a_2, a_3\})) \end{aligned}$$

Since $R \bowtie Pow(\emptyset) = \{R\}$, the leftmost set in the above union is $\{\{a_1, a_2, a_3\}\}$. In general, the following recurrence relation holds.

Lemma 1 *Let $S = \{a_1, \ldots, a_n\}$. We have:*

$$R \bowtie Pow(S) = \{R \cup S\} \cup \bigcup_{i=n,\ldots,1} (R \cup \{a_1, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\})$$

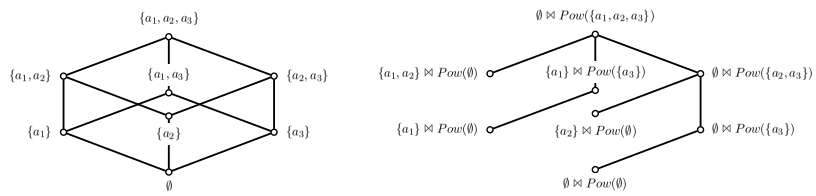


Figure 1: Lattice of subsets and reverse binary counting.

Proof The proof is by induction on n. The base case $n = 0$ is trivial: $R \bowtie Pow(\emptyset) = \{R\}$ by definition. Consider now $n > 0$. Since $Pow(S) = (\{a_1\} \bowtie Pow(S \setminus \{a_1\})) \cup Pow(S \setminus \{a_1\})$, we have: $R \bowtie Pow(S) = R \bowtie ((\{a_1\} \bowtie Pow(\{a_2, \ldots, a_n\})) \cup (\emptyset \bowtie Pow(\{a_2, \ldots, a_n\})))$. Since the $\bowtie$ operator satisfies:

$$R \bowtie (P_1 \cup P_2) = (R \bowtie P_1) \cup (R \bowtie P_2) \quad \text{and} \quad R_1 \bowtie (R_2 \bowtie P) = (R_1 \cup R_2) \bowtie P$$

we have: $R \bowtie Pow(S) = ((R \cup \{a_1\}) \bowtie Pow(\{a_2, \ldots, a_n\})) \cup (R \bowtie Pow(\{a_2, \ldots, a_n\}))$. By induction hypothesis on the leftmost occurrence of $\bowtie$:

$$\begin{aligned} R \bowtie Pow(S) &= \{R \cup \{a_1\} \cup \{a_2, \ldots, a_n\}\} \cup \\ &\quad \bigcup_{i=n,\ldots,2} (R \cup \{a_1\} \cup \{a_2, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\}) \cup \\ &\quad R \bowtie Pow(\{a_2, \ldots, a_n\}) \\ &= \{R \cup S\} \cup \bigcup_{i=n,\ldots,1} (R \cup \{a_1, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\}) \end{aligned}$$

■

This result can be readily translated into a procedure **subset(R, S)** for the enumeration of elements in $R \bowtie Pow(S)$. In particular, since $\emptyset \bowtie Pow(S) = Pow(S)$, **subset($\emptyset$, S)** generates all subsets of S. The procedure is shown as Algorithm 1. The search space of the procedure is the tree of the recursive calls of the procedure. The search space for $n = 3$ is reported in Figure 1 (right). According to line 1 of Algorithm 1, the subset outputted at a node labelled as $R \bowtie Pow(S)$ is $R \cup S$. Hence, the output for $n = 3$ is the reverse counting ordering: $\{a_1, a_2, a_3\}$, $\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_1\}$, $\{a_2, a_3\}$, $\{a_2\}$, $\{a_3\}$, $\{\}$. Two key properties of Algorithm 1 will be relevant for the rest of the paper.

Remark 2 *A set $R' \cup S'$ generated at a non-root node of the search tree of Algorithm 1 is obtained by removing an element from the set $R \cup S$ generated at its father node. In particular, $R' \cup S' = R \cup S \setminus \{a\}$ for some $a \in S$.*

The invariant $|R' \cup S'| = |R \cup S|$ readily holds for the loop at lines 4–8 of Algorithm 1. Before the recursive call at line 6, an element of S is removed from R', hence the set $R' \cup S'$ outputted at a child node has one element less than the set $R \cup S$ outputted at its father node.

Algorithm 1 subset(R, S) enumerates $R \bowtie Pow(S)$

1: **output** $R \cup S$
2: $R' \leftarrow R \cup S$
3: $S' \leftarrow \emptyset$
4: **for** $a_i \in S$ **do**
5: $\quad R' \leftarrow R' \setminus \{a_i\}$
6: $\quad$ subset(R', S')
7: $\quad S' \leftarrow S' \cup \{a_i\}$
8: **end for**

Remark 3 *The selection order of $a_i \in S$ at line 4 of Algorithm 1 is irrelevant.*

The procedure does not rely on any specific order of selecting members of S, which is a form of *don't care non-determinism* in the visit of the lattice. Any choice generates all elements in $R \bowtie Pow(S)$. In case of an *a priori* positional order of attributes, namely line 4 is "**for** $a_i \in S$ **order by** i **desc do**", Algorithm 1 produces precisely the reversed binary counting order. However, if the selection order varies from one recursive call to another, then the output is still an enumeration of subsets.

4. Generating All Distinct Decision Trees

We build on the subset generation procedure to devise an algorithm for the enumeration of all distinct decision trees built on subsets of the predictive features.

4.1. On Top-Down Greedy Decision Tree Induction

Let us first introduce some notation and assumptions. Let $F = \{a_1, \ldots, a_N\}$ be the set of predictive features, and $S \subseteq F$ a subset of them. We write $T = DT(S)$ to denote the decision tree built from features in S on a fixed training set. Throughout the paper, we make the following assumption on the node split criterion in top-down greedy decision tree induction.

Assumption 4 *Let $T = DT(S)$. A split attribute at a decision node of T is chosen as $argmax_{a \in S} f(a, C)$, where $f()$ is a quality measure and C are the cases of the training set reaching the node.*

While the assumption regards univariate splits, it can be restated for bi-variate or multi-variate split conditions, and the theoretical results in this paper can be adapted to such general cases. However, since we build on a software that deals with univariate splits only (see Section 7), experiments are restricted to such a case. Moreover, the results will hold for any quality measure $f()$ as far as the split attributes are chosen as the ones that maximize $f()$. Examples of quality measures used in this way include Information Gain (IG), Gain Ratio[1] (GR), and the Gini index, which are adopted in the C4.5 (Quinlan, 1993) and in the CART systems (Breiman et al., 1984). A second assumption regards the stopping criterion

1. Gain Ratio normalizes Information Gain over the Split Information (SI) of an attribute, i.e., GR = IG/SI. This definition does not work well for attributes which are (almost) constants over the cases C,

in top-down decision tree construction. Let $stop(S, C)$ be the boolean result of the stopping criterion at a node with cases C and predictive features S.

Assumption 5 *If $stop(S, C) = true$ then $stop(S', C) = true$ for every $S' \subseteq S$.*

The assumption states that either: (1) the stopping criterion does not depend on S; or, if it does, then (2) stopping is monotonic with regard to the set of predictive features. (1) is a fairly general assumption, since typical stopping criteria are based on the size of cases C at a node and/or on the purity of the class attribute in C. We will later on consider the stopping criterion of C4.5 which halts tree construction if the number of cases of the training set reaching the current node is lower than a minimum threshold m (formally, $stop(S, C)$ is true iff $|C| < m$). Another widely used stopping criterion satisfying (1) consists of setting a maximum depth of the decision tree. (2) applies to criteria which require minimum quality of features for splitting a node. E.g., the C4.5 additional criterion of stopping if IG of all features is below a minimum threshold satisfies the assumption. The following remark, which is part of the decision tree folklore (see e.g., Caruana and Freitag (1994)), states a useful consequence of Assumptions 4 and 5. Removing any feature not used in a decision tree from the initial set of features does not affect the result of tree building.

Lemma 6 *Let $features(T)$ denote the set of split attributes in a decision tree $T = DT(S)$. For every S' such that $S \supseteq S' \supseteq features(T)$, $DT(S') = T$.*

Proof If a decision tree T built from S uses only features from $U = features(T) \subseteq S$, then at any decision node of T it must be true that $argmax_{a \in S} f(a, C) = argmax_{a \in U} f(a, C)$. Hence, removing any unused attribute in $S \setminus U$ will not change the result of maximizing the quality measure and then, by Assumption 4, the split attribute at a decision node. Moreover, by Assumption 5, a leaf node in T will remain a leaf node for any subset of S. ■

4.2. Enumerating Distinct Decision Trees

Consider a subset of features $R \cup S$, where R must be necessarily used by a decision tree and $S = \{a_1, \ldots, a_n\}$ may be used or not. Let $T = DT(R \cup S)$, and $U = features(T) \supseteq R$ be the features used in split nodes of T. Therefore, $S \cap U = \{a_1, \ldots, a_k\}$ is the set of features in S actually selected as split features, and $S \setminus U = \{a_{k+1}, \ldots, a_n\}$ is the set of features never selected as split features. By Lemma 6, the decision tree T is equal to the one built starting from features $R \cup \{a_1, \ldots, a_k\}$ plus any subset of $\{a_{k+1}, \ldots, a_n\}$. In symbols, all the decision trees for feature subsets in $(R \cup \{a_1, \ldots, a_k\}) \bowtie Pow(\{a_{k+1}, \ldots, a_n\})$ do coincide with T. We will use this observation to remove from the recurrence relation of Lemma 1 some sets in $R \bowtie Pow(S)$ which lead to duplicate decision trees. Formally, when searching for feature subsets that lead to distinct decision trees, the recurrence relation can be modified as:

$$R \bowtie Pow(S) = \{R \cup S\} \cup \bigcup_{i=k,\ldots,1} (R \cup \{a_1, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\})$$

i.e., when SI ≈ 0. Quinlan (1986) proposed the heuristic of restricting the evaluation of GR only to attributes with above *average* IG. The heuristic is implemented in the C4.5 system (Quinlan, 1993). It clearly breaks Assumption 4, making the selection of the split attribute dependent on the set S. An heuristic that satisfies Assumption 4 consists of restricting the evaluation of GR only for attributes with IG higher than a *minimum* threshold.

Algorithm 2 DTdistinct(R, S) enumerates distinct decision trees using feature subsets in $R \bowtie Pow(S)$.

1: **build tree** $T = DT(R \cup S)$
2: $U \leftarrow features(T)$
3: **if** $R \subseteq U$ **then**
4: **output** T
5: **end if**
6: $R' \leftarrow R \cup (S \cap U)$
7: $S' \leftarrow S \setminus U$
8: **for** $a_i \in S \cap U$ **order by** $rk_{frontier}(T)$ **do**
9: $R' \leftarrow R' \setminus \{a_i\}$
10: DTdistinct(R', S')
11: $S' \leftarrow S' \cup \{a_i\}$
12: **end for**

since the missing union:

$$\bigcup_{i=n,\ldots,k+1} (R \cup \{a_1, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\}) \tag{1}$$

is included in $(R \cup \{a_1, \ldots, a_k\}) \bowtie Pow(\{a_{k+1}, \ldots, a_n\})$, and then it contains sets of features V such that $DT(V) = DT(R \cup S)$. In particular, this implies the following property, for any error estimation function $err()$, which will be useful later on:

$$err(DT(R \cup \{a_1, \ldots, a_k\})) = err(DT(V)) \tag{2}$$
$$\text{for all } V \in \textstyle\bigcup_{i=n,\ldots,k+1} (R \cup \{a_1, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\}).$$

The simplified recurrence relation prunes from the the search space feature subsets that lead to duplicated decision trees. However, we will show in Example 1 that such a pruning alone is not sufficient to generate distinct decision trees only, i.e., duplicates may still exist.

Algorithm 2 provides an enumeration of *all and only the distinct decision trees*. It builds on the subset generation procedure. Line 1 constructs a tree T from features $R \cup S$. Features in the set $S \setminus U$ of unused features in T are not iterated over in the loop at lines 8–12, since those iterations would yield the same tree as T. This is formally justified by the modified recurrence relation above. The tree T is outputted at line 4 only if $R \subseteq U$, namely features *required* to be used (i.e., R) are *actually* used in decision node splits. We will shows that such a test characterizes a uniqueness condition for all feature subsets that lead to a same decision tree. Hence, it prevents outputting more than once a decision tree that can be obtained from multiple paths of the search tree.

Example 1 *Let $F = \{a_1, a_2, a_3\}$. Assume that a_1 has no discriminatory power unless data has been split by a_3. More formally, $DT(S) = DT(S \setminus \{a_1\})$ if $a_3 \notin S$. The visit of feature subsets of Figure 1 (right) gives rise to the trees built by* ***DTdistinct(***$\emptyset$***, F)*** *as shown in Figure 2 (left). For instance, the subset $\{a_1, a_2\}$ visited at the node labelled $\{a_1, a_2\} \bowtie \emptyset$ in Figure 1 (right), produces the decision tree $DT(\{a_1, a_2\})$. By assumption, such a tree*

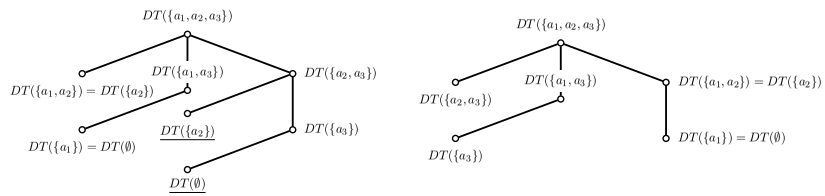


Figure 2: Search spaces of Algorithm 2 for different selection orders.

is equal to $DT(\{a_2\})$, which is a duplicate tree produced in another node – underlined in Figure 2 (left) – corresponding to the feature set visited at the node labelled $\{a_2\} \bowtie \emptyset$. Another example regarding $DT(\{a_1\}) = DT(\emptyset)$ is shown in Figure 2 (left), together with its underlined duplicated tree. Unique trees for two or more duplicates are characterized by the fact that features appearing to the left of $\bowtie$ must necessarily be used as split features by the constructed decision tree. In the two previous example cases, the nodes underlined output their decision trees, while the other duplicates do not pass the test at line 3 of Algorithm 2.

The following non-trivial result holds.

Theorem 7 ***DTdistinct(****R, S)* *outputs the distinct decision trees built on sets of features in $R \bowtie Pow(S)$.*

Proof The search space of **DTdistinct** is a pruning of the search space of **subset**. Every tree built at a node and outputted is then constructed from a subset in $R \bowtie Pow(S)$. By Remark 3, the order of selection of $a_i \in S \cap U$ at line 8 is irrelevant, since any order will lead to the same space $R \bowtie Pow(S)$.

Let us first show that decision trees in output are all distinct. The key observation here is that, by line 4, all features in R are used as split features in the outputted decision tree. The proof proceeds by induction on the size of S. If $|S| = 0$, then there is at most one decision tree in output, hence the conclusion. Assume now $|S| > 0$, and let $S = \{a_1, \ldots, a_n\}$. By Lemma 1, any two recursive calls at line 10 have parameters $(R \cup \{a_1, \ldots, a_{i-1}\}, \{a_{i+1}, \ldots, a_n\})$ and $(R \cup \{a_1, \ldots, a_{j-1}\}, \{a_{j+1}, \ldots, a_n\})$, for some $i < j$. Observe that a_i is missing as a predictive attribute in the trees in output from the first call, while by inductive hypothesis it must be a split attribute in the trees in output by the second call. Hence, the trees in output from recursive calls are all distinct among them. Moreover, they are all different from $T = DT(R \cup S)$, because recursive calls do not include some feature $a_i \in S \cap U$ that is by definition used in T.

Let us now show that trees pruned at line 8 or at line 4 are already outputted elsewhere, which implies that every distinct decision tree is outputted at least once. First, by Lemma 6, the trees that would have been outputted in the pruned iterations at line 8 (i.e., for $a_i \in S \setminus U$) are equal to the tree of $T = DT(R \cup S)$. Second, if the tree T is not outputted at line 4, because $R \not\subseteq U$, we have that it is outputted at another node of the search tree. The proof is by induction on $|R|$. For $|R| = 0$ it is trivial, because the the premise $R \not\subseteq U$ does not hold. Let $R = \{a_1, \ldots, a_n\}$, with $n > 0$, and let $a_1, \ldots, a_n$ be

in the order they have been added by recursive calls. Fix $R' = \{a_1, \ldots, a_{i-1}\}$ such that $a_i \notin U$ and $R' \subseteq U$. There is a sibling node or a sibling of an ancestor node in the search tree corresponding to a call with parameters R' and $S' \supseteq \{a_{i+1}, \ldots, a_n\} \cup S$. By inductive hypothesis on $|R'| < |R|$, the distinct decision trees with features in $R' \bowtie Pow(S')$ are all outputted, including T because T has split features in $R \cup S \setminus \{a_i\}$ which belongs to $R' \bowtie Pow(S')$. ■

The proof of Theorem 7 does not assume any specific order at line 8 of Algorithm 2. Any order would produce the enumeration of distinct decision trees – this is a consequence of Remark 3. However, the order may impact on the size of the search space.

Example 2 *Reconsider Example 1. The order of selection of a_i's in the visit of Figure 2 (left) is by descending i's. This ordering does not take into account the fact that a_3 has more discriminatory power than a_1, i.e., its presence gives rise to more distinct decision trees. Consider instead having a_3 removed in the rightmost child of the root, e.g., the selection order a_1, a_2, and a_3. The search space of* ***DTdistinct(∅, F)*** *is reported in Figure 2 (right). Notice that no duplicate decision tree is built here, and that the size of the search space is smaller than in the previous example. In fact, the node labelled as $DT(\{a_1, a_2\}) = DT(\{a_2\})$ corresponds to the exploration of $\emptyset \bowtie \{a_1, a_2\}$. The a_1 attribute is unused and hence, it is pruned at line 8 of Algorithm 2. The sub-space to be searched consists then of only the subsets of $\{a_1\}$, not all subsets of $\{a_1, a_2\}$. The child node finds $DT(\{a_1\}) = DT(\emptyset)$ and then it stops recursion since there are no used attributes to iterate over at line 8.*

Algorithm 2 adopts a specific order intended to be effective in pruning the search space. In particular, our objective is to minimize the number of duplicated trees built. In fact, even though duplicates are not outputted, building them has a computational cost that should be minimized. Duplicated decision trees are detected through the test $R \subseteq U$ at line 4 of Algorithm 2. Thus, we want to minimize the number of recursive calls **DTdistinct(R', S')** where attributes in R' have lower chances of being used. Since required attributes are removed from R' one at a time at line 9 (and added to the set of possibly used attributes S' at line 11), this means ranking attributes in $S \cap U$ by increasing chances of being used in decision trees built in recursive calls. How do we estimate such chances?

Example 3 *Consider the sample decision tree at Figure 3(a). It is built on the set of features $F = \{a_1, a_2, a_3, a_4\}$. Which attributes have the highest chance of being used if included in a subset of F? Whenever included in the subset, a_1 will be certainly used at the root node. In fact, it already maximizes the quality measure on F, hence by Assumption 4 it will also maximizes the quality measure over any subset of F. Assume now a_1 is selected for inclusion. Both a_2 and a_3 will be certainly selected as split attributes, for the same reason as above. Which one should be preferred first? Let us look at the sub-trees rooted at a_2 and a_3. If a_2 is selected, then the sub-tree rooted at a_2 uses no further attribute. In fact, a_1 cannot be counted as a further attribute because it is known to be already used. Conversely, if a_3 is selected, the sub-tree rooted at a_3 ensures that attribute a_4 can be used. Therefore, having a_3 gives more chances of using further attributes in decision tree building. Therefore, a_3 should be selected for inclusion before a_2. Finally, sub-trees rooted at a_2 and a_4 use no*

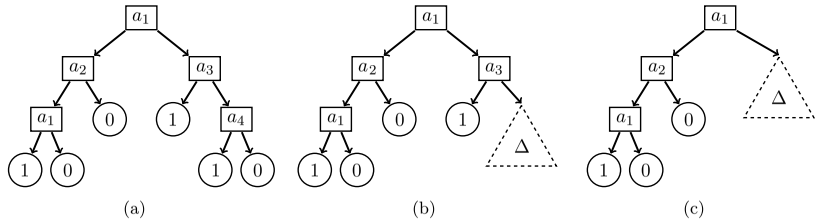


Figure 3: A sample decision tree, and two subtrees replaced with an oracle Δ. Internal nodes are labelled with split attributes, and leaves are labelled with class value.

further attributes, and we break the tie by selecting a_2 before a_4. In summary, the rank of attributes in F with increasing chances of being used is a_4, a_2, a_3, a_1.

Let us formalize the intuition of this example. We define an *a-frontier node* of a decision tree T w.r.t. a set R' of features, as a decision node that uses $a \notin R'$ for the first time in a path from the root to a node. A frontier node is any a-frontier node, for any feature a. Based on the previous example, we should count the number of attributes that are used in sub-trees rooted at frontier nodes.

Definition 8 *We define $frontier(T, R', a)$ as the number of distinct features not in R' that are used in sub-trees of T rooted at a-frontier nodes of T w.r.t. R'.*

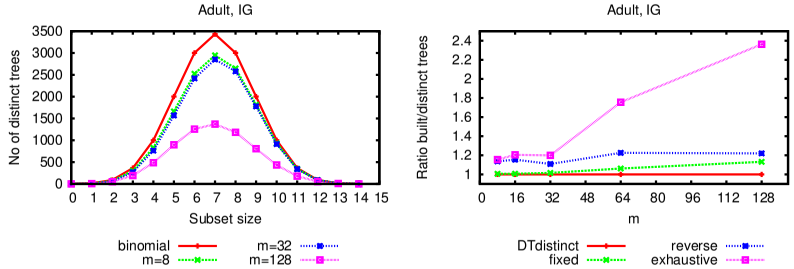
Notice that attributes in R' are excluded from the counting in $frontier()$. The idea is that R' will include attributes that must already appear somewhere in a decision tree (in between the root and frontier nodes), and thus their presence in the sub-tree rooted at a does not imply further usability of such attributes. We are now in the position to introduce the ranking $rk_{frontier}$ based on ascending $frontier()$.

Definition 9 *Let $T = DT(R \cup S)$ and $U = features(T)$. $rk_{frontier}(T)$ is the order $r_1, \ldots, r_k$ of elements in $S \cap U$ such that, for $i = k, \ldots, 1$:*

$$r_i = argmax_{a \in (S \cap U) \setminus \{r_{i+1}, \ldots, r_k\}} \; frontier(T, R \cup \{r_{i+1}, \ldots, r_k\}, a).$$

The definition of the ranking iterates from the last to the first position, and at each step selects the feature which maximizes the $frontier()$ measure. Iteration is necessary due to the fact that the frontier nodes depend on the features selected at the previous step. Intuitively, the ordering tries to keep as much as possible sub-trees with large sets of not yet ranked attributes. This is in line with the objective of ranking features based on increasing chances of being used in decision trees of recursive calls.

Example 4 *Reconsider Example 3 and the decision tree $T = DT(F)$ in Figure 3(a). Assume that $R = \emptyset$ and $S = F = \{a_1, a_2, a_3, a_4\}$. The set of used features is $U = F$.*



(a) Distribution of distinct decision trees. (b) Ratio duplicate/distinct decision trees.

Figure 4: Distinct decision trees and overhead of **DTdistinct** on the Adult dataset.

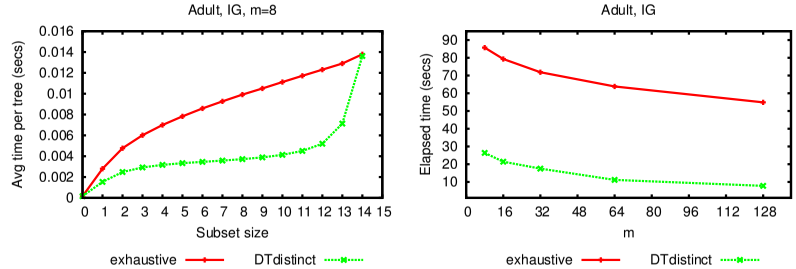
The rank of attributes starts by defining $r_4 = argmax_{a \in \{a_1,a_2,a_3,a_4\}}$ $frontier(T, \emptyset, a)$ *which is trivially the split attribute* a_1 *at root node of* T *– the only frontier node of* T *w.r.t.* $\emptyset$*. Next,* $r_3 = argmax_{a \in \{a_2,a_3,a_4\}}$ $frontier(T, \{a_1\}, a)$ *is defined by looking at the frontier nodes, which are those using* a_2 *and* a_3*. As discussed in Example 3,* $frontier(T, \{a_1\}, a_2) = 1$ *and* $frontier(T, \{a_1\}, a_3) = 2$*. Thus, we have* $r_3 = a_3$*. At the third step,* $r_2 = argmax_{a \in \{a_2,a_4\}}$ $frontier(T, \{a_1, a_2\}, a)$ *is defined by looking at the frontier nodes using* a_2 *and* a_4*. Both have a frontier of* 1*, so we fix* $r_2 = a_2$*. Finally,* r_1 *must be necessarily be* a_4*. Summarizing, the ordering provided by* $rk_{frontier}(T)$ *is* a_4, a_2, a_3, a_1 *as stated in Example 3.*

Let us now point out some properties of **DTdistinct**.

Property 1: linear space complexity. Consider the set F of all features, with N elements. **DTdistinct(**$\emptyset$**,** F**)** is computationally linear in space (per number of trees built) in N. In fact, there are at most N nested calls, since the size of the second parameter decreases at each call. Summarizing, at most N decision trees are built and stored in the nested calls, i.e., space complexity is linear per number of trees built. An exhaustive search would instead keep in memory the distinct decision trees built in order to check whether a new decision tree is a duplicate. Similarly, so will do applications based on complete search that exploit duplicate pruning through caching of duplicates (Caruana and Freitag, 1994). Those approaches would require exponential space, since the number of distinct trees can be exponential as shown in the next example.

Example 5 *Let us consider the well-known Adult dataset*[2] *(Lichman, 2013), consisting of 48,842 cases, and with* $N = 14$ *predictive features and a binary class. Figure 4(a) shows, for the IG split criterion, the distributions of the number of distinct decision trees w.r.t. the size of feature subset. The distributions are plotted for various values of the stopping parameter* m *(formally,* $stop(S, C)$ *is true iff* $|C| < m$*). For low* m *values, the distribution approaches the binomial; hence, the number of distinct decision trees approaches* 2^N*.*

2. See Section 7 for the experimental settings.



(a) Average elapsed time for tree building. (b) Total elapsed time.

Figure 5: **DTdistinct** elapsed times on the Adult dataset.

Property 2: reduced overhead. Algorithm 2 may construct duplicated decision trees at line 1, which, however, are not outputted due to the test at line 3. It is legitimate to ask ourselves how many duplicates are constructed. Or, in other words, how effective is the selection order at line 8 based on $rk_{frontier}()$. Formally, we measure such an overhead as the ratio of all decision trees constructed at line 1 over the number of distinct decision trees. An ideal ratio of 1 means that no duplicate decision tree is constructed at all.

Example 6 (Ctd.) *Figure 4(b) shows the overhead at the variation of* m *for three possible orderings of selection at line 8 of Algorithm 2. One is the the ordering stated by* ***DTdistinct****, based on* $rk_{frontier}()$*. The second one is the reversed order, namely* $a_n, \ldots, a_1$ *for* $rk_{frontier}()$ *being* $a_1, \ldots, a_n$*. The third one is based on assigning a static index* $i \in [1, N]$ *to features* a_i*'s, and then ordering over* i*. The* $rk_{frontier}()$ *ordering used by* ***DTdistinct*** *is impressively effective, with a ratio of almost* 1 *everywhere.*

The effectiveness of the $rk_{frontier}()$ ordering will be confirmed in the experimental section. Figure 4(b) also reports the ratio of the number of trees in an exhaustive search (which are 2^N for N features) over the number of distinct trees. Smaller m's lead to a smaller ratio, because built trees are larger in size and hence there are more distinct ones. Thus, for small m values, pruning duplicate trees does not guarantee alone a considerably more efficient enumeration than exhaustive search. The next property will help in such cases.

Property 3: feature-incremental tree building. The construction of each single decision tree at line 1 of Algorithm 2 can be sped up by Remark 2. The decision tree T' at a child node of the search tree differs from the decision tree T built at the father node by one missing attribute a_i. The construction of T' can then benefit from this observation. In the implementation of Algorithm 2, we first recursively clone T and then re-build only sub-trees rooted at nodes whose split attribute is a_i. This requires maintaining in memory the trees built along recursive calls, which gives rise to the linear space complexity (in the number of trees) of the algorithm. However, it allows for incrementally building T' from T.

Example 7 (Ctd.) *Figure 5(a) shows the average elapsed time required to built a decision tree w.r.t. the size of the feature subset, for the fixed parameter $m = 8$. The exhaustive search requires an average time linear in the size of the feature subset. Due to incremental tree building,* ***DTdistinct*** *requires instead a sub-linear time.*

Property 4: multi-core parallelization. The parallelization of greedy decision tree algorithms is non-trivial, due to their recursive nature, with some upper bounds on the maximum speedup achievable (Aldinucci et al., 2014). Exhaustive exploration of feature subsets is, instead, parallelizable with perfect scalability, due the exponential number of independent tasks of tree construction. Regarding our pruned search, the loop at lines 8–12 of Algorithm 2 has no strong dependency between iterations, and it can also be easily parallelized on multi-core platforms. Our implementation of Algorithm 2 runs the loop at lines 8–12 in task parallel threads. The construction of the tree at line 1 is also executed using nested task parallelism. For fair comparison, the implementation of exhaustive search exploits nested parallelism as well in the enumeration and in the construction of trees.

Example 8 (Ctd.) *Figure 5(b) contrasts the total elapsed times of exhaustive search and* ***DTdistinct****. For small values of m, the number of trees built by exhaustive search approaches the number of distinct decision trees (see Figure 4(b)). Nevertheless, the running time of* ***DTdistinct*** *is constantly better than the exhaustive search. This improvement is due to the incremental building of decision trees. The computational efficiency in terms of absolute elapsed times is, in addition, due the effectiveness of parallel implementation, which in the example at hand runs on an low-cost 8-core machine reaching a 7× speedup.*

5. Best and Acceptable Feature Subsets

Consider an error estimation function $err(T)$ for a decision tree T. A best feature subset is such that the estimated error of the tree built on such features is minimum among decision trees built on any feature subset. It is δ-acceptable, if the estimated error is lower or equal than the minimum plus δ.

Definition 10 *Let F be a set of features. We define $err_{bst} = min_{S \subseteq F} err(DT(S))$, and call it the best estimated error w.r.t. feature subsets. For $\delta \geq 0$, a feature subset $S \subseteq F$ is δ-acceptable if:*

$$err(DT(S)) \leq \delta + err_{bst}.$$

When $\delta = 0$, we call S a best feature subset. Finally, $DT(S)$ is called a δ-acceptable decision tree.

The *δ-acceptable feature subset problem* consists of finding a δ-acceptable feature subset. In particular, for $\delta = 0$, it consists of finding a best feature subset.

We make no assumption on the error estimation function $err(T)$. In experiments, unless otherwise stated, we adhere to the wrapper model (John et al., 1994; Kohavi and John, 1997), by assuming that the available training set is split into a *building* set, used to build the decision trees $T = DT(S)$ on a subset S of features F, and a *search* set. Error is

Algorithm 3 **DTaccept**$_\delta(R, S)$ finds a δ-acceptable feature subset S_{acc} in $R \bowtie Pow(S)$.

```
1: // e_acc is initialized outside to ∞
2: build tree T = DT(R ∪ S)
3: U ← features(T)
4: if err(T) ≤ e_acc then
5:    e_acc ← err(T)
6:    S_acc ← U
7: end if
8: U ← greedy_δ(U, R, S)
9: R' ← R ∪ (S ∩ U)
10: S' ← S \ U
11: for a_i ∈ S ∩ U order by rk_frontier(T) do
12:    R' ← R' \ {a_i}
13:    DTaccept(R', S')
14:    S' ← S' ∪ {a_i}
15: end for
```

Algorithm 4 **greedy**$_\delta(U, R, S)$

```
1: W ← U
2: for a_i ∈ S ∩ U order by rk_frontier(T) do
3:    Ŵ ← W \ {a_i}
4:    if e_acc ≤ lberr(R, S ∩ Ŵ) + δ then
5:       W ← Ŵ
6:    end if
7: end for
8: return W
```

estimated as the empirical misclassification error on the search set, and it is computed using the C4.5's distribution imputation method[3].

Algorithm 3 builds on the procedure for the enumeration of distinct decision trees by implementing a further pruning of the search space. In particular, a call **DTaccept**$_\delta(R, S)$ searches for a δ-acceptable feature subset S_{acc} among all subsets in $R \bowtie Pow(S)$. The global variable e_{acc} stores the best error estimate found so far, and it is initialized outside the call to ∞. The structure of **DTaccept**$_\delta$ follows the one of Algorithm 2, from which it differs in two main points.

The first difference regards lines 4-8, which instead of just outputting the feature subset, they update the best error estimation found so far in case the estimated error of T is lower or equal[4] than it. The set of features S_{acc} is also updated.

3. Predictions of instances with no missing value follows a path from the decision tree root to a leaf node. For instances with missing value of the attribute tested at a decision node, several options are available (Saar-Tsechansky and Provost, 2007; Twala, 2009). In C4.5, all branches of the decision node are followed, and the prediction of a leaf in a branch contributes in proportion to the weight of the branch's child node (fraction of cases reaching the decision node that satisfy the test outcome of the child). The class value predicted for the instance is the one with the largest total contribution.
4. We break ties in favor of smaller feature subsets.

The second difference regards the set U of used features which, at line 8, is possibly pruned by the call $\mathbf{greedy}_\delta(U, R, S)$. Such a function tries and relax the pruning of features subsets in formula (2) in order to include additional attributes. Let $S = \{a_1, \ldots, a_n\}$. In particular, we aim at finding a minimal set $W = \{a_1, \ldots, a_k\} \subseteq S \cap U$ such that:

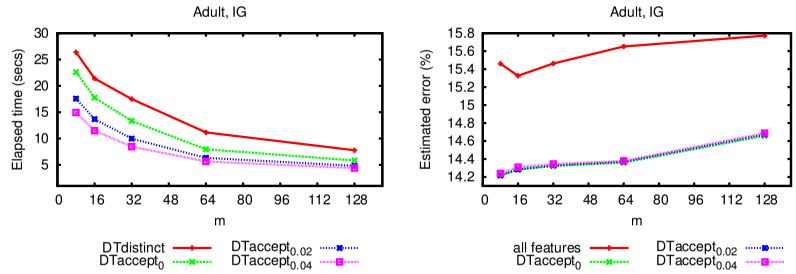
$$e_{acc} \leq err(DT(V)) + \delta \quad (3)$$
$$\text{for all } V \in \bigcup_{i=n,\ldots,k+1} (R \cup \{a_1, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\}).$$

If such a condition holds[5], we can prune from the search space the feature subsets in the quantifier of the condition, because even in the case that a feature subset with the global best error is pruned this way, the best error found so far e_{acc} is within the δ bound from the error of the pruned decision tree. Practically, this means that we can continue using W in the place of U in the rest of the search – and, in fact, lines 9–15 of $\mathbf{DTaccept}_\delta$ coincide with lines 6–12 of **DTdistinct**. The search space is pruned if $W \subset U$, because the loop at lines 11-15 iterates over a smaller set of features. The approach that $\mathbf{greedy}_\delta$ adopts for determining $\{a_1, \ldots, a_k\}$ is a greedy one, which tries and removes one candidate feature from $S \cap U$ at a time. The order of removal is by[6] $rk_{frontier}(T)$ as in the main loop of **DTdistinct**. The function $\mathbf{greedy}_\delta$ relies on a lower-bound function $lberr()$ for which the test condition[7] $e_{acc} \leq lberr(R, \{a_1, \ldots, a_k\}) + \delta$ is required to imply that (3) holds. This is obviously true when:

$$lberr(R, \{a_1, \ldots, a_k\}) \leq err(DT(V)) \quad (4)$$
$$\text{for all } V \in \bigcup_{i=n,\ldots,k+1} (R \cup \{a_1, \ldots, a_{i-1}\}) \bowtie Pow(\{a_{i+1}, \ldots, a_n\}).$$

hence, the adjective "lower-bound" function for $lberr()$. Our lower-bound function is defined as follows for a candidate $\{a_1, \ldots, a_k\}$. We start from the decision tree $T = DT(R \cup S) = DT(R \cup (S \cap U))$ and remove sub-trees in T rooted at frontier nodes with split features in $(S \cap U) \setminus \{a_1, \ldots, a_k\}$. Let T' be the partial tree obtained and call the removed frontier nodes the "to-be-expanded" nodes. The features used in T' belong to $R \cup \{a_1, \ldots, a_k\}$, which is included in any V quantified over in (4). Due to the Assumptions 4-5, any tree $DT(V)$ may differ from T only at the nodes to-be-expanded and their sub-trees, i.e., T' is a sub-tree of $DT(V)$. At the best, the error of the sub-trees in V that expand T' will be zero[8]. Thus, we have $err(T') \leq err(DT(V))$, where $err(T')$ adds 0 as estimated error at nodes to-be-expanded. Since T' is defined only starting from T, and not from any specific

5. A direct extension of (2) is to require $err(DT(R \cup S)) \leq err(DT(V)) + \delta$, i.e., that the error of the last built tree is within the δ bound from the error of any tree over quantified attributes V. The relaxed condition (3) allows for a better pruning, namely that the best error found so far is within the δ bound.
6. The rationale is to try and remove first features that lead to minimal changes in the decision tree, and, consequently, to minimal differences in its misclassification error. This is a direct generalization of the approach of **DTdistinct** of removing unused features, which lead to no change in misclassification.
7. Cf. line 4 of Algorithm 4, where $\{a_1, \ldots, a_k\}$ is $S \cap \hat{W}$.
8. Consider the wrapper model, namely error is estimated as the empirical misclassification error over a search set. Since decision tree error is always not lower than the Bayes error (Devroye et al., 1996), a better lower bound can be obtained at each node to-be-expanded as: $1 - \frac{1}{|I|} \sum_{\mathbf{x} \in I} g(\mathbf{x})$. Here, I includes the instances in the search set reaching the node to-be-expanded, and $g(\mathbf{x}) = max_c\ k^c_{\mathbf{x}}/n_{\mathbf{x}}$, where $n_{\mathbf{x}}$ is the number of instances in I whose predictive attribute values are the same as for $\mathbf{x}$, and $k^c_{\mathbf{x}}$ is the number of such instances that also have class value equal to c. Unfortunately, the computational cost of calculating this lower bound function makes it impractical.



(a) Elapsed running time. (b) Misclassification error on search set.

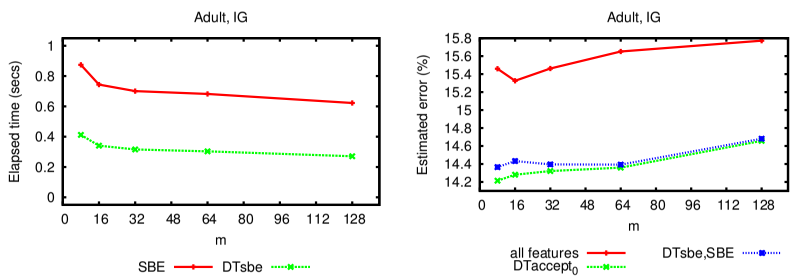
Figure 6: $\mathbf{DTaccept}_\delta$ elapsed times and estimated errors on the Adult dataset.

V, we then define $lberr(R, \{a_1, \ldots, a_k\}) = err(T')$ and have that (4) holds. In summary, we have the following result.

Theorem 11 *Let F be a set of features. $\mathbf{DTaccept}_\delta(R, S)$ finds a δ-acceptable feature subset among all subsets in $R \bowtie Pow(S)$, if it exists. In particular, $\mathbf{DTaccept}_\delta(\emptyset, F)$ finds a δ-acceptable feature subset.*

Example 9 *Consider again the sample decision tree T at Figure 3(a). Assume $R = \emptyset$ and $S = S \cap U = \{a_1, a_2, a_3, a_4\}$. Also, let e_{acc} the best error estimate found so far. $\mathbf{greedy}_\delta$ will consider removing attributes in the order provided by $rk_{frontier}(T)$, which is a_4, a_2, a_3, a_1 (see Example 4). At the first step, the sub-tree rooted at a_4 is tentatively removed, and replaced with an oracle with zero error estimate, as shown in Figure 3(b). If the error $lb = lberr(\emptyset, \{a_4\})$ of such a tree is such that $e_{acc} \leq lb + \delta$, we can commit the removal of a_4. Assume this is the case. In the next step, the sub-tree rooted at a_2 is also tentatively removed and replaced with an oracle. Again, if the estimated error $lb = lberr(\emptyset, \{a_4, a_2\})$ of such a tree is such that $e_{acc} \leq lb + \delta$, we can commit the removal of a_2. Assume this is not the case, and a_2 is not removed. In the third step, the sub-tree rooted at a_3 is tentatively removed and replaced with an oracle, as shown in Figure 3(c). If the estimated error $lb = lberr(\emptyset, \{a_4, a_3\})$ of such a tree is such that $e_{acc} \leq lb + \delta$, we can commit the removal of a_3. Assume this is the case. In the last step, we try and remove the sub-tree rooted at a_1, and replace it with an oracle. The estimated error of such a tree is $lb = lberr(\emptyset, \{a_4, a_3, a_1\}) = 0$. Condition $e_{acc} \leq lb + \delta$ does not hold (otherwise, it would have been satisfied at the second step as well). In summary, $\mathbf{greedy}_\delta$ returns $\{a_2, a_1\}$, whilst $\{a_4, a_3\}$ can be safely not iterated over at step 11 of Algorithm 3.*

Example 10 *Reconsider Example 5. Figure 6(a) shows the elapsed running times of $\mathbf{DTaccept}_\delta(\emptyset, F)$ where F is the set of all features of the Adult dataset. Results are shown for several values of the parameter δ. It is worth noting that, for $\delta = 0$, the elapsed time is smaller than the enumeration of distinct trees by **DTdistinct**. In fact, when we*



(a) Elapsed running time. (b) Misclassification error on search set.

Figure 7: **SBE** and **DTsbe** elapsed times and estimated errors on the Adult dataset.

*search for a best feature subset, **DTaccept**$_0$ prunes from the search space those (distinct) decision trees for which the lower bound on the estimated error is higher than the best error estimation found during the search. Figure 6(b) shows the estimated error (misclassification error on the search set) of the decision tree built on features returned by **DTaccept**$_\delta$ and on all features F. The difference between the estimated error of $DT(F)$ and the estimated error of a best feature subset ($\delta = 0$) provides the range of error estimations that may be returned by heuristic feature selection approaches adhering to the wrapper model. Notice that the estimated error of δ-acceptable feature subsets for $\delta = 0.02$ and $\delta = 0.04$ is very close to the estimated error of a best feature subset (only +2% and +4% respectively).*

The example shows that the pruning strategy of **DTaccept**$_\delta$ is effective in the specific case of the Adult dataset. In the worst case, however, the search space remains the one of distinct decision trees, which, for low m values, is exponential in the number of features. In Section 7, we will test performances on datasets of larger dimensionalities.

As a final note, we observe that our approach can be easily adapted to other variants of the feature selection problem. One variant consists of regularizing the error estimation with a penalty ϵ for every feature in a subset. In such a case, the only changes in Algorithm 3 would be: the test at line 4 becomes $err(T) + |U| \cdot \epsilon \leq e_{acc}$; the assignment at line 5 becomes $e_{acc} \leftarrow err(T) + |U| \cdot \epsilon$. Regarding the lower bound function $lberr(R, S \cap \hat{W})$ called at line 4 of **greedy**$_\delta$, by adding the penalty $|S \cap \hat{W}| \cdot \epsilon$ we obtain a lower bound on the regularized error estimation of trees built from subsets V pruned in (4).

6. A White-Box Optimization of SBE

On the practical side, **DTaccept**$_\delta$ does not scale to large dimensional datasets. Moreover, acceptability/best error on the search set may be obtained at the cost of overfitting (Doak, 1992; Reunanen, 2003) and instability (Nogueira and Brown, 2016). We will discuss these issues in the experimental section. The ideas underlying our approach, however, can impact also on the efficiency of well-performing heuristic searches. In particular, we consider here the cornerstone sequential backward elimination (**SBE**) heuristic. It starts building a classifier T using the set S of all features, i.e., $S = F$. For every $a \in S$, a classifier T_a is built using features in $S \setminus \{a\}$. If no T_a's has a lower or equal estimated error than T, the algorithm stops returning S as the subset of selected features. Otherwise, the procedure is repeated removing $\hat{a}$ from S, where $T_{\hat{a}}$ is the classifier with the smallest estimated error. In summary, features are eliminated one at a time in a greedy way. **SBE** is a black-box approach. The procedure applies to any type of classifier. A white-box optimization can be devised for decision tree classifiers that satisfy the assumptions of Section 4.1. Let $U = features(T)$ be the set of features used in the current decision tree $T = DT(S)$. By Lemma 6, for a non-used feature $a \in S \setminus U$, it turns out that $T_a = DT(S \setminus \{a\}) = DT(S) = T$. Thus, only trees T_a for $a \in U$ need to be built and evaluated at each step, saving the construction of $|S \setminus U|$ decision trees. The following other pruning can be devised. Let $\hat{b}$ be the feature removed at the current step, and $a \in U = features(T)$ such that $\hat{b} \notin features(T_a)$, where $features(T_a)$ is known from the previous step. By Lemma 6, it turns out that $(T_{\hat{b}})_a = DT(S \setminus \{\hat{b}, a\}) = DT(S \setminus \{a\}) = T_a$. In summary, $(T_{\hat{b}})_a$ can be pruned if $a \notin features(T_{\hat{b}})$ or if $\hat{b} \notin features(T_a)$. The optimizations of incremental tree building and parallelization discussed for **DTdistinct** readily apply for such heuristic as well. We call the resulting white-box optimized algorithm **DTsbe**.

Example 11 *Figure 7(a) contrasts the elapsed running times of **SBE** and **DTsbe** on the Adult dataset. The efficiency improvement of **DTsbe** over **SBE** is consistently in the order of 2×. Figure 7(b) shows instead their estimated error (misclassification error on the search set), which are obviously the same. Estimates are very close to the estimated error of a best feature subset provided by **DTaccept**$_0$.*

7. Experiments

7.1. Datasets and Experimental Settings

We perform experiments on 20 small and large dimensional benchmark datasets publicly available from the UCI ML repository (Lichman, 2013). Some of the datasets have been used in the NIPS 2003 challenge on feature selection, and are described in detail by Guyon et al. (2006a). Table 1 reports the number of instances and features of the datasets.

Following (Kohavi, 1995; Reunanen, 2003), the generalization error of a classifier is estimated by repeated stratified 10-fold cross validation[9]. Cross-validation is repeated 10 times. At each repetition, the available dataset is split into 10 folds, using stratified random sampling. Each fold is used to compute the misclassification error of the classifier built on the remaining 9 folds used as training set for building classification models. The generalization error is then the average misclassification error over the 100 classification models (10 models times 10 repetitions). The following classification models will be considered:

baseline: the decision tree $DT(F)$ built on all available features F;

DTsbe and SBE: the decision tree built on the feature subset selected by **DTsbe** (or equivalently by **SBE**);

9. Cross-validation is a nearly unbiased estimator (Kohavi, 1995), yet highly variable for small datasets. Kohavi's recommendation is to adopt a stratified version of it. Variability of the estimator is accounted for by adopting repetitions (Kim, 2009).

DTaccept$_0$: the decision tree built on the feature subset selected by **DTaccept$_0$** (i.e, a best feature subset);

RF: a random forest[10] of 100 decision trees;

DTsbe+RF: a random forest of 100 decision trees, where the available set of features is restricted to those selected by **DTsbe**.

Random forests are included in experiments in order to relate results to state-of-the-art classification models. With reference to the feature selection methods (**DTsbe**, **SBE** and **DTaccept$_0$**), the error estimator used adheres to the wrapper model. The training set is split into 70% building set and 30% search set using stratified random sampling. The building set is used for constructing decision trees over subsets of features, and the search set for estimating their error using empirical misclassification. The estimated (search) error of a feature selection strategy is the average misclassification error on the 100 search sets of the decision tree built on the feature subset selected by the strategy. Thus, best feature subsets are those with minimum estimated (search) error, but not necessarily with minimum generalization (cross-validation) error. Different feature selection strategies are applied to the same triples of building, search, and test sets. Hence, paired t-tests can be adopted to evaluate statistically significant differences of their means.

Information Gain (IG) is used as quality measure in node splitting during tree construction. No form of tree simplification (Breslow and Aha, 1997) is considered in this section. Appendix A reports additional results about the impact of C4.5 error-based pruning and the Gain Ratio quality measure.

All procedures described in this paper are implemented by extending the YaDT system (Ruggieri, 2002, 2004; Aldinucci et al., 2014). It is a state-of-the-art main-memory C++ implementation of C4.5 with many algorithmic and data structure optimizations as well as with multi-core data parallelism in tree building. The extended YaDT version is publicly available from the author's home page[11]. Tests were performed on a PC with Intel 8 cores i7-6900K at 3.7 GHz, without hyperthreading, 16 Gb RAM, and Windows Server 2016 OS.

7.2. On the Search Space Visit of DTdistinct

For the Adult dataset, the order $rk_{frontier}(T)$ used in **DTdistinct** was impressively effective in producing a visit of the feature subset space with a negligible fraction of duplicates (see Figure 4(b)). Figure 8(a) shows the ratio of the number of built trees over the number of distinct trees for other small to medium dimensionality datasets – namely, those for which **DTdistinct** terminates within a time-out of 1h. The ratios are below 1.00035, i.e., there are at most 0.035% duplicate trees built by **DTdistinct**.

7.3. Elapsed Running Times

Table 1 compares the elapsed running times of **SBE** and its computational optimization **DTsbe**. For all datasets, the tree building stopping parameter m is set to the small value 2, which is the default for C4.5. The ratio of elapsed times shows a speedup in the range

10. At each decision node, only a random subset of attributes are considered for splitting the node. The number of attributes in the subset is logaritmic in the number of available features.
11. http://pages.di.unipi.it/ruggieri

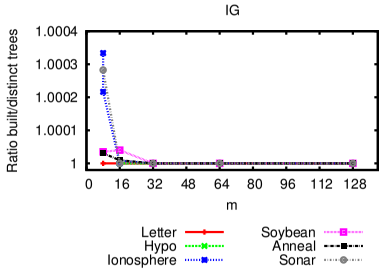
Table 1: Experimental datasets, and average elapsed running times (seconds). IG and m=2.

dataset	inst.	feat.	DTsbe	SBE	ratio	DTaccept$_0$	RF
Adult	48,842	14	0.63	1.24	0.510	32.42	1.94
Letter	20,000	16	0.39	0.69	0.567	231.64	0.87
Hypo	3,772	29	0.06	0.21	0.283	56.12	0.05
Ionosphere	351	34	0.01	0.09	0.109	0.74	0.01
Soybean	683	35	0.06	0.20	0.281	1843.12	0.03
Kr-vs-kp	3196	36	0.06	0.23	0.272	>1h	0.07
Anneal	898	38	0.01	0.08	0.152	18.27	0.01
Census	299,285	40	35.02	117.85	0.297	>1h	18.64
Spambase	4601	57	0.69	4.84	0.144	>1h	0.13
Sonar	208	60	0.01	0.40	0.031	18.77	0.01
Optdigits	3823	64	0.47	4.61	0.103	>1h	0.14
Coil2000	9,822	85	2.13	23.74	0.090	>1h	0.28
Clean1	476	166	0.13	20.43	0.006	>1h	0.03
Clean2	6,598	166	3.91	151.94	0.026	>1h	0.22
Madelon	2,600	500	9.79	>1h	-	>1h	0.27
Isolet	7,797	617	226.75	>1h	-	>1h	2.09
Gisette	7,000	5,000	87.40	>1h	-	>1h	11.69
P53mutants	31,420	5,408	616.67	>1h	-	>1h	9.58
Arcene	900	10,000	25.59	>1h	-	>1h	4.33
Dexter	2,600	20,000	1060.91	>1h	-	>1h	0.18

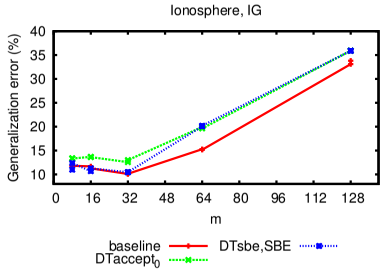
2× - 100×. For large dimensional datasets, the black-box **SBE** does not even terminate within a time-out of 1h on the first fold of the first iteration of repeated cross-validation. This is a relevant result for machine learning practitioners, extending the applicability of the **SBE** heuristic, a key reference in feature selection strategies.

Table 1 also reports the elapsed times for **DTaccept$_0$**. It makes it possibile to search for best feature subsets in a reasonable amount of time for datasets with up to 60 features. In general, the elapsed time depends on: (1) the size of the search space, which is bounded by the number of distinct decision trees; and on: (2) the size of the dataset and the stopping parameter m, which affect the time to build single decision trees. In turn, (1) depends on the number of available features and on (2) itself – in fact, decision trees for small datasets or for large m's values are necessarily small in size. In summary, efficiency of **DTaccept$_0$** can be achieved for datasets with limited number of features *and* (limited number of instances *or* large values of the stopping parameter m). For example, time-out bounds in Table 1 are reached for medium-to-large number of features (>60) or for medium-to-large number of instances (e.g., the *Kr-vs-kp*, *Census*, and *Spambase* datasets). Notice that while large m's values can speed up the search, they negatively affect the predictive accuracy of decision trees (see next subsection).

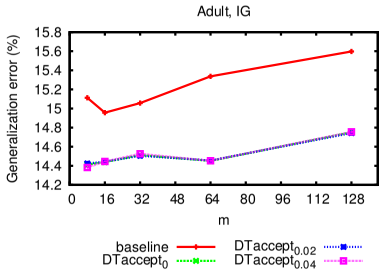
Finally, Table 1 also includes the elapsed times of building random forests. They are very low compared to all the other times. In fact, the number of trees in a forest (set to 100) is much smaller than the number of trees built by sequential backward elimination (quadratic in the number of features, in the worst case) and by complete search (exponential, in the worst case).



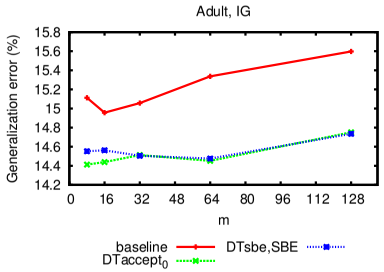
(a) Ratio built/distinct trees in **DTdistinct**.



(b) Cross-validation error.



(c) Cross-validation errors.



(d) Cross-validation error.

Figure 8: Procedure performances.

7.4. Estimated Errors and Generalization Errors

Best feature subsets found by complete search minimize the estimated error (misclassification on the search set), but there is no guarantee that this extends to unseen cases, i.e., to the generalization error evaluated with cross-validation.

Figures 8(c) and 8(d) show generalization errors on the Adult dataset for decision trees constructed on all available features (baseline), and on features selected by **DTaccept**$_0$ (best feature subset), **DTaccept**$_{0.02}$ and **DTaccept**$_{0.04}$ (0.02 and 0.04-acceptable feature subsets), and **SBE** (same as **DTsbe**). For such a dataset, the best feature subsets have the best generalization error, the 0.02- and 0.04-acceptable feature subsets are very close to it, and **SBE** has a similar performance except for low m values.

Figures 8(b) highlights a different result. For the Ionosphere dataset and large m values, the best generalization error is for the baseline, then for **SBE**, and finally for the best feature subsets. Ionosphere is a small dataset, hence using all instances for training (particularly for large m values) results in a better strategy than splitting training into building and search sets for feature selection.

Table 2: Estimated (search) errors and generalization (cross-validation) errors (mean ± stdev). IG and m=2. Best method in **bold**. "⋆" labels methods not different from the best one at 95% significance level.

	estimated (search) error			generalization (cross-val.) error		
dataset	**baseline**	**DTsbe**	**DTaccept**$_0$	**baseline**	**DTsbe**	**DTaccept**$_0$
Adult	16.13 ± 0.29	14.35 ± 0.29	**14.19 ± 0.21**	15.76 ± 0.43	14.45 ± 0.45	**14.30 ± 0.42**
Letter	14.82 ± 0.50	13.98 ± 0.47	**13.87 ± 0.44**	12.46 ± 0.76⋆	**12.38 ± 0.71**	12.38 ± 0.76⋆
Hypo	0.55 ± 0.26	0.32 ± 0.17	**0.29 ± 0.17**	**0.42 ± 0.32**	0.52 ± 0.38	0.57 ± 0.43
Ionosphere	11.47 ± 2.82	5.78 ± 1.86	**2.12 ± 1.14**	11.40 ± 5.69	**9.77 ± 4.67**	11.29 ± 5.61
Soybean	16.38 ± 2.68	7.78 ± 2.11	**5.75 ± 1.39**	13.12 ± 4.30	13.15 ± 4.00	**11.53 ± 4.15**
Kr-vs-kp	1.10 ± 0.40	**0.64 ± 0.28**	-	**0.55 ± 0.44**	1.17 ± 0.61	-
Anneal	1.39 ± 0.86	0.84 ± 0.53	**0.60 ± 0.45**	**0.99 ± 0.92**	1.61 ± 1.34	2.14 ± 1.86
Census	6.16 ± 0.07	**4.97 ± 0.13**	-	6.03 ± 0.12	**4.98 ± 0.16**	-
Spambase	8.14 ± 0.83	**5.58 ± 0.61**	-	7.46 ± 1.20⋆	**7.29 ± 1.21**	-
Sonar	28.02 ± 6.72	14.28 ± 3.77	**2.12 ± 1.30**	26.40 ± 9.51⋆	**25.84 ± 9.05**	26.40 ± 9.40⋆
Optdigits	11.11 ± 0.95	**7.79 ± 0.67**	-	**9.88 ± 1.22**	10.27 ± 1.66	-
Coil2000	9.32 ± 0.49	**7.16 ± 0.33**	-	9.03 ± 0.68	**8.76 ± 0.72**	-
Clean1	22.22 ± 3.90	**10.53 ± 2.33**	-	**17.38 ± 6.76**	19.45 ± 6.96	-
Clean2	4.21 ± 0.49	**1.54 ± 0.37**	-	3.31 ± 0.67	**2.80 ± 0.71**	-
Madelon	29.76 ± 3.74	**15.67 ± 2.17**	-	25.91 ± 3.53	**22.51 ± 3.20**	-
Isolet	18.65 ± 0.86	**11.95 ± 0.66**	-	17.10 ± 1.23	**16.80 ± 1.39**	-
Gisette	6.65 ± 0.62	**3.12 ± 0.43**	-	6.14 ± 0.85	**5.91 ± 0.94**	-
P53mutants	0.62 ± 0.06	**0.31 ± 0.04**	-	0.58 ± 0.10	**0.51 ± 0.11**	-
Arcene	48.75 ± 3.17	**30.83 ± 2.58**	-	**48.04 ± 5.46**	48.21 ± 5.08⋆	-
Dexter	48.37 ± 1.69	**31.82 ± 1.38**	-	48.88 ± 2.94⋆	**48.20 ± 3.47**	-
wins/⋆	0/0	13/0	7/0	6/4	12/1	2/2

Table 2 reports the estimated errors and the generalization errors for all datasets. Here and in the following tables, the best method is shown in bold. Other methods are labelled with "⋆" if the null hypothesis in a paired t-test with the best method at 95% significance level cannot be rejected. Three conclusions can be made from the table.

First, heuristic search performs well in comparison to complete search as far as estimated error is concerned with. In fact, the estimated error of **DTsbe** (or equivalently, of **SBE**) is close to the estimated error of best feature subsets in 3 cases out of 7, and is always better than the baseline. Only for Ionosphere and Sonar, it departs from the estimated error of the best feature subsets.

Second, heuristic search generalizes to unseen cases better than baseline and complete search. Globally, **DTsbe** wins in 12 cases and it is not statistically different from the winner in another case. The larger the dimensionality of datasets, the more clear is the advantage of **DTsbe** over the baseline. Comparing **DTsbe** and **DTaccept**$_0$ between them, **DTsbe** wins in 2 cases (Ionosphere and Anneal), loses in 2 cases, and it is not statistically different in 3 cases (we count here Hypo, where the baseline wins). Thus we cannot conclude that complete search leads to better generalization errors than heuristic search.

Third, the difference between estimated and generalization error is greater for heuristic search than for the baseline, and for complete search compared to heuristic search. In the

Table 3: Generalization (cross-validation) errors (mean ± stdev). IG and m=2.

dataset	generalization (cross-val.) error RF	DTsbe+RF
Adult	**14.21 ± 0.43**	14.29 ± 0.43
Letter	**4.04 ± 0.46**	4.66 ± 0.63
Hypo	0.70 ± 0.39	**0.48 ± 0.36**
Ionosphere	**6.50 ± 4.01**	8.38 ± 4.35
Soybean	**7.37 ± 2.76**	10.95 ± 3.58
Kr-vs-kp	1.35 ± 0.67⋆	**1.26 ± 0.66**
Anneal	**0.94 ± 0.88**	1.69 ± 1.38
Census	**4.85 ± 0.07**	4.92 ± 0.12
Spambase	**5.63 ± 1.12**	5.79 ± 1.05
Sonar	**18.42 ± 8.62**	23.26 ± 8.90
Optdigits	**2.03 ± 0.64**	2.76 ± 0.88
Coil2000	**6.01 ± 0.12**	6.11 ± 0.20
Clean1	**10.84 ± 4.62**	15.24 ± 5.68
Clean2	2.46 ± 0.56⋆	**2.41 ± 0.54**
Madelon	36.25 ± 3.06	**20.93 ± 2.87**
Isolet	**6.34 ± 0.86**	6.46 ± 0.84⋆
Gisette	4.17 ± 0.78	**3.80 ± 0.71**
P53mutants	**0.45 ± 0.04**	0.45 ± 0.06⋆
Arcene	45.47 ± 2.61	**44.59 ± 2.59**
Dexter	48.95 ± 2.07	**43.59 ± 2.88**
wins/⋆	13/2	7/2

case of Sonar, for instance, the differences are considerably large. This implies that oversearching may incur in feature subsets that overfit the data, thus reinforcing the conclusions of Doak (1992); Quinlan and Cameron-Jones (1995); Reunanen (2003) that oversearching increases overfitting.

7.5. Comparison with Random Forests

Random forests are a state-of-the-art classification model that is important to relate with. Single decision trees may be preferrable when interpretability of the model is a requirement that can be traded-off with a lower generalization error. How much lower? A first natural question is then how complete and heuristic searches perform compared to random forests. Table 3 reports the generalization errors of random forest models (**RF**). Contrasting them with the generalization errors in Table 2, random forests perform worse for Hypo and Madelon only. In all other cases, they perform better, often considerably better.

A second question is whether the coupling of heuristic search with random forests may enhance the performance of the random forests alone. Table 3 shows also the generalization errors of **DTsbe+RF**. Recalling that datasets are listed in order of increasing dimensionality (see Table 1), it is immediate to note that for large dimensionality datasets, there is an advantage in doing feature selection before random forests. In particular, we correlate the dataset dimensionality with the difference of mean generalization errors normalized by standard deviation of **RF**. The rank correlation coefficient is $\rho = 0.49$, and the Spearman's test rejects the null hypotesis of zero correlation at 95% confidence level. Intuitively, ran-

Table 4: Number of features and sample Pearson's correlation coefficient in cross-validation (mean ± stdev). IG and m=2.

dataset	number of features baseline	DTsbe	DTaccept$_0$	$\phi_{Pearson}$ baseline	DTsbe	DTaccept$_0$
Adult	13.92 ± 0.27	6.37 ± 1.81	**5.66 ± 1.18**	**0.85 ± 0.36**	0.45 ± 0.27	0.55 ± 0.22
Letter	16.00 ± 0.00	11.35 ± 1.47	**10.21 ± 1.02**	**1.00 ± 0.00**	0.53 ± 0.25	0.70 ± 0.17
Hypo	15.57 ± 2.02	**6.64 ± 1.23**	8.85 ± 3.05	**0.78 ± 0.16**	0.76 ± 0.13	0.51 ± 0.22
Ionosphere	11.44 ± 1.54	**5.35 ± 1.64**	7.20 ± 1.63	**0.43 ± 0.16**	0.33 ± 0.21	0.05 ± 0.19
Soybean	27.52 ± 0.95	**15.28 ± 2.02**	17.12 ± 2.67	**0.87 ± 0.10**	0.29 ± 0.19	0.27 ± 0.18
Kr-vs-kp	28.88 ± 1.07	**19.56 ± 2.10**	-	**0.84 ± 0.11**	0.63 ± 0.14	-
Anneal	10.13 ± 0.68	**6.65 ± 0.95**	10.26 ± 2.38	**0.95 ± 0.06**	0.67 ± 0.20	0.36 ± 0.20
Census	37.94 ± 0.28	**8.97 ± 3.62**	-	**0.97 ± 0.08**	0.50 ± 0.17	-
Spambase	48.89 ± 2.45	**30.98 ± 4.05**	-	**0.53 ± 0.15**	0.35 ± 0.12	-
Sonar	13.32 ± 1.79	**6.78 ± 1.23**	9.11 ± 1.38	**0.25 ± 0.16**	0.15 ± 0.19	0.03 ± 0.15
Optdigits	46.40 ± 1.68	**28.13 ± 2.68**	-	**0.75 ± 0.08**	0.38 ± 0.11	-
Coil2000	62.63 ± 2.14	**39.92 ± 4.65**	-	**0.77 ± 0.07**	0.43 ± 0.10	-
Clean1	28.53 ± 2.46	**14.78 ± 2.29**	-	**0.31 ± 0.13**	0.17 ± 0.11	-
Clean2	79.03 ± 4.67	**37.72 ± 4.67**	-	**0.31 ± 0.10**	0.20 ± 0.09	-
Madelon	113.99 ± 9.87	**49.25 ± 5.91**	-	**0.23 ± 0.05**	0.20 ± 0.06	-
Isolet	240.40 ± 7.61	**105.26 ± 7.71**	-	**0.36 ± 0.04**	0.27 ± 0.05	-
Gisette	129.40 ± 3.95	**56.62 ± 5.01**	-	**0.24 ± 0.04**	0.19 ± 0.06	-
P53mutants	73.73 ± 4.27	**22.05 ± 4.23**	-	**0.20 ± 0.05**	0.09 ± 0.06	-
Arcene	69.19 ± 2.21	**35.02 ± 4.57**	-	**0.05 ± 0.03**	0.02 ± 0.03	-
Dexter	284.85 ± 8.40	**126.10 ± 9.84**	-	**0.38 ± 0.03**	0.28 ± 0.04	-
wins/⋆	0/0	18/0	2/0	20/0	0/0	0/0

dom forests are more robust to large dimensionality effects than single decision trees due to the (random) selection of a logarithmic number of features to be considered in splits at decision nodes. However, when the number of available features increases, the logarithmic reduction is not sufficient anymore. It is worth noting that such an experimental analysis can only be made due to the efficiency improvements of **DTsbe** over **SBE**. In fact, from Table 1, we know that **SBE** does not terminate within a reasonable time-out.

7.6. On Feature Reduction and Stability

Feature selection is typically a multi-objective problem. In addition to the selection of a feature subset with minimum error, one may be interested in other performance measures, e.g., in minimizing the size of the subset and in minimizing the variability due to perturbations in the training set (stability). The two objectives are contrasting between them, since one can reach low variability by always using all available features. Table 4 reports on the left hand side the mean and standard deviation of the number of features actually used by the decision trees built during cross-validation. The baseline method shows that the embedded feature selection used in tree construction uses a restricted number of features w.r.t. the available ones. **DTsbe** leads to using half or even fewer features. It is the best performing selection strategy with regard to such a measure. **DTaccept$_0$** wins in 2 datasets

Table 5: Elapsed running times (seconds) and generalization (cross-validation) errors (mean ± stdev). IG, m=2, max depth = 5, and misclassification on training set as error estimator in **DTsbe**. (1) one-hot encoding of discrete features required by OCT.

	elapsed time		generalization error		
dataset	**DTsbe**	**OCT**	**baseline**	**DTsbe**	**OCT**
Adult(1)	0.34	5.53	14.91 ± 0.45⋆	**14.87 ± 0.45**	15.2 ± 0.24
Letter	0.03	0.68	49.85 ± 0.92	**49.44 ± 0.92**	55.3 ± 1.61
Hypo(1)	0.01	0.20	**0.81 ± 0.41**	0.83 ± 0.44⋆	1.72 ± 0.72
Ionosphere	0.01	0.09	12.32 ± 4.64	**9.58 ± 4.60**	19.25 ± 5.93
Soybean(1)	0.03	0.19	17.88 ± 3.88	**10.83 ± 3.00**	35.82 ± 5.26
Kr-vs-kp(1)	0.02	0.17	5.91 ± 1.18	6.10 ± 1.22	**5.8 ± 0.71**
Anneal(1)	0.01	0.33	**1.50 ± 1.51**	1.74 ± 1.42⋆	4.9 ± 2.0
Census(1)	10.27	88.15	5.41 ± 0.06	**5.28 ± 0.08**	5.36 ± 0.05
Spambase	0.05	0.38	9.35 ± 1.30⋆	**9.32 ± 1.29**	11.58 ± 1.15
Sonar	0.02	0.09	26.29 ± 9.52⋆	**25.91 ± 9.86**	37.36 ± 6.95
Optdigits	0.07	0.07	20.03 ± 1.97	**18.64 ± 2.07**	66.32 ± 1.47
Coil2000	0.15	0.60	**6.01 ± 0.13**	6.11 ± 0.19	7.64 ± 0.5
Clean1	0.04	0.17	24.85 ± 6.17	**21.51 ± 6.99**	36.97 ± 3.61
Clean2	0.30	1.16	6.01 ± 1.16	**5.19 ± 0.72**	8.81 ± 1.04
Madelon	0.44	1.84	23.92 ± 3.83	**21.23 ± 2.84**	44.24 ± 3.71
Isolet	11.33	21.79	35.40 ± 1.50	**34.01 ± 1.37**	38.78 ± 1.56
Gisette	5.54	32.26	6.61 ± 0.91	**6.24 ± 0.96**	9.06 ± 0.74
P53mutants	150.99	142.69	0.47 ± 0.09⋆	**0.47 ± 0.09**	0.75 ± 0.11
Arcene	3.36	7.47	47.36 ± 5.55⋆	**47.22 ± 4.89**	48.89 ± 1.83
Dexter	3.02	25.53	46.02 ± 2.51⋆	**45.86 ± 2.59**	48.58 ± 1.41
wins/⋆			3/6	16/2	1/0

out of 7. While the differences with **DTsbe** are statistically significant, they are not large – about 1-2 features.

Table 4 reports on the right hand side the mean and standard deviation of the sample Pearson's correlation coefficient over feature subsets used by decision trees[12] built during cross-validation. The mean value corresponds to the $\hat{\boldsymbol{\Phi}}_{\boldsymbol{Pearson}}$ measure of stability introduced by Nogueira and Brown (2016). Differently from other measures of stability, it is unbiased w.r.t. dimensionality of the dataset. From the table, we can conclude that the baseline method has the highest stability of selected features (also with the smallest variance), where the value 1 means that any two distinct folds in cross-validation produce decision trees that use the same subset of features. However, such a stability is obtained at the expenses of a greater number of used features. Feature selection strategies have a considerably lower stability, in some cases half of the values of the baseline. **DTsbe** wins over $\mathbf{DTaccept}_0$ in 4 cases out of 7 and is equivalent in another. $\mathbf{DTaccept}_0$ has a better stability only for the 2 lower dimensional datasets. In summary, we can conclude that, in the case of decision tree classifiers, oversearching increases instability as well.

12. Stability is calculated on the subset of features *used* by decision trees, not on the feature subsets *selected* by a strategy. This allows for measuring variability of the baseline approach, which otherwise would result to have zero variability, and to contrast it to the feature selection strategies.

7.7. Comparison with Optimal Decision Trees

The non-greedy OCT approach by Bertsimas and Dunn (2017) transforms decision tree learning into a mixed-integer optimization problem. The objective function to minimize is the empirical misclassification error on the training set regularized by a complexity penalty cp on tree size. Such a decision tree is optimal over all possible ways of setting the split attribute at all nodes. An optimal decision tree may not be produceable by greedy top-down algorithms for any feature subset. In this section, we compare OCT[13] with **DTsbe**. For uniformity of comparison, we set the error estimator in **DTsbe** to the misclassification error on the training set (instead of adhering to the wrapper model). In both cases, we set as stopping criterion a maximum tree depth of 5. This is useful for two reasons. First, decision trees are small and easily interpretable, which is the main advantage of using single decision trees over the more powerful model of random forests. Second, exploring the search space of optimal tree search becomes feasible.

Table 5 shows the elapsed times and generalization errors of the two approaches. It also reports the baseline generalization error of depth-bounded C4.5 decision tree. The elapsed times of **DTsbe** are less than the half of the ones of **OCT**, for all datasets except *P53mutants*. The generalization error of **DTsbe** is the best one in 16 out of 20 datasets. Contrasting with Table 2, the restriction on maximum tree depth leads to higher errors – considerably higher for *Letter*, *Kr-vs-kp*, *Optdigits*, and *Isolet*. The generalization errors of **OCT** are almost always higher than the ones of **DTsbe**, and even than the ones of the baseline. **OCT** wins for only one dataset. Such a low performance of **OCT** can be attributed to oversearching, as already observed for $\mathbf{DTaccept}_0$ in Table 2. The much better generalization error of **OCT** reported by Bertsimas and Dunn (2017) can instead be attributed to the parameter focusing approach used there to set the complexity penaly cp in the regularization of the empirical misclassification error.

8. Conclusions

We have introduced an original pruning algorithm of the search space of feature subsets which allows for enumerating all and only the distinct decision trees. The lattice of feature subsets is explored by distinguishing features that must be necessarily used from those that may be possibly used. The order of the visit is impressively effective, and it relies on an estimation of the chances of generating distinct trees. Based on the enumeration of distinct decision trees, we introduced an algorithm for finding δ-acceptable feature subsets, which depart by at most δ from the best estimated error of decision trees built from any feature subset. The framework is stated in general terms for any top-down greedy decision tree induction algorithm, any quality measure used to select split attributes, and any error estimation function. Coupled with a few computational optimizations and a multi-core parallel implementation, this makes it possible to investigate properties of complete search for datasets of up to 60 features. Beyond such a limit, we contributed by exploiting ideas and optimizations proposed in the paper to a white-box computational improvement of the sequential backward elimination heuristic, which extends its practical applicability to large

13. OCT parameters: 8 core processes in Julia, max depth = 5, other parameters set to default (min bucket = 1, local search = true, cp = 0.01). No focusing of parameters performed. Non-binary discrete features of experimental datasets are processed with one-hot encoding as required by OCT.

dimensional datasets. Experimental results reinforce, in the case of decision trees, previous findings that oversearching increases overfitting, and, in addition, they highlight that oversearching also increases instability. Sequential backward elimination performs better than complete search over the feature subsets, and also better than the OCT optimal (non-greedy) decision tree algorithm. It also improves the generalization error of random forest models for medium-to-large dimensional datasets.

Acknowledgments. I am grateful to the anonymous reviewers for several helpful comments. I also thank Jack Dunn for providing the OCT system.

References

M. Aldinucci, S. Ruggieri, and M. Torquati. Decision tree building on multi-core using FastFlow. *Concurrency and Computation: Practice and Experience*, 26(3):800–820, 2014.

E. Amaldi and V. Kann. On the approximation of minimizing non zero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209:237–260, 1998.

D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, 34(3):483–519, 2013.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth Publishing Company, 1984.

L. A. Breslow and D. W. Aha. Simplifying decision trees: A survey. *The Knowledge Engineering Review*, 12:1–40, 1997.

R. Caruana and D. Freitag. Greedy attribute selection. In *Proc. of the Int. Conf. on Machine Learning (ICML 1994)*, pages 28–36. Morgan Kaufmann, 1994.

R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proc. of the Int. Conf. on Machine Learning (ICML 2006)*, volume 148, pages 161–168. ACM, 2006.

T. M. Cover. On the possible ordering on the measurement selection problem. *Trans. Systems, Man, and Cybernetics*, 9:657–661, 1977.

M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4): 131–156, 1997.

M. Fernández Delgado, E. Cernadas, S. Barro, and D. Gomes Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

L. Devroye, L. Györfi, and G. A. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

J. Doak. An evaluation of feature selection methods and their application to computer security. Technical Report CSE-92-18, University of California Davis, 1992.

F. Esposito, D. Malerba, and G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):476–491, 1997.

L. Fan. Accurate robust and efficient error estimation for decision trees. In *Proc. of the Int. Conf. on Machine Learning (ICML 2016)*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 239–247, 2016.

I. Foroutan and J. Sklansky. Feature selection for automatic classification of non-gaussian data. *Trans. Systems, Man, and Cybernetics*, 17(2):187–198, 1987.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):93:1–93:42, December 2018.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Design and analysis of the NIPS2003 challenge. In I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors, *Feature Extraction: Foundations and Applications*, pages 237–263. Springer, 2006a.

I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors. *Feature Extraction: Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*. Springer, 2006b.

J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proc. of the Int. Conf. on Machine Learning (ICML 1994)*, pages 121–129. Morgan Kaufmann, 1994.

J.-H. Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745, 2009.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI 1995)*, pages 1137–1145. Morgan Kaufmann, 1995.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97 (1-2):273–324, 1997.

T. N. Lal, O. Chapelle, J. Weston, and A. Elisseeff. Embedded methods. In I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors, *Feature Extraction: Foundations and Applications*, pages 137–165. Springer, 2006.

M. Lichman. UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml.

H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

H. Liu, H. Motoda, and M. Dash. A monotonic measure for optimal feature selection. In *Proc. of the European Conf. on Machine Learning (ECML 1998)*, volume 1398 of *Lecture Notes in Computer Science*, pages 101–106. Springer, 1998.

M. Menickelly, O. Günlük, J. Kalagnanam, and K. Scheinberg. Optimal generalized decision trees via integer programming. *CoRR*, abs/1612.03225, 2016.

A. Miller. *Subset Selection in Regression*. Chapman and Hall, 2 edition, 2002.

S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2:345–389, 1998.

N. Narodytska, A. Ignatiev, F. Pereira, and J. Marques-Silva. Learning optimal decision trees with SAT. In *Proc. of Int. Joint Conf. on Artificial Intelligence (IJCAI 2018)*, pages 1362–1368. ijcai.org, 2018.

S. Nogueira and G. Brown. Measuring the stability of feature selection. In *Proc. of Machine Learning and Knowledge Discovery in Databases (ECML-PKDD 2016) Part II*, volume 9852 of *LNCS*, pages 442–457, 2016.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

J. R. Quinlan and M. Cameron-Jones. Oversearching and layered search in empirical learning. In *Proc. of Int. Joint Conf. on Artificial Intelligence (IJCAI 1995)*, pages 1019–1024. Morgan Kaufmann, 1995.

J. Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382, 2003.

S. Ruggieri. Efficient C4.5. *IEEE Transactions on Knowledge and Data Engineering*, 14: 438–444, 2002.

S. Ruggieri. YaDT: Yet another Decision tree Builder. In *Proc. of Int. Conf. on Tools with Artificial Intelligence (ICTAI 2004)*, pages 260–265. IEEE, 2004.

S. Ruggieri. Subtree replacement in decision tree simplification. In *Proc. of the SIAM Conf. on Data Mining (SDM 2012)*, pages 379–390. SIAM, 2012.

S. Ruggieri. Enumerating distinct decision trees. In *Proc. of the Int. Conf. on Machine Learning (ICML 2017)*, number 70 in JMLR Workshop and Conference Proceedings, pages 2960–2968, 2017.

M. Saar-Tsechansky and F. Provost. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1625–1657, 2007.

S. S. Skiena. *The Algorithm Design Manual*. Springer, 2 edition, 2008.

M. Stone. Asymptotics for and against cross-validation. *Biometrika*, 64:29–?35, 1997.

B. Twala. An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, 23(5):373–405, 2009.

S. Verwer and Y. Zhang. Learning decision trees with flexible constraints and objectives using integer optimization. In *Proc. of Int. Conf. on Integration of AI and OR Techniques in Constraint Programming (CPAIOR 2017)*, volume 10335 of *Lecture Notes in Computer Science*, pages 94–103. Springer, 2017.

Table 6: Cross-validation errors (mean ± stdev). IG+EBP and m=2.

dataset	baseline	DTsbe	DTaccept$_0$	RF	DTsbe+RF
Adult	14.62 ± 0.38	14.18 ± 0.44	**14.09 ± 0.40**	**14.04 ± 0.38**	14.21 ± 0.45
Letter	11.91 ± 0.71⋆	**11.90 ± 0.70**	11.95 ± 0.75⋆	**4.13 ± 0.50**	4.77 ± 0.65
Hypo	**0.46 ± 0.31**	0.53 ± 0.38	0.57 ± 0.40	0.83 ± 0.44	**0.50 ± 0.38**
Ionosphere	11.48 ± 5.87	**9.58 ± 4.51**	11.06 ± 5.58	**6.53 ± 4.08**	8.52 ± 4.46
Soybean	**9.87 ± 3.70**	12.31 ± 3.80	11.51 ± 4.20	**8.64 ± 2.96**	11.83 ± 3.78
Kr-vs-kp	**0.64 ± 0.47**	1.06 ± 0.57	-	1.94 ± 0.81⋆	**1.82 ± 0.84**
Anneal	**0.93 ± 0.86**	1.41 ± 1.26	2.22 ± 1.99	**1.18 ± 1.02**	1.65 ± 1.50
Census	5.14 ± 0.10	**4.93 ± 0.10**	-	**4.87 ± 0.07**	4.91 ± 0.12
Spambase	7.06 ± 1.23⋆	**6.95 ± 1.21**	-	**5.64 ± 1.19**	5.81 ± 1.06
Sonar	**25.97 ± 9.34**	25.99 ± 9.08⋆	26.69 ± 9.40⋆	**17.50 ± 8.48**	22.88 ± 8.67
Optdigits	**9.63 ± 1.20**	9.90 ± 1.67⋆	-	**2.00 ± 0.71**	2.79 ± 0.84
Coil2000	5.98 ± 0.11⋆	**5.97 ± 0.05**	-	**5.97 ± 0.05**	5.97 ± 0.05⋆
Clean1	**17.53 ± 6.87**	19.41 ± 7.15	-	**10.73 ± 4.46**	15.35 ± 5.55
Clean2	3.29 ± 0.67	**2.76 ± 0.71**	-	2.41 ± 0.54⋆	**2.38 ± 0.55**
Madelon	25.73 ± 3.45	**21.77 ± 3.39**	-	35.72 ± 3.06	**20.03 ± 3.06**
Isolet	16.68 ± 1.15	**16.34 ± 1.36**	-	**6.11 ± 0.78**	6.35 ± 0.80
Gisette	5.94 ± 0.83	**5.57 ± 0.87**	-	4.01 ± 0.72	**3.76 ± 0.76**
P53mutants	0.52 ± 0.09	**0.49 ± 0.10**	-	0.46 ± 0.04⋆	**0.45 ± 0.07**
Arcene	**48.09 ± 5.39**	48.57 ± 4.96⋆	-	46.34 ± 3.25	**44.51 ± 2.99**
Dexter	48.92 ± 2.94⋆	**48.13 ± 3.51**	-	48.96 ± 1.89	**43.74 ± 3.00**
wins/⋆	8/4	11/3	1/2	12/3	8/1

Appendix A. Additional Experimental Results

First, we consider tree simplification, a well-known strategy to address the common problems of overfitting the training data and of trading accuracy for simplicity. Decision tree pruning does not satisfy Assumption 5, hence it cannot be embedded within the heuristic or complete search procedures. We apply then the C4.5 Error Based Pruning (EBP) (Breslow and Aha, 1997) approach as a post-processing on the decision trees built over the feature subset selected by the strategies considered. The running time added by such a post-processing is negligible. Table 6 reports the generalization errors, including, for comparison purposes, also the random forest models. The winners/starred strategies are almost identical to Table 2. Generalization errors are now almost always smaller for all strategies (for *Coil2000* the improvement is vast), confirming that tree simplification is an effective strategy (Ruggieri, 2012). Finally, contrasting Table 6 with Table 3, there is no evidence that adding pruning to random forests improves their accuracy.

Next, we consider whether the results extend to quality measures other than IG. Table 7 and Table 8 report the elapsed times and the generalization errors for the Gain Ratio (GR) (see footnote 1). Running times are generally greater than the ones of IG shown in Table 1 – up to 2.5× for **DTsbe** and up to 10× for **DTaccept$_0$**. Regarding generalization errors, the baseline wins or is not stastitically different from the winner in 3 additional cases with respect to IG (cfr. Table 2), thus lowering the gap with the search heuristics. Also, the benefits of adding **DTsbe** to random forests for large dimensional datasets is not as marked as in Table 3 and in Table 6. The improved generalization performances can be attributed to the better discriminative power of Gain Ratio over Information Gain.

Table 7: Average elapsed running times (seconds). GR and m=2.

dataset	DTsbe	SBE	ratio	DTaccept$_0$	RF
Adult	0.78	1.40	0.559	48.68	1.97
Letter	0.47	0.77	0.617	302.62	0.90
Hypo	0.06	0.22	0.282	309.80	0.05
Ionosphere	0.01	0.10	0.106	1.01	0.01
Soybean	0.05	0.15	0.312	>1h	0.03
Kr-vs-kp	0.07	0.23	0.285	>1h	0.07
Anneal	0.03	0.14	0.196	185.51	0.02
Census	51.23	135.70	0.378	>1h	21.03
Spambase	0.85	5.11	0.167	>1h	0.13
Sonar	0.01	0.42	0.030	22.42	0.01
Optdigits	0.61	4.96	0.123	>1h	0.14
Coil2000	2.77	24.09	0.115	>1h	0.32
Clean1	0.19	25.79	0.007	>1h	0.03
Clean2	7.50	195.43	0.038	>1h	0.24
Madelon	21.81	>1h	-	>1h	0.28
Isolet	492.55	>1h	-	>1h	2.29
Gisette	211.04	>1h	-	>1h	11.08
P53mutants	1012.83	>1h	-	>1h	9.70
Arcene	66.52	>1h	-	>1h	4.58
Dexter	1656.07	>1h	-	>1h	0.18

Table 8: Cross-validation errors (mean ± stdev). GR and m=2.

dataset	baseline	DTsbe	DTaccept$_0$	RF	DTsbe+RF
Adult	14.62 ± 0.43	13.91 ± 0.50	**13.77 ± 0.44**	**13.50 ± 0.40**	13.78 ± 0.48
Letter	12.76 ± 0.78	12.25 ± 0.77	**12.14 ± 0.80**	**3.78 ± 0.41**	4.42 ± 0.66
Hypo	**0.47 ± 0.35**	0.52 ± 0.42⋆	0.54 ± 0.41⋆	0.54 ± 0.38⋆	**0.48 ± 0.39**
Ionosphere	10.57 ± 5.01⋆	**9.94 ± 4.93**	11.46 ± 5.05	**6.21 ± 3.83**	8.60 ± 4.70
Soybean	**8.45 ± 3.02**	9.18 ± 3.43	-	**5.69 ± 2.31**	7.92 ± 2.71
Kr-vs-kp	**0.63 ± 0.52**	1.05 ± 0.57	-	**1.03 ± 0.59**	1.12 ± 0.66⋆
Anneal	**1.34 ± 1.16**	1.85 ± 1.53	2.31 ± 1.85	**0.62 ± 0.88**	1.82 ± 1.65
Census	5.50 ± 0.10	**5.03 ± 0.29**	-	**4.51 ± 0.08**	4.67 ± 0.15
Spambase	7.57 ± 1.28⋆	**7.37 ± 1.25**	-	**5.44 ± 1.04**	5.63 ± 1.03
Sonar	25.63 ± 9.08⋆	**25.14 ± 8.83**	26.17 ± 8.97⋆	**19.70 ± 8.01**	23.61 ± 9.27
Optdigits	**10.23 ± 1.65**	10.60 ± 1.91	-	**2.01 ± 0.63**	2.59 ± 0.76
Coil2000	9.00 ± 0.70⋆	**8.88 ± 0.73**	-	**6.01 ± 0.10**	6.05 ± 0.14
Clean1	**17.68 ± 5.93**	18.59 ± 6.02⋆	-	**9.93 ± 4.20**	13.15 ± 4.91
Clean2	3.18 ± 0.72	**2.91 ± 0.71**	-	**2.36 ± 0.56**	2.38 ± 0.55⋆
Madelon	30.40 ± 4.05	**24.96 ± 3.73**	-	35.35 ± 2.53	**23.00 ± 3.73**
Isolet	**16.51 ± 1.27**	16.58 ± 1.16⋆	-	**6.38 ± 0.74**	6.44 ± 0.82⋆
Gisette	6.62 ± 0.99	**5.97 ± 1.02**	-	**4.10 ± 0.74**	4.20 ± 0.81⋆
P53mutants	0.59 ± 0.11	**0.52 ± 0.09**	-	**0.45 ± 0.04**	0.45 ± 0.06⋆
Arcene	48.67 ± 5.64⋆	**47.50 ± 4.85**	-	45.28 ± 2.52	**44.37 ± 1.91**
Dexter	48.55 ± 2.56⋆	**48.25 ± 3.29**	-	49.36 ± 1.56	**45.08 ± 2.80**
wins/⋆	7/6	11/3	2/2	16/1	4/5

Journal of Machine Learning Research 20 (2019) 1-23 Submitted 2/18; Revised 3/19; Published 7/19

Optimal Transport: Fast Probabilistic Approximation with Exact Solvers

Max Sommerfeld MAX.SOMMERFELD@MATHEMATIK.UNI-GOETTINGEN.DE
Felix-Bernstein Institute for Mathematical Statistics in the Biosciences
University of Göttingen
Goldschmidtstr. 7, 37077 Göttingen

Jörn Schrieber JOERN.SCHRIEBER-1@MATHEMATIK.UNI-GOETTINGEN.DE
Institute for Mathematical Stochastics
University of Göttingen
Goldschmidtstr. 7, 37077 Göttingen

Yoav Zemel YOAV.ZEMEL@MATHEMATIK.UNI-GOETTINGEN.DE
Felix-Bernstein Institute for Mathematical Statistics in the Biosciences
University of Göttingen
Goldschmidtstr. 7, 37077 Göttingen

Axel Munk MUNK@MATH.UNI-GOETTINGEN.DE
Institute for Mathematical Stochastics and Felix-Bernstein Institute for Mathematical Statistics in the Biosciences
University of Göttingen
Goldschmidtstr. 7, 37077 Göttingen
and
Max-Planck-Institute for Biophysical Chemistry
Am Faßberg 11, 37077 Göttingen

Editor: Animashree Anandkumar

Abstract

We propose a simple subsampling scheme for fast randomized approximate computation of optimal transport distances on finite spaces. This scheme operates on a random subset of the full data and can use any exact algorithm as a black-box back-end, including state-of-the-art solvers and entropically penalized versions. It is based on averaging the exact distances between empirical measures generated from independent samples from the original measures and can easily be tuned towards higher accuracy or shorter computation times. To this end, we give non-asymptotic deviation bounds for its accuracy in the case of discrete optimal transport problems. In particular, we show that in many important instances, including images (2D-histograms), the approximation error is independent of the size of the full problem. We present numerical experiments that demonstrate that a very good approximation in typical applications can be obtained in a computation time that is several orders of magnitude smaller than what is required for exact computation of the full problem.

Keywords: computational vs statistical accuracy, covering numbers, empirical optimal transport, resampling, risk bounds, spanning tree, Wasserstein distance

©2019 Max Sommerfeld, Jörn Schrieber, Yoav Zemel, and Axel Munk.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-079.html.

1. Introduction

Optimal transport distances, a.k.a. Wasserstein, earth-mover's, Monge-Kantorovich-Rubinstein or Mallows distances, as metrics to compare probability measures (Rachev and Rüschendorf, 1998; Villani, 2008) have become a popular tool in a wide range of applications in computer science, machine learning and statistics. Important examples are image retrieval (Rubner et al., 2000) and classification (Zhang et al., 2007), computer vision (Ni et al., 2009), but also therapeutic equivalence (Munk and Czado, 1998), generative modeling (Bousquet et al., 2017), biometrics (Sommerfeld and Munk, 2018), metagenomics (Evans and Matsen, 2012) and medical imaging (Ruttenberg et al., 2013).

Optimal transport distances compare probability measures by incorporating a suitable ground distance on the underlying space, typically driven by the particular application, e.g. Euclidean distance. This often makes it preferable to competing distances such as total-variation or χ^2-distances, which are oblivious to any metric or similarity structure on the ground space. Note that total variation is the Wasserstein distance with respect to the trivial metric, which usually does not carry the geometry of the underlying ground space. In this setting, optimal transport distances have a clear and intuitive interpretation as the amount of 'work' required to transport one probability distribution onto the other. This notion is typically well-aligned with human perception of similarity (Rubner et al., 2000).

1.1. Computation

The outstanding theoretical and practical performance of optimal transport distances is contrasted by its excessive computational cost. For example, optimal transport distances can be computed with an auction algorithm (Bertsekas, 1992). For two probability measures supported on N points this algorithm has a worst case run time of $\mathcal{O}(N^3 \log N)$. Other methods like the transportation simplex have sub-cubic empirical average runtime (compare Gottschlich and Schuhmacher, 2014), but exponential worst case runtimes.

Therefore, many attempts have been made to design improved algorithms. We give some selective references: Ling and Okada (2007) proposed a specialized algorithm for L_1-ground distance and $\mathcal{X}$ a regular grid and report an empirical runtime of $\mathcal{O}(N^2)$. Gottschlich and Schuhmacher (2014) improved existing general purpose algorithms by initializing with a greedy heuristic. Their *Shortlist* algorithm achieves an empirical average runtime of the order $\mathcal{O}(N^{5/2})$. Schmitzer (2016) solves the optimal transport problem by solving a sequence of sparse problems. The theoretical runtime of his algorithm is not known, but it exhibits excellent performance on two-dimensional grids (Schrieber et al., 2016). The literature on this topic is rapidly growing and we refer for further recent work to Liu et al. (2018), Dvurechensky et al. (2018), Lin et al. (2019), and the references given there.

Despite these efforts, still many practically relevant problems remain well outside the scope of available algorithms. See Schrieber et al. (2016) for an overview and a numerical comparison of state-of-the-art algorithms for discrete optimal transport. This is true in particular for two or three dimensional images and spatio temporal imaging, which constitute an important area of potential applications. Here, N is the number of pixels or voxels and is typically of size 10^5 to 10^7. Naturally, this problem is aggravated when many distances have to be computed, as is the case for Wasserstein barycenters (Agueh and Carlier, 2011; Cuturi and Doucet, 2014), which have become an important use case.

To bypass the computational bottleneck, also many surrogates for optimal transport distances that are more amenable to fast computation have been proposed. Shirdhonkar and Jacobs (2008) proposed to use an equivalent distance based on wavelets that can be computed in linear time but cannot be calibrated to approximate the Wasserstein distance with arbitrary accuracy. Pele and Werman (2009) threshold the ground distance to reduce the complexity of the underlying linear program, obtaining a lower bound for the exact distance. Cuturi (2013) altered the optimization problem by adding an entropic penalty term in order to use faster and more stable algorithms, see

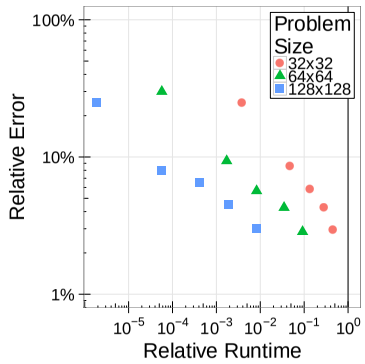


Figure 1: Relative error and relative runtime compared to the exact computation of the proposed scheme. Optimal transport distances and its approximations were computed between images of different sizes (32×32, 64×64, 128×128). Each point represents a specific parameter choice in the scheme and is a mean over different problem instances, solvers and cost exponents. For the relative runtimes the geometric mean is reported. For details on the parameters see Figure 2.

also Altschuler et al. (2017). Bonneel et al. (2015) consider the 1D Wasserstein distances of radial projections of the original measures, exploiting the fact that, in one dimension, computing the Wasserstein distance amounts to sorting the point masses and hence has quasi-linear computation time.

1.2. Contribution

We do *not* propose a new algorithm to solve the optimal transport problem. Instead, we propose a simple probabilistic scheme as a meta-algorithm that can use any algorithm (e.g., those mentioned above) solving finitely supported optimal transport problems as a black-box back-end and gives a random but fast approximation of the exact distance. This scheme

a) is extremely easy to implement, to parallelize and to tune towards higher accuracy or shorter computation time as desired (see Figure 1);

b) can be used with any algorithm for transportation problems as a back-end, including general LP solvers, specialized network solvers and algorithms using entropic penalization (Cuturi, 2013);

c) comes with theoretical non-asymptotic guarantees for the approximation error of the Wasserstein distance—in particular, this error is independent of the size of the original problem in many important cases, including images;

d) works well in practice. For example, the Wasserstein distance between two 128^2-pixel images can typically be approximated with a relative error of less than 5% in only 1% of the time required for exact computation.

2. Problem and Algorithm

Although our meta-algorithm is applicable to exact solvers for any optimal transport distance between probability measures, for example the Sinkhorn distance (Cuturi, 2013), the theory we present here concerns the Kantorovich (1942) transport distance, often also denoted as *Wasserstein distance*.

Wasserstein Distance Consider a fixed finite space $\mathcal{X} = \{x_1, \dots, x_N\}$ with a metric $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$. Every probability measure on $\mathcal{X}$ is given by a vector $\boldsymbol{r}$ in

$$\mathcal{P}_{\mathcal{X}} = \left\{ \boldsymbol{r} = (r_x)_{x \in \mathcal{X}} \in \mathbb{R}^{\mathcal{X}}_{\geq 0} : \sum_{x \in \mathcal{X}} r_x = 1 \right\},$$

via $P_{\boldsymbol{r}}(\{x\}) = r_x$. We will not distinguish between the vector $\boldsymbol{r}$ and the measure it defines. For $p \geq 1$, the *p-th Wasserstein distance* between two probability measures $\boldsymbol{r}, \boldsymbol{s} \in \mathcal{P}_{\mathcal{X}}$ is defined as

$$W_p(\boldsymbol{r}, \boldsymbol{s}) = \left(\min_{\boldsymbol{w} \in \Pi(\boldsymbol{r}, \boldsymbol{s})} \sum_{x, x' \in \mathcal{X}} d^p(x, x') w_{x,x'} \right)^{1/p}, \tag{1}$$

where $\Pi(\boldsymbol{r}, \boldsymbol{s})$ is the set of all probability measures on $\mathcal{X} \times \mathcal{X}$ with marginal distributions $\boldsymbol{r}$ and $\boldsymbol{s}$, respectively. The minimization in (1) can be written as a linear program

$$\min \sum_{x, x' \in \mathcal{X}} w_{x,x'} d^p(x, x') \quad \textbf{s.t.} \quad \sum_{x' \in \mathcal{X}} w_{x,x'} = r_x, \quad \sum_{x \in \mathcal{X}} w_{x,x'} = s_{x'}, \quad w_{x,x'} \geq 0, \tag{2}$$

with N^2 variables $w_{x,x'}$ and $2N$ constraints, where the weights $d^p(x, x')$ are known and have been precalculated.

2.1. Approximating the Wasserstein Distance

The idea of the proposed algorithm is to replace a probability measure $\boldsymbol{r} \in \mathcal{P}(\mathcal{X})$ with an empirical measure $\hat{\boldsymbol{r}}_S$ based on i.i.d. picks $X_1, \dots, X_S \sim \boldsymbol{r}$ for some integer S:

$$\hat{r}_{S,x} = \frac{1}{S} \# \{k : X_k = x\}, \quad x \in \mathcal{X}. \tag{3}$$

Likewise, replace $\boldsymbol{s}$ with $\hat{\boldsymbol{s}}_S$. Then, use the *empirical optimal transport distance* (EOT) $W_p(\hat{\boldsymbol{r}}_S, \hat{\boldsymbol{s}}_S)$ as a random approximation of $W_p(\boldsymbol{r}, \boldsymbol{s})$.

Algorithm 1 Statistical approximation of $W_p(\boldsymbol{r}, \boldsymbol{s})$

Input: Probability measures $\boldsymbol{r}, \boldsymbol{s} \in \mathcal{P}_{\mathcal{X}}$, sample size S and number of repetitions B
for $i = 1 \dots B$ **do**
  Sample i.i.d. $X_1, \dots, X_S \sim \boldsymbol{r}$ and independently $Y_1, \dots, Y_S \sim \boldsymbol{s}$
  $\hat{r}_{S,x} \leftarrow \# \{k : X_k = x\} / S$ **for all** $x \in \mathcal{X}$
  $\hat{s}_{S,x} \leftarrow \# \{k : Y_k = x\} / S$ **for all** $x \in \mathcal{X}$
  Compute $\hat{W}^{(i)} \leftarrow W_p(\hat{\boldsymbol{r}}_S, \hat{\boldsymbol{s}}_S)$
end for
Return: $\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) \leftarrow B^{-1} \sum_{i=1}^{B} \hat{W}^{(i)}$

In each of the B iterations in Algorithm 1, the Wasserstein distance between two sets of S point masses has to be computed. For the exact Wasserstein distance, two measures on N points need to be compared. If we take for example the super-cubic runtime of the auction algorithm as a basis, Algorithm 1 has worst case runtime

$$\mathcal{O}(BS^3 \log S)$$

compared to $\mathcal{O}(N^3 \log N)$ for the exact distance. This means a dramatic reduction of computation time if S (and B) are small compared to N.

The application of Algorithm 1 to other optimal transport distances is straightforward. One can simply replace $W_p(\hat{\boldsymbol{r}}_S, \hat{\boldsymbol{s}}_S)$ with the desired distance, e.g., the Sinkhorn distance (Cuturi, 2013), see also our numerical experiments below. Further, the algorithm can be applied to non-discrete instances as long as we can sample from the measures. However, the theoretical results below only apply to the EOT on a finite ground space $\mathcal{X}$.

3. Theoretical Results

We give general non-asymptotic guarantees for the quality of the approximation $\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) = B^{-1} \sum_{i=1}^{B} W_p(\hat{\boldsymbol{r}}_{S,i}, \hat{\boldsymbol{s}}_{S,i})$ (where $\hat{\boldsymbol{r}}_{S,i}$ are independent empirical measures of size S from $\boldsymbol{r}$; see Algorithm 1) in terms of the expected L_1-error. That is, we give bounds of the form

$$E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) - W_p(\boldsymbol{r}, \boldsymbol{s})\right|\right] \leq g(S, \mathcal{X}, p), \tag{4}$$

for some function g. We are particularly interested in the dependence of the bound on the size N of $\mathcal{X}$ and on the sample size S as this determines how the number of sampling points S (and hence the computational effort of Algorithm 1) must be increased for increasing problem size N in order to retain (on average) a certain approximation quality. In a second step, we obtain deviation inequalities for $\hat{W}^{(S)}(\boldsymbol{r}, \boldsymbol{s})$ via concentration of measure techniques.

Related work The question of the convergence of empirical measures to the true measure in expected Wasserstein distance has been considered in detail by Boissard and Le Gouic (2014) and Fournier and Guillin (2015). The case of the underlying measures being different (that is, the convergence of $EW_p(\hat{\boldsymbol{r}}_S, \hat{\boldsymbol{s}}_S)$ to $W_p(\boldsymbol{r}, \boldsymbol{s})$ when $\boldsymbol{r} \neq \boldsymbol{s}$) has not been considered to the best of our knowledge. Theorem 1 is reminiscent of the main result of Boissard and Le Gouic (2014). However, we give a result here, which is explicitly tailored to finite spaces and makes explicit the dependence of the constants on the size N of the underlying set $\mathcal{X}$. In fact, when we consider finite spaces $\mathcal{X}$ which are subsets of $\mathbb{R}^D$ later in Theorem 3, we will see that in contrast to the results of Boissard and Le Gouic (2014), the rate of convergence (in S) does not change when the dimension gets large, but rather the dependence of the constants on N changes. This is a valuable insight as our main concern here is how the subsample size S (driving the computational cost) must be chosen when N grows in order to retain a certain approximation quality.

3.1. Expected Absolute Error

Recall that, for $\delta > 0$ the *covering number* $\mathcal{N}(\mathcal{X}, \delta)$ of $\mathcal{X}$ is defined as the minimal number of closed balls with radius δ and centers in $\mathcal{X}$ that is needed to cover $\mathcal{X}$. Note that in contrast to continuous spaces, $\mathcal{N}(\mathcal{X}, \delta)$ is bounded by N for all $\delta > 0$.

Theorem 1. *Let $\hat{\boldsymbol{r}}_S$ be the empirical measure obtained from i.i.d. samples $X_1, \ldots, X_S \sim \boldsymbol{r}$, then*

$$E\left[W_p^p(\hat{\boldsymbol{r}}_S, \boldsymbol{r})\right] \leq \mathcal{E}_q / \sqrt{S}, \tag{5}$$

where the constant $\mathcal{E}_q := \mathcal{E}_q(\mathcal{X}, p)$ is given by

$$\mathcal{E}_q = 2^{p-1} q^{2p} (\operatorname{diam}(\mathcal{X}))^p \left(q^{-(l_{\max}+1)p} \sqrt{N} + \sum_{l=0}^{l_{\max}} q^{-lp} \sqrt{\mathcal{N}(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}))} \right) \tag{6}$$

for any $2 \leq q \in \mathbb{N}$ and $l_{\max} \in \mathbb{N}$.

Remark 1. Since Theorem 1 holds for any integer $q \geq 2$ and $l_{\max} \in \mathbb{N}$, they can be chosen freely to minimize the constant $\mathcal{E}_q$. In the proof they appear as the branching number and depth of a spanning tree that is constructed on $\mathcal{X}$ (see appendix). In general, an optimal choice of q and $l_{\max}$ cannot be given. However, in the Euclidean case, the optimal values for q and $l_{\max}$ will be determined, and in particular we will show that $q = 2$ is optimal (see the discussion after Theorem 3, and Lemma 1).

Remark 2 (covering by arbitrary sets)**.** At the price of a factor 2^p, we can replace the balls defining the covering numbers $\mathcal{N}$ with arbitrary sets, and obtain the bound

$$\mathcal{E}_q = 2^{2p-1} q^{2p} (\operatorname{diam}(\mathcal{X}))^p \left(q^{-(l_{\max}+1)p} \sqrt{N} + \sum_{l=0}^{l_{\max}} q^{-lp} \sqrt{\mathcal{N}_1(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}))} \right),$$

where $\mathcal{N}_1(\mathcal{X}, \delta)$ is the minimal number of closed sets of diameter $\leq 2\delta$ needed to cover $\mathcal{X}$. The proof is given in the appendix. These alternative covering numbers lead to better bounds in high-dimensional Euclidean spaces when $p > 2.5$ (see Remark 3).

Based on Theorem 1, we can formulate a bound for the mean approximation error of Algorithm 1. A mean squared error version is given below, in Theorem 5.

Theorem 2. *Let $\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s})$ be as in Algorithm 1 for any choice of $B \in \mathbb{N}$. Then for every integer $q \geq 2$*

$$E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) - W_p(\boldsymbol{r}, \boldsymbol{s})\right|\right] \leq 2\mathcal{E}_q^{1/p} S^{-1/(2p)}. \tag{7}$$

Proof. The statement is an immediate consequence of the reverse triangle inequality for the Wasserstein distance, Jensen's inequality and Theorem 1,

$$\begin{aligned} E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) - W_p(\boldsymbol{r}, \boldsymbol{s})\right|\right] &\leq E\left[W_p(\hat{\boldsymbol{r}}_S, \boldsymbol{r}) + W_p(\hat{\boldsymbol{s}}_S, \boldsymbol{s})\right] \\ &\leq E\left[W_p^p(\hat{\boldsymbol{r}}_S, \boldsymbol{r})\right]^{1/p} + E\left[W_p^p(\hat{\boldsymbol{s}}_S, \boldsymbol{s})\right]^{1/p} \leq 2\mathcal{E}_q^{1/p} / S^{1/(2p)}. \end{aligned}$$

□

Measures on Euclidean Space While the constant $\mathcal{E}_q$ in Theorem 1 may be difficult to compute or estimate in general, we give explicit bounds in the case when $\mathcal{X}$ is a finite subset of a Euclidean space. They exhibit the dependence of the approximation error on $N = |\mathcal{X}|$. In particular, it comprises the case when the measures represent images (two- or more dimensional).

Theorem 3. *Let $\mathcal{X}$ be a finite subset of $\mathbb{R}^D$ with the usual Euclidean metric. Then,*

$$\mathcal{E}_2 \leq D^{p/2} 2^{3p-1} (\operatorname{diam}(\mathcal{X}))^p \cdot C_{D,p}(N),$$

where $N = |\mathcal{X}|$ and

$$C_{D,p}(N) = \begin{cases} 1/(1 - 2^{D/2-p}) & D < 2p, \\ 2 + D^{-1} \log_2 N & D = 2p, \\ N^{1/2-p/D}[2 + 1/(2^{D/2-p} - 1)] & D > 2p. \end{cases} \tag{8}$$

One can obtain bounds for $\mathcal{E}_q$, $q > 2$ (see the proof), but the choice $q = 2$ leads to the smallest bound (Lemma 1a, page 17). Further, if p is an integer, then

$$C_{D,p}(N) \leq \begin{cases} 2 + \sqrt{2} & D < 2p, \\ 2 + D^{-1} \log_2 N & D = 2p, \\ (3 + \sqrt{2}) N^{1/2-p/D} & D > 2p \end{cases}$$

(see Lemma 1b).

In particular, we have for the most important cases $p = 1, 2$:

Corollary 1. *Under the conditions of Theorem 3,*

$$p = 1 \quad \Longrightarrow \quad \mathcal{E}_2 \leq 4D^{1/2}\,\mathrm{diam}(\mathcal{X}) \cdot \begin{cases} 1/(1-2^{D/2-1}) & D < 2, \\ 2 + (1/2)\log_2 N & D = 2, \\ N^{1/2-1/D}[2 + 1/(2^{D/2-1}-1)] & D > 2. \end{cases}$$

$$p = 2 \quad \Longrightarrow \quad \mathcal{E}_2 \leq 32D(\,\mathrm{diam}(\mathcal{X}))^2 \cdot \begin{cases} 1/(1-2^{D/2-2}) & D < 4, \\ 2 + (1/4)\log_2 N & D = 4, \\ N^{1/2-2/D}[2 + 1/(2^{D/2-2}-1)] & D > 4. \end{cases}$$

Remark 3 (improved bounds in high dimensions)**.** The term $D^{p/2}$ appears because in the proof of Theorem 3 we switch between the Euclidean norm and the supremum norm. One may wonder whether this change of norms is necessary. We can stay in the Euclidean setting, and may assume without loss of generality that $\mathcal{X}$ is included in $B_{\mathrm{diam}(\mathcal{X})}(0)$, where $B_r(x) = \{y : \|y - x\|_2 \leq r\}$ is the closed ball of radius r around x. According to Verger-Gaugry (2005), there exists an absolute constant C such that $\mathcal{N}(B_1(0), \epsilon) \leq C^2 D^{5/2} \epsilon^{-D}$. Using this would allow to replace $D^{p/2}$ by $C2^{D/2}D^{5/4}$, or, combining the alternative covering numbers $\mathcal{N}_1$ (Remark 2), by $C2^p D^{5/4}$. This is better than $D^{p/2}$ when $p > 2.5$ and D is large.

Theorem 3 gives control over the error made by the approximation $\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s})$ of $W_p(\boldsymbol{r}, \boldsymbol{s})$. Of particular interest is the behavior of this error as N gets large (e.g., for high resolution images). We distinguish three cases. In the *low-dimensional case* $p' = D/2 - p < 0$, we have $C_{D,p}(N) = \mathcal{O}(1)$ and the approximation error is $\mathcal{O}(S^{-\frac{1}{2p}})$ independent of the size of the image. In the *critical case* $p' = 0$ the approximation error is no longer independent of N but is of order $\mathcal{O}\left(\log(N) S^{-\frac{1}{2p}}\right)$. Finally, in the *high-dimensional case* the dependence on N becomes stronger with an approximation error of order

$$\mathcal{O}\left(\left(\frac{N^{(1-\frac{2p}{D})}}{S}\right)^{\frac{1}{2p}}\right).$$

In all cases one can choose $S = o(N)$ while still guaranteeing vanishing approximation error for $N \to \infty$. In practice, this means that S can typically be chosen (much) smaller than N to obtain a good approximation of the Wasserstein distance. In particular, this implies that for low-dimensional applications with two or three dimensional histograms (for example grayscale images, where N corresponds to the number of pixels / voxels and $\boldsymbol{r}, \boldsymbol{s}$ correspond to the grey value distribution after normalization), the approximation error is essentially not affected by the size of the problem when p is not too small, e.g., $p = 2$.

While the three cases in Theorem 3 resemble those given by Boissard and Le Gouic (2014), the rate of convergence in S as seen in Theorem 1 is $\mathcal{O}(S^{-1/2})$, regardless of the dimension of the underlying space $\mathcal{X}$. The constant depends on D, however, roughly at the polynomial rate $D^{p/2}$ and through $C_{D,p}(N)$. It is also worth mentioning that by considering the dual transport problem, our results can be recast in the framework of Shalev-Shwartz et al. (2010).

Remark 4. The results presented here extend to the case where $\mathcal{X}$ is a bounded, countable subset of $\mathbb{R}^D$. However, our bounds for $\mathcal{E}_q$ contain the term $C_{D,p}(N)$, which is finite as $N \to \infty$ in the low-dimensional case ($D < 2p$) but infinite otherwise. Finding a better bound for $\mathcal{E}_q$ when $\mathcal{X}$ is countable is challenging and an interesting topic for further research.

3.2. Concentration Bounds

Based on the bounds for the expected approximation error we now give non-asymptotic guarantees for the approximation error in the form of deviation bounds using standard concentration of measure techniques.

Theorem 4. *If $\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s})$ is obtained from Algorithm 1, then for every $z \geq 0$*

$$P\left[|\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) - W_p(\boldsymbol{r}, \boldsymbol{s})| \geq z + \frac{2\mathcal{E}_q^{1/p}}{S^{1/2p}}\right] \leq 2\exp\left(-\frac{SBz^{2p}}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right). \tag{9}$$

Note that while the mean approximation quality $2\mathcal{E}_q^{1/p}/S^{1/(2p)}$ only depends on the subsample size S, the stochastic variability (see the right hand side term in Equation 9) depends on the product SB. This means that the repetition number B cannot decrease the expected error but it decreases the magnitude of fluctuation around it.

From these concentration bounds we can obtain a mean squared error version of Theorem 2:

Theorem 5. *Let $\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s})$ be as in Algorithm 1 for any choice of $B \in \mathbb{N}$. Then for every integer $q \geq 2$ the mean squared error of the EOT can be bounded as*

$$E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) - W_p(\boldsymbol{r}, \boldsymbol{s})\right|^2\right] \leq 18\mathcal{E}_q^{2/p} S^{-1/p} = \mathcal{O}(S^{-1/p}).$$

Remark 5. The power 2 can be replaced by any $\alpha \leq 2p$ with rate $S^{-\alpha/(2p)}$, as can be seen from a straightforward modification of the first lines of the proof.

For example, in view of Theorem 3, when $\mathcal{X}$ is a finite subset of a $\mathbb{R}^D$ and $q = 2$, we obtain

$$E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) - W_p(\boldsymbol{r}, \boldsymbol{s})\right|^2\right] \leq 3^2 2^{7-2/p} D C_{D,p}^{2/p}(N)[\,\mathrm{diam}(\mathcal{X})]^2 S^{-1/p}.$$

with the constant $C_{D,p}(N)$ given in (8). Thus, we qualitatively observe the same dependence on N as in Theorem 3, e.g., the mean squared error is independent of N when $D < 2p$.

4. Simulations

This section covers the numerical findings of the simulations. Runtimes and returned values of Algorithm 1 for each back-end solver are reported in relation to the results of that solver on the original problem. Four different solvers are tested.

4.1. Simulation Setup

The setup of our simulations is identical to that of Schrieber et al. (2016). One single core of a Linux server (AMD Opteron Processor 6140 from 2011 with 2.6 GHz) was used. The original and subsampled instances were run under the same conditions.

Three of the four methods featured in this simulation are exact linear programming solvers. The transportation simplex is a modified version of the network simplex solver tailored towards optimal transport problems. Details can be found for example in Luenberger and Ye (2008). The shortlist method (Gottschlich and Schuhmacher, 2014) is a modification of the transportation simplex, that performs an additional greedy step to quickly find a good initial solution. The parameters were chosen as the default parameters described in that paper. The third method is the network simplex solver of CPLEX (`www.ibm.com/software/commerce/optimization/cplex-optimizer/`). For the transportation simplex and the shortlist method the implementations provided in the R package *transport* (Schuhmacher et al., 2014) were used. The models for the CPLEX solver were created and solved via the R package *Rcplex* (Bravo and Theussl, 2016).

Additionally, the Sinkhorn scaling algorithm (Cuturi, 2013) was tested in our simulation. This method computes an entropy regularized optimal transport distance. The regularization parameter was chosen according to the heuristic in Cuturi (2013). Note that the Sinkhorn distance is not

covered by the theoretical results from Section 3. The errors reported for the Sinkhorn scaling are relative to the values returned by the algorithm on the full problems, which themselves differ from the actual Wasserstein distances.

The instances of optimal transport considered here are discrete instances of two different types: regular grids in two dimensions, that is, images in various resolutions, as well as point clouds in $[0,1]^D$ with dimensions $D = 2$, 3 and 4. For the image case, from the DOTmark, which contains images of various types intended to be used as optimal transport instances in the form of two-dimensional histograms, three instances were chosen: two images of each of the classes White Noise, Cauchy Density, and Classic Images, which are then treated in the three resolutions 32×32, 64×64 and 128×128. Images are interpreted as finitely supported measures. The mass of a pixel is given by the grayscale value and the support of the measure is the grid $\{1, \ldots, R\} \times \{1, \ldots, R\}$ for an image with resolution $R \times R$.

In the White Noise class the grayscale values of the pixels are independent of each other, the Cauchy Density images show bivariate Cauchy densities with random centers and varying scale ellipses, while Classic Images contains grayscale test images. See Schrieber et al. (2016) for further details on the different image classes and example images. The instances were chosen to cover different types of images, while still allowing for the simulation of a large variety of parameters for subsampling.

The point cloud type instances were created as follows: The support points of the measures are independently, uniformly distributed on $[0,1]^D$. The number of points N was chosen 32^2, 64^2 and 128^2 in order to match the size of the grid based instances. For each choice of D and N, three instances were generated with regards to the three images types used in the grid based case. Two measures on the points are drawn from the Dirichlet distribution with all parameters equal to one. That is, the masses on different points are independent of each other, similar to the white noise images. To create point cloud versions of the Cauchy Density and Classic Images classes the grayscale values of the same images were used to get the mass values for the support points. In three and four dimensions, the product measure of the images with their sum of columns and with themselves, respectively, was used.

All original instances were solved by each back-end solver in each resolution for the values $p = 1$, $p = 2$, and $p = 3$ in order to be compared to the approximative results for the subsamples in terms of runtime and accuracy, with the exception of CPLEX, where the 128×128 instances could not be solved due to memory limitations. Algorithm 1 was applied to each of these instances with parameters $S \in \{100, 500, 1000, 2000, 4000\}$ and $B \in \{1, 2, 5\}$. For every combination of instance and parameters, the subsampling algorithm was run 5 times in order to mitigate the randomness of the results.

Since the linear programming solvers had a very similar performance on the grid based instances (see below), only one of them–the transportation simplex—was tested on the point cloud instances.

4.2. Computational Results

As mentioned before, all results of Algorithm 1 are relative to the results of the methods applied to the original problems. We are mainly interested in the reduction in runtime and accuracy of the returned values. Many important results can be observed in Figures 2 and 3. The points in the diagram represent averages over the different methods, instances, and multiple tries, but are separated in resolution and choices of the parameters S and B in Algorithm 1.

For images we observe a decrease in relative runtimes with higher resolution, while the average relative error is independent of the image resolution. In the point cloud case, however, the relative error increases slightly with the instance size. The number S of sampled points seems to considerably affect the relative error. An increase of the number of points results in more accurate values, with average relative errors as low as about 3% for $S = 4000$, while still maintaining a speedup of two orders of magnitude on 128×128 images. Lower sample sizes yield higher average errors, but also

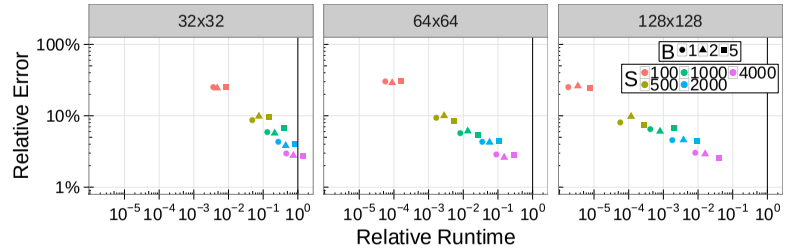


Figure 2: Relative errors $|\hat{W}_p^{(S)}(\boldsymbol{r}, \boldsymbol{s}) - W_p(\boldsymbol{r}, \boldsymbol{s})|/W_p(\boldsymbol{r}, \boldsymbol{s})$ vs. relative runtimes $\hat{t}/t$ for different parameters S and B and different problem sizes for images. $\hat{t}$ is the runtime of Algorithm 1 and t is the runtime of the respective back-end solver without subsampling.

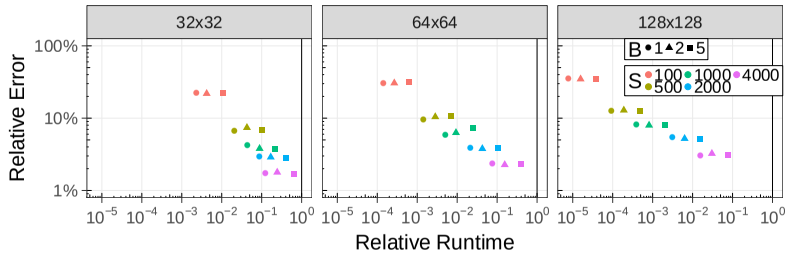


Figure 3: Relative errors vs. relative runtimes for different parameters S and B and different problem sizes for point clouds. The number of support points matches the number of pixels in the images.

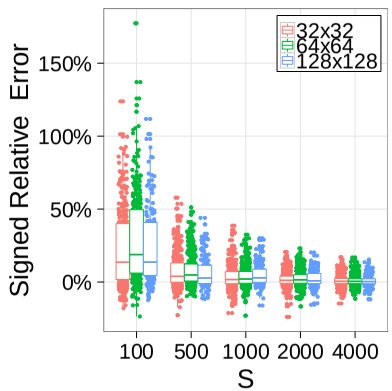


Figure 4: The signed relative approximation error $\left(\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})\right)/W_p(\boldsymbol{r},\boldsymbol{s})$ showing that the approximation overestimates the exact distance for small S but the bias vanishes for larger S.

lower runtimes. With $S = 500$ the runtime is reduced by over four orders of magnitude with an average relative error of less than 10%. As to be expected, runtime increases linearly with the number of repetitions B. However, the impact on the relative errors is rather inconsistent. This is due to the fact, that the costs returned by the subsampling algorithm are often overestimated, therefore averaging over multiple tries does not yield improvements (see Figure 4). This means that in order to increase the accuracy of the algorithm it is advisable to keep $B = 1$ and instead increase the sample size S. However, increasing B can be useful to lower the variability of the results.

On the contrary, there is a big difference in accuracy between the image classes. While Algorithm 1 has consistently low relative errors on the Cauchy Density images, the exact optimal costs for White Noise images cannot be approximated as reliably. The relative errors fluctuate more and are generally much higher, as one can see from Figure 5 (left). In images with smooth structures and regular features the subsamples are able to capture that structure and therefore deliver a more precise representation of the images and a more precise value. This is not possible in images that are very irregular or noisy, such as the White Noise images, which have no structure to begin with. The Classic Images contain both regular structures and more irregular regions, therefore their relative errors are slightly higher than in the Cauchy Density cases. The algorithm has a similar performance on the point cloud instances, that are modelled after the Cauchy Density and Classic Images classes, while the Dirichlet instances have a more desirable accuracy compared to the White Noise images, as seen in Figure 5 (right).

There are no significant differences in performance between the different back-end solvers for the Wasserstein distance. As Figure 6 shows, accuracy seems to be better for the Sinkhorn distance compared to the other three solvers which report the exact Wasserstein distance.

In the results of the point cloud instances we can observe the influence of the value $p' = (D/2) - p$ on the scaling of the relative error with the instance size N for constant sample size ($S = 4000$). This is shown in Figure 7. We observe an increase of the relative error with p', as expected from the theory. However, we are not able to clearly distinguish between the three cases $p' < 0$, $p' = 0$

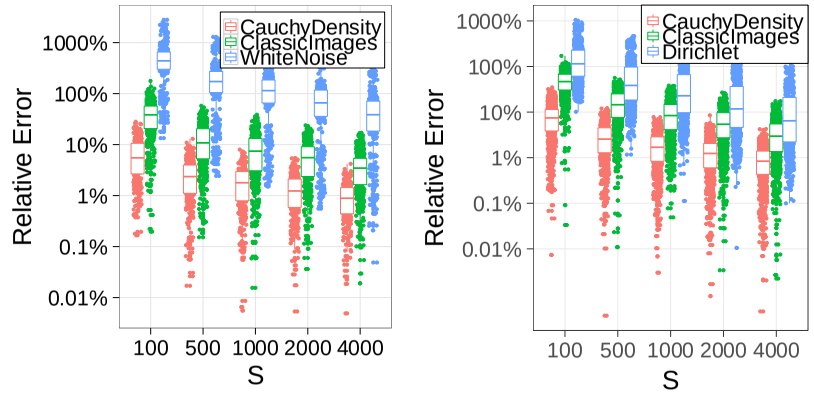


Figure 5: A comparison of the relative errors for different image classes (left) and point cloud instance classes (right).

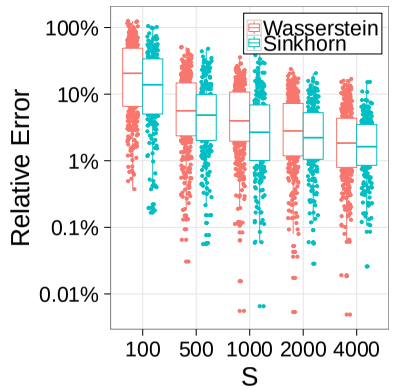


Figure 6: A comparison between the approximations of the Wasserstein and Sinkhorn distances.

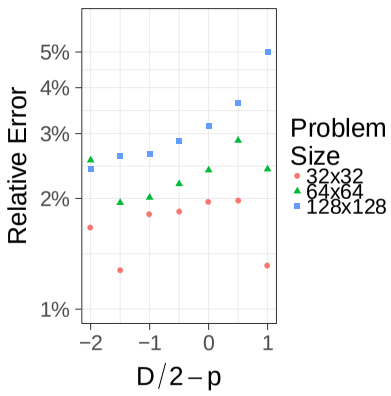


Figure 7: A comparison of the mean relative errors in the point cloud instances with sample size $S = 4000$ for different values of $p' = (D/2) - p$.

and $p' > 0$. This might be due to the relatively small instance sizes N in the experiments. While we see that the relative errors are independent of N in the image case (compare Figure 2), for the point clouds N has an influence on the accuracy that depends on p'.

5. Discussion

As our simulations demonstrate, subsampling is a simple yet powerful tool to obtain good approximations to Wasserstein distances with only a small fraction of required runtime and memory. It is especially remarkable that in the case of two dimensional images for a fixed amount of subsampled points, and therefore a fixed amount of time and memory, the relative error is independent of the resolution/size of the images. Based on these results, we expect the subsampling algorithm to return similarly precise results with even higher resolutions of the images it is applied to, while the effort to obtain them stays the same. Even in point cloud instances the relative error only scales mildly with the original input size N and is dependent on the value p'.

The numerical results (Figure 2) show an inverse polynomial decrease of the approximation error with S, in accordance with the theoretical results. In fact, the rate $\mathcal{O}(S^{-1/2p})$ is optimal. Indeed, when $\boldsymbol{r} = \boldsymbol{s}$ (are nontrivial measures), Sommerfeld and Munk (2018) show that $Z_S = S^{1/2p}[W_p(\hat{\boldsymbol{r}}_S, \hat{\boldsymbol{s}}_S) - W_p(\boldsymbol{r}, \boldsymbol{s})]$ has a nondegenerate limiting distribution Z. For each $R > 0$ the function $x \mapsto \min(R, |x|)$ is nonnegative, continuous and bounded, so

$$\liminf_{S\to\infty} E\{S^{1/2p}|W_p(\hat{\boldsymbol{r}}_S, \hat{\boldsymbol{s}}_S) - W_p(\boldsymbol{r}, \boldsymbol{s})|\} = \liminf_{S\to\infty} E\{|Z_S|\} \geq \liminf_{S\to\infty} E\min\{R, |Z_S|\} = E\min(R, |Z|).$$

Letting $R \to \infty$ and using the monotone convergence theorem yields

$$\liminf_{S\to\infty} E\{S^{1/2p}|W_p(\hat{\boldsymbol{r}}_S, \hat{\boldsymbol{s}}_S) - W_p(\boldsymbol{r}, \boldsymbol{s})|\} \geq E|Z| > 0.$$

When applying the algorithm, it is important to note that the quality of the returned values depends on the structure of the data. In very irregular instances it is necessary to increase the

sample size in order to obtain similarly precise results, while in regular structures a small sample size suffices.

Our scheme allows the parameters S and B to be easily tuned towards faster runtimes or more precise results, as desired. Increases and decreases of the sample size S will increase/decrease the mean approximation of W_p by $\hat{W}_p^{(S)}$, while B will only affect the concentration around $E\hat{W}_p^{(S)}$. Empirically, we found that for fixed computational cost, the best performance is achieved when $B = 1$ (compare Figure 2), suggesting that the bias is more dominant than the variance in the mean squared error.

The scheme presented here can readily be applied to other optimal transport distances, as long as a solver is available, as we demonstrated with the Sinkhorn distance (Cuturi, 2013). Empirically, we can report good performance in this case, suggesting that entropically regularized distances might be even more amenable to subsampling approximation than the Wasserstein distance itself. Extending the theoretical results to this case would require an analysis of the mean speed of convergence of empirical Sinkhorn distances, which is an interesting task for future research.

All in all, subsampling proves to be a general, powerful and versatile tool that can be used with virtually any optimal transport solver as back-end and has both theoretical approximation error guarantees, and a convincing performance in practice. It is a challenge to extend this method in a way which is specifically tailored to the geometry of the underlying space $\mathcal{X}$, which may result in further improvements.

Acknowledgments

Jörn Schrieber and Max Sommerfeld acknowledge support by the German Research Foundation (DFG) through the RTG 2088. Yoav Zemel is supported by Swiss National Science Foundation Grant #178220. We thank an Associate Editor and three reviewers for insightful comments on previous versions of this work.

Appendix A. Proofs

This Appendix contains proofs of all our theoretical results.

A.1. Proof of Theorem 1

Proof strategy The method used in this proof has been employed before to bound the mean rate of convergence of the empirical Wasserstein distance on a general metric space $(\mathcal{X}, d)$ (Boissard and Le Gouic, 2014; Fournier and Guillin, 2015). In essence, it constructs a tree on the space $\mathcal{X}$ and bounds the Wasserstein distance with some transport metric in the tree, which can either be computed explicitly or bounded easily (see also Heinrich and Kahn, 2018, who use a coarse-graining tree in order to bound the Wasserstein distance in the context of mixture models). Our construction is specifically tailored to finite spaces, and allows to obtain a better dependence on $N = |\mathcal{X}|$ in Theorem 3 while preserving the rate $S^{-1/2}$.

More precisely, in our case of finite spaces, let $\mathcal{T}$ be a spanning tree on $\mathcal{X}$ (that is, a tree with vertex set $\mathcal{X}$ and edge lengths given by the metric d on $\mathcal{X}$) and $d_{\mathcal{T}}$ the metric on $\mathcal{X}$ defined by the path lengths in the tree. Clearly, the tree metric $d_{\mathcal{T}}$ dominates the original metric d on $\mathcal{X}$ and hence $W_p(\boldsymbol{r}, \boldsymbol{s}) \leq W_p^{\mathcal{T}}(\boldsymbol{r}, \boldsymbol{s})$ for all $\boldsymbol{r}, \boldsymbol{s} \in \mathcal{P}(\mathcal{X})$, where $W_p^{\mathcal{T}}$ denotes the Wasserstein distance evaluated with respect to the tree metric. The goal is now to bound $E\left[(W_p^{\mathcal{T}}(\hat{\boldsymbol{r}}_S, \boldsymbol{r}))^p\right]$. We refer to Tameling and Munk (2018) for examples and comparisons of different spanning trees on two-dimensional grids.

Assume $\mathcal{T}$ is rooted at $\mathrm{root}(\mathcal{T}) \in \mathcal{X}$. Then, for $x \in \mathcal{X}$ and $x \neq \mathrm{root}(\mathcal{T})$ we may define $\mathrm{par}(x) \in \mathcal{X}$ as the immediate neighbor of x in the unique path connecting x and $\mathrm{root}(\mathcal{T})$. We set $\mathrm{par}(\mathrm{root}(\mathcal{T})) = \mathrm{root}(\mathcal{T})$. We also define $\mathrm{children}(x)$ as the set of vertices $x' \in \mathcal{X}$ such that there exists a sequence

$x' = x_1, \dots, x_l = x \in \mathcal{X}$ with $\mathrm{par}(x_j) = x_{j+1}$ for $j = 1, \dots, l-1$. Note that with this definition $x \in \mathrm{children}(x)$. Additionally, define the linear operator $S_{\mathcal{T}} \colon \mathbb{R}^{\mathcal{X}} \to \mathbb{R}^{\mathcal{X}}$

$$(S_{\mathcal{T}} \boldsymbol{u})_x = \sum_{x' \in \mathrm{children}(x)} u_{x'}. \tag{10}$$

Building the tree We build a q-ary tree on $\mathcal{X}$. To this end, we split $\mathcal{X}$ to $l_{\max} + 2$ groups and build the tree in such a way that a node at level $l+1$ has a unique parent at level l with edge length q^{-l}. The formal construction follows.

For $l \in \{0, \dots, l_{\max}\}$ we let $Q_l \subset \mathcal{X}$ be the center points of a $q^{-l} \operatorname{diam}(\mathcal{X})$ covering of $\mathcal{X}$, that is

$$\bigcup_{x \in Q_l} B(x, q^{-l} \operatorname{diam}(\mathcal{X})) = \mathcal{X}, \text{ and } |Q_l| = \mathcal{N}(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X})),$$

where $B(x, \epsilon) = \{x' \in \mathcal{X} : d(x, x') \le \epsilon\}$. Additionally set $Q_{l_{\max}+1} = \mathcal{X}$. Now define $\tilde{Q}_l = Q_l \times \{l\}$; we will build a tree structure on $\cup_{l=0}^{l_{\max}+1} \tilde{Q}_l$.

Since we must have $|\tilde{Q}_0| = 1$ we can take this element as the root. Assume now that the tree already contains all elements of $\cup_{j=0}^{l} \tilde{Q}_j$. Then, we add to the tree all elements of $\tilde{Q}_{l+1}$ by choosing for $(x, l+1) \in \tilde{Q}_{l+1}$ (exactly one) parent element $(x', l) \in \tilde{Q}_l$ such that $d(x, x') \le q^{-l} \operatorname{diam}(\mathcal{X})$. This is possible, since Q_l is a $q^{-l} \operatorname{diam}(\mathcal{X})$ covering of $\mathcal{X}$. We set the length of the connecting edge to $q^{-l} \operatorname{diam}(\mathcal{X})$.

In this fashion we obtain a spanning tree $\mathcal{T}$ of $\cup_{l=0}^{l_{\max}+1} \tilde{Q}_l$ and a partition $\{\tilde{Q}_l\}_{l=0,\dots,l_{\max}+1}$. About this tree we know that:

- it is in fact a tree. First, it is connected, because the construction starts with one connected component and in every subsequent step all additional vertices are connected to it. Second, it contains no cycles. To see this let $((x_1, l_1), \dots, (x_K, l_K))$ be a cycle in $\mathcal{T}$. Without loss of generality we may assume $l_1 = \min\{l_1, \dots, l_K\}$. Then, (x_1, l_1) must have at least two edges connecting it to vertices in a $\tilde{Q}_l$ with $l \ge l_1$ which is impossible by construction.
- $|\tilde{Q}_l| = \mathcal{N}(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}))$ for $0 \le l \le l_{\max}$.
- $d(x, \mathrm{par}(x)) = q^{-l+1} \operatorname{diam}(\mathcal{X})$ whenever $x \in \tilde{Q}_l$, $l \ge 1$.
- $d(x, x') \le d_{\mathcal{T}}((x, l_{\max}+1), (x', l_{\max}+1))$.

Since the leaves of $\mathcal{T}$ can be identified with $\mathcal{X}$, a measure $\boldsymbol{r} \in \mathcal{P}(\mathcal{X})$ canonically defines a probability measure $\boldsymbol{r}^{\mathcal{T}} \in \mathcal{P}(\mathcal{T})$ for which $r^{\mathcal{T}}_{(x, l_{\max}+1)} = r_x$ and $r^{\mathcal{T}}_{(x,l)} = 0$ for $l \le l_{\max}$. In slight abuse of notation we will denote the measure $\boldsymbol{r}^{\mathcal{T}}$ simply by $\boldsymbol{r}$. With this notation, we have $W_p(\boldsymbol{r}, \boldsymbol{s}) \le W_p^{\mathcal{T}}(\boldsymbol{r}, \boldsymbol{s})$ for all $\boldsymbol{r}, \boldsymbol{s} \in \mathcal{P}(\mathcal{X})$.

Wasserstein distance on trees Note also that $\mathcal{T}$ is *ultra-metric*, that is, all its leaves are at the same distance from the root. For trees of this type, we can define a height function $h : \mathcal{X} \to [0, \infty)$ such that $h(x) = 0$ if $x \in \mathcal{X}$ is a leaf and $h(\mathrm{par}(x)) - h(x) = d_{\mathcal{T}}(x, \mathrm{par}(x))$ for all $x \in \mathcal{X} \setminus \mathrm{root}(\mathcal{X})$. There is an explicit formula for the Wasserstein distance on ultra-metric trees (Kloeckner, 2015). Indeed, if $\boldsymbol{r}, \boldsymbol{s} \in \mathcal{P}(\mathcal{X})$ then

$$(W_p^{\mathcal{T}}(\boldsymbol{r}, \boldsymbol{s}))^p = 2^{p-1} \sum_{x \in \mathcal{X}} (h(\mathrm{par}(x))^p - h(x)^p) \left| (S_{\mathcal{T}} \boldsymbol{r})_x - (S_{\mathcal{T}} \boldsymbol{s})_x \right|, \tag{11}$$

with the operator $S_{\mathcal{T}}$ as defined in (10). For the tree $\mathcal{T}$ constructed above and $x \in \tilde{Q}_l$ with $l = 0, \dots, l_{\max}$ we have

$$h(x) = \sum_{j=l}^{l_{\max}} q^{-j} \operatorname{diam}(\mathcal{X}),$$

and therefore $\operatorname{diam}(\mathcal{X}) q^{-l} \le h(x) \le 2 \operatorname{diam}(\mathcal{X}) q^{-l}$. This yields

$$(h(\mathrm{par}(x))^p - (h(x))^p) \le (\operatorname{diam}(\mathcal{X}))^p q^{-(l-2)p}.$$

Then (11) yields

$$E\left[W_p^p(\hat{\boldsymbol{r}}_S, \boldsymbol{r})\right] \le 2^{p-1} q^{2p} (\operatorname{diam}(\mathcal{X}))^p \sum_{l=0}^{l_{\max}+1} q^{-lp} \sum_{x \in \tilde{Q}_l} E|(S_{\mathcal{T}} \hat{\boldsymbol{r}}_S)_x - (S_{\mathcal{T}} \boldsymbol{r})_x|.$$

Since $(S_{\mathcal{T}} \hat{\boldsymbol{r}}_S)_x$ is the mean of S i.i.d. Bernoulli variables with expectation $(S_{\mathcal{T}} \boldsymbol{r})_x$ we have

$$\begin{aligned} \sum_{x \in \tilde{Q}_l} E|(S_{\mathcal{T}} \hat{\boldsymbol{r}}_S)_x - (S_{\mathcal{T}} \boldsymbol{r})_x| &\le \sum_{x \in \tilde{Q}_l} \sqrt{\frac{(S_{\mathcal{T}} \boldsymbol{r})_x (1 - (S_{\mathcal{T}} \boldsymbol{r})_x)}{S}} \\ &\le \frac{1}{\sqrt{S}} \left(\sum_{x \in \tilde{Q}_l} (S_{\mathcal{T}} \boldsymbol{r})_x \right)^{1/2} \left(\sum_{x \in \tilde{Q}_l} (1 - (S_{\mathcal{T}} \boldsymbol{r})_x) \right)^{1/2} \le \sqrt{|\tilde{Q}_l| / S}, \end{aligned}$$

using Hölder's inequality and the fact that $\sum_{x \in \tilde{Q}_l} (S_{\mathcal{T}} \boldsymbol{r})_x = 1$ for all $l = 0, \dots, l_{\max}+1$. This finally yields

$$\begin{aligned} E\left[W_p^p(\hat{\boldsymbol{r}}_S, \boldsymbol{r})\right] &\le 2^{p-1} q^{2p} (\operatorname{diam}(\mathcal{X}))^p \left(q^{-(l_{\max}+1)p} \sqrt{N} + \sum_{l=0}^{l_{\max}} q^{-lp} \sqrt{\mathcal{N}(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}))} \right) / \sqrt{S} \\ &\le \mathcal{E}_q(\mathcal{X}, p) / \sqrt{S}. \end{aligned}$$

□

Covering by arbitrary sets We now explain how to obtain the second formula for $\mathcal{E}_q$ as stated in Remark 2. The idea is to define the coverings with arbitrary sets, not necessarily balls. Let

$$\mathcal{N}_1(\mathcal{X}, \delta) = \inf\{m : \exists A_1, \dots, A_m \subseteq \mathcal{X},\ \operatorname{diam}(A_i) \le 2\delta, \cup A_i \supseteq \mathcal{X}\}.$$

Since balls satisfy the diameter condition, $\mathcal{N}_1 \le \mathcal{N}$. Furthermore, if $\mathcal{X}' \supseteq \mathcal{X}$, then $\mathcal{N}_1(\mathcal{X}, \delta) \le \mathcal{N}_1(\mathcal{X}', \delta)$, which is not the case for $\mathcal{N}$. For example, let $\mathcal{X} = \{-1, 1\} \subset \{-1, 0, 1\} = \mathcal{X}'$ and observe that

$$\mathcal{N}_1(\mathcal{X}, 1) = 1 = \mathcal{N}_1(\mathcal{X}', 1), \qquad \text{but} \qquad \mathcal{N}(\mathcal{X}, 1) = 2 > 1 = \mathcal{N}(\mathcal{X}', 1).$$

The tree construction with respect to the new covering numbers is done in a similar manner. For each $0 \le l \le l_{\max}$ let Q'_l be a collection of disjoint sets of diameter $2q^{-l} \operatorname{diam}(\mathcal{X})$ that cover $\mathcal{X}$ and $|Q'_l| = \mathcal{N}_1(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}))$. Let $Q_l = \{x_1, \dots, x_{|Q'_l|}\} \subseteq \mathcal{X}$ be an arbitrary collection of representatives from the sets in Q'_l. Such representatives exist by minimality of $|Q'_l|$ and they are different by the disjoint nature of Q'_l. Additionally set $Q_{l_{\max}+1} = \mathcal{X}$. Construct the tree in the same way, except that now we only have the bound $d(x, x') \le 2q^{-l} \operatorname{diam}(\mathcal{X})$ for $(x, l+1) \in \tilde{Q}_{l+1}$ and a corresponding $(x, l) \in \tilde{Q}_l$, so we need to set the edge length to be $2q^{-l} \operatorname{diam}(\mathcal{X})$, twice as much as in the original construction. The proof then goes in the same way, with an extra factor 2^p. We obtain an alternative bound

$$\mathcal{E}_q = 2^{2p-1} q^{2p} (\operatorname{diam}(\mathcal{X}))^p \left(q^{-(l_{\max}+1)p} \sqrt{N} + \sum_{l=0}^{l_{\max}} q^{-lp} \sqrt{\mathcal{N}_1(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}))} \right).$$

In comparison with (6), we replaced $\mathcal{N}$ by $\mathcal{N}_1$. The price to pay for this is an additional factor of 2^p. □

A.2. Proof of Theorem 3

We may assume without loss of generality that $\mathcal{X} \subseteq [0, \operatorname{diam}(\mathcal{X})]^D$. The covering numbers of the cube with Euclidean balls behave badly in high dimensions, so it will prove useful to replace the Euclidean norm by the infinity norm $\|x\|_\infty = \max_i |x_i|$, $x = (x_1, \ldots, x_D) \in \mathbb{R}^D$. With this norm we have $\mathcal{N}([0, \operatorname{diam}(\mathcal{X})]^D, \epsilon \operatorname{diam}(\mathcal{X}), \|\cdot\|_\infty) \leq (\lceil 1/(2\epsilon) \rceil)^D$. If q is an integer, then

$$\mathcal{N}(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}), \|\cdot\|_\infty) \leq \mathcal{N}([0, \operatorname{diam}(\mathcal{X})]^D, q^{-l} \operatorname{diam}(\mathcal{X})/2, \|\cdot\|_\infty) \leq \lceil q^l \rceil^D = q^{lD}.$$

This yields

$$\sum_{l=0}^{l_{\max}} q^{-lp} \sqrt{\mathcal{N}(\mathcal{X}, q^{-l} \operatorname{diam}(\mathcal{X}))} \leq \sum_{l=0}^{l_{\max}} q^{l(D/2-p)} = \begin{cases} (1 - q^{(l_{\max}+1)(D/2-p)})/(1 - q^{D/2-p}) & D \neq 2p, \\ l_{\max} + 1 & D = 2p. \end{cases}$$

Denote for brevity $p' = D/2 - p$ and plug this into (6) to bound $S^{1/2} E\left[W_p^p(\hat{\boldsymbol{r}}_S, \boldsymbol{r}, \|\cdot\|_\infty)\right]$ by

$$2^{p-1} q^{2p} (\operatorname{diam}(\mathcal{X}))^p \left[q^{-p(l_{\max}+1)} \sqrt{N} + \begin{cases} (1 - q^{(l_{\max}+1)p'})/(1 - q^{p'}) & p' \neq 0, \\ l_{\max} + 1 & p' = 0. \end{cases} \right]$$

If $p' < 0$, then let $l_{\max} \to \infty$. Otherwise, choose $l_{\max} = \lfloor D^{-1} \log_q N \rfloor$ (giving the best dependence on N), so that the element inside the square brackets is smaller than

$$\begin{cases} 1/(1 - q^{p'}) & p' < 0, \\ 2 + D^{-1} \log_q N & p' = 0, \\ N^{1/2-p/D} + (N^{1/2-p/D} q^{p'} - 1)/(q^{p'} - 1) & p' > 0 \end{cases} \leq \begin{cases} 1/(1 - q^{p'}) & p' < 0, \\ 2 + D^{-1} \log_q N & p' = 0, \\ (2q^{p'} - 1) N^{1/2-p/D}/(q^{p'} - 1) & p' > 0. \end{cases} \tag{12}$$

The right-hand side is $C_{D,p}(N)$ for $q = 2$. To get back to the Euclidean norm use $\|a\|_2 \leq \|a\|_\infty \sqrt{D}$, so that

$$E\left[W_p^p(\hat{\boldsymbol{r}}_S, \boldsymbol{r})\right] \leq D^{p/2} E\left[W_p^p(\hat{\boldsymbol{r}}_S, \boldsymbol{r}, \|\cdot\|_\infty)\right] \leq D^{p/2} 2^{p-1} q^{2p} (\operatorname{diam}(\mathcal{X}))^p C_{D,p}(N)/\sqrt{S},$$

which is the desired conclusion. □

Lemma 1. *(a) Let $\tilde{C}_{D,p}(q, N)$ denote the right-hand side of (12). Then the minimum of the function $q \mapsto q^{2p} \tilde{C}_{D,p}(q, N)$ on $[2, \infty)$ is attained at $q = 2$.*

(b) Let $q \geq 2$, p, D integers, and $p' = D/2 - p$. If $p' < 0$, then $1/(1 - q^{p'}) \leq 2 + \sqrt{2}$ and if $p' > 0$, then $2 + 1/(q^{p'} - 1) \leq 3 + \sqrt{2}$.

Proof. We begin with (b). If $p' < 0$ then $1/(1 - q^{p'})$ is decreasing in q and increasing in p'. The integer constraints on D and p imply that the maximal value p' can attain is -0.5. The smallest value q can attain is 2. Thus

$$1/(1 - q^{p'}) \leq 1/(1 - 2^{-0.5}) = \frac{\sqrt{2}}{\sqrt{2} - 1} = \sqrt{2}(\sqrt{2} + 1) = 2 + \sqrt{2}.$$

When $p' > 0$ the term $2 + 1/(q^{p'} - 1)$ is decreasing in $p' \geq 0.5$ and in $q \geq 2$, so it is bounded by

$$2 + 1/(\sqrt{2} - 1) = 3 + \sqrt{2}.$$

To prove (a) we shall differentiate the function $q^{2p} \tilde{C}_{D,p}(q, N)$ with respect to q and show that the derivative is positive for all $q \geq 2$, and $p, D, N \geq 1$.

For negative p' consider the function

$$f_1(q) = \frac{q^{2p}}{1 - q^{p'}}, \qquad q \geq 2; p \geq 1; p' < 0.$$

Its derivative is

$$f_1'(q) = \frac{2pq^{2p-1}(1 - q^{p'}) + p' q^{p'-1} q^{2p}}{(1 - q^{p'})^2} = \frac{q^{2p-1}}{1 - q^{p'}} \left[2p + \frac{p' q^{p'}}{1 - q^{p'}} \right].$$

It suffices to show that the term in square brackets is positive, since $1 - q^{p'} > 0$. Let us bound $q^{p'}$ and the denominator $(1 - q^{p'})^{-1}$. Since $e^x \geq 1 + x$ for $x \geq 0$, $e^{-x} \leq 1/(1 + x)$ and setting $x = -p' \log q$ gives

$$q^{p'} = e^{p' \log q} \leq \frac{1}{1 - p' \log q}.$$

Hence

$$1 - q^{p'} \geq 1 - \frac{1}{1 - p' \log q} = \frac{1 - p' \log q - 1}{1 - p' \log q} = \frac{-p' \log q}{1 - p' \log q},$$

so that

$$\frac{q^{p'}}{1 - q^{p'}} \leq \frac{1}{1 - p' \log q} \frac{1 - p' \log q}{-p' \log q} = \frac{1}{-p' \log q}.$$

Conclude that, since $p' < 0$,

$$2p + \frac{p' q^{p'}}{1 - q^{p'}} \geq 2p + p' \frac{1}{-p' \log q} = 2p + \frac{1}{-\log q} = 2p - \frac{1}{\log q} \geq 2p - \frac{1}{\log 2} \geq 2 - \frac{1}{\log 2} > 0.$$

For $p' = 0$ consider the function

$$f_2(q) = q^{2p}(2 + D^{-1} \log_q N) = 2q^{2p} + \frac{q^{2p} \log N}{D \log q}, \qquad q \geq 2; D = 2p \geq 2.$$

Its derivative is

$$f_2'(q) = 4pq^{2p-1} + \frac{\log N}{D(\log q)^2} \left[2pq^{2p-1} \log q - q^{-1} q^{2p} \right] = q^{2p-1} \left[4p + \frac{\log N}{D(\log q)^2} (2p \log q - 1) \right] > 0$$

since $2p \log q \geq 2 \log 2 > 1$.

For $p' > 0$ consider the function

$$f_3(q) = q^{2p}[2 + 1/(q^{p'} - 1)] = 2q^{2p} + \frac{q^{2p}}{q^{p'} - 1} = 2q^{2p} - f_1(q), \qquad q \geq 2; p \geq 1; p' > 0.$$

The derivative is

$$4pq^{2p-1} - \frac{q^{2p-1}}{1 - q^{p'}} \left[2p + \frac{p' q^{p'}}{1 - q^{p'}} \right] = 4pq^{2p-1} + \frac{q^{2p-1}}{q^{p'} - 1} \left[2p - \frac{p' q^{p'}}{q^{p'} - 1} \right].$$

This function is more complicated and we need to split into cases according to small, large or moderate values of p'.

Case 1: $p' \leq 0.5$**.** Then the negative term can be bounded using $q^{p'} - 1 \geq p' \log q$ as

$$\frac{p' q^{p'}}{q^{p'} - 1} = p' + \frac{p'}{q^{p'} - 1} \leq p' + \frac{1}{\log q} \leq p' + \frac{1}{\log 2} \leq 0.5 + \frac{1}{\log 2} < 2 \leq 2p.$$

Thus $f_3'(q) \geq 0$ in this case.

To deal with larger values of p' rewrite the derivative as

$$q^{2p-1}\left[4p + \frac{2p}{q^{p'}-1} - \frac{p'q^{p'}}{(q^{p'}-1)^2}\right],$$

and bound the negative part:

$$\frac{p'q^{p'}}{(q^{p'}-1)^2} = \frac{p'}{q^{p'}-1} + \frac{p'}{(q^{p'}-1)^2} \leq \frac{1}{\log q} + \frac{1}{(q^{p'}-1)\log q}.$$

Case 2: $p' \geq 1$**.** Then $q^{p'}-1 \geq 1$ so this is smaller than

$$\frac{1}{\log 2} + \frac{1}{\log 2} = \frac{2}{\log 2} < 4 \leq 4p.$$

Hence the derivative is positive in this case.

Case 3: $p' \geq 1/2$ **and** $q \geq e$**.** Then this is smaller than

$$1 + \frac{1}{e^{1/2}-1} \leq 1 + \frac{1}{\sqrt{2}-1} = 2+\sqrt{2} < 4 \leq 4p.$$

Hence the derivative is positive in this case.

Case 4: $q \leq e$ **and** $p' \in [1/2, 1]$**.** The negative term is bounded above by

$$\frac{1}{\log q} + \frac{1}{(q^{p'}-1)\log q} \leq \frac{1}{\log 2} + \frac{1}{(q^{p'}-1)\log 2} \leq \frac{1}{\log 2} + \frac{1}{(\sqrt{2}-1)\log 2} = \frac{2+\sqrt{2}}{\log 2} \approx 4.93,$$

whereas the positive term can be bounded below as

$$4p + \frac{2p}{q^{p'}-1} \geq 4 + \frac{2}{e-1} \approx 5.16 > 4.93.$$

This completes the proof. □

A.3. Proof of Theorem 4

We introduce some additional notation. For $(x,y),(x',y') \in \mathcal{X}^2$ we set

$$d_{\mathcal{X}^2}((x,y),(x',y')) = \{d^p(x,x') + d^p(y,y')\}^{1/p}$$

We further define the function $Z : (\mathcal{X}^2)^{SB} \to \mathbb{R}$ via

$$((x_{11},y_{11}),\ldots,(x_{SB},y_{SB})) \mapsto \frac{1}{B}\sum_{i=1}^{B}\left[W_p\left(\frac{1}{S}\sum_{j=1}^{S}\delta_{x_{ji}}, \frac{1}{S}\sum_{j=1}^{S}\delta_{y_{ji}}\right) - W_p(\boldsymbol{r},\boldsymbol{s})\right].$$

Since $W_p^p(\cdot,\cdot)$ is jointly convex (Villani, 2008, Theorem 4.8),

$$W_p\left(\frac{1}{S}\sum_{j=1}^{S}\delta_{x_j}, \frac{1}{S}\sum_{j=1}^{S}\delta_{y_j}\right) \leq \left\{\frac{1}{S}\sum_{j=1}^{S} W_p^p(\delta_{x_j},\delta_{y_j})\right\}^{1/p} = S^{-1/p}\left\{\sum_{j=1}^{S} d^p(x_j,y_j)\right\}^{1/p}.$$

Our first goal is to show that Z is Lipschitz continuous. Let $((x_{11},y_{11}),\ldots,(x_{SB},y_{SB}))$ and $((x'_{11},y'_{11}),\ldots,(x'_{SB},y'_{SB}))$ be arbitrary elements of $(\mathcal{X}^2)^{SB}$. Then, using the reverse triangle inequality and the relations above

$$\begin{aligned}
&|Z((x_{11},y_{11}),\ldots,(x_{SB},y_{SB})) - Z((x'_{11},y'_{11}),\ldots,(x'_{SB},y'_{SB}))| \\
&\leq \frac{1}{B}\sum_{i=1}^{B}\left|W_p\left(\frac{1}{S}\sum_{j=1}^{S}\delta_{x_{ji}}, \frac{1}{S}\sum_{j=1}^{S}\delta_{y_{ji}}\right) - W_p\left(\frac{1}{S}\sum_{j=1}^{S}\delta_{x'_{ji}}, \frac{1}{S}\sum_{j=1}^{S}\delta_{y'_{ji}}\right)\right| \\
&\leq \frac{1}{B}\sum_{i=1}^{B}\left[W_p\left(\frac{1}{S}\sum_{j=1}^{S}\delta_{x_{ji}}, \frac{1}{S}\sum_{j=1}^{S}\delta_{x'_{ji}}\right) + W_p\left(\frac{1}{S}\sum_{j=1}^{S}\delta_{y_{ji}}, \frac{1}{S}\sum_{j=1}^{S}\delta_{y'_{ji}}\right)\right] \\
&\leq \frac{S^{-1/p}}{B}\sum_{i=1}^{B}\left[\left\{\sum_{j=1}^{S} d^p(x_{ji},x'_{ji})\right\}^{1/p} + \left\{\sum_{j=1}^{S} d^p(y_{ji},y'_{ji})\right\}^{1/p}\right] \\
&\leq \frac{S^{-1/p}}{B}(2B)^{\frac{p-1}{p}}\left\{\sum_{i,j} d^p_{\mathcal{X}^2}((x_{ji},y_{ji}),(x'_{ji},y'_{ji}))\right\}^{1/p}.
\end{aligned}$$

Hence, $Z/2$ is Lipschitz continuous with constant $(SB)^{-1/p}$ relative to the p-metric generated by $d_{\mathcal{X}^2}$ on $(\mathcal{X}^2)^{SB}$.

For $\tilde{\boldsymbol{r}} \in \mathcal{P}(\mathcal{X}^2)$ let $H(\cdot \mid \tilde{\boldsymbol{r}})$ denote the relative entropy with respect to $\tilde{\boldsymbol{r}}$. Since $\mathcal{X}^2$ has $d_{\mathcal{X}^2}$-diameter $2^{1/p}\operatorname{diam}(\mathcal{X})$, we have by Bolley and Villani (2005, Particular case 2.5, page 337) that for every $\tilde{\boldsymbol{s}}$

$$W_p(\tilde{\boldsymbol{r}},\tilde{\boldsymbol{s}}) \leq \left(8\operatorname{diam}(\mathcal{X})^{2p}H(\tilde{\boldsymbol{r}} \mid \tilde{\boldsymbol{s}})\right)^{1/2p}. \tag{13}$$

If $X_{11},\ldots,X_{SB} \sim \boldsymbol{r}$ and $Y_{11},\ldots,Y_{SB} \sim \boldsymbol{s}$ are all independent, we have

$$Z((X_{11},Y_{11}),\ldots,(X_{SB},Y_{SB})) \sim \hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s}).$$

The Lipschitz continuity of Z and the transportation inequality (13) yields a concentration result for this random variable. In fact, by Gozlan and Léonard (2007, Lemma 6) we have

$$P\left[\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s}) \geq E\left[\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})\right] + z\right] \leq \exp\left(\frac{-SBz^{2p}}{8\operatorname{diam}(\mathcal{X})^{2p}}\right).$$

for all $z \geq 0$. Note that $-Z$ is Lipschitz continuous as well and hence, by the union bound,

$$P\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})\right| \geq E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})\right|\right] + z\right] \leq 2\exp\left(\frac{-SBz^{2p}}{8\operatorname{diam}(\mathcal{X})^{2p}}\right).$$

Now, with the reverse triangle inequality, Jensen's inequality and Theorem 1,

$$\begin{aligned}
E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})\right|\right] &\leq E\left[W_p(\hat{\boldsymbol{r}}_S,\boldsymbol{r}) + W_p(\hat{\boldsymbol{s}}_S,\boldsymbol{s})\right] \\
&\leq E\left[W_p^p(\hat{\boldsymbol{r}}_S,\boldsymbol{r})\right]^{1/p} + \left[W_p^p(\hat{\boldsymbol{s}}_S,\boldsymbol{s})\right]^{1/p} \leq 2\mathcal{E}_q^{1/p}/S^{1/(2p)}.
\end{aligned}$$

Together with the last concentration inequality above, this concludes the proof of Theorem 4. □

A.4. Proof of Theorem 5

Denote $V = |\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})|$, $C = 2\mathcal{E}_q^{1/p}/S^{1/(2p)} \geq 0$, and observe that

$$E\left[V^2\right] = \int_0^\infty P(V > \sqrt{t})dt = 2\int_0^\infty P(V > s)sds = 2\int_{-C}^\infty P(V > z+C)(z+C)dz$$

$$\leq 2\int_{-C}^{C}(z+C)dz + 4\int_{C}^{\infty} P(V > z+C)zdz \leq 4C^2 + 8\int_{C}^{\infty} z\exp\left(-\frac{SBz^{2p}}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right)dz,$$

by Theorem 4. Changing variables and using the inequality $y^{2p} \geq y^2$ (valid for $y, p \geq 1$) gives

$$\begin{aligned} 8\int_{C}^{\infty} z\exp\left(-\frac{SBz^{2p}}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right)dz &= 8C^2\int_{1}^{\infty} y\exp\left(-\frac{SB(Cy)^{2p}}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right)dy \\ \leq 8C^2\int_{1}^{\infty} y\exp\left(-\frac{SBC^{2p}y^2}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right)dy &= 8C^2\frac{4(\ \mathrm{diam}(\mathcal{X}))^{2p}}{SBC^{2p}}\exp\left(-\frac{SBC^{2p}}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right) \\ &= 4C^2\frac{(\ \mathrm{diam}(\mathcal{X}))^{2p}}{2^{2p-3}\mathcal{E}_q^2 B}\exp\left(-\frac{4^p\mathcal{E}_q^2 B}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right), \end{aligned}$$

where we have used $C^2 = 4\mathcal{E}_q^{2/p}S^{-1/p}$. Deduce that

$$E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})\right|^2\right] \leq 16\mathcal{E}_q^{2/p}\left\{1 + \frac{(\ \mathrm{diam}(\mathcal{X}))^{2p}}{2^{2p-3}\mathcal{E}_q^2 B}\exp\left(-\frac{4^p\mathcal{E}_q^2 B}{8\ \mathrm{diam}(\mathcal{X})^{2p}}\right)\right\}S^{-1/p}.$$

Now note that (6) implies $\mathcal{E}_q^2 \geq 2^{6p-2}[\mathrm{diam}(\mathcal{X})]^{2p}$ and hence $[\mathrm{diam}(\mathcal{X})]^{2p}/[B2^{2p-3}\mathcal{E}_q^2] \leq 2^{5-8p} \leq 1/8$, so the term in parentheses is smaller than $1+1/8$. Consequently the mean squared error is bounded by $18\mathcal{E}_q^{2/p}S^{-1/p}$.

Similar computations show that $E\left[\left|\hat{W}_p^{(S)}(\boldsymbol{r},\boldsymbol{s}) - W_p(\boldsymbol{r},\boldsymbol{s})\right|^{\alpha}\right] = \mathcal{O}(S^{-\alpha/(2p)})$ for all $0 \leq \alpha \leq 2p$. □

References

Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM J. Math. Anal.*, 43(2):904–924, 2011.

Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, pages 1964–1974, 2017.

Dimitri P. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1(1):7–66, 1992.

Emmanuel Boissard and Thibaut Le Gouic. On the mean speed of convergence of empirical and occupation measures in Wasserstein distance. *Ann. Inst. H. Poincaré Probab. Statist.*, 50(2): 539–563, 2014.

François Bolley and Cédric Villani. Weighted Csiszár-Kullback-Pinsker inequalities and applications to transportation inequalities. *Annales de La Faculté Des Sciences de Toulouse: Mathématiques*, 14(3):331–352, 2005.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. From optimal transport to generative modeling: the VEGAN cookbook. 2017. URL `https://arxiv.org/abs/1705.07642`.

Hector Corrada Bravo and Stefan Theussl. Rcplex: R interface to cplex, 2016. URL `https://CRAN.R-project.org/package=Rcplex`. R package version 0.3-3.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In *International Conference on Machine Learning*, pages 1367–1376. 2018.

Steven N. Evans and Frederick A. Matsen. The phylogenetic Kantorovich–Rubinstein metric for environmental sequence samples. *J. R. Stat. Soc. B*, 74(3):569–592, 2012.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probab. Theory Relat. Fields*, 162(3-4):707–738, 2015.

Carsten Gottschlich and Dominic Schuhmacher. The Shortlist method for fast computation of the earth mover's distance and finding optimal solutions to transportation problems. *PLoS ONE*, 9 (10):e110214, 2014.

Nathael Gozlan and Christian Léonard. A large deviation approach to some transportation cost inequalities. *Probab. Theory Relat. Fields*, 139(1-2):235–283, 2007.

Philippe Heinrich and Jonas Kahn. Strong identifiability and optimal minimax rates for finite mixture estimation. *Ann. Stat.*, 46(6A):2844–2870, 2018.

Leonid Vitaliyevich Kantorovich. On the translocation of masses. *(Dokl.) Acad. Sci. URSS 37*, 3: 199–201, 1942.

Benoît R. Kloeckner. A geometric study of Wasserstein spaces: Ultrametrics. *Mathematika*, 61(1): 162–178, 2015.

Tianyi Lin, Nhat Ho, and Michael I Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. 2019. URL `https://arxiv.org/abs/1901.06482`.

Haibin Ling and Kazunori Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007.

Jialin Liu, Wotao Yin, Wuchen Li, and Yat Tin Chow. Multilevel optimal transport: A fast approximation of Wasserstein-1 distances. 2018. URL `https://arxiv.org/abs/1810.00118`.

David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, New York, 2008.

Axel Munk and Claudia Czado. Nonparametric validation of similar distributions and assessment of goodness of fit. *J. R. Stat. Soc. B*, 60(1):223–241, 1998.

Kangyu Ni, Xavier Bresson, Tony Chan, and Selim Esedoglu. Local histogram based segmentation using the Wasserstein distance. *International Journal of Computer Vision*, 84(1):97–111, 2009.

Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *IEEE 12th International Conference on Computer Vision*, pages 460–467, 2009.

Svetlozar T. Rachev and Ludger Rüschendorf. *Mass Transportation Problems, Volume 1: Theory*. Springer, New York, 1998.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

Brian E. Ruttenberg, Gabriel Luna, Geoffrey P. Lewis, Steven K. Fisher, and Ambuj K. Singh. Quantifying spatial relationships from whole retinal images. *Bioinformatics*, 29(7):940–946, 2013.

Bernhard Schmitzer. A sparse multi-scale algorithm for dense optimal transport. *Journal of Mathematical Imaging and Vision*, 56(2):238–259, 2016.

Jörn Schrieber, Dominic Schuhmacher, and Carsten Gottschlich. DOTmark — a benchmark for discrete optimal transport. *IEEE Access*, 5:271–282, 2016. doi: 10.1109/ACCESS.2016.2639065.

Dominic Schuhmacher, Carsten Gottschlich, and Bjoern Baehre. R-package transport: Optimal transport in various forms, 2014. URL `https://cran.r-project.org/package=transport`. R package version 0.6-3.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11(Oct):2635–2670, 2010.

Sameer Shirdhonkar and David W. Jacobs. Approximate earth mover's distance in linear time. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

Max Sommerfeld and Axel Munk. Inference for empirical Wasserstein distances on finite spaces. *J. R. Stat. Soc. B*, 80(1):219–238, 2018.

Carla Tameling and Axel Munk. Computational strategies for statistical inference based on empirical optimal transport. In *2018 IEEE Data Science Workshop*, pages 175–179. IEEE, 2018.

Jean-Louis Verger-Gaugry. Covering a ball with smaller equal balls in $\mathbb{R}^n$. *Discrete & Computational Geometry*, 33(1):143–155, 2005.

Cédric Villani. *Optimal Transport: Old and New*. Springer, New York, 2008.

Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

Journal of Machine Learning Research 20 (2019) 1-25 Submitted 3/18; Revised 1/19; Published 7/19

Solving the OSCAR and SLOPE Models Using a Semismooth Newton-Based Augmented Lagrangian Method

Ziyan Luo ZYLUO@BJTU.EDU.CN
Department of Mathematics
Beijing Jiaotong University
Beijing, P. R. China

Defeng Sun DEFENG.SUN@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hong Kong

Kim-Chuan Toh MATTOHKC@NUS.EDU.SG
Department of Mathematics, and Institute of Operations Research and Analytics
National University of Singapore
Singapore

Naihua Xiu NHXIU@BJTU.EDU.CN
Department of Mathematics
Beijing Jiaotong University
Beijing, P. R. China

Editor: Benjamin Recht

Abstract

The octagonal shrinkage and clustering algorithm for regression (OSCAR), equipped with the ℓ_1-norm and a pair-wise ℓ_∞-norm regularizer, is a useful tool for feature selection and grouping in high-dimensional data analysis. The computational challenge posed by OSCAR, for high dimensional and/or large sample size data, has not yet been well resolved due to the non-smoothness and non-separability of the regularizer involved. In this paper, we successfully resolve this numerical challenge by proposing a sparse semismooth Newton-based augmented Lagrangian method to solve the more general SLOPE (the sorted L-one penalized estimation) model. By appropriately exploiting the inherent sparse and low-rank property of the generalized Jacobian of the semismooth Newton system in the augmented Lagrangian subproblem, we show how the computational complexity can be substantially reduced. Our algorithm offers a notable computational advantage in the high-dimensional statistical regression settings. Numerical experiments are conducted on real data sets, and the results demonstrate that our algorithm is far superior, in both speed and robustness, to the existing state-of-the-art algorithms based on first-order iterative schemes, including the widely used accelerated proximal gradient (APG) method and the alternating direction method of multipliers (ADMM).

Keywords: Linear Regression, OSCAR, Sparsity, Augmented Lagrangian Method, Semismooth Newton method

©2019 Ziyan Luo, Defeng Sun, Kim-Chuan Toh and Naihua Xiu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-172.html.

1. Introduction

Feature selection and grouping is highly beneficial in learning with high-dimensional data containing spurious features, and thus has found wide applications in statistics (Hocking, 1976; Miller, 2002), computer vision (Mairal et al., 2014), signal processing (Chen et al., 1998; Tropp, 2006; Figueiredo et al., 2007), bioinformatics (Wang et al., 2005; Rapaport et al., 2007). The octagonal shrinkage and clustering algorithm for regression (OSCAR) proposed by (Bondell and Reich, 2008), serves as an efficient sparse modeling tool with automatic feature grouping by employing the ℓ_1-norm regularizer together with a pairwise ℓ_∞ penalty. The OSCAR penalized problem for linear regression with the least squares loss function takes the form of

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|^2 + w_1\|x\|_1 + w_2\sum_{i<j}\max\{|x_i|,|x_j|\}\,, \tag{1}$$

where $b\in\mathbb{R}^m$ is the response vector, $A\in\mathbb{R}^{m\times n}$ is the design matrix, $x\in\mathbb{R}^n$ is the vector of unknown coefficients to be estimated, w_1 and w_2 are two nonnegative tuning parameters for the tradeoff of the sparsity and equality of coefficients for correlated features promoted by the ℓ_1-norm and the pairwise ℓ_∞ term, respectively. Note that in high dimensional statistical regressions, we often have $n \gg m$, that is, the number of features is larger than the sample size.

The OSCAR penalized problem (1) is a convex optimization problem. When the pairwise ℓ_∞ term is removed, the problem (1) is reduced to the well-known LASSO model proposed by Tibshirani (1996) in statistics and a rich variety of algorithms have been proposed, most of them have taken the advantage of the componentwise separability of the ℓ_1-norm in their algorithmic design. With the additional pairwise ℓ_∞ term, the problem (1) becomes understandably more challenging due to the lack of separability of the OSCAR regularization term. In Bondell and Reich (2008), the traditional quadratic programming (QP) and sequential quadratic programming (SQP) based algorithms are employed for solving (1) with numerical implementations limited to small data sets. Efficient numerical algorithms are in dire need especially for large scale problems resulting from the explosion in the size and complexity of modern data sets in practical applications. In Zhong and Kwok (2012), the accelerated proximal gradient (APG) method, proposed by Nesterov (1983) and coined as FISTA for the ℓ_1-norm regularization problem by Beck and Teboulle (2009), is adopted for solving relatively large scale instances by taking advantage of the efficient computation of the proximal mapping of the OSCAR penalty function, which is explained in the next paragraph.

We begin by introducing some notation. For a given vector $x\in\mathbb{R}^n$, denote $|x|$ to be the vector obtained from x by taking the absolute value of its components. Let $|x|_{(i)}$ be the i-th largest component of $|x|$ such that $|x|_{(1)} \geq |x|_{(2)} \cdots \geq |x|_{(n)}$. With the above notation, the OSCAR penalty can be written as

$$w_1\|x\|_1 + w_2\sum_{i=1}^n \max_{i<j}\{|x_i|,|x_j|\} \;=\; \sum_{i=1}^n \lambda_i |x|_{(i)}, \tag{2}$$

where $\lambda_i = w_1 + w_2(n-i)$, $i=1,\ldots,n$, satisfy the property that $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_n\geq 0$. The resulting regularization function $\kappa_\lambda(x) := \sum_{i=1}^n \lambda_i|x|_{(i)}$ for any $x\in\mathbb{R}^n$, termed as the

decreasing weighted sorted ℓ_1-norm (DWSL1) by Zeng and Figueiredo (2014b), is exactly the weighted Ky Fan norm as studied in Wu et al. (2014) as long as $\lambda_1 > 0$. The computation of the proximal mapping of DWSL1 has been studied in the literature (see, e.g., Zeng and Figueiredo, 2014a,b; Bogdan et al., 2015). It is heavily related to the pool adjacent violators algorithm (PAVA) for solving isotonic regression problems (Barlow and Brunk, 1972) in the field of ordered statistics (see, e.g., Robertson et al., 1988; Silvapulle and Sen, 2011).

As a more general framework of the OSCAR problem (1), the least-squares problem with the DWSL1 regularization term is called the sorted L-one penalized estimation (SLOPE), which has been shown to have good performance for controlling the false discovery rate (FDR) in sparse statistical models in Bogdan et al. (2015). The APG method is employed in the latter paper for solving the SLOPE model by relying on the efficient numerical evaluation of the proximal mapping of the sorted ℓ_1-norm. As can be seen, most of the existing methods for solving the OSCAR model and the more general SLOPE model in the large scale settings are based on the first-order information of the underlying nonsmooth optimization model. However, as demonstrated by the works of Li et al. (2018a) for the LASSO and Li et al. (2018b) for the fused LASSO, there are compelling evidences to suggest that one can design a much more efficient algorithm if one can fully exploit the inherent second-order sparsity and low-rank property present in the OSCAR model or the more general SLOPE model. In this paper, we will show how this can be achieved by focusing on the following SLOPE model:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|^2 + \sum_{i=1}^{n} \lambda_i |x|_{(i)} \tag{3}$$

with parameters $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ and $\lambda_1 > 0$. Note that here the parameter vector λ is a general vector satisfying the previous condition. It needs not be restricted to the parameter vector associated with the OSCAR penalty in (2).

The main goal of this paper is to design a semismooth Newton-based augmented Lagrangian method (Newt-ALM for short) for solving the SLOPE model (2) from the dual perspective. As a main contribution of this paper, in Section 2 and Subsection 3.4, we will see that one can extract some special low-rank and sparsity structures in the generalized Jacobian of the proximal mapping associated with the sorted ℓ_1-norm. In turn, the Hessian matrices involved in the ALM subproblems also inherit the special structures which we can wisely exploit to design a very efficient semismooth Newton method to solve the subproblems. The latter fact, combined with the fast linear convergence of the augmented Lagrangian method which we will establish in Section 3, will enable our Newt-ALM algorithm to perform highly efficiently later in the numerical experiments on large scale instances. The comparison of our algorithm with the inexact ADMM (iADMM) proposed in Chen et al. (2017) and the APG method implemeneted in the SLOPE solver in Bogdan et al. (2015) for solving OSCAR problems indicates that our Newt-ALM can outperform these state-of-the-art first-order algorithms substantially.

The remaining parts of the paper are organized as follows. In Section 2, some analytical properties of the proximal mapping of the sorted ℓ_1-norm and their generalized Jacobians are reviewed and developed. These properties are critical for the subsequent analysis on the local convergence rate of the algorithm in the next section. Section 3 is dedicated to the semismooth Newton augmented Lagrangian method and its convergence analysis. In

addition, we also extract the low-rank and sparsity structures present in the generalized Jacobians of the proximal mapping of the sorted ℓ_1-norm. These structures are crucial for the efficient numerical computation in the semismooth Newton method. Numerical results are reported in Section 4 to demonstrate the high efficiency and robustness of our algorithm. We conclude our paper in Section 5. Technical proofs are provided in Appendix A.

2. The generalized Jacobian of the proximal mapping of the DWSL1 norm

As mentioned in the introduction, a key factor contributing to the high computational efficiency of our proposed Newt-ALM is the characterization of the generalized Jacobian matrix of the proximal mapping for the DWSL1 norm (or sorted ℓ_1-norm). In particular, the characterization will enable us to extract the underlying low-rank and sparsity structures present in the generalized Jacobian which we can fully exploit for computational efficiency within the semismooth Newton method for solving the subproblem in each iteration of the augmented Lagrangian method. The purpose of this section is to present the characterization of the generalized Jacobain of the proximal mapping for the DWSL1 norm and its analytical properties.

Let $\mathbf{\Pi^s_n}$ be the set of all signed permutation matrices in $\mathbb{R}^{n\times n}$. Recall that an $n\times n$ signed permutation matrix is a matrix whose rows are the permutation of those of the $n\times n$ identity matrix and the only non-zero element in each row can take the value ± 1. Note that the cardinality of $\mathbf{\Pi^s_n}$ is $2^n n!$. For any given vector $y\in\mathbb{R}^n$, denote

$$\Pi^s(y) := \left\{\pi \in \mathbf{\Pi^s_n} \mid (\pi y)_i = |y|_{(i)},\ i=1,\ldots,n\right\}.$$

Let $\kappa_\lambda(x) := \sum_{i=1}^n \lambda_i |x|_{(i)}$ with $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. The proximal mapping of κ_λ is

$$\mathrm{Prox}_{\kappa_\lambda}(y) = \arg\min_x \left\{\frac{1}{2}\|x-y\|^2 + \kappa_\lambda(x)\right\}, \quad \forall y\in\mathbb{R}^n.$$

Since the involved objective function is strongly convex (see, e.g., Wu et al., 2014; Bogdan et al., 2015) and piecewise quadratic, the proximal mapping $\mathrm{Prox}_{\kappa_\lambda}$ is then piecewise affine, a result known from Sun (1986) or (Rockafellar and Wets, 1998, Proposition 12.30). Define

$$x_\lambda(w) := \arg\min_x \left\{\frac{1}{2}\|x-w\|^2 + \lambda^\top x \mid Bx \geq 0\right\}, \quad w\in\Re^n, \tag{4}$$

where

$$Bx = [x_1-x_2, x_2-x_3, \ldots, x_{n-1}-x_n, x_n]^\top \in \mathbb{R}^n.$$

It is known from (Bogdan et al., 2015, Proposition 2.2) that for any $y\in\mathbb{R}^n$ and $\pi\in\Pi^s(y)$, $\mathrm{Prox}_{\kappa_\lambda}(\pi y) = x_\lambda(\pi y)$. Furthermore, for any $\lambda\in\mathbb{R}^n_+$ satisfying $\lambda_1 \geq \cdots \geq \lambda_n$, and any vector $y\in\mathbb{R}^n$, we have

$$\mathrm{Prox}_{\kappa_\lambda}(y) = \pi^{-1} x_\lambda(\pi y), \quad \forall \pi\in\Pi^s(y) \subseteq \mathbf{\Pi^s_n}. \tag{5}$$

Given the structure of $x_\lambda(\cdot)$, one can see that the Jacobian of $x_\lambda(\cdot)$ at any $w\in\mathbb{R}^n$, as constructed in (Han and Sun, 1997), is given by

$$\mathcal{P}(w) = \left\{P\in\mathbb{R}^{n\times n} \mid P = I - B_\Gamma^\top \left(B_\Gamma B_\Gamma^\top\right)^{-1} B_\Gamma, \Gamma\in\mathcal{K}(w)\right\}. \tag{6}$$

Here

$$\mathcal{K}(w) := \{\Gamma \subseteq \{1,\ldots,n\} \mid \mathrm{Supp}(z_\lambda(w)) \subseteq \Gamma \subseteq \mathbf{I}(x_\lambda(w))\},$$

where $z_\lambda(w) = (BB^\top)^{-1}B(w-\lambda-x_\lambda(w))$ is an optimal dual multiplier vector associated with the inequality constraints in (4), $\mathbf{I}(x_\lambda(w)) = \{i \in \{1,\ldots,n\} \mid (Bx_\lambda(w))_i = 0\}$ is the set of active indices (the indices of the active constraints) in (4), and B_Γ is the submatrix obtained by extracting the rows of B with indices in Γ. In the above, $\mathrm{Supp}(z_\lambda(w))$ is the support of $z_\lambda(w)$, i.e., the index set of nonzero compomonents of $z_\lambda(w)$. Observe that each element of $\mathcal{P}(w)$ is the projection onto the null space of B_Γ for some index set Γ sandwiched between the set of active indices $\mathbf{I}(x_\lambda(w))$ and $\mathrm{Supp}(z_\lambda(w))$.

It is known from Lemma 2.1 in Han and Sun (1997) that for any $w \in \mathbb{R}^n$, there exists a neighborhood W of w such that for all $w' \in W$,

$$\begin{cases} \mathcal{K}(w') \subseteq \mathcal{K}(w), \\ \mathcal{P}(w') \subseteq \mathcal{P}(w), \\ x_\lambda(w') - x_\lambda(w) - P(w'-w) = 0, \quad \forall P \in \mathcal{P}(w'). \end{cases} \tag{7}$$

Define the multifunction $\mathcal{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^{n\times n}$ by

$$\mathcal{M}(y) := \left\{M \in \mathbb{R}^{n\times n} \mid M = \pi^{-1}P\pi, \pi \in \Pi^s(y), P \in \mathcal{P}(\pi y)\right\}. \tag{8}$$

Recall that the set-valued mapping $\mathcal{M} : \Re^n \rightrightarrows \Re^{n\times n}$ is said to be upper semicontinuous (Aubin and Frankowska, 1990, Definition 1.4.1) at a certain point $y \in \Re^n$ if for any neighborhood $\mathcal{N}$ of $\mathcal{M}(y)$, there exists a constant $\rho > 0$ such that

$$\mathcal{M}(y') \subset \mathcal{N}, \quad \forall y' \in \mathbb{B}(y,\rho) := \left\{y' \in \Re^n \mid \|y'-y\| \le \rho\right\}.$$

Then we have the following theorem which is adapted from (Li et al., 2018b, Proposition 7). Its proof is given in Appendix A.

Theorem 1 *Let $\lambda \in \mathbb{R}^n_+$ be such that $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$. Then $\mathcal{M}(\cdot)$ is a nonempty and compact valued, upper semicontinuous multifunction, and for any given $y \in \mathbb{R}^n$, every $M \in \mathcal{M}(y)$ is symmetric and positive semidefinite. Moreover, there exists a neighborhood U of y such that for all $y' \in U$,*

$$\mathrm{Prox}_{\kappa_\lambda}(y') - \mathrm{Prox}_{\kappa_\lambda}(y) - M(y'-y) = 0, \quad \forall M \in \mathcal{M}(y'). \tag{9}$$

Example. Next, we present an example to illustrate the result in equation (9) of Theorem 1 explicitly. Consider the vector $y = [4,3,0]^\top$ and the parameter vector $\lambda = [3,1,1]^\top$. For any $y' = [y'_1, y'_2, y'_3]^\top$ that is sufficiently close to y, say $\|y'-y\| \le 0.1$, we can show by using the pool adjacent violators algorithm that

$$\mathrm{Prox}_{\kappa_\lambda}(y) = \begin{bmatrix} 1.5 \\ 1.5 \\ 0 \end{bmatrix}, \quad \mathrm{Prox}_{\kappa_\lambda}(y') = \begin{bmatrix} \frac{y'_1+y'_2-4}{2} \\ \frac{y'_1+y'_2-4}{2} \\ 0 \end{bmatrix}.$$

Thus

$$\mathrm{Prox}_{\kappa_\lambda}(y') - \mathrm{Prox}_{\kappa_\lambda}(y) = M(y'-y) \quad \text{with} \quad M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In this case $\mathcal{M}(y') = \{M\}$.

Next, we discuss the semismoothness property of the proximal mapping $\mathrm{Prox}_{\kappa_\lambda}$. Recall from Mifflin (1977); Kummer (1988); Qi and Sun (1993); Sun and Sun (2002) or directly from (Li et al., 2018b, Definition 1) that the semismoothness with respect to a given nonempty compact valued, upper semicontinuous multifunction is defined as follows.

Let $\mathcal{O} \subseteq \mathbb{R}^n$ be any given open set, $\mathcal{K} : \mathcal{O} \rightrightarrows \mathbb{R}^{m\times n}$ be a nonempty compact valued, upper semicontinuous multifunction, and $F : \mathcal{O} \to \mathbb{R}^m$ be a locally Lipschitz continuous function, i.e., for any $x \in \mathcal{O}$, there exist positive constants L_x and δ_x such that for all $y, y' \in \mathcal{O}$ satisfying $\|y-x\| \le \delta_x$ and $\|y'-x\| \le \delta_x$, we get $\|F(y)-F(y')\| \le L_x\|y-y'\|$. F is said to be *semismooth* at $x \in \mathcal{O}$ with respect to the multifunction $\mathcal{K}$ if F is directionally differentiable at x and for any $V \in \mathcal{K}(x+d)$ with $d \to 0$,

$$F(x+d) - F(x) - Vd = o(\|d\|).$$

Let γ be a positive scalar. F is said to be γ-order semismooth (stongly semismooth if $\gamma = 1$) at $x \in \mathcal{O}$ with respect to $\mathcal{K}$ if F is directionally differentiable at x and for any $V \in \mathcal{K}(x+d)$ with $d \to 0$,

$$F(x+d) - F(x) - Vd = O(\|d\|^{1+\gamma}).$$

F is said to be a semismooth (γ-order semismooth, stongly) function on $\mathcal{O}$ with respect to $\mathcal{K}$ if F is semismooth (γ-order semismooth, strongly semismooth) everywhere in $\mathcal{O}$ with respect to $\mathcal{K}$. It is known from Theorem 1 that $\mathrm{Prox}_{\kappa_\lambda}$ is γ-order semismooth on $\mathbb{R}^n$ with respect to $\mathcal{M}$ for any given positive γ.

3. A semismooth Newton augmented Lagrangian method

3.1. The algorithmic framework

Given $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $\lambda_1 \ge \cdots \ge \lambda_n \ge 0$ with $\lambda_1 > 0$, the DWSL1 regularized least squares problem can be rewritten as

$$(P) \quad \max_{x\in\mathbb{R}^n} \left\{-f(x) := -\frac{1}{2}\|Ax-b\|^2 - \kappa_\lambda(x)\right\}. \tag{10}$$

Its dual problem takes the form of

$$(D) \quad \min_{y\in\mathbb{R}^m, \xi\in\mathbb{R}^n} \left\{\frac{1}{2}\|y\|^2 + \langle b, y\rangle + \kappa^*_\lambda(\xi) \middle| A^\top y + \xi = 0\right\}, \tag{11}$$

where $\kappa^*_\lambda(v) := \sup_{x\in\mathbb{R}^n}\{\langle x, v\rangle - \kappa_\lambda(x)\}$ is the Fenchel conjugate function of κ_λ. For any given scalar $\sigma > 0$, the corresponding reduced augmented Lagrangian function associated with (D) is defined by

$$\begin{aligned} L_\sigma(y;x) &:= \inf_{\xi\in\mathbb{R}^n} \left\{\frac{1}{2}\|y\|^2 + \langle b, y\rangle + \kappa^*_\lambda(\xi) - \langle A^\top y + \xi,\, x\rangle + \frac{\sigma}{2}\|A^\top y + \xi\|^2\right\} \\ &= \frac{1}{2}\|y\|^2 + \langle b, y\rangle + \inf_{\xi\in\mathbb{R}^n} \left\{\kappa^*_\lambda(\xi) + \frac{\sigma}{2}\|A^\top y + \xi - \sigma^{-1}x\|^2 - \frac{1}{2\sigma}\|x\|^2\right\} \\ &= \frac{1}{2}\|y\|^2 + \langle b, y\rangle - \frac{1}{2\sigma}\|x\|^2 + \sigma\phi_{\kappa^*_\lambda/\sigma}\left(\sigma^{-1}x - A^\top y\right), \end{aligned}$$

where $\phi_{\kappa_\lambda^*/\sigma}$ is the Moreau-Yosida regularization of κ_λ^*/σ defined as

$$\phi_{\kappa_\lambda^*/\sigma}(x) := \min_{u\in\mathbb{R}^n}\left\{\frac{1}{\sigma}\kappa_\lambda^*(u)+\frac{1}{2}\|u-x\|^2\right\}, \quad \forall x\in\mathbb{R}^n.$$

The inexact augmented Lagrangian method (Rockafellar, 1976b) together with the semismooth Newton method will be employed to solve (D) with the algorithmic framework as described in Algorithm 1. Note that the most expensive part in each iteration of the ALM is in solving the subproblem in Step 1.

Algorithm 1: An inexact augmented Lagrangian method for (D) (Newt-ALM)

Choose $\sigma_0>0$ and $\left(y^0,x^0\right)\in\mathbb{R}^m\times\mathbb{R}^n$. For $k=0,1,\ldots$, perform the following steps in each iteration:

Step 1. Compute $y^{k+1}\approx\arg\min_{y\in\mathbb{R}^m}\left\{\Psi_k(y):=L_{\sigma_k}(y;x^k)\right\}$;

Step 2. Compute $x^{k+1}=\mathrm{Prox}_{\sigma_k\kappa_\lambda}\left(x^k-\sigma_kA^\top y^{k+1}\right)$;

Step 3. Update $\sigma_{k+1}\uparrow\sigma_\infty\le\infty$.

The stopping criteria for the approximation in Step 1 of the inexact augmented Lagrangian method have been well discussed in Rockafellar (1976b,a). Given two summable sequences of nonnegative numbers, $\{\epsilon_k\}_{k\ge0}$ and $\{\delta_k\}_{k\ge0}$, and a nonnegative convergent sequence $\{\delta_k'\}_{k\ge0}$ with limit 0, the stopping criteria can be simplified as follows in our case:

(A) $\|\nabla\Psi_k(y^{k+1})\|\le\epsilon_k/\sqrt{\sigma_k}$;

(B1) $\|\nabla\Psi_k(y^{k+1})\|\le(\delta_k/\sqrt{\sigma_k})\,\|x^{k+1}-x^k\|$;

(B2) $\|\nabla\Psi_k(y^{k+1})\|\le(\delta_k'/\sigma_k)\,\|x^{k+1}-x^k\|$.

3.2. Convergence theory

The piecewise linear-quadratic property of f as defined in (10) leads to the polyhedral multifunction ∂f (the sub-differential of f). By a fundamental result in Robinson (1981), this further implies that ∂f satisfies the error bound condition with a common modulus, say a_f. Specifically, the error bound condition states that for the optimal solution set $(\partial f)^{-1}(0)$ of (P), which we denote by S^*, there exists some $\varepsilon>0$ such that for any $x\in\mathbb{R}^n$ satisfying $\mathrm{dist}(0,\partial f(x))\le\varepsilon$, it holds that

$$\mathrm{dist}(x,S^*)\le a_f\ \mathrm{dist}(0,\partial f(x)). \tag{12}$$

Similarly, consider the Lagrangian function associated with (D) which is defined by

$$l(y;x):=\frac{1}{2}\|y\|^2+\langle b,y\rangle+\kappa_\lambda^*(\xi)-\langle A^\top y+\xi,\,x\rangle.$$

For the polyhedral multifunction T_l defined as $T_l(y,x)=\{(y',x')\mid(y',-x')\in\partial l(y;x)\}$, there exist some a_l and $\varepsilon'>0$ such that for any $(y,x)\in\mathbb{R}^m\times\mathbb{R}^n$ satisfying $\mathrm{dist}(0,T_l(y,x))\le\varepsilon'$, it has

$$\mathrm{dist}\left((y,x),\{y^*\}\times S^*\right)\le a_l\ \mathrm{dist}(0,T_l(y,x)), \tag{13}$$

where y^* is the unique optimal solution of (D). Following the results on global and local convergence of the ALM as stated in (Rockafellar, 1976b,a; Li et al., 2018a,b), we can readily obtain the following convergence results on Algorithm 1 with the above stopping criteria. As the proofs of the theorems are almost the same as those appeared in (Li et al., 2018a, Theorems 3.2 and 3.3), we will omit them here.

Theorem 2 (Global convergence) *Let $\left\{(y^k,x^k)\right\}$ be the infinite sequence generated by Algorithm 1 with stopping criterion (A) applied to Ψ_k in Step 1. Then $\left\{x^k\right\}$ converges to an optimal solution to (P), and $\left\{y^k\right\}$ converges to the unique optimal solution of (D).*

Theorem 3 (Local linear-rate convergence) *Let $\left\{(y^k,x^k)\right\}$ be the infinite sequence generated by Algorithm 1 with stopping criteria (A) and (B1) applied to Ψ_k in Step 1. Then for all k sufficiently large,*

$$\mathrm{dist}\left(x^{k+1},S^*\right)\le\theta_k\mathrm{dist}\left(x^k,S^*\right),$$

where

$$\theta_k:=\left(\frac{a_f}{\sqrt{a_f^2+\sigma_k^2}}+2\delta_k\right)\frac{1}{1-\delta_k}\ \to\ \theta_\infty:=\frac{a_f}{\sqrt{a_f^2+\sigma_\infty^2}}<1$$

as $k\to+\infty$, and a_f is from (12). Additionally, if the criterion (B2) is also adopted, then for all k sufficiently large,

$$\|y^{k+1}-y^*\|\le\theta_k'\|x^{k+1}-x^k\|,$$

where

$$\theta_k':=\frac{a_l(1+\delta_k')}{\sigma_k}\to\frac{a_l}{\sigma_\infty}$$

as $k\to+\infty$, and a_l is from (13).

Remark 4 (Global linear-rate convergence) *Besides the local linear-rate convergence as stated in Theorem 3, one can also obtain the global Q-linear convergence of the primal sequence $\{x^k\}$ and the global R-linear convergence of the dual infeasibility and the duality gaps for the sequence generated by Algorithm 1 based on (Cui et al., 2018, Proposition 2 and Lemma 3) or by mimicking the proofs of (Zhang et al., 2018, Theorem 4.1 and Remark 4.1) since problem (P) possesses the following property: For any positive scalar r, there exists $t>0$ such that*

$$\mathrm{dist}(x,S^*)\le t\,\mathrm{dist}(0,\partial f(x)),\ \forall x\in\mathbb{R}^n\ \textit{satisfying}\ \mathrm{dist}(x,S^*)\le r,$$

(see, Zhang et al., 2018, Proposition 2.2). We omit the details here.

3.3. The semismooth Newton method for solving the subproblem in Step 1

It is known from Moreau (1965) or (Rockafellar, 1970, Theorem 31.5) that Ψ_k is continuously differentiable and

$$\nabla\Psi_k(y)=y+b-A\mathrm{Prox}_{\sigma_k\kappa_\lambda}(x^k-\sigma_kA^\top y),\quad\forall y\in\mathbb{R}^m.$$

Since Ψ_k is strongly convex with bounded level sets, the unique solution of the minimization subproblem, $\min_{y\in\mathbb{R}^m}\Psi_k(y)$, in Step 1 of the ALM can be computed by solving the following first-order optimality condition

$$\nabla\Psi_k(y)=0. \tag{14}$$

For any $y\in\mathbb{R}^m$, define

$$\mathcal{G}_k(y):=\left\{V\in\mathbb{R}^{m\times m}\mid V=I_m+\sigma_k AMA^\top,\ M\in\mathcal{M}\left((\sigma_k)^{-1}x^k-A^\top y\right)\right\},$$

where $\mathcal{M}$ is defined in (8). The following semismooth Newton (SSN) method is then applied to solve the semismooth equation (14), as presented in Algorithm 2.

Algorithm 2: A semismooth Newton method for solving (14)

Choose $\mu\in(0,1/2)$, $\bar{\eta}\in(0,1)$, $\delta\in(0,1)$, $\tau\in(0,1]$, $y^0\in\mathbb{R}^m$. For $j=0,1,\ldots$, perform the following steps in each iteration:

Step 1. (Computing the Newton direction) Choose an element $M_j\in\mathcal{M}((\sigma_k)^{-1}x^k-A^\top y^j)$ and set $V_j:=I_m+\sigma_k AM_jA^\top$. Solve the Newton equation

$$V_jd=-\nabla\Psi_k(y^j) \tag{15}$$

exactly or by the conjugate gradient (CG) algorithm to get d^j such that $\|V_jd^j+\nabla\Psi_k(y^j)\|\le\min\{\bar{\eta},\|\nabla\Psi_k(y^j)\|^{1+\tau}\}$.

Step 2. (Line search) Set $\alpha_j=\delta^{m_j}$, where m_j is the least nonnegative integer m satisfying

$$\Psi_k(y^j+\delta^m d^j)\le\Psi_k(y^j)+\mu\delta^m\langle\nabla\Psi_k(y^j),d^j\rangle.$$

Step 3. Set $y^{j+1}=y^j+\alpha_jd^j$.

3.4. Efficient implementations of the semismooth Newton method

In this subsection, the sparsity and low-rank structure of the coefficient matrix in the linear system (15) will first be uncovered. Then the structures will be exploited through designing novel numerical techniques for solving the large scale system (15) to achieve efficient implementations of the semismooth Newton method in Algorithm 2. For any given index set $\Gamma\subseteq\{1,\ldots,n\}$, define the diagonal matrix $\Sigma_\Gamma\in\mathbb{R}^{n\times n}$ by

$$(\Sigma_\Gamma)_{ii}=\begin{cases}1, & \text{if } i\in\Gamma;\\ 0, & \text{otherwise.}\end{cases}$$

Similar to the case in (Li et al., 2018b, Proposition 6), there exists some positive integer N such that Σ_Γ can be rewritten as a block diagonal matrix

$$\Sigma_\Gamma=\mathrm{Diag}(\Lambda_1,\ldots,\Lambda_N)$$

with $\Lambda_i\in\{O_{n_i},I_{n_i}\}$ for each $i\in\{1,\ldots,N\}$ where any two consecutive blocks Λ_i and Λ_{i+1} are not of the same type. Note that O_{n_i} denotes the n_i by n_i zero matrix. Denote

$$J=\{j\in\{1,\ldots,N\}\mid\Lambda_j=I_{n_j}\}.$$

Then we have

$$P=I_n-B_\Gamma^\top(B_\Gamma B_\Gamma^\top)^{-1}B_\Gamma=\mathrm{Diag}(P_1,\ldots,P_N),$$

where

$$P_i=\begin{cases}\frac{1}{n_i+1}e_{n_i+1}e_{n_i+1}^\top, & \text{if } i\in J \text{ and } i\ne N;\\ O_{n_i}, & \text{if } i\in J \text{ and } i=N;\\ I_{n_i-1}, & \text{if } i\notin J \text{ and } i\ne 1;\\ I_{n_i}, & \text{if } i\notin J \text{ and } i=1\end{cases}$$

with the convention $I_0=\emptyset$. This block diagonal matrix P can be further decomposed into the sum of a sparse diagonal term and a low-rank term as $P=H+UU^T$, where $H=\mathrm{Diag}(H_1,\ldots,H_N)\in\mathbb{R}^{n\times n}$ with

$$H_i=\begin{cases}O_{n_i+1}, & \text{if } i\in J \text{ and } i\ne N;\\ O_{n_i}, & \text{if } i\in J \text{ and } i=N;\\ I_{n_i-1}, & \text{if } i\notin J \text{ and } i\ne 1;\\ I_{n_i}, & \text{if } i\notin J \text{ and } i=1\end{cases}$$

and $U\in\mathbb{R}^{n\times N}$ with its (k,j)-th entry given by

$$U_{kj}=\begin{cases}1/\sqrt{n_j+1}, & \text{if } \sum_{t=1}^{j-1}n_t+1\le k\le\sum_{t=1}^{j}n_t+1 \text{ and } j\in J\backslash\{N\};\\ 0, & \text{otherwise.}\end{cases}$$

Define $\alpha:=\{j\in\{1,\ldots,n\}\mid H_{ii}=1\}=\{1,\ldots,n\}\backslash\Gamma$, and let U_{J_N} be the submatrix of U generated by extracting its columns indexed by $J\backslash\{N\}$. Then for any given $A\in\mathbb{R}^{m\times n}$, any $\Gamma\in\{1,\ldots,n\}$ with its corresponding matrix P defined as above, and any signed permutation matrix π, we have

$$\begin{aligned}A\pi^\top P\pi A^\top &= A\pi^\top H\pi A^\top+A\pi^\top UU^\top\pi A^\top\\ &= A\pi(\alpha,:)^\top\pi(\alpha,:)A^\top+\widetilde{A}\widetilde{U}_{J_N}\widetilde{U}_{J_N}^\top\widetilde{A}^\top\\ &=: V_1V_1^\top+V_2V_2^\top,\end{aligned} \tag{16}$$

where $V_1=A\pi(\alpha,:)^\top$, $V_2=\widetilde{A}\widetilde{U}_{J_N}$ with $\widetilde{U}_{J_N}$ being the submatrix of U_{J_N} obtained by dropping all its zero rows and $\widetilde{A}$ is the submatrix obtained from the permuted matrix $A\pi^\top$ by dropping the columns corresponding to those zero rows in U_{J_N}. We call the structure uncovered in (16) that is inherited from the sparse plus low-rank structure of the generalized Jacobian P as the *second-order sparsity*.

Based on the structure in (16), the cost of computing $A\pi^\top P\pi A^\top$ is dramatically reduced from $\mathcal{O}(mn(n+m))$ by naive computation to $\mathcal{O}(m^2(r_1+r_2))$, where r_1 is number of columns in V_1 and r_2 is the number of columns in V_2. Here r_1 refers to the number of inactive constraints in $Bx\ge 0$, and r_2 refers to the number of distinct nonzero identical components in Bx, both of which are generally no larger than the number of nonzero components of x. In the setting of high-dimensional sparse grouping linear regression models, m, r_1, r_2 and N are generally much smaller than n, therefore the aforementioned reduction of the computational cost can be highly significant.

If m is not too large, we can use the (sparse) Cholesky factorization to directly solve the linear system (15). In the case where $r_1 + r_2 \ll m$, the cost of solving (15) can be further reduced by using the Sherman-Morrison-Woodbury (SMW) formula as follows:

$$\left(I_m + \sigma A\pi^\top P\pi A^\top\right)^{-1} = (I_m + WW^\top)^{-1} = I_m - W(I_{r_1+r_2} + W^\top W)^{-1}W^\top,$$

where $W = \sqrt{\sigma}[V_1 \;\; V_2] \in \mathbb{R}^{m\times(r_1+r_2)}$. In the event when m is extremely large and $r_1 + r_2$ is not small so that using the SMW formula is also expensive, we can use the preconditioned conjugate gradient (PCG) method to solve the linear system (15).

4. Numerical experiments

The performance of our proposed sparse semismooth Newton-based augmented Lagrangian method (Newt-ALM) for solving SLOPE (3) and the special case of the OSCAR model in (1) will be evaluated by comparing it with the following first-order methods:

- the accelerated proximal gradient (APG) algorithm implemented in Bogdan et al. (2015) with its Matlab code available at `http://statweb.stanford.edu/~candes/SortedL1`;
- the semi-proximal alternating direction method of multipliers (sPADMM) (see, e.g., Fazel et al. (2013)) applied to the dual problem (with implementation details presented in Subsection 4.2).

All the computational results are obtained from a desktop computer running on 64-bit Windows Operating System having 4 cores with Intel(R) Core(TM) i5-5257U CPU at 2.70GHz and 8 GB memory.

4.1. Stopping criteria

To measure the accuracy of an approximate optimal solution (y, x) for the dual problem (11) and the primal problem (10), the relative duality gap and the dual infeasibility will be adopted. Specifically, denote

$$\mathrm{Obj}_P := \frac{1}{2}\|Ax - b\|^2 + \kappa_\lambda(x), \text{ and } \mathrm{Obj}_D := -b^\top y - \frac{1}{2}\|y\|^2.$$

Then the relative duality gap can be defined by

$$\eta_G := \frac{|\mathrm{Obj_P} - \mathrm{Obj_D}|}{\max\{1, |\mathrm{Obj}_P|\}}.$$

Note that $\kappa_\lambda^*(\cdot)$ is actually the indicator function induced by the closed convex set

$$\mathcal{C}_\lambda := \left\{ z \;\middle|\; \sum_{j\le i} |z|_{(j)} \le \sum_{j\le i} \lambda_j,\; i = 1, \ldots, n \right\},$$

which is exactly the unit ball of the dual norm to κ_λ (see, e.g., Bogdan et al., 2015; Wu et al., 2014). To characterize the dual infeasibility of y, or equivalently $-A^\top y \in \mathcal{C}_\lambda$, we adopt the measure proposed in Bogdan et al. (2015) which is

$$\eta_D := \max\left\{0, \max_{1\le i\le n} \sum_{j\le i} \left(|A^\top(Ax - b)|_{(j)} - \lambda_j\right)\right\}.$$

For given accuracy parameters ε_G and ε_D, our algorithm Newt-ALM will be terminated once

$$\eta_G \le \varepsilon_G \text{ and } \eta_D \le \varepsilon_D, \tag{17}$$

while both the sPADMM and the APG method will be terminated if (17) holds or if the number of iterations reaches the maximum of $50,000$. In our numerical experiments, we choose $\varepsilon_G = \varepsilon_D =$ 1e-6.

The relative KKT residual

$$\eta = \frac{\|x - \mathrm{Prox}_{\kappa_\lambda}(x - A^\top(Ax - b))\|}{1 + \|x\| + \|A^\top(Ax - b)\|}$$

is adopted to measure the accuracy of an approximate optimal solution x generated from any of the algorithms tested in the numerical experiments.

4.2. ADMM for the dual problem (11)

The implementation details of the (semi-proximal) ADMM for solving problem (11) are elaborated in this subsection. Recall the dual problem (11)

$$(D) \qquad \min_{y\in\mathbb{R}^m, \xi\in\mathbb{R}^n} \left\{ \frac{1}{2}\|y\|^2 + \langle b, y\rangle + \kappa_\lambda^*(\xi) \;\middle|\; A^\top y + \xi = 0 \right\}.$$

We can apply the ADMM framework to solve (D) as follows:

$$\begin{cases} y^{k+1} \approx \arg\min_y \mathcal{L}_\sigma(y, \xi^k; x^k) + \frac{1}{2}\|y - y^k\|_S^2, \\ \xi^{k+1} \approx \arg\min_\xi \mathcal{L}_\sigma(y^{k+1}, \xi; x^k) + \frac{1}{2}\|\xi - \xi^k\|_T^2, \\ x^{k+1} = x^k - \tau\sigma\left(A^\top y^{k+1} + \xi^{k+1}\right), \end{cases} \tag{18}$$

where $\sigma > 0$ is a given penalty parameter, $\tau \in (0, \frac{1+\sqrt{5}}{2})$ is the dual steplength, which is typically chosen to be 1.618,

$$\mathcal{L}_\sigma(y, \xi; x) := \frac{1}{2}\|y\|^2 + \langle b, y\rangle + \kappa_\lambda^*(\xi) - \langle x, A^\top y + \xi\rangle + \frac{\sigma}{2}\|A^\top y + \xi\|^2$$

is the augmented Lagrangian function associated with (D) and S and T are two symmetric positive semidefinite matrices. The convergence results of such a general ADMM including the classical ones with the subproblems solved exactly have been discussed in Fazel et al. (2013) under some mild conditions. An inexact version for the general semi-proximal ADMM scheme and its convergence proof can be found in the recent paper by Chen et al. (2017).

In (18), the subproblem for updating y can be handled by solving the following linear system corresponding to its optimality condition:

$$\Big(I_m + \sigma AA^\top + S\Big)\, y = Ax^k - b - \sigma A\xi^k + Sy^k.$$

The weight matrix S in the proximal term can be simply chosen to be the zero matrix when $I_m + \sigma AA^\top$ admits a relatively cheap Cholesky factorization. Otherwise, we can adopt the rule elaborated in Subsection 7.1 in Chen et al. (2017) for choosing S appropriately.

The subproblem for updating ξ can be reformulated as:

$$\xi^{k+1} \approx \arg\min_{\xi} \kappa^*_\lambda(\xi)/\sigma + \frac{1}{2}\|\xi - (x^k/\sigma - A^\top y^{k+1})\|^2 + \frac{1}{2\sigma}\|\xi - \xi^k\|^2_T.$$

By utilizing the efficient algorithm for computing the proximal mapping $\mathrm{Prox}_{\sigma\kappa_\lambda}(w^k)$ (Bogdan et al., 2015), together with the Moreau identity, we can simply choose $T = 0$ and update ξ as follows:

$$\xi^{k+1} = \mathrm{Prox}_{\kappa^*_\lambda/\sigma}(w^k/\sigma) = \frac{1}{\sigma}\Big(w^k - \mathrm{Prox}_{\sigma\kappa_\lambda}(w^k)\Big),$$

where $w^k := x^k - \sigma A^\top y^{k+1}$.

4.3. Results on solving the OSCAR model

In this subsection, we will test our proposed algorithm Newt-ALM for solving the OSCAR model and benchmark it against the APG algorithm implemented in the SLOPE package from Bogdan et al. (2015) and the ADMM scheme (Gabay and Mercier, 1976; Glowinski and Marrocco, 1975) presented in Subsection 4.2. The comparison among these three algorithms will be in terms of the computation time, the iteration number, and the accuracy measured via the relative KKT residual on several selected data from the UCI data repository Lichman (2013) and the BioNUS data set considered in Li et al. (2018b). To demonstrate the performance of these three methods for solving the OSCAR model, with less consideration on the tuning parameter adjustment for pursuing a nice statistical behavior of the regularization model, here we manually choose the tuning parameters w_1 and w_2 as follows:

$$w_1 = a\|A^\top b\|_\infty \text{ and } w_2 = w_1/\sqrt{n} \tag{19}$$

with several testing values for the factor a. The sparsity is recorded in terms of the minimum number k such that the first k largest components in magnitude contribute a percentage of no less than 99.9% for the ℓ_1-norm. Results are shown in Table 1 and the data description is listed in Table 2. The "nnz" column in Table 1 counts the number of nonzeros in the solution x obtained by Newt-ALM such that $\mathrm{nnz} = \min\{t : \sum_{i=1}^t |x|_{(i)} \geq 0.999\|x\|_1\}$.

Table 1: The comparison results obtained by testing real data from the UCI and the BioNUS data sets with (w_1, w_2) set as in (19). A_1: the ADMM with $\tau = 1.618$; A_2: the APG method implemented by the SLOPE package; A_3: our Newt-ALM.

No.	a	nnz	η $(A_1\|A_2\|A_3)$	Time(s) $(A_1\|A_2\|A_3)$	Iter No. $(A_1\|A_2\|A_3)$
p1	1e-7	1	4.37e-07|8.47e-05|5.26e-10	6.86|3.01|1.97	51|25|7
	1e-8	2	2.53e-07|1.56e-05|3.37e-09	328.20|3071.39|7.11	2251|32704|37
	1e-9	158	9.72e-08|7.37e-04|1.83e-10	779.00|4678.39|25.33	4054|50000|52
p2	1e-7	1	4.00e-07|4.53e-05|2.60e-09	2.48|1.78|1.14	51|25|7
	1e-8	3	5.88e-07|3.14e-04|3.61e-08	99.28|961.80|3.86	2101|16525|35
	1e-9	256	1.89e-07|3.77e-04|1.16e-11	277.84|2605.04|48.97	4627|50000|64
p3	1e-3	6	9.07e-09|1.61e-07|6.48e-09	549.90|396.43|4.07	7827|5522|23
	1e-4	81	3.29e-08|1.18e-06|1.78e-09	4521.18|3652.90|8.77	50000|50000|39
	1e-5	100	3.87e-06|3.26e-05|2.62e-10	5781.13|3776.61|38.20	50000|50000|48
p4	1e-3	245	5.97e-10|1.62e-07|1.51e-11	351.95|203.92|9.22	802|607|7
	1e-4	343	1.04e-07|3.52e-07|1.02e-09	2498.72|7069.69|21.94	4139|21205|23
	1e-5	419	1.84e-04|3.69e-05|3.63e-09	10800.80|17138.70|100.32	9893|50000|41
p5	1e-4	11	1.22e-07|6.48e-07|1.90e-08	144.99|629.91|2.69	1292|17531|21
	1e-5	26	2.07e-07|1.31e-05|3.66e-09	815.35|1618.06|5.32	4896|50000|37
	1e-6	70	2.44e-07|3.72e-04|7.74e-09	859.92|1613.81|24.21	4320|50000|44
p6	1e-6	2	2.30e-07|1.95e-08|2.04e-10	215.07|712.95|4.31	1761|10696|23
	1e-7	10	2.14e-07|1.52e-05|1.42e-09	583.87|2899.70|8.19	3509|50000|33
	1e-8	51	9.61e-09|1.11e-04|5.70e-10	1662.10|2886.64|19.88	6153|50000|46
p7	1e-3	8	2.41e-08|1.70e-07|2.04e-08	286.68|89.86|3.03	2713|1401|16
	1e-4	39	9.98e-07|5.40e-07|4.96e-09	373.67|1745.63|4.64	1782|29896|24
	1e-5	120	9.75e-07|6.81e-06|1.18e-08	1303.22|3032.56|11.39	4266|50000|32
p8	1e-3	3	7.75e-08|2.54e-07|1.70e-07	2.20|5.73|0.73	302|881|13
	1e-4	14	1.06e-07|2.64e-06|1.57e-07	4.95|35.55|0.95	513|8013|21
	1e-5	60	1.20e-07|3.90e-06|2.05e-08	16.28|106.13|1.61	1302|31678|32
p9	1e-3	1	1.75e-09|1.42e-08|2.56e-07	2.50|1.01|0.55	51|28|4
	1e-4	6	6.64e-07|3.99e-07|5.24e-07	8.20|22.47|0.97	151|715|10
	1e-5	15	4.66e-07|1.20e-06|2.50e-08	46.97|220.01|2.02	701|10905|15
p10	1e-2	3	4.05e-08|7.39e-07|3.33e-08	4.20|0.84|0.72	623|82|20
	1e-3	130	2.76e-08|1.24e-06|9.58e-09	13.94|194.86|3.13	1910|25299|33
	1e-4	160	9.66e-07|2.75e-06|1.53e-09	29.99|220.75|5.77	2963|50000|45
p11	1e-3	32	2.93e-08|1.85e-06|7.21e-09	42.61|586.28|2.41	3755|50000|34
	1e-4	155	3.97e-07|1.83e-05|1.83e-08	32.15|713.13|3.25	2216|50000|40
	1e-5	193	2.71e-07|4.08e-05|8.81e-10	93.55|352.37|11.47	4608|50000|54
p12	1e-3	34	3.17e-08|7.60e-07|6.19e-09	9.84|218.69|1.06	3024|42154|34
	1e-4	54	5.97e-07|3.78e-06|2.31e-09	14.02|267.84|2.11	3766|50000|39
	1e-5	59	7.55e-07|7.58e-06|8.24e-10	46.34|267.88|4.22	11501|50000|52
p13	1e-3	7	2.71e-08|8.99e-06|1.93e-08	524.93|842.24|2.17	38456|50000|44
	1e-4	38	2.86e-08|4.68e-05|7.02e-09	761.06|840.22|5.83	50000|50000|45
	1e-5	151	2.28e-07|1.33e-04|3.05e-10	246.02|670.25|10.42	8130|50000|53
p14	1e-2	5	4.93e-08|6.27e-07|4.16e-08	2.20|4.07|0.62	997|1187|22
	1e-3	33	4.65e-07|6.93e-07|5.91e-09	2.89|68.43|1.11	1302|22498|34
	1e-4	51	5.87e-07|3.37e-06|1.61e-08	7.89|154.98|2.08	4063|50000|39
p15	1e-2	1	5.46e-09|4.47e-09|5.08e-09	3.71|1.72|0.72	1002|234|23
	1e-3	11	6.78e-08|2.50e-06|9.02e-09	9.89|300.71|1.17	2531|50000|32
	1e-4	76	1.40e-08|1.79e-05|7.98e-09	13.93|304.82|4.67	3144|50000|41

Table 2: The problem names and sizes

No.	Problem name	$[m, n]$
p1	E2006.train	[16087,150360]
p2	E2006.test	[3308,150358]
p3	pyrim_scaled-expanded5	[74, 201376]
p4	triazines-scaled-expanded4	[186,635376]
p5	abalone_scale_expanded7	[4177,6435]
p6	bodyfat_scale_expanded7	[252,116280]
p7	housing_scale_expanded7	[506,77520]
p8	mpg_scale_expanded7	[392, 3432]
p9	space_ga_scale_expanded9	[3107,5005]
p10	DLBCL_H	[160, 7399]

p11	lung_H1	[203, 12600]
p12	NervousSystem	[60, 7129]
p13	ovarian_P	[253, 15153]
p14	DLBCL_S	[47, 4026]
p15	lung_M	[96, 7129]

From Table 1, we observe that all the 45 tested instances are successfully solved by Newt-ALM within 2 minutes (for most of the cases within less than half a minute), while 3 and 22 cases have failed (i.e., not achieving our stopping criteria) to be solved by ADMM and SLOPE, respectively. This shows that our Hessian based Newt-ALM algorithm is more robust compared to the first-order methods (ADMM and the accelerated proximal gradient method implemented in SLOPE) in its ability to successfully solve difficult problems. Both the solution accuracy (as shown in the column under "η") and the computation time (as shown in the column under "Time(s)") also show a tremendous computational advantage of Newt-ALM comparing to ADMM and SLOPE. In particular, for many of the instances corresponding to `p3, p4, p5, p6, p7, p13`, our algorithm can be more than 100 times faster than ADMM and SLOPE.

It is noteworthy that the dual-based ADMM also works better than SLOPE for majority of the tested instances. The performance profiles of these three algorithms for all 45 tested problems are presented in Figure 1. Recall that a point (x, y) on a particular profile curve implies that the algorithm can solve $(100y)\%$ of all the tested instances up to the desired accuracy within at most x times of the fastest algorithm for each instance. More specifically, for $x = 150$, we can see from Figure 1 that even by consuming more than 150 times of the computation time taken by Newt-ALM, there are still around 40% and 10% of tested instances which are not successfully solved by SLOPE and ADMM, respectively.

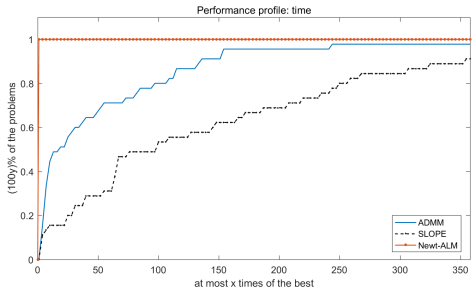


Figure 1: Time comparison for ADMM, SLOPE and Newt-ALM.

4.4. Results on real data sets with group structures

Two real-world data sets with group structures are used to test our proposed algorithm Newt-ALM against the other two first-order algorithms discussed in the previous subsection.

The first one is the breast cancer data set compiled by Van de Vijver et al. (2002), which consists of gene expression data for 8,141 genes in 295 breast cancer tumors (78 metastatic and 217 non-metastatic). We restrict the analysis to 3510 genes which are in at least one pathway. Since the data set is very unbalanced, we adopt the balancing scheme in Jacob et al. (2009) by using 3 replicates of each metastasis tumor and yield a total number of 451 samples. The comparison results are listed in Table 3.

Table 3: The comparison results obtained by testing the breast cancer data set with (w_1, w_2) set as in (19) with different values for a. A_1: ADMM with $\tau = 1.618$; A_2: the APG method in the SLOPE package; A_3: our Newt-ALM.

a	nnz	η $(A_1\|A_2\|A_3)$	Time(s) $(A_1\|A_2\|A_3)$	Iter No. $(A_1\|A_2\|A_3)$
1e-3	145	1.43e-08|7.09e-07|2.48e-09	2.42|26.02|0.86	402|6220|20
1e-4	306	3.66e-08|8.74e-07|9.35e-10	10.21|70.51|2.80	1310|18016|33
1e-5	335	1.86e-09|7.12e-06|8.31e-10	34.63|205.39|4.68	3051|50000|43

The second data set is the NCEP/NCAR reanalysis 1 data set from Kalnay et al. (1996) which contains the monthly means of climate data measurements spread across the globe in a grid of $2.5^o \times 2.5^o$ resolutions (longitude and latitude 144×73) from 1948/01/01 to 2018/05/31. Each grid point has 7 predictive variables including the air temperature, precipitable water, relative humidity, pressure, sea level pressure, horizontal wind speed and vertical wind speed, which leads to a natural group structure in the data set (each group of length 7). The resulting measurement matrix A is of size $[m, n] = 845 \times 73584$. Similar to the parameter scheme chosen in Ndiaye et al. (2016) and Zhang et al. (2018), we manually choose the tuning parameters w_1 and w_2 as follows:

$$w_1 = 0.4w(t) \text{ and } w_2 = 0.6w(t) \tag{20}$$

with $w(t) = 10^{-5+[3(t-1)/99]} \times \|A^\top b\|_\infty$. The numerical results with different t's are listed in Table 4.

Table 4: The comparison results obtained by testing the NCEP/NCAR reanalysis 1 data set with (w_1, w_2) set as in (20). A_1: ADMM with $\tau = 1.618$; A_2: the APG method in the SLOPE package; A_3: our Newt-ALM.

t	η $(A_1\|A_2\|A_3)$	Time(s) $(A_1\|A_2\|A_3)$	Iter No. $(A_1\|A_2\|A_3)$
10	6.85e-07|9.76e-07|6.07e-11	12.75|6.19|1.50	136|104|4
20	1.65e-10|5.29e-07|7.74e-10	18.73|11.68|0.94	201|203|2
30	1.09e-07|8.68e-07|1.33e-11	32.22|12.70|1.20	330|221|3
40	3.02e-07|7.55e-07|3.57e-12	47.10|15.14|0.98	506|265|2
50	3.38e-08|5.16e-07|1.73e-12	55.77|26.52|1.36	608|469|4
60	2.06e-07|8.96e-07|1.16e-10	71.26|28.92|1.12	761|504|2
70	1.35e-08|8.37e-07|1.51e-10	63.14|37.32|1.11	685|653|2
80	5.41e-07|1.24e-03|4.89e-08	574.57|2841.53|4.06	5451|50000|13
90	1.00e-03|2.06e-02|3.17e-08	4911.28|2969.07|3.72	50000|50000|18
100	1.24e-03|1.33e-02|4.05e-08	5833.23|2809.22|5.17	50000|50000|26

From Tables 3 and 4, we can draw a similar conclusion as the experiments on the UCI and the BioNUS data sets discussed in Subsection 4.3. That is, our Newt-ALM method is far superior to the tested first-order methods in terms of computational efficiency and the ability to successfully solve the problems to the required level of accuracy.

4.5. The pathwise solution for a microarray data

The behavior of the OSCAR model for sparse feature selection and grouping for each specific instance relies heavily on the tuning parameters w_1 and w_2. To get a reliable and effective estimation for the coefficients of all involved predictors in the context of linear regression, a two-dimensional grid of various w_1 and w_2 values are tested to generate a solution path. The task of generating a solution path can be costly since each single pair of parameters (w_1, w_2) will lead to a different instance of the OSCAR model. The path usually begins with appropriately chosen parameters that shrink all the coefficients to zero, and moves on until we are near the un-regularized solution by varying the values of the parameters. During the construction of the solution path, the warm start strategy (Friedman et al., 2007, 2010) is always used to accelerate the entire process by using the previous near-by solution as the initial point for the next problem.

Here, we will use the microarray data set reported in Scheetz et al. (2006) and processed it by following Huang et al. (2008); Gu et al. (2018), where the design matrix $A \in \mathbb{R}^{m\times n}$ and the response vector $b \in \mathbb{R}^m$ with $m = 120$ and $n = 3000$. A partial solution path with the parameter w_2 fixed at the value $\|A^\top b\|_\infty/n^2$, and the parameter w_1 varying evenly in the interval $\left[10^{-4}, 10^{-2}\right] \times \|A^\top b\|_\infty$ for 100 different values will be constructed. The first 10 largest coefficients in magnitude of all the 100 numerical solutions are collected in Figure 2 by using ADMM, SLOPE and Newt-ALM, respectively. The timing comparison for generating the partial solution paths by these three algorithms is presented in Table 5.

Table 5: Computation time comparison among Newt-ALM, ADMM and SLOPE for generating the partial solution paths, where the row "Ratio" reports the ratios of the computation time of each single algorithm to that of the fastest algorithm.

	Newt-ALM	ADMM	SLOPE
Time(s)	27.74	323.65	1149.73
Ratio	1	11.7	41.4

As Figure 2 shows, all the three algorithms obtain almost the same partial solution paths for the microarray data with the above parameters setting. This is due to fact that the size of the data is relatively small ($[m, n] = [120, 3000]$), and all the instances corresponding to the chosen tuning parameter pairs have rather sparse solutions (most of them have less than 10 nonzero components in the numerical solutions) and hence have been successfully solved by all three algorithms. Even for such a nice scenario, the Newt-ALM is still more than 10 times and 40 times faster than ADMM and SLOPE, respectively, as shown in Table 5. For difficult cases, such as large scale problems or those with relatively dense solutions in the

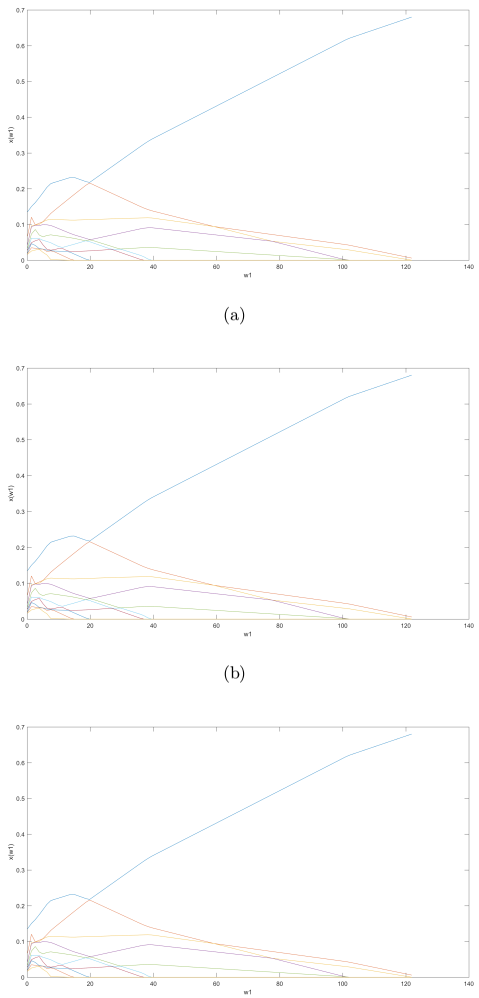


Figure 2: The partial solution paths with the first 10 largest coefficients in magnitude for the microarray data: (a) Newt-ALM; (b) ADMM; (c) SLOPE

high-dimensional linear regression, both the advantage in computation time and solution accuracy of our Newt-ALM will certainly be more significant, as can be observed in Table 1 and deduced from the computational complexity analysis in Subsection 3.4.

With a two-dimensional grid of varying w_1 and w_2 values, we can also construct a three-dimensional scattergram to show those first k (e.g., $k = 10$) largest components in magnitude for the microarray data. Figure 3 shows such a case, from which we can get a partial solution path along w_1 (or w_2) with any fixed w_2 (or w_1), or along any set of (w_1, w_2) pairs on the grid. Figure 3 shows the scattergram which collects the first 10 largest components in magnitude with w_1 and w_2 varying evenly in $\left[10^{-7}, 10^{-4}\right] \times \|A^\top b\|_\infty$ and $\left[10^{-4}, 10^{-2}\right] \times \|A^\top b\|_\infty / n$, respectively. All the $10,000$ problems are solved by our algorithm Newt-ALM in a total of about 70 minutes.

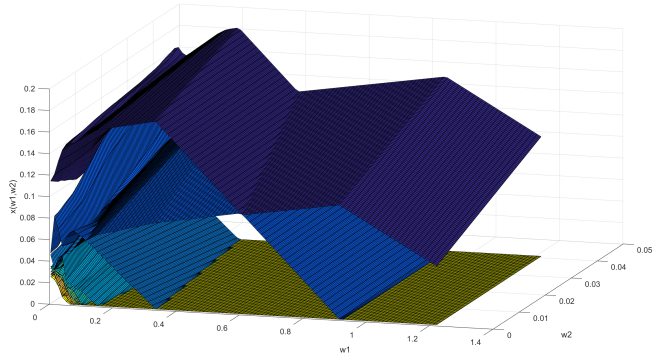


Figure 3: The first 10 largest components in magnitude of solutions with a two-dimensional grid of w_1 and w_2 values for the microarray data

5. Discussions

In this paper we have proposed an efficient semismooth Newton-based augmented Lagrangian method for solving the OSCAR and SLOPE models in high-dimensional statistical regressions from the dual perspective. Numerical results have demonstrated the overwhelming superiority of the proposed algorithm on high-dimensional real data sets, comparing to the widely-used APG and ADMM. It is noteworthy that the original OSCAR and SLOPE models have been transformed to their dual counterparts before applying our method to take the advantage of the high-dimensional setting (i.e., the number of coefficients to be estimated is far larger than the sample size). The success of our second-order iterative method, both in accuracy and in computation time, relies heavily on the subtle second-order sparsity structure present in the generalized Jacobian matrix that corresponds to the

second-order differential information of the underlying structured regularizer. Besides the least squares loss function adopted in the OSCAR and SLOPE models, our method is also applicable for the case of the logistic loss function, in which the desired nice properties of the corresponding subproblems are maintained to guarantee the efficiency and robustness of the algorithm. For classical statistical regression with larger sample size, our method is still applicable. But we may have to explore whether it is more efficient to apply our algorithmic framework directly to the OSCAR and SLOPE models, instead of our current application to the dual problem. The efficiency and effectiveness of our algorithm in solving high-dimensional linear regression with the OSCAR and SLOPE regularizers will greatly facilitate data analysis in statistical learning and related applications across a broad range of fields.

Acknowledgments

We thank the referees for the helpful suggestions to improve the presentation of the paper. We would like to acknowledge the financial support for the National Natural Science Foundation of China (11771038, 11431002) and “111” Project of China (B16002), a start-up research grant from the Hong Kong Polytechnic University, and the Academic Research Fund (R-146-000-257-112) of the Ministry of Education of Singapore.

Appendix A.

In this appendix we prove the following theorem from Section 2:

Theorem *Let $\lambda \in \mathbb{R}^n_+$ be such that $\lambda_1 \geq \cdots \geq \lambda_n$. Then $\mathcal{M}(\cdot)$ is a nonempty and compact valued, upper semicontinuous multifunction, and for any given $y \in \mathbb{R}^n$, every $M \in \mathcal{M}(y)$ is symmetric and positive semidefinite. Moreover, there exists a neighborhood U of y such that for all $y' \in U$,*

$$\mathrm{Prox}_{\kappa_\lambda}(y') - \mathrm{Prox}_{\kappa_\lambda}(y) - M(y' - y) = 0, \quad \forall M \in \mathcal{M}(y'). \tag{21}$$

Proof. Let $y \in \mathbb{R}^n$ be an arbitrary point. Then it is obvious that $\mathcal{M}(y)$ is a nonempty and compact set. The symmetric and positive semidefiniteness of $M \in \mathcal{M}(y)$ is trivial by the definitions in (6) and (8). Now we claim that there exists a neighborhood V of $y \in \mathbb{R}^n$ such that

$$\Pi^s(y') \subseteq \Pi^s(y), \quad \forall y' \in V.$$

This claim is trivial for $y = 0$ since $\Pi^s(0) = \mathbf{\Pi^s_n}$. For the case of a nonzero $y \in \mathbb{R}^n$, let r be the number of distinct values in $|y|$, and $t_1, \ldots, t_r$ be all those distinct values satisfying $t_1 > t_2 > \cdots > t_r \geq 0$. Consider the following two cases:

Case I: If $t_r > 0$, set $\delta := \frac{1}{3} \min \left\{ t_r, \min\limits_{1 \leq i \leq r-1} \{t_i - t_{i+1}\} \right\}$;

Case II: If $t_r = 0$, set $\delta := \frac{1}{3} \min\limits_{1 \leq i \leq r-1} \{t_i - t_{i+1}\}$.

It is easy to verify that in both cases $\delta > 0$ and

$$\Pi^s(y') \subseteq \Pi^s(y), \quad \forall y' \in \mathbf{B}(y, \delta), \tag{22}$$

where $\mathbf{B}(y, \delta)$ is the 2-norm ball centered at y with radius δ. The upper semicontinuity of $\mathcal{M}$ then can be obtained from (22) and (7). The remaining part is to show (21). For any $y' \in \mathbf{B}(y, \delta)$ with δ defined as above, it is known from (5) and the inclusion property in (22) that

$$\mathrm{Prox}_{\kappa_\lambda}(y') - \mathrm{Prox}_{\kappa_\lambda}(y) = \pi^{-1}\left(x_\lambda(\pi y') - x_\lambda(\pi y)\right), \quad \forall \pi \in \Pi^s(y'). \tag{23}$$

By combining the properties in (7) and the fact that $\|\pi y' - \pi y\| = \|y' - y\|$, we know that there exists a neighborhood $U \subseteq \mathbf{B}(y, \delta)$ of y such that for all $y' \in U$,

$$x_\lambda(\pi y') - x_\lambda(\pi y) = P(\pi y' - \pi y), \quad \forall P \in P(\pi y'), \forall \pi \in \Pi^s(y'),$$

which together with (23) leads to the desired result in (21). This completes the proof.

References

Jean-Pierre Aubin and Hélène Frankowska. *Set-Valued Analysis*. Birkhäuser, 1990.

Richard E. Barlow and H. D. Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Małgorzata Bogdan, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J. Candès. SLOPE-adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 9(3):1103–1140, 2015.

Howard D. Bondell and Brian J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1):115–123, 2008.

Liang Chen, Defeng Sun, and Kim-Chuan Toh. An efficient inexact symmetric Gauss-Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Mathematical Programming, Series A*, 161(1-2):237–270, 2017.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

Ying Cui, Defeng Sun, and Kim-Chuan Toh. On the R-superlinear convergence of the KKT residuals generated by the augmented lagrangian method for convex composite conic programming. *Mathematical Programming*, DOI: 10.1007/s10107-018-1300-6, 2018.

Maryam Fazel, Ting Kei Pong, Defeng Sun, and Paul Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 34(3):946–977, 2013.

Mario A. T. Figueiredo, Robert D. Nowak, and Stephen J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Jounral of Selected Topics in Signal Processing*, 1:586–597, 2007.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1):17–40, 1976.

Roland Glowinski and Americo Marrocco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'atomatique, Informatique Recherche Opérationelle. Analyse Numérique*, 9(2):41–76, 1975.

Yuwen Gu, Jun Fan, Lingchen Kong, Shiqian Ma, and Hui Zou. ADMM for high-dimensional sparse penalized quantile regression. *Technometrics*, 60(3):319–331, 2018.

Jiye Han and Defeng Sun. Newton and quasi-Newton methods for normal maps with polyhedral sets. *Journal of Optimization Theory and Applications*, 94(3):659–676, 1997.

Ronald R. Hocking. The analysis and selection of variables in linear regression. *Biometrics*, 32:1–49, 1976.

Jian Huang, Shuangge Ma, and Cun-Hui Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18(4):1603–1618, 2008.

Laurent Jacob, Guillaume Obozinski, and Jean Philippe Vert. Group lasso with overlap and graph lasso. In *International Conference on Machine Learning*, pages 433–440, 2009.

E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mo, C. Ropelewski, J. Wang, A. Leetmaa, R. Reynolds, R. Jenne, and Joseph J. D. The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American Meteorological Society*, 77:437–472, 1996.

Bernd Kummer. Newton's method for non-differentiable functions. *Advances in Mathematical Optimization*, 45:114–125, 1988.

Xudong Li, Defeng Sun, and Kim-Chuan Toh. An efficient linearly convergent semismooth Netwon-CG augmented lagrangian method for Lasso problems. *SIAM Journal on Optimization*, 28(1):433–458, 2018a.

Xudong Li, Defeng Sun, and Kim-Chuan Toh. On efficiently solving the subproblems of a level-set method for fused lasso problems. *SIAM Journal on Optimization*, 28:1842–1866, 2018b.

Moshe Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml/datasets.html`, 2013.

Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8:85–283, 2014.

Robert Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization*, 15(6):959–972, 1977.

Alan Miller. *Subset Selection in Regression*. Chapman & Hall, London, UK, 2002.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93(2):273–299, 1965.

Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. GAP safe screening rules for sparse-group lasso. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pages 388–396, 2016.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Liqun Qi and Jie Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58(1):353–367, 1993.

Franck Rapaport, Andrei Zinovyev, Marie Dutreix, Emmanuel Barillot, and Jean-Philippe Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:35, 2007.

Tim Robertson, Farroll T. Wright, and Richard Dykstra. *Order Restricted Statistical Inference*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, Chichester, 1988. ISBN 0-471-91787-7.

Stephen M. Robinson. Some continuity properties of polyhedral multifunctions. *Mathematical Programming Study*, 16:206–214, 1981.

Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

Ralph Tyrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976a.

Ralph Tyrell Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976b.

Ralph Tyrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Springer-Verlag, Berlin, 1998.

Todd E. Scheetz, Kwang-Youn A. Kim, Ruth E. Swiderski, Alisdair R. Philp, Terry A. Braun, Kevin L. Knudtson, Anne M. Dorrance, Gerald F. DiBona, Jian Huang, and Thomas L. Casavant. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.

Mervyn J. Silvapulle and Pranab Kumar Sen. *Constrained Statistical Inference: Order, Inequality, and Shape Constraints*, volume 912. John Wiley & Sons, 2011.

Defeng Sun and Jie Sun. Semismooth matrix-valued functions. *Mathematics of Operations Research*, 27(1):150–169, 2002.

Jie Sun. *On monotropic piecewise quadratic programming*. PhD thesis, University of Washington, 1986.

Robert Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society*, 58:267–288, 1996.

Joel A. Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, 51:1030–1051, 2006.

M. J. Van de Vijver, Y. D. He, L. J. Vant Veer, H. Hart A.A. Dai, Voskuil D. W., Schreiber G.J., Peterse J. L., Roberts C., Marton M. J., Parrish M., Atsma D., Witteveen A., Glas A., Delahaye L., Van der Velde T., Bartelink H., Rodenhuis S., Rutgers E.T., Friend S. H., and Bernards R. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347:1999–2009, 2002.

Yuhang Wang, Fillia S. Makedon, James C. Ford, and Justin Pearlman. Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics*, 21:1530–1537, 2005.

Bin Wu, Chao Ding, Defeng Sun, and Kim-Chuan Toh. On the Moreau–Yosida regularization of the vector k-norm related functions. *SIAM Journal on Optimization*, 24(2): 766–794, 2014.

Xiangrong Zeng and Mário AT Figueiredo. Solving OSCAR regularization problems by proximal splitting algorithms. *Digital Signal Processing*, 31:124–135, 2014a.

Xiangrong Zeng and Mário AT Figueiredo. Decreasing weighted sorted ℓ_1 regularization. *IEEE Signal Processing Letters*, 21(10):1240–1244, 2014b.

Yangjing Zhang, Ning Zhang, Defeng Sun, and Kim-Chuan Toh. An efficient hessian based algorithm for solving large-scale sparse group lasso problems. *Mathematical Programming*, DOI:10.1007/s10107-018-1329-6, 2018.

Leon Wenliang Zhong and James T. Kwok. Efficient sparse modeling with automatic feature grouping. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1436–1447, 2012.

Journal of Machine Learning Research 20 (2019) 1-34 Submitted 4/18; Revised 2/19; Published 6/19

Scalable Interpretable Multi-Response Regression via SEED

Zemin Zheng ZHENGZM@USTC.EDU.CN
School of Management and School of Data Science
International Institute of Finance
University of Science and Technololgy of China
Hefei, Anhui 230026, China

M. Taha Bahadori BAHADORI@GATECH.EDU
School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Yan Liu YANLIU.CS@USC.EDU
Computer Science Department
Viterbi School of Engineering
University of Southern California
Los Angeles, CA 90089, USA

Jinchi Lv JINCHILV@MARSHALL.USC.EDU
Data Sciences and Operations Department
Marshall School of Business
University of Southern California
Los Angeles, CA 90089, USA

Editor: Xiaotong Shen

Abstract

Sparse reduced-rank regression is an important tool for uncovering meaningful dependence structure between large numbers of predictors and responses in many big data applications such as genome-wide association studies and social media analysis. Despite the recent theoretical and algorithmic advances, scalable estimation of sparse reduced-rank regression remains largely unexplored. In this paper, we suggest a scalable procedure called sequential estimation with eigen-decomposition (SEED) which needs only a single top-r sparse singular value decomposition from a generalized eigenvalue problem to find the optimal low-rank and sparse matrix estimate. Our suggested method is not only scalable but also performs simultaneous dimensionality reduction and variable selection. Under some mild regularity conditions, we show that SEED enjoys nice sampling properties including consistency in estimation, rank selection, prediction, and model selection. Moreover, SEED employs only basic matrix operations that can be efficiently parallelized in high performance computing devices. Numerical studies on synthetic and real data sets show that SEED outperforms the state-of-the-art approaches for large-scale matrix estimation problem.

Keywords: reduced-rank regression, scalability, high dimensionality, greedy algorithm, sparse eigenvector estimation

©2019 Zemin Zheng, M. Taha Bahadori, Yan Liu and Jinchi Lv.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-200.html.

1. Introduction

Identifying complex dependence structures among predictors and responses is an important problem in statistics and machine learning, since these structures reveal hidden domain knowledge about the data. For example, in bioinformatics, identifying gene regulatory networks is crucial for understanding gene regulatory paths and gene functions, which helps disease prediction and diagnosis. Similarly, in social media analysis, inferring the influence networks from user activities, that is, *Diffusion Network Inference* problem (Leskovec et al., 2008; Zhou et al., 2013; Embar et al., 2014) is an important problem and it has found applications in social media marketing (Gomez-Rodriguez et al., 2012) and crisis management (Starbird and Palen, 2012). In these big data applications, inferring the dependence structures is challenging since the responses and predictors may be related through a few latent pathways and/or associated through only a subset of responses and predictors. Moreover, the curse of dimensionality and massive amounts of data, that is, scalability issues make the dependence structure discovery problem even harder to solve. To recover sparse response-predictor associations and latent predictors, regularization methods such as lasso (Tibshirani, 1996) and group lasso (Yuan and Lin, 2006), and reduced-rank regression approaches (Izenman, 1975; Velu and Reinsel, 2013) have been proposed, respectively. Chen et al. (2012) and Chen and Chan (2015) have proposed sparse reduced-rank regression approaches by combining the regularization and reduced-rank regression techniques to find the complex dependence structures between responses and predictors.

Sparse reduced-rank regression works by modeling the associations between the predictor and response variables via a sparse and low-rank representation of the coefficient matrix. It not only enhances the interpretability of the estimated matrix by eliminating irrelevant features (Chen et al., 2012), but also reduces the number of free parameters of the model and thus the number of observations required for desired estimation consistency (Yuan et al., 2007; Bunea et al., 2011; Candès and Plan, 2011; Negahban and Wainwright, 2011; Chen et al., 2013). Sparse reduced-rank regression has found applications in micro-array biclustering (Chen et al., 2012), subspace clustering (Wang et al., 2013), social network community discovery (Richard et al., 2012; Zhou et al., 2013), and motion segmentation (Feng et al., 2014). In these applications, joint sparsity and low-rankness has been used to enforce a clustered dependence structure among data points. In particular, the key idea is to estimate a similarity matrix among data points that is simultaneously sparse and low-rank and then permute the rows and columns of the matrix to yield approximately block-diagonal structures, which naturally lead to clustering of data points into several groups. Note that Chandrasekaran et al. (2010), Agarwal et al. (2012), and the references therein have considered estimating matrices with a low-rank plus sparse representation which is different from our work as we are interested in estimating a matrix that is jointly low-rank and sparse.

A natural approach to solving the sparse reduced-rank regression problem is to simultaneously penalize the parameter matrix using the L_1 and nuclear norm regularizers, as they are convex relaxations to sparsity and low-rankness of a matrix, respectively. The resulting optimization problem is convex and can be solved using the alternating direction method of multipliers (ADMM) (Boyd et al., 2010) as proposed by Richard et al. (2012) and Zhou et al. (2013). In Bunea et al. (2012), an alternative approach, called rank constrained group

lasso (RCGL), was proposed which directly penalizes the rank and the number of nonzero rows of the parameter matrix. They showed oracle rates for the estimated matrix and also provided a practical algorithm which iteratively and jointly solves a L_1-regularization and low-rank estimation problem. To further improve the estimation accuracy, Chen et al. (2012) borrowed ideas from adaptive Lasso (Zou, 2006) and proposed the iterative exclusive extraction algorithm (IEEA) which finds a locally optimal solution in the neighborhood of the initial value. They also showed model selection consistency and asymptotic normality results along with the improved empirical performance of IEEA on microarray biclustering analysis data.

All the above approaches for sparse reduced-rank regression achieve both desirable theoretical properties and strong empirical results. However, they cannot scale to large matrix estimation problems in many big data applications. The ADMM algorithm of Richard et al. (2012) and Zhou et al. (2013) uses iterative singular value thresholding (Cai et al., 2010) for solving the joint L_1 and nuclear norm regularization. Iterative singular value thresholding is known to be computationally expensive since it performs a full singular value decomposition of the parameter matrix in each iteration. On the other hand, RCGL (Bunea et al., 2012) is computationally much faster than ADMM since it only performs top-r singular value decomposition for estimating a rank-r matrix in each iteration. Despite a lower computational cost per iteration, it is unclear how many iterations RCGL needs for convergence. IEEA (Chen et al., 2012) performs nested loops of alternating L_1-penalized regression for each singular vector which can be expensive, especially on parallel computing devices. The iterative nature of these three approaches makes them not scalable and renders them inefficient for large matrix estimation even on high performance computing devices.

To overcome the scalability issues of the previous approaches, we propose a simple and scalable sparse reduced-rank regression procedure called sequential estimation with eigen-decomposition (SEED). SEED is designed for high-performing computing platforms. It converts the sparse and low-rank regression problem to a sparse generalized eigenvalue problem and then solves the problem using the recent algorithms for sparse eigenvalue decomposition (Cai et al., 2013; Ma, 2013; Yuan and Zhang, 2013). As a pure learning algorithm, SEED is expected to perform only a single top-r sparse eigenvalue decomposition for estimating a rank-r matrix, which makes it truly scalable and efficient for large matrix estimation problems.

The main contributions of our paper are threefold. First, for the sparse reduced-rank regression problem, our proposed procedure SEED provides a scalable approach to uncovering the sparse predictor-response association network while simultaneously achieving dimension reduction and variable selection. Second, for the high-dimensional settings, our theoretical analysis shows that SEED can consistently estimate the singular vectors, latent factors as well as the regression coefficient matrix, identify the correct rank of the matrix, accurately predict the multivariate response vector, and recover the support of the singular vectors under mild conditions. Note that, compared with Chen et al. (2012), we do not make any assumption on the positive definiteness of the design matrix for proving our consistency results. Third, we empirically demonstrate that SEED can not only be efficiently implemented on both central processing units (CPU) and graphics processing units (GPU) for large-scale applications, but it also outperforms the state-of-the-art sparse reduced-rank regression approaches.

Recently, Mishra et al. (2017) proposed the sequential co-sparse factor regression in a similar sparse and low-rank model setting, where a unit rank sparse coefficient matrix was recovered at each step such that both the left and the right singular vectors could be estimated with co-sparse pattern (an earlier version of the current paper was cited in theirs). Compared with their method, our approach separates the estimation of the singular vectors at each step with the left ones obtained through a generalized sparse eigenvalue problem and the right ones recovered directly based on the left ones. Moreover, we provide comprehensive theoretical properties to justify the validity of sequential estimation for sparse and low-rank coefficient matrices in high dimensions.

The rest of this paper is organized as follows. Section 2 introduces our SEED method. We discuss the implementation details of SEED in Section 3 and present its asymptotic properties in Section 4. We demonstrate the advantages of SEED on both synthetic and real data sets in Section 5, and in Section 6 we discuss some extensions of our SEED method. The proofs of some main results are relegated to the Appendix. Additional technical details are provided in the Supplementary Material.

2. Sequential Estimation with Eigen-Decomposition

We will first introduce the model setup and then the proposed methodology.

2.1. Model and Problem Formulation

Denote by $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ n observations in the fixed design setting, where $\mathbf{x}_i \in \mathbb{R}^p$ and $\mathbf{y}_i \in \mathbb{R}^q$ represent the ith predictor and the corresponding response vectors, respectively. Given a predictor vector $\mathbf{x}$, the corresponding response vector $\mathbf{y}$ is drawn from the following model

$$\mathbf{y} = \mathbf{C}^{*T}\mathbf{x} + \boldsymbol{\varepsilon},$$

where the noise vector $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is a q-dimensional multivariate Gaussian random vector with mean zero and covariance matrix $\boldsymbol{\Sigma}$, and $\mathbf{C}^* \in \mathbb{R}^{p\times q}$ is the regression coefficient matrix.[1] We can rewrite the model in the matrix form as follows

$$\mathbf{Y} = \mathbf{X}\mathbf{C}^* + \mathbf{E}, \tag{1}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^T$, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$, and $\mathbf{E} = [\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_n]^T$ denote the matrices of stacked response, predictor and noise vectors, respectively.

Let $\mathbf{P} = n^{-1}\mathbf{X}^T\mathbf{X}$ be the Gram matrix of the predictors. We consider model (1) from a latent factor point of view, where the true regression coefficient matrix $\mathbf{C}^*$ is jointly low-rank and sparse, similar to Bunea et al. (2012) and Chen et al. (2012). In particular, $\text{rank}(\mathbf{C}^*) = r^*$ with the matrix rank $r^* \ll \min(p, q)$. Without loss of generality, we assume that $\text{rank}(\mathbf{X}\mathbf{C}^*) = \text{rank}(\mathbf{C}^*)$ since if $\text{rank}(\mathbf{X}\mathbf{C}^*) < \text{rank}(\mathbf{C}^*)$, the redundant part of $\mathbf{C}^*$ can be removed such that it reflects the true number of latent factors. Then $\mathbf{C}^*$ will have the following decomposition

$$\mathbf{C}^* = \sum_{k=1}^{r^*} \mathbf{u}_k^* \mathbf{v}_k^{*T} = \sum_{k=1}^{r^*} \mathbf{C}_k^*, \tag{2}$$

1. Note that the Gaussianity of noise variables is not essential to either our procedure or the theoretical results and similar results would hold under the sub-Gaussian assumption (Bühlmann and van de Geer, 2011, Chapter 14).

where the left singular vectors $\mathbf{u}_k^* \in \mathbb{R}^p$ are $\mathbf{P}$-orthogonal with unit length, that is, $\mathbf{u}_k^{*T}\mathbf{P}\mathbf{u}_{k'}^* = 0$ if $k \neq k'$ and $\|\mathbf{u}_k^*\|_2 = 1$, while the right singular vectors $\mathbf{v}_k^* \in \mathbb{R}^q$ are orthogonal, that is, $\mathbf{v}_k^{*T}\mathbf{v}_{k'}^* = 0$ for $k \neq k'$, and $\mathbf{C}_k^*$ is the layer k unit rank matrix of $\mathbf{C}^*$. The singular vectors are sorted by the magnitudes of the singular values $\sigma_k = \frac{1}{\sqrt{nq}}\|\mathbf{X}\mathbf{C}_k^*\|_F$ in descending order, consistent with their contributions to the prediction of $\mathbf{Y}$.

We consider the left singular vectors (both the population and estimated ones) in the constraint space $\mathbf{u} \perp \mathrm{Ker}(\mathbf{P})$, where $\mathrm{Ker}(\mathbf{P})$ denotes the null space of $\mathbf{P}$, to guarantee the model identifiability. Otherwise, $\mathbf{u}$ would contain certain component $\widetilde{\mathbf{u}}$ such that $\mathbf{X}\widetilde{\mathbf{u}} = 0$, which does not contribute to the prediction of $\mathbf{Y}$. It is worth noticing that the decomposition in the form of $\mathbf{C}^* = \sum_{k=1}^{r^*}\mathbf{u}_k^*\mathbf{v}_k^{*T}$ is not unique without orthogonality constraints and the entrywise sparsity in the singular vectors is not invariant to rotations. The decomposition in (2) is a special one since the $\mathbf{P}$-orthogonality of $\mathbf{u}_k^*$ arises naturally from the power method and leads to a latent factor analysis interpretation of the reduced rank estimation in which the latent factors $\mathbf{X}\mathbf{u}_k^*$ are uncorrelated. Moreover, decomposition (2) coincides with the singular value decomposition of $\mathbf{X}\mathbf{C}^*$ except for different scalings on the singular vectors. We defer the discussion on the identifiability of decomposition (2) (existence and uniqueness up to simultaneous sign changes of $\mathbf{u}_k^*$ and $\mathbf{v}_k^*$) to Supplementary Material.

The aforementioned modeling of the regression coefficient matrix gives a latent factor model with r^* latent factors, where $\mathbf{X}\mathbf{u}_k^*$ is the kth latent factor and $\mathbf{v}_k^*$ describes the impacts of the kth factor on the response variables. As illustrated in Yuan et al. (2007), the low-rankness of $\mathbf{C}^*$ renders dimension reduction such that all responses can be predicted by a relatively small set of common factors. Furthermore, the left singular vectors $\mathbf{u}_k^*$ are assumed to be sparse, yielding the necessity of predictor selection. Similar sparsity assumptions were made in Bunea et al. (2012) and Chen et al. (2012) to enhance model interpretability by removing irrelevant features in high dimensions. Specifically, Chen et al. (2012) assumed that both the left and right singular vectors are sparse while Bunea et al. (2012) imposed restriction on the number of nonzero rows of the regression coefficient matrix. In this paper, we are interested in two cases: (i) when the right singular vectors are not required to be sparse and (ii) when it is desirable to have sparse right singular vectors, which entails the response selection. We will show that both cases are efficiently accommodated by our procedure.

Our goal is to accurately estimate the singular vectors $\mathbf{u}_k^*$ and $\mathbf{v}_k^*$, and the true rank r^* such that we can recover the latent factors, their impacts, and the underlying number of latent factors as well as the significant predictors. As a singular vector can have two opposite directions, we always assume that the estimated left singular vector takes the correct one, that is, the angles between estimated and population left singular vectors are no more than a right angle. Once the estimated rank and singular vectors are obtained, the estimate $\widehat{\mathbf{C}}$ of the true matrix $\mathbf{C}^*$ follows immediately from (2). Unlike most existing sparse and low-rank estimation methods which adopt the regularization framework of minimizing a loss function plus certain penalties, we will show that the proposed procedure SEED is indeed a pure learning algorithm that predicts $\mathbf{Y}$ using $\mathbf{X}\widehat{\mathbf{C}}$.

2.2. Description of SEED

The following proposition provides us insight into estimating the top-r^* left and right singular vectors of $\mathbf{C}^*$.

Proposition 1 *Consider the noiseless case $\mathbf{Y}^* = \mathbf{X}\mathbf{C}^*$ with $\mathbf{C}^* = \sum_{k=1}^{r^*}\mathbf{u}_k^*\mathbf{v}_k^{*T}$ defined in (2). Then $\mathbf{u}_1^*, \ldots, \mathbf{u}_{r^*}^*$ are the r^* non-degenerate left singular vectors of $\mathbf{C}^*$ if and only if they are eigenvectors of the following generalized eigenvalue problem*

$$\mathbf{X}^T\mathbf{Y}^*\mathbf{Y}^{*T}\mathbf{X}\mathbf{u} = \lambda\mathbf{X}^T\mathbf{X}\mathbf{u} \tag{3}$$

with respect to the nonzero eigenvalues $\lambda_1, \cdots, \lambda_{r^*}$, where $\lambda_k = nq\sigma_k^2$ is the kth largest eigenvalue of $\mathbf{Y}^*\mathbf{Y}^{*T}$ with $\sigma_k = \|\mathbf{X}\mathbf{C}_k^*\|_F/\sqrt{nq}$ defined in Section 2.1. Furthermore, given the left singular vector $\mathbf{u}_k^*$, the corresponding right singular vector $\mathbf{v}_k^*$ can be written as

$$\mathbf{v}_k^* = \frac{1}{\mathbf{u}_k^{*T}\mathbf{X}^T\mathbf{X}\mathbf{u}_k^*}\mathbf{Y}^{*T}\mathbf{X}\mathbf{u}_k^*. \tag{4}$$

Proposition 1 shows that the problem of estimating the singular vectors can be transformed into the generalized eigenvalue problem in (3), thanks to the $\mathbf{P}$-orthogonality of the left singular vectors. It motivates us to estimate the left singular vectors in the noisy case $\mathbf{Y} = \mathbf{X}\mathbf{C}^* + \mathbf{E}$ by solving the following generalized eigenvalue problem

$$\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{u} = \lambda\mathbf{X}^T\mathbf{X}\mathbf{u}. \tag{5}$$

The estimation consistency can be guaranteed by the matrix perturbation theory (see Section 4 for details). On the other hand, it is not difficult to see that the eigenvectors with respect to different eigenvalues of problem (5) are $\mathbf{P}$-orthogonal, which further gives the orthogonality of the right singular vectors estimated according to (4). It implies that the right and left singular vectors obtained by solving the generalized eigenvalue problem will automatically be orthogonal and $\mathbf{P}$-orthogonal, respectively.

Related results of principal component analysis in low dimensions can be found in Baldi and Hornik (1989), Diamantaras and Kung (1996), and De La Torre and Black (2003). Note that in the high-dimensional setting, the regime of interest for this paper, the Gram matrix $\mathbf{P}$ can not be invertible and the generalized eigenvalue problem is potentially challenging to solve. We will address the implementation challenges in Section 3.

Based on Proposition 1, our proposed procedure SEED performs a two-step estimation for the regression coefficient matrix. It first solves the generalized eigenvalue problem (5) to obtain the estimated left singular vectors $\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_r$ with unit length and then finds the estimated right singular vectors $\widehat{\mathbf{v}}_1, \ldots, \widehat{\mathbf{v}}_r$ according to (4) as follows

$$\widehat{\mathbf{v}}_k = \frac{1}{\widehat{\mathbf{u}}_k^T\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}_k}\mathbf{Y}^T\mathbf{X}\widehat{\mathbf{u}}_k. \tag{6}$$

The maximum rank r depends on the magnitude of the estimated singular value $\widehat{\sigma}_k = \|\mathbf{X}\widehat{\mathbf{C}}_k\|_F/\sqrt{nq}$ with $\widehat{\mathbf{C}}_k = \widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^T$ (whether it is larger than a threshold μ) and the optimal rank can be tuned by cross validation or the information criterion described in Section 4.

Algorithm 1: SEED

Input: (1) Data $\mathbf{Y} \in \mathbb{R}^{n\times q}$ and $\mathbf{X} \in \mathbb{R}^{n\times p}$ (2) Termination accuracy μ and sparsity parameter θ

1 Compute matrices: $\mathbf{P} \leftarrow \frac{1}{n}\mathbf{X}^T\mathbf{X}$, $\mathbf{R} \leftarrow \frac{1}{n}\mathbf{Y}^T\mathbf{X}$, $\mathbf{Q} \leftarrow \frac{1}{q}\mathbf{R}^\top\mathbf{R}$
2 $k \leftarrow 1$
3 **repeat**
4 $\quad(\widehat{\mathbf{u}}_k, \widehat{\sigma}_k) \leftarrow k$th θ-sparse eigenvector and eigenvalue of $\mathbf{Qu} = \lambda\mathbf{Pu}$
5 $\quad$**if** $\widehat{\sigma}_k > \mu$ **then**
6 $\quad\quad\widehat{\mathbf{v}}_k \leftarrow \frac{1}{\widehat{\mathbf{u}}_k^\top\mathbf{P}\widehat{\mathbf{u}}_k}\mathbf{R}\widehat{\mathbf{u}}_k$ $\quad$ *#Optional thresholding of $\widehat{\mathbf{v}}_k$ for sparsity*
7 $\quad\quad\widehat{\mathbf{C}}_k \leftarrow \widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^\top$
8 $\quad\quad k \leftarrow k+1$
9 $\quad$**end**
10 **until** $\widehat{\sigma}_k < \mu$
11 tune the optimal rank $\widehat{r}$
12 **return** $\widehat{\mathbf{C}} = \sum_{k=1}^{\widehat{r}}\widehat{\mathbf{C}}_k$

The details of the procedure are provided in Algorithm 1. To achieve a sparse solution, we need to find the rank-r sparse matrix via a sparse eigenvalue decomposition procedure in Line 4 of Algorithm 1. This can be done in a sequential way and practical methods will be provided in Section 3. The practical methods need a sparsity parameter θ, such as a threshold (Ma, 2013) or a sparsity size (Yuan and Zhang, 2013). We will show in Section 5 that SEED is robust to the choices of parameters θ and μ. If the right singular vectors are also required to be sparse, we perform a simple element-wise thresholding on $\widehat{\mathbf{v}}_k$ after we obtain it in Line 6.

3. Scalable Implementation of SEED

Algorithm 1 requires a sparse solution of (5) which is a generalized eigenvalue problem with a rank deficient matrix $\mathbf{P}$. In this section, we propose two different procedures to solve (5) and study multiple practical aspects of SEED.

3.1. Fast Approach

The bottleneck in speeding up Algorithm 1 is Line 4 where we need to solve a sparse generalized eigenvalue problem. To overcome this bottleneck, we propose a new solution to estimating the left singular vectors by rewriting equation (5) as

$$\mathbf{X}^T(\mathbf{YY}^T - \lambda\mathbf{I})\mathbf{Xu} = \mathbf{0}.$$

Similar to Proposition 1, when $\mathbf{X}$ is of full row rank (which is easy to satisfy in the high-dimensional setting), the above equation shares the same nonzero eigenvalues with $\mathbf{YY}^T$. Even if $\mathbf{X}$ is row rank deficient, we still have nonzero solution $\mathbf{u}$ when the perturbation is relatively small, ensured by the perturbation theory in Lemma 6. Thus, we propose the following two-step procedure for Line 4 of Algorithm 1:

(1) $\lambda \leftarrow \lambda_{\max}(\mathbf{YY}^T)$.

(2) $\widehat{\mathbf{u}}_1 \leftarrow$ sparse eigenvector relative to the zero eigenvalue of $\mathbf{X}^T\mathbf{YY}^T\mathbf{X} - \lambda\mathbf{X}^T\mathbf{X}$.

Based on $\widehat{\mathbf{u}}_1$, the corresponding right singular vector $\widehat{\mathbf{v}}_1$ and the unit rank matrix $\widehat{\mathbf{C}}_1$ will be obtained. To continue, we compute the residual $\mathbf{Y}_k = \mathbf{Y} - \mathbf{X}\sum_{j=1}^{k-1}\widehat{\mathbf{C}}_j$ in the kth step and replace $\mathbf{Y}$ with $\mathbf{Y}_k$ in the above two-step procedure to obtain the kth left singular vector $\widehat{\mathbf{u}}_k$. Overall, the first step requires calculation of the top-r eigenvalues for an $n \times n$ matrix (or $q \times q$ if $q < n$) while the second step finds the corresponding eigenvectors by solving a regular sparse eigenvalue problem.

The above procedure significantly accelerates the speed of SEED as it converts a degenerate sparse generalized eigenvalue problem to two simpler regular sparse eigenvalue problems. Existing procedures such as the iterative thresholding method (Ma, 2013) can be used to compute both eigenvalue problems efficiently. Generally speaking, SEED will be extremely efficient in applications with low rank structure such as image processing as it stops early after achieving a few important signals. Moreover, the speed of SEED can be greatly enhanced by parallel implementation on high performance computing devices such as GPU, due to the fact that it employs only basic matrix operations.

3.2. Alternative with Enhanced Stability

In cases where the perturbation can be large, for numerical stability purposes, we can also solve (5) by the following modified problem with a very small positive ρ:

$$\mathbf{X}^T\mathbf{YY}^T\mathbf{Xu} = \lambda(\mathbf{X}^T\mathbf{X} + \rho\mathbf{I})\mathbf{u}. \tag{7}$$

Note that $\mathbf{X}^T\mathbf{X} + \rho\mathbf{I}$ is invertible since the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are nonnegative. Denote by $\widetilde{\mathbf{X}} \in \mathbb{R}^{p\times p}$ the modified predictor matrix such that $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} = \mathbf{X}^T\mathbf{X} + \rho\mathbf{I}$, which can be obtained via the Cholesky decomposition, and $\widetilde{\mathbf{Y}} = (\widetilde{\mathbf{X}}^T)^{-1}\mathbf{X}^T\mathbf{Y}$ the modified response matrix. Then the above equation (7) can be rewritten as

$$\widetilde{\mathbf{X}}^T\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{X}}\mathbf{u} = \lambda\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}\mathbf{u}, \tag{8}$$

which adopts the same form as (5).

A computationally efficient technique for solving equation (8) is to solve the sparse eigenvalue problem $\widetilde{\mathbf{P}}^{-1}\widetilde{\mathbf{Q}}\mathbf{u} = \lambda\mathbf{u}$, where the modified Gram matrix $\widetilde{\mathbf{P}} = n^{-1}\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Q}}$ is defined accordingly as in Algorithm 1. We can compute $\widetilde{\mathbf{P}}^{-1}$ by the Sherman-Morrison-Woodbury formula as follows:

$$(\rho\mathbf{I}_p + \mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{\rho}\mathbf{I}_p - \frac{1}{\rho^2}\mathbf{X}^T(\mathbf{I}_n + \frac{1}{\rho}\mathbf{XX}^T)^{-1}\mathbf{X}.$$

The above equation requires inversion of an $n \times n$ matrix instead of a $p \times p$ matrix, which is significantly faster in the high-dimensional setting when $p \gg n$.

Remark. The formulation of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ can be regarded as a generalization of the ridge regression to the multivariate response setting. In fact, since $\mathbf{C}^*$ is the minimizer

of $\|\mathbf{Y}^* - \mathbf{X}\mathbf{C}\|_F^2$, we can enhance the stability by adding a small Frobenius norm regularization as follows:

$$\widetilde{\mathbf{C}} = \text{argmin}_{\mathbf{C}} \left\{ \|\mathbf{Y} - \mathbf{X}\mathbf{C}\|_F^2 + \rho \|\mathbf{C}\|_F^2 \right\},$$

where the Frobenius norm is defined as $\|\mathbf{C}\|_F^2 = \sum_{i,j} \mathbf{C}_{i,j}^2$ for any matrix $\mathbf{C}$. After completing the squares, we get

$$\widetilde{\mathbf{C}} = \text{argmin}_{\mathbf{C}} \left\{ \|\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\mathbf{C}\|_F^2 \right\},$$

which means that $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ are the corresponding predictor and response matrices that take into account the shrinkage effects (James and Stein, 1961; Zheng et al., 2014).

3.3. Practical Aspects

Now we analyze several practical aspects of SEED as follows.

Refitting. In Algorithm 1, a refitting can be performed during the eigenvalue decomposition in Line 4 to enhance the stability. The refitting procedure is as follows. In the kth step, we perform the top-k singular value decomposition $\widehat{\mathbf{C}} = \mathbf{U}\widehat{\mathbf{S}}\mathbf{V}^T$ for $\widehat{\mathbf{C}} = \sum_{j=1}^{k} \widehat{\mathbf{C}}_j$ and refit the solution by finding $\widetilde{\mathbf{S}} = \text{argmin}_{\mathbf{S}} \left\|\mathbf{Y} - \mathbf{X}\mathbf{U}\mathbf{S}\mathbf{V}^T\right\|_F^2$. The estimate with refitting is defined as $\widetilde{\mathbf{C}} = \mathbf{U}\widetilde{\mathbf{S}}\mathbf{V}^T$. In practice, we find this approach more stable and report the results based on this variation of SEED in numerical studies.

Sparse eigenvector estimation. The previous approaches for solving the generalized eigenvalue decomposition problem indicate that we can solve the problem via regular sparse eigenvalue decomposition. This allows us to reuse the existing procedures for sparse eigenvalue decomposition such as Cai et al. (2013), Ma (2013), Yuan and Zhang (2013), and Lei and Vu (2015) to solve the problem in (5). In numerical studies, we use the iterative thresholding method (Ma, 2013), which is detailed in Algorithm 2 for estimating the sparse eigenvector relative to the largest eigenvalue of a given sparse eigenvalue problem $\mathbf{S}\mathbf{u} = \lambda\mathbf{u}$ with sparsity parameter θ (threshold level).

It is worth pointing out that although sparse eigenvalue decomposition is a nonconvex problem, Ma (2013) proved that the sparse eigenvectors obtained by the iterative thresholding algorithm will converge to their population counterparts with asymptotic probability one within $O(\log n)$ iterations. The initial estimate $\widehat{\mathbf{u}}^{(0)}$ is generated from the diagonal thresholding sparse PCA algorithm proposed in Johnstone and Lu (2009), which is a regular eigenvalue problem after constraining on the significant coordinates (variances). This iterative thresholding algorithm can also be applied to recover several sparse eigenvectors simultaneously.

Parallel implementation. In order to scale up the procedures, we often need to utilize the parallel computing tools. Given that SEED only uses basic matrix operations, we can employ parallel implementation of the large matrix operations in high performance computing devices to accelerate SEED. In Section 5, our experiments with GPU which contain thousands of processing units show that the matrix operations in SEED can be efficiently parallelized and it significantly enhances the speed of SEED. Whenever the data can not be loaded into the memory of a single device, efficient distributed algorithms can be used. See for example, Kang et al. (2011) and the references therein.

Algorithm 2: Iterative thresholding

Input: Matrix $\mathbf{S} \in \mathbb{R}^{p\times p}$, threshold level θ, and initial estimate $\widehat{\mathbf{u}}^{(0)} \in \mathbb{R}^p$.

$k \leftarrow 1$
repeat
  Multiplication: $\mathbf{t}^{(k)} = (t_{ij}^{(k)}) = \mathbf{S}\widehat{\mathbf{u}}^{(k-1)}$
  Thresholding: $\widehat{\mathbf{t}}^{(k)} = (\widehat{t}_{ij}^{(k)})$ with $\widehat{t}_{ij}^{(k)} = t_{ij}^{(k)} 1_{(|t_{ij}|\geq\theta)}$
  QR factorization: $\widehat{\mathbf{u}}^{(k)} R^{(k)} = \widehat{\mathbf{t}}^{(k)}$ with $R^{(k)} = \|\widehat{\mathbf{t}}^{(k)}\|_2$
  $k \leftarrow k+1$.
until *convergence*
return $\widehat{\mathbf{u}}^{(k)}$ *at convergence*

4. Asymptotic Properties of SEED

In this section, we will analyze the statistical properties of SEED. Define the maximum sparsity level of the left singular vectors as $s^* = \max_{k=1}^{r^*} \|\mathbf{u}_k^*\|_0 \ll p$, which is assumed mainly for theoretical analysis (see Condition 1 below) and unknown in our practical implementation. We consider the estimated left singular vectors with the number of nonzero elements less than certain sparsity level $s > s^*$, that is, $\|\widehat{\mathbf{u}}_k\|_0 \leq s$ for $k = 1, \cdots, r$, where $r > r^*$ is an upper bound of the estimated rank that can be controlled by the algorithm in practice. When the generalized eigenvalue problem (5) does not have an s-sparse solution, the estimated largest s-sparse eigenvalue can be reformulated as

$$\widehat{\lambda} = \max_{\mathbf{u}\neq\mathbf{0}, \|\mathbf{u}\|_0 \leq s} \frac{\mathbf{u}^T(\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X})\mathbf{u}}{\mathbf{u}^T(\mathbf{X}^T\mathbf{X})\mathbf{u}}$$

due to the extremal property of eigenvalues, and the corresponding maximizer $\widehat{\mathbf{u}}$ will be treated as the estimated left singular vector. Note that there is no sparsity constraint on either the population or the estimated right singular vectors since $\mathbf{v}_k^*$ can be recovered from (4) based on accurate estimation of $\mathbf{u}_k^*$.

4.1. Technical Conditions

Here we list a few technical conditions and discuss their relevance in detail.

Condition 1 (Restricted Isometry) *There exists a positive constant ϕ_s such that the Gram matrix $\mathbf{P}$ satisfies*

$$\phi_s \leq \min_{\mathbf{z}\in\mathbb{R}^p} \left\{ \frac{\|\mathbf{P}\mathbf{z}\|_2}{\|\mathbf{z}\|_2} : \|\mathbf{z}\|_0 \leq 2s \right\} \leq \max_{\mathbf{z}\in\mathbb{R}^p} \left\{ \frac{\|\mathbf{P}\mathbf{z}\|_2}{\|\mathbf{z}\|_2} : \|\mathbf{z}\|_0 \leq 2s \right\} \leq \phi_s^{-1}$$

for some $s > s^*$.

Condition 2 (Minimum Singular Value Separation) *The non-zero singular values σ_k satisfy $\sigma_k^2 - \sigma_{k+1}^2 \geq d_\sigma > 0$ for some positive constant d_σ and $k = 1, \ldots, r^*$.*

Condition 3 (Bounded Eigenvalues) *The eigenvalues of the population covariance matrix of the noise vector $\boldsymbol{\varepsilon}$ satisfy $0 < \gamma_l^2 \le \lambda_j(\boldsymbol{\Sigma}) \le \gamma_u^2 < \infty$ for $j = 1, \ldots, q$, where γ_l and γ_u are positive constants with $\gamma_u \le c_\gamma \sigma_{r^*}$ for some positive constant c_γ.*

Condition 4 (Minimum Signal Strength) *There exists some positive constant $\delta \in (0,1)$ such that the following lower bounds on the magnitudes of the non-zero elements of $\mathbf{u}_k^*$ and $\mathbf{v}_k^*$ hold for any $1 \le k \le r^*$*

$$\min_{i \in \mathrm{supp}(\mathbf{u}_k^*)} |u_i^*| \ge 3C_u \sqrt{\frac{s}{n} \log \frac{pq}{\delta}},$$
$$\min_{i \in \mathrm{supp}(\mathbf{v}_k^*)} \frac{1}{\sqrt{q}} |v_i^*| \ge 3C_v \sqrt{\frac{s}{n} \log \frac{pq}{\delta}},$$

where C_u and C_v are constants defined in Theorem 1.

Condition 1 imposes bounds on the $2s$-sparse eigenvalues of $\mathbf{P}$, which is weaker than the regular bounded eigenvalue assumption since the sparse eigenvalues do not grow as fast as the regular eigenvalues when the dimensionality p increases. As a typical condition in high dimensions, it restricts the correlations between small numbers of features and thus guarantees the identifiability of the true support. See, for instance, Candès and Tao (2005) and Zhang (2011) for more discussion on it.

Recall that σ_k is the kth largest singular value of $\mathbf{X}\mathbf{C}^*/\sqrt{nq}$. Condition 2 requires strict separation among the singular values such that the left singular vectors are distinguishable. For ease of presentation, we assume d_σ to be a constant and indicate the roles of d_σ clearly in the constants of the theoretical results. In fact, $\mathbf{X}\mathbf{C}^*$ adopts a spiked eigen-structure (Johnstone, 2001; Shen et al., 2016) with the spiked singular values allowed to diverge at the rate of $\sqrt{nq}$. This rate is reasonable for an n by q matrix as we do not impose any sparsity on the columns of $\mathbf{C}^*$.

The elements of the unobserved noise vector $\boldsymbol{\varepsilon}$ were assumed to be independent and identically distributed (i.i.d.) in Bunea et al. (2012). We relax it a bit in Condition 3 by imposing bounded eigenvalues for the noise covariance matrix for recovering the true rank in Theorem 2. Our technical argument still applies when either $\gamma_l \to 0$ or $\gamma_u \to \infty$ as long as their rates of convergence can be controlled within certain magnitudes.

The two inequalities in Condition 4 are imposed for the model selection consistency of the predictors and responses, respectively. The magnitude of the minimum signal strength is $O\left(\sqrt{s\log(pq)/n}\right)$, which is relatively mild as it converges to zero in our setting. Since $\mathbf{u}_k^*$ is assumed to have unit length, the singular values of $\mathbf{C}^*$ are absorbed into $\mathbf{v}_k^*$ in view of decomposition (2) so that there is an extra scaling factor $\frac{1}{\sqrt{q}}$ in the second inequality.

4.2. Main Results

Denote by $P^2 = \max_{j=1}^p \mathbf{P}_{jj}$ and $\gamma^2 = \max_{j=1}^q \boldsymbol{\Sigma}_{jj}$ the maximum diagonal components of the Gram matrix and noise covariance matrix, respectively. Without loss of generality, we assume that $V = \max_{k=1}^{r^*} \frac{1}{\sqrt{q}} \|\mathbf{v}_k^*\|_2$ is finite for the q-dimensional vectors $\mathbf{v}_k^*$. Moreover, it is clear that under Conditions 1 and 3, P and γ are also finite constants. The estimated regression coefficient matrix is given by $\widehat{\mathbf{C}} = \sum_{k=1}^{\tilde{r}} \widehat{\mathbf{C}}_k$, where $\tilde{r}$ is the optimal rank tuned

by information criterion (9) and $\widehat{\mathbf{C}}_k = \widehat{\mathbf{u}}_k \widehat{\mathbf{v}}_k^T$. The following theorem bounds the estimation errors of SEED with the estimated left singular vectors $\widehat{\mathbf{u}}_k$ taking the correct signs as discussed before.

Theorem 1 (Consistency of Estimation and Prediction) *Suppose that Conditions 1 and 2 hold, γ is finite, and $s\log(pq) = o(n)$, then with probability at least $1-\delta$ for any $\delta \in (0,1)$ and uniformly over $k = 1, \ldots, r^*$, we have*

$$\|\widehat{\mathbf{u}}_k - \mathbf{u}_k^*\|_2 \le C_u \sqrt{\frac{s}{n} \log \frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n} \log \frac{pq}{\delta}}\right),$$
$$\frac{1}{\sqrt{q}} \|\widehat{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 \le C_v \sqrt{\frac{s}{n} \log \frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n} \log \frac{pq}{\delta}}\right),$$
$$\frac{\|\mathbf{X}(\widehat{\mathbf{u}}_k - \mathbf{u}_k^*)\|_2}{\sqrt{n}} \le \frac{C_u}{\sqrt{\phi_s}} \sqrt{\frac{s}{n} \log \frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n} \log \frac{pq}{\delta}}\right),$$
$$\frac{1}{\sqrt{q}} \|\widehat{\mathbf{C}}_k - \mathbf{C}_k^*\|_F \le (VC_u + C_v)\left(\sqrt{\frac{s}{n} \log \frac{pq}{\delta}}\right) + o\left(\sqrt{\frac{s}{n} \log \frac{pq}{\delta}}\right),$$
$$\frac{\|\mathbf{X}(\widehat{\mathbf{C}}_k - \mathbf{C}_k^*)\|_F}{\sqrt{nq}} \le \frac{(VC_u + C_v)}{\sqrt{\phi_s}}\left(\sqrt{\frac{s}{n} \log \frac{pq}{\delta}}\right) + o\left(\sqrt{\frac{s}{n} \log \frac{pq}{\delta}}\right),$$

where the constants $C_u = \frac{4\gamma P \sigma_1}{d_\sigma \phi_s^{5/2}}$ and $C_v = 2\sqrt{2}\phi_s^{-3/2}(2V\phi_s^{-1/2} + \sigma_1)C_u + 2\sqrt{2}\phi_s^{-1}\gamma P$.

Theorem 1 is established based on the perturbation theory of sparse generalized eigenvalue problem (Lemma 6). It shows that the uniform estimation error bounds for both top-r^* singular vectors $\mathbf{u}_k^*$ and $\frac{1}{\sqrt{q}}\mathbf{v}_k^*$, top-$r^*$ latent factors $\frac{1}{\sqrt{n}}\mathbf{X}\mathbf{u}_k^*$ and unit rank matrices $\frac{1}{\sqrt{q}}\mathbf{C}_k^*$, and the uniform prediction error bounds of the top-r^* latent factors are all in the same order of $O\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right)$. The tail probability δ can decay to zero quickly as p and q grow with rates such as $\delta \propto (pq)^{-\alpha}$ for some positive constant $\alpha > 1$. This is due to the fact that when $\delta \propto (pq)^{-\alpha}$, we have $\sqrt{\frac{s}{n}\log\frac{pq}{\delta}} \to 0$ under the assumption that $s\log(pq) = o(n)$. Furthermore, the estimation and prediction accuracy would then be within the rate of $O\left(\sqrt{s\log(pq)/n}\right)$, where the factor $\log(pq)$ reflects the curse of dimensionality as there are pq parameters in total from the regression coefficient matrix $\mathbf{C}^*$.

If the true rank r^* can be correctly identified, it is not difficult to see that the estimation accuracy for $\frac{1}{\sqrt{q}}\mathbf{C}^*$ will be within the rate of $O\{\sqrt{r^* s\log(pq)/n}\}$ (see Corollary 3 below), which coincides with the minimax error bound for estimating the regression coefficient vector in the univariate response setting (Raskutti et al., 2011) with the dimensionality p and sparsity level s replaced by the overall dimensionality pq and the product r^*s, respectively.

Similarly, with the true rank r^*, the normalized prediction error $\|\mathbf{X}(\widehat{\mathbf{C}} - \mathbf{C}^*)\|_F/\sqrt{nq}$ will be within the rate of $O\left\{\sqrt{r^* s\log(pq)/n}\right\}$, which is similar to the prediction error bound established in Bunea et al. (2012). But we have an extra scaling factor of $\sqrt{q}$ since the singular values of $\mathbf{X}\mathbf{C}^*$ are allowed to diverge at the rate of $\sqrt{nq}$, larger than the rate $\sqrt{n} + \sqrt{q}$ in Bunea et al. (2012). Our theoretical results show that when the left singular vectors are consistently estimated, the normalized prediction error will converge to

zero asymptotically. Overall speaking, the prediction error bound in Bunea et al. (2012) was derived from the regularization framework, while the prediction accuracy of SEED is obtained from the matrix perturbation theory (Lemmas 5 and 6).

Based on the discussion before, a desirable statistical property of any low-rank estimation procedure is accurately recovering the true rank of the parameter matrix. Similar to Lasso, the nuclear norm regularization needs to be enhanced by techniques such as adaptive regularization to accurately recover the rank of the matrix (Bach, 2008; Chen et al., 2013). In contrast, we can directly control the rank of the solution in SEED by limiting the number of steps. In particular, we propose a GIC-type (Fan and Tang, 2013) information criterion that guarantees rank recovery by SEED when the optimal rank is tuned according to it.

Theorem 2 (Consistency of Rank Recovery) *Suppose Conditions 1-3 hold,* $s\log(pq) = o(n)$, $r^* = o\left(\frac{1}{\sqrt{(\log\log n)\vee\sqrt{s}}}\cdot\left[\frac{n}{\log(pq)}\right]^{\frac{1}{4}}\right)$, *and* $r = o\left(\left[\frac{n}{s\log(pq)}\right]^{\frac{1}{4}}\right)$. *Define the following information criterion*

$$\mathcal{C}_n = \sqrt{n}\log\mathcal{L}_n(\mathbf{Y},\mathbf{X},\widehat{\mathbf{C}}) + \mathrm{rank}(\widehat{\mathbf{C}})\sqrt{\log(pq)}\log\log n, \tag{9}$$

where $\mathcal{L}_n(\mathbf{Y},\mathbf{X},\mathbf{C}) = \frac{1}{qn}\|\mathbf{Y}-\mathbf{X}\mathbf{C}\|_F^2$. *Under the above information criterion, with probability at least* $1-(pq)^{-\alpha}$ *for some positive constant* $\alpha > 1$ *and sufficiently large* n, *SEED will select the true rank, that is,* $\mathrm{rank}(\widehat{\mathbf{C}}) = \mathrm{rank}(\mathbf{C}^*)$.

In the high-dimensional setting where the number of predictors can increase exponentially with the sample size, it is demonstrated in Fan and Tang (2013) that we need some power of the logarithmic factor of dimensionality ($\sqrt{\log(pq)}$ for our setting) in the model complexity penalty of the information criterion to consistently identify the true model, and the slow diverging rate $\log\log n$ is set to prevent underfitting. The proof of Theorem 2 indeed shows that information criterion (9) will keep decreasing until the estimated rank reaches the true rank r^*, where in each step the amount of decrease in the objective function $\mathcal{L}_n(\mathbf{Y},\mathbf{X},\widehat{\mathbf{C}})$ equals to the squared singular value obtained by solving the generalized eigenvalue problem (5). After reaching the true rank, the estimated singular value becomes small such that the model complexity penalty will overweight the decrease and then information criterion (9) would start increasing. Therefore, in the sequence of solutions generated by SEED, the estimate $\widehat{\mathbf{C}}$ with rank r^* will be the minimizer of (9) such that the true rank can be correctly identified.

As discussed before, correct identification of the true rank will yield the estimation accuracy of $\mathbf{C}^*$ as well as the prediction accuracy of $\mathbf{X}\mathbf{C}^*$. Therefore, combined with Theorem 2, it is immediate that the results in Theorem 1 give the following corollary.

Corollary 3 (Consistency of Overall Estimation and Prediction) *Given Conditions 1-3,* $s\log(pq) = o(n)$, $r^* = o\left(\frac{1}{\sqrt{(\log\log n)\vee\sqrt{s}}}\cdot\left[\frac{n}{\log(pq)}\right]^{\frac{1}{4}}\right)$, *and* $r = o\left(\left[\frac{n}{s\log(pq)}\right]^{\frac{1}{4}}\right)$, *if the optimal rank is tuned by information criterion (9), then with probability at least* $1-(pq)^{-\alpha}$

for any constant $\alpha > 0$ *and sufficiently large* n, *we have*

$$\frac{1}{\sqrt{q}}\|\widehat{\mathbf{C}}-\mathbf{C}^*\|_F \le (1+\alpha)(VC_u+C_v)\left(\sqrt{\frac{r^*s\log(pq)}{n}}\right) + o\left(\sqrt{\frac{r^*s\log(pq)}{n}}\right),$$
$$\frac{\|\mathbf{X}(\widehat{\mathbf{C}}-\mathbf{C}^*)\|_F}{\sqrt{qn}} \le \frac{(1+\alpha)(VC_u+C_v)}{\sqrt{\phi_s}}\left(\sqrt{\frac{r^*s\log(pq)}{n}}\right) + o\left(\sqrt{\frac{r^*s\log(pq)}{n}}\right).$$

Besides estimation consistency and rank recovery, SEED is also able to find the true support of the singular vectors. To achieve this goal, after selecting the optimal rank, we need to further refine the model selection procedure by performing a hard-thresholding. See, for example, Fan and Lv (2013) for applications of thresholding in high-dimensional sparse modeling. Specifically, denote by $T_\theta(\mathbf{z})$ the estimator after the hard-thresholding operation on every element of $\mathbf{z} = (z_1,\cdots,z_p)\in\mathbb{R}^p$ such that

$$T_\theta(z_i) = \begin{cases} 0 & \text{if } |z_i| < \theta \\ z_i & \text{otherwise} \end{cases}, \quad i = 1,\ldots,p.$$

Based on the results of Theorems 1 and 2 and the signal strength assumption in Condition 4, we have the following properties for the estimator with a further thresholding.

Theorem 4 (Support Recovery of Singular Vectors) *Given Conditions 1-4,* $s\log(pq) = o(n)$, $r^* = o\left(\frac{1}{\sqrt{(\log\log n)\vee\sqrt{s}}}\cdot\left[\frac{n}{\log(pq)}\right]^{\frac{1}{4}}\right)$, *and* $r = o\left(\left[\frac{n}{s\log(pq)}\right]^{\frac{1}{4}}\right)$, *for every pair of singular vectors,* $(\widehat{\mathbf{u}}_k,\widehat{\mathbf{v}}_k)$, $k = 1,\ldots,r^*$, *the following results hold.*

a) If the threshold $\theta\in(\frac{5}{4}T_u,\frac{7}{4}T_u)$ *with* $T_u = C_u\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}$, *then with probability at least* $1-\delta$, *we have* $\mathrm{supp}(T_\theta(\widehat{\mathbf{u}}_k)) = \mathrm{supp}(\mathbf{u}_k^*)$;

b) If the threshold $\theta\in(\frac{5}{4}T_v,\frac{7}{4}T_v)$ *with* $T_v = C_v\sqrt{\frac{qs}{n}\log\frac{pq}{\delta}}$, *then with probability at least* $1-\delta$, *we have* $\mathrm{supp}(T_\theta(\widehat{\mathbf{v}}_k)) = \mathrm{supp}(\mathbf{v}_k^*)$.

Theorem 4 shows that both supports of the left and right singular vectors can be accurately recovered with properly chosen tuning parameter θ. Together with the correctly identified true rank r^*, the above results indeed yield consistent selection of both predictors and responses. In practice, this threshold θ can be tuned by criteria such as cross-validation.

Besides the statistical properties established before, the proposed procedure SEED enjoys great flexibility in the sense that it does not rely on exact eigenvalue decomposition and the perturbation errors in the generalized eigenvalue problem (5) will be linearly incorporated into the estimated singular vectors $\widehat{\mathbf{u}}_k$ and $\widehat{\mathbf{v}}_k$. Furthermore, our analysis does not reply on the positive definiteness of the Gram matrix (Chen et al., 2012) in high dimensions.

5. Numerical Studies

In this section, we conduct experiments on three data sets, including two simulation data sets (one for a medium-scale experiment and one for a large-scale experiment) and one application data set in social media analysis, to examine the empirical performance of SEED.

5.1. Simulation Studies

5.1.1. Simulation Example 1

We generate a medium-scale synthetic data set as follows: the predictors $\mathbf{x}$ are drawn from a multivariate Gaussian distribution as $\mathbf{x} \sim \mathcal{N}(\mathbf{0}_{p\times 1}, \mathbf{\Sigma}_X)$, where $\mathbf{\Sigma}_X$ is the $p \times p$ covariance matrix with auto-regressive structure, that is, $\Sigma_{X,i,j} = \rho^{|i-j|}$ for some $0 < \rho < 1$ which will be specified later. The responses $\mathbf{y}$ are drawn according to conditional distribution $\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{C}^\top \mathbf{x}, \gamma \mathbf{\Sigma}_E)$ where the noise covariance matrix $\mathbf{\Sigma}_E$ is also selected to have the autoregressive structure with $\rho = 0.5$ and we set $\gamma = 0.3$. We generate the parameter matrix $\mathbf{C}$ as follows: first we generate a block-sparse matrix $\widetilde{\mathbf{C}}$ with 5% non-zero elements. Each non-zero element of $\widetilde{\mathbf{C}}$ is drawn from a $\mathcal{N}(0, 1)$. To achieve a low-rank structure, we find the top-r singular value decomposition of $\widetilde{\mathbf{C}}$ as $\widetilde{\mathbf{C}} = \mathbf{USV}^T$, and then set the elements of $\mathbf{U}$ and $\mathbf{V}$ whose magnitude is smaller than 0.01 to zero to obtain $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$. The final parameter matrix is obtained as $\mathbf{C} = \bar{\mathbf{U}}\bar{\mathbf{S}}\bar{\mathbf{V}}^T$ where the first r diagonal elements of the diagonal matrix $\bar{\mathbf{S}}$ are set to $100, 99, \ldots, 101 - r$. Without loss of generality, we add a few vectors to the design matrix to ensure the orthogonality condition of Section 4.1. In all of the simulation experiments, we generate 100 data sets and report the mean and standard error of performance for different methods.

We compare the performance of SEED with two state-of-art methods: (1) RCGL (Bunea et al., 2012) and (2) Penalized regression with simultaneous L_1 and nuclear-norm penalization. The optimization problem is solved by the popular alternating direction method of multipliers (Boyd et al., 2010) and we will refer to this baseline as the "ADMM" algorithm. For a fair comparison, all model parameters are set based on a separate validation set with size $n_{\text{valid}} = 500$. To tune the parameters in SEED, we created a grid of sparsity thresholds θ and for each value of θ, the validation errors were recorded while increasing the rank of the solution matrices. The robustness of sparsity threshold θ and termination parameter μ will also be analyzed.

The quality of the estimator $\widehat{\mathbf{C}}$ is evaluated via four performance metrics listed as follows. (1) *Normalized Prediction Error* defined as

$$\text{Normalized Prediction Error} = \frac{\|\mathbf{Y}_{\text{test}} - \mathbf{X}_{\text{test}}\widehat{\mathbf{C}}\|_F}{\|\mathbf{Y}_{\text{test}}\|_F}.$$

(2) *Normalized Parameter Estimation Error* defined as

$$\text{Normalized Parameter Estimation Error} = \frac{\|\widehat{\mathbf{C}} - \mathbf{C}\|_F}{\|\mathbf{C}\|_F},$$

where $\mathbf{C}$ is the true parameter matrix.

(3) *Rank Recovery Error* defined as

$$\text{Rank Recovery Error} = |\text{rank}(\widehat{\mathbf{C}}) - \text{rank}(\mathbf{C})|.$$

Since the solution of the nuclear norm always leads to small non-zero singular values (which prevents $\widehat{\mathbf{C}}$ from being low-rank), we threshold the singular values of $\widehat{\mathbf{C}}$ that are more than 100 times smaller than its largest singular value to have a fair comparison.

(4) *Support Recovery AUC*, that is, the area under the receiver operating characteristic (ROC) curve of comparing support of $\widehat{\mathbf{C}}$ with the ground truth, which is always between 0 and 1. It is computed by varying the decision threshold and obtaining the false positive and true positive curve. Then the area under the false positive and true positive curve is reported as AUC. The value of AUC indicates the probability that a procedure assigns a higher value to a randomly chosen non-zero element than a randomly chosen zero element (Hanley and McNeil, 1982). It is an appropriate metric for measuring support recovery accuracy because in sparse support recovery we have more zeros than non-zeros which inflates the result of the simple 0-1 accuracy measure. In contrast, AUC is more robust to imbalanced positive/negative prediction labels.

Table 1 shows the results of all algorithms on a variety of regimes by varying the dimensionality p and the rank r. We can see that SEED is superior or comparable to the baseline algorithms across all four measures. As the results show and the theory predicts, in most high-dimensional cases, nuclear norm usually overestimates the true rank of the matrix. Furthermore, we find that the iterative SVD procedure in the RCGL algorithm often results in significant underestimation of the true rank, when the true rank is large. Note that in addition to accuracy, SEED also significantly reduces the variance of the estimation.

Figure 1a shows the solution path for SEED on one example data set ($p = 400$, $r = 5$, $q = 200$, $\rho = 0.5$, and $n = 100$). The corresponding singular values are set to $30, 27, 24, 21$, and 18. In the horizontal axis, we show the termination parameter μ normalized by $\|\mathbf{Y}\|_F^2/(nq)$. We can see that SEED can identify the correct rank with medium values of μ. Figure 1b shows the solution path for the top left singular vector $\mathbf{u}_1$ of $\mathbf{C}$ on an example data set ($p = 200$, $q = 100$, $n = 50$, $\rho = 0.5$, and $r = 1$). Both of the solution paths indicate that SEED is robust to the particular choice of parameters and in a large range of parameters SEED is able to successfully recover the true rank of the matrix and the support of the singular vectors.

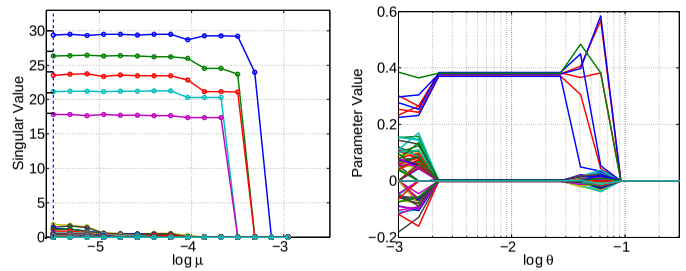
5.1.2. Simulation Example 2

In order to study scalability of SEED, we conduct the experiments on two computing environment, including: (1) an off-the-shelf personal computer (PC) and (2) a graphics processing unit (GPU), to demonstrate the runtime efficiency and the parallelization capability of SEED.

First, we run our experiments on an off-the-shelf PC with Intel i7 at 3.4GHz and 8GB of memory. The system runs MATLAB R2013b on the Windows operating system. We generate 5 data sets with $r = 1$, non-zero ratio of 10%, $q = 1000$, and $n = 1000$. Figure 2a shows the average CPU runtime of three algorithms as the dimension p increases. We can see that SEED can achieve a speed up of 10-100 times in runtime compared with baseline methods.

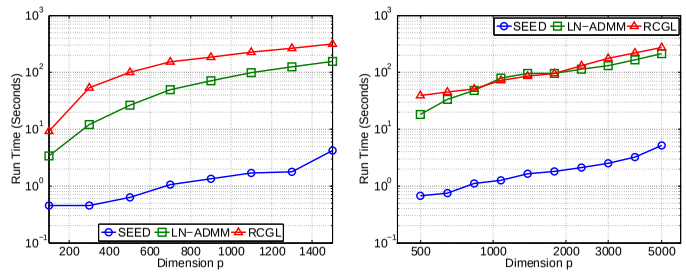
p	Algorithm	Normalized Prediction Error	Normalized Estimation Error	Rank Recovery Error	Support Recovery AUC
$n = 100$, $q = 200$, $\mathbf{r} = \mathbf{3}$, and $\rho = 0.5$					
100	SEED	**0.039 (0.013)**	**0.012 (0.006)**	**0.06 (0.445)**	**0.980 (0.024)**
	ADMM	0.041 (0.012)	0.015 (0.004)	0.01 (0.100)	0.975 (0.026)
	RCGL	0.053 (0.013)	0.040 (0.009)	0.03 (0.171)	0.979 (0.028)
400	SEED	**0.026 (0.005)**	**0.011 (0.002)**	**0 (0)**	**0.970 (0.015)**
	ADMM	0.035 (0.007)	0.035 (0.008)	0.06 (0.278)	0.961 (0.018)
	RCGL	0.158 (0.050)	0.193 (0.066)	0 (0)	0.934 (0.027)
800	SEED	**0.019 (0.003)**	**0.010 (0.002)**	**0 (0)**	**0.970 (0.014)**
	ADMM	0.076 (0.020)	0.127 (0.031)	4.14 (1.614)	0.954 (0.020)
	RCGL	0.482 (0.072)	0.603 (0.074)	0.01 (0.100)	0.834 (0.030)
1500	SEED	**0.015 (0.002)**	**0.010 (0.002)**	**0 (0)**	**0.971 (0.011)**
	ADMM	0.072 (0.015)	0.145 (0.024)	3.02 (0.141)	0.968 (0.010)
	RCGL	0.698 (0.054)	0.820 (0.022)	0.09 (0.379)	0.809 (0.047)
2000	SEED	**0.014 (0.001)**	**0.010 (0.002)**	**0 (0)**	**0.973 (0.011)**
	ADMM	0.066 (0.010)	0.136 (0.016)	3 (0)	0.970 (0.011)
	RCGL	0.746 (0.041)	0.857 (0.021)	0.2 (0.603)	0.789 (0.045)
$n = 100$, $q = 200$, $\mathbf{r} = \mathbf{30}$, and $\rho = 0.5$					
100	SEED	**0.010 (0.001)**	**0.043 (0.003)**	**0.29 (0.456)**	**0.989 (0.003)**
	ADMM	0.015 (0.002)	0.066 (0.006)	48.51 (0.882)	0.964 (0.027)
	RCGL	0.058 (0.010)	0.220 (0.017)	0.04 (0.243)	0.774 (0.064)
400	SEED	**0.035 (0.018)**	**0.155 (0.082)**	**1.17 (1.092)**	**0.770 (0.023)**
	ADMM	0.037 (0.002)	0.139 (0.007)	42.34 (3.085)	0.783 (0.018)
	RCGL	0.395 (0.029)	0.558 (0.023)	0.02 (0.200)	0.610 (0.010)
800	SEED	**0.003 (0.000)**	**0.005 (0.000)**	**0.02 (0.141)**	**0.999 (0.000)**
	ADMM	0.014 (0.003)	0.022 (0.004)	9.34 (2.388)	0.999 (0.001)
	RCGL	0.286 (0.037)	0.429 (0.020)	0.13 (0.733)	0.966 (0.004)
1500	SEED	**0.003 (0.000)**	**0.007 (0.000)**	**0 (0)**	**0.999 (0.000)**
	ADMM	0.021 (0.001)	0.088 (0.004)	23.61 (1.348)	0.999 (0.000)
	RCGL	0.315 (0.024)	0.457 (0.017)	2.31 (3.243)	0.970 (0.002)
2000	SEED	**0.002 (0.000)**	**0.007 (0.000)**	**0 (0)**	**0.999 (0.000)**
	ADMM	0.020 (0.001)	0.091 (0.003)	22.54 (1.275)	0.999 (0.000)
	RCGL	0.335 (0.019)	0.472 (0.012)	3.10 (3.538)	0.974 (0.002)

Table 1: Simulation Results



(a) Rank (b) Support

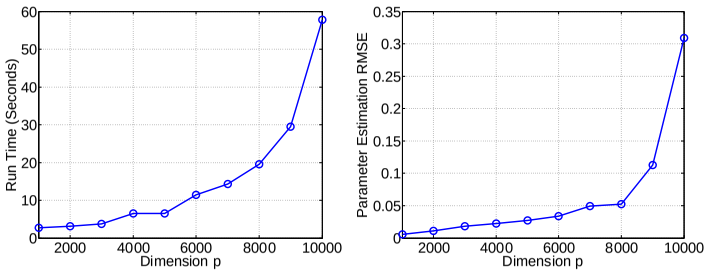
Figure 1: (a) Solution path for the singular values of the estimated matrices. The plot show the value of top five singular values of the solution $\widehat{\mathbf{C}}$ as we change the stopping error μ. (b) Solution path for the top left singular vector $\mathbf{u}$ of the estimated matrices. Only seven coefficients are non-zero. The range of the parameters are generated as follows: $\mu = \text{logspace}(-5, -1, 5)$ and $\theta = \text{logspace}(-1, \log_{10}(20), 10)$, where $\text{logspace}(a, b, n)$ indicates the minimum value 10^a, maximum value 10^b, and total number n.



(a) CPU (b) GPU

Figure 2: Speedup by SEED on (a) CPU and (b) GPU devices. Note that the vertical axis is in logarithmic scale.

Next, in order to test scalability of SEED in extremely large data sets, we use a machine that is equipped with a Tesla K40 GPU which has 2880 processing cores at 745MHz and 12GB of memory. We perform our experiments with MATLAB R2013b on a Debian Linux operating system. GPUs are built to have many less-powerful processing units which makes them ideal for parallel implementation (Bekkerman et al., 2012, Chapter 5). Therefore we apply the two-step fast eigenvalue decomposition described in Section 3, which involves only simple matrix operation and can be paralleled easily. The experiment results shown in



(a) (b)

Figure 3: Scalability experiments on very large data sets on GPU using the fast approach. Accuracy results are normalized.

Figures 2b and 3 are obtained under the setting of $q = 10000$, $n = 5000$, $r = 1$ and non-zero ratio of 10%. The results indicate that while SEED is fast on the GPU, it also achieves reasonable accuracy. Note that the results show that SEED is able to estimate a sparse and low-rank matrix with 10^8 elements in less than a minute, confirming its extreme scalability.

5.2. Real Data Analysis

Diffusion Network Inference, that is, the task of inferring influence networks from user activities, is one of the central tasks in social networks analysis (Gomez-Rodriguez et al., 2012) because it helps improve social marketing by finding the influential users in a network. It is a challenging problem because: (i) in many social networks the influence is expressed implicitly (Gomez-Rodriguez et al., 2012) and (ii) empirical studies show that common metrics such as number of friends or followers fail to accurately measure the social influence of the users (Cha et al., 2010).

A popular computational approach in estimating social influence among users is to count the number of users' activities over a time span (in regularly or irregularly spaced intervals) and analyze the resulting time series data (Truccolo et al., 2005). Many different models have been developed, among which the vector auto-regressive model arises as a simple and robust solution (Trusov et al., 2009; Bahadori and Liu, 2013). That is, every user i is described by a time series $x_i(t)$ for $t = 1, \ldots, T$ and $i = 1, \ldots, q$, such that

$$x_i(t) = \sum_{j=1}^{q} \boldsymbol{\beta}_{i,j}^T \mathbf{x}_j^{t,Lagged} + \varepsilon_i(t),$$

where $\boldsymbol{\beta}_{i,j}$ is the vector of coefficients modeling the effects of time series x_j, $\mathbf{x}_j^{t,Lagged} = [x_j(t-L), \ldots, x_j(t-1)]^T$ is the history of x_j up to time t with L the maximal time lag, and $\varepsilon_i(t)$ is the random noise at time t. Denoting by $\mathbf{x}(t) = [x_1(t), \ldots, x_q(t)]^T$, we have the following multi-response regression model:

$$\mathbf{x}(t) = \mathbf{B}^T \mathbf{x}^{t,Lagged} + \boldsymbol{\varepsilon}(t),$$

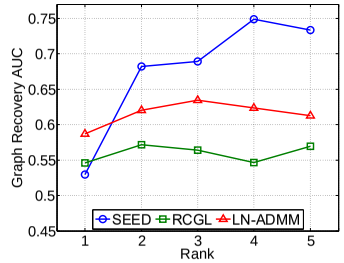


Figure 4: The graph recovery accuracy of the algorithms as the rank of solution varies.

where the predictor vector $\mathbf{x}^{t,Lagged} = [(\mathbf{x}_1^{t,Lagged})^T, \ldots, (\mathbf{x}_q^{t,Lagged})^T]^T$, $\mathbf{B}^T = (\boldsymbol{\beta}_{i,j}^T)_{1 \le i,j \le q}$ is the evolution matrix, and $\boldsymbol{\varepsilon}(t) = [\varepsilon_1(t), \ldots, \varepsilon_q(t)]^T$. The influence network can be built from the evolution matrix by establishing an edge from node j to node i if $\|\boldsymbol{\beta}_{i,j}\|_1$ is significantly larger than zero.

In this experiment, we gather a Twitter data set with tweets on the "Haiti earthquake" and apply vector auto-regressive model to identify the potential top influencers on this topic (that is, those Twitter accounts with the largest impact on the others). We divide the 17 days after the Haiti Earthquake on Jan. 12, 2010 into 1000 uniformly spaced intervals and generate a multivariate time series data set by counting the number of tweets on this topic for the top 1000 users who tweeted most about it. For accurate modeling, we remove the users that were highly correlated with each other, most of which were operated by the same users and tweeted exactly the same content. We also remove robot-like user-accounts which tweeted on very regular intervals, which in total led to a subset of 270 users. We analyze this data by a VAR model with the maximal time lag $L = 5$ based on the intuition about the maximum retweeting delay, which requires estimation of a $q = 270$ dimensional response vector using $p = 1350$ predictors while we have only $n = 995$ observations.

Since we do not have access to the true influence network, we use the retweet network as a surrogate of the ground truth following the evaluation convention in the social networks community. The retweet network is constructed by adding an edge from user i to user j if user j has retweeted at least 4 of the tweets of user i. Clearly, the retweet network is not the actual underlying temporal dependency graph, mainly because there are possible implicit influence patterns as well. However, it is the best possible metric that we could obtain for graph estimation accuracy evaluation in our data set (Cha et al., 2010). The retweet network for the 270 selected users is sparse; it has only 0.11% of possible edges.

We apply SEED, ADMM, and RCGL algorithms to uncover the influence network in our twitter data set. Figure 4 shows the accuracy of the procedures in uncovering the true influence network in terms of AUC. For every value of the rank parameter, we tune the sparsity by 5-fold cross-validation. Given the fact that exact rank constraint cannot be enforced directly in the ADMM algorithm, we find the best value of the nuclear norm regularization parameter λ_L by 5-fold cross-validation. Then, we compute the low-rank approximations of the parameter matrix and evaluate the accuracy at each rank.

SEED	ADMM	RCGL
2.83	127.34	256.87

Table 2: Run time (in seconds) of the algorithms on the application data set.

The results in Figure 4 show that SEED significantly outperforms the baseline procedures. They also indicate that, in all of the algorithms, as we increase the rank of the solution matrix, the accuracy is improved initially and then quickly saturates. SEED achieves the highest accuracy when the rank is 4. Note that this result also confirms other studies that the social network connections may be strongly influenced by a few unobserved exogenous variables (Myers et al., 2012). The results in Table 2 demonstrate the significant speedup achieved by SEED compared to the baselines.

6. Discussion

In this paper, we propose to convert the problem of sparse reduced-rank regression into a sparse generalized eigenvalue problem, which allows us to efficiently employ the recently developed sparse eigenvalue decomposition techniques. After this transformation, the left singular vectors can be estimated in simple steps and the estimation of both sparse and dense right singular vectors is unified in a single framework. As a pure learning algorithm, SEED deviates from traditional regularization frameworks (that is, a loss function plus certain penalties), leading to computational efficiency and scalability. Furthermore, we prove that SEED achieves nice estimation and prediction accuracy similar to the minimax error bound in the univariate regression setting (Raskutti et al., 2011).

Some interesting problems for future research include extending the current formulation of the regression coefficient matrix in (2) to the case where the singular values can be repeated such that the left singular vectors (which correspond to latent factors) are not identifiable. Then we will need to estimate the eigenspaces spanned by important singular vectors and characterize the estimation accuracy by some new criterion, such as the one in Cai et al. (2013) and Ma (2013). Another research direction is to explore the theory of random design matrices and this can be addressed by using an extended version of perturbation theory (Lemma 6), where the perturbation in $\mathbf{P}$ is also included in the analysis.

Moreover, it is computationally straightforward to extend SEED to the generalized linear models by adapting the sequential quadratic programming framework. For this extension, we first approximate the loss function by the quadratic loss function and find the optimal unit rank matrix. Then we can add the unit rank matrix to the solution and re-approximate the loss function with another quadratic function around this new solution. By performing these three steps sequentially, we can efficiently estimate the low-rank coefficient matrix. Statistical properties of such estimator can be analyzed by extending the results in Lozano et al. (2011) for greedy sparse procedures to reduced-rank regression.

Acknowledgments

We would like to acknowledge support for this project from NSF CAREER Awards DMS-0955316 and IIS-1254206, NSFC-11601501, 11671374 and 71731010, Anhui Provincial Natural Science Foundation-1708085QA02, Fundamental Research Funds for Central Universities-WK2040160028, Okawa Foundation Research Award, a grant from the Simons Foundation, and Adobe Data Science Research Award. The authors would like to thank Emre Demirkaya and Sanjay Purushotham for helpful discussions. The authors also sincerely thank the Action Editor and referees for their valuable comments that helped improve the article substantially.

Appendix A. Proofs of Theorems 1 and 2

We need the following two lemmas in the proofs of Theorems 1 and 2.

Lemma 5 *Suppose $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_q, \boldsymbol{\Sigma})$ and $\gamma^2 = \max_j \boldsymbol{\Sigma}_{jj}$. We have stacked n realization of these random vectors in the rows of $n \times q$ matrix $\mathbf{E}$. Denote by $P^2 = \max_j [\frac{1}{n}\mathbf{X}^T\mathbf{X}]_{jj}$ given a deterministic $n \times p$ matrix $\mathbf{X}$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\frac{1}{n}\|\mathbf{E}^T\mathbf{X}\|_{2,s} \leq \sqrt{\frac{2qs(\gamma P)^2}{n}\log\frac{2pq}{\delta}}.$$

Lemma 6 *(Perturbation Theory for Sparse Symmetric Generalized Eigenvalue Problems) Suppose $\mathbf{Q}, \mathbf{P} \in \mathbb{S}_p^+$ are two $p \times p$ semi-definite matrices. For the following generalized eigenvalue problem and its perturbed variant with sparse eigenvectors*

$$\mathbf{Q}\mathbf{u} = \lambda\mathbf{P}\mathbf{u}, \tag{10}$$

$$(\mathbf{Q} + \boldsymbol{\delta}_Q)\widehat{\mathbf{u}} = \widehat{\lambda}\mathbf{P}\widehat{\mathbf{u}}, \tag{11}$$

where $\mathbf{u}, \widehat{\mathbf{u}} \perp \mathrm{Ker}(\mathbf{P})$ with $\|\mathbf{u}\|_0 \leq s^*$, $\|\widehat{\mathbf{u}}\|_0 \leq s$ and $s^* < s$, under Condition 1, we have uniformly over $k = 1, \cdots, p - \dim\{\mathrm{Ker}(\mathbf{P})\}$,

$$|\widehat{\lambda}_k - \lambda_k| \leq \phi_s^{-1}\|\boldsymbol{\delta}_Q\|_{2,s}. \tag{12}$$

Here λ_k and $\widehat{\lambda}_k$ are the kth largest eigenvalues of equations (10) and (11), respectively, $\|\boldsymbol{\delta}_Q\|_{2,s}$ denotes the s-sparse largest singular value of $\boldsymbol{\delta}_Q$, and ϕ_s is defined in Condition 1.

Furthermore, denote by $\mathbf{u}_k$ and $\widehat{\mathbf{u}}_k$ the unit length sparse eigenvectors correspond to λ_k and $\widehat{\lambda}_k$, respectively, with $\widehat{\mathbf{u}}_k$ taking the correct directions. When there exists some positive eigengap d_λ which is the minimum difference between non-zero eigenvalues of equation (10) and the perturbation of $\mathbf{Q}$ satisfies $\|\boldsymbol{\delta}_Q\|_{2,s} = o(d_\lambda)$, then uniformly over k such that $\lambda_k \neq 0$,

$$\|\widehat{\mathbf{u}}_k - \mathbf{u}_k\|_2 \leq \sqrt{2}\phi_s^{-2}d_\lambda^{-1}\|\boldsymbol{\delta}_Q\|_{2,s} + o(d_\lambda^{-1}\|\boldsymbol{\delta}_Q\|_{2,s}). \tag{13}$$

A.1. Proof of Theorem 1

The proof is established based on analyzing the impact of perturbation of the matrices in the solution of equation (3). To do so, in the first step, we bound the amount of perturbation in the matrix of the generalized eigenvalue problem in Proposition 1. In the second step, using

the eigenvalue perturbation theory (Lemma 6), we derive the error bound for $\widehat{\mathbf{u}}_k$. Using the bound for $\widehat{\mathbf{u}}_k$, in the third step, the error bound for $\widehat{\mathbf{v}}_k$ is calculated.

Step 1: Bounding the perturbation of matrix $\mathbf{Q}$. For notational simplicity, define the noise free response variables as $\mathbf{Y}^* = \mathbf{X}\mathbf{C}^*$, $\mathbf{Q} = \frac{1}{qn^2}\mathbf{X}^T\mathbf{Y}^*\mathbf{Y}^{*T}\mathbf{X}$ and its noisy version $\widehat{\mathbf{Q}} = \frac{1}{qn^2}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$. Denote by $\|\widehat{\mathbf{Q}} - \mathbf{Q}\|_{2,s}$ the s-sparse largest singular value of $\widehat{\mathbf{Q}} - \mathbf{Q}$. Since $\mathbf{Y} = \mathbf{Y}^* + \mathbf{E}$, we can derive the following bound,

$$\|\widehat{\mathbf{Q}} - \mathbf{Q}\|_{2,s} = \frac{1}{qn^2}\left\|\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X} - \mathbf{X}^T\mathbf{Y}^*\mathbf{Y}^{*T}\mathbf{X}\right\|_{2,s} \le \frac{2}{n^2q}\left\|\mathbf{X}^T\mathbf{Y}^*\mathbf{E}^T\mathbf{X}\right\|_{2,s} + \frac{1}{n^2q}\|\mathbf{E}^T\mathbf{X}\|_{2,s}^2,$$

where the last inequality is due to the expansion of $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$ and an application of the triangular inequality.

By Condition 1 and Proposition 1, we have $\|\mathbf{X}\|_{2,s} \le \sqrt{n}\phi_s^{-1/2}$ and

$$\|\mathbf{X}^T\mathbf{Y}^*\|_2 \le \|\mathbf{X}\|_{2,s}\cdot\|\mathbf{Y}^*\|_2 = \|\mathbf{X}\|_{2,s}\cdot\sqrt{nq}\sigma_1 \le n\sqrt{q}\sigma_1\phi_s^{-1/2}, \tag{14}$$

where the first inequality holds since the unit length vector $\mathbf{v}$ satisfying $\|\mathbf{X}^T\mathbf{Y}^*\mathbf{v}\|_2 = \|\mathbf{X}^T\mathbf{Y}^*\|_2$ must be one of the $\mathbf{v}_k^*$ in (2) due to the strict separation of the singular values by Condition 2, so that $\mathbf{X}^T\mathbf{Y}^*\mathbf{v}_k^* = \mathbf{X}^T\mathbf{X}\mathbf{C}^*\mathbf{v}_k^*$ with $\mathbf{C}^*\mathbf{v}_k^*$ yielding an s-sparse vector for $\mathbf{X}^T\mathbf{X}$. Therefore, we get

$$\frac{2}{n^2q}\left\|\mathbf{X}^T\mathbf{Y}^*\mathbf{E}^T\mathbf{X}\right\|_{2,s} \le \frac{2}{n^2q}\left\|\mathbf{X}^T\mathbf{Y}^*\right\|_2\cdot\left\|\mathbf{E}^T\mathbf{X}\right\|_{2,s} \le \frac{2\sigma_1\phi_s^{-1/2}}{n\sqrt{q}}\left\|\mathbf{E}^T\mathbf{X}\right\|_{2,s}.$$

Let $a^* = \sigma_1\phi_s^{-1/2}$. It follows that

$$\|\widehat{\mathbf{Q}} - \mathbf{Q}\|_{2,s} \le \frac{2a^*}{n\sqrt{q}}\|\mathbf{E}^T\mathbf{X}\|_{2,s} + \frac{1}{n^2q}\|\mathbf{E}^T\mathbf{X}\|_{2,s}^2. \tag{15}$$

Step 2: Error bounds for $\widehat{\mathbf{u}}_k$ *and* $\mathbf{X}\widehat{\mathbf{u}}_k$. Using Lemma 5, for any $\delta \in (0,1)$ with probability at least $1-\delta$, we have

$$\frac{1}{n\sqrt{q}}\|\mathbf{E}^T\mathbf{X}\|_{2,s} \le \gamma P\sqrt{\frac{2s}{n}\log\frac{2pq}{\delta}}.$$

Since $\frac{s}{n}\log\frac{pq}{\delta} \to 0$, substituting the above bound in equation (15) yields

$$\|\widehat{\mathbf{Q}} - \mathbf{Q}\|_{2,s} \le \gamma Pa^*\sqrt{\frac{8s}{n}\log\frac{2pq}{\delta}} + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right). \tag{16}$$

Therefore, under Conditions 1 and 2, applying Lemma 6 with $\widehat{\mathbf{Q}} = \frac{1}{qn^2}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$, $\mathbf{Q} = \frac{1}{qn^2}\mathbf{X}^T\mathbf{Y}^*\mathbf{Y}^{*T}\mathbf{X}$, and $\mathbf{P} = \frac{1}{n}\mathbf{X}^T\mathbf{X}$ gives the following bound for $\widehat{\mathbf{u}}_k$,

$$\|\widehat{\mathbf{u}}_k - \mathbf{u}_k^*\|_2 \le \frac{4\gamma Pa^*}{d_\sigma\phi_s^2}\sqrt{\frac{s}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right) = C_u\sqrt{\frac{s}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right),$$

where $C_u = \frac{4\gamma Pa^*}{d_\sigma\phi_s^2} = \frac{4\gamma P\sigma_1}{d_\sigma\phi_s^{5/2}}$ is a constant. Since $\|\widehat{\mathbf{u}}_k - \mathbf{u}_k\|_0 \le 2s$, further applying Condition 1 yields

$$\frac{1}{\sqrt{n}}\|\mathbf{X}(\widehat{\mathbf{u}}_k - \mathbf{u}_k^*)\|_2 \le \phi_s^{-1/2}C_u\sqrt{\frac{s}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right).$$

Step 3: Error bound for $\widehat{\mathbf{v}}_k$. With the estimated left singular vector $\widehat{\mathbf{u}}_k$, we can further estimate the corresponding right singular vector as

$$\widehat{\mathbf{v}}_k = \frac{1}{\widehat{\mathbf{u}}_k^\top\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}_k}\mathbf{Y}^T\mathbf{X}\widehat{\mathbf{u}}_k,$$

and compare it to $\mathbf{v}_k^*$, which by Proposition 1 can be expressed as

$$\mathbf{v}_k^* = \frac{1}{\mathbf{u}_k^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}_k^*}\mathbf{Y}^{*T}\mathbf{X}\mathbf{u}_k^*.$$

For simplicity of notation, we drop the index k of $\mathbf{v}_k$ and $\mathbf{u}_k$ and write

$$\|\widehat{\mathbf{v}} - \mathbf{v}^*\|_2 = \left\|\frac{1}{\mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^* - e_1}\left(\mathbf{Y}^{*T}\mathbf{X}\mathbf{u}^* - \mathbf{e}_2\right) - \frac{1}{\mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^*}\mathbf{Y}^{*T}\mathbf{X}\mathbf{u}^*\right\|_2,$$

where $e_1 \triangleq \mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^* - \widehat{\mathbf{u}}^\top\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}$ and $\mathbf{e}_2 \triangleq \mathbf{Y}^{*T}\mathbf{X}\mathbf{u}^* - \mathbf{Y}^T\mathbf{X}\widehat{\mathbf{u}}$. Using the Taylor expansion of $\frac{1}{1-x}$ at $x=0$, we can write

$$\begin{aligned}\|\widehat{\mathbf{v}} - \mathbf{v}^*\|_2 &\le \left\|\frac{\mathbf{Y}^{*T}\mathbf{X}\mathbf{u}^*}{\mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^*}\frac{e_1}{\mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^*}\right\|_2 + \left\|\frac{\mathbf{e}_2}{\mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^*}\right\|_2 + T_e \\ &= (\mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^*/n)^{-1}\left(\|\mathbf{v}^*\|_2|e_1|/n + \|\mathbf{e}_2\|_2/n\right) + T_e \\ &\le \phi_s^{-1}(\|\mathbf{v}^*\|_2|e_1|/n + \|\mathbf{e}_2\|_2/n) + T_e,\end{aligned} \tag{17}$$

where $T_e = \left(\frac{\|\mathbf{v}^*\|_2|e_1|/n}{\mathbf{u}^{*\top}\mathbf{P}\mathbf{u}^*} + \frac{\|\mathbf{e}_2\|_2/n}{\mathbf{u}^{*\top}\mathbf{P}\mathbf{u}^*}\right)\sum_{\ell=1}^{\infty}\left(\frac{|e_1|/n}{\mathbf{u}^{*\top}\mathbf{P}\mathbf{u}^*}\right)^\ell$ denotes the higher order terms in the Taylor expansion and the last step is by Condition 1.

Bounding $|e_1|$*:* Let $\widehat{\mathbf{u}} = \mathbf{u}^* + \boldsymbol{\delta}_\mathbf{u}$. Since $\|\mathbf{u}^*\|_0 \le s^*$, $\|\widehat{\mathbf{u}}\|_0 \le s$ and $s^* < s$, we have $\|\boldsymbol{\delta}_\mathbf{u}\|_0 \le 2s$. As $\mathbf{u}^*$ is a unit length vector, it yields from Condition 1 that

$$\begin{aligned}|e_1|/n &= \left|\frac{1}{n}(\mathbf{u}^* + \boldsymbol{\delta}_\mathbf{u})^\top\mathbf{X}^T\mathbf{X}(\mathbf{u}^* + \boldsymbol{\delta}_\mathbf{u}) - \frac{1}{n}\mathbf{u}^{*\top}\mathbf{X}^T\mathbf{X}\mathbf{u}^*\right| \\ &= |2\mathbf{u}^{*\top}\mathbf{P}\boldsymbol{\delta}_\mathbf{u} + \boldsymbol{\delta}_\mathbf{u}^T\mathbf{P}\boldsymbol{\delta}_\mathbf{u}| \le \phi_s^{-1}(2\|\boldsymbol{\delta}_\mathbf{u}\|_2 + \|\boldsymbol{\delta}_\mathbf{u}\|_2^2).\end{aligned} \tag{18}$$

Bounding $\|\mathbf{e}_2\|_2$*:* Similarly, let $\mathbf{Y}^T\mathbf{X} = \mathbf{Y}^{*T}\mathbf{X} + \mathbf{E}^T\mathbf{X}$. We obtain

$$\begin{aligned}\|\mathbf{e}_2\|_2/n &= \|(\mathbf{Y}^{*T}\mathbf{X} + \mathbf{E}^T\mathbf{X})(\mathbf{u}^* + \boldsymbol{\delta}_\mathbf{u}) - \mathbf{Y}^{*T}\mathbf{X}\mathbf{u}^*\|_2/n \\ &\le \|\mathbf{Y}^{*T}\mathbf{X}\boldsymbol{\delta}_\mathbf{u}\|_2/n + \|\mathbf{E}^T\mathbf{X}\mathbf{u}^*\|_2/n + \|\mathbf{E}^T\mathbf{X}\boldsymbol{\delta}_\mathbf{u}\|_2/n \\ &\le a^*\sqrt{q}\|\boldsymbol{\delta}_\mathbf{u}\|_2 + \|\mathbf{E}^T\mathbf{X}\|_{2,s}/n + \|\mathbf{E}^T\mathbf{X}\|_{2,s}\|\boldsymbol{\delta}_\mathbf{u}\|_2/n,\end{aligned} \tag{19}$$

where in the last step we used $\|\mathbf{Y}^{*T}\mathbf{X}\boldsymbol{\delta}_{\mathbf{u}}\|_2/n \le a^*\sqrt{q}\|\boldsymbol{\delta}_{\mathbf{u}}\|_2$ similarly as in (14).

Since $\frac{1}{\sqrt{q}}\|\mathbf{v}_k^*\|_2 \le V$, substituting inequalities (18) and (19) in inequality (17) with reorganization yields

$$\begin{aligned}\|\widehat{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 \le & \frac{V\sqrt{q}\|\boldsymbol{\delta}_{\mathbf{u}}\|_2}{\phi_s^2}(2+\|\boldsymbol{\delta}_{\mathbf{u}}\|_2) + \frac{1}{n\phi_s}\|\mathbf{E}^T\mathbf{X}\|_{2,s}(1+\|\boldsymbol{\delta}_{\mathbf{u}}\|_2) \\ & + \phi_s^{-1}a^*\sqrt{q}\|\boldsymbol{\delta}_{\mathbf{u}}\|_2 + T_e.\end{aligned}$$

When $s\log(pq/\delta) = o(n)$, we can see that both the events $\mathcal{E}_1 = \left\{\|\boldsymbol{\delta}_{\mathbf{u}}\|_2 = O\big(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\big)\right\}$ and $\mathcal{E}_2 = \left\{T_e = O\big(\sqrt{\frac{qs}{n}\log\frac{pq}{\delta}}\big)\right\}$ occur, if the event $\mathcal{E}_0 = \left\{\frac{1}{n\sqrt{q}}\|\mathbf{E}^T\mathbf{X}\|_{2,s} = O\big(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\big)\right\}$ holds for some $\delta \in (0,1)$, where in the second event we have used the results in inequalities (18) and (19) together with an application of geometric series sum. Note that the first term of T_e is $O\big(\sqrt{\frac{qs}{n}\log\frac{pq}{\delta}}\big)$ and the common ratio is $O\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right)$.

Therefore, by applying the result in Lemma 5 for bounding $\|\mathbf{E}^T\mathbf{X}\|_{2,s}/n$ and the result for the estimation error bound of $\|\widehat{\mathbf{u}}_k - \mathbf{u}_k^*\|_2$ in **Step 2**, we can conclude with probability at least $1-\delta$ that

$$\begin{aligned}\|\widehat{\mathbf{v}}_k - \mathbf{v}_k^*\|_2 &\le \frac{(2V\phi_s^{-1}+a^*)C_u + \gamma P}{\phi_s}\sqrt{\frac{2qs}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{qs}{n}\log\frac{pq}{\delta}}\right) + T_e \\ &\le \frac{2\{(2V\phi_s^{-1}+a^*)C_u + \gamma P\}}{\phi_s}\sqrt{\frac{2qs}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{qs}{n}\log\frac{pq}{\delta}}\right) \\ &= C_v\sqrt{\frac{qs}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{qs}{n}\log\frac{pq}{\delta}}\right),\end{aligned}$$

where $C_v = 2\sqrt{2}\phi_s^{-1}\left\{(2V\phi_s^{-1/2}+\sigma_1)\phi_s^{-1/2}C_u + \gamma P\right\}$.

Step 4: Error bounds for $\frac{1}{\sqrt{q}}\|\widehat{\mathbf{C}}_k - \mathbf{C}_k^*\|_F$ *and* $\frac{1}{\sqrt{qn}}\|\mathbf{X}(\widehat{\mathbf{C}}_k - \mathbf{C}_k^*)\|_F$. With the estimation error bounds of $\widehat{\mathbf{u}}_k$ and $\widehat{\mathbf{v}}_k$, we write

$$\frac{1}{qn}\|\mathbf{X}(\mathbf{C}_k^* - \widehat{\mathbf{C}}_k)\|_F^2 = \frac{1}{q}\mathrm{trace}\{(\mathbf{C}_k^* - \widehat{\mathbf{C}}_k)^T\mathbf{P}(\mathbf{C}_k^* - \widehat{\mathbf{C}}_k)\} = \tilde{e}_1 + \tilde{e}_2 + \tilde{e}_3, \tag{20}$$

where the last equality follows from the decomposition

$$\mathbf{C}_k^* - \widehat{\mathbf{C}}_k = \mathbf{u}_k^*\mathbf{v}_k^{*T} - \widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^T = (\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)\mathbf{v}_k^{*T} + \widehat{\mathbf{u}}_k(\mathbf{v}_k^* - \widehat{\mathbf{v}}_k)^T \tag{21}$$

such that

$$\begin{aligned}\tilde{e}_1 &= \frac{1}{q}\mathrm{trace}\{\mathbf{v}_k^*(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)^T\mathbf{P}(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)\mathbf{v}_k^{*T}\}, \\ \tilde{e}_2 &= \frac{2}{q}\mathrm{trace}\{(\mathbf{v}_k^* - \widehat{\mathbf{v}}_k)\widehat{\mathbf{u}}_j^T\mathbf{P}(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)\mathbf{v}_k^{*T}\}, \\ \tilde{e}_3 &= \frac{1}{q}\mathrm{trace}\{(\mathbf{v}_k^* - \widehat{\mathbf{v}}_k)\widehat{\mathbf{u}}_k^T\mathbf{P}\widehat{\mathbf{u}}_k(\mathbf{v}_k^* - \widehat{\mathbf{v}}_k)^T\}.\end{aligned}$$

We will bound them separately. Since the estimation error bounds of $\widehat{\mathbf{u}}_k$ and $\widehat{\mathbf{v}}_k$, $k = 1,\cdots,r^*$, hold with probability at least $1-\delta$. It yields that

$$\begin{aligned}\tilde{e}_1 &= \frac{1}{q}\mathrm{trace}\{(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)^T\mathbf{P}(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)\mathbf{v}_k^{*T}\mathbf{v}_k^*\} = V^2\cdot(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)^T\mathbf{P}(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k) \\ &\le V^2\phi_s^{-1}\|\mathbf{u}_k^* - \widehat{\mathbf{u}}_k\|_2^2 \le V^2\phi_s^{-1}C_u^2\left(\frac{s}{n}\log\frac{pq}{\delta}\right) + o\left(\frac{s}{n}\log\frac{pq}{\delta}\right),\end{aligned}$$

where the first inequality follows from Condition 1 and the fact $\|\mathbf{u}_k^* - \widehat{\mathbf{u}}_k\|_0 \le 2s$. Similarly, we have

$$\begin{aligned}\tilde{e}_2 &\le 2V\phi_s^{-1}C_uC_v\left(\frac{s}{n}\log\frac{pq}{\delta}\right) + o\left(\frac{s}{n}\log\frac{pq}{\delta}\right), \\ \tilde{e}_3 &\le \phi_s^{-1}C_v^2\left(\frac{s}{n}\log\frac{pq}{\delta}\right) + o\left(\frac{s}{n}\log\frac{pq}{\delta}\right).\end{aligned}$$

In view of (20), the above bounds together give

$$\frac{1}{qn}\|\mathbf{X}(\widehat{\mathbf{C}}_k - \mathbf{C}_k^*)\|_F^2 \le \phi_s^{-1}(VC_u + C_v)^2\left(\frac{s}{n}\log\frac{pq}{\delta}\right) + o\left(\frac{s}{n}\log\frac{pq}{\delta}\right).$$

Thus we have

$$\frac{1}{\sqrt{qn}}\|\mathbf{X}(\widehat{\mathbf{C}}_k - \mathbf{C}_k^*)\|_F \le \frac{(VC_u + C_v)}{\phi_s^{1/2}}\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right) + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right).$$

By a similar but simpler argument, it follows from the decomposition (21) that

$$\begin{aligned}\frac{1}{\sqrt{q}}\|\mathbf{C}_k^* - \widehat{\mathbf{C}}_k\|_F &\le \frac{1}{\sqrt{q}}\|(\mathbf{u}_k^* - \widehat{\mathbf{u}}_k)\mathbf{v}_k^{*T}\|_F + \frac{1}{\sqrt{q}}\|\widehat{\mathbf{u}}_k(\mathbf{v}_k^* - \widehat{\mathbf{v}}_k)^T\|_F \\ &= \frac{1}{\sqrt{q}}\|\mathbf{u}_k^* - \widehat{\mathbf{u}}_k\|_2\cdot\|\mathbf{v}_k^*\|_2 + \frac{1}{\sqrt{q}}\|\widehat{\mathbf{u}}_k\|_2\cdot\|\mathbf{v}_k^* - \widehat{\mathbf{v}}_k\|_2 \\ &\le (VC_u + C_v)\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right) + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right).\end{aligned}$$

It concludes the proof.

A.2. Proof of Theorem 2

The main idea for proving the rank consistency result is noting that $\widehat{\sigma}_k^2 = (\widehat{\mathbf{u}}_k^\top\widehat{\mathbf{Q}}\widehat{\mathbf{u}}_k)/(\widehat{\mathbf{u}}_k^\top\mathbf{P}\widehat{\mathbf{u}}_k)$ which relates to the termination criteria is in fact the kth largest eigenvalue of the problem $\widehat{\mathbf{Q}}\mathbf{u} = \lambda\mathbf{P}\mathbf{u}$. Thus, we will show that the amount of decrease in the objective function in the kth step equals to the the kth largest eigenvalue of the problem $\widehat{\mathbf{Q}}\mathbf{u} = \lambda\mathbf{P}\mathbf{u}$. Given the perturbation bounds in Lemma 6, we can bound its difference from the true eigenvalue and show that after r^* greedy steps, the eigenvalues become almost zero.

To prove the result in Theorem 2, we need to bound $\log\mathcal{L}_{k-1} - \log\mathcal{L}_k$, where $\mathcal{L}_k = \frac{1}{nq}\|\mathbf{Y} - \mathbf{X}\mathbf{C}_k\|_F^2$ is the objective function with $\mathbf{C}_k = \sum_{j=1}^k\widehat{\mathbf{C}}_j$ the estimated coefficient matrix up to the kth step. By using the fact that $1 - \frac{1}{x} \le \log(x) \le x - 1$ for $x > 0$, we can write

$$\frac{\mathcal{L}_{k-1} - \mathcal{L}_k}{\mathcal{L}_{k-1}} \le \log\left(\frac{\mathcal{L}_{k-1}}{\mathcal{L}_k}\right) \le \frac{\mathcal{L}_{k-1} - \mathcal{L}_k}{\mathcal{L}_k}.$$

Next, we show that the amount of the decrease in the objective function in the kth step is equal to the the kth largest eigenvalue of the problem $\widehat{\mathbf{Q}}\mathbf{u} = \lambda\mathbf{P}\mathbf{u}$, which satisfies the following equality

$$\begin{aligned}\mathcal{L}_{k-1} - \mathcal{L}_k &= \frac{1}{nq}\|\mathbf{Y} - \mathbf{X}\mathbf{C}_{k-1}\|_F^2 - \frac{1}{nq}\|\mathbf{Y} - \mathbf{X}\mathbf{C}_k\|_F^2 \\ &= \frac{1}{nq}\|\mathbf{Y} - \mathbf{X}\mathbf{C}_{k-1}\|_F^2 - \frac{1}{nq}\|\mathbf{Y} - \mathbf{X}\mathbf{C}_{k-1} - \mathbf{X}\widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^\top\|_F^2 \\ &= \frac{1}{nq}\left(2\left\langle \mathbf{Y} - \mathbf{X}\mathbf{C}_{k-1}, \mathbf{X}\widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^\top\right\rangle - \|\mathbf{X}\widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^\top\|_F^2\right).\end{aligned}$$

Now, using the $\mathbf{P}$-orthogonality of $\widehat{\mathbf{u}}_k$'s, we have

$$\begin{aligned}\mathcal{L}_{k-1} - \mathcal{L}_k &= \frac{1}{nq}\left(2\left\langle \mathbf{Y}, \mathbf{X}\widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^\top\right\rangle - \|\mathbf{X}\widehat{\mathbf{u}}_k\widehat{\mathbf{v}}_k^\top\|_F^2\right) \\ &= \frac{1}{nq}\left(2\frac{\widehat{\mathbf{u}}_k^\top\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\widehat{\mathbf{u}}_k}{\widehat{\mathbf{u}}_k^\top\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}_k} - \frac{\widehat{\mathbf{u}}_k^\top\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\widehat{\mathbf{u}}_k}{\widehat{\mathbf{u}}_k^\top\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}_k}\right) \\ &= \frac{\widehat{\mathbf{u}}_k^\top\widehat{\mathbf{Q}}\widehat{\mathbf{u}}_k}{\widehat{\mathbf{u}}_k^\top\mathbf{P}\widehat{\mathbf{u}}_k} = \widehat{\sigma}_k^2,\end{aligned}$$

where the second step is due to the substitution of the solution for $\widehat{\mathbf{v}}_k$ in (6) and a few steps of algebraic rearrangement.

Underfitted regime $k \le r^*$. Under Condition 1, using the perturbation bound in Lemma 6 for the eigenvalues σ_k^2's with $k \le r^*$, we can write

$$\widehat{\sigma}_k^2 \ge \sigma_k^2 - \phi_s^{-1}\|\mathbf{\Delta}_Q\|_{2,s}.$$

Further applying inequality (16), we know that there exists some positive constant C such that

$$\phi_s^{-1}\|\mathbf{\Delta}_Q\|_{2,s} \le 2a^*\phi_s^{-1}\gamma P\sqrt{\frac{2s}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right) \le C\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}.$$

Therefore, it follows that with probability at least $1-\delta$,

$$\widehat{\sigma}_k^2 \ge \sigma_k^2 - C\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}. \tag{22}$$

Then we can derive the following lower bound

$$\sqrt{\frac{\mathcal{L}_{k-1} - \mathcal{L}_k}{\mathcal{L}_{k-1}}} \ge \frac{\sqrt{\sigma_k^2 - C\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}}}{\frac{1}{\sqrt{nq}}\left(\sum_{j=k}^{r^*}\|\mathbf{X}\mathbf{C}_j^*\|_F + \sum_{j=1}^{k-1}\|\mathbf{X}(\mathbf{C}_j^* - \widehat{\mathbf{C}}_j)\|_F + \|\mathbf{E}\|_F\right)}, \tag{23}$$

where $\mathbf{C}_j^* = \mathbf{u}_j^*\mathbf{v}_j^{*T}$, $\widehat{\mathbf{C}}_j = \widehat{\mathbf{u}}_j\widehat{\mathbf{v}}_j$, in the nominator we have used the result obtained in (22) and the denominator is the result of the triangular inequality. We will then bound the three terms in the denominator successively.

For the first term, given the definition of $\mathbf{C}_j^* = \mathbf{u}_j^*\mathbf{v}_j^{*T}$, $\mathbf{v}_j^{*T}\mathbf{v}_j^* = qa_j^2$ and $\mathbf{u}_j^{*T}\mathbf{P}\mathbf{u}_j^* = c_j^2$, we have

$$\frac{\|\mathbf{X}\mathbf{C}_j^*\|_F}{\sqrt{nq}} = \sqrt{\frac{\mathrm{trace}(\mathbf{X}\mathbf{u}_j^*\mathbf{v}_j^{*T}\mathbf{v}_j^*\mathbf{u}_j^{*T}\mathbf{X}^T)}{nq}} = a_j\sqrt{\frac{\mathbf{u}_j^{*T}\mathbf{X}^T\mathbf{X}\mathbf{u}_j^*}{n}} = a_jc_j = \sigma_j.$$

It follows that $\frac{1}{\sqrt{nq}}\sum_{j=k}^{r^*}\|\mathbf{X}\mathbf{C}_j^*\|_F = \sum_{j=k}^{r^*}\sigma_j$.

To bound the second term, by Theorem 1, we have

$$\frac{1}{nq}\|\mathbf{X}(\mathbf{C}_j^* - \widehat{\mathbf{C}}_j)\|_F^2 \le \phi_s^{-1}(VC_u + C_v)^2\left(\frac{s}{n}\log\frac{pq}{\delta}\right) + o\left(\frac{s}{n}\log\frac{pq}{\delta}\right),$$

which gives

$$\sum_{j=1}^{k-1}\frac{\|\mathbf{X}(\mathbf{C}_j^* - \widehat{\mathbf{C}}_j)\|_F}{\sqrt{nq}} \le \frac{(k-1)(VC_u + C_v)}{\phi_s^{1/2}}\sqrt{\frac{s}{n}\log\frac{pq}{\delta}} + o\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right). \tag{24}$$

To bound the last term, as the components of $\mathbf{E}\mathbf{\Sigma}^{-1/2}$ are independent and identically distributed with the standard Gaussian distribution, given the tail bound for the χ^2 distribution in (Laurent and Massart, 2000, Lemma 1), we can see that with probability at least $1-\delta$,

$$\|\mathbf{E}\mathbf{\Sigma}^{-1/2}\|_F^2/nq \le 1 + 2\sqrt{\frac{1}{nq}\log\frac{1}{\delta}} + \frac{2}{nq}\log\frac{1}{\delta}.$$

Moreover, by Condition 3, we have

$$\|\mathbf{E}\mathbf{\Sigma}^{-1/2}\|_F^2 = \sum_{i=1}^n\|\mathbf{E}_{i:}\mathbf{\Sigma}^{-1/2}\|_2^2 \ge \sum_{i=1}^n\|\mathbf{E}_{i:}\|_2^2/\gamma_u^2 = \|\mathbf{E}\|_F^2/\gamma_u^2.$$

It gives

$$\|\mathbf{E}\|_F/\sqrt{nq} \le \gamma_u\left(1 + \sqrt{\frac{2}{nq}\log\frac{1}{\delta}}\right). \tag{25}$$

Thus, applying the union bound, we can see that with probability at least $1-2\delta$, the results in Theorem 1 (including inequality (16)), inequalities (24) and (25) hold simultaneously. In view of (23), it yields that

$$\sqrt{\frac{\mathcal{L}_{k-1} - \mathcal{L}_k}{\mathcal{L}_{k-1}}} \ge \frac{\sigma_k + O\left(\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right)}{\sum_{j=k}^{r^*}\sigma_j + \gamma_u + O\left(r^*\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right)}, \tag{26}$$

where we used the inequality $\sqrt{b-x} \ge \sqrt{b} - x/\sqrt{b}$ for $0 \le x \le b$ in the numerator. Let $\delta^{-1} = O\{(pq)^\alpha\}$ for some positive constant $\alpha > 1$. Since $\sum_{j=k}^{r^*}\sigma_j \le (r^* - k + 1)\sigma_k$,

$\gamma_u \le c_\gamma \sigma_{r^*} \le c_\gamma \sigma_k$ by Condition 3 and $r^* \left\{\frac{s\log(pq)}{n}\right\}^{1/4} = o(1)$, we get

$$\sqrt{\frac{\mathcal{L}_{k-1}-\mathcal{L}_k}{\mathcal{L}_{k-1}}} \ge \frac{1}{r^*-k+1+c_\gamma} + O\left(r^*\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right) = \frac{1}{r^*-k+1+c_\gamma} + o\left(\frac{1}{r^*}\right). \tag{27}$$

Overfitted regime $k > r^*$. Similar to the previous argument, by Condition 1 and Lemma 6, with probability at least $1-\delta$, for all $k > r^*$ we have

$$\widehat{\sigma}_k^2 \le \phi_s^{-1}\|\mathbf{\Delta}_Q\|_{2,s} \le C\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}. \tag{28}$$

Then we derive the following upper bound

$$\sqrt{\frac{\mathcal{L}_{k-1}-\mathcal{L}_k}{\mathcal{L}_k}} \le \frac{O\left\{\left(\frac{s}{n}\log\frac{pq}{\delta}\right)^{1/4}\right\}}{\frac{1}{\sqrt{nq}}\left(\|\mathbf{E}\|_F - \sum_{j=1}^{r^*}\|\mathbf{X}(\mathbf{C}_j^*-\widehat{\mathbf{C}}_j)\|_F - \sum_{j=r^*+1}^{k}\|\mathbf{X}\widehat{\mathbf{C}}_j\|_F\right)}. \tag{29}$$

Bounds on the three terms in the denominator will be derived successively. First, another application of the tail bound for the χ^2 distribution in (Laurent and Massart, 2000, Lemma 1) gives that with probability at least $1-\delta$,

$$\|\mathbf{E}\mathbf{\Sigma}^{-1/2}\|_F^2/nq \ge 1 - 2\sqrt{\frac{1}{nq}\log\frac{1}{\delta}}.$$

Similarly, together with Condition 3, we have

$$\|\mathbf{E}\|_F/\sqrt{nq} \ge \gamma_l\left(1 - 2\sqrt{\frac{1}{nq}\log\frac{1}{\delta}}\right).$$

Moreover, by inequality (24), we get

$$\sum_{j=1}^{r^*}\frac{1}{\sqrt{nq}}\left\|\mathbf{X}(\mathbf{C}_j^*-\widehat{\mathbf{C}}_j)\right\|_F \le O\left(r^*\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right).$$

For the last term, by the estimation procedure of $\widehat{\mathbf{v}}_j$ in equation (6) and the fact that $\widehat{\mathbf{u}}_j$ is the eigenvector of the generalized eigenvalue problem $\widehat{\mathbf{Q}}\mathbf{u} = \lambda\mathbf{P}\mathbf{u}$ with respect to the jth largest eigenvalue $\widehat{\sigma}_j^2$, we have

$$\frac{1}{nq}\|\mathbf{X}\widehat{\mathbf{C}}_j\|_F^2 = \frac{1}{nq}\mathrm{trace}(\widehat{\mathbf{v}}_j\widehat{\mathbf{u}}_j^T\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}_j\widehat{\mathbf{v}}_j^T) = \frac{1}{nq}(\widehat{\mathbf{u}}_j^T\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}_j)\widehat{\mathbf{v}}_j^T\widehat{\mathbf{v}}_j = \frac{1}{nq}\left(\frac{\widehat{\mathbf{u}}_j^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\widehat{\mathbf{u}}_j}{\widehat{\mathbf{u}}_j^T\mathbf{X}^T\mathbf{X}\widehat{\mathbf{u}}_j}\right) = \frac{\widehat{\mathbf{u}}_j^T\widehat{\mathbf{Q}}\widehat{\mathbf{u}}_j}{\widehat{\mathbf{u}}_j^T\mathbf{P}\widehat{\mathbf{u}}_j} = \widehat{\sigma}_j^2.$$

Thus, it follows from inequality (28) that

$$\sum_{j=r^*+1}^{k}\frac{1}{\sqrt{nq}}\|\mathbf{X}\widehat{\mathbf{C}}_j\|_F = \sum_{j=r^*+1}^{k}\widehat{\sigma}_j \le O\left\{(r-r^*)\left(\frac{s}{n}\log\frac{pq}{\delta}\right)^{1/4}\right\}.$$

In view of inequality (29), using the same argument as in (26), the above bounds yield

$$\sqrt{\frac{\mathcal{L}_{k-1}-\mathcal{L}_k}{\mathcal{L}_k}} \le \frac{O\left\{\left(\frac{s}{n}\log\frac{pq}{\delta}\right)^{1/4}\right\}}{\gamma_l + O\left(r^*\sqrt{\frac{s}{n}\log\frac{pq}{\delta}}\right) + O\left\{(r-r^*)\left(\frac{s}{n}\log\frac{pq}{\delta}\right)^{1/4}\right\}}.$$

Since $\delta^{-1} = O\{(pq)^\alpha\}$, $r^*\left\{\frac{s\log(pq)}{n}\right\}^{1/4} = o(1)$ and $r\left\{\frac{s\log(pq)}{n}\right\}^{1/4} = o(1)$, we conclude that

$$\sqrt{\frac{\mathcal{L}_{k-1}-\mathcal{L}_k}{\mathcal{L}_k}} \le O\left\{\left(\frac{s}{n}\log\frac{pq}{\delta}\right)^{1/4}\right\}. \tag{30}$$

In view of the bounds in (27) and (30), using the union bound, it is not difficult to see that with probability at least $1-3\delta$ (both bounds are based on the event in Lemma 5 such that the results in Theorem 1 hold), the following bounds hold,

$$\text{For } k \le r^*: \quad \log\left(\frac{\mathcal{L}_{k-1}}{\mathcal{L}_k}\right) \ge \frac{1}{(r^*-k+1+c_\gamma)^2} + o\left\{(r^*)^{-2}\right\},$$
$$\text{For } k > r^*: \quad \log\left(\frac{\mathcal{L}_{k-1}}{\mathcal{L}_k}\right) \le O\left\{\left(\frac{s}{n}\log\frac{pq}{\delta}\right)^{1/2}\right\}.$$

Thus, given the information criterion $\mathcal{C}_n = \mathrm{rank}(\widehat{\mathbf{C}})\sqrt{\log(pq)}\log\log n + \sqrt{n}\log\mathcal{L}_n$ and the assumption $(r^*)^2\sqrt{\log(pq)} = o(\sqrt{n}/\log\log n)$, by setting $\delta = \frac{1}{3}(pq)^{-\alpha}$, we can show that the following statements hold with probability at least $1-(pq)^{-\alpha}$:

$$\mathcal{C}_n(\mathrm{rank}(\widehat{\mathbf{C}}) = k-1) - \mathcal{C}_n(\mathrm{rank}(\widehat{\mathbf{C}}) = k) > 0, \qquad \text{if } k \le r^*;$$
$$\mathcal{C}_n(\mathrm{rank}(\widehat{\mathbf{C}}) = k-1) - \mathcal{C}_n(\mathrm{rank}(\widehat{\mathbf{C}}) = k) < 0, \qquad \text{if } k > r^*.$$

The above equations indicate that $\mathcal{C}_n$ will attain its minimum value when $\mathrm{rank}(\widehat{\mathbf{C}}) = r^*$, which means the algorithm will stop at $\mathrm{rank}(\widehat{\mathbf{C}}) = r^*$ with probability at least $1-(pq)^{-\alpha}$ for sufficiently large n, which concludes the proof of Theorem 2.

References

Alekh Agarwal, Sahand Negahban, and Martin J. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, 40(2): 1171–1197, 2012.

Francis R. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9(2):1019–1048, 2008.

M. Taha Bahadori and Yan Liu. An examination of practical granger causality inference. *SIAM International Conference on Data Mining*, pages 467–475, 2013.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

Ron Bekkerman, Mikhail Bilenko, and John Langford. *Scaling Up Machine Learning: Parallel and Distributed Approaches*. Cambridge University Press, 2012.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media, 2011.

Florentina Bunea, Yiyuan She, and Marten H. Wegkamp. Optimal selection of reduced rank estimators of high-dimensional matrices. *The Annals of Statistics*, 39(2):1282–1309, 2011.

Florentina Bunea, Yiyuan She, and Marten H. Wegkamp. Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *The Annals of Statistics*, 40 (5):2359–2388, 2012.

Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Tony T. Cai, Zongming Ma, and Yihong Wu. Sparse PCA: Optimal rates and adaptive estimation. *The Annals of Statistics*, 41(6):3074–3110, 2013.

Emmanuel J. Candès and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory*, 57(4):2342–2359, 2011.

Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. Measuring user influence in twitter: The million follower fallacy. *International AAAI Conference on Weblogs and Social*, 10:10–17, 2010.

Venkat Chandrasekaran, Pablo Parrilo, and Alan S. Willsky. Latent variable graphical model selection via convex optimization. *Annual Allerton Conference on Communication, Control, and Computing*, pages 1610–1613, 2010.

Kun Chen and Kung-Sik Chan. A note on rank reduction in sparse multivariate regression. *Journal of Statistical Theory and Practice*, 10(1):100–120, 2015.

Kun Chen, Kung-Sik Chan, and Nils Chr. Stenseth. Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):203–221, 2012.

Kun Chen, Hongbo Dong, and Kung-Sik Chan. Reduced rank regression via adaptive nuclear norm penalization. *Biometrika*, 100(4):901–920, 2013.

Fernando De La Torre and Michael J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

Konstantinos I. Diamantaras and Sun Y. Kung. *Principal Component Neural Networks: Theory and Applications*. John Wiley & Sons, 1996.

Varun R. Embar, Rama K. Pasumarthi, and Indrajit Bhattacharya. A Bayesian framework for estimating properties of network diffusions. *International Conference on Knowledge Discovery and Data Mining*, pages 1216–1225, 2014.

Yingying Fan and Jinchi Lv. Asymptotic equivalence of regularization methods in thresholded parameter space. *Journal of the American Statistical Association*, 108(503):1044–1061, 2013.

Yingying Fan and Chengyong Tang. Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):531–552, 2013.

Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan. Robust subspace segmentation with block-diagonal prior. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3818–3825, 2014.

Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data*, 5(4):21, 2012.

James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

Alan J. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.

Willard James and Charles Stein. Estimation with quadratic loss. *Berkeley Symposium on Mathematical Statistics and Probability*, 1:361–379, 1961.

Iain M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, 2001.

Iain M. Johnstone and Arthur Y. Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486): 682–693, 2009.

U Kang, Brendan Meeder, and Christos Faloutsos. Spectral analysis for billion-scale graphs: discoveries and implementation. *Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 13–25, 2011.

Béatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Jing Lei and Vincent Q. Vu. Sparsistency and agnostic inference in sparse PCA. *The Annals of Statistics*, 43(1):299–322, 2015.

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: an approach to modeling networks. *Journal of Machine Learning Research*, 11(3):985–1042, 2008.

Aurélie C. Lozano, Grzegorz Swirszcz, and Naoki Abe. Group orthogonal matching pursuit for logistic regression. *International Conference on Artificial Intelligence and Statistics*, pages 452–460, 2011.

Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

Aditya Mishra, Dipak K. Dey, and Kun Chen. Sequential co-sparse factor regression. *Journal of Computational and Graphical Statistics*, 26(4):814–825, 2017.

Seth A. Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. *International Conference on Knowledge Discovery and Data Mining*, pages 33–41, 2012.

Sahand Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q balls. *IEEE Transactions on Information Theory*, 57(10):6976–6994, 2011.

Emile Richard, Stéphane Gaïffas, and Nicolas Vayatis. Link prediction in graphs with autoregressive features. *Journal of Machine Learning Research*, 15(1):489–517, 2012.

Dan Shen, Haipeng Shen, and J. S. Marron. A general framework for consistency of principal component analysis. *Journal of Machine Learning Research*, 17(1):5218–5251, 2016.

Kate Starbird and Leysia Palen. (How) will the revolution be retweeted?: information diffusion and the 2011 Egyptian uprising. *ACM Conference on Computer Supported Cooperative Work*, pages 7–16, 2012.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 58(1):267–288, 1996.

Wilson Truccolo, Uri T. Eden, Matthew R. Fellows, John P. Donoghue, and Emery N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–89, 2005.

Michael Trusov, Randolph E. Bucklin, and Koen Pauwels. Effects of word-of-mouth versus traditional marketing: findings from an internet social networking site. *Journal of Marketing*, 73(5):90–102, 2009.

Raja Velu and Gregory C. Reinsel. *Multivariate Reduced-Rank Regression: Theory and Applications*. Springer Science & Business Media, 2013.

Yu-Xiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: when LRR meets SSC. *International Conference on Neural Information Processing Systems*, pages 64–72, 2013.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Ming Yuan, Ali Ekici, Zhaosong Lu, and Renato Monteiro. Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):329–346, 2007.

Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14(1):899–925, 2013.

Tong Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, 57(7):4689–4708, 2011.

Zemin Zheng, Yingying Fan, and Jinchi Lv. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):627–649, 2014.

Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. *International Conference on Artificial Intelligence and Statistics*, 31:641–649, 2013.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Journal of Machine Learning Research 20 (2019) 1-30 Submitted 5/18; Revised 4/19; Published 6/19

Maximum Likelihood for Gaussian Process Classification and Generalized Linear Mixed Models under Case-Control Sampling

Omer Weissbrod OWEISSBROD@HSPH.HARVARD.EDU
Epidemiology Department
Harvard T.H. Chan School of Public Health
Boston, MA 02115, USA

Shachar Kaufman SHACHARK@POST.TAU.AC.IL
Department of Statistics and Operations Research
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv 6139101, Israel

David Golan DAVID@VIZ.AI
Viz.ai
Tel-Aviv 6121001, Israel

Saharon Rosset SAHARON@POST.TAU.AC.IL
Department of Statistics and Operations Research
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv 6139101, Israel

Editor: Ryan Adams

Abstract

Modern data sets in various domains often include units that were sampled non-randomly from the population and have a latent correlation structure. Here we investigate a common form of this setting, where every unit is associated with a latent variable, all latent variables are correlated, and the probability of sampling a unit depends on its response. Such settings often arise in case-control studies, where the sampled units are correlated due to spatial proximity, family relations, or other sources of relatedness. Maximum likelihood estimation in such settings is challenging from both a computational and statistical perspective, necessitating approximations that take the sampling scheme into account. We propose a family of approximate likelihood approaches which combine composite likelihood and expectation propagation. We demonstrate the efficacy of our solutions via extensive simulations. We utilize them to investigate the genetic architecture of several complex disorders collected in case-control genetic association studies, where hundreds of thousands of genetic variants are measured for every individual, and the underlying disease liabilities of individuals are correlated due to genetic similarity. Our work is the first to provide a tractable likelihood-based solution for case-control data with complex dependency structures.

Keywords: Gaussian Processes, Expectation Propagation, Composite Likelihood, Selection Bias, Linear Mixed Models

©2019 Omer Weissbrod, Shachar Kaufman, David Golan and Saharon Rosset.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-298.html.

1. Introduction

In the analysis of scientific data, a common phenomenon is the existence of complex dependencies between analyzed units. This is encountered in diverse fields such as epidemiology, econometrics, ecology, geostatistics, psychometrics and genetics, and can arise due to spatial correlations, temporal correlations, family relations, or other sources of heterogeneity (Pfeiffer, 2008; Rabe-Hesketh et al., 2005; Bolker et al., 2009; Rabe-Hesketh et al., 2004; Yang et al., 2014; Burton et al., 1999; Diggle et al., 1998). This idea is often captured through the use of Gaussian processes (GPs; Rasmussen and Williams 2006) or equivalently, through generalized linear mixed models (GLMMs; McCulloch et al. 2008) or latent Gaussian models (Fahrmeir and Tutz, 2001). Such models associate sampled units with latent variables, and express the dependencies through covariance matrices of latent variables.

A second important concept is that of *ascertainment*, where the probability of sampling a unit depends on its response. Ascertainment is especially common in case-control studies where a binary response variable has a rare outcome, such as a rare disease, leading to oversampling of disease cases relative to their population prevalence (Breslow, 1996).

In this paper we consider situations that contain both elements—a complex covariance structure and case-control sampling—and the statistical modeling solutions available for these situations. Our interest lies in an extreme form of this combination, involving:

(a) Unit-level ascertainment, where the sampling probability of a unit depends only on its response (Figure 1a). This stands in contrast to common study designs such as family studies (Neuhaus et al., 2002), ascertained longitudinal studies (Liang and Zeger, 1986) or clustered case-control studies (Neuhaus and Jewell, 1990). In such studies each cluster is either entirely selected or entirely omitted from the study, such that the dependency structure in the sample and in the population are the same.

(b) A full-rank covariance matrix, indicating that dependencies cannot be captured by a small number of variables (Figure 1b). Such settings are common in modern data sets due to either high-dimensional settings or to the use of kernels or basis expansions, which implicitly project a small number of features into a large (possibly infinite-dimensional) space (Diggle et al., 1998; Rasmussen and Williams, 2006).

(c) A non-sparse covariance matrix, indicating that all units are correlated (Figure 1c). Such settings exacerbate both the computational challenge, because the density does not factorize into multiplicative terms, and the statistical challenge, because classic statistical theory requires a large number of independent units.

Special cases of this combination have been addressed in the literature (Glidden and Liang, 2002; Epstein et al., 2002; Neuhaus et al., 2006, 2014), but to our knowledge, there is a limited set of available solutions for the general setting, which is indeed very challenging.

A major motivating application for our study is genome-wide association studies of diseases with a case-control sampling design (GWAS-CC) (Price et al., 2015; Visscher et al., 2017). In GWAS-CC, the genomes of individuals affected with a disease and of unaffected controls are collected in an effort to uncover the genetic mechanisms driving disease risk. Studies in this field have diverse goals, reflected in the diversity of the statistical inference tasks they seek to solve: testing for association between genetic variants and disease (Yang

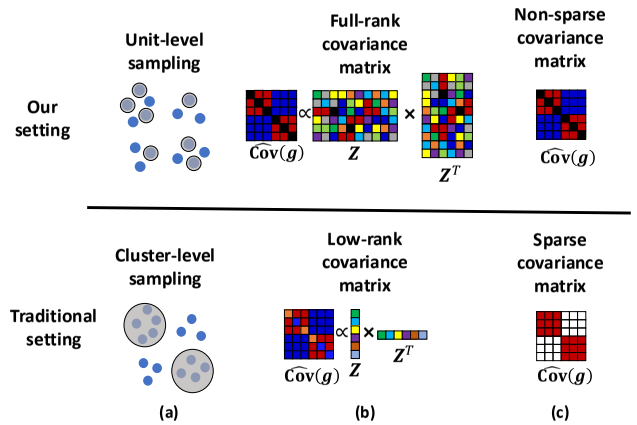


Figure 1: The properties that are unique to our setting of interest (top row) compared to more traditional settings (bottom row). (a) Unit-level sampling, where the decision whether to sample a unit depends only on their response, in contrast to studies that either sample or omit an entire cluster of units. (b) A full-rank covariance matrix of latent variables, indicating a complex dependency structure. (c) A non-sparse covariance matrix, indicating that the latent variables of every pair of units are correlated.

et al., 2014; Weissbrod et al., 2015), estimating disease heritability (Yang et al., 2010; Golan et al., 2014; Weissbrod et al., 2018), risk prediction (Zhou et al., 2013; Golan and Rosset, 2014; Weissbrod et al., 2016), and more.

GWAS-CC typically employ case-control designs, where patients are recruited in hospitals or clinics whereas healthy controls are recruited independently, owing to the small prevalence of complex genetic diseases—even common diseases like type 1 diabetes or schizophrenia typically have a population prevalence $\leq$1%. Furthermore, statistical models for such studies typically treat the effects of genetic variants on disease as random effects sampled from a distribution. This is because such studies include millions of variants, and the effects are typically very small (Yang et al., 2010; Golan et al., 2014; Golan and Rosset, 2014). Hence, GWAS-CC give rise to settings with case-control sampling and a full-rank and non-sparse covariance structure: every individual has a latent genetic effect (given by the inner product of her genotype and the random effects), and the genetic effects are correlated due to genetic similarity. Despite the extensive interest that GWAS-CC have attracted in recent years (Wellcome Trust Case Control Consortium et al., 2007; Ehret et al., 2011; Sawcer et al., 2011; Ripke et al., 2014; Okada et al., 2014), the statistical modeling problems this setting generates have been discussed in a limited manner, with application of heuristic methods that do not formally take the probabilistic structure of the problem into account

(Lee et al., 2011; Hayeck et al., 2015; Weissbrod et al., 2015; Chen et al., 2016; Jiang et al., 2016a).

Similar settings arise in other scientific domains, where case-control sampling and a full-rank, non-sparse covariance structure are observed. Prominent examples include disease mapping studies with a smoothing kernel (Diggle et al., 1998; Kelsall and Diggle, 1998; Held et al., 2005) and GP-based classification of data collected in case-control studies (Chu et al., 2010; Ziegler et al., 2014; Young et al., 2013). The analyses employed in these examples often ignore the effects of case-control sampling, a practice we would like to avoid and whose fundamental flaws we discuss and illustrate below.

The problem we consider poses substantial statistical and computational challenges. The main statistical framework for inference with binary responses and latent variables are GP classifiers, which are mathematically equivalent to latent Gaussian models, and recover GLMMs as a special case when using a linear kernel. Such models provide a likelihood-based solution but can pose significant computational difficulties. Modern approach to alleviate computational difficulties include (1) Pairwise likelihood (PL; Renard et al. 2004), which approximates the joint distribution of all variables as a product of marginal distributions of pairs of variables; (2) Expectation propagation (EP) (Minka, 2001; Seeger, 2005), which replaces multiplicative terms in the distribution with simpler terms from an exponential family distribution; (3) Variational approximation (Opper and Archambeau, 2009; Hensman et al., 2015), which approximates the distribution with the closest distribution from a more tractable class; (4) Markov chain Monte Carlo (MCMC) sampling combined with thermodynamic integration (Kuss and Rasmussen, 2005; Nickisch and Rasmussen, 2008; Gelman and Meng, 1998), and (5) Laplace approximation (Tierney and Kadane, 1986; Raudenbush et al., 2000) and its close variant, penalized quasi likelihood approximation (Breslow and Clayton, 1993; Wolfinger and O'connell, 1993), which approximate the distribution as a Gaussian distribution via a second-order Taylor expansion. Of these, PL and EP have proved to often outperform the alternatives (Kuss and Rasmussen, 2005; Nickisch and Rasmussen, 2008; Varin et al., 2011) and thus form the basis of our proposed approach.

The main approach for statistical inference in the presence of ascertainment is called *ascertained maximum likelihood* (AML). AML consists of defining a binary variable indicating inclusion in the study for every unit, and conditioning the analysis on these variables (Scott and Wild, 2001). However, none of the above GP approximation methods is suitable for AML inference. The only existing method that is both scalable and statistically sound is a method-of-moments approach called phenotype-correlation-genotype-correlation (PCGC; Golan et al. 2014). However, this approach is not likelihood-based and thus cannot naturally be used for model comparison, inference, prediction, and hypothesis testing.

Here we propose two approaches for approximate likelihood computation in our setting of interest. Our approaches combine (1) GP + AML + PL, which provides a tractable likelihood approximation but is very sensitive to model misspecification; and (2) GP + AML + modified EP, which is more computationally intensive, but is more robust and is closer to traditional maximum likelihood estimation. We evaluate the merits of our approaches on both synthetic and real data sets of genetic studies involving thousands of individuals and hundreds of thousands of features treated as having random effects.

2. Formal Problem Description

Here we provide a formal description of our problem and the statistical and computational challenges.

2.1. Gaussian Processes / Generalized Linear Mixed Models

We are presented with n units, with each unit i having c features $\boldsymbol{X}_i \in \mathbb{R}^c$ associated with fixed (non-random) effects $\boldsymbol{\beta} \in \mathbb{R}^c$, m additional features $\boldsymbol{Z}_i \in \mathbb{R}^m$ that are not associated with fixed effects, and an outcome variable y_i (here we consider binary outcomes, but the framework can also be applied to other types). We associate every unit i with a latent variable $g_i(\boldsymbol{Z}_i)$ that depends on $\boldsymbol{Z}_i$ such that $P\left(y_i = 1 \mid \boldsymbol{X}_i, g_i, \boldsymbol{\beta}\right) = h\left(\boldsymbol{X}_i^T\boldsymbol{\beta} + g_i\right)$, where $h\left(\cdot\right)$ is a likelihood function (which is closely related to an inverse link function in GLMM terminology), such as probit or logit.

GPs impose a Gaussian process prior over the the functional form of the latent variables, $g \sim \mathcal{GP}\left(\mathbf{0}, k(\cdot,\cdot\,;\,\boldsymbol{\theta})\right)$, where $k(\cdot,\cdot\,;\,\boldsymbol{\theta})$ is a *kernel function* parameterized by $\boldsymbol{\theta}$. This indicates that for every finite set of units $i = 1 \ldots n$, the vector $\boldsymbol{g} = [g_1(\boldsymbol{Z}_1), \ldots, g_n(\boldsymbol{Z}_n),]^T$ follows a multivariate normal distribution:

$$\boldsymbol{g} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{K}(\boldsymbol{Z};\boldsymbol{\theta})\right),$$

where $\boldsymbol{K}$, which is non-sparse and full-rank in our setting, has entries $K_{ij} = k(\boldsymbol{Z}_i, \boldsymbol{Z}_j; \boldsymbol{\theta})$.

GLMMs are a class of models that are mathematically equivalent to GPs with linear kernels, i.e., $\boldsymbol{K}_{ij} = \theta \boldsymbol{Z}_i^T \boldsymbol{Z}_j$, where θ is a scalar hyperparameter called a *variance component*. GLMM literature typically uses the alternative notation $g_i = \boldsymbol{Z}_i^T \boldsymbol{b}$, where $\boldsymbol{b} \sim \mathcal{N}\left(\mathbf{0}, \sqrt{\theta}\boldsymbol{I}\right)$ is a vector of *random effects*. The equivalence can be extended to non-linear kernels by making use of Mercer's theorem (König, 2013), which allows expressing every kernel function as a (possibly infinite) linear combination of basis functions, $k(\boldsymbol{Z}_i, \boldsymbol{Z}_j; \boldsymbol{\theta}) = \sum_{r=1}^{\infty} \phi_r(\boldsymbol{Z}_i; \boldsymbol{\theta}) \cdot \phi_r(\boldsymbol{Z}_j; \boldsymbol{\theta})$. Hence, every GP can be written as a GLMM over the feature space $\{\phi_r(\boldsymbol{Z})\}$ with iid random effects $b_r \sim \mathcal{N}(0,1)$. GLMMs can also be defined with non-normal random effects, in which case the equivalence with GPs breaks down, but we do not consider such models here. We focus on linear kernels because we are interested in extremely high-dimensional settings, where non-linear kernels often overfit (Weissbrod et al., 2016). A graphical model of GPs and GLMMs is shown in Figure 2.

Our main aim in this work is estimating the kernel hyperparameters $\boldsymbol{\theta}$ (i.e., training the model). Given a vector of observed outcomes $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ and the matrices $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n]^T$, $\boldsymbol{Z} = [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n]^T$, the GP likelihood is given by:

$$L\left(\boldsymbol{\beta}, \boldsymbol{\theta}\right) = P\left(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\beta}\right) = \int P\left(\boldsymbol{g} | \boldsymbol{Z}, \boldsymbol{\theta}\right) \prod_i P(y_i | \boldsymbol{X}_i, g_i, \boldsymbol{\beta}) d\boldsymbol{g}. \tag{1}$$

As we do not impose a prior over the hyperparameters $\boldsymbol{\beta}, \boldsymbol{\theta}$, we can estimate them via type-II maximum likelihood, by finding the values that maximize Equation 1.

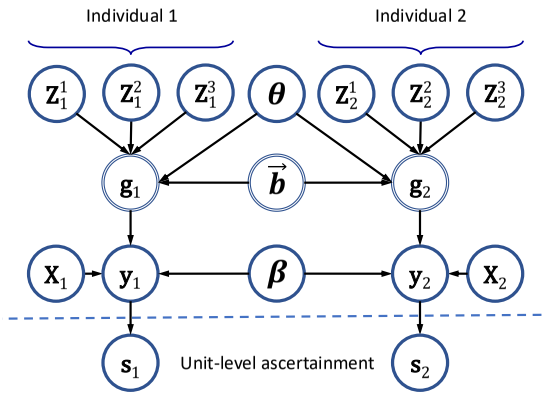


Figure 2: A directed graphical model for a GP with one feature X associated with a fixed effect β, three features Z^1, Z^2, Z^3 associated with an implicit vector of random effects $\vec{b}$ and with a hyperparameter θ, and two sampled units (indicated by subscript indices) with latent variables g_1, g_2 and observed responses y_1, y_2. Also shown is the extension to unit-level ascertainment, which consists of adding a sampling indicator s_i that depends on y_i and is equal to 1 for every sampled unit. Latent (non-observed) random variables are marked with a double-lined border.

2.2. The Implications of Ignoring Ascertainment in GPs

Up until now we implicitly assumed that n units were sampled completely at random from an underlying population. We now assume that the sample is ascertained, i.e., that the probability of sampling cases (units with $y_i = 1$) and controls ($y_i = 0$) is different.

We first demonstrate that using GPs while ignoring ascertainment leads to nonsensical conclusions which stand in contrast to fundamental motivations for GP use, like the central limit theorem. We focus on binary GPs, which can be formulated according to the liability threshold model (Dempster and Lerner, 1950). Under this model, every unit i has a latent liability $l_i = g_i + \epsilon_i$, where ϵ_i is an iid latent residual variable whose distribution depends on the likelihood function (e.g. normally distributed for probit, or logit distributed for logit), and unit i is a case (having $y_i = 1$) if and only if $l_i > t$ for some cutoff t. The *prevalence* K is the proportion of units in the population having $l_i > t$. We emphasize that a normally distributed ϵ_i is completely equivalent to a standard GP with a probit likelihood $h(\cdot)$.

It is common to use likelihood functions associated with a smooth and symmetrically distributed ϵ_i, such as logit or probit, which leads to a smooth and symmetric distribution of liabilities in the population. However, due to the ascertainment mechanism, the liabilities and latent variables g_i in an ascertained sample follow a non-symmetric and possibly discontinuous distribution (Figure 3a), and thus cannot be analyzed with standard likelihood functions. This problem motivates our proposed solutions for analysis of case-control

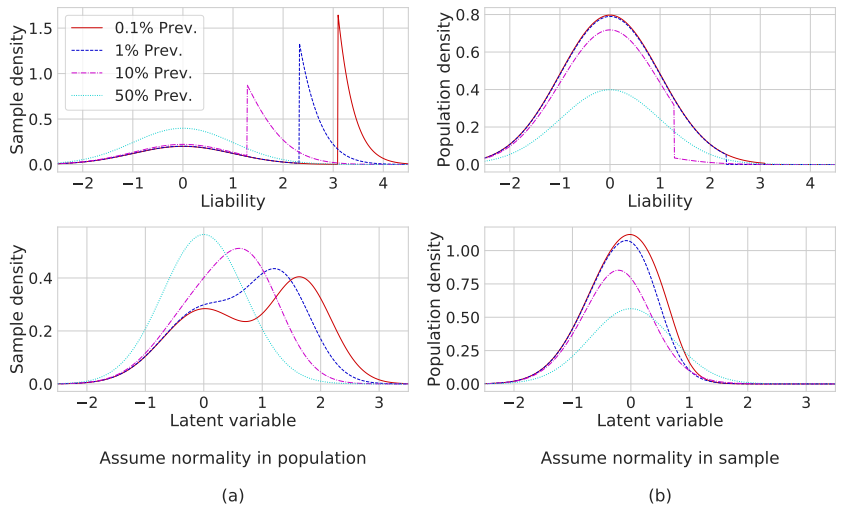


Figure 3: The implications of assuming normality of latent variables in the population from which units are sampled (panel a) or in a case-control study (panel b), for a GP with a probit likelihood and a sample consisting of 50% cases. The liability is given by $l_i = g_i + \epsilon_i,\ g_i, \epsilon_i \sim \mathcal{N}(0, \sqrt{0.5})$. Units with liabilities greater than their (1-prevalence) population quantile are cases. (a) When assuming normality in the population, latent variables and liabilities in a case-control study are not normally distributed (unless the cases prevalence is 50%, in which case there is no ascertainment). (b) When naively assuming normality of latent variables in a case-control study, the latent variables and the liabilities are *not* normally distributed in the population from which the data was sampled, in contradiction to the liability threshold model. Specifically, the liabilities distribution is discontinuous, and the latent variables distribution has a heavy left tail. All distributions were computed analytically by conditioning on the sampling indicators defined in Section 2.3.

studies. Many studies in practice ignore the complexities above, and instead use common likelihood functions such as a logit or a probit in case-control studies (e.g. Chen et al. 2016; Jiang et al. 2015; Kramer et al. 2017; Qi et al. 2017). However, this solution implies a non-symmetric and discontinuous distribution of latent variables in the population from which units are sampled, in stark contrast to the central limit theorem assumptions (Figure 3b). Thus, ignoring the ascertainment scheme in GPs may lead to nonsensical probabilistic settings under common assumptions.

The practical implications of modeling violations due to ascertainment have been investigated extensively in the statistical genetics literature. These include severe biases in estimation of quantities such as disease heritability (Golan et al., 2014; Weissbrod et al., 2018), inaccurate risk prediction (Golan and Rosset, 2014) and loss of power in hypothesis testing (Weissbrod et al., 2015; Hayeck et al., 2015; Yang et al., 2014).

2.3. Modeling Ascertainment in GPs

Analysis of ascertained data is typically performed by (1) defining a binary sampling indicator s_i for every unit i such that s_i depends only on y_i (Figure 2); and (2) performing all statistical inference tasks conditional on $s_1 = 1, \ldots, s_n = 1$. This requires specifying two numbers: the sampling probabilities of cases $P(s_i = 1|y_i = 1)$ and of controls $P(s_i = 1|y_i = 0)$. There are two main approaches for estimating the hyperparameters $\boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}$ of the distribution $P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z})$ with such indicators, differing with respect to how the sampling probabilities are determined.

Maximum profile likelihood estimates $\boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}$ by jointly maximizing the so-called profile likelihood $P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}, s_1 = 1, \ldots, s_n = 1)$ over both $\boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}$ and the nuisance hyperparameters $\boldsymbol{\theta}_{s|y}$ of the distributions $P(s_i = 1|y_i)$ of every possible value of y_i, i.e., $\hat{\boldsymbol{\theta}}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}, \hat{\boldsymbol{\theta}}_{s|y} = \text{argmax}_{\boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}, \boldsymbol{\theta}_{s|y}} P\left(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}, s_1 = 1, \ldots, s_n = 1, \boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}, \boldsymbol{\theta}_{s|y}\right)$ (Scott and Wild, 2001). The resulting estimator is maximally efficient as it attains the Cramér-Rao lower bound.

Ascertained Maximum Likelihood (AML) estimates $\boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}$ using a pre-specified assignment $\boldsymbol{\theta}_{s|y} = \boldsymbol{\theta}^0_{s|y}$, i.e., $\hat{\boldsymbol{\theta}}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}} = \text{argmax}_{\boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}} P\left(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}, s_1 = 1, \ldots, s_n = 1, \boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}}, \boldsymbol{\theta}^0_{s|y}\right)$ (Scott and Wild, 1997). For a binary outcome with a population prevalence K and an in-sample prevalence P, any assignment $\boldsymbol{\theta}^0_{s|y}$ obeying the constraint $\frac{P(s_i=1|y_i=0)}{P(s_i=1|y_i=1)} = \frac{K(1-P)}{(1-K)P}$ guarantees consistent estimates (i.e., estimators converge to the true parameter values as sample size tends to infinity if the model is true), because it yields the observed case-control ratio in expectation. This approach is often termed pseudo likelihood or conditional likelihood (Manski, 1981; Hsieh et al., 1985), but as both terms have alternative meanings in GLMM literature, we use the term ascertained likelihood instead. AML is less statistically efficient than maximum profile likelihood, in the sense that the estimator has a larger variance. However, the loss of efficiency has been shown to be negligible in practice (Wild, 1991; Scott and Wild, 1997). AML has previously been used for family-based studies (Glidden and Liang, 2002; Epstein et al., 2002), but to our knowledge it has not been used under the combination of a dependency structure and unit-level sampling.

To combine GPs with the AML framework, we define the ascertained GP likelihood and apply Bayes' law as follows:

$$L^*(\boldsymbol{\beta}, \boldsymbol{\theta}) = P(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s} = 1, \boldsymbol{\theta}, \boldsymbol{\beta}) = \frac{P(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\beta})}{P(\boldsymbol{s} = \boldsymbol{1} \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\beta})} \prod_i P(s_i = 1|y_i), \quad (2)$$

where $\boldsymbol{s} = 1$ is a shorthand notation for $s_1 = 1, \ldots, s_n = 1$ and $\boldsymbol{\theta}_{\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z}} = \{\boldsymbol{\theta}, \boldsymbol{\beta}\}$. The last term in the rhs of Equation 2 is considered known under AML and requires no special treatment. The numerator is the likelihood of a standard GP under no ascertainment, and the denominator is the likelihood of a GP in which the outcome is s_i instead of y_i. A naive approach is to approximate the numerator and denominator separately. However, obtaining an accurate estimate of the ratio is extremely challenging, because both the numerator and denominator are challenging to approximate, and any inaccuracy is compounded by the division. In our experience, this approach does not lead to reasonable estimators.

3. Approximate Inference in GPs under Ascertainment

We now propose two methods for approximate inference in GPs under ascertainment.

3.1. Ascertained Pairwise Likelihood

PL is a composite likelihood approximation, which approximates multivariate joint densities via products of marginal bivariate densities: (Varin et al., 2011; Renard et al., 2004):

$$P\left(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\beta}\right) \underset{\sim}{\propto} \prod_{i \neq j} P\left(y_i, y_j \mid \boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{Z}_i, \boldsymbol{Z}_j, \boldsymbol{\theta}, \boldsymbol{\beta}\right), \tag{3}$$

where i,j iterate over all pairs of units, and $\underset{\sim}{\propto}$ indicates approximate proportionality with respect to the hyperparameters $\boldsymbol{\theta}$, $\boldsymbol{\beta}$. The maximum pairwise likelihood estimate is approximately the maximum likelihood estimate. PL is computationally efficient owing to its quadratic dependency on the sample size, and is statistically consistent under suitable regularity conditions (Varin et al., 2011).

Ascertained PL (APL) is an extension of PL that approximates the ascertained likelihood in Equation 2 by modifying Equation 3 to condition on $\boldsymbol{s} = 1$:

$$\begin{aligned} P\left(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s} = \boldsymbol{1}\right) &\underset{\sim}{\propto} \prod_{i \neq j} P\left(y_i, y_j \mid \boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{Z}_i, \boldsymbol{Z}_j, s_i = 1, s_j = 1\right) \\ &= \prod_{i \neq j} \frac{P\left(y_i, y_j \mid \boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{Z}_i, \boldsymbol{Z}_j\right)}{P\left(s_i = s_j = 1 \mid \boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{Z}_i, \boldsymbol{Z}_j\right)} P\left(s_i = s_j = 1 \mid y_i, y_j\right), \end{aligned}$$

where $P\left(s_i = s_j = 1 \mid y_i, y_j\right) = P\left(s_i = 1 \mid y_i\right) P\left(s_j = 1 \mid y_j\right)$ are known constants which can be ignored, and we omitted the hyperparameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ for brevity. The terms in the numerator and the denominator can be separately evaluated as in standard PL, where we treat the denominator as a GP with a suitable likelihood function. Unlike Equation 2, the evaluation of the ratio is accurate since both the numerator and denominator can be computed exactly. In certain settings, PL evaluation can be substantially accelerated via a Taylor approximation around $\boldsymbol{Z}_i^T \boldsymbol{Z}_j = 0$, which enables factoring each bivariate distribution into a product of marginal distributions (Appendix A).

3.2. Ascertained Expectation Propagation

EP is a popular approach for approximating complex distributions by iteratively replacing every multiplicative term in the joint distribution of the observed and latent variables with a simpler term from an exponential family distribution (Minka, 2001; Rasmussen and Williams, 2006; Seeger, 2005). This joint distribution in GPs is given by $P\left(\boldsymbol{g} | \boldsymbol{Z}, \boldsymbol{\theta}\right) \prod_i P(y_i | \boldsymbol{X}_i, g_i, \boldsymbol{\beta})$. EP replaces every term in this product by an unnormalized Gaussian, $P\left(y_i | \boldsymbol{X}_i, g_i\right) \approx t_i\left(g_i\right) \triangleq r_i \mathcal{N}\left(g_i; \widetilde{\alpha}_i, \widetilde{\gamma}_i\right)$, where we omitted the hyperparameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ for brevity, and the site parameters r_i, $\widetilde{\alpha}_i$, $\widetilde{\gamma}_i$ implicitly depend on $\boldsymbol{X}_i$, y_i and $\boldsymbol{\beta}$. EP iteratively updates the terms $t_i(g_i)$, such that each term minimizes the generalized Kullback Leibler divergence (GKL) between the functions $q_{-i}(g_i) t_i(g_i)$ and $q_{-i}(g_i) P(y_i | \boldsymbol{X}_i, g_i)$ (i.e., the KL divergence between these functions after standardizing them to integrate to unity), where the cavity distribution $q_{-i}(g_i) \propto \int P(\boldsymbol{g} | \boldsymbol{Z}) \prod_{j \neq i} t_j(g_j) d\boldsymbol{g}_{j \neq i}$ represents the current approximation of $P\left(g_i | \boldsymbol{Z}, \boldsymbol{y}_{j \neq i}\right)$.

Given an EP approximation, the GP likelihood can be approximated as:

$$P\left(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{Z}\right) \approx \int P\left(\boldsymbol{g} | \boldsymbol{Z}\right) \prod_i t_i\left(g_i\right) d\boldsymbol{g}.$$

This expression can be evaluated analytically because it is an integral of a product of (unnormalized) Gaussian densities. EP has proven to consistently outperform alternative approximation methods for binary data (Nickisch and Rasmussen, 2008), and recent theoretical analysis has demonstrated that is is statistically consistent under certain modeling assumptions (Dehaene and Barthelmé, 2016, 2018).

Ascertained EP (AEP) is our proposed method to generalize standard EP to handle ascertainment. AEP approximates the ascertained likelihood in Equation 2 by replacing the standard EP step with a modified step that equates the functions $\int q_{-i}(g_i) t_i(g_i) dg_i$ and $\frac{\int q_{-i}(g_i) P(y_i, s_i | g_i, \boldsymbol{X}_i) dg_i}{\int q_{-i}(g_i) P(s_i | g_i, \boldsymbol{X}_i) dg_i}$. Unlike standard EP, we cannot minimize the GKL divergence beteween these functions, because this will lead to the same solution as standard EP, up to a scaling constant. Instead, AEP finds the unnormalized Gaussian $t_i(g_i)$ which makes these functions and their first two partial derivatives with respect to μ_{-i} (the mean of the Gaussian $q_{-i}(g_i)$) have the same value when evaluated at μ_{-i} (see Appendix B for details).

EP is a special case of AEP, because the proposed step objective coincides with the standard EP objective in the absence of ascertainment (i.e., when $P(s_i | y_i)$ is a constant regardless of y_i). To see this, observe that EP minimizes the GKL divergence between $\hat{m}(g_i) \triangleq q_{-i}(g_i) P(y_i | g_i, \boldsymbol{X}_i)$ and $\tilde{m}(g_i) \triangleq q_{-i}(g_i) \cdot t_i(g_i)$. Since $\tilde{m}(g_i)$ is an unnormalized Gaussian, EP minimizes the GKL divergence by equating its zeroth, first and second moments with those of $\hat{m}(g_i)$ (Rasmussen and Williams, 2006). Hence, standard EP requires computing the mean $\hat{\mu}_i$ and variance $\hat{\sigma}_i^2$ of $\hat{m}(g_i)$. A straightforward but lengthy derivation shows that we can compute these quantities as follows:

$$\begin{aligned} \hat{\mu}_i &= \frac{\partial}{\partial \mu_{-i}} \left[\log \int \hat{m}(g_i) dg_i\right] \sigma_{-i}^2 + \mu_{-i} \\ \hat{\sigma}_i^2 &= \frac{\partial^2}{\partial \left(\mu_{-i}\right)^2} \left[\log \int \hat{m}(g_i) dg_i\right] \left(\sigma_{-i}^2\right)^2 + \sigma_{-i}^2, \end{aligned}$$

where μ_{-i}, σ_{-i}^2 are the mean and variance of the Gaussian $q_{-i}(g_i)$, and the derivatives are evaluated at the actual value of μ_{-i}. Hence, there is a one-to-one correspondence between the first two moments of $\hat{m}(g_i)$ and its first two partial derivatives with respect to μ_{-i} (when evaluated at μ_{-i}). Consequently, each step of standard EP can alternatively be described as imposing the constraint that the zeroth, first and second derivatives of the integrals of $\tilde{m}(g_i)$ and $\hat{m}(g_i)$ with respect to μ_{-i} are the same. This is the same constraint used in AEP. Hence, EP and AEP coincide in the absence of ascertainment, where $P(s_i = 1 | y_i)$ is constant regardless of the value of y_i.

The sampling variance of the AEP maximum likelihood estimator can be estimated efficiently via jackknife sampling, by reusing the functions $t_i(g_i)$ (Opper and Winther, 2000; Qi et al., 2004; Vehtari et al., 2016). The evaluation of each jackknife sample requires inverting a matrix that is a submatrix of a matrix that was inverted in the original computation, with one row and one column removed. Such an inversion can be computed rapidly while retaining numerical stability, by combining a Cholesky decomposition with a series of Givens

rotations (Seeger, 2004). A formal analysis of AEP is difficult because there are relatively few theoretical guarantees for standard EP, which is a special case of AEP, except under relatively strong assumptions (Dehaene and Barthelmé, 2016, 2018). However, we sketch a heuristic argument supporting the objective function of AEP in Appendix B.

4. Results

We evaluated the performance of our methods using extensive simulations and real data analysis. We first describe our simulation studies, and then present the results obtained on real data.

4.1. Simulations Overview

We simulated data that closely mimics real GWAS-CC using the liability threshold model, where each individual has liability $l_i = \boldsymbol{X}_i^T\boldsymbol{\beta} + g_i + \epsilon_i$, g_i is a GP latent variable, and individuals with l_i greater than some cutoff are cases. In most simulations we used a linear kernel with a single hyperparameter, $\boldsymbol{\theta} = \{\sigma_g^2\}$, though we also investigate radial basis function (RBF) kernels below. Our aim was estimating the hyperparameter σ_g^2, which we call a variance component. Importantly, $\sigma_g^2/\mathrm{var}(l_i)$ is an estimator of genetic heritability, defined as the proportion of $\mathrm{var}(l_i)$ explained by genetics.

We simulated genetic data based on single nucleotide polymorphisms (SNPs), which can be encoded as 0/1/2, according to the number of minor alleles carried by an individual at a specific position in the genome. We first generated a minor allele frequency $f^j \sim \mathcal{U}(0.05, 0.5)$ for every SNP j, and then sampled a matrix $\boldsymbol{Z}$ of SNP, such that $Z_{ij} \sim \mathrm{Bin}(2, f^j)$. Finally, we standardized each column in the matrix $\boldsymbol{Z}$ by subtracting the mean and dividing by the standard deviation corresponding to its allele frequency.

To simulate unit-level ascertainment, we (1) generated a population of 1,000,000 individuals, where for every individual i we generated a vector of standardized genotypes $\boldsymbol{Z}_i \in \mathbb{R}^m$ as described above, and a vector $\boldsymbol{X}_i \sim \mathcal{N}(0, \boldsymbol{I}) \in \mathbb{R}^c$ representing additional standardized risk factors such as sex or age; (2) generated vectors $\boldsymbol{b} \sim \mathcal{N}\left(0, \sqrt{\sigma_g^2/m}\boldsymbol{I}\right)$ of random effects, and $\boldsymbol{\beta} \sim \mathcal{N}\left(0, \sqrt{\sigma_c^2/c}\boldsymbol{I}\right)$ of fixed effects; (3) assigned a latent variable $g_i = \boldsymbol{Z}_i^T\boldsymbol{b}$ and a liability $l_i = g_i + \boldsymbol{X}_i^T\boldsymbol{\beta} + \epsilon_i$ to every individual i, where $\epsilon_i \sim \mathcal{N}\left(0, \sqrt{1 - \sigma_c^2 - \sigma_g^2}\right)$ iid; (4) defined all individuals with l_i greater than the $1 - K$ quantile of the liability distribution as cases (i.e., $y_i = 1$), where K is the desired prevalence; and (5) selected a subset of $\frac{n}{2}$ cases and $\frac{n}{2}$ controls for the case-control study, where n is the desired study size. Unless stated otherwise, we used $m = 500$, $n = 500$, $K = 1\%$, $\sigma_g^2 = 0.25, \sigma_c^2 = 0.25$, and $c = 1$. We generated 100 datasets for each unique combination of evaluated settings.

We evaluated three likelihood-based methods: AEP, APL and plain EP, which does not account for ascertainment. We additionally evaluated the moment-based method PCGC, which is considered the state of the art approach for hyperparameter estimation in genetic case-control studies (Golan et al., 2014). We estimated hyperparameters in the likelihood-based methods via maximum likelihood, and in PCGC by finding the values that minimize the squared loss between the observed and expected values of $\mathrm{cov}(y_i, y_j)$ across all pairs of individuals i, j. In all settings, we first estimated the fixed effects via a novel ascertainment-

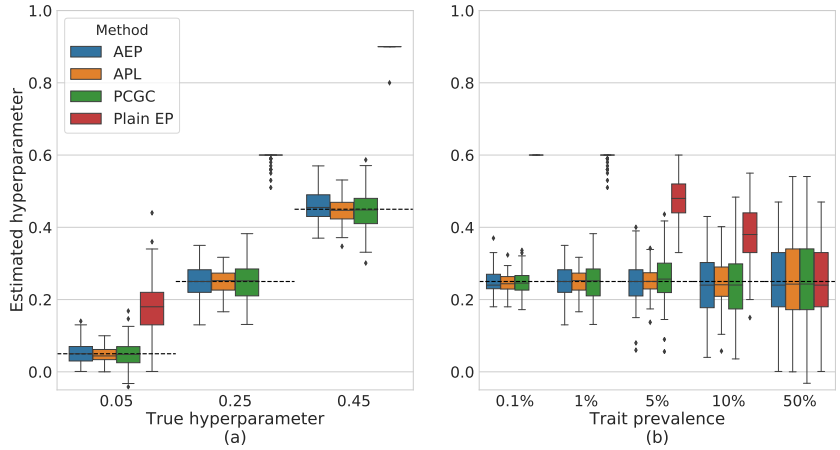


Figure 4: Evaluating hyperparameter estimation accuracy. Shown are box-plots depicting the estimates of each method across 100 different simulations, under data sets with an equal number of cases and controls, and a model with a single scale hyperparameter σ_g^2. The dashed horizontal lines represent the true underlying values of σ_g^2 used to generate the data. (a) AEP, APL and PCGC provide accurate estimates of σ_g^2 when the true trait prevalence (the prevalence of cases in the population) is 1%, for various values of σ_g^2, whereas plain EP is severely biased. (b) All methods except for plain EP accurately estimate σ_g^2 regardless of the underlying trait prevalence. Plain EP is accurate only when the prevalence is 50%, in which case there is no ascertainment.

aware generalized estimating equations (AGEE) approach that we developed, and then adjusted the affection cutoffs accordingly (Appendix C).

4.2. Simulation Studies: Estimating hyperparameters

Our first experiment evaluated variance component estimation accuracy. All methods except plain EP yielded empirically unbiased estimates, whereas plain EP was severely biased (Figure 4a). We also generated data under different prevalence values K and verified that all methods except plain EP remained accurate regardless of K, whereas plain EP was only accurate when $K = 0.5$, in which case there is no ascertainment (Figure 4b).

In the next experiment we examined sensitivity to sample size n and dimensionality m (corresponding to the number of rows and columns in the matrix $\boldsymbol{Z}$, respectively). We first verified that all methods became increasingly accurate with increasing sample size, but PCGC had a consistently larger sampling variance, because it uses a moment-based rather than a likelihood-based estimator (Figure 5a). We also observed that all methods became increasingly accurate with increasing dimensionality, but AEP was substantially

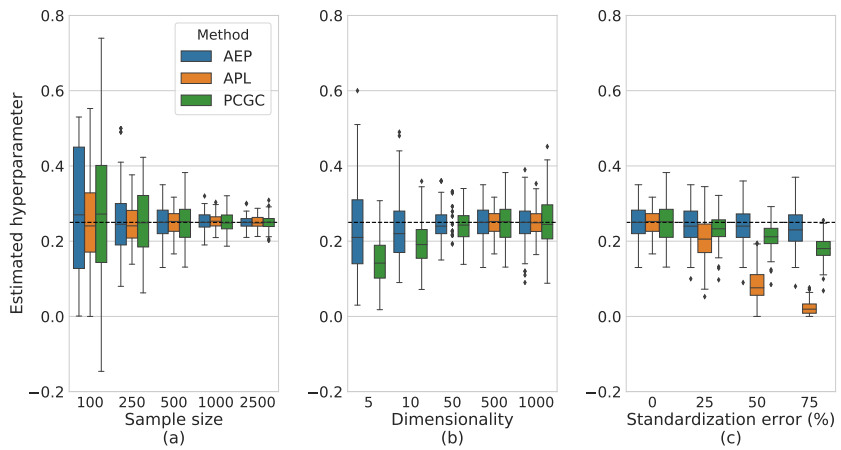


Figure 5: Investigating how hyperparameter estimation performance is affected by sample size, data dimensionality and modeling violations. (a) All the methods gain accuracy as the sample size increases. The sampling variance of PCGC is consistently larger, because it uses a moment-based rather than a likelihood-based estimator. (b) All the methods gain accuracy as dimensionality increases. AEP is substantially more accurate than the other methods in the presence of a small number of features, because the other two methods use a first-order Taylor expansion around $\boldsymbol{Z}\boldsymbol{Z}^T = \boldsymbol{I}$, which is less accurate in the presence of a small number of features. APL estimates for numbers <50 are equal to 1.0, and are omitted for clarity. (c) AEP is robust to feature standardization misspecification (see main text), whereas PCGC is moderately sensitive and APL is highly sensitive.

more accurate when $m < 50$ (Figure 5b). This is because the other two methods use a first-order Taylor expansion around $\boldsymbol{Z}\boldsymbol{Z}^T = \boldsymbol{I}$, which is less accurate when m is small.

We also examined robustness to modeling violations by introducing noise into the feature standardization procedure. We multiplied the estimated frequency of every binary variable j by $r_j \sim U\left(\frac{1}{1+e}, 1+e\right)$ before standardizing it, where $e \in [0,1]$ is the error magnitude, and used this value for estimation, but not for the true generative model. This noise model is motivated by GWAS, where genetic variants are often standardized according to (somewhat noisy) estimates of their population frequency rather than their sample frequency, to prevent bias due to ascertainment. AEP was highly robust to such modeling misspecification, whereas PCGC was moderately sensitive and APL was highly sensitive to such misspecification (Figure 5c). These experiments indicate that AEP is more reliable than the other methods under a wide variety of modeling assumptions, and is thus the method of choice for GP likelihood-based hyperparameter estimation in case-control studies.

Next, we examined estimation accuracy under non-linear kernels. We generated data with a scaled RBF kernel, $K_{ij} = \sigma_g^2 \exp\left(-\|\boldsymbol{Z}_i - \boldsymbol{Z}_j\|^2 / (2\gamma^2)\right)$, and estimated the hyperparameters $\boldsymbol{\theta} = \{\sigma_g^2, \gamma\}$. Our generative model used $\sigma_g^2 = 0.25$, $\gamma = 0.5$, and the same values as in the linear kernel simulations for all other parameters, except for restricting to m =10 normally-distributed features. This is because RBF kernels tend to overfit under a large m, yielding K_{ij} that is very close to either 0 or σ_g^2 regardless of $\boldsymbol{Z}$.

A technical challenge of the RBF experiments is that our simulations first generate true latent variables g_i for a population of 1M units. This requires computing a $1M \times 1M$ RBF covariance matrix $\boldsymbol{K}$ and sampling $\boldsymbol{g} = [g_1, \ldots, g_n]^T$ from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$, which is computationally intractable under a non-linear kernel. Instead we (1) generated a base population of 10K feature vectors $\boldsymbol{Z}_i$ and a corresponding $10K \times 10K$ RBF kernel matrix $\boldsymbol{K}_{10K}$; (2) sampled 10K g_i values from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_{10K})$; and (3) created a population of 1M units, such that each unit has a vector $\boldsymbol{Z}_i$ and a corresponding g_i value selected at random from the base population, along with uniquely generated values of $\boldsymbol{X}_i$ and ϵ_i. Afterwards we followed the same procedure as in the linear kernel simulations of (1) generating a liability $l_i = \boldsymbol{X}_i^T \boldsymbol{\beta} + g_i + \epsilon_i$ for each unit; (2) determining the liability cutoff according to the desired prevalence; and (3) sampling $n/2$ cases and $n/2$ controls. We modified PCGC and APL to ignore pairs of units with identical features in these experiments.

AEP was empirically unbiased in the RBF experiments, having an average estimation bias of -0.016 (stdev 0.068) for σ_g^2 and of 0.022 (stdev 0.076) for γ across 100 simulations. In contrast, APL, PCGC and plain EP were severely biased, with an average bias >0.2 in the estimation of σ_g^2 and >0.15 (in absolute value) in the estimation of γ. We verified that the bias was not due to the modified data generation scheme by repeating the same experiments with a linear kernel, wherein PCGC and APL were empirically unbiased. These results likely arise because PCGC and APL both use a first-order Taylor expansion around $K_{ij} = 0$, which may be less accurate in the presence of non-linear kernels. We conclude that PCGC and APL cannot be trivially modified to handle non-linear kernels, whereas AEP can be used in more general settings.

Finally, we examined the computational speed of the methods. PCGC and APL are very efficient compared to AEP, because they scale quadratically with the sample size whereas AEP scales cubically, like standard EP (Nickisch and Rasmussen, 2008). Nevertheless, AEP can perform maximum likelihood estimation in data sets with 3,000 units in less than two hours, and is thus applicable to solve reasonably sized real-world problems. AEP can potentially be scaled up using novel methods developed for GPs (see Discussion).

4.3. Real Data Analysis

We evaluated the accuracy of GP hyperparameter estimation in real data sets. To this end we estimated the heritability of seven complex disorders, having population prevalence between 0.1% and 6%, based on data sets with ∼3,700 individuals and ∼280,000 genetic variants from the Wellcome Trust 1 case-control consortium (Wellcome Trust Case Control Consortium et al., 2007). We performed stringent preprocessing to avoid confounding artifacts, as reported in our previous publication (Weissbrod et al., 2016). We used the same model as in the simulation studies. Specifically, we used a liability threshold model (i.e., a scaled probit likelihood) and a linear kernel, $K_{ij} = \theta \boldsymbol{Z}_i^T \boldsymbol{Z}_j$, where $\boldsymbol{Z}_i$,$\boldsymbol{Z}_j$ are 280,000-

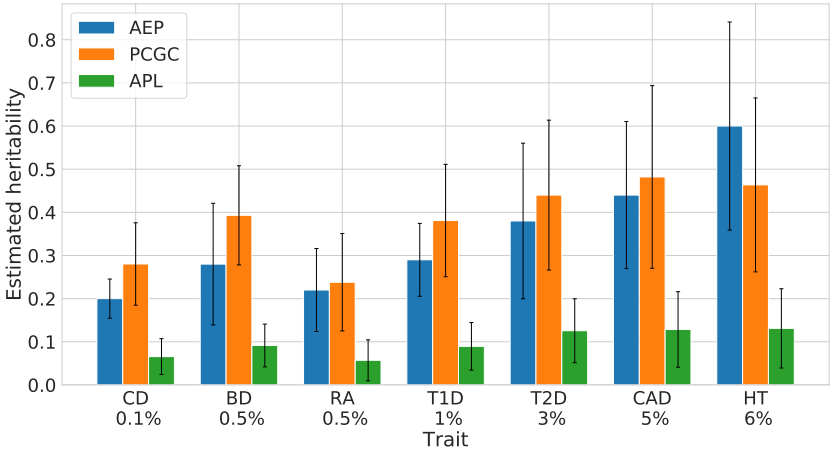


Figure 6: Shown are estimates of heritability (the proportion of liability variance explained by genetic factors) of seven complex disorders from (Wellcome Trust Case Control Consortium et al., 2007). The error bars are the standard deviation multiplied by 1.96, as estimated via jackknife. The disorders are Crohns disease (CD), rheumatoid arthritis (RA), bipolar disorder (BD), type 1 diabetes (T1D), type 2 diabetes (T2D), coronary artery disease (CAD) and hypertension (HT). The population prevalence of each trait is shown below its name. The estimates of AEP and PCGC are relatively concordant, whereas the APL estimates are significantly down-biased, in agreement with the modeling misspecification simulations.

dimensional vectors encoding the number of minor alleles in SNPs (standardized as in the simulation studies), and θ is a linear scaling factor. We associated sex with a fixed effect and estimated its effect via AGEE. Standard errors were computed via jackknife sampling.

The SNP heritability estimates of the investigated disorders lied in the range 20%-60% (Figure 6). There was a high degree of concordance between PCGC and AEP, whereas the APL estimates were substantially lower, consistent with the simulation studies. To further demonstrate the capabilities of AEP we estimated the posterior distribution of the GP latent variables g_i under both AEP and plain EP, using the standard EP approximation of the posterior distribution of latent variables (Rasmussen and Williams, 2006). Inferring the latent variables has a clinical utility as a measure of disease severity: individuals with larger values have a stronger genetic load of disease-inducing variants. The latent variables follow a clear bimodal distribution in all cases, but AEP provides a stronger separation between cases and controls (Figure 7). This result cannot be obtained with existing methods, because no existing likelihood-based GP approximation can model ascertainment.

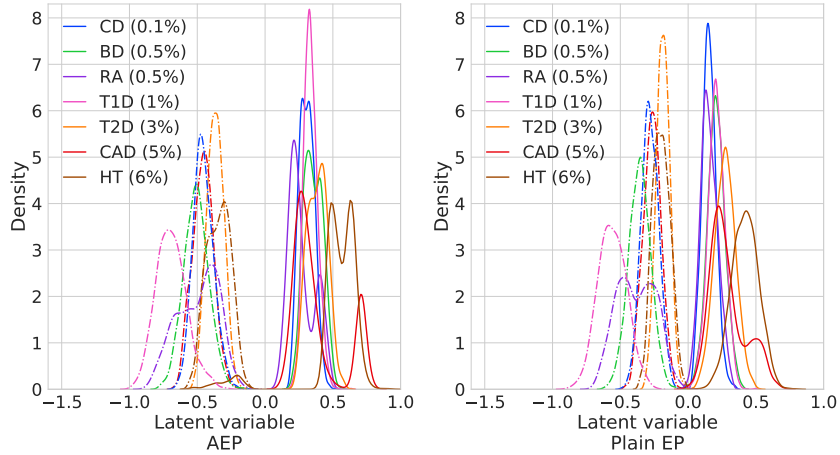


Figure 7: Inference of latent variables of complex genetic disorders. Shown is the distribution of the posterior mean of latent variables g_i provided by AEP (left) and plain EP (right), estimated via Gaussian kernel density estimation separately for controls (dashed lines) and cases (solid lines). The trait names are the same as in Figure 6, and prevalences are shown in parentheses. Individuals with larger posterior mean estimates carry a greater genetic load of disease-inducing variants. AEP provides a clearer separation between cases and controls by exploiting knowledge about the prevalence and sampling scheme. For both traits, the distribution variance increases with heritability and prevalence. Note that these distributions are not analogous to the ones in Figure 3 because they are based on posterior rather than marginal prior distributions.

5. Discussion

We presented several methods for inference of GP hyperparameters in settings with unit-level ascertainment, and a full-rank, non-sparse covariance structure. This was done by combining the ascertained likelihood framework with GPs and GLMMs, which form the statistical backbone of likelihood based analysis of non-iid data.

We proposed two approximate likelihood-based methods for the ascertained GP framework, AEP and APL, and empirically compared them with PCGC—the current state of the art method for estimating variance components in genetic case-control studies—which uses a moment-based rather than a likelihood-based estimator. APL is very computationally efficient but is sensitive to model misspecification. AEP, which is the most complex and empirically best approximation of maximum likelihood we proposed, is slower and is more technically complex than APL and PCGC, but is consistently more accurate than PCGC, and is less sensitive to modeling assumptions in our simulations. AEP additionally has

the advantage of providing a full probabilistic model with a well-defined likelihood, and it recovers standard EP as a special case under random ascertainment. On the other hand, PCGC has a principled underlying approximation, whereas APL and AEP are less well understood. Hence, the three methods are complementary in terms of their strengths and weaknesses, and we encourage future case-control studies to use multiple methods to gain a deeper understanding of high dimensional dependency structures.

The combination of unit-level ascertainment and a full-rank, non-sparse covariance structure is very common in statistical genetics (Golan et al., 2014), but is often encountered in other scientific domains, such as geostatistics and GP classification (Diggle et al., 1998; Chu et al., 2010; Ziegler et al., 2014; Young et al., 2013). Ascertained sampling is almost inevitable when studying rare phenomena, and the increasing dimensionality of studied data often necessitates the introduction of random rather than fixed effects, which in turn induce full-rank, non-sparse dependency structures. Additionally, it is often more convenient to perform dense sampling in a small number of clusters rather than collecting a large number of clusters (Bellamy et al., 2005; Zhang, 2004; Glidden and Vittinghoff, 2004), leading to non-sparse, full-rank dependency structures at the cluster level. Hence, we expect our work to be applicable in diverse scientific fields.

In this study we extended the well-known EP algorithm (Minka, 2001) to approximate GP likelihood. Another potential approach is MCMC sampling coupled with an integration scheme such as thermodynamic integration (Kuss and Rasmussen, 2005; Nickisch and Rasmussen, 2008; Gelman and Meng, 1998), but in our experience such approaches are too slow and complex for modern sized data sets. In recent years, Bayesian approaches have proven to be potential alternatives to likelihood based approaches in GPs (Ferkingstad and Rue, 2015). However, such approaches can be sensitive to the choice of prior distribution, and require prohibitively computationally expensive MCMC sampling. Several analytical approximations exist, but these are often inaccurate in the presence of binary data (Fong et al., 2010). The potential use of sampling-based approaches for inference in GPs under case-control ascertainment remains to be explored.

In recent years, genetic biobanks with hundreds of thousands of individuals have become available (Bycroft et al., 2018). AEP scales cubically with sample size and is thus not scalable to such datasets. GP approximation techniques from the machine learning community, such as mixture-of-experts models (Deisenroth and Ng, 2015), inducing points (Snelson and Ghahramani, 2006; Wilson and Nickisch, 2015; Gardner et al., 2018), random feature expansions (Rahimi and Recht, 2008; Le et al., 2013; Yang et al., 2015) and stochastic variational approximations (Hensman et al., 2013; Wilson et al., 2016; Cheng and Boots, 2017), or from the GWAS community (e.g. Loh et al. 2015) can potentially be used to scale up AEP to such large datasets.

A major challenge of non-sparse dependencies is that statistical theory is relatively undeveloped for this case. Specifically, assuming a study with r mutually independent clusters of m units, statistical theory is well developed for the asymptotic behavior $r/m \to \infty$, but is limited for $r/m \to 0$, which is our setting of interest (as $r=1$ when the covariance matrix is non-sparse). The statistical consistency of estimators in such cases has been established in limited settings, including GEEs (Xie and Yang, 2003), maximum penalized quasi likelihood (Bellamy et al., 2005), composite likelihood approximation (Heagerty and Lele, 1998), Laplace approximations (Shun and McCullagh, 1995), and specific geostatistical

models (Zhang and Zimmerman, 2005; Du et al., 2009). Several studies have established the statistical consistency of maximum likelihood estimators for linear mixed models (i.e., GPs with a linear kernel and a normal likelihood) in similar settings using random matrix theory (Bonnet et al., 2015; Jiang et al., 2016b; Dicker and Erdogdu, 2016), but to our knowledge such results have not been derived for GPs with non-normal likelihoods. We conclude that there is a major gap in statistical theory regarding $r/m \to 0$ asymptotics, representing questions of both theoretical and practical importance.

Several topics that remain unexplored in this work are GPs with more advanced kernels, outcome prediction and testing of fixed effects, for which several heuristic methods have been proposed in the statistical genetics literature (Hayeck et al., 2015; Weissbrod et al., 2015; Chen et al., 2016; Jiang et al., 2016a). Extending our approach to handle these topics is a potential avenue for future work.

Acknowledgments

This work was supported by grant 1804/16 from the Israel Science Foundation. This study makes use of data generated by the Wellcome Trust Case Control Consortium. A full list of the investigators who contributed to the generation of the data is available from www.wtccc.org.uk. Funding for the project was provided by the Wellcome Trust under award 076113. We thank Malka Gorfine for fruitful discussions.

Appendix A

Here we describe a fast Taylor expansion-based approximation to APL estimation of GPs with a probit likelihood and a scaling parameter σ^2. Denote $\text{cov}(g_i, g_j) = \rho\sigma_g^2$, where g_i is the latent variable of unit i and ρ depends on $\boldsymbol{Z}_i$,$\boldsymbol{Z}_j$, and on all the other kernel hyperparameters. The joint likelihood of each pair of units can be written as:

$$P(y_i = a, y_j = b \,|\, \boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{Z}_i, \boldsymbol{Z}_j, s_i = s_j = 1) = \frac{A_{ab}(\rho)}{B(\rho)} P(s_i = 1 \,|\, y_i) P(s_j = 1 \,|\, y_j), \quad (4)$$

where $A_{ab}(\rho) \triangleq P(y_i = a, y_j = b \,|\, \boldsymbol{X}_i, \boldsymbol{X}_j, \rho)$, $B(\rho) \triangleq P(s_i = s_j = 1 \,|\, \boldsymbol{X}_i, \boldsymbol{X}_j, \rho)$, and we omitted the hyperparameters $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ for brevity. Using the law of total probability, we can write: $B(\rho) = (s^1)^2 A_{11}(\rho) + s^1 s^0 (A_{10}(\rho) + A_{01}(\rho)) + (s^0)^2 A_{00}(\rho)$, where $s^t = P(s_i = 1 | y_i = t)$. Next, we explicitly evaluate these quantities at $\rho = 0$: $A_{ab}(0) = K_i^a (1 - K_i)^{1-a} K_j^b (1 - K_j)^{1-b}$, $B(0) = \left(s^0(1 - K_i) + s^1 K_i\right)\left(s^0(1 - K_j) + s^1 K_j\right)$, where $K_i = P(y_i = 1 | \boldsymbol{X}_i)$, and we omitted the dependence on $\boldsymbol{Z}_i$ because we assume that $g_i \sim \mathcal{N}(0, \sigma_g^2)$ marginally regardless of $\boldsymbol{Z}_i$. The above equations hold because y_i, y_j and s_i, s_j are independent given $\rho = 0$.

We next compute the partial derivatives of both expressions with respect to ρ at $\rho = 0$. Following (Golan et al., 2014), we have:

$$\begin{aligned}
\frac{d}{d\rho}A_{ab}(\rho)|_{\rho=0} &= \phi(t_i)\phi(t_j)\sigma^2(-1)^{a\neq b} \\
\frac{d}{d\rho}B(\rho)|_{\rho=0} &= (s^1)^2\frac{d}{d\rho}A_{11}(\rho)|_{\rho=0} + 2s^1s^0\frac{d}{d\rho}A_{a\neq b}(\rho)|_{\rho=0} + (s^0)^2\frac{d}{d\rho}A_{00}(\rho)|_{\rho=0} \\
&= \phi(t_i)\phi(t_j)\sigma^2\left((s^1)^2 + (s^0)^2 - 2s^1s^0\right),
\end{aligned}$$

where $\phi(\cdot)$ is the standard normal density, and $t_i = \Phi^{-1}(1-K) - \boldsymbol{X}_i^T\boldsymbol{\beta}$ is the liability cutoff for unit i, with $\Phi(\cdot)$ representing the standard normal cumulative density and K being the prevalence of cases in the population.

Finally, we plug in the above expressions into the Taylor expansion of Equation 4 at $\rho = 0$, which can be written as follows:

$$\frac{A_{ab}(\rho)}{B(\rho)}P(s_i\,|\,y_i)P(s_j\,|\,y_j) = \left(\frac{A'_{ab}(0)B(0) - B'(0)A_{ab}(0)}{B(0)^2}\rho + \mathcal{O}(\rho^2)\right)P(s_i\,|\,y_i)P(s_j\,|\,y_j).$$

Appendix B

Here we provide an informal analysis motivating the use of AEP. A formal analysis of AEP is difficult because there are relatively few theoretical guarantees for standard EP (Dehaene and Barthelmé, 2016, 2018). Instead, we state several assumptions and then provide an informal analysis under these assumptions. Throughout this Appendix, the notation $\boldsymbol{s}$ is a shorthand notation for $s_1 = \ldots = s_n = 1$, $\boldsymbol{u}_{-i}$ indicates the vector $\boldsymbol{u}$ with the ith entry removed, and we omit the dependence on $\boldsymbol{\beta}, \boldsymbol{\theta}$ for brevity.

The parametric form of the AEP site approximation

We first demonstrate that in order for AEP to approximate the ascertained likelihood $P(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ (up to a scaling factor) at its fixed point, the site function $t_i(g_i)$ needs to approximately take the following parametric form:

$$t_i(g_i) \approx \frac{P(y_i, s_i|g_i, \boldsymbol{X}_i)}{P(s_i|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}_{-i})}. \tag{5}$$

Note that this is different from the standard EP approximation $t_i(g_i) \approx P(y_i|X_i, g_i)$.

Our derivation requires an additional approximation:

Approximation 1

$$P(\boldsymbol{s}|\boldsymbol{X}, \boldsymbol{Z}) \approx \frac{1}{C(\boldsymbol{X}, \boldsymbol{Z})}\prod_i P(s_i|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}_{-i}), \quad \text{for some proportionality factor } C(\boldsymbol{X}, \boldsymbol{Z}).$$

Approximation 1 is motivated by the theory of composite likelihood estimators (Varin et al., 2011). Specifically, the composite maximum likelihood estimator of $\boldsymbol{\beta}, \boldsymbol{\theta}$ is asymptotically normally distributed around their true values under suitable regularity conditions (Cox and Reid, 2004). Since both the full and the composite maximum likelihood estimators

are asymptotically normal with the same mean, the composite likelihood is approximately proportional to the full likelihood around this mean, with the approximation accuracy depending on the ratio between their variances. This ratio depends on the ratio between the diagonal entries of the Fisher and the Godambe information matrices (Varin et al., 2011). Note that if Approximation 1 holds, this implies that it also holds when replacing $\boldsymbol{s}$ with $\boldsymbol{s}_{-j}$ and omitting j from the product. This property will be used in the derivations below.

Under Equation 5 and Approximation 1, the AEP likelihood $\int P(\boldsymbol{g}|\boldsymbol{Z})\prod_i t_i(g_i)d\boldsymbol{g}$ is approximately proportional to the ascertained likelihood $P(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$:

$$\begin{aligned}
\int P(\boldsymbol{g}|\boldsymbol{Z})\prod_i t_i(g_i)d\boldsymbol{g} &\overset{\text{Equation 5}}{\approx} \int P(\boldsymbol{g}|\boldsymbol{Z})\prod_i \frac{P(y_i, s_i|g_i, \boldsymbol{X}_i)}{P(s_i|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}_{-i})}d\boldsymbol{g} \\
&= \int \frac{P(\boldsymbol{g}|\boldsymbol{Z})\prod_i P(s_i|g_i, \boldsymbol{X}_i)}{\prod_i P(s_i|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}_{-i})}\prod_i P(y_i|s_i, g_i, \boldsymbol{X}_i)d\boldsymbol{g} \\
&\overset{\text{Approximation 1}}{\approx} C(\boldsymbol{X}, \boldsymbol{Z})\int \frac{P(\boldsymbol{g}|\boldsymbol{Z})P(\boldsymbol{s}|\boldsymbol{g}, \boldsymbol{X})}{P(\boldsymbol{s}|\boldsymbol{X}, \boldsymbol{Z})}P(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{X})d\boldsymbol{g} \\
&= C(\boldsymbol{X}, \boldsymbol{Z})\int P(\boldsymbol{g}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})P(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{g}, \boldsymbol{X})d\boldsymbol{g} \\
&= C(\boldsymbol{X}, \boldsymbol{Z}) \cdot P(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}).
\end{aligned}$$

The last two equalities use the fact that $\boldsymbol{y}$, $\boldsymbol{s}$ are conditionally independent of $\boldsymbol{Z}$ given $\boldsymbol{g}$. We conclude that if $t_i(g_i)$ approximately takes the form of Equation 5 then the hyperparameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ which maximize the AEP likelihood are approximately the maximum likelihood estimates.

Derivation of the AEP Step

Here we provide a heuristic motivation for the AEP step procedure. Recall from Section 3.2 that the AEP step consists of finding the unnormalized Gaussian $t_i(g_i)$ that optimizes the following approximation:

$$\int q_{-i}(g_i)t_i(g_i)dg_i \approx \frac{\int q_{-i}(g_i)P(y_i, s_i|g_i, \boldsymbol{X}_i)dg_i}{\int q_{-i}(g_i)P(s_i|g_i, \boldsymbol{X}_i)dg_i}, \tag{6}$$

where $q_{-i}(g_i) \propto \int P(\boldsymbol{g}|\boldsymbol{Z})\prod_{j\neq i} t_j(g_j)d\boldsymbol{g}_{-i}$, and the optimization is performed by matching the zeroth, first, and second derivatives of both functions with respect to μ_{-i}.

We first write down the natural analogue of the standard EP step objective for AEP. According to Equation 5, this objective finds the unnormalized Gaussian $t_i(g_i)$ that optimizes the approximation:

$$\int q_{-i}(g_i)t_i(g_i)dg_i \approx \int q_{-i}(g_i)\frac{P(y_i, s_i|g_i, \boldsymbol{X}_i)}{P(s_i|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}_{-i})}dg_i. \tag{7}$$

Unlike standard EP, we cannot minimize the GKL divergence between the functions in the integrals in Equation 7, because this will lead to the same solution $t_i(g_i)$ as in standard EP up to a scaling factor. To see this, note that the function in the rhs of Equation 7 can be

written as $q_{-i}(g_i)P(y_i|g_i,\boldsymbol{X}_i)W$, where $W = \frac{P(s_i|y_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}$ is constant with respect to g_i. Hence, minimizing the GKL divergence will lead to the same approximation as in standard EP, up to the scaling factor W.

Instead of minimizing the GKL divergence, we will approximate the rhs of Equation 7 as follows:

$$\int q_{-i}(g_i)\frac{P(y_i,s_i|g_i,\boldsymbol{X}_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}dg_i \approx \frac{\int q_{-i}(g_i)P(y_i,s_i|g_i,\boldsymbol{X}_i)dg_i}{\int q_{-i}(g_i)P(s_i|g_i,\boldsymbol{X}_i)dg_i}. \tag{8}$$

The AEP step in Equation 6 is obtained by equating the lhs of Equation 7 with the rhs of Equation 8.

It remain to derive Equation 8. Our derivation uses the following assumption:

Assumption 1 *Weak dependence between $\boldsymbol{y}_{-i}$ and s_i conditional on $\boldsymbol{X}$ and on $\boldsymbol{Z}$:*

$$P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}) \approx P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i}).$$

The derivation additionally uses the following two approximations, which we derive below by using Equation 5, Approximation 1 and Assumption 1:

Approximation 2 *At the fixed point we have: $q_{-i}(g_i) \approx P(g_i,\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})$.*

Approximation 3 $\int q_{-i}(g_i)\frac{P(y_i,s_i|g_i,\boldsymbol{X}_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}dg_i \approx P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},s_i)$.

We complete the derivation of Equation 8 by first using Approximation 3 to approximate the rhs of Equation 7 and the lhs of Equation 8 as $P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},s_i)$, and then using Approximation 2 and the graphical model structure (Figure 2) to approximate $P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},s_i)$ via $q_{-i}(g_i)$ as follows:

$$\begin{aligned}
P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},s_i) &= \frac{P(y_i,s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})} \\
&= \frac{\int P(g_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})P(y_i,s_i|\boldsymbol{X}_i,g_i)dg_i}{\int P(g_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})P(s_i|\boldsymbol{X}_i,g_i)dg_i} \\
&\approx \frac{\int q_{-i}(g_i)P(y_i,s_i|\boldsymbol{X}_i,g_i)dg_i}{\int q_{-i}(g_i)P(s_i|\boldsymbol{X}_i,g_i)dg_i}.
\end{aligned}$$

This completes the derivation.

Derivation of Approximations 2–3

We now provide heuristic derivations of Approximations 2–3.

Approximation 2 $q_{-i}(g_i) \approx P(g_i,\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})$.

Our derivation consists of two stages. First, we define the unnormalized cavity distribution $q^*_{-i}(g_i) \triangleq \int P(\boldsymbol{g}|\boldsymbol{Z})\prod_{j\neq i}t_j(g_j)d\boldsymbol{g}_{-i}$, and show that $q^*_{-i}(g_i) \approx C(\boldsymbol{X},\boldsymbol{Z})\cdot P(g_i,\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})$:

$$\begin{aligned}
q^*_{-i}(g_i) &\triangleq \int P(\boldsymbol{g}|\boldsymbol{Z})\prod_{j\neq i}t_j(g_j)d\boldsymbol{g}_{-i} \\
&\overset{\text{Equation 5}}{\approx} \int P(\boldsymbol{g}|\boldsymbol{Z})\prod_{j\neq i}\frac{P(y_j,s_j|g_j,\boldsymbol{X}_j)}{P(s_j|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-j})}d\boldsymbol{g}_{-i} \\
&\overset{\text{Approximation 1}}{\approx} C(\boldsymbol{X},\boldsymbol{Z})\int \frac{P(\boldsymbol{g}|\boldsymbol{Z})}{P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}\prod_{j\neq i}P(y_j,s_j|g_i,\boldsymbol{X}_j)d\boldsymbol{g}_{-i} \\
&= C(\boldsymbol{X},\boldsymbol{Z})\int \frac{P(\boldsymbol{g}_{-i}|\boldsymbol{Z})P(g_i|\boldsymbol{g}_{-i},\boldsymbol{Z})}{P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}P(\boldsymbol{y}_{-i},\boldsymbol{s}_{-i}|\boldsymbol{g}_{-i},\boldsymbol{X}_{-i})d\boldsymbol{g}_{-i} \\
&\overset{\text{rearrangement}}{=} C(\boldsymbol{X},\boldsymbol{Z})\int \frac{P(\boldsymbol{g}_{-i}|\boldsymbol{Z})P(\boldsymbol{s}_{-i}|\boldsymbol{g}_{-i},\boldsymbol{X}_{-i})}{P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}P(\boldsymbol{y}_{-i},|\boldsymbol{g}_{-i},\boldsymbol{s}_{-i},\boldsymbol{X}_{-i})P(g_i|\boldsymbol{g}_{-i},\boldsymbol{Z})d\boldsymbol{g}_{-i} \\
&\overset{\text{Bayes rule}}{=} C(\boldsymbol{X},\boldsymbol{Z})\int P(\boldsymbol{g}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})P(\boldsymbol{y}_{-i},|\boldsymbol{g}_{-i},\boldsymbol{s}_{-i},\boldsymbol{X}_{-i})P(g_i|\boldsymbol{g}_{-i},\boldsymbol{Z})d\boldsymbol{g}_{-i} \\
&\overset{\text{graphical model}}{=} C(\boldsymbol{X},\boldsymbol{Z})\int P(\boldsymbol{g}_{-i},\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})P(g_i|\boldsymbol{g}_{-i},\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})d\boldsymbol{g}_{-i} \\
&= C(\boldsymbol{X},\boldsymbol{Z})\int P(\boldsymbol{g},\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})d\boldsymbol{g}_{-i} \\
&= C(\boldsymbol{X},\boldsymbol{Z})\cdot P(g_i,\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})
\end{aligned}$$

Next, we note that since $q_{-i}(g_i) \triangleq \frac{q^*_{-i}(g_i)}{\int q^*_{-i}(g'_i)dg'_i}$ is a normalized distribution over g_i, we have $q_{-i}(g_i) \approx P(g_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},\boldsymbol{s}_{-i})$. Finally, we note that g_i is conditionally independent of $\boldsymbol{s}_{-i}$ given $\boldsymbol{y}_{-i}$ due to the graphical model structure, yielding $q_{-i}(g_i) \approx P(g_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})$.

Approximation 3 $\int q_{-i}(g_i)\frac{P(y_i,s_i|g_i,\boldsymbol{X}_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}dg_i \approx P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},s_i)$.

First, we invoke Approximation 2 and the graphical model structure to obtain the following approximation:

$$\begin{aligned}
\int q_{-i}(g_i)\frac{P(y_i,s_i|g_i,\boldsymbol{X}_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}dg_i &\overset{\text{Approximation 2}}{\approx} \int P(g_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})\frac{P(y_i,s_i|g_i,\boldsymbol{X}_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}dg_i \\
&= \int \frac{P(g_i|\boldsymbol{X},\boldsymbol{Z})P(\boldsymbol{y}_{-i}|g_i,\boldsymbol{X},\boldsymbol{Z})}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z})}\frac{P(y_i,s_i|g_i,\boldsymbol{X}_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}dg_i \\
&= \int \frac{P(g_i|\boldsymbol{X},\boldsymbol{Z})P(\boldsymbol{y}_{-i}|g_i,\boldsymbol{X},\boldsymbol{Z})}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z})}\frac{P(y_i|g_i,\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i})P(s_i|y_i)}{P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}dg_i \\
&= \frac{\int P(g_i|\boldsymbol{X},\boldsymbol{Z})P(\boldsymbol{y}|g_i,\boldsymbol{X},\boldsymbol{Z})dg_iP(s_i|y_i)}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z})P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})} \\
&= \frac{P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z})P(s_i|y_i)}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z})P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}.
\end{aligned} \tag{9}$$

Next, we multiply the rhs of Equation 9 by $\frac{P(\boldsymbol{s}_{-i}|\boldsymbol{y}_{-i})P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}{P(\boldsymbol{s}_{-i}|\boldsymbol{y}_{-i})P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}$ and invoke Assumption 1:

$$\begin{aligned}
\frac{P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z})P(s_i|y_i)}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z})P(s_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})}\frac{P(\boldsymbol{s}_{-i}|\boldsymbol{y}_{-i})P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}{P(\boldsymbol{s}_{-i}|\boldsymbol{y}_{-i})P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})} &= \frac{P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z})}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z})}\frac{P(\boldsymbol{s}|\boldsymbol{y})P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}{P(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{Z})P(\boldsymbol{s}_{-i}|\boldsymbol{y}_{-i})} \\
&= \frac{P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z})\frac{P(\boldsymbol{s}|\boldsymbol{y})}{P(\boldsymbol{s}|\boldsymbol{X},\boldsymbol{Z})}}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z})\frac{P(\boldsymbol{s}_{-i}|\boldsymbol{y}_{-i})}{P(\boldsymbol{s}_{-i}|\boldsymbol{X},\boldsymbol{Z})}} \\
&= \frac{P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s})}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})} \\
&= \frac{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s})P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},\boldsymbol{s})}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})} \\
&\overset{\text{Assumption 1}}{\approx} \frac{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},\boldsymbol{s})}{P(\boldsymbol{y}_{-i}|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}_{-i})} \\
&= P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},\boldsymbol{s}) \\
&= P(y_i|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{y}_{-i},s_i)
\end{aligned}$$

Appendix C

Here we describe our novel development of ascertained generalized estimating equations (AGEE). GEEs are extensions of generalized linear models that can estimate fixed effects while accounting for dependencies without requiring a probabilistic model (Liang and Zeger, 1993). GEEs require a correct specification of the mean of the outcome conditional on the features, $\mu_i = E\left[y_i|\boldsymbol{X}_i,\boldsymbol{\beta}\right]$, and a (possibly misspecified) working covariance matrix of the outcomes, denoted as $\boldsymbol{\Omega}\left(\theta^{\Omega}\right)$ and parameterized by θ^{Ω}. Given these, $\boldsymbol{\beta}$ is estimated by solving the estimating equation $\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\beta}}\boldsymbol{\Omega}\left(\theta^{\Omega}\right)^{-1}\left(\boldsymbol{y}-\boldsymbol{\mu}\left(\boldsymbol{\beta}\right)\right)=0$. GEEs yield consistent estimates of $\boldsymbol{\beta}$ and its sampling variance even if the covariance structure is misspecified and is non-sparse (Xie and Yang, 2003).

GEEs can naturally be adapted to case-control settings by using the ascertained conditional mean function $E\left[y_i|\boldsymbol{X}_i,s_i=1,\boldsymbol{\beta}\right]=P\left(y_i=1\,|\,\boldsymbol{X}_i,s_i=1,\boldsymbol{\beta}\right)$. We now show how the GEE fixed effect estimates can be plugged into GPs. In the general case it is not possible to reconcile fixed effect estimates of GEEs and GPs, because GEEs assume that the conditional mean of the outcome is affected only by the fixed effects, whereas GPs assume that it is affected by both the fixed effects and the GP latent variable. Fortunately, the probit likelihood provides a convenient way to reconcile the two approaches. Denote $\boldsymbol{\beta}_{\text{GEE}}$ and $\boldsymbol{\beta}_{\text{GP}}$ as the vectors of fixed effects used by GEE and GP, respectively. When using a probit likelihood, the GEE conditional mean is given by $\Phi(\boldsymbol{X}_i^T\boldsymbol{\beta}_{\text{GEE}})$, where $\Phi(\cdot)$ is the standard normal cumulative density. In contrast, the GP conditional mean is given by $\Phi\left(\frac{\boldsymbol{X}_i^T\boldsymbol{\beta}_{\text{GP}}}{(\text{var}(g_i)+1)^{1/2}}\right)$. If $\text{var}(g_i)$ is constant for every unit i (which corresponds to a constant value on the diagonal of the covariance matrix of $\boldsymbol{g}$), the two approaches can be reconciled by defining $\boldsymbol{\beta}_{\text{GP}}=\boldsymbol{\beta}_{\text{GEE}}\left(\text{var}(g_i)+1\right)^{1/2}$. In practice, the diagonal of the covariance matrix of $\boldsymbol{g}$ is often exactly or almost exactly constant, which enables exploiting the above relation. Therefore, we can use the GEE estimates in a GP by setting $\boldsymbol{\beta}_{\text{GP}}=\boldsymbol{\beta}_{\text{GEE}}\left(\text{var}(g_i)+1\right)^{1/2}$.

Our implementation of AGEE closely followed that of (Jiang et al., 2016a), with a suitable modification of the conditional mean to encode ascertainment, as described above.

References

Scarlett L. Bellamy, Yi Li, Xihong Lin, and Louise M. Ryan. Quantifying PQL bias in estimating cluster-level covariate effects in generalized linear mixed models for group-randomized trials. *Stat. Sin.*, 15(4):1015–1032, 2005.

Benjamin M. Bolker, Mollie E. Brooks, Connie J. Clark, Shane W. Geange, John R. Poulsen, M. Henry H. Stevens, and Jada-Simone S. White. Generalized linear mixed models: a practical guide for ecology and evolution. *Trends Ecol. Evol.*, 24(3):127–35, 2009.

Anna Bonnet, Elisabeth Gassiat, and Cline Lvy-Leduc. Heritability estimation in high dimensional sparse linear mixed models. *Electron. J. Stat.*, 9(2):2099–2129, 2015.

Norman E. Breslow. Statistics in epidemiology: the case-control study. *J. Am. Stat. Assoc.*, 91(433):14–28, 1996.

Norman E. Breslow and David G. Clayton. Approximate inference in generalized linear mixed models. *J. Am. Stat. Assoc.*, 88(421):9–25, 1993.

Paul R. Burton, Katrina J. Tiller, Lyle C. Gurrin, William OCM Cookson, A. William Musk, and Lyle J. Palmer. Genetic variance components analysis for binary phenotypes using generalized linear mixed models (GLMMs) and Gibbs sampling. *Genet. Epidemiol.*, 17(2):118–140, 1999.

Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared OConnell, et al. The uk biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203, 2018.

Han Chen, Chaolong Wang, Matthew P. Conomos, Adrienne M. Stilp, Zilin Li, Tamar Sofer, Adam A. Szpiro, Wei Chen, John M. Brehm, Juan C. Celedón, et al. Control for population structure and relatedness for binary traits in genetic association studies via logistic mixed models. *Am. J. Hum. Genet.*, 98(4):653–666, 2016.

Ching-An Cheng and Byron Boots. Variational inference for Gaussian process models with linear complexity. In *Advances in Neural Information Processing Systems*, pages 5184–5194, 2017.

Carlton Chu, Peter Bandettini, John Ashburner, Andre Marquand, and Stefan Kloeppel. Classification of neurodegenerative diseases using Gaussian process classification with automatic feature determination. In *Workshop on brain decoding: Pattern recognition challenges in neuroimaging (WBD), 2010*, pages 17–20. IEEE, 2010.

David R. Cox and Nancy Reid. A note on pseudolikelihood constructed from marginal densities. *Biometrika*, 91(3):729–737, 2004.

Guillaume Dehaene and Simon Barthelmé. Bounding errors of Expectation-Propagation. In *Advances in Neural Information Processing Systems*, pages 244–252, 2016.

Guillaume Dehaene and Simon Barthelmé. Expectation Propagation in the large data limit. *J. R. Stat. Soc. B*, 80(1):199–217, 2018.

Marc Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *International Conference on Machine Learning*, volume 37, pages 1481–1490, 07–09 Jul 2015.

Everett R. Dempster and I. Michael Lerner. Heritability of threshold characters. *Genetics*, 35(2):212–36, 1950.

Lee H. Dicker and Murat A. Erdogdu. Maximum likelihood for variance estimation in high-dimensional linear models. In *International Conference on Artificial Intelligence and Statistics*, pages 159–167, 2016.

Peter J. Diggle, Jonathan A. Tawn, and Rana Moyeed. Model-based geostatistics. *J. R. Stat. Soc. C*, 47(3):299–350, 1998.

Juan Du, Hao Zhang, and Vidyadhar Mandrekar. Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *Ann. Stat.*, 37(6A):3330–3361, 2009.

Georg B. Ehret, Patricia B. Munroe, Kenneth M. Rice, Murielle Bochud, Andrew D. Johnson, Daniel I. Chasman, Albert V. Smith, Martin D. Tobin, Germaine C. Verwoert, Shih-Jen Hwang, et al. Genetic variants in novel pathways influence blood pressure and cardiovascular disease risk. *Nature*, 478(7367):103–109, 2011.

Michael P. Epstein, Xihong Lin, and Michael Boehnke. Ascertainment-adjusted parameter estimates revisited. *Am. J. Hum. Genet.*, 70(4):886–95, 2002.

Ludwig Fahrmeir and Gerhard Tutz. *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer Series in Statistics. Springer New York, Berlin, 2nd edition, 2001. ISBN 978-0-387-95187-4.

Egil Ferkingstad and Håvard Rue. Improving the INLA approach for approximate Bayesian inference for latent Gaussian models. *Electron. J. Stat.*, 9(2):2706–2731, 2015.

Youyi Fong, Håvard Rue, and Jon Wakefield. Bayesian inference for generalized linear mixed models. *Biostatistics*, 11(3):397–412, 2010.

Jacob Gardner, Geoff Pleiss, Ruihan Wu, Kilian Weinberger, and Andrew Wilson. Product kernel interpolation for scalable Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1407–1416, 2018.

Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Stat. Sci.*, 13(2):163–185, 1998.

David V. Glidden and Kung-Yee Liang. Ascertainment adjustment in complex diseases. *Genet. Epidemiol.*, 23(3):201–208, 2002.

David V. Glidden and Eric Vittinghoff. Modelling clustered survival data from multicentre clinical trials. *Stat. Med.*, 23(3):369–88, 2004.

David Golan and Saharon Rosset. Effective genetic-risk prediction using mixed models. *Am. J. Hum. Genet.*, 95(4):383–93, 2014.

David Golan, Eric Lander, and Saharon Rosset. Measuring missing heritability: Inferring the contribution of common variants. *Proc. Natl. Acad. Sci. USA*, 111(49):E5272–81, 2014.

Tristan J. Hayeck, Noah A. Zaitlen, Po-Ru Loh, Bjarni Vilhjalmsson, Samuela Pollack, Alexander Gusev, Jian Yang, Guo-Bo Chen, Michael E. Goddard, Peter M. Visscher, et al. Mixed model with correction for case-control ascertainment increases association power. *Am. J. Hum. Genet.*, 96(5):720–730, 2015.

Patrick J Heagerty and Subhash R Lele. A composite likelihood approach to binary spatial data. *J. Am. Stat. Assoc.*, 93(443):1099–1111, 1998.

Leonhard Held, Isabel Natrio, Sarah Elaine Fenton, Håvard Rue, and Nikolaus Becker. Towards joint disease mapping. *Stat. Methods Med. Res.*, 14(1):61–82, 2005.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, 2013.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. MCMC for variationally sparse Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1648–1656, 2015.

David A Hsieh, Charles F Manski, and Daniel McFadden. Estimation of response probabilities from augmented retrospective observations. *J. Am. Stat. Assoc.*, 80(391):651–662, 1985.

Duo Jiang, Joelle Mbatchou, and Mary Sara McPeek. Retrospective association analysis of binary traits: overcoming some limitations of the additive polygenic model. *Hum. Hered.*, 80(4):187–195, 2015.

Duo Jiang, Sheng Zhong, and Mary Sara McPeek. Retrospective binary-trait association test elucidates genetic architecture of Crohn disease. *Am. J. Hum. Genet.*, 98(2):243–255, 2016a.

Jiming Jiang, Cong Li, Debashis Paul, Can Yang, and Hongyu Zhao. On high-dimensional misspecified mixed model analysis in genome-wide association study. *Ann. Stat.*, 44(5): 2127–2160, 2016b.

Julia E Kelsall and Peter J. Diggle. Spatial variation in risk of disease: a nonparametric binary regression approach. *J. Royal Stat. Soc. C*, 47(4):559–573, 1998.

H. König. *Eigenvalue Distribution of Compact Operators*. Operator Theory: Advances and Applications. Birkhäuser Basel, 2013. ISBN 9783034862783.

Holly J Kramer, Adrienne M. Stilp, Cathy C. Laurie, Alex P. Reiner, James Lash, Martha L. Daviglus, Sylvia E. Rosas, Ana C. Ricardo, Bamidele O. Tayo, Michael F. Flessner, et al. African ancestry–specific alleles and kidney disease risk in Hispanics/Latinos. *J. Am. Soc. Nephrol.*, 28(3):915–922, 2017.

Malte Kuss and Carl Edward Rasmussen. Assessing approximate inference for binary Gaussian process classification. *J. Mach. Learn. Res.*, 6:1679–1704, 2005.

Quoc Le, Tamas Sarlos, and Alexander Smola. Fastfood - computing Hilbert space expansions in loglinear time. In *International Conference on Machine Learning*, pages 244–252, 2013.

Sang Hong Lee, Naomi R. Wray, Michael E. Goddard, and Peter M. Visscher. Estimating missing heritability for disease from genome-wide association studies. *Am. J. Hum. Genet.*, 88(3):294–305, 2011.

Kung-Yee Liang and Scott L Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986.

Kung-Yee Liang and Scott L Zeger. Regression analysis for correlated data. *Annu. Rev. Public Health*, 14(1):43–68, 1993.

Po-Ru Loh, Gaurav Bhatia, Alexander Gusev, Hilary K Finucane, Brendan K Bulik-Sullivan, Samuela J Pollack, Teresa R de Candia, Sang Hong Lee, Naomi R Wray, Kenneth S Kendler, et al. Contrasting genetic architectures of schizophrenia and other complex diseases using fast variance-components analysis. *Nat. Genet.*, 47(12):1385–1392, 2015.

Charles F. Manski. *Alternative estimators and sample designs for discrete choice analysis*. The MIT Press, 1981.

Charles E McCulloch, Shayle R Searle, and John M Neuhaus. *Generalized, Linear, and Mixed Models*. Wiley Series in Probability and Statistics, 2nd edition, 2008. ISBN 0-470-01181-5.

Thomas P Minka. Expectation Propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence*, pages 362–369, 2001. ISBN 1-55860-800-1.

J. M. Neuhaus, Alastair J. Scott, Chris J. Wild, Y. Jiang, C. E. McCulloch, and R. Boylan. Likelihood-based analysis of longitudinal data from outcome-related sampling designs. *Biometrics*, 70(1):44–52, 2014.

John M. Neuhaus and Nicholas P. Jewell. The effect of retrospective sampling on binary regression models for clustered data. *Biometrics*, 46(4):977–990, 1990.

John M. Neuhaus, Alastair H. Scott, and Chris J. Wild. The analysis of retrospective family studies. *Biometrika*, 89(1):23–37, 2002.

John M. Neuhaus, Alastair J. Scott, and Chris J. Wild. Family-specific approaches to the analysis of case-control family data. *Biometrics*, 62(2):488–94, 2006.

Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary Gaussian process classification. *J. Mach. Learn. Res.*, 9:2035–2078, 2008.

Yukinori Okada, Di Wu, Gosia Trynka, Towfique Raj, Chikashi Terao, Katsunori Ikari, Yuta Kochi, Koichiro Ohmura, Akari Suzuki, Shinji Yoshida, et al. Genetics of rheumatoid arthritis contributes to biology and drug discovery. *Nature*, 506(7488):376–381, 2014.

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural Comput*, 21(3):786–92, 2009.

Manfred Opper and Ole Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Comput*, 12(11):2655–2684, 2000.

Dirk Pfeiffer. *Spatial Analysis in Epidemiology*. Oxford University Press, 2008.

Alkes L. Price, Chris C. A. Spencer, and Peter Donnelly. Progress and promise in understanding the genetic basis of common diseases. *Proc Biol Sci*, 282(1821), 2015.

Qibin Qi, Adrienne M Stilp, Tamar Sofer, Jee-Young Moon, Bertha Hidalgo, Adam A Szpiro, Tao Wang, Maggie CY Ng, Xiuqing Guo, Yii-Der Ida Chen, et al. Genetics of type 2 diabetes in US Hispanic/Latino individuals: results from the Hispanic Community Health Study/Study of Latinos (HCHS/SOL). *Diabetes*, 66(5):1419–1425, 2017.

Yuan Alan Qi, Thomas P Minka, Rosalind W Picard, and Zoubin Ghahramani. Predictive automatic relevance determination by Expectation Propagation. In *International Conference on Machine Learning*, page 85, 2004.

Sophia Rabe-Hesketh, Anders Skrondal, and Andrew Pickles. Generalized multilevel structural equation modeling. *Psychometrika*, 69(2):167–190, 2004.

Sophia Rabe-Hesketh, Anders Skrondal, and Andrew Pickles. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *J. Econom.*, 128(2):301–323, 2005.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 978-0-262-18253-9.

Stephen W. Raudenbush, Meng-Li Yang, and Matheos Yosef. Maximum likelihood for generalized linear models with nested random effects via high-order, multivariate Laplace approximation. *J. Comp. Graph. Stat.*, 9(1):141–157, 2000.

Didier Renard, Geert Molenberghs, and Helena Geys. A pairwise likelihood approach to estimation in multilevel probit models. *Comput. Stat. Data Anal.*, 44(4):649–667, 2004.

Stephan Ripke, Benjamin M Neale, Aiden Corvin, James TR Walters, Kai-How Farh, Peter A Holmans, Phil Lee, Brendan Bulik-Sullivan, David A Collier, Hailiang Huang, et al. Biological insights from 108 schizophrenia-associated genetic loci. *Nature*, 511(7510):421–427, 2014.

Stephen Sawcer, Garrett Hellenthal, Matti Pirinen, Chris CA Spencer, Nikolaos A Patsopoulos, Loukas Moutsianas, Alexander Dilthey, Zhan Su, Colin Freeman, Sarah E Hunt, et al. Genetic risk and a primary role for cell-mediated immune mechanisms in multiple sclerosis. *Nature*, 476(7359):214–9, 2011.

Alastair J Scott and Chris J Wild. Fitting regression models to case-control data by maximum likelihood. *Biometrika*, 84(1):57–71, 1997.

Alastair J. Scott and Chris J. Wild. Maximum likelihood for generalised case-control studies. *J. Stat. Plan. Inference*, 96(1):3–27, 2001.

Matthias Seeger. Low rank updates for the cholesky decomposition. Technical report, 2004.

Matthias Seeger. Expectation propagation for exponential families. Technical report, 2005.

Zhenming Shun and Peter McCullagh. Laplace approximation of high dimensional integrals. *J. R. Stat. Soc. B*, 57(4):749–760, 1995.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2006.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *J. Am. Stat. Assoc.*, 81(393):82–86, 1986.

Cristiano Varin, Nancy Reid, and David Firth. An overview of composite likelihood methods. *Stat. Sin.*, 21(1):5–42, 2011.

Aki Vehtari, Tommi Mononen, Ville Tolvanen, Tuomas Sivula, and Ole Winther. Bayesian leave-one-out cross-validation approximations for Gaussian latent variable models. *J. Mach. Learn. Res.*, 17(1):3581–3618, 2016.

Peter M. Visscher, Naomi R. Wray, Qian Zhang, Pamela Sklar, Mark I. McCarthy, Matthew A. Brown, and Jian Yang. 10 years of GWAS discovery: biology, function, and translation. *Am. J. Hum. Genet.*, 101(1):5–22, 2017.

Omer Weissbrod, Christoph Lippert, Dan Geiger, and David Heckerman. Accurate liability estimation improves power in ascertained case-control studies. *Nat. Methods*, 12(4):332–4, 2015.

Omer Weissbrod, Dan Geiger, and Saharon Rosset. Multikernel linear mixed models for complex phenotype prediction. *Genome Res.*, 26(7):969–79, 2016.

Omer Weissbrod, Jonathan Flint, and Saharon Rosset. Estimating SNP-based heritability and genetic correlation in case-control studies directly and with summary statistics. *Am. J. Hum. Genet.*, 103(1):89–99, 2018.

Wellcome Trust Case Control Consortium et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, 2007.

Chris J. Wild. Fitting prospective regression models to case-control data. *Biometrika*, 78 (4):705–717, 1991.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

Andrew G Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016.

Russ Wolfinger and Michael O'connell. Generalized linear mixed models: a pseudo-likelihood approach. *J. Stat. Comput. Simul.*, 48(3-4):233–243, 1993.

Minge Xie and Yaning Yang. Asymptotics for generalized estimating equations with large cluster sizes. *Ann. Stat.*, 31(1):310–347, 2003.

Jian Yang, Beben Benyamin, Brian P. McEvoy, Scott Gordon, Anjali K. Henders, Dale R. Nyholt, Pamela A. Madden, Andrew C. Heath, Nicholas G Martin, Grant W. Montgomery, et al. Common SNPs explain a large proportion of the heritability for human height. *Nat. Genet.*, 42(7):565–569, 2010.

Jian Yang, Noah A. Zaitlen, Michael E. Goddard, Peter M. Visscher, and Alkes L. Price. Advantages and pitfalls in the application of mixed-model association methods. *Nat. Genet.*, 46(2):100–106, 2014.

Zichao Yang, Andrew Wilson, Alex Smola, and Le Song. A la Carte – learning fast kernels. In *International Conference on Artificial Intelligence and Statistics*, pages 1098–1106, 2015.

Jonathan Young, Marc Modat, Manuel J. Cardoso, Alex Mendelson, Dave Cash, Sebastien Ourselin, the Alzheimer's Disease Neuroimaging Initiative, et al. Accurate multimodal probabilistic prediction of conversion to Alzheimer's disease in patients with mild cognitive impairment. *Neuroimage Clin.*, 2:735–745, 2013.

Hao Zhang. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *J. Am. Stat. Assoc.*, 99(465):250–261, 2004.

Hao Zhang and Dale L. Zimmerman. Towards reconciling two asymptotic frameworks in spatial statistics. *Biometrika*, 92(4):921–936, 2005.

Xiang Zhou, Peter Carbonetto, and Matthew Stephens. Polygenic modeling with Bayesian sparse linear mixed models. *PLoS Genet.*, 9(2):e1003264, 2013.

Gabriel Ziegler, Gerard R. Ridgway, Robert Dahnke, Christian Gaser, the Alzheimer's Disease Neuroimaging Initiative , et al. Individualized Gaussian process-based prediction and detection of local and global gray matter abnormalities in elderly subjects. *Neuroimage*, 97:333–348, 2014.

Journal of Machine Learning Research 20 (2019) 1-30 Submitted 5/18; Published 3/19

Learning Unfaithful K-separable Gaussian Graphical Models

De Wen Soh SOHDW@IHPC.A-STAR.EDU.SG
Institute of High Performance Computing
1 Fusionopolis Way, #16-16 Connexis
Singapore, 138632
Sekhar Tatikonda SEKHAR.TATIKONDA@YALE.EDU
Department of Electrical Engineering
Yale University
New Haven, CT 06511, USA

Editor: Alexander Ihler

Abstract

The global Markov property for Gaussian graphical models ensures graph separation implies conditional independence. Specifically if a node set S graph separates nodes u and v then X_u is conditionally independent of X_v given X_S. The opposite direction need not be true, that is, $X_u \perp X_v \mid X_S$ need not imply S is a node separator of u and v. When it does, the relation $X_u \perp X_v \mid X_S$ is called faithful. In this paper we provide a characterization of faithful relations and then provide an algorithm to test faithfulness based only on knowledge of other conditional relations of the form $X_i \perp X_j \mid X_S$. We study two classes of separable Gaussian graphical models, namely, weakly K-separable and strongly K-separable Gaussian graphical models. Using the above test for faithfulness, we introduce algorithms to learn the topologies of weakly K-separable and strongly K-separable Gaussian graphical models with $\Omega(K \log p)$ sample complexity. For strongly K-separable Gaussian graphical models, we additionally provide a method with error bounds for learning the off-diagonal precision matrix entries.

Keywords: Gaussian graphical model selection, separable graphs, high-dimensional statistical learning, faithful conditional independence relations, structural consistency

1. Introduction

Graphical models (Pearl (1988); Lauritzen (1996); Whittaker (1990); Wainwright and Jordan (2008)) are a popular and important means of representing certain conditional independence relations between random variables. In particular, the Gaussian graphical model is a popular model with applications to many areas such as object recognition and tracking (Sudderth (2006)), protein sequencing (Durbin et al. (1999)), pyschological modeling (Epskamp et al. (2018)), gene networks (Mohan et al. (2012); van Wieringen et al. (2018)), computer vision (Isard (2003)) and neuroimaging (Ryali et al. (2012); Belilovskym et al. (2016)). In a Gaussian graphical model, each variable is associated with a node in a graph, and any two nodes are connected by an undirected edge if and only if their two corresponding variables are dependent conditioned on the rest of the variables. An edge between two nodes therefore corresponds directly to the non-zero entries of the precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, where $\mathbf{\Sigma}$ is the covariance matrix of the multivariate Gaussian distribution in question. With the graphical model defined in this way, the Gaussian distribution satisfies the global Markov property: for any pair of nodes i and j, if all paths between the two pass through a set of nodes S, then the variables associated with i and j are conditionally independent given the variables associated with S.

The converse of the global Markov property does not always hold. When it does hold for a conditional independence relation, that relation is called faithful. If it holds for all relations in a model, that model is faithful. Faithfulness is important for the use of conditional independence relations in structural estimation of graphical models, that is, identifying the zeros of $\mathbf{\Omega}$. It can be challenging to simply invert $\mathbf{\Sigma}$. With faithfulness, to determine an edge between nodes i and j, one could run through all possible separator sets S and test for conditional independence (Liang et al. (2015); Koldanov et al. (2017)). If S is small, the computation becomes more accurate. In the work of (Meinshausen and Bühlmann (2006); Ravikumar et al. (2011); Anandkumar et al. (2012); Wu et al. (2013)), different assumptions are used to bound S to this end.

The main problem of faithfulness in graphical models is one of identifiability. Can we distinguish between a faithful graphical model and an unfaithful one? The idea of faithfulness was first explored for conditional independence relations that were satisfied in a family of graphs, using the notion of θ-Markov perfectness (Frydenberg (1990); Kauermann (1996)). For Gaussian graphical models with a tree topology the the distribution has been shown to be faithful (Becker et al. (2005); Malouche and Rajaratnam (2009)). In directed graphical models, the class of unfaithful distributions has been studied in (Spirites et al. (1993); Meek (1995)). In more recent work, a notion of strong-faithfulness as a means of relaxing the conditions of faithfulness was defined (Uhler et al. (2013); Lin et al., and some answers were given as to whether a faithful graph representation existed for a given probability distribution (Sadeghi (2017)).

Being able to distinguish between faithful and unfaithful relations is the first step. Our main goal is to learn the structure of the graphical model. Many literature (Meinshausen and Bühlmann (2006); Ravikumar et al. (2011); Ren et al. (2015); Dalal and Rajaratnam (2017)) involve a form of sparsity to make learning of the graphical model easier. This sparsity comes in the form of node degree, so that a graph is sparse if the node degree is small. However, in studying the use of conditional independence relations, another form of sparsity may prove to be more useful. This form of sparsity is the size of the set S that we are conditioning upon. We can make the set S small, by limiting the number of vertex disjoint paths between any two nodes in a graph that are not neighbors. Graphs that exhibit this property are what is known as K-separable graphs (Cicalese and Melanič (2012)). It is natural therefore to study these kinds of graphs. We will examine graph learning for this graph class in this paper.

In our paper, we make the following contributions:

- We propose an algorithm to test the faithfulness of a conditional independence relation of the form $X_u \perp X_v \mid \boldsymbol{X}_S$, where X_u, X_v and $\boldsymbol{X}_S$ are the random variables associated with the nodes u, v and the node set S. This algorithm uses other conditional independence relations of the form $X_i \perp X_j \mid \boldsymbol{X}_S$, where $i, j \notin S$ to determine this. The faithfulness test does not require any assumption on the population version of covariance matrix $\mathbf{\Sigma}$ and can be applied to any Gaussian graphical model. To the best of our knowledge, this is the first algorithm that uses local information of

©2019 De Wen Soh and Sekhar Tatikonda.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-329.html.

a matrix to test for the faithfulness of a conditional independence relation. We also provide sample complexity bounds for this algorithm.

- We propose a structure learning algorithm for weakly K-separable Gaussian graphical models. The quantity K controls the size of S that we need to condition on to learn the graph. This algorithm searches through different possible node sets S and makes use of the faithfulness test to identify when a conditional independence relation implies that there is no edge between two nodes in the topology of the graphical model. There is no particular assumption on how small K needs to be, except that K scales according to $O(n/\log p)$ where n is the number of samples and p is the dimension of the multivariate Gaussian distribution. There are two main novelties of this algorithm. The first is that we make no assumption on the model of the graph other than it is weakly K-separable. Therefore, we do not require assumptions for the graph to be faithful or to satisfy certain irrepresentability conditions, which tend to be hard to verfiy. The second novelty is that the weakly K-separable condition subsumes other known degree bound (sparsity) assumptions. This means that our algorithm caters to a larger class of Gaussian graphical models.

- We propose a precision matrix learning algorithm for strongly K-separable Gaussian graphical models. This algorithm not only learns the structure of the graph, it learns the entries of the precision matrix (edge weights) as well. Of course, to do so we can simply invert the matrix. However, with small K, we can get better sample complexity in the high-dimensional setting. The algorithm is similar to the weakly K-separable learning algorithm in the sense that it again runs through different node sets S to condition on. However, it uses the faithfulness test somewhat differently to identify separator nodes S. The goal is, for any pair of nodes i and j, to find S such that all paths from i to j that is of edge length greater than one must pass through some node in S. This gives us a nice expression for the precision matrix entries $\boldsymbol{\Omega}_{ij}$.

This paper is structured as follows: In Section 2, we discuss some preliminaries about Gaussian graphical models. In Section 3, we state our algorithm for testing the faithfulness of a conditional independence relation and the theoretical guarantees of the algorithm. In Section 4, we lay out an algorithm that learns the structure of a weakly K-separable Gaussian graphical model. In Section 5, we introduce our algorithm for learning the structure and precision matrix entries of a strongly K-separable Gaussian graphical model. In Section 6, we look at some examples where our algorithm can perform structural estimation while others that rely on sparsity cannot.

1.1. Related Work

The use of conditional independence relations to infer graph structure in the high dimensional setting was first introduced in Anandkumar et al. (2012). The authors in this work also use a search algorithm to search through different node sets S for every node pair i and j so as to determine whether an edge exists between i and j. They consider walk-summable Gaussian graphical models that satisfy a separability condition known as the local separability property. Because conditional independence relations are used to imply something about the topology of the graphical model, the problem of unfaithful conditional

independence relations needed to be dealt with. An assumption on the precision matrix (Assumption A4) restricts their models to faithful graphical models. We also note here that this assumption along with a minimum precision matrix entry assumption (Assumption A1) implies that the maximum node degree of the graph is $O(\sqrt{n/\log p})$.

Another class of papers that work on high-dimensional Gaussian graphical models structural estimation are those that make use of convex optimization on sparse precision matrices (Meinshausen and Bühlmann (2006); Ravikumar et al. (2011); Ren et al. (2015)). These papers make use of optimization procedures such as ℓ_1-regularization to learn the structure of the graphical model. These techniques exploit the sparsity of the precision matrix of the graphical model, which takes the form of bounded node degree. The order of the maximum node degree is different for different papers, and we discuss this later when we compare our results to theirs. Besides the sparsity element, the optimization methods (Meinshausen and Bühlmann (2006); Ravikumar et al. (2011)) also require some assumptions on the precision matrix that are hard to verify in general, which are the incoherence assumptions and neighborhood stability assumptions. Part of the motivation of our work was to overcome the need for such assumptions.

2. Preliminaries

2.1. Linear Algebra

We first define some linear algebra notation. For a matrix $\boldsymbol{M}$, let $\boldsymbol{M}^T$ denote its transpose and let $|\boldsymbol{M}|$ denote its determinant. If I is a subset of its row indices and J a subset of its column indices, then we define the submatrix $\boldsymbol{M}_{IJ}$ as the $|I| \times |J|$ matrix with elements with both row and column indices from I and J respectively. If $I = J$, we use the notation $\boldsymbol{M}_I$ for convenience. In the same way, for a vector $\boldsymbol{v}$, we define $\boldsymbol{v}_I$ to be the subvector of $\boldsymbol{v}$ with indices from I. Let the spectral norm of $\boldsymbol{M}$ be denoted by $\|\boldsymbol{M}\|_2$, and let the trace of $\boldsymbol{M}$ be denoted by $\mathrm{tr}(\boldsymbol{M})$. For a square matrix $\boldsymbol{M}$, we denote its maximum and minimum eigenvalues by $\lambda_{\max}(\boldsymbol{M})$ and $\lambda_{\min}(\boldsymbol{M})$. In this paper, we will often refer to the p by p covariance matrix $\boldsymbol{\Sigma}$, so we will use the shorthand $\lambda_{\max} = \lambda_{\max}(\boldsymbol{\Sigma})$ and $\lambda_{\min} = \lambda_{\min}(\boldsymbol{\Sigma})$.

Let $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ and let $\mathcal{W} = \{1, \ldots, p\}$ be the index set of $\boldsymbol{M}$. Let $S \subset \mathcal{W}$ and let $S^c = \mathcal{W} \setminus S$. The matrix $\boldsymbol{M}$ has the block structure

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_S & \boldsymbol{M}_{SS^c} \\ \boldsymbol{M}_{S^cS} & \boldsymbol{M}_{S^c} \end{bmatrix}. \tag{1}$$

The Schur complement of $\boldsymbol{M}_S$ in $\boldsymbol{M}$ is defined by

$$\boldsymbol{M}_{S^c|S} = \boldsymbol{M}_{S^c} - \boldsymbol{M}_{S^cS}\boldsymbol{M}_S^{-1}\boldsymbol{M}_{SS^c}. \tag{2}$$

Using the Schur complement, we can write the inverse of $\boldsymbol{M}^{-1}$ in the form

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{M}_{S|S^c}^{-1} & -\boldsymbol{M}_{S|S^c}^{-1}\boldsymbol{M}_{S^cS}\boldsymbol{M}_S^{-1} \\ -\boldsymbol{M}_S^{-1}\boldsymbol{M}_{SS^c}\boldsymbol{M}_{S|S^c}^{-1} & \boldsymbol{M}_S^{-1} + \boldsymbol{M}_S^{-1}\boldsymbol{M}_{SS^c}\boldsymbol{M}_{S|S^c}^{-1}\boldsymbol{M}_{S^cS}\boldsymbol{M}_S^{-1} \end{bmatrix}. \tag{3}$$

2.2. Graph Theory

Let $\mathcal{G} = (\mathcal{W}, \mathcal{E})$ be an undirected graph, where $\mathcal{W} = \{1, \ldots, p\}$ is the set of nodes and $\mathcal{E}$ is the set of edges, namely, a subset of the set of all unordered pairs $\{(u, v) \mid u, v \in$

$\mathcal{W}\}$. In our paper, we are dealing with graphs that have no self-loops and no multiple edges between the same pair of nodes. For $I \subseteq \mathcal{W}$, we denote the induced subgraph on nodes I by $\mathcal{G}_I$. For two distinct nodes u and v, a path of length t from u to v is a series $\{(u, w_1), (w_1, w_2), \ldots, (w_{t-2}, w_{t-1}), (w_{t-1}, v)\}$ of edges in $\mathcal{E}$, where $w_1, \ldots, w_{k-1} \in \mathcal{W}$.

Let $S \subseteq \mathcal{W} \setminus \{u, v\}$. We say that the node set S is a node separator of u and v if all paths from u to v must pass through some node in S. The graph $\mathcal{G}$ is connected if for any distinct nodes $u, v \in \mathcal{W}$, there is at least one path from u to v. Otherwise, the graph $\mathcal{G}$ is disjoint. A connected component of $\mathcal{G}$ is a subgraph of $\mathcal{G}$ that is connected. A disjoint graph can be divided into a number of connected components, where nodes from distinct connected components are not connected by a path. Therefore, S is a node separator of u and v if and only if $\mathcal{G}_{S^c}$ is a disjoint graph with u and v in distinct connected components.

Definition 1 *A graph $\mathcal{G} = (\mathcal{W}, \mathcal{E})$ is weakly K-separable if for every $(u, v) \notin \mathcal{E}$, there exists a node set $S \subset \mathcal{W} \setminus \{u, v\}$, $|S| \leq K$, such that S separates nodes i and j in $\mathcal{G}$.*

Definition 2 *A graph $\mathcal{G} = (\mathcal{W}, \mathcal{E})$ is strongly K-separable if for any two nodes u and v, there exists a node set $S \subset \mathcal{W} \setminus \{u, v\}$, $|S| \leq K$, such that every path from u to v consisting of more than two nodes must contain a node in S.*

For any two nodes, let $\mathcal{G}_{-(u,v)}$ be the resulting graph with the edge (u, v) deleted from $\mathcal{G}$. This means that $\mathcal{G}_{-(u,v)} = \mathcal{G}$ if and only if $(u, v) \notin \mathcal{E}$. The following proposition is an equivalent definition for a strongly K-separable graph.

Proposition 3 *Let $\mathcal{G}$ be an undirected graph. $\mathcal{G}$ is strongly K-separable if and only if for any two nodes u and v, there exists a node set $S \subset \mathcal{W} \setminus \{u, v\}$, $|S| \leq K$, such that $(\mathcal{G}_{-(u,v)})_{S^c}$ is a disjoint graph with nodes u and v in distinct connected components.*

A graph that is strongly K-separable graph is weakly K-separable as well.

Example 1 (Tree Graphs) *A tree graph is a weakly 1-separable graph, since any two nodes on a tree graph that are not neighbors are connected by one unique path (there are no cycles), so any node on that unique path would separate the two nodes. A tree graph is also trivially strongly 1-separable, since any two nodes connected by an edge is separated in the resultant graph where that particular edge is removed.*

Example 2 (Degree Bounded Graphs) *A graph $\mathcal{G}$ where the degree of each node is bounded by K is a weakly K-separable graph. For any pair of non-neighboring nodes u and v, the neighborhood of u separates u from v. Since this neighborhood size is bounded by K, so the graph must also be a weakly K-separable graph. This degree bounded graph is also a strongly K-separable graph as well, by the same logic. Let v be a neighbor node of u, that is, they are connected by the edge (u, v). Then the rest of the neighbors of u must separate u and v in $\mathcal{G}_{-(u,v)}$. This fact, along with the fact that the graph is already weakly K-separable, makes $\mathcal{G}$ strongly K-separable as well. It is important to note here that weakly K-separable graphs are not degree bounded graphs. A star graph of arbitrary degree is a tree and is thus a weakly 1-separable graph. However, it is not degree bounded.*

Example 3 (Locally Separated Graphs) *In Anandkumar et al. (2012), the notion of local separability was introduced. Let $\mathcal{G}$ be a graph with p nodes. For any two nodes u, v, we say that the number of hops required to reach v from u is the minimum number of edges in a path from u to v. If v is a neighbor of u, then v is 1-hop from u. For any $i \in \mathcal{W}$, let $B_\gamma(i)$ be the set of all nodes that are k-hop from i, including the node i, where $k \leq \gamma$. Let $\mathcal{G}_{B_\gamma(u) \cup B_\gamma(v)}$ be the induced subgraph of $\mathcal{G}$ on the node set $B_\gamma(u) \cup B_\gamma(v)$. The graph satisfies the (η, γ)-local separation property if for any two non-neighboring nodes u and v, the minimum number of nodes required to separate u and v in $\mathcal{G}_{B_\gamma(u) \cup B_\gamma(v)}$ is less than or equal to η. Of course, any graph satisfies the local separation property with the appropriate η and γ values. Let S be such a local separator node set for nodes u and v in the induced subgraph $\mathcal{G}_{B_\gamma(u) \cup B_\gamma(v)}$. Then any path in $\mathcal{G}$ from u to v that does not pass through S must have at least 2γ number of edges. Using the pigeonhole principle, it is easy to see that the graph is also weakly K-separable, where $K = \eta + \frac{p-\eta}{2\gamma}$. The graph may not be strongly K-separable however.*

Example 4 (Weakly K-separable Graphs that are not Strongly K-separable) *There are many examples of graphs that are weakly K-separable but not strongly K-separable. A simple example is the complete graph, with number of nodes $p > K + 2$. Because each node is a neighbor of every other node, it is trivially weakly K-separable. However, there are at least $p - 2$ vertex disjoint paths from u to v in $\mathcal{G}_{-(u,v)}$, namely, the paths of length 2 that go from u to any other node in $\mathcal{G}$, excluding u and v, to v. By Menger's theorem, at least $p - 2$ nodes are required to separate u and v in $\mathcal{G}_{-(u,v)}$, thus making the complete graph not strongly K-separable for $K < p - 2$.*

2.3. Gaussian Graphical Model

Let $\boldsymbol{X} = (X_1, \ldots, X_p)$ be a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. For the rest of this paper, we will only consider zero mean Gaussian distributions, that is, $\boldsymbol{\mu} = \mathbf{0}$. Let $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ be the precision or concentration matrix of the graph. The random variable $\boldsymbol{X}$ has the distribution function

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^p |\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Omega} (\boldsymbol{x} - \boldsymbol{\mu})\right\}. \tag{4}$$

For any set $S \subset \mathcal{W}$, we define $\boldsymbol{X}_S = \{X_i \mid i \in S\}$. We note here that $\boldsymbol{\Sigma}_{uv} = 0$ if and only if X_u is independent of X_v, which we denote by $X_u \perp X_v$. If X_u is independent of X_v conditioned on some random variable Z, we denote this independence relation by $X_u \perp X_v \mid Z$. Note that $\boldsymbol{\Omega}_{uv} = 0$ if and only if $X_u \perp X_v \mid \boldsymbol{X}_{\mathcal{W} \setminus \{u,v\}}$.

For any set $S \subseteq \mathcal{W}$, the conditional distribution of $\boldsymbol{X}_{S^c}$ given $\boldsymbol{X}_S = \boldsymbol{x}_S$ follows a multivariate Gaussian distribution with conditional mean $\boldsymbol{\mu}_{S^c} - \boldsymbol{\Sigma}_{S^c S} \boldsymbol{\Sigma}_S^{-1} (\boldsymbol{x}_S - \boldsymbol{\mu}_S)$ and conditional covariance matrix

$$\boldsymbol{\Sigma}_{S^c|S} = \boldsymbol{\Sigma}_{S^c} - \boldsymbol{\Sigma}_{S^c S} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\Sigma}_{S S^c}. \tag{5}$$

Observe that the conditional covariance is the Schur complement of $\boldsymbol{\Sigma}_S$ in $\boldsymbol{\Sigma}$. For distinct nodes $u, v \in \mathcal{W}$ and $S \subseteq \mathcal{W} \setminus \{u, v\}$, we have

$$(\boldsymbol{\Sigma}_{S^c|S})_{uv} = \boldsymbol{\Sigma}_{uv} - \boldsymbol{\Sigma}_{uS} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\Sigma}_{Sv}. \tag{6}$$

The following property easily follows.

Proposition 4 *$X_u \perp X_v \mid \boldsymbol{X}_S$ if and only if $(\boldsymbol{\Sigma}_{S^c|S})_{uv} = 0$.*

To denote conditional covariance, we use the notations $\boldsymbol{\Sigma}(u, v \mid S)$ and $(\boldsymbol{\Sigma}_{S^c|S})_{uv}$ interchangeably.

The concentration graph $\mathcal{G}_{\boldsymbol{\Sigma}} = (\mathcal{W}, \mathcal{E})$ of a multivariate Gaussian distribution $\boldsymbol{X}$ is defined as follows: We have node set $\mathcal{W} = \{1, \ldots, p\}$, with random variable X_u associated with node u, and edge set $\mathcal{E}$ where unordered pair (u, v) is in $\mathcal{E}$ if and only if $\boldsymbol{\Omega}_{uv} \neq 0$. The multivariate Gaussian distribution, along with its concentration graph, is also known as a Gaussian graphical model. Any Gaussian graphical model satisfies the global Markov property, which is the following.

Proposition 5 (Global Markov Property) *If S is a node separator of nodes u and v in $\mathcal{G}_{\boldsymbol{\Sigma}}$, then $X_u \perp X_v \mid \boldsymbol{X}_S$.*

The converse, however, is not necessarily true. Therefore, this motivates us to define faithfulness in a graphical model.

Definition 6 *The conditional independence relation $X_u \perp X_v \mid \boldsymbol{X}_S$ is said to be faithful if S is a node separator of u and v in the concentration graph $\mathcal{G}_{\boldsymbol{\Sigma}}$. Otherwise, it is unfaithful. A multivariate Gaussian distribution is faithful if all its conditional independence relations are faithful. The distribution is unfaithful if it is not faithful.*

Example 5 (Example of an unfaithful Gaussian distribution) *Consider the multivariate Gaussian distribution $\boldsymbol{X} = (X_1, X_2, X_3, X_4)$ with zero mean and positive definite covariance matrix*

$$\boldsymbol{\Sigma} = \begin{bmatrix} 3 & 2 & 1 & 2 \\ 2 & 4 & 2 & 1 \\ 1 & 2 & 7 & 1 \\ 2 & 1 & 1 & 6 \end{bmatrix}. \tag{7}$$

By Proposition 4, we have $X_1 \perp X_3 \mid X_2$ since $\boldsymbol{\Sigma}_{13} = \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{23}$. However, the precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ has no zero entries, so the concentration graph is a complete graph. This means that node 2 is not a node separator of nodes 1 and 3. The independence relation $X_1 \perp X_3 \mid X_2$ is thus not faithful and the distribution $\boldsymbol{X}$ is not faithful as well.

We can think of the submatrix $\boldsymbol{\Sigma}_{S\cup\{u,v\}}$ as a local patch of the covariance matrix $\boldsymbol{\Sigma}$. When $X_u \perp X_v \mid \boldsymbol{X}_S$, nodes u and v are not connected by an edge in the concentration graph of the local patch $\boldsymbol{\Sigma}_{S\cup\{u,v\}}$, that is, we have $(\boldsymbol{\Sigma}^{-1}_{S\cup\{u,v\}})_{uv} = 0$. This does not imply that u and v are not connected in the concentration graph $\mathcal{G}_{\boldsymbol{\Sigma}}$. If $X_u \perp X_v \mid \boldsymbol{X}_S$ is faithful, then the implication follows. If $X_u \perp X_v \mid \boldsymbol{X}_S$ is unfaithful, then u and v may be connected in $\mathcal{G}_{\boldsymbol{\Sigma}}$ (See Figure 1).

Faithfulness is important in structural estimation, especially in high-dimensional settings. If we assume faithfulness, then finding a node set S such that $X_u \perp X_v \mid \boldsymbol{X}_S$ would imply that there is no edge between u and v in the concentration graph. When we have access only to the sample covariance instead of the population covariance matrix, if the size of S is small compared to n, the error of computing $X_u \perp X_v \mid \boldsymbol{X}_S$ is much less than

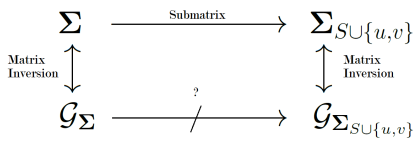


Figure 1: Even though $\boldsymbol{\Sigma}_{S\cup\{u,v\}}$ is a submatrix of $\boldsymbol{\Sigma}$, $\mathcal{G}_{\boldsymbol{\Sigma}_{S\cup\{u,v\}}}$ need not be a subgraph of $\mathcal{G}_{\boldsymbol{\Sigma}}$. Edge properties do not translate as well. That means the local patch $\boldsymbol{\Sigma}_{S\cup\{u,v\}}$ need not reflect the edge properties of the global graph structure of $\boldsymbol{\Sigma}$.

the error of inverting the entire covariance matrix. This method of searching through all possible node separator sets of a certain size is employed in Anandkumar et al. (2012); Wu et al. (2013). As mentioned before, these authors impose other restrictions on their models to overcome the problem of unfaithfulness. We do not place any restriction on the Gaussian models.

2.4. Sample Covariance

Let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^p$ be n samples of the random variable $\boldsymbol{X}$ with distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. The scatter matrix $\boldsymbol{S}$ is defined as

$$\boldsymbol{S} = \sum_{i=1}^{n} \boldsymbol{x}^{(i)}(\boldsymbol{x}^{(i)})^T. \tag{8}$$

The sample covariance matrix determined by these n samples is defined as

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\boldsymbol{S}. \tag{9}$$

In determining the sample conditional covariances, we will make use of the scatter matrix $\boldsymbol{S}$ instead of $\widehat{\boldsymbol{\Sigma}}$. Let u and v be distinct elements of $\mathcal{W}$ and let $S \subseteq \mathcal{W} \setminus \{u, v\}$. The sample conditional covariance of X_u and X_v given $\boldsymbol{X}_S$ is denoted by

$$\widehat{\boldsymbol{\Sigma}}(u, v \mid S) = \frac{1}{n - |S|}\left(\boldsymbol{S}_{uv} - \boldsymbol{S}_{uS}\boldsymbol{S}_S^{-1}\boldsymbol{S}_{Sv}\right). \tag{10}$$

In our algorithms, we usually have to decide whether a conditional independence relation holds. We have to determine whether $X_u \perp X_v \mid \boldsymbol{X}_S$ or $X_u \not\perp X_v \mid \boldsymbol{X}_S$. To do so with the sample covariance matrix, we need to define a conditional independence threshold $\alpha > 0$, such that if

$$|\widehat{\boldsymbol{\Sigma}}(u, v \mid S)| < \alpha, \tag{11}$$

we will decide that $X_u \perp X_v \mid \boldsymbol{X}_S$. Otherwise, we decide that $X_u \not\perp X_v \mid \boldsymbol{X}_S$. In our analysis, α will scale depending on p, n and $|S|$.

3. Testing Conditional Independence Relations

In this section, we will describe a novel algorithm for testing faithfulness of a conditional independence relation $X_u \perp X_v \mid \boldsymbol{X}_S$. Our main goal is to decide if a conditional independence relation $X_u \perp X_v \mid \boldsymbol{X}_S$ is faithful or not. For convenience, we will denote $\mathcal{G}_{\boldsymbol{\Sigma}}$ simply by $\mathcal{G} = (\mathcal{W}, \mathcal{E})$ for the rest of this paper. Now let us suppose that it is faithful; S is a node separator for u and v in $\mathcal{G}$. Then we should not be able to find a path from u to v in the induced subgraph $\mathcal{G}_{S^c}$. The main idea therefore is to search for a path between u and v in $\mathcal{G}_{S^c}$. If this fails, then we know that the conditional independence relation is faithful.

By the global Markov property, for any two distinct nodes $i, j \in S^c$, if $X_i \not\perp X_j \mid \boldsymbol{X}_S$, then we know that there is a path between i and j in $\mathcal{G}_{S^c}$. Thus, if we find some $w \in \mathcal{W} \setminus (S \cup \{i, j\})$ such that $X_u \not\perp X_w \mid \boldsymbol{X}_S$ and $X_v \not\perp X_w \mid \boldsymbol{X}_S$, then a path exists from u to w and another exists from v to w, so u and v are connected in $\mathcal{G}_{S^c}$. This would imply that $X_u \perp X_v \mid \boldsymbol{X}_S$ is unfaithful. However, testing for paths this way does not necessarily rule out all possible paths in $\mathcal{G}_{S^c}$. The problem is that some paths may be obscured by other unfaithful conditional independence relations. There may be some w whereby $X_u \not\perp X_w \mid \boldsymbol{X}_S$ and $X_v \perp X_w \mid \boldsymbol{X}_S$, but the latter relation is unfaithful. This path from u to v through w is thus not detected by these two independence relations.

We will show however, that if there is no path from u to v in $\mathcal{G}_{S^c}$, then we cannot find a series of distinct nodes $w_1, \ldots, w_t \in \mathcal{W} \setminus (S \cup \{u, v\})$ for some natural number $t > 0$ such that $X_u \not\perp X_{w_1} \mid \boldsymbol{X}_S$, $X_{w_1} \not\perp X_{w_2} \mid \boldsymbol{X}_S$, …, $X_{w_{t-1}} \not\perp X_{w_t} \mid \boldsymbol{X}_S$, $X_{w_t} \not\perp X_v \mid \boldsymbol{X}_S$. This is to be expected because of the global Markov property. What is more surprising about our result is that the converse is true. If we cannot find such nodes $w_1, \ldots, w_t$, then u and v are not connected by a path in $\mathcal{G}_{S^c}$. This means that if there is a path from u to v in $\mathcal{G}_{S^c}$, even though it may be hidden by some unfaithful conditional independence relations, ultimately there are enough conditional dependence relations to reveal that u and v are connected by a path in $\mathcal{G}_{S^c}$. This gives us an equivalent condition for faithfulness that is in terms of conditional independence relations.

This is the backbone of our algorithm to test for the faithfulness of a conditional independence relation. We define a new graph $\bar{\mathcal{G}} = (\bar{\mathcal{W}}, \bar{\mathcal{E}})$, where $\bar{\mathcal{W}} = S^c$, and $(i, j) \in \bar{\mathcal{E}}$ if and only if $X_i \not\perp X_j \mid \boldsymbol{X}_S$. The algorithm seeks to find a path from u to v in $\bar{\mathcal{G}}$. If there exists a path, then $X_u \perp X_j \mid \boldsymbol{X}_S$ is unfaithful. Otherwise, it is faithful. If we consider each test of whether two nodes are conditionally independent given $\boldsymbol{X}_S$ as one step, the running time of the algorithm is the that of the algorithm used to determine set U. If a breadth-first search is used, the running time is $O(|S^c|^2)$.

Theorem 7 *Suppose $X_u \perp X_v \mid \boldsymbol{X}_S$. If S is a node separator of u and v in the concentration graph, then Algorithm 1 will classify $X_u \perp X_v \mid \boldsymbol{X}_S$ as faithful. Otherwise, Algorithm 1 will classify $X_u \perp X_v \mid \boldsymbol{X}_S$ as unfaithful.*

Example 6 (Testing an Unfaithful Distribution (1)) *Let us take a look again at the 4-dimensional Gaussian distribution in Example 5. Suppose we want to test if $X_1 \perp X_3 \mid X_2$ is faithful or not. From its covariance matrix, we have $\boldsymbol{\Sigma}_{14} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_{24} = 2 - 2 \cdot 1/4 = 3/2 \neq 0$, so this implies that $X_1 \not\perp X_4 \mid X_2$. Similarly, $X_3 \not\perp X_4 \mid X_2$. So there exists a path from X_1 to X_3 in $\mathcal{G}_{\{1,3,4\}}$ that passes through node 4. Indeed there is such a path since the concentration graph is complete. Therefore, the relation $X_1 \perp X_3 \mid X_2$ is unfaithful.*

Algorithm 1: Testing Faithfulness of Relation $X_u \perp X_v \mid \boldsymbol{X}_S$

Input: Covariance matrix $\boldsymbol{\Sigma}$.

1. Generate set U to be the set of all nodes in $\bar{\mathcal{W}}$ that are connected to u by a path in $\bar{\mathcal{G}}$, including u. (A breadth-first search could be used.)
2. If $v \in U$, there exists a path from u to v in $\bar{\mathcal{G}}$, output $X_u \perp X_v \mid \boldsymbol{X}_S$ as unfaithful.
3. If $v \notin U$, let $V = \bar{W} \setminus U$. Output $X_u \perp X_v \mid \boldsymbol{X}_S$ as faithful.

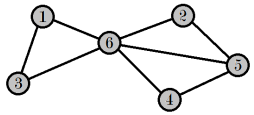


Figure 2: The concentration graph of the distribution in Example 8.

Example 7 (Testing an Unfaithful Distribution (2)) *Consider a 6-dimensional Gaussian distribution $\boldsymbol{X} = (X_1, \ldots, X_6)$ that has the covariance matrix*

$$\boldsymbol{\Sigma} = \begin{bmatrix} 7 & 1 & 2 & 2 & 3 & 4 \\ 1 & 8 & 2 & 1 & 2.25 & 3 \\ 2 & 2 & 10 & 4 & 3 & 8 \\ 2 & 1 & 4 & 9 & 1 & 6 \\ 3 & 2.25 & 3 & 1 & 11 & 9 \\ 4 & 3 & 8 & 6 & 9 & 12 \end{bmatrix}. \tag{12}$$

We want to test if the relation $X_1 \perp X_2 \mid X_6$ is faithful or unfaithful. Working out the necessary conditional independence relations to obtain $\bar{\mathcal{G}}$ with $S = \{6\}$, we observed that $(1,3), (3,5), (5,4), (4,2) \in \bar{\mathcal{E}}$ This means that 2 is reachable from 1 in $\mathcal{G}$, so the relation is unfaithful. In fact, the concentration graph is the complete graph K_6, and 6 is not a node separator of 1 and 2.

Example 8 (Testing a Faithful Distribution) *We consider a 6-dimensional Gaussian distribution $\boldsymbol{X} = (X_1, \ldots, X_6)$ that has a covariance matrix which is similar to the distribution in Example 7,*

$$\boldsymbol{\Sigma} = \begin{bmatrix} 7 & 1 & 2 & 2 & 3 & 4 \\ 1 & 8 & 2 & 1 & 2.25 & 3 \\ 2 & 2 & 10 & 4 & 6 & 8 \\ 2 & 1 & 4 & 9 & 1 & 6 \\ 3 & 2.25 & 6 & 1 & 11 & 9 \\ 4 & 3 & 8 & 6 & 9 & 12 \end{bmatrix}. \tag{13}$$

Observe that only $\boldsymbol{\Sigma}_{35}$ is changed. We again test the relation $X_1 \perp X_2 \mid X_6$. Running the algorithm produces a viable partition with $U = \{1, 3\}$ and $V = \{2, 4, 5\}$. This agrees with the concentration graph, as shown in Figure 2.

The proof of Theorem 7 reflects another interesting property of the graph $\bar{\mathcal{G}}$, which is that the connected components of $\bar{\mathcal{G}}$ are exactly the same connected components of $\mathcal{G}_{S^c}$. Suppose $\bar{\mathcal{G}}$ is a disjoint graph. Then the node set S^c can be partitioned into sets $S_1, \ldots, S_k$, for some $k \geq 2$, such that $\bar{\mathcal{G}}_{S_i}$ is connected for all $i = 1, \ldots, k$, and there are no edges in $\bar{\mathcal{E}}$ between nodes belonging to different partitions. This also implies that the graph $\mathcal{G}_{S^c}$ is disjoint in the same manner, in that $(\mathcal{G}_{S^c})_{S_i}$ is connected for all $i = 1, \ldots, k$, and there are no edges in $\mathcal{E}$ between nodes belonging to different partitions.

So far, we have not placed any assumption on the multivariate Gaussian distribution. Given the exact covariance matrix $\mathbf{\Sigma}$, for any general Gaussian graphical model, we can determine whether a node set S is a separator of two nodes u and v using conditional independence relationships to test for faithfulness. If the node set S is indeed a separator set of i and j, we can then conclude that there is no edge between nodes i and j. The next step therefore is to learn the entire structure of the graphical model.

3.1. Faithfulness Test Using Sample Conditional Covariances

Suppose now instead of the true covariance matrix $\mathbf{\Sigma}$, we only have access to n samples $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^p$ of the p-dimensional multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Since the faithfulness test only considers conditional relationships between pairs of nodes, we make use of the scatter matrix $\boldsymbol{S}$ to calculate the conditional covariances, according to Section 2.4.

Theorem 8 *Let $\boldsymbol{S}$ be used according to* (10) *to determine the sample conditional covariances in Algorithm 1 for testing the faithfulness of a conditional independence relation $X_u \perp X_v \mid \boldsymbol{X}_S$. Let $\beta = \min_{X_i \not\perp X_j \mid \boldsymbol{X}_S} |\mathbf{\Sigma}(i, j \mid S)|$, and let $\alpha = \beta/2$. Let Υ be the event in which:*

- *Using $\boldsymbol{S}$, Algorithm 1 correctly outputs whether or not X_u is conditional independent of X_v given $\boldsymbol{X}_S$.*
- *If $X_u \perp X_v \mid \boldsymbol{X}_S$, Algorithm 1 correctly outputs whether or not it is faithful using $\boldsymbol{S}$.*

Then,

$$\mathbb{P}(\Upsilon) \geq 1 - \epsilon. \tag{14}$$

for $n = \Omega\left(\frac{\lambda_{\max}^2 + \beta\lambda_{\max}}{\beta^2}\left(\log(p - |S|) + \log(\epsilon^{-1})\right) + |S|\right)$.

Some remarks:

- The algorithm to test the faithfulness of a conditional independent relation requires no assumption on the true covariance matrix $\mathbf{\Sigma}$. We only require that the distribution be multivariate Gaussian.
- The threshold α is used to decide if two variables are conditionally dependent or independent. If $|\widehat{\mathbf{\Sigma}}(i, j \mid S)| < \alpha$, then we decide $X_i \perp X_j \mid \boldsymbol{X}_S$, otherwise $X_i \not\perp X_j \mid \boldsymbol{X}_S$. Since we are using α to decide whether $\mathbf{\Sigma}(i, j \mid S)$ is zero or not, it is natural to set $\alpha = \beta/2$ where $\beta = \min_{X_i \not\perp X_j \mid \boldsymbol{X}_S} |\mathbf{\Sigma}(i, j \mid S)|$.

- Using the conditional independence test to determine faithfulness, we see that the sample complexity, if the dimension p is much larger than separator size $|S|$, then the $\log(p - |S|)$ term dominates. In this case, the separator size $|S|$ behaves like a kind of sparsity measure of the graph. However, as $|S|$ increases and gets closer to p, we see that the term $|S|$ dominates in the sample complexity. This is to be expected because when $|S|$ tends towards $p - 2$, we get closer to inverting almost the entire matrix. Of course, when $|S| = p - 2$, there is no faithfulness test needed to be done since $X_u \perp X_v \mid \boldsymbol{X}_{\mathcal{W} \setminus \{i,j\}}$ implies that there is no edge between nodes u and v. However the sample complexity needed is larger. Therefore, the faithfulness test allows us to reduce the size of $|S|$ and with that the sample complexity, even though there are more conditional relations to test between variables in S^c.
- The quantity $\beta = \min_{X_i \not\perp X_j \mid \boldsymbol{X}_S} |\mathbf{\Sigma}(i, j \mid S)|$ measures how far away the conditional covariance is from zero when it isn't zero. When this term is close to zero, more samples are required to test whether the conditional covariance is zero or non-zero for a pair of random variables. The relationship between β and the entries of precision matrix will be discussed later when we compare our recovery results with some of the other literature. Intuitively, it makes sense that this entry plays a role in the testing for faithfulness because the foundation of the algorithm is the testing for whether a pair of random variables are conditionally independent or not.

4. Weakly K-separable Gaussian graphical models

In the process of using conditional independence relationships to infer the structure of a Gaussian graphical model, if we let the separator size $|S|$ be $p - 2$, then we do not have to use any faithfulness test to determine whether any two nodes i and j are connected by an edge. Therefore, the true value of the faithfulness test is in the case where $|S|$ is small compared to the number of nodes p.

The main idea of this section is to learn the structure of a weakly K-separable Gaussian graphical model. For two nodes i and j that are not connected by an edge, there exists a node set S with $|S| \leq K$, such that S separates i and j. Thus, the variable K places a bound on the minimal node separator size for a node set that separates i and j. Consequently, K affects directly the sample complexity required for structural estimation. For K significantly smaller than p, the sample complexity involved in computing each of the conditional independence relations $X_i \perp X_j \mid \boldsymbol{X}_S$ is also significantly smaller than inverting the entire covariance matrix. Using the faithfulness test described in Algorithm 1 of the previous section, Algorithm 2 is able to learn the structure of a weakly K-separable graph, that is, it can estimate the edge set $\mathcal{E}$.

Again, considering a computation of a conditional independence relation as one step, the running time of the algorithm is $O(p^{K+4})$. This comes from exhaustively checking through all $\binom{p-2}{K}$ possible separation sets S for each of the $\binom{p}{2}$ (i, j) pairs. Each time there is a conditional independence relation, we have to check for faithfulness using Algorithm 1, and the running time for that is $O(p^2)$. The novelty of the algorithm is in its ability to learn graphical models that are unfaithful.

Algorithm 2: Learning topology of weakly K-separable GGM

Input: Covariance matrix $\mathbf{\Sigma}$.

1. For each node pair (i, j):
   - Let $F = \{S \subset \mathcal{W} \setminus \{i, j\} : |S| = K, X_i \perp X_j \mid \boldsymbol{X}_S, \text{ and it is faithful}\}$.
   - If $F \neq \phi$, output $(i, j) \notin \mathcal{E}$. If $F = \phi$, output $(i, j) \in \widehat{\mathcal{E}}$.
2. Output $\widehat{\mathcal{E}}$.

Theorem 9 *For a weakly K-separable Gaussian graphical model, given the exact covariance matrix $\mathbf{\Sigma}$ as the input in Algorithm 2, the corresponding output will be the correct edge set $\mathcal{E}$.*

Theorem 10 *Let $\mathcal{G}$ be a weakly K-separable graph. Let $\boldsymbol{S}$ be used according to* (10) *to determine the sample conditional covariances in Algorithm 2, instead of the true covariance matrix $\mathbf{\Sigma}$. Let $\beta \leq \min_{|S|=K, X_i \not\perp X_j | \boldsymbol{X}_S} |\mathbf{\Sigma}(i, j \mid S)|$, and let $\alpha = \beta/2$. Then,*

$$\mathbb{P}\left(\hat{\mathcal{E}} = \mathcal{E}\right) \geq 1 - \epsilon, \tag{15}$$

for $n = \Omega(\frac{\lambda_{\max}^2 + \beta\lambda_{\max}}{\beta^2}(K \log p + \log(\epsilon^{-1})) + K)$.

4.1. Comparisons to existing work

We compare this graph recovery algorithm to ones in existing work in the following aspects:

- *Faithfulness Assumptions*: In our result, we make no assumptions about the conditional independence relations having to be faithful. The purpose of Algorithm 1 is specifically to test for the faithfulness or unfaithfulness of a conditional independence relation. In Anandkumar et al. (2012), the authors make use of conditional covariances in the same way to test whether nodes are neighbors or not. Because they are using conditional covariances to infer graph structure, they naturally have to overcome the hurdle of unfaithful conditional independence relations. This is done by assumption A4 of their paper, which is an assumption on the covariance matrix $\mathbf{\Sigma}$ ensuring that pairs of nodes which are not neighbors and not separated by a set S will not have small conditional covariances when conditioned on the variables $\boldsymbol{X}_S$. This assumption is strong; together with Assumption A1, it implies that the node degree of the graph is bounded by $O(\sqrt{n/\log p})$. Our algorithm is novel in the sense that we can test for faithfulness and unfaithfulness using only conditional independence relations. Thus, we do not need an assumption like that of Assumption A4, that prevents the conditional independence relations from being unfaithful.

- *Degree Bounds*: The only assumption placed on the graph structure is that it is weakly K-separable. According to Theorem 10, for consistent structural estimation, we require

$$K = O\left(\frac{n}{\log p}\right). \tag{16}$$

This subsumes all graphs that have bounded node degree K. Degree boundedness is one of the main type of sparsity constraint in other high-dimensional Gaussian graphical model learning. In Anandkumar et al. (2012), even though the main structural assumption is the local separation property, their assumptions A1 and A4 imply that their graphs have node degree bounded by $O(\sqrt{n/\log p})$. In Ravikumar et al. (2011), the degree of each node is bounded by the same order as well. In Ren et al. (2015); Meinshausen and Bühlmann (2006), the degree bound required is $O(n/\log p)$. Thus, the weakly K-separable assumption subsumes all the assumptions on degree bounds, and Algorithm 2 caters to a wider class of graphical models.

- *General Gaussian Graphs*: Besides requiring the graph to be weakly K-separable, we place no other assumptions on the Gaussian graphical model. In Anandkumar et al. (2012), the authors require the model to be walk-summable, so that the normalized covariance matrix can be written using the Neumann series for matrix inverses. They also require assumption A4 to hold, which is a restriction on the entries of the precision matrix with respect to its corresponding row entries. In Ravikumar et al. (2011) and other works using ℓ-1 minimization, certain incoherence conditions need to hold, and these incoherence conditions are a restriction on the precision matrix and are hard to check in general. In Meinshausen and Bühlmann (2006), they place assumptions on neighborhood stability that allows them to do support recovery. The neighborhood stability assumption is hard to verify and the precision matrices that satisfy form a subset of diagonally dominant matrices.

5. Strongly K-separable Gaussian graphical models

In this section, we impose an additional assumption on the graphical model so that we can not only learn the topology of the graph, but learn the entries of the precision matrix as well. We can think of the precision matrix as the matrix of edge weights, where $\mathbf{\Omega}_{ij}$ is the weight of edge (i, j). If the edge weight is zero, it means that there is no edge between the corresponding two nodes.

The additional assumption is that the graphical model must not only be weakly K-separable, but it must be strongly K-separable as well. This means that even for edges (i, j) that belong to $\mathcal{E}$, there is a node set $S \subset \mathcal{W} \setminus \{i, j\}$, $|S| \leq K$, such that the removal of edge (i, j) from $\mathcal{G}$ results in a graph where S separates nodes i and j. This separation property is exactly where we can apply our faithfulness test. If we can remove the edge (i, j) from the graph and use our faithfulness test to find a node separator S of i and j in the resultant graph $\mathcal{G}_{-(i,j)}$, we can deduce the precision matrix entry $\mathbf{\Omega}_{ij}$. As shown in the following algorithm, if we can find such an S, the entry $\mathbf{\Omega}_{ij}$ can be calculated from the conditional covariance of X_i and X_j, and the conditional variances of X_i and of X_j, all of which are conditioned on $\boldsymbol{X}_S$.

The main idea behind the algorithm is to find such a node separator S, by appropriately "removing" the edge (i, j). We cannot simply condition on $\boldsymbol{X}_S$, because, if (i, j) is in $\mathcal{E}$, we

will have the condition dependence relation $X_i \not\perp X_j \mid \boldsymbol{X}_S$. We could remove the influence of the edge (i,j) in the graph by conditioning on $\boldsymbol{X}_{S\cup\{i,j\}}$, however this does not ensure that i and j are separated by S in $\mathcal{G}_{-(i,j)}$.

To overcome this problem, we condition on both $\boldsymbol{X}_{S\cup\{i\}}$ and $\boldsymbol{X}_{S\cup\{j\}}$. We use the conditional independence relations given these random variables to deduce that S is a node separator of i and j in $\mathcal{G}_{-(i,j)}$. Running through node subsets $S \subset \mathcal{W}\setminus\{i,j\}$ of size k, we first condition on $\boldsymbol{X}_{S\cup\{j\}}$ to see how S separates $\mathcal{G}_{S^c\setminus\{j\}}$. We then condition on $\boldsymbol{X}_{S\cup\{i\}}$ to see how S separates $\mathcal{G}_{S^c\setminus\{i\}}$. Using these two pieces of information, we can infer whether S separates i and j in $\mathcal{G}_{-(i,j)}$.

In the faithfulness test for a relation $X_u \perp X_v \mid \boldsymbol{X}_S$ in Section 3.1, we defined the graph of conditional dependences, $\bar{\mathcal{G}}$. The connectivity of $\bar{\mathcal{G}}$ reflects the connectivity of $\mathcal{G}_{S^c}$, which further implies whether S separates u and v. Here, we need to define $\bar{\mathcal{G}}$ for different node subsets S. For any subset $S \subset \mathcal{W}$, we denote the graph $\bar{\mathcal{G}}^{S^c} = (S^c, \bar{\mathcal{E}}^{S^c})$, where $(i,j) \in \bar{\mathcal{E}}^{S^c}$ if and only if $\boldsymbol{X}_i \not\perp X_j \mid \boldsymbol{X}_S$. For a node $h \in S^c$, let the connected node set component of $\bar{\mathcal{G}}_{S^c}$ containing h be denoted by $\bar{U}_{S^c}(h)$. Therefore $\bar{U}_{S^c}(h)$ is the set of nodes in S^c that are connected to h by a path in $\bar{\mathcal{G}}_{S^c}$, including h.

For any node $i \in \mathcal{W}$, we denote the set

$$\Gamma_{(i,j)} = \{S \subset \mathcal{W}\setminus\{i,j\} : |S| \le K\}. \tag{17}$$

of all possible node subsets S of size K in $\mathcal{W}\setminus\{i,j\}$. We define a subset of this set, which is

$$\Gamma_{i|j} = \{S \in \Gamma_{(i,j)} : \exists h \in S^c\setminus\{i,j\} \text{ s.t. } \boldsymbol{\Sigma}(i,h \mid S\cup\{j\}) = 0 \text{ and is faithful}\}. \tag{18}$$

This quantity encompasses the different sets S such that $\mathcal{G}_{S\cup\{j\}}$ is a disjoint graph. However, this set does not subsume all possible S that separate i and j in $\mathcal{G}_{-(i,j)}$. To include all such possible node sets S, we specify a subset of $\Gamma_{i|j}$, namely,

$$\Psi_{i|j} = \{S \in \Gamma_{(i,j)} : |S| \le K, \boldsymbol{\Sigma}(i,h \mid S\cup\{j\}) = 0,\ \forall h \in S^c\setminus\{i,j\}\}. \tag{19}$$

These quantities allow us to bring definition to Λ_1, Λ_2 and Λ_3, which are given in Algorithm 3. Basically, all S in $\Lambda_1\cup\Lambda_2\cup\Lambda_3$ are node sets that separate i and j in $\mathcal{G}_{-(i,j)}$. The set $\Lambda_1\cup\Lambda_2\cup\Lambda_3$ also has to be non-empty, which we prove in the appendix.

We only need to find one element of the set $\Lambda_1\cup\Lambda_2\cup\Lambda_3$. It is easy to test if S is an element of the set $\Psi_{i|j}$ or $\Psi_{j|i}$. Finding an element of $\Psi_{i|j}$ or $\Psi_{j|i}$ is a special case of the test for faithfulness. To do so, in the case of $\Psi_{i|j}$, run through the subsets S of size K, and determine if $\boldsymbol{\Sigma}(i,h \mid S\cup\{j\}) = 0$ for all $h \in S^c\setminus\{i,j\}$.

To test if S belongs to $\Gamma_{i|j}$, we have to run through the $K+1$ node set of $S\cup\{h\}$, where $h \in S^c\setminus\{i,j\}$. For each node set $S\cup\{h\}$, we employ Algorithm 1, the faithfulness test, to test whether $\boldsymbol{\Sigma}(i,h \mid S\cup\{j\})$ is zero and is a faithful conditional independence relation. If we can find such a node set $S\cup\{h\}$, then S belongs to $\Gamma_{i|j}$. If there is no $h \in S^c\setminus\{i,j\}$ whereby $\boldsymbol{\Sigma}(i,h \mid S\cup\{j\})$ is zero and faithful, then S belongs to $\Gamma_{(i,j)}\setminus\Gamma_{i|j}$.

In this way, using Algorithm 1, we can determine whether S belongs to the set $\Lambda_1\cup\Lambda_2\cup\Lambda_3$. Let each computation of a conditional independence relation be one step. For each (i,j) pair, there are $\binom{p-2}{K}$ possible sets S. For each S, there are $p-2-K$ possible nodes $h \in S^c\setminus\{i,j\}$ to pick from, with each h possibly require a faithfulness test, of which the running time is $O(p^2)$. Therefore, Algorithm 3 has a running time of $O(p^{K+5})$.

Algorithm 3: Learning the precision matrix of strongly K-separable GGM

Input: Covariance matrix $\boldsymbol{\Sigma}$

1. For each pair (i,j):
   - Let $\Lambda_1 = \big\{S \in \Gamma_{i|j}\cap\Gamma_{j|i} : \bar{U}_{S^c\setminus\{i\}}(j) \subseteq \big(S^c\setminus\bar{U}_{S^c\setminus\{j\}}(i)\big)\big\}$.
   - Let $\Lambda_2 = \Psi_{j|i}$.
   - Let $\Lambda_3 = \Psi_{i|j}$.
   - Choose a set $S^* \in \Lambda_1\cup\Lambda_2\cup\Lambda_3$.
   - Let $\widehat{\boldsymbol{\Omega}}_{ij} = -\boldsymbol{\Sigma}(i,j \mid S^*)/\left[\boldsymbol{\Sigma}(i,i \mid S^*)\boldsymbol{\Sigma}(j,j \mid S^*) - \boldsymbol{\Sigma}(i,j \mid S^*)^2\right]$.
2. Output $\widehat{\boldsymbol{\Omega}}$.

Theorem 11 *For a strongly K-separable Gaussian graphical model, given the exact covariance matrix $\boldsymbol{\Sigma}$ as the input in Algorithm 3, the corresponding output will be the correct precision matrix $\boldsymbol{\Omega}$.*

For each node pair i,j, we define

$$\Lambda_0(i,j) = \{S \in \Gamma_{(i,j)}, S \text{ separates } i \text{ and } j \text{ in } \mathcal{G}_{-(i,j)}\}. \tag{20}$$

Theorem 12 *Let $K \le p-2$. Let $\mathcal{G}$ be a strongly K-separable graph, and let $\boldsymbol{S}$ be used according to (10) to determine the sample conditional covariances in Algorithm 3, instead of the true covariance matrix $\boldsymbol{\Sigma}$. Let L be a constant such that $0 < L < 1$. Let*

$$C_1 = \frac{L\lambda_{\min}^2 + 2\lambda_{\max}(1+2\lambda_{\max})}{(1-L)L\lambda_{\min}^4}. \tag{21}$$

Then, for

$$\epsilon \in \left(0, C_1\cdot\min\left\{1, \frac{L\lambda_{\min}^2}{2(1+2\lambda_{\max})}\right\}\right], \tag{22}$$

we have

$$\mathbb{P}\left(\max_{\substack{i,j\in\mathcal{W}, i\neq j,\\ S\in\Lambda_0(i,j)}} |\widehat{\boldsymbol{\Omega}}_{ij} - \boldsymbol{\Omega}_{ij}| \ge \epsilon\right) \le 4p^{K+2}\exp\left\{-\frac{(n-K)\epsilon^2}{6C_1^2\lambda_{\max}^2 + 4C_1\epsilon\lambda_{\max}}\right\}. \tag{23}$$

Again, when we use the scatter matrix $\boldsymbol{S}$ to determine $\widehat{\Omega}_{ij}$, we get

$$\widehat{\boldsymbol{\Omega}}_{ij} = -\frac{\widehat{\boldsymbol{\Sigma}}(i,j \mid S^*)}{\widehat{\boldsymbol{\Sigma}}(i,i \mid S^*)\widehat{\boldsymbol{\Sigma}}(j,j \mid S^*) - \widehat{\boldsymbol{\Sigma}}(i,j \mid S^*)^2}, \tag{24}$$

where the sample conditional covariance are derived from $\boldsymbol{S}$ using (10).

Some remarks:

- Similar to the weakly K-separable case, we make no faithfulness assumptions about the model and we also do not make other assumptions besides the graph being strongly K-separable.

- Structural Estimation of a strongly K-separable graph using samples depends on accurately identifying the conditional independence relations. Therefore, the sample complexity required to estimate the structure of a strongly K-separable graph is the same as that of a weakly K-separable graph.

- The bounded degree assumption of other work is still subsumed under the strongly K-separable assumption. A K bounded degree graph is a strongly K-separable graph. As such, this algorithm learns a broader variety of Gaussian graphical models.

6. Discussion

So far, in both the weakly and strongly K-separable cases, the graphs with bounded degree can be learned as well. In this section, we will examine some graphs without bounded degree that our recovery algorithms can learn.

The complete graph: The complete graph on p nodes, as shown before, is a weakly K-separable graph, for any $K \geq 1$. In particular, it is weakly 1-separable graph. However, the node degree is the maximum possible, which is $p-1$. In this dense graph, our algorithm for weakly K-separable graphs can be used to learn the structure of the complete graph. We can learn the structure of the complete graph in $O(p^3)$, since every conditional independence in a weakly 1-separable graph is faithful. Edge estimation methods that require degree bounds will fail in this case. In fact, a weakly K-separable graph can contain many cliques of arbitrary size (). The presence of a large clique in a graph would render these methods inaccurate. For example, consider the p node graph with $p-1$ of its nodes forming a clique and the last node is connected to only one of the other $p-1$ nodes. This graph is weakly 1-separable as well. Therefore, our algorithm is able to deal with graphs that have arbitrarily large cliques in them. These graphs do not have bounded degree. However, the complete graph and graphs with large cliques are not strongly K-separable.

The star graph: Consider the star graph on p nodes. One node has degree $p-1$ and every other node has degree 1. This graph is weakly 1-separable and it is strongly 1-separable as well. Using Algorithm 3, we can learn the precision matrix entries of the star graph with low sample complexity. This algorithm has a running time of $O(p^4)$ in order to recover the structure of the graph, since all conditional independence relations in strongly K-separable graph are faithful. The precision matrix entries can then be derived according to 3. The ensemble of star graphs, however, clearly do not fall under the bounded degree regime. The star graph, in its essence, describes other graphs that are strongly K-separable, but not degree bounded. For example, as long as there is a node that is connected to a large number of degree 1 nodes, the graph is no longer degree bounded. However, in this case, the graph is still strongly K-separable, and we can use our algorithm to retrieve the graph structure and the entries of the precision matrix.

7. Conclusion

We have presented an equivalence condition for faithfulness in Gaussian graphical models and an algorithm to test whether a conditional independence relation is faithful or not. Gaussian distributions are special because its conditional independence relations depend on its covariance matrix, whose inverse, the precision matrix, provides us with a graph structure. The question of faithfulness in other Markov random fields, like Ising models, is an area of study that has much to be explored. The same questions can be asked, such as when unfaithful conditional independence relations occur, and whether they can be identified. Being able to test faithfulness allows us to learn a wider class of Gaussian graphical models, such as the weakly and strong K-separable graphs. In the future, we plan to extend some of these results to other Markov random fields.

Acknowledgments

This work was partially supported by the National Science Foundation under Grant CNS-0963989 and Grant CCF-1217023.

Appendix A. Correctness of Algorithms based on exact covariance matrix $\boldsymbol{\Sigma}$

A.1 Proof of Theorem 7

Proof Suppose Algorithm 1 decides that $X_u \perp X_v \mid \boldsymbol{X}_S$ is unfaithful. It does so by finding a series of distinct nodes $w_1, \ldots, w_t \in S^c \setminus \{u, v\}$ for some natural number $t > 0$ such that $X_u \not\perp X_{w_1} \mid \boldsymbol{X}_S$, $X_{w_1} \not\perp X_{w_2} \mid \boldsymbol{X}_S$, ..., $X_{w_{t-1}} \not\perp X_{w_t} \mid \boldsymbol{X}_S$, $X_{w_t} \not\perp X_v \mid \boldsymbol{X}_S$. By the global Markov property, this means that u is connected to w_1 by a path in $\mathcal{G}_{S^c}$, w_i is connected to w_{i+1} a path in $\mathcal{G}_{S^c}$ for $i = 1, \ldots, t-1$, and w_t is connected to v by a path in $\mathcal{G}_{S^c}$. This implies u is connected to v by a path in $\mathcal{G}_{S^c}$, so $X_u \perp X_v \mid \boldsymbol{X}_S$ is unfaithful and Algorithm 1 has correctly deduced that it is so.

Now suppose Algorithm 1 decides that $X_u \perp X_v \mid \boldsymbol{X}_S$ is faithful. That means that there is no path from u to v in $\bar{\mathcal{G}}$. Thus, $\bar{\mathcal{G}}$ is a disjoint graph with u and v in separate distinct components. The graph $\bar{\mathcal{G}}_U$ is the connected component that contains u. By the way we defined $\bar{\mathcal{G}}$, it follows that $X_i \perp X_j \mid \boldsymbol{X}_S$ for all $i \in U$ and $j \in V$. Equivalently, by Proposition 4,

$$(\boldsymbol{\Sigma}_{S^c|S})_{ij} = 0, \quad \forall i \in U,\ j \in V. \tag{25}$$

The matrix $\boldsymbol{\Sigma}_{S^c|S}$ therefore has a block diagonal structure, with

$$(\boldsymbol{\Sigma}_{S^c|S})_{UV} = (\boldsymbol{\Sigma}_{S^c|S})_{VU}^T = \mathbf{0}. \tag{26}$$

From (5) and (3), it follows that

$$(\boldsymbol{\Sigma}_{S^c|S})^{-1} = (\boldsymbol{\Sigma}_{S^c} - \boldsymbol{\Sigma}_{S^cS}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{\Sigma}_{SS^c})^{-1} = \boldsymbol{\Omega}_{S^c}. \tag{27}$$

Since the inverse of a block diagonal matrix is also block diagonal, it follows that

$$(\boldsymbol{\Omega}_{S^c})_{UV} = \boldsymbol{\Omega}_{UV} = \mathbf{0}. \tag{28}$$

As the non-zero entries of $\mathbf{\Omega}_{S^c}$ reflect the edges between the nodes in $\mathcal{G}_{S^c}$, the last equation implies that for any $i \in U$ and $j \in V$, the edge (i,j) is not in the edge set $\mathcal{E}$. This means u is not connected to v by a path in $\mathcal{G}_{S^c}$, which further implies that S is a separator of u and v in $\mathcal{G}$. Thus, Algorithm 1 has correctly deduced that $X_u \perp X_v \mid \boldsymbol{X}_S$ is a faithful conditional independence relation. ■

A.2 Proof of Theorem 9

Proof If $F \neq \phi$, F is non-empty so there exists an S such that $X_i \perp X_j \mid \boldsymbol{X}_S$ is faithful. Therefore, S separates i and j in $\mathcal{G}$ and $(i,j) \notin \mathcal{E}$. If $F = \phi$, then for any $S \subseteq \mathcal{W}$, $|S| \leq K$, we have either $X_i \not\perp X_j \mid \boldsymbol{X}_S$ or $X_i \perp X_j \mid \boldsymbol{X}_S$ but it is unfaithful. In both cases, S does not separate i and j in $\mathcal{G}$, for any $S \subseteq \mathcal{W}$, $|S| \leq K$. By the assumption on the graphical model, (i,j) must be in $\mathcal{E}$. This shows that Algorithm 2 will correctly output the edges of $\mathcal{G}$. ■

A.3 Conditional covariance in terms of the precision matrix

We establish some properties of the covariance matrix in terms of the precision matrix. In most of the work in this paper, we are trying to learn properties of the precision matrix, such as the support or the entries of the matrix, using conditional independence relations. These conditional independence relations are reflected by entries of the covariance matrix. Here, in this section, we further describe some of the relationship between the covariance matrix $\mathbf{\Sigma}$ and the precision matrix $\mathbf{\Omega}$.

Let i, j be two elements of the index set $\mathcal{W} = \{1, \ldots, p\}$ of the square matrix $\mathbf{\Sigma}$ and let $Q = \{i, j\}$. Let S be a subset of $\mathcal{W} \setminus Q$. Let $S^c = \mathcal{W} \setminus S$ and let $T = S^c \setminus Q$. Consider the matrix $\mathbf{\Omega}_{S^c}$. Computing the Schur complement of $\mathbf{\Omega}_{S^c \setminus Q}$ with respect to $\mathbf{\Omega}_{S^c}$ and using (3), we get

$$[(\mathbf{\Omega}_{S^c}^{-1})_Q]^{-1} = \mathbf{\Omega}_Q - \mathbf{\Omega}_{QT}\mathbf{\Omega}_T^{-1}\mathbf{\Omega}_{TQ}. \tag{29}$$

Using (27), we get

$$[(\mathbf{\Sigma}_{S^c|S})_Q]^{-1} = \mathbf{\Omega}_Q - \mathbf{\Omega}_{QT}\mathbf{\Omega}_T^{-1}\mathbf{\Omega}_{TQ}. \tag{30}$$

Now $(\mathbf{\Sigma}_{S^c|S})_Q$ has the form

$$(\mathbf{\Sigma}_{S^c|S})_Q = \begin{bmatrix} \mathbf{\Sigma}(i,i \mid S) & \mathbf{\Sigma}(i,j \mid S) \\ \mathbf{\Sigma}(i,j \mid S) & \mathbf{\Sigma}(j,j \mid S) \end{bmatrix}. \tag{31}$$

Therefore, comparing off-diagonal entries in (30), we get

$$\frac{-\mathbf{\Sigma}(i,j \mid S)}{\mathbf{\Sigma}(i,i \mid S)\mathbf{\Sigma}(j,j \mid S) - \mathbf{\Sigma}(i,j \mid S)^2} = \mathbf{\Omega}_{ij} - \mathbf{\Omega}_{iT}\mathbf{\Omega}_T^{-1}\mathbf{\Omega}_{Tj}. \tag{32}$$

We will make use of this last equation to learn the entries of the precision matrix for a strongly K-separable graph.

A.4 Proof of Theorem 11

Proof If S^* separates i and j in the edge-truncated graph $\mathcal{G}_{-(i,j)}$, this means that there exists node sets $T_i, T_j \subset \mathcal{W} \setminus S^*$, such that

- $T_i \cup T_j = \mathcal{W} \setminus S^*$ and $T_i \cap T_j = \phi$;
- $i \in T_i$, $j \in T_j$;
- $\mathbf{\Omega}_{h_1 h_2} = 0$, for all $h_1 \in T_i, h_2 \in T_j$, $(h_1, h_2) \neq (i,j)$.

This implies that

$$\begin{aligned} \mathbf{\Omega}_{iT}\mathbf{\Omega}_T^{-1}\mathbf{\Omega}_{Tj} &= \begin{bmatrix} \mathbf{\Omega}_{iT_i} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{\Omega}_{T_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_{T_j} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{\Omega}_{T_j j} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{\Omega}_{iT_i} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{\Omega}_{T_i}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_{T_j}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{\Omega}_{T_j j} \end{bmatrix} \\ &= 0. \end{aligned} \tag{33}$$

This reduces (32) to

$$\mathbf{\Omega}_{ij} = \frac{-\mathbf{\Sigma}(i,j \mid S)}{\mathbf{\Sigma}(i,i \mid S)\mathbf{\Sigma}(j,j \mid S) - \mathbf{\Sigma}(i,j \mid S)^2}, \tag{34}$$

which is the correct output that we want from the algorithm.

To complete the proof, we need to show that the set $\Lambda_1 \cup \Lambda_2 \cup \Lambda_3$ is non-empty for any pair of nodes i and j, and that any element of $\Lambda_1 \cup \Lambda_2 \cup \Lambda_3$ separates i and j in $\mathcal{G}_{-(i,j)}$. Let

$$\Lambda_0 = \{S \in \Gamma_{(i,j)} : S \text{ separates } i \text{ and } j \text{ in } \mathcal{G}_{-(i,j)}\}. \tag{35}$$

If $\Lambda_0 = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3$, by the definition of a strongly K-separable graph, there exists a set $S \in \Gamma_{i,j}$ such that S separates i and j in $\mathcal{G}_{-(i,j)}$. This means that Λ_0 is non-empty, and by association, $\Lambda_1 \cup \Lambda_2 \cup \Lambda_3$ is non-empty as well. It also follows that any element of $\Lambda_1 \cup \Lambda_2 \cup \Lambda_3$ separates i and j in $\mathcal{G}_{-(i,j)}$.

Therefore, it remains to show that $\Lambda_0 = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3$. Suppose $S \in \Lambda_0$, that is, $S \in \Gamma_{i,j}$ that separates i and j in $\mathcal{G}_{-(i,j)}$. If i is an isolated node in $(\mathcal{G}_{-(i,j)})_{S^c}$, then $S \in \Lambda_3$. If j is an isolated node in $(\mathcal{G}_{-(i,j)})_{S^c}$, then $S \in \Lambda_2$. Suppose i and j are both not isolated nodes. Let $U_i \in S^c$ be the set of nodes connected to i by some path in $(\mathcal{G}_{-(i,j)})_{S^c}$ including i, and let $U_j \in S^c$ be the set of nodes connected to j by some path in $(\mathcal{G}_{-(i,j)})_{S^c}$ including j. Since i and j are not isolated nodes, both U_i and U_j contains at least two elements. Let $h \in U_i$ such that $h \neq i$. Then $\mathbf{\Sigma}(j,h \mid S \cup \{i\}) = 0$ and is faithful since $S \cup \{j\}$ separates h and j in $\mathcal{G}$. This implies that $S \in \Gamma_{j|i}$. Similarly, $S \in \Gamma_{i|j}$. Observe also that

$$\bar{U}_{S^c \setminus \{i\}}(j) \subseteq U_j \subseteq (S^c \setminus U_i) \subseteq \left(S^c \setminus \bar{U}_{S^c \setminus \{j\}}(i)\right). \tag{36}$$

The first relation follows from the fact that $\bar{U}_{S^c \setminus \{i\}}(j)$ contains only the nodes from U_j that are connected to j by some path in $(\mathcal{G}_{-(i,j)})_{S^c}$ that does not pass through node i. The last relation also follows from a similar argument. The middle relation follows because S separates i and j in $\mathcal{G}_{-(i,j)}$. This implies that $S \in \Lambda_1$. Therefore $\Lambda_0 \subseteq \Lambda_1 \cup \Lambda_2 \cup \Lambda_3$.

Now, let $S \in \Lambda_1$. For the sake of contradiction, suppose that there is a path in $(\mathcal{G}_{-(i,j)})_{S^c}$ from i to j. This implies that there is a node h such that

- i is connect by a path to h in $(\mathcal{G}_{-(i,j)})_{S^c}$ that does not pass through j;
- j is connect by a path to h in $(\mathcal{G}_{-(i,j)})_{S^c}$ that does not pass through i.

The first property is equivalent to there being a path from i to h in $(\mathcal{G}_{-(i,j)})_{S^c\setminus\{j\}}$, which is if and only if $h \in \bar{U}_{S^c\setminus\{j\}}(i)$. Similarly, $h \in \bar{U}_{S^c\setminus\{i\}}(j)$. This implies that $h \notin (S^c \setminus \bar{U}_{S^c\setminus\{j\}}(i))$, which means that $\bar{U}_{S^c\setminus\{i\}}(j) \not\subseteq (S^c \setminus \bar{U}_{S^c\setminus\{j\}}(i))$, which contradicts $S \in \Lambda_1$. Therefore, there is no path in $(\mathcal{G}_{-(i,j)})_{S^c}$ from i to j, which implies that S is a node separator of i and j in $\mathcal{G}_{-(i,j)}$, that is, $S \in \Lambda_0$.

Next, let $S \in \Lambda_2$. This means that $S \in \Psi_{j|i}$. Suppose that j is connected by a path to i in $(\mathcal{G}_{-(i,j)})_{S^c}$. Then there exists a node h on this path such that j is connected to h by a path in $(\mathcal{G}_{-(i,j)})_{S^c\setminus i}$ that does not pass through i. Therefore, $\mathbf{\Sigma}(i,h \mid S \cup \{i\})$ cannot be zero and faithful at the same time. This contradicts $S \in \Psi_{j|i}$, by definition. Therefore, j is not connected by a path to i in $(\mathcal{G}_{-(i,j)})_{S^c}$, which also means that $S \in \Lambda_0$. Thus $\Lambda_2 \subseteq \Lambda_0$. Similarly, by symmetry $\Lambda_3 \subseteq \Lambda_0$. Therefore, $\Lambda_0 = \Lambda_1 cup \Lambda_2 \cup \Lambda_3$.

■

Appendix B. Sample Analysis

B.1 Wishart Distribution

Let $\boldsymbol{X} = (X_1, \ldots, X_p) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ be n independent samples of $\boldsymbol{X}$. The random scatter matrix $\boldsymbol{S}$ follows a Wishart distribution, which depends on parameters $\mathbf{\Sigma}$, p and n. We denote the Wishart distribution by $W(\mathbf{\Sigma}, p, n)$. For convenience, we denote

$$A_{n,\mathbf{\Sigma}} = \frac{1}{n}\|\mathbf{\Sigma}\|_2 \left(\mathrm{tr}(\mathbf{\Sigma}) + \|\mathbf{\Sigma}\|_2\right), \tag{37}$$

$$B_{n,\mathbf{\Sigma}} = \frac{2}{n}\mathrm{tr}(\mathbf{\Sigma}). \tag{38}$$

We will make use of the following proposition from Zhu (2012) in our sample analysis.

Proposition 13 *Let $\boldsymbol{S} \sim W(\mathbf{\Sigma}, p, n)$. The following inequality*

$$\mathbb{P}\left(\left\|\frac{1}{n}\boldsymbol{S} - \mathbf{\Sigma}\right\|_2 \geq \epsilon\right) \leq 2p \exp\left\{-\frac{\epsilon^2}{2A_{n,\mathbf{\Sigma}} + 2\epsilon B_{n,\mathbf{\Sigma}}}\right\}. \tag{39}$$

holds for all $\epsilon \geq 0$.

We can simplify the above proposition to the following form.

Lemma 14 *Let $\boldsymbol{S} \sim W(\mathbf{\Sigma}, p, n)$. Then,*

$$\mathbb{P}\left(\left\|\frac{1}{n}\boldsymbol{S} - \mathbf{\Sigma}\right\|_2 \geq \epsilon\right) \leq 2p \exp\left\{-\frac{n\epsilon^2}{(2+C)(p+1)\|\mathbf{\Sigma}\|_2^2}\right\}. \tag{40}$$

for all $\epsilon \in \left[0, \frac{C\|\mathbf{\Sigma}\|_2}{4}\left(1 + \frac{\|\mathbf{\Sigma}\|_2}{tr(\mathbf{\Sigma})}\right)\right]$, where $C > 0$ is a constant.

Proof
For $\epsilon \in \left[0, \frac{C\|\mathbf{\Sigma}\|_2}{4}\left(1 + \frac{\|\mathbf{\Sigma}\|_2}{\mathrm{tr}(\mathbf{\Sigma})}\right)\right]$, we have

$$2\epsilon B_{n,\mathbf{\Sigma}} \leq 2B_{n,\mathbf{\Sigma}} \cdot \frac{C\|\mathbf{\Sigma}\|_2}{4}\left(1 + \frac{\|\mathbf{\Sigma}\|_2}{\mathrm{tr}(\mathbf{\Sigma})}\right) = CA_{n,\mathbf{\Sigma}}. \tag{41}$$

Also, since

$$\mathrm{tr}(\mathbf{\Sigma}) \leq p\|\mathbf{\Sigma}\|_2, \tag{42}$$

we have

$$A_{n,\mathbf{\Sigma}} \leq \frac{1}{n}(p+1)\|\mathbf{\Sigma}\|_2^2. \tag{43}$$

Therefore, applying both (41) and (43), we get

$$\begin{aligned} \exp\left\{-\frac{\epsilon^2}{2A_{n,\mathbf{\Sigma}} + 2\epsilon B_{n,\mathbf{\Sigma}}}\right\} &\leq \exp\left\{-\frac{\epsilon^2}{(C+2)A_{n,\mathbf{\Sigma}}}\right\} \\ &\leq \exp\left\{-\frac{n\epsilon^2}{(2+C)(p+1)\|\mathbf{\Sigma}\|_2^2}\right\}, \end{aligned} \tag{44}$$

as required.

■

This inequality provides a bound for the spectral norm of the sample covariance matrix with respect to the actual covariance matrix. We however want an entry-wise bound for the sample conditional covariance matrix, conditioned on $\boldsymbol{X}_S$, where $S \subseteq \{1, \ldots, p\} = \mathcal{W}$. To do so, we make use of the following proposition from Eaton (2007), which gives us the distribution of the sample conditional covariance matrix. Let $S^c = \mathcal{W} \setminus S$. We define $\widehat{\mathbf{\Sigma}}^{|S}$ to be the $p - |S|$ by $p - |S|$ matrix, where

$$\widehat{\mathbf{\Sigma}}^{|S}_{ij} = \widehat{\mathbf{\Sigma}}(i, j \mid S), \tag{45}$$

with $\widehat{\mathbf{\Sigma}}(i, j \mid S)$ defined by (10).

Proposition 15 *(Eaton (2007)) The conditonal covariance matrix $\boldsymbol{S}_{S^c|S}$ follows a Wishart distribution with parameters $\mathbf{\Sigma}_{S^c|S}$, $p-|S|$ and $n-|S|$, that is, $\boldsymbol{S}_{S^c|S} \sim W(\mathbf{\Sigma}_{S^c|S}, p-|S|, n-|S|)$.*

Using this fact, we can now provide an element-wise bound for the sample conditional covariance.

Lemma 16 *For any $i, j \in \mathcal{W} \setminus S$, the sample conditional covariance satisfies*

$$\mathbb{P}\left(\left|\widehat{\mathbf{\Sigma}}(i, j \mid S) - \mathbf{\Sigma}(i, j \mid S)\right| \geq \epsilon\right) \leq 4 \exp\left\{-\frac{(n-|S|)\epsilon^2}{6\lambda_{\max}^2 + 4\epsilon\lambda_{\max}}\right\}, \tag{46}$$

for all $\epsilon \geq 0$.

Proof The submatrix $\boldsymbol{S}_{S\cup\{i,j\}}$ of the scatter matrix $\boldsymbol{S}$ follows a Wishart distribution with parameters $\boldsymbol{\Sigma}_{S\cup\{i,j\}}, |S|+2$ and n, that is, $\boldsymbol{S}_{S\cup\{i,j\}} \sim W(\boldsymbol{\Sigma}_{S\cup\{i,j\}}, |S|+2, n)$. Let $Q = \{i,j\}$, and let

$$\boldsymbol{\Sigma}_{Q|S} = \boldsymbol{\Sigma}_Q - \boldsymbol{\Sigma}_{QS}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{\Sigma}_{SQ}. \tag{47}$$

By Proposition 15, we have $\boldsymbol{S}_{Q|S} \sim W(\boldsymbol{\Sigma}_{Q|S}, 2, n-|S|)$, where

$$\boldsymbol{S}_{Q|S} = \boldsymbol{S}_Q - \boldsymbol{S}_{QS}\boldsymbol{S}_S^{-1}\boldsymbol{S}_{SQ}. \tag{48}$$

Applying Proposition **??** to $\boldsymbol{S}_{Q|S}$, we get

$$\begin{aligned}&\mathbb{P}\left(\left\|\frac{1}{n-|S|}\boldsymbol{S}_{Q|S} - \boldsymbol{\Sigma}_{Q|S}\right\|_2 \geq \epsilon\right)\\ &\leq 4\exp\left\{-\frac{(n-|S|)\epsilon^2}{2\|\boldsymbol{\Sigma}_{Q|S}\|_2\left(\mathrm{tr}(\boldsymbol{\Sigma}_{Q|S}) + \|\boldsymbol{\Sigma}_{Q|S}\|_2\right) + 4\epsilon(\mathrm{tr}(\boldsymbol{\Sigma}_{Q|S}))}\right\}.\end{aligned} \tag{49}$$

for all $\epsilon \geq 0$. Using the eigenvalue interlacing properties for the Schur complement and submatrices Smith (1992), we have

$$\lambda_{\min}^2 \leq \lambda_{\min}^2(\boldsymbol{\Sigma}_{Q\cup S}) \leq \|\boldsymbol{\Sigma}_{Q|S}\|_2^2 \leq \|\boldsymbol{\Sigma}_{Q\cup S}\|_2^2 \leq \|\boldsymbol{\Sigma}\|_2^2 = \lambda_{\max}. \tag{50}$$

Also, we have

$$\mathrm{tr}(\boldsymbol{\Sigma}_{Q|S}) \leq 2\lambda_{\max}(\boldsymbol{\Sigma}_{Q|S}) \leq 2\lambda_{\max}. \tag{51}$$

This give us the probabilistic bound

$$\mathbb{P}\left(\left\|\frac{1}{n-|S|}\boldsymbol{S}_{Q|S} - \boldsymbol{\Sigma}_{Q|S}\right\|_2 \geq \epsilon\right) \leq 4\exp\left\{-\frac{(n-|S|)\epsilon^2}{6\lambda_{\max}^2 + 4\epsilon\lambda_{\max}}\right\}. \tag{52}$$

for all $\epsilon \geq 0$.

For any matrix, the maximum of the absolute value of its entries is bounded above by the spectral norm. Since $\widehat{\boldsymbol{\Sigma}}(i,j \mid S) = \frac{1}{n-|S|}(\boldsymbol{S}_{Q|S})_{ij}$, we have

$$\left|\widehat{\boldsymbol{\Sigma}}(i,j \mid S) - \boldsymbol{\Sigma}(i,j \mid S)\right| \leq \left\|\frac{1}{n-|S|}\boldsymbol{S}_{Q|S} - \boldsymbol{\Sigma}_{Q|S}\right\|_2. \tag{53}$$

This gives us,

$$\mathbb{P}\left(\left|\widehat{\boldsymbol{\Sigma}}(i,j \mid S) - \boldsymbol{\Sigma}(i,j \mid S)\right| \geq \epsilon\right) \leq \mathbb{P}\left(\left\|\frac{1}{n-|S|}\boldsymbol{S}_{Q|S} - \boldsymbol{\Sigma}_{Q|S}\right\|_2 \geq \epsilon\right), \tag{54}$$

which completes the proof. ■

Using Lemma 16, we prove the following two useful corollaries.

Corollary 17 *Let $S \subset \mathcal{W}$, with $|S| \leq p-2$. Then*

$$\mathbb{P}\left(\max_{i,j\in S^c}\left|\widehat{\boldsymbol{\Sigma}}(i,j \mid S) - \boldsymbol{\Sigma}(i,j \mid S)\right| \geq \epsilon\right) \leq 4(p-|S|)^2\exp\left\{-\frac{(n-|S|)\epsilon^2}{6\lambda_{\max}^2 + 4\epsilon\lambda_{\max}}\right\}, \tag{55}$$

for all $\epsilon \geq 0$.

Proof Let $A_{i,j}$ be the event $\left\{\left|\widehat{\boldsymbol{\Sigma}}(i,j \mid S) - \boldsymbol{\Sigma}(i,j \mid S)\right| \geq \epsilon\right\}$. Then,

$$\mathbb{P}\left(\max_{i,j\in S^c}\left|\widehat{\boldsymbol{\Sigma}}(i,j \mid S) - \boldsymbol{\Sigma}(i,j \mid S)\right| \geq \epsilon\right) = \mathbb{P}\left(\bigcup_{i,j\in S^c} A_{i,j}\right). \tag{56}$$

Applying the union bound, we get

$$\mathbb{P}\left(\bigcup_{i,j\in S^c} A_{i,j}\right) \leq \sum_{i,j\in S^c} P\left(A_{i,j}\right) \leq 4(p-|S|)^2\exp\left\{-\frac{(n-|S|)\epsilon^2}{6\lambda_{\max}^2 + 4\epsilon\lambda_{\max}}\right\}, \tag{57}$$

since there are $\binom{p-|S|}{2}$ possible choices of $\{i,j\}$. ■

Corollary 18 *Let $K < p-2$. The following inequality*

$$\mathbb{P}\left(\max_{\substack{S\subset\mathcal{W},|S|=K,\\ i,j\in S^c}}\left|\widehat{\boldsymbol{\Sigma}}(i,j \mid S) - \boldsymbol{\Sigma}(i,j \mid S)\right| \geq \epsilon\right) \leq 4p^{K+2}\exp\left\{-\frac{(n-K)\epsilon^2}{6\lambda_{\max}^2 + 4\epsilon\lambda_{\max}}\right\}, \tag{58}$$

holds for all $\epsilon \geq 0$.

Proof The proof follows that of Corollary 17, except that the union bound now runs over all different choices of i,j, and S, with $|S| = K$. There are altogether $\binom{p}{2}\binom{p-2}{K}$ such choices, which gives us the resulting inequality. ■

B.2 Proof of Theorem 8

Proof Instead of the event Υ, we define a subset of event Υ that is more useful. Let ξ be the event that, using $\boldsymbol{S}$, Algorithm 1 correctly identifies, for all $i,j \in S^c$, whether X_i is conditionally independent of X_j or not given $\boldsymbol{X}_S$. If each of these $\binom{p-|S|}{2}$ pairs of conditional independence relations are identified correctly, Algorithm 1 will be able to correctly identify whether or not $X_u \perp X_v \mid \boldsymbol{X}_S$ is faithful. Thus, if ξ occurs, Υ occurs as well.

There are two types of events that occurs in the complement event space ξ^c of ξ. The first event or error, is when $X_i \perp X_j \mid \boldsymbol{X}_S$, but Algorithm 1 outputs this relation conditionally dependent. We name this event $\xi_{ij}^{(1)}$. Thus, the event $\xi_{ij}^{(1)}$ occurs when $\boldsymbol{\Sigma}(i,j \mid S) = 0$ but $|\widehat{\boldsymbol{\Sigma}}(i,j \mid S)| \geq \alpha$, where $\widehat{\boldsymbol{\Sigma}}(i,j \mid S)$ is defined according to (10). The second type of error occurs when $X_i \not\perp X_j \mid \boldsymbol{X}_S$ but Algorithm 1 outputs this relation conditionally independent. Let this event be $\xi_{ij}^{(2)}$. Event $xi_{ij}^{(2)}$ occurs when $\boldsymbol{\Sigma}(i,j \mid S) \neq 0$, but $|\widehat{\boldsymbol{\Sigma}}(i,j \mid S)| \leq \alpha$.

As a result, we have

$$\mathbb{P}\left(\xi^c\right) = \mathbb{P}\left(\bigcup_{i,j\in S^c, X_i\perp X_j|\boldsymbol{X}_S}\xi_{ij}^{(1)}\right) + \mathbb{P}\left(\bigcup_{i,j\in S^c, X_i\not\perp X_j|\boldsymbol{X}_S}\xi_{ij}^{(2)}\right). \tag{59}$$

We bound the first term of the expression on the right hand side of the equation. When $\xi_{ij}^{(1)}$ occurs for some i and j, we immediately have

$$\left|\widehat{\mathbf{\Sigma}}(i,j \mid S) - \mathbf{\Sigma}(i,j \mid S)\right| = \left|\widehat{\mathbf{\Sigma}}(i,j \mid S)\right| \geq \alpha. \tag{60}$$

This gives us the upper bound on the probability

$$\mathbb{P}\left(\bigcup_{i,j\in S^c, X_i \perp X_j \mid \boldsymbol{X}_S} \xi_{ij}^{(1)}\right) \leq \mathbb{P}\left(\max_{i,j\in S^c}\left|\widehat{\mathbf{\Sigma}}(i,j \mid S) - \mathbf{\Sigma}(i,j \mid S)\right| \geq \alpha\right). \tag{61}$$

Applying Corollary 17, we have

$$\mathbb{P}\left(\bigcup_{i,j\in S^c, X_i \perp X_j \mid \boldsymbol{X}_S} \xi_{ij}^{(1)}\right) \leq 4(p-|S|)^2 \exp\left\{-\frac{(n-|S|)\alpha^2}{6\lambda_{\max}^2 + 4\alpha\lambda_{\max}}\right\}, \tag{62}$$

Since $\alpha = \beta/2$, for

$$n \geq \frac{24\lambda_{\max}^2 + 8\beta\lambda_{\max}}{\beta^2}\left(\log 8 + 2\log(p-|S|) + \log\frac{1}{\epsilon}\right) + |S|, \tag{63}$$

we have

$$\mathbb{P}\left(\bigcup_{i,j\in S^c, X_i \perp X_j \mid \boldsymbol{X}_S} \xi_{ij}^{(1)}\right) \leq \frac{\epsilon}{2}. \tag{64}$$

The second term is bounded similarly. Since $|\mathbf{\Sigma}(i,j \mid S)| \geq \beta$ when $\mathbf{\Sigma}(i,j \mid S) \neq 0$, we have

$$\mathbb{P}\left(\bigcup_{i,j\in S^c, X_i \not\perp X_j \mid \boldsymbol{X}_S} \xi_{ij}^{(2)}\right) \leq \mathbb{P}\left(\max_{i,j\in S^c}\left|\widehat{\mathbf{\Sigma}}(i,j \mid S) - \mathbf{\Sigma}(i,j \mid S)\right| \geq \beta - \alpha\right). \tag{65}$$

Since $\beta - \alpha = \beta/2$, we again have

$$\mathbb{P}\left(\bigcup_{i,j\in S^c, X_i \not\perp X_j \mid \boldsymbol{X}_S} \xi_{ij}^{(2)}\right) \leq \frac{\epsilon}{2}, \tag{66}$$

for n satisfying (63). Substituting equations (64) and (66) into (59), we get

$$\mathbb{P}(\Upsilon) \geq \mathbb{P}(\xi) \geq 1 - \epsilon, \tag{67}$$

as required.

■

B.3 Proof of Theorem 10

Proof The proof follows closely the proof of Theorem 8. In the proof of Theorem 8, the separator set S is fixed, and the success of the algorithm is based on correctly deciding whether each of the conditional relation between X_i and X_j given $\boldsymbol{X}_S$ is dependent or independent for all $i,j \in S^c$. In learning the structure of a weakly K-separable graph however, we have to run through various separator sets S and apply our faithfulness algorithm on it. This means that if we correctly decide whether X_i and X_j given $\boldsymbol{X}_S$ is conditionally dependent or independent for all possible i,j and S, that is sufficient for us to correctly determine $\mathcal{E}$.

Instead of applying Corollary 17, we apply Corollary 18, which gives us

$$\mathbb{P}\left(\hat{\mathcal{E}} \neq \mathcal{E}\right) \leq \epsilon, \tag{68}$$

for

$$n \geq \frac{24\lambda_{\max}^2 + 8\beta\lambda_{\max}}{\beta^2}\left(\log 8 + (K+2)\log p + \log\frac{1}{\epsilon}\right) + K. \tag{69}$$

■

B.4 Proof of Theorem 12

Lemma 19 *Let i and j be separate nodes in $\mathcal{W}$ and let S separate i and j in the graph $\mathcal{G}_{-(i,j)}$. Let $L < 1$ be a positive constant, and let*

$$\epsilon \in \left(0, \frac{L\lambda_{\min}^2 + 2\lambda_{\max}(1+2\lambda_{\max})}{(1-L)L\lambda_{\min}^4} \cdot \min\left\{1, \frac{L\lambda_{\min}^2}{2(1+2\lambda_{\max})}\right\}\right]. \tag{70}$$

Also, define

$$\delta = \frac{(1-L)L\lambda_{\min}^4}{L\lambda_{\min}^2 + 2\lambda_{\max}(1+2\lambda_{\max})} \cdot \epsilon. \tag{71}$$

If we have

$$\left|\widehat{\mathbf{\Sigma}}(i,j \mid S) - \mathbf{\Sigma}(i,j \mid S)\right|, \left|\widehat{\mathbf{\Sigma}}(i,i \mid S) - \mathbf{\Sigma}(i,i \mid S)\right|, \left|\widehat{\mathbf{\Sigma}}(j,j \mid S) - \mathbf{\Sigma}(j,j \mid S)\right| \leq \delta, \tag{72}$$

then

$$\left|\widehat{\mathbf{\Omega}}_{ij} - \mathbf{\Omega}_{ij}\right| \leq \epsilon. \tag{73}$$

Proof For convenience, we denote $\mathbf{\Sigma}(i,j \mid S), \mathbf{\Sigma}(i,i \mid S)$, and $\mathbf{\Sigma}(j,j \mid S)$ by a, b, and c. We will also denote $\widehat{\mathbf{\Sigma}}(i,j \mid S), \widehat{\mathbf{\Sigma}}(i,i \mid S)$, and $\widehat{\mathbf{\Sigma}}(j,j \mid S)$ by $\hat{a}, \hat{b}$, and $\hat{c}$. Let D be the determinant $bc - a^2$ and let $\hat{D} = \hat{b}\hat{c} - \hat{a}^2$. Rewriting (72), we have

$$\max\left\{|\hat{a} - a|, |\hat{b} - b|, |\hat{c} - c|\right\} \leq \delta. \tag{74}$$

We first bound $|\hat{D} - D|$. We have

$$\begin{aligned} |\hat{D} - D| &= |(\hat{b}\hat{c} - \hat{a}^2) - (bc - a^2)| \\ &\leq |\hat{b}\hat{c} - bc| + |\hat{a}^2 - a^2| \end{aligned} \tag{75}$$

The first term is bounded by

$$\begin{aligned} |\hat{b}\hat{c} - bc| &= |(\hat{b} - b)(\hat{c} - c) + c(\hat{b} - b) + b(\hat{c} - c)| \\ &\leq \delta^2 + |c|\delta + |b|\delta \\ &\leq \delta(1 + 2\lambda_{\max}), \end{aligned} \tag{76}$$

where in the last inequality, we make use of the condition that $\delta \leq 1$, which is derived from (70) and (71), and the bound on the matrix entries

$$\max\{|a|, |b|, |c|\} \leq \|\mathbf{\Sigma}_{Q|S}\|_2 \leq \|\mathbf{\Sigma}\|_2 = \lambda_{\max}. \tag{77}$$

We can then bound the second term $|\hat{a} - a|$ similarly, which gives us

$$|\hat{D} - D| \leq 2\delta(1 + 2\lambda_{\max}). \tag{78}$$

From (70), we have

$$\delta \leq \frac{L\lambda_{\min^2}}{2(1 + 2\lambda_{\max})}, \tag{79}$$

which give us

$$|\hat{D}| \geq |D| - L\lambda_{\min}^2 \geq (1 - L)|D|, \tag{80}$$

where the last inequality follows from the fact that the two eigenvalues of $\mathbf{\Sigma}_{Q|S}$ are both bounded below by $\lambda_{\min}$ by the eigenvalue interlacing property.

We are now ready to establish the upper bound for $|\widehat{\mathbf{\Omega}}_{ij} - \mathbf{\Omega}_{ij}|$. We have

$$\begin{aligned} |\widehat{\mathbf{\Omega}}_{ij} - \mathbf{\Omega}_{ij}| &= \left|\frac{\hat{a}}{\hat{D}} - \frac{a}{D}\right| \\ &= \frac{|\hat{a}D - a\hat{D}|}{|\hat{D}||D|} \\ &\leq \frac{|D||\hat{a} - a| + |a||\hat{D} - D|}{|\hat{D}||D|} \\ &\leq \frac{1}{(1 - L)|D|}\left(\delta + \frac{|a|}{L|D|} \cdot 2\delta(1 + 2\lambda_{max})\right) \\ &\leq \frac{1}{(1 - L)\lambda_{\min}^2}\left(\delta + \frac{\lambda_{\max}}{L\lambda_{\min}^2} \cdot 2\delta(1 + 2\lambda_{\max})\right). \end{aligned} \tag{81}$$

Substituting δ from (70), we get

$$|\widehat{\mathbf{\Omega}}_{ij} - \mathbf{\Omega}_{ij}| \leq \epsilon, \tag{82}$$

which is the result of this lemma. ■

We are now ready to prove Theorem 12.

Proof We again define δ according to (71), which can be rewritten as

$$\delta = \frac{\epsilon}{C_1}. \tag{83}$$

Applying Lemma 19, we have

$$\mathbb{P}\left(\max_{\substack{i,j \in \mathcal{W}, i \neq j, \\ S \in \Lambda_0(i,j)}} |\widehat{\mathbf{\Omega}}_{ij} - \mathbf{\Omega}_{ij}| \geq \epsilon\right) \leq \mathbb{P}\left(\max_{\substack{S \subset \mathcal{W}, |S| = K, \\ i,j \in S^c}} \left|\widehat{\mathbf{\Sigma}}(i,j \mid S) - \mathbf{\Sigma}(i,j \mid S)\right| \geq \delta\right), \tag{84}$$

for

$$\epsilon \in \left(0, C_1 \cdot \min\left\{1, \frac{L\lambda_{\min}^2}{2(1 + 2\lambda_{\max})}\right\}\right]. \tag{85}$$

Using the result of Corollary 18, we have

$$\mathbb{P}\left(\max_{\substack{S \subset \mathcal{W}, |S| = K, \\ i,j \in S^c}} \left|\widehat{\mathbf{\Sigma}}(i,j \mid S) - \mathbf{\Sigma}(i,j \mid S)\right| \geq \delta\right) \leq 4p^{K+2} \exp\left\{-\frac{(n - K)\delta^2}{6\lambda_{\max}^2 + 4\delta\lambda_{\max}}\right\}. \tag{86}$$

Combining the domains for ϵ, we get the result. ■

References

A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky. High-dimensional gaussian graphical model selection: walk-summability and local separation criterion. *J. Machine Learning Research*, 13:2293–2337, Aug 2012.

A. Becker, D. Geiger, and C. Meek. Perfect tree-like markovian distributions. *Probability and Mathematical Statistics*, 25(2):231–239, 2005.

E. Belilovskym, G. Varoquaux, and M. B. Blaschko. Testing for differences in gaussian graphical models: Applications to brain connectivity. In *Advances in Neural Information Processing Systems*, Dec 2016.

F. Cicalese and M. Melanič. Graphs of separability at most 2. *Discrete Applied Mathematics*, 160 (6):685–696, April 2012.

O. Dalal and B. Rajaratnam. Sparse gaussian graphical model estimation via alternating minimization. *Biometrika*, 104(2):379–395, 2017.

R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

M. L. Eaton. *Multivariate Statistics: A Vector Space Approach*. Wiley, 2007.

S. Epskamp, L. J. Waldorp, R. Mõttus, and D. Borsboom. The gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, pages 1–28, 2018.

M. Frydenberg. Marginalisation and collapsibility in graphical interaction models. *Annals of Statistics*, 18:790–805, 1990.

M. Isard. Pampas: real-valued graphical models for computer vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun 2003.

G. Kauermann. On a dualization of graphical gaussian models. *Scandinavian Journal of Statistics*, 23(1):105–116, 1996.

P. Koldanov, A. Koldanov, V. Kalyagin, and P. Pardalos. Uniformly most powerful unbiased test for conditional independence in gaussian graphical model. *Statistics and Probability Letters*, 122: 90–95, 2017.

S. L. Lauritzen. *Graphical models*. Oxford University Press, New York, 1996.

F. Liang, Q. Song, and P. Qiu. An equivalent measure of partial correlation coefficients for high-dimensional gaussian graphical models. *Journal of the American Statistical Association*, 110: 1248–1265, 2015.

S. Lin, C. Uhler, B. Sturmfels, and P. Bühlmann. Hypersurfaces and their singularities in partial correlation testing. *Preprint*.

D. Malouche and B. Rajaratnam. Gaussian covariance faithful markov trees. *Technical report, Department of Statistics, Stanford University*, 2009.

C. Meek. Strong completeness and faithfulness in bayesian networks. In *Proceedings of the eleventh international conference on uncertainty in artificial intelligence*, 1995.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

K. Mohan, M. J. Chung, S. Han, D. Witten, S. Lee, and M. Fazel. Structured learning of gaussian graphical models. In *Advances in Neural Information Processing Systems*, Dec 2012.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High dimensional covariance estimation by minimizing ℓ-1 penalized log-determinant divergence. *Electronic Journal in Statistics*, 4:935–980, 2011.

Z. Ren, T. Sun, C. Zhang, and H. Zhou. Asymptotic normality and optimalities in estimation of large gaussian graphical model. *Annals of Statistics*, 43(3):991–1026, 2015.

S. Ryali, T. Chen, K. Supekar, and V. Menon. Estimation of functional connectivity in fmri data using stability selection-based sparse partial correlation with elastic net penalty. *Neuroimage*, 59 (4):3852–3861, February 2012.

K. Sadeghi. Faithfulness of probability distributions and graphs. *Journal of Machine Learning Research*, 18(148):1–29, 2017.

L. Smith. Some interlacing properties of schur complement of a hermitian matrix. *Linear Algebra Appl.*, 177:137–144, 1992.

P. Spirites, C. Glymore, and R. Scheines. *Causation, prediction and search*. Springer Verlag, New York, 1993.

E. B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 2006.

C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of faithfulness assumption in causal inference. *Annals of Statistics*, 41:436–463, 2013.

W. N. van Wieringen, C. F. W. Peeters, R. X. de Menezes, and M. A. van de Wiel. Testing for pathway (in)activation by using gaussian graphical models. *Journal of Royal Statistical Society*, 2018.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, 1990.

R. Wu, R. Srikant, and J. Ni. Learning loosely connected markov random fields. *Stochastic Systems*, 3, 2013.

S. Zhu. A short note on the tail bound of wishart distribution. *ArXiv e-prints*, arXiv:1212.5860, 2012.

Journal of Machine Learning Research 20 (2019) 1-30 Submitted 6/18; Revised 5/19; Published 7/19

A Representer Theorem for Deep Neural Networks

Michael Unser MICHAEL.UNSER@EPFL.CH
Biomedical Imaging Group,
École polytechnique fédérale de Lausanne (EPFL),
CH-1015 Lausanne, Switzerland

Editor: Lorenzo Rosasco

Abstract

We propose to optimize the activation functions of a deep neural network by adding a corresponding functional regularization to the cost function. We justify the use of a second-order total-variation criterion. This allows us to derive a general representer theorem for deep neural networks that makes a direct connection with splines and sparsity. Specifically, we show that the optimal network configuration can be achieved with activation functions that are nonuniform linear splines with adaptive knots. The bottom line is that the action of each neuron is encoded by a spline whose parameters (including the number of knots) are optimized during the training procedure. The scheme results in a computational structure that is compatible with existing deep-ReLU, parametric ReLU, APL (adaptive piecewise-linear) and MaxOut architectures. It also suggests novel optimization challenges and makes an explicit link with ℓ_1 minimization and sparsity-promoting techniques.

Keywords: splines, regularization, sparsity, learning, deep neural networks, activation functions

1. Introduction

The basic regression problem in machine learning is to find a parametric representation of a function $f : \mathbb{R}^N \to \mathbb{R}$ given a set of data points $(\boldsymbol{x}_m, y_m) \in \mathbb{R}^{N+1}$ such that $f(\boldsymbol{x}_m)$ is close to y_m for $m = 1, \ldots, M$ in an appropriate sense (Bishop, 2006). Classically, there are two steps involved. The first is the *design*, which can be abstracted in the choice of a given parametric class of functions $\boldsymbol{x} \mapsto f(\boldsymbol{x}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ encodes the parameters. For instance, $f(\boldsymbol{x}|\boldsymbol{\theta})$ could be a neural network with weights $\boldsymbol{\theta}$. The second is the *training*, which basically amounts to an interpolation/approximation problem where the chosen model is fit to the data. In practice, the optimal parameter $\boldsymbol{\theta}_0$ is determined via the functional minimization

$$\boldsymbol{\theta}_0 = \arg\min_{\boldsymbol{\theta}} \sum_{m=1}^{M} E\big(y_m, f(\boldsymbol{x}_m|\boldsymbol{\theta})\big), \tag{1}$$

where $E : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ is a convex error function that quantifies the discrepancy of the fit to the data. A classical choice is $E\big(y_m, f(\boldsymbol{x}_m|\boldsymbol{\theta})\big) = |y_m - f(\boldsymbol{x}_m|\boldsymbol{\theta})|^2$, which yields the least-squares solution.

The most delicate step is the design, because it has to deal with two conflicting requirements. First is the desire for universality, meaning that the parametric model $f(\boldsymbol{x}|\boldsymbol{\theta})$ should be flexible enough to allow for the faithful representation of a large class of functions—ideally, the complete family of continuous functions $\mathbb{R}^N \to \mathbb{R}$, as the dimensionality of $\boldsymbol{\theta}$ goes to infinity. Second is the quest for parsimony, meaning that the model should have a small number of parameters, which is expected to lead to an increase in robustness and trustworthiness.

This work aims at unifying the design of neural networks based on variational principles inspired by kernel methods. To set up the stage, we now briefly review two relevant approaches to supervised learning.

1.1. Kernel Methods

A kernel estimator is a linear model with adjustable parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M) \in \mathbb{R}^M$ and predefined data centers $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M \in \mathbb{R}^N$ of the form

$$f(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{m=1}^{M} \theta_m h(\boldsymbol{x}, \boldsymbol{x}_m), \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the input variable of the model and where $h : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is a positive-definite kernel, a preferred choice being the Gaussian kernel $h(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{e}^{-\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2/\sigma^2}$ (Hofmann et al., 2008; Alvarez et al., 2012). This expansion is at the heart of the whole class of kernels methods, including radial-basis functions and support-vector machines (Schölkopf et al., 1997; Vapnik, 2013; Schölkopf and Smola, 2002).

The elegance of kernel estimators lies in that they can be justified based on regularization theory (Poggio and Girosi, 1990; Evgeniou et al., 2000; Poggio and Smale, 2003). The incentive there is to remove some of the arbitrariness of model selection by formulating the learning task as a global minimization problem that takes care of the design and training jointly. The property that makes such an integrated approach feasible is that any Hilbert space $\mathcal{H}$ of continuous functions on $\mathbb{R}^N$ has a unique *reproducing kernel* $h_{\mathcal{H}} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ such that (*i*) $h_{\mathcal{H}}(\cdot, \boldsymbol{x}_m) \in \mathcal{H}$; and (*ii*) $\langle f, h_{\mathcal{H}}(\cdot, \boldsymbol{x}_m)\rangle_{\mathcal{H}} = f(\boldsymbol{x}_m)$ for any $\boldsymbol{x}_m \in \mathbb{R}^N$ and $f \in \mathcal{H}$ (Aronszajn, 1950). The idea, then, is to formulate the "regularized" version of Problem (1) as

$$f_{\mathrm{RKHS}} = \arg\min_{f \in \mathcal{H}} \left(\sum_{m=1}^{M} E\big(y_m, f(\boldsymbol{x}_m)\big) + \lambda \|f\|_{\mathcal{H}}^2 \right), \tag{3}$$

where the second term penalizes solutions with a large $\|\cdot\|_{\mathcal{H}}$-norm and $\lambda \in \mathbb{R}^+$ is an adjustable tradeoff factor. Under the assumption that the loss function E is convex, the representer theorem (Kimeldorf and Wahba, 1971; Schölkopf et al., 2001; Schölkopf and Smola, 2002) states that the solution of (3) exists, is unique, and is such that $f_{\mathrm{RKHS}} \in \mathrm{span}\{h_{\mathcal{H}}(\cdot, \boldsymbol{x}_m)\}_{m=1}^{M}$. This ultimately results in the same linear expansion as (2). The argument also applies the other way round since any positive-definite kernel h specifies a unique reproducing-kernel Hilbert space (RKHS) $\mathcal{H}$, which then provides the regularization functional $\|f\|_{\mathcal{H}}^2$ in (3) that is matched to the kernel estimator specified by (2).

The other remarkable feature of kernel expansions is their *universality*, under mild conditions on h (Micchelli et al., 2006). In other words, one has the guarantee that the generic linear model of (2) can reproduce *any* continuous function $f : \mathbb{R}^N \to \mathbb{R}$ to a desired degree of accuracy by including sufficiently many centers, with the error vanishing as $M \to \infty$. Moreover, because of the tight connection between kernels, RKHS, and splines (de Boor and

©2019 Michael Unser.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-418.html.

Lynch, 1966; Micchelli, 1986; Wahba, 1990), one can invoke standard results in approximation theory to obtain quantitative estimates of the approximation error of smooth functions as a function of M and of the widest gap between data centers (Wendland, 2005). Finally, there is a well-known link between kernel methods derived from regularization theory and neural networks, albeit "shallow" ones that involve a single nonlinear layer (Poggio and Girosi, 1990).

1.2. Deep Neural Networks

While kernel methods have been a major (and winning) player in machine learning since the mid '90s, they have been recently outperformed by deep neural networks (DNNs) in many real-world applications such as image classification (Krizhevsky et al., 2012), speech recognition (Hinton et al., 2012), and image segmentation (Ronneberger et al., 2015).

The leading idea of deep learning is to build more powerful learning architectures via the stacking/composition of simpler entities (see the review papers by LeCun, Bengio and Hinton (LeCun et al., 2015), Schmidhuber (Schmidhuber, 2015), and Goodfellow's recent textbook (Goodfellow et al., 2016) for more detailed explanations). In this work, we focus on the popular class of feedforward networks that involve a layered composition of affine transformations (linear weights) and pointwise nonlinearities. The deep structure of such a network is specified by its *node descriptor* $(N_0, N_1, \ldots, N_L)$, where L is the total number of layers (depth of the network) and N_ℓ is the number of neurons at the ℓth layer. The action of a (scalar) neuron (or node) indexed by (n, ℓ) is described by $\sigma(\mathbf{w}_{n,\ell}^T \boldsymbol{x} - b_{n,\ell})$ where $\boldsymbol{x} \in \mathbb{R}^{N_{\ell-1}}$ denotes the multivariate input of the neuron, $\sigma : \mathbb{R} \to \mathbb{R}$ is a predefined activation function such as a sigmoid or a ReLU (rectified linear unit), $\mathbf{w}_{n,\ell} \in \mathbb{R}^{N_{\ell-1}}$ a set of linear weights, and $b_{n,\ell} \in \mathbb{R}$ an additive bias. The outputs of layer ℓ are then fed as the inputs of layer $(\ell + 1)$ for $\ell = 1, \ldots, L-1$.

To obtain a global description, we group the neurons within a given layer ℓ and specify two corresponding vector-valued maps.

1. Linear step $\boldsymbol{f}_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ (affine transformation)

$$\boldsymbol{f}_\ell : \boldsymbol{x} \mapsto \boldsymbol{f}_\ell(\boldsymbol{x}) = \mathbf{W}_\ell \boldsymbol{x} - \mathbf{b}_\ell \tag{4}$$

   with weighting matrix $\mathbf{W}_\ell = [\mathbf{w}_{1,\ell} \cdots \mathbf{w}_{N_\ell,\ell}]^T \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and bias vector $\mathbf{b}_\ell = (b_{1,\ell}, \ldots, b_{N_\ell,\ell}) \in \mathbb{R}^{N_\ell}$.

2. Nonlinear step $\boldsymbol{\sigma}_\ell : \mathbb{R}^{N_\ell} \to \mathbb{R}^{N_\ell}$ (activation functions)

$$\boldsymbol{\sigma}_\ell : \boldsymbol{x} = (x_1, \ldots, x_{N_\ell}) \mapsto \boldsymbol{\sigma}_\ell(\boldsymbol{x}) = \big(\sigma_{1,\ell}(x_1), \ldots, \sigma_{N_\ell,\ell}(x_{N_\ell})\big) \tag{5}$$

   with the possibility to adapt the scalar activation functions $\sigma_{n,\ell}$ on a per-node basis.

This allows us to describe the overall action of the full L-layer deep network by

$$\mathbf{f}_{\text{deep}}(\boldsymbol{x}) = (\boldsymbol{\sigma}_L \circ \boldsymbol{f}_L \circ \boldsymbol{\sigma}_{L-1} \circ \cdots \circ \boldsymbol{\sigma}_2 \circ \boldsymbol{f}_2 \circ \boldsymbol{\sigma}_1 \circ \boldsymbol{f}_1)(\boldsymbol{x}), \tag{6}$$

which makes its compositional structure explicit. The design step therefore consists in fixing the architecture of the deep neural net: One must specify $(N_0, N_1, \ldots, N_L)$ together with

the activation functions $\boldsymbol{\sigma}_\ell : \mathbb{R}^{N_\ell} \to \mathbb{R}^{N_\ell}$. The activations are traditionally chosen to be not only the same for all neurons within a layer, but also the same across layers. This results in a computational structure with adjustable parameters $\boldsymbol{\theta} = (\mathbf{W}_1, \ldots, \mathbf{W}_L, \mathbf{b}_1, \ldots, \mathbf{b}_L)$ (weights of the linear steps). These are then set during training via the minimization of (1), which is achieved by stochastic gradient descent with efficient error backpropagation (Rumelhart et al., 1986).

While researchers have considered a variety of possible activation functions, such as the traditional sigmoid, a preferred choice that has emerged over the years is the rectified linear unit $\mathrm{ReLU}(x) = \max(x, 0)$ (Glorot et al., 2011). The reasons that support this choice are multiple. The initial motivation was to promote sparsity (in the sense of a decrease in the number of active units), capitalizing on the property that ReLU acts as a gate and works well in combination with ℓ_1-regularization (Glorot et al., 2011). Second is the empirical observation that the training of very deep networks is much faster if the hidden layers are composed of ReLU activation functions (LeCun et al., 2015). Last but not least is the connection between deep ReLU networks and splines—to be further developed in this paper.

A key observation is that a deep ReLU network implements a multivariate input-output relation that is continuous and piecewise-linear (CPWL) (Montufar et al., 2014). This remarkable property is due to the ReLU itself being a linear spline, which has prompted Poggio et al. to interpret deep neural networks as hierarchical splines (Poggio et al., 2015). Moreover, it has been shown that any CPWL function admits a deep ReLU implementation (Wang and Sun, 2005; Arora et al., 2016), which is quite significant since the CPWL family has universal-approximation properties.

The ability of splines to effectively represent arbitrary (univariate) functions (de Boor, 1978; Schumaker, 1981; Unser, 1999) has also been exploited at the more local level of a neuron/node in a network. Several authors have proposed to use spline-related parametric models to optimize the shape of neural activation units. Existing designs include B-spline receptive fields (Lane et al., 1991), Catmul-Rom splines (Vecci et al., 1998), cubic spline activations (Guarnieri et al., 1999), adaptive piecewise-linear (APL) units (Agostinelli et al., 2015), and smooth piecewise-polynomial functions (Hou et al., 2017).

1.3. Road Map

Our purpose in this paper is to strengthen the connection between splines and multilayer ReLU networks even further. To that end, we formulate the design of a deep neural network globally within the context of regularization theory, in direct analogy with the variational formulation of kernel estimators given by (3). The critical aspect, of course, is the selection of an appropriate regularization functional which, for reasons that will be exposed next, will take us outside of the traditional realm of RKHS.

Having set the deep architecture of the neural network, we then formulate the training as a global optimization task whose outcome is a combined set of optimal neuronal activation functions and linear weights. The foundational role of the representer theorem (Theorem 4) is that it will provide us with the parametric representation of the optimal activations, which can then be leveraged to obtain a numerical implementation that is compatible with current architectures; in particular, the popular deep RELU networks.

2. From Deep Neural Networks to Deep Splines

Given the generic structure of a deep neural network, we are interested in investigating the possibility of optimizing the shape of the activation function(s) on a node-by-node basis. We now show how this can be achieved within the context of infinite-dimensional regularization theory.

2.1. Choice of Regularization Functional

For practical relevance, the scheme should favor simple solutions such as identity or linear scaling. This will retain the possibility of performing a classical linear regression. It is also crucial that the activation function σ be differentiable to be compatible with the chain rule when the backpropagation algorithm is used to train the network. Lastly, we want to promote activation functions that are locally linear (such as the ReLU) since these appear to work best in practice. If the two aforementioned constraints are satisfied, then the activation function is CPWL. As this property is conserved through (multivariate) composition, it implies that the resulting map $\boldsymbol{f}_{\text{deep}} : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ is CPWL as well, which is highly desirable for applications (Strang, 2018). Hence, an idealized solution would be a function $\sigma : \mathbb{R} \to \mathbb{R}$ whose second derivative vanishes almost everywhere.

As measure of sparsity, we use the "total-variation" norm $\|\cdot\|_{\mathcal{M}}$ associated with the Banach space

$$\mathcal{M}(\mathbb{R}) = \{f \in \mathcal{S}'(\mathbb{R}) : \|f\|_{\mathcal{M}} \triangleq \sup_{\varphi \in \mathcal{S}(\mathbb{R}) : \|\varphi\|_\infty \leq 1} \langle f, \varphi \rangle < \infty\}, \tag{7}$$

where $\mathcal{S}'(\mathbb{R})$ is Schwartz' space of tempered distributions, which is the continuous dual of $\mathcal{S}(\mathbb{R})$, the space of smooth and rapidly decreasing test functions on $\mathbb{R}$. Note that our definition of $\|\cdot\|_{\mathcal{M}}$ (by duality) is equivalent to the notion of total variation used in measure theory (Rudin, 1987). The critical point for us is that the latter is a slight extension of the L_1-norm: The basic property is that $\|f\|_{L_1} \triangleq \int_{\mathbb{R}} |f(x)| \mathrm{d}x = \|f\|_{\mathcal{M}}$ for any $f \in L_1(\mathbb{R})$, which implies that $L_1(\mathbb{R}) \subseteq \mathcal{M}(\mathbb{R})$. However, the shifted Dirac distribution $\delta(\cdot - x_m) \notin L_1(\mathbb{R})$ for any shift $x_m \in \mathbb{R}$, while $\delta(\cdot - x_m) \in \mathcal{M}(\mathbb{R})$ with $\|\delta(\cdot - x_m)\|_{\mathcal{M}} = 1$, which shows that the space $\mathcal{M}(\mathbb{R})$ is (slightly) larger than $L_1(\mathbb{R})$.

To favor neuronal activation functions $\sigma : \mathbb{R} \to \mathbb{R}$ with "sparse" second derivatives, we shall therefore impose a bound on their second total variation, defined as

$$\mathrm{TV}^{(2)}(\sigma) \triangleq \|\mathrm{D}^2 \sigma\|_{\mathcal{M}} = \sup_{\varphi \in \mathcal{S}(\mathbb{R}) : \|\varphi\|_\infty \leq 1} \langle \mathrm{D}^2 \sigma, \varphi \rangle \tag{8}$$

where $\mathrm{D}^2 = \frac{\mathrm{d}^2}{\mathrm{d}x^2}$ is the second-derivative operator. The connection with ReLU is that $\mathrm{D}^2\mathrm{ReLU} = \delta$, which confirms that the ReLU activation function is intrinsically sparse with $\mathrm{TV}^{(2)}(\mathrm{ReLU}) = 1$.

Since our formulation involves a joint optimization of all network components, it is important to decouple the effect of the various stages. The only operation that is common to linear transformations and pointwise nonlinearities is a linear scaling, which is therefore transferable from one level to the next. Since most regularization schemes are scale-sensitive, it is essential to prevent such a transfer. We achieve this by restricting the class of admissible

weight vectors $\mathbf{w}_{n,\ell}$ acting on a given node indexed by (n, ℓ) to those that have a unit norm. In other words, we shall normalize the scale of all linear modules with the introduction of the new variable $\mathbf{u}_{n,\ell} = \mathbf{w}_{n,\ell} / \|\mathbf{w}_{n,\ell}\|$.

2.2. Topological Properties of the Search Space

The formal definition of our native space (i.e., the space over which the optimization is performed) is

$$\mathrm{BV}^{(2)}(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} \ \text{ s.t. } \ \|\mathrm{D}^2 f\|_{\mathcal{M}} < \infty\}, \tag{9}$$

which is the class of functions with bounded second total variation.

To provide the proper mathematical context for our derivations, we also need to specify a suitable Banach topology for $\mathrm{BV}^{(2)}(\mathbb{R})$. In particular, doing so allows us to show that $\mathrm{BV}^{(2)}(\mathbb{R})$ has universal approximation properties. An important property in this respect is that $\mathrm{BV}^{(2)}(\mathbb{R})$ is a subspace of

$$C_{\mathrm{b},1}(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} \text{ s.t. } f \text{ is continuous and } \|f\|_{\infty,1} = \sup_{x \in \mathbb{R}} (1 + |x|)^{-1} |f(x)| < \infty\}, \tag{10}$$

which is the space of continuous function with at most linear growth. The key features of $\mathrm{BV}^{(2)}(\mathbb{R})$ that constitute the theoretical backbone of our formulation can then be summarized as follows.

Theorem 1 *The space* $\mathrm{BV}^{(2)}(\mathbb{R})$ *equipped with the norm*

$$\|f\|_{\mathrm{BV}^{(2)}} \triangleq \|\mathrm{D}^2 f\|_{\mathcal{M}} + \sqrt{|f(0)|^2 + |f(1) - f(0)|^2} \tag{11}$$

has the following properties:

1. *It is a Banach space.*
2. *The Dirac sampling functional* $\delta(\cdot - x_m) : f \mapsto f(x_m)$ *is weak* continuous on* $\mathrm{BV}^{(2)}(\mathbb{R})$ *for any* $x_m \in \mathbb{R}$. *Moreover, it satisfies the continuity bound*

$$|f(x_m)| = |\langle \delta(\cdot - x_m), f \rangle| \leq (1 + |x_m|) \ \|f\|_{\mathrm{BV}^{(2)}},$$

   for any $f \in \mathrm{BV}^{(2)}(\mathbb{R})$.
3. *Continuous embeddings:* $\mathcal{S}(\mathbb{R}) \subseteq \mathrm{BV}^{(2)}(\mathbb{R}) \subseteq C_{\mathrm{b},1}(\mathbb{R})$.

The reader is referred to Appendix A for the explanation of the concept of weak* continuity, and to Appendices B and C for the details of the proof. Except for Property 1, which is a corollary of (Unser et al., 2017, Theorem 5), the results in Theorem 1 are new. Property 2 is crucial for the proof of Lemma 2 which, as we shall see, is fundamental to our argumentation. Property 3 is very relevant as well since it implies that $\mathrm{BV}^{(2)}(\mathbb{R})$ is rich enough to reproduce any continuous function—in the present case, any neuronal activation—with an arbitrary degree of precision. This can be deduced by transitivity from the universal approximation properties of Schwartz' space $\mathcal{S}(\mathbb{R})$, which even happens to be dense in $\mathcal{S}'(\mathbb{R})$ (Schwartz, 1966).

2.3. Supporting Optimality Results

As preparation for our representer theorem, we now present our lemma on the $\mathrm{TV}^{(2)}$-optimality of piecewise-linear interpolation. This enabling result is deduced from the general spline theory presented in (Unser et al., 2017), as detailed in Appendix D. We then provide arguments to disqualify the use of the more conventional Sobolev-type regularization.

The subproblem of interest is to search for the optimal interpolant of a series of data points within the native space $\mathrm{BV}^{(2)}(\mathbb{R})$.

Lemma 2 ($\mathrm{TV}^{(2)}$**-optimality of piecewise-linear interpolants)** *Consider a series of scalar data points $(x_m, y_m), m = 1, \ldots, M$ with $M > 2$ and $x_1 \neq x_2$. Then, under the hypothesis of feasibility (i.e., $y_{m_1} = y_{m_2}$ whenever $x_{m_1} = x_{m_2}$), the extremal points of the interpolation problem*

$$\arg \min_{f \in \mathrm{BV}^{(2)}(\mathbb{R})} \|\mathrm{D}^2 f\|_{\mathcal{M}} \quad s.t. \quad f(x_m) = y_m, m = 1, \ldots, M$$

are nonuniform splines of degree 1 (a.k.a. piecewise-linear functions) with no more than $(M-2)$ adaptive knots.

The proof, together with the relevant background in functional analysis, is given in the appendix. The feasibility hypothesis in Lemma 2 is not restrictive since a function returns a single value for each input point. We are aware of two antecedents to Lemma 2 (i.e., (Fisher and Jerome, 1975, Corollary 2.2), (Mammen and van de Geer, 1997, Proposition 1)); these earlier results, however, are not in a form suitable to our purpose because they restrict the domain of f to a finite interval. Moreover, in both cases, the authors implicitly assume the weak* continuity of the sampling functionals, which is not obvious a priori. In fact, it is the groundwork for establishing this property—namely, the characterization of the predual space $C_{0,\mathrm{D}^2}(\mathbb{R})$ in Theorem 11—that is the most involved part of our derivation (see Appendix B). The statement in Lemma 2 is also more precise because it yields the full solution set (as the convex hull of the extremal points) and gives a stronger bound on the maximum number of knots.

Lemma 2 implies that there exists an optimal interpolator, not necessarily unique, whose generic parametric form is given by

$$f_{\mathrm{spline}}(x) = b_1 + b_2 x + \sum_{k=1}^{K} a_k (x - \tau_k)_+, \tag{12}$$

where $(x)_+ \triangleq \max(x, 0) = \mathrm{ReLU}(x)$, with the caveat that the intrinsic spline descriptors, given by the (minimal) number K of knots and the knot locations $\tau_1, \ldots, \tau_K \in \mathbb{R}$ are not known beforehand. This means that these descriptors need to be optimized jointly with the expansion coefficients $\boldsymbol{b} = (b_1, b_2) \in \mathbb{R}^2$ and $\boldsymbol{a} = (a_1, \ldots, a_K) \in \mathbb{R}^K$. Ultimately, this translates into a solution that has a polygonal graph with breakpoints $f_{\mathrm{spline}}(\tau_k), k = 1, \ldots, K$ and that perfectly interpolates the data points otherwise, as shown in Figure 1.

Since $\mathrm{TV}^{(2)}$-regularization penalizes the variations of the derivative, it will naturally produce (sparse) solutions with a small number of knots. This means that an optimal spline will typically have fewer knots than there are data points, while the list $\{\tau_1, \ldots, \tau_K\}$

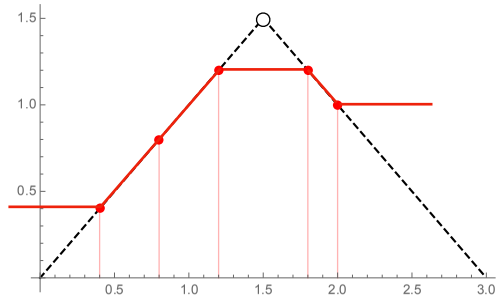


Figure 1: Conventional (solid line) vs. sparse (dashed line) piecewise-linear interpolants. The 5 data points are shown as dots; they coincide with the knots of the conventional interpolant. The sparse solution, by contrast, has a single knot at $\tau_1 = 1.5$ (circle), an argument value that is found in none of the data points.

of its knots, with $K < M$, may not necessarily be a subset of $\{x_1, \ldots, x_M\}$, as illustrated in Figure 1. This push towards model simplification (Occam's razor) is highly desirable. It distinguishes this formulation of splines from the more conventional one, which, in the case of interpolation, simply tells us "to connect the dots" with $K = M$ and $\tau_m = x_m$ for $m = 1, \ldots, M$ (see the solid-line curve in Figure 1).

It is well known that the classical linear interpolator is the solution of the variational problem in Proposition 3, which we like to see as the precursor of RKHS kernel methods (Prenter, 1975; Wahba, 1990).

Proposition 3 (Sobolev optimality of piecewise-linear interpolation) *Let the native space be the first-order Sobolev space $H^1(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} \,|\, \|\mathrm{D}f\|_{L_2}^2 + |f(0)|^2 < \infty\}$. Given a series of distinct data points $(x_m, y_m), m = 1, \ldots, M$, the interpolation problem*

$$\arg \min_{f \in H^1(\mathbb{R})} \int_{\mathbb{R}} |\mathrm{D}f(x)|^2 \mathrm{d}x \ s.t. \ \ f(x_m) = y_m, \ m = 1, \ldots, M$$

has a unique piecewise-linear solution that can be written as

$$s_2(x) = b_1 + \sum_{m=1}^{M} a_m (x - x_m)_+. \tag{13}$$

While the result is elegant and translates into a straightforward implementation, the scheme can be cumbersome for large data sets because the number of parameters in (13) increases with the number of data points. The other limitation is that the use of $\|\mathrm{D}f\|_{L_2}$-regularization disqualifies the simple linear solution $f(x) = ax$, which has an infinite cost.

As one may expect, there are also direct extensions of Lemma 2 and Proposition 3 for regularized least-squares approximations. Moreover, the distinction between the two types of solutions—smoothing splines (Schoenberg, 1964) vs. adaptive regression splines (Mammen and van de Geer, 1997)—is even more striking[1] for noisy-data fitting applications, which brings us back to our initial goal: the design and training of neural networks.

2.4. Representer Theorem for Deep Neural Networks

Our aim is to determine the optimal activation functions for a deep neural network in a task-dependent fashion. This problem is inherently ill-posed because activations are infinite-dimensional entities while we only have access to finite data. As in the case of interpolation, we resolve the ambiguity by imposing an appropriate form of regularization. Having singled out $\mathrm{TV}^{(2)}$ as the most favorable choice, we now proceed with the enunciation of our representer theorem for deep neural networks. We have purposefully stated the optimization problem in a generic form that is compatible with the current practice in DNN. Specifically, the cost function in (15) includes a standard data term that penalizes data misfit plus a regularization to constrain the values of the linear weights of the network (e.g., $R_\ell(\mathbf{U}_\ell) = \|\mathbf{U}_\ell\|_{\mathrm{F}}^2$ in the case of the popular weight-decay penalty). The novelty is the additional optimization over the neuronal activations $\sigma_{n,\ell}$ and the insertion of the $\mathrm{TV}^{(2)}$ term to regularize their shape.

Theorem 4 ($\mathrm{TV}^{(2)}$**-optimality of deep spline networks)** *Let the L-layer feedforward neural network $\mathbf{f}:\mathbb{R}^{N_0}\to\mathbb{R}^{N_L}$ with node descriptor $(N_0,N_1,\ldots,N_L)$ take the form*

$$\boldsymbol{x}\mapsto\mathbf{f}(\boldsymbol{x})=\left(\boldsymbol{\sigma}_L\circ\boldsymbol{\ell}_L\circ\boldsymbol{\sigma}_{L-1}\circ\cdots\circ\boldsymbol{\ell}_2\circ\boldsymbol{\sigma}_1\circ\boldsymbol{\ell}_1\right)(\boldsymbol{x}), \tag{14}$$

which is an alternating composition of the normalized linear transformations $\boldsymbol{\ell}_\ell:\mathbb{R}^{N_{\ell-1}}\to\mathbb{R}^{N_\ell},\boldsymbol{x}\mapsto\mathbf{U}_\ell\boldsymbol{x}$ with linear weights $\mathbf{U}_\ell=[\mathbf{u}_{1,\ell}\ \cdots\ \mathbf{u}_{N_\ell,\ell}]^T\in\mathbb{R}^{N_\ell\times N_{\ell-1}}$ such that $\|\mathbf{u}_{n,\ell}\|=1$ and the nonlinear activations $\boldsymbol{\sigma}_\ell:\mathbb{R}^{N_\ell}\to\mathbb{R}^{N_\ell},\boldsymbol{x}\mapsto\big(\sigma_{1,\ell}(x_1),\ldots,\sigma_{N_\ell,\ell}(x_n)\big)$ with $\sigma_{1,\ell},\ldots,\sigma_{N_\ell,\ell}\in\mathrm{BV}^{(2)}(\mathbb{R})$. Given a series of data points $(\boldsymbol{x}_m,\boldsymbol{y}_m)_{m=1}^M$, we then define the training problem

$$\arg\min_{(\mathbf{U}_\ell),(\sigma_{n,\ell}\in\mathrm{BV}^{(2)}(\mathbb{R}))}\left(\sum_{m=1}^M E\big(\boldsymbol{y}_m,\mathbf{f}(\boldsymbol{x}_m)\big)+\mu\sum_{\ell=1}^N R_\ell(\mathbf{U}_\ell)+\lambda\sum_{\ell=1,}^L\sum_{n=1}^{N_\ell}\mathrm{TV}^{(2)}(\sigma_{n,\ell})\right), \tag{15}$$

where $E:\mathbb{R}^{N_L}\times\mathbb{R}^{N_L}\to\mathbb{R}^+$ is an arbitrary convex error function such that $E(\boldsymbol{y},\boldsymbol{y})=0$ for any $\boldsymbol{y}\in\mathbb{R}^{N_\ell}$, $R_\ell:\mathbb{R}^{N_\ell\times N_\ell}\to\mathbb{R}^+$ is some arbitrary convex cost that favors certain types of linear transformations, and $\lambda,\mu\in\mathbb{R}^+$ are two adjustable regularization parameters. If the solution of (15) exists, then it is achieved by a deep spline network with individual activations of the form

$$\sigma_{n,\ell}(x)=b_{1,n,\ell}+b_{2,n,\ell}x+\sum_{k=1}^{K_{n,\ell}}a_{k,n,\ell}(x-\tau_{k,n,\ell})_+, \tag{16}$$

1. In the least-squares setting, one can adjust the strength of $\mathrm{TV}^{(2)}$-regularization to control the number of knots and thereby produce solutions with $K\ll M$.

with adaptive parameters $K_{n,\ell}\le(M-2)$, $\tau_{1,n,\ell},\ldots,\tau_{K_{n,\ell},n,\ell}\in\mathbb{R}$, and $b_{1,n,\ell},b_{2,n,\ell},a_{1,n,\ell},\ldots,a_{K_{n,\ell},n,\ell}\in\mathbb{R}$.

Proof Let the function $\tilde{\mathbf{f}}:\mathbb{R}^{N_0}\to\mathbb{R}^{N_L}$ be a (not necessarily unique) solution of the problem summarized by (15). This solution is described by (14) with some optimal choice of transformation matrices $\tilde{\mathbf{U}}_\ell$ and pointwise nonlinearities $\tilde\sigma_{n,\ell}:\mathbb{R}\to\mathbb{R}$ for $\ell=1,\ldots,L$ and $n=1,\ldots,N_\ell$.

As we apply $\tilde{\mathbf{f}}$ to the data point $\boldsymbol{x}=\boldsymbol{x}_m$ and progressively move through the layers of the network, we generate a series of vectors $\boldsymbol{z}_{m,\ell}\in\mathbb{R}^{N_\ell}$, according to a recursive definition as follows.

- Initialization (input of the network): $\tilde{\boldsymbol{y}}_{m,0}=\boldsymbol{x}_m$.
- Recursive update: For $\ell=1,\ldots,L$, calculate

$$\boldsymbol{z}_{m,\ell}=(z_{1,m,\ell},\ldots,z_{N_\ell,m,\ell})=\tilde{\mathbf{U}}_\ell\tilde{\boldsymbol{y}}_{m,\ell-1} \tag{17}$$

  and construct $\tilde{\boldsymbol{y}}_{m,\ell}=(\tilde y_{1,m,\ell},\ldots,\tilde y_{N_\ell,m,\ell})\in\mathbb{R}^{N_\ell}$ with

$$\tilde y_{n,m,\ell}=\tilde\sigma_{n,\ell}(z_{n,m,\ell})\quad n=1,\ldots,N_\ell. \tag{18}$$

At the output level, we get $\tilde{\mathbf{f}}(\boldsymbol{x}_m)=\tilde{\boldsymbol{y}}_{m,L}$ for $m=1,\ldots,M$, which are the values that determine the data-fidelity part of the criterion associated with the optimal network and represented by the term $\sum_{m=1}^M E\big(\boldsymbol{y}_m,\mathbf{f}(\boldsymbol{x}_m)\big)$ in (15). Likewise, the specification of the optimal linear transforms $\tilde{\mathbf{U}}_1,\ldots,\tilde{\mathbf{U}}_L$ fixes the regularization cost $\sum_{\ell=1}^L R_\ell(\mathbf{U}_\ell)$. Having set these quantities, we concentrate on the final element of the problem: the characterization of the "optimal" activations $\tilde\sigma_{n,\ell}:\mathbb{R}\to\mathbb{R}$ in-between the locations $z_{n,m,\ell}$ associated with the "auxiliary" data points $\tilde y_{n,m,\ell}=\tilde\sigma_{n,\ell}(z_{n,m,\ell})$, $m=1,\ldots,M$. The key is to recognize that we can now consider the various activation functions individually because the variation of $\tilde\sigma_{n,\ell}$ in-between data points is entirely controlled by $\mathrm{TV}^{(2)}(\tilde\sigma_{n,\ell})$ without any influence on the other terms of the cost functional. Since the solution $\tilde{\mathbf{f}}$ achieves the global optimum, we have that

$$\tilde\sigma_{n,\ell}\in\arg\min_{f\in\mathrm{BV}^{(2)}(\mathbb{R})}\|\mathrm{D}^2f\|_{\mathcal{M}}\quad\text{s.t.}\quad f(z_{n,m,\ell})=\tilde y_{n,m,\ell},\quad m=1,\ldots,M,$$

where the "auxiliary" data pairs $(z_{n,m,\ell},\tilde y_{n,m,\ell})$ are specified by (18). After this reformulation, we can apply Lemma 2, which proves that, at each node (n,ℓ), the minimum is achieved by a nonuniform spline with a number $K_{n,\ell}$ of knots smaller than the number of data points.

Since the hypothesis of feasibility is implicit in the construction, there is only one case not covered by Lemma 2: the singular scenario where all the auxiliary data points associated to a node are equal. Fortunately, this does not break the argument because such a configuration calls for a (zero-cost) solution of the form b_1+b_2x (which is a special case of (12) with $K=0$), except for the twist that there are now infinitely many possibilities with $b_1+b_2z_1=\tilde y_1$. ■

This result translates into a computational structure where each node of the network (with fixed index (n,ℓ)) is characterized by

- its number $0 \leq K = K_{n,\ell}$ of knots (ideally, much smaller than M);
- the location $\{\tau_{k,n,\ell}\}_{k=1}^{K_{n,\ell}}$ of these knots (equivalent to ReLU biases);
- the expansion coefficients $b_{1,n,\ell}, b_{2,n,\ell}, a_{1,n,\ell}, \ldots, a_{K,n,\ell}$, also written as $\mathbf{b} = (b_1, b_2) \in \mathbb{R}^2$ and $\mathbf{a} = (a_1, \ldots, a_K) \in \mathbb{R}^K$ to avoid notational overload.

The fundamental point is that these parameters (including the number of knots) are data-dependent and adjusted automatically through the minimization of (15). All this takes place during training.

3. Interpretation and Discussion

Theorem 4 tells us that we can configure a neural network optimally by restricting our attention to piecewise-linear activation functions $\sigma_{n,\ell}$, or *spline activations*, for short. In effect, this means that the "infinite-dimensional" minimization problem specified by (15) can be converted into a tractable finite-dimensional problem where, for each node (n,ℓ), the parameters to be optimized are the number $K_{n,\ell}$ of knots, the locations $\{\tau_{k,n,\ell}\}_{k=1}^{K_{n,\ell}}$ of the spline knots, and the linear weights $b_{1,n,\ell}, b_{2,n,\ell}, a_{1,n,\ell}, \ldots, a_{K_{n,\ell},n,\ell} \in \mathbb{R}$. The enabling property is going to be (21), which converts the continuous-domain regularization into a discrete ℓ_1-norm. This is consistent with the expectation that bounding the second-order total-variation favors solutions with sparse second derivatives—i.e., linear splines with the fewest possible number of knots. The idea is that ℓ_1-minimization promotes a reduction of the number of active coefficients $a_{k,n,\ell}$ (Donoho, 2006; Unser et al., 2016).

The other important feature is that the knots are adaptive and that they can be learned during training using the standard backpropagation algorithm. What is required is the derivative of the activation functions. It is given by

$$\sigma'_{n,\ell}(x) = b_{2,n,\ell} + \sum_{k=1}^{K_{n,\ell}} a_{k,n,\ell} \mathbb{1}_{[\tau_{k,n,\ell},+\infty)}(x), \tag{19}$$

where $\mathbb{1}_{[\tau,+\infty)}(x)$ is an indicator function that is zero for $x < \tau$ and 1 otherwise. These derivatives are piecewise-constant splines with jumps of height $a_{k,n,\ell}$ at the knot locations $\tau_{k,n,\ell}$. By differentiating (19) once more, we get that

$$\sigma''_{n,\ell}(x) = \sum_{k=1}^{K_{n,\ell}} a_{k,n,\ell} \delta(x - \tau_{k,n,\ell}), \tag{20}$$

where δ is the Dirac distribution. Owing to the property that $\|\delta(\cdot - \tau_{k,n,\ell})\|_{\mathcal{M}} = 1$, we then readily deduce that

$$\mathrm{TV}^{(2)}\{\sigma_{n,\ell}\} = \|\sigma''_{n,\ell}\|_{\mathcal{M}} = \sum_{k=1}^{K_{n,\ell}} |a_{k,n,\ell}| = \|\mathbf{a}_{n,\ell}\|_1, \tag{21}$$

which converts the continuous-domain regularization into a more familiar minimum ℓ_1-norm constraint on the underlying expansion coefficients.

3.1. Link with Existing Techniques

What is even more interesting, from a practical point of view, is that the corresponding system translates into a deep ReLU network modulo a slight modification of the standard architecture described by (6). Indeed, the primary basis functions in (16) are shifted ReLUs, so that each spline activation $\sigma_{n,\ell}$ can be realized by way of a simple one-layer ReLU subnetwork with the spline knots being encoded in the biases. In particular, when the only active coefficients is $a_{n,\ell} = a_{1,n,\ell}$ (i.e., $b_{1,n,\ell} = 0$, $b_{2,n,\ell} = 0$, and $K_{n,\ell} = 1$), we have a perfect equivalence with the classical deep ReLU structure described by (6) with $\sigma_{n,\ell}(x) = (x)_+$. The enabling property is that

$$(\mathbf{w}_{n,\ell}^T \boldsymbol{x} - z_{n,\ell})_+ = (a_{n,\ell} \mathbf{u}_{n,\ell}^T \boldsymbol{x} - z_{n,\ell})_+ = a_{n,\ell}(\mathbf{u}_{n,\ell}^T \boldsymbol{x} - \tau_{n,\ell})_+,$$

with $\mathbf{u}_{n,\ell} = \mathbf{w}_{n,\ell}/\|\mathbf{w}_{n,\ell}\|$, $a_{n,\ell} = \|\mathbf{w}_{n,\ell}\|$, and $\tau_{n,\ell} = z_{n,\ell}/a_{n,\ell}$. Concretely, this means that, for every layer ℓ, we can absorb the single ReLU coefficients $a_{n,\ell}, n = 1, \ldots, N_\ell$ into the prior linear transformation and consider unnormalized transformations $\mathbf{W}_\ell = [\mathbf{w}_{1,\ell} \ \ldots \ \mathbf{w}_{N_\ell,\ell}]^T$ (as in (4)) rather than the normalized ones of Theorem 4 with $\mathbf{u}_{n,\ell} = \mathbf{w}_{n,\ell}/\|\mathbf{w}_{n,\ell}\|$.

Theorem 4 then suggests that the next step in complexity is to add the linear term $b_{1,n,\ell} + b_{2,n,\ell} x$ to each node, since its regularization cost vanishes. Interestingly, the suggested configuration—one ReLU plus an adjustable linear term per neuron—is equivalent to the parametric ReLU model (PReLU) of He et al. (2015) which has been found to systematically outperform the baseline ReLU configuration in real-world applications. The other design extreme is to let $\lambda \to \infty$, in which case the whole network collapses, leading to an affine mapping of the form $\mathbf{f}(\boldsymbol{x}) = \mathbf{W}\boldsymbol{x} - \mathbf{b}$ with $\mathbf{W} \in \mathbb{R}^{N_L \times N_0}$ and $\mathbf{b} \in \mathbb{R}^{N_L}$. More generally, the framework provides us with the possibility of controlling the number of knots (and, hence, the complexity of the network) through the simple adjustment of the regularization parameter λ, with the number of knots increasing as $\lambda \to 0$.

Among the various attempts in the literature to optimize the shape of the activation functions in deep neural networks, there is one scheme that is remarkably close to the optimal solution suggested by our theorem: the APL (adaptive piecewise-linear activation) framework of Agostinelli et al. (2015) in which each neuron is represented as a linear combination of shifted ReLUs, with the parameter being determined during training. The only difference is that their number of ReLUs is fixed a priori and that their model does not include the linear term $b_1 + b_2 x$. While the formulation of Agostinelli et al.'s does not involve any explicit regularization, they found in their experiments that is was helpful to add some mild ℓ_2 penalty on the ReLU coefficients (in contrast to the sparsity-promoting ℓ_1-penalty that results from our theorem) to avoid numerical instability. The good news in support of our theorem is that they report substantial improvement (9.4% and 7.5% relative-error decrease, respectively) on state-of-the-art CNN (with fixed RELU activations) on the CIFAR-10 and CIFAR-100 classification benchmarks.

A characteristic property of deep spline networks, to be considered here as a superset of the traditional deep ReLU networks, is that they produce an input-output relation that is continuous and piecewise-linear (CPWL) in the following sense: the corresponding function $\mathbf{f}$ is continuous $\mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$; its domain $\mathbb{R}^{N_0} = \bigcup_{k=1}^{K} P_k$ can be partitioned into a finite set of non-overlapping convex polytopes P_k over which it is affine (Tarela and Martinez, 1999; Wang and Sun, 2005). More precisely, $\mathbf{f}(\boldsymbol{x}) = \boldsymbol{f}_k(\boldsymbol{x})$ for all $\boldsymbol{x} \in P_k$ where $\boldsymbol{f}_k : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$

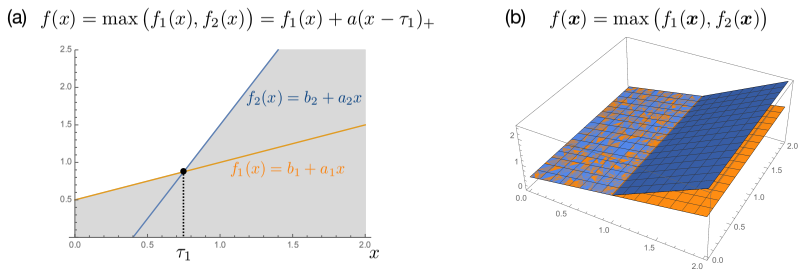


Figure 2: Deep-spline implementation of MaxOut with $N = 2$. (a) In one dimension, the spline parameters are $a = (a_2 - a_1)$ and $\tau_1 = (b_2 - b_1)/(a_2 - a_1)$. (b) In two or more dimensions, $f(\boldsymbol{x}) = b_1 + \boldsymbol{a}_1^T \boldsymbol{x} + a(\boldsymbol{u}^T \boldsymbol{x} - \tau_1)_+$, where $\boldsymbol{u}$ is the unit vector perpendicular to the hinge.

has the same parametric form as in (4). This simply follows from the observation that $\boldsymbol{\sigma}_\ell = (\sigma_{1,\ell}, \ldots, \sigma_{N_\ell,\ell})$, with $\sigma_{n,\ell}$ as specified by (16), is CPWL and that the CPWL property is conserved through functional composition. In fact, the CPWL property for $N = 1$ is equivalent to the function being a nonuniform spline of degree 1.

Another powerful architecture that is known to generate CPWL functions is the MaxOut network (Goodfellow et al., 2013). There, the nonlinear steps $\boldsymbol{\sigma}_\ell$ in (6) are replaced by max-pooling operations. It turns out that these operations are also expressible in terms of deep splines, as illustrated in Figure 2 for the simple case where the maximum is taken over two inputs. Interestingly, this conversion requires the use of the linear term which is absent in conventional ReLU networks. This reinforces the argument made by Goodfellow et al. concerning the capability of MaxOut to learn activation functions.

An attractive feature that is offered by the deep-spline parameterization is the possibility of suppressing a network layer or, rather, of merging two adjacent ones when the optimal solution is such that $K_{n,\ell} = 0$ for ℓ fixed and $n = 1, \ldots, N_\ell$. This is a property that results from the presence of the linear component and has not been exploited so far.

3.2. Generalizations

The optimality result in Theorem 4 holds for a remarkably broad family of cost functions, which should cover all cases of practical interest. The first condition is that the data term, as its name suggest, be solely dependent upon $\boldsymbol{y}_m$ and $\boldsymbol{f}(\boldsymbol{x}_m)$. The second is that the regularization of the weights (the part that constrains the linear steps) and the regularization of the individual activation functions be decoupled from each others. Obvious generalizations of the result include

- cases where the optimization in (15) is performed over a subset of the components while other network elements such as critical linear weights, activation functions[2], or even pooling operators are fixed beforehand (in particular, this includes the important subclass of networks that are not fully connected);
- configurations such as those found in convolutional networks where some (tunable) activation functions are shared among multiple nodes;
- generalized forms of regularization where $\mathrm{TV}^{(2)}(\sigma_{n,\ell})$ is substituted by $\psi\big(\mathrm{TV}^{(2)}(\sigma_{n,\ell})\big)$, where $\psi : \mathbb{R}^+ \to \mathbb{R}^+$ is any monotonically increasing function.

2. A prominent example is the use of the softmax function (Bishop, 2006; Goodfellow et al., 2016) to convert the output of a neural network into a set of pseudo-probabilities.

While the first two scenarios require a slight reformulation of the optimization problem, it is still possible to invoke the same kind of "interpolation" argument as in the proof of Theorem 4. The third generalization is obvious since the (constrained) miminization of $\mathrm{TV}^{(2)}(\sigma_{n,\ell})$ is equivalent to the minimization of $\psi\big(\mathrm{TV}^{(2)}(\sigma_{n,\ell})\big)$.

The statement in Theorem 4 refers to the global optimum of (15), which is often hard to reach in practice because the underlying problem is highly non-convex. It turns out that the argument of the proof is also applicable to local minima and/or saddle points of the cost functional.

By relying on the supporting mathematics in Appendix D for general spline-admissible operators L, it is possible to revisit the proof of Theorem 4 to determine the parametric form of the optimal activations for higher-order versions of TV regularization; i.e., $\mathrm{TV}^{(n)}(\sigma) = \|\mathrm{D}^n \sigma\|_{\mathcal{M}}$. This yields optimal activations that are nonuniform polynomial splines of degree $n > 2$. While such solutions have a higher order of differentiability, they are less favorable globally because the underlying spline property is not retained through composition, meaning that the larger the number of layers, the larger the polynomal degree of the "polytopes" of the resulting network. By contrast, the CPWL property of the linear splines in (16) is preserved through composition, so that the resulting deep spline DNN can be also be interpreted as a flat (or shallow) multidimensional piecewise-linear spline. The other way of inducing CPWL activations is through the quadratic Sobolev 1 regularization of Proposition 3. However, this solution has two shortcomings: (i) its inability to represent the identity, which would result in an infinite cost, and (ii) its lack of sparsity.

3.3. Comparison with Kernel Methods

We like to contrast the result in Theorem 4 with the classical representer theorem of machine learning (Schölkopf et al., 2001). The commonality is that both theorems provide a parametric representation of the solution in the form of a linear "kernel" expansion. The primary distinction is that the classical representer theorem is restricted to "shallow" networks with $L = 1$. Yet, there is another difference even more crucial for our purpose: the fact that the knots τ_k in (16) are *adaptive* and few ($K \ll M$), while the centers $\boldsymbol{x}_m$ in (2) are *fixed* and as numerous as there are data points in the training set. In addition, the ReLU function $(x)_+$ is not a kernel in the traditional sense of the term because it is not positive-definite. We note, however, that it can be substituted by another equivalent spline generator $|x|$, which is conditionally positive-definite (Micchelli, 1986; Wendland, 2005). Again, the property that makes this feasible is the presence of the linear term $b_{1,n,\ell} + b_{2,n,\ell} x$.

There is also a conceptual similarity between the result of Theorem 4 and a recent representer theorem for deep kernel networks (Bohn et al., 2018) that results in a solution that is a composition of L multivalued kernel estimators of the classical RKHS form given

by (2). Again, the two main differences with the present framework are: (i) each layer of the deep kernel network is a multivariate nonlinear map, which does not necessarily allow for affine transformations (e.g., linear regressions) and (ii) the kernel expansion in each layer requires as many basis functions as there are training data; this amounts to a total of $L \times M$ linear parameters. This can rapidly become prohibitive, not to mention the complexity of the underlying (non-convex) optimization task. The first shortcoming can easily be fixed by inserting intermediate affine transformations, in direct analogy with the type of architecture covered by Theorem 4. The second limitation is more fundamental and can probably only be removed by adopting some kind of generalized TV regularization in the spirit of Unser et al. (2017); in short, this calls for an extension of Theorem 4 to multivariate activations, which is currently work in progress.

3.4. Towards a Practical Implementation

While the solution of Theorem 4 is conceptually appealing, it can be expected to be harder to implement than fixed-kernel/RKHS methods since the optimization is not only over the linear weights $\boldsymbol{a}_{n,\ell}$ and $\boldsymbol{b}_{n,\ell}$, but also over the number and position of the corresponding spline knots. There is also always a risk that an increase in the number of degrees of freedom may compromise the generalization ability of the resulting network, which means that the method will need to be carefully tested and validated on real data. A possible strategy for making the optimization easier is to constrain the ReLU units to lie on a grid, in the spirit of Gupta et al. (2018), and then to rely on standard iterative ℓ_1-norm minimization techniques to produce a sparse solution (Donoho, 2006; Foucart and Rauhut, 2013; Unser et al., 2016). Such a scheme may still require some explicit knot-deletion step, either as post-processing or during the training iterations, to effectively trim down the number of parameters. A potential difficulty is that minimum $\mathrm{TV}^{(2)}$ interpolants are typically non-unique, because the underlying regularization is semi-convex. This means that the solution found by an iterative algorithm, assuming that the minimum of the regularization energy is achieved, is not necessarily the sparsest one within the (convex) solution set. Designing an algorithm that can effectively deal with this issue will be a very valuable contribution to the field.

4. Conclusion

The main contribution of this work is to provide the theoretical foundations for an integrated approach to neural networks where a subpart of the design—the optimal shaping of activations—can be transferred to the training part of the process and formulated as a global optimization problem. It also deepens the connection between splines and multilayer ReLU networks, as a pleasing side product. While the concept seems promising and includes the theoretical possibility of suppressing unnecessary layers, it raises a number of issues that can only be answered through extensive experimentation with real data. There are already strong indications in the literature (e.g., the improved performance of PReLU) of the practical usefulness of the linear-activation component that is suggested by the theory and not present in traditional ReLU systems. The task ahead is to demonstrate the capability of more complex spline activations to improve upon the state-of-the-art. (Except

for a potential risk of over-parameterization, deep-spline networks should perform at least as well as deep ReLU, PReLU, or APL networks since the latter constitute a subset of the former.)

We expect the greatest challenge for training a deep spline network to be the proper optimization of the number of knots at each neuron, given that the solution with the fewest parameters is the most desirable. In short, we are still in need of a practical and efficient solution for training a deep neural network with fully adaptable activations that globally produces a continuous and piecewise-linear input-output relation; in other words, a DNN that implements an adaptive multidimensional linear spline.

Appendices

The proof of Lemma 2 is based on some foundational results in (Unser et al., 2017) that rely on compactness arguments requiring the weak* topology. We therefore start with a brief review of the relevant notions from functional analysis (Appendix A). We then specify the topology of $\mathrm{BV}^{(2)}(\mathbb{R})$ in Appendix B and precisely delineate its predual space in Theorem 11. This latter is the key to the proof of Theorem 1 and Lemma 2 that are presented in Appendix C and D, respectively.

Appendix A. Background on Continuity and Weak* Continuity

Definition 5 *Let $v : u \mapsto \langle v, u\rangle$ be a linear functional on a Banach space $\mathcal{U}$ equipped with the norm $\|\cdot\|_{\mathcal{U}}$. Then, $v : \mathcal{U} \to \mathbb{R}$ is said to be continuous if $\lim_{n\to\infty}\langle v, u_n\rangle = \langle v, u\rangle$ for any sequence (u_n) in $\mathcal{U}$ such that $\lim_{n\to\infty}\|u_n - u\|_{\mathcal{U}} = 0$.*

We recall that $\mathcal{U}'$ (the continuous dual of $\mathcal{U}$) is the vector space that is formed of the linear functionals that are continuous on $\mathcal{U}$; it is a Banach space equipped with the dual norm

$$\|v\|_{\mathcal{U}'} = \sup_{\varphi\in\mathcal{U}:\|u\|_{\mathcal{X}}\leq 1} \langle v, u\rangle.$$

By following up the above property, one specifies the space of linear functionals that are continuous on $\mathcal{U}'$, which yields the Banach space $\mathcal{U}''$. A standard result in functional analysis is that $\mathcal{U}$ is continuously embedded in its bidual $\mathcal{U}''$, which is indicated as $\mathcal{U} \hookrightarrow \mathcal{U}''$, with the two spaces being isometrically isomorphic (i.e., $\mathcal{U} = \mathcal{U}''$) if and only if $\mathcal{U}$ is reflexive (Rudin, 1991). In other words, the construction of the bidual $\mathcal{U}''$ gets us back to the initial space in the reflexive case only.

A primary case of interest for this paper is $\mathcal{U}' = \mathcal{M}(\mathbb{R})$, which is not reflexive. For such a scenario, the proper way to deduce the *predual* space $\mathcal{U}$ from $\mathcal{V} = \mathcal{U}'$ is through the identification of the linear functionals that are weak*-continuous on $\mathcal{V}$.

Definition 6 (weak* topology) *A sequence $(v_n)_{n=1}^{\infty}$ in $\mathcal{V} = \mathcal{U}'$ is said to converge to v in the weak* topology if $\lim_{n\to\infty}\langle v_n - v, u\rangle = 0$ for all $u \in \mathcal{U}$.*

Definition 7 (weak* continuity) *A linear functional $u : \mathcal{U}' \to \mathbb{R}$ is said to be weak*-continuous if $\lim_{n\to\infty}\langle u, v_n\rangle = \langle u, v\rangle$ for any sequence (v_n) that converges to v in the weak* topology.*

Proposition 8 (see (Reed and Simon, 1980, Theorem IV.20, p. 114)) *The only weak* continuous linear functionals on $\mathcal{U}'$ are the elements of $\mathcal{U}$.*

The main point is that, despite the qualifier "weak", the functional property of weak* continuity is actually more restrictive than continuity.

In practice, it is relatively straightforward to establish the continuity of $u : \mathcal{V} \to \mathbb{R}$ since the property is equivalent to the existence of a constant $C > 0$ such

$$|\langle u, v\rangle| \leq C\|v\|_{\mathcal{V}}$$

for all $v \in \mathcal{V}$, which also yields $\|u\|_{\mathcal{V}'} \leq C < \infty$. By contrast, proving that $v : \mathcal{V} \to \mathbb{R}$ is weak*-continuous in the non-reflexive scenario requires the precise characterization of the predual of $\mathcal{V}$, which is typically more demanding mathematically. For instance, the property that $\mathcal{M}(\mathbb{R}) = \big(C_0(\mathbb{R})\big)'$ is a fundamental result in measure theory known as the Riesz-Markov theorem (Rudin, 1987). For example, the functionals $\varphi \mapsto \langle 1, \varphi\rangle$ and $\varphi \mapsto \langle \mathbb{1}_{[0,1]}, \varphi\rangle$ are continuous on $\mathcal{M}(\mathbb{R}) = \big(C_0(\mathbb{R})\big)'$ because the "generalized" functions 1 and $\mathbb{1}_{[0,1]}$ are bounded in the sup-norm. However, they both fail to be weak*-continuous. Indeed, $1 \notin C_0(\mathbb{R})$ because the unit-valued function does not decay at infinity, and $\mathbb{1}_{[0,1]} \notin C_0(\mathbb{R})$ because it is not continuous everywhere. In the latter example, we may recover weak* continuity by considering a smoothed version of the indicator function.

These considerations are central to the proof of Lemma 2 because it requires the weak* continuity of the sampling functional $\delta(\cdot - x_m)$. While the sampling operation is continuous on $\mathrm{BV}^{(2)}(\mathbb{R})$, it is not necessarily weak*-continuous; at least, not in the canonical topology that is proposed in (Unser et al., 2017, p. 780) (e.g., polynomial-spline example with $N_0 = 2$ and $\boldsymbol{\phi} = (\delta, -\delta')$). This is the reason why we need to revisit the construction of our native space, as detailed in Appendix B, and establish a new operational criterion for testing weak* continuity (Theorem 11).

Appendix B. Banach Structure of $\mathrm{BV}^{(2)}(\mathbb{R})$ and of Its Predual Space

While the definition of $\mathrm{BV}^{(2)}(\mathbb{R})$ given in (9) is convenient for expository purposes, it is not directly usable for mathematical analysis because the functional $\|\mathrm{D}^2 f\|_{\mathcal{M}}$ is only a semi-norm. We therefore rely on (Unser et al., 2017, Theorem 5) with $\mathrm{L} = \mathrm{D}^2$ to characterize $\mathrm{BV}^{(2)}(\mathbb{R})$ as a Banach space, with the choice of the biorthogonal system $(\boldsymbol{\phi}, \boldsymbol{p})$ for $\mathcal{N}_{\mathrm{D}^2}$ still being left open.

Proposition 9 (Banach structure of $\mathrm{BV}^{(2)}(\mathbb{R})$) *Let $(\boldsymbol{\phi}, \boldsymbol{p})$ be a biorthogonal system for $\mathcal{N}_{\mathrm{D}^2} = \mathrm{span}\{1, x\}$. Then, $\mathrm{BV}^{(2)}(\mathbb{R})$ equipped with the norm*

$$\|f\| = \|\mathrm{D}^2 f\|_{\mathcal{M}} + \sqrt{|\langle \phi_1, f\rangle|^2 + |\langle \phi_2, f\rangle|^2} \tag{22}$$

is a (non-reflexive) Banach space. Moreover, every $f \in \mathrm{BV}^{(2)}(\mathbb{R})$ has the unique direct-sum decomposition

$$f = \mathrm{G}_{\boldsymbol{\phi}}\{w\} + p, \tag{23}$$

where $w = \mathrm{D}^2 f \in \mathcal{M}(\mathbb{R})$, $p = \sum_{n=1}^{2} \langle f, \phi_n\rangle p_n \in \mathcal{N}_{\mathrm{D}^2}$, and $\mathrm{G}_{\boldsymbol{\phi}} : w \mapsto \int_{\mathbb{R}} g_{\boldsymbol{\phi}}(\cdot, y) w(y) \mathrm{d}y$, with

$$g_{\boldsymbol{\phi}}(x, y) = (x - y)_+ - \sum_{n=1}^{2} p_n(x) \langle \phi_n, (\cdot - y)_+\rangle. \tag{24}$$

Central to our formulation is the unique operator $\mathrm{G}_{\boldsymbol{\phi}} : \mathcal{M}(\mathbb{R}) \to \mathrm{BV}^{(2)}(\mathbb{R})$ such that

$$\mathrm{D}^2 \mathrm{G}_{\boldsymbol{\phi}}\{w\} = w \qquad \text{(right-inverse property)} \tag{25}$$

$$\langle \phi_1, \mathrm{G}_{\boldsymbol{\phi}}\{w\}\rangle = 0, \quad \langle \phi_2, \mathrm{G}_{\boldsymbol{\phi}}\{w\}\rangle = 0 \qquad \text{(boundary conditions)} \tag{26}$$

for all $w \in \mathcal{M}(\mathbb{R})$. Specifically, (26) ensures the orthogonality of the two components of the direct-sum decomposition of f in (23), while (25) and the biorthogonality of $(\boldsymbol{\phi}, \boldsymbol{p})$ guarantees its unicity.

Since $\mathrm{BV}^{(2)}(\mathbb{R})$ is non-reflexive, the characterization of its predual is required for testing the hypothesis of weak* continuity. To that end, we first recall that the predual of $\mathcal{M}(\mathbb{R}) = \big(C_0(\mathbb{R})\big)'$ is the space $C_0(\mathbb{R})$ of continuous functions that vanish at infinity equipped with the sup-norm (Rudin, 1987). Moreover, since $\mathcal{S}(\mathbb{R})$ (Schwartz' space of smooth and rapidly decaying functions) is dense in $C_0(\mathbb{R})$ (Schwartz, 1966), the latter can also be described as the completion of $\mathcal{S}(\mathbb{R})$ equipped with the sup-norm, in conformity with the definition of $\mathcal{M}(\mathbb{R})$ given by (7).

We now present an explicit construction and characterization of the predual of $\mathrm{BV}^{(2)}(\mathbb{R})$. This description is consistent with an earlier theorem of ours (Unser et al., 2017, Theorem 6) applicable to general spline spaces; however, it contributes two novel elements: (i) the operational criterion for space membership provided by the first property, and (ii) the construction of the predual space $C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ via the completion of $\mathcal{S}(\mathbb{R})$, which requires additional hypotheses on $\boldsymbol{\phi}$.

Definition 10 *Let $\boldsymbol{p} = (p_1, p_2)$ be a basis of $\mathcal{N}_{\mathrm{D}^2} = \mathrm{span}\{1, x\}$ and $\boldsymbol{\phi} = (\phi_1, \phi_2)$ a complementary set of (generalized) functions whose Fourier transforms are denoted by $\widehat{\phi}_1, \widehat{\phi}_2$. Then, the system $(\boldsymbol{p}, \boldsymbol{\phi})$ is said to be admissible for D^2 if*

1. *the basis functions are biorthogonal; i.e., $\langle \phi_m, p_n\rangle = \delta_{m,n}$, $(m, n = 1, 2)$*
2. *$\widehat{\phi}_1, \widehat{\phi}_2 \in L_{1,2}(\mathbb{R}) = \{f : \mathbb{R} \mapsto \mathbb{R} \;\big|\; \int_{\mathbb{R}} (1 + |\omega|)^{-2} |f(\omega)| \mathrm{d}\omega < \infty\}$, with the two functions being continuously differentiable twice at $\omega = 0$.*

Theorem 11 (Predual of native space) *Let $(\boldsymbol{\phi}, \boldsymbol{p})$ be an admissible system in the sense of Definition 10. Then, the function space*

$$C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R}) = \{g = \mathrm{D}^2 v + a_1 \phi_1 + a_2 \phi_2 : v \in C_0(\mathbb{R}), \mathbf{a} = (a_1, a_2) \in \mathbb{R}^2\} \tag{27}$$

has the following properties:

1. *every $g \in C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ has a unique direct-sum representation as in (27) with $v = \mathrm{G}^*_{\boldsymbol{\phi}}\{g\}$, $a_1 = \langle p_1, g\rangle$, and $a_2 = \langle p_2, g\rangle$, where $\mathrm{G}^*_{\boldsymbol{\phi}}$ is the adjoint of $\mathrm{G}_{\boldsymbol{\phi}}$ specified by (24);*

2. *$C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ is a (non-reflexive) Banach space equipped with the norm*

$$\|g\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}} \triangleq \max(\|\mathrm{G}^*_{\boldsymbol{\phi}} g\|_\infty, \|\boldsymbol{p}(g)\|_2) = \max(\|v\|_\infty, \|\boldsymbol{a}\|_2); \tag{28}$$

3. *$C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ is the predual of $\mathrm{BV}^{(2)}(\mathbb{R})$ in Proposition 9; i.e., $\mathrm{BV}^{(2)}(\mathbb{R}) = \big(C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})\big)'$;*

4. *$C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ is the completion of $\mathcal{S}(\mathbb{R})$ equipped with the $\|\cdot\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}}$-norm.*

Proof : The main idea is that the construction expressed by (27) is the direct sum of the two linear spaces $\mathcal{U}$ and $\mathcal{N}_{\boldsymbol{\phi}}$ whose Banach topology and completion properties are revealed next.

(i) Topology of the Space $\mathcal{N}_{\boldsymbol{\phi}} = \mathrm{span}\{\phi_1, \phi_2\}$ and of Its Dual
The space $\mathcal{N}_{\boldsymbol{\phi}}$ collects the two last components of g in (27) and is equipped with the discrete ℓ_2-norm $\|\phi\|_{\mathcal{N}_{\boldsymbol{\phi}}} = \|\mathbf{a}\|_2$ with $\mathbf{a} = \boldsymbol{p}(\phi) = (\langle p_1, \phi\rangle, \langle p_2, \phi\rangle)$. We also specify the projection operator $C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R}) \to \mathcal{N}_{\boldsymbol{\phi}}$:

$$\mathrm{Proj}_{\mathcal{N}_{\boldsymbol{\phi}}}\{g\} = \langle p_1, g\rangle \phi_1 + \langle p_2, g\rangle \phi_2.$$

The complementary space is $\mathcal{N}_{\boldsymbol{p}} = \mathrm{span}\{p_1, p_2\}$ equipped with the norm $\|p\|_{\mathcal{N}_{\boldsymbol{p}}} = \|\boldsymbol{p}(p)\|_2 = \|\mathbf{b}\|_2$ with $\mathbf{b} = \boldsymbol{\phi}(p) = (\langle \phi_1, p\rangle, \langle \phi_2, p\rangle)$. Thanks to the biorthogonality of $\boldsymbol{\phi}$ and $\boldsymbol{p}$, for all $p = b_1 p_1 + b_2 p_2 \in \mathcal{N}_{\boldsymbol{p}}$, we have that

$$\|p\|_{\mathcal{N}'_{\boldsymbol{\phi}}} = \sup_{\phi \in \mathcal{N}_{\boldsymbol{\phi}} : \|\phi\|_{\mathcal{N}_{\boldsymbol{\phi}}} \le 1} \langle \phi, p\rangle = \sup_{\mathbf{a} \in \mathbb{R}^2 : \|\mathbf{a}\|_2 \le 1} \mathbf{a}^T \mathbf{b} = \|\mathbf{b}\|_2 = \|p\|_{\mathcal{N}_{\boldsymbol{p}}},$$

which shows that $\mathcal{N}_{\boldsymbol{p}} = \mathcal{N}'_{\boldsymbol{\phi}}$ is the continuous dual of $\mathcal{N}'_{\boldsymbol{\phi}}$.

(ii) Range of the Operator $\mathrm{G}^*_{\boldsymbol{\phi}}$
To derive the required properties, we restrict the domain of $\mathrm{G}^*_{\boldsymbol{\phi}}$ to the subspace

$$\mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R}) = \big\{\psi \in \mathcal{S}(\mathbb{R}) : \langle p_1, \psi\rangle = 0, \langle p_2, \psi\rangle = 0\big\} \subset \mathcal{S}(\mathbb{R}).$$

By using the explicit form (24) of the kernel of $\mathrm{G}_{\boldsymbol{\phi}}$, we find that, for any $\psi \in \mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R})$,

$$\begin{aligned}
\mathrm{G}^*_{\boldsymbol{\phi}}\{\psi\}(x) &= \int_{\mathbb{R}} \big((y-x)_+ - q_1(x)p_1(y) - q_2(x)p_2(y)\big)\psi(y)\mathrm{d}y \\
&= \int_{\mathbb{R}} (y-x)_+ \psi(y)\mathrm{d}y - q_1(x)\underbrace{\langle p_1, \psi\rangle}_{0} - q_2(x)\underbrace{\langle p_2, \psi\rangle}_{0} \\
&= \mathrm{D}^{-2*}\{\psi\}(x),
\end{aligned} \tag{29}$$

where $q_n(y) = \langle \phi_n, (\cdot - y)_+\rangle$ for $n = 1, 2$, and D^{-2*} is the 2-fold (adjoint) integration operator whose frequency response is $\big(-1/\omega^2 - \mathrm{i}\pi\delta'(\omega)\big) = \mathcal{F}\{(-x)_+\}(\omega)$. Based on (29), we then show that

$$\forall \psi \in \mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R}) : \quad \mathrm{G}^*_{\boldsymbol{\phi}}\{\psi\} \in C_0(\mathbb{R}), \tag{30}$$

which, as we shall see, implies the boundedness of $\mathrm{G}^*_{\boldsymbol{\phi}} : \mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R}) \to C_0(\mathbb{R}) \hookrightarrow \mathcal{S}'(\mathbb{R})$. Property (30) is established by examining the Fourier transform[3] of $f = \mathrm{D}^{-2*}\{\psi\}$ given by

$$\hat{f}(\omega) = -\widehat{\psi}(\omega)/\omega^2 - \mathrm{i}\pi\big(\widehat{\psi}(0)\delta'(\omega) - \widehat{\psi}^{(1)}(0)\delta(\omega)\big) \tag{31}$$

with $\widehat{\psi} = \mathcal{F}\{\psi\} \in \mathcal{S}(\mathbb{R})$. Since $\boldsymbol{p}(\psi) = \mathbf{0} \Leftrightarrow \widehat{\psi}(0) = \widehat{\psi}^{(1)}(0) = 0$, we first simplify (31) to $\hat{f}(\omega) = \big(-\widehat{\psi}(\omega)/\omega^2\big)$ and then invoke a Taylor-series argument to deduce the continuity of $\hat{f}(\omega)$ at $\omega = 0$. This, together with the boundedness and rapid decay of $\widehat{\psi}(\omega)$, implies that $\hat{f} \in L_1(\mathbb{R})$. The announced result—the continuity, boundedness, and decay of $f(x)$ at infinity—then follows from the Riemann-Lebesgue lemma.

(iii) The Banach Topology of $\mathcal{U}$
The definition of $\mathcal{U}$, which corresponds to the first component in (27), is

$$\mathcal{U} = \{f = \mathrm{D}^2 v : v \in C_0(\mathbb{R})\},$$

equipped with the norm $\|\mathrm{D}^2 v\|_{\mathcal{U}} = \|v\|_\infty$, which establishes an isometric isomorphism with $C_0(\mathbb{R})$. Our intent now is to prove that $\|f\|_{\mathcal{U}} = \|\mathrm{G}^*_{\boldsymbol{\phi}} f\|_\infty$ for all $f \in \mathcal{U}$, which is equivalent to showing that $\mathrm{G}^*_{\boldsymbol{\phi}}$ is the inverse of $\mathrm{D}^2 : C_0(\mathbb{R}) \to \mathcal{U}$.

We shall achieve this through an extension process that builds upon the properties of the operator $\mathrm{G}^*_{\boldsymbol{\phi}}$ established in Step (*ii*). We start by considering the semi-norm $\psi \mapsto \|\psi\|_{\tilde{\mathcal{U}}} \triangleq \|\mathrm{G}^*_{\boldsymbol{\phi}}\psi\|_\infty$, which is well-defined over $\mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R}) \subset \mathcal{S}(\mathbb{R})$. Since $\mathrm{G}^*_{\boldsymbol{\phi}}\psi = \mathrm{D}^{-2*}\psi$ for all $\psi \in \mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R})$ and $\mathrm{D}^2\mathrm{D}^{-2*}\varphi = \varphi$ any $\varphi \in \mathcal{S}(\mathbb{R})$, we have that $\|\mathrm{G}^*_{\boldsymbol{\phi}}\psi\|_\infty = 0 \Leftrightarrow \psi = 0$, which shows that $\|\cdot\|_{\tilde{\mathcal{U}}}$ is a norm over $\mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R})$, as expected. This allows us to rephrase the inclusion property from Step (*ii*) as: $\mathrm{G}^*_{\boldsymbol{\phi}}$ isometrically maps $(\mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R}), \|\cdot\|_{\tilde{\mathcal{U}}})$ to the Banach space $(C_0(\mathbb{R}), \|\cdot\|_\infty)$, which is the form suitable for the bounded linear transformation (BLT) extension theorem.

Theorem 12 (Reed and Simon (1980, Theorem I.7, p. 9)) *Let G be a bounded linear transformation from a normed space $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ to a complete normed space $(\mathcal{Y}, \|\cdot\|_{\mathcal{Y}})$. Then, G has a unique extension to a bounded linear transformation (with the same bound) from the completion of $\mathcal{X}$ to $(\mathcal{Y}, \|\cdot\|_{\mathcal{Y}})$.*

Consequently, the restricted operator from Step (*ii*) uniquely extends to an isometry $\mathrm{G}^*_{\boldsymbol{\phi}} : \tilde{\mathcal{U}} \to C_0(\mathbb{R})$ where the Banach space $\tilde{\mathcal{U}}$ is the completion of $\mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R})$ in the $\|\cdot\|_{\tilde{\mathcal{U}}}$-norm. The final element is that $\mathrm{D}^2\mathrm{G}^*_{\boldsymbol{\phi}}\psi = \mathrm{D}^2\mathrm{D}^{-2*}\psi = \psi$ for all $\mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R}) \subseteq \tilde{\mathcal{U}}$, which indicates that D^2 is the inverse of $\mathrm{G}^*_{\boldsymbol{\phi}}$ on $\mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R})$. Since the latter is a dense subset of $\tilde{\mathcal{U}}$, we can extend the property to the entire space, which ultimately proves that $\mathcal{U} = \tilde{\mathcal{U}}$.

(iv) The space $C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R}) = \mathcal{U} \oplus \mathcal{N}_{\boldsymbol{\phi}}$
The inclusion $g \in C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ is equivalent to $g = f + \phi$ where $f = \mathrm{D}^2 v$ with $v \in C_0(\mathbb{R})$ and $\phi = a_1\phi_1 + a_2\phi_2$. The components (f, ϕ) are retrieved as $f = \mathrm{Proj}_{\mathcal{U}}\{g\} = \mathrm{D}^2\mathrm{G}^*_{\boldsymbol{\phi}} g$ and $\phi = \mathrm{Proj}_{\mathcal{N}_{\boldsymbol{\phi}}}\{g\}$. The conditions $\mathrm{G}^*_{\boldsymbol{\phi}}\phi = 0$ and $\mathrm{Proj}_{\mathcal{N}_{\boldsymbol{\phi}}}\{\mathrm{D}^2 v\} = 0$ for all $\phi \in \mathcal{N}_{\boldsymbol{\phi}}$ and $v \in C_0(\mathbb{R})$ ensure that $\mathcal{U} \cap \mathcal{N}_{\boldsymbol{\phi}} = \{0\}$ so that the sum is direct. The other relevant identity

3. We use the product rule $\psi(\cdot)\delta' = \psi(0)\delta' - \psi'(0)\delta$, which follows from the definition of the distribution $\delta' : \varphi \mapsto \langle \delta', \varphi\rangle = -\varphi'(0)$.

from Step (*iii*) is $f = \mathrm{D}^2\mathrm{G}_{\boldsymbol{\phi}}^* f$ for all $f \in \mathcal{U}$. Consequently, $C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R}) = \mathcal{U} \oplus \mathcal{N}_{\boldsymbol{\phi}}$ is a Banach space when equipped with $\|(f,\phi)\|_\infty$, which is the composite norm given by (28).

(v) The Identification of $\mathrm{BV}^{(2)}(\mathbb{R}) = \big(C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})\big)'$

First, we identify the norm of $\mathcal{U}'$ by applying a standard duality argument:

$$\begin{aligned}\|u^*\|_{\mathcal{U}'} &= \sup_{u\in\mathcal{U}:\|u\|_{\mathcal{U}}\le 1} \langle u^*, u\rangle = \sup_{v\in C_0(\mathbb{R}):\|v\|_\infty\le 1} \langle u^*, \mathrm{D}^2 v\rangle \\ &= \sup_{v\in\mathcal{S}(\mathbb{R}):\|v\|_\infty\le 1} \langle \mathrm{D}^2 u^*, v\rangle = \|\mathrm{D}^2 u^*\|_{\mathcal{M}}\end{aligned}$$

where we have used the identity $u = \mathrm{D}^2 v$ with $v \in C_0(\mathbb{R})$ and the denseness of $\mathcal{S}(\mathbb{R})$ in $C_0(\mathbb{R})$. The dual of $C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ in Step (*iv*) is then given by $\mathcal{U}' \oplus \mathcal{N}'_{\boldsymbol{\phi}} = \mathcal{U}' \oplus \mathcal{N}_{\boldsymbol{p}}$ equipped with the composite norm $\|(u^*,p)\|_1 = \|u^*\|_{\mathcal{U}'} + \|\boldsymbol{\phi}(p)\|_2 = \|\mathrm{D}^2 f\|_{\mathcal{M}} + \|\boldsymbol{\phi}(f)\|_2 = \|f\|_{\mathrm{BV}^{(2)}}$, which is the dual norm of $\|(f,\phi)\|_\infty = \max(\|f\|_{\mathcal{U}}, \|\boldsymbol{p}(\phi)\|_2)$.

(vi) The Space $C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ *Is The Completion of* $\mathcal{S}(\mathbb{R})$ *in the* $\|\cdot\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}}$*-Norm*

The idea is to amend the extension technique of Step (*iii*) by selecting a second biorthogonal system $(\boldsymbol{\varphi}, \boldsymbol{p})$ such that $\mathcal{N}_{\boldsymbol{\varphi}} = \mathrm{span}\{\varphi_1, \varphi_2\} \subset \mathcal{S}(\mathbb{R})$. This yields the direct-sum decomposition of $\varphi = \tilde{\psi} + \tilde{\phi} \in \mathcal{S}(\mathbb{R})$ with $\tilde{\phi} = \mathrm{Proj}_{\mathcal{N}_{\boldsymbol{\varphi}}}\{\varphi\} \in \mathcal{N}_{\boldsymbol{\varphi}}$ and $\tilde{\psi} = (\varphi - \tilde{\phi}) \in \mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R})$. While we already know that $\mathrm{G}_{\boldsymbol{\phi}}^*\tilde{\psi} \in C_0(\mathbb{R})$, the delicate point is to make sure that the same holds true for $\mathrm{G}_{\boldsymbol{\phi}}^*\tilde{\phi}$. Since $\tilde{\phi} \in \mathrm{span}\{\varphi_1, \varphi_2\}$, the last requirement is equivalent to

$$\mathrm{G}_{\boldsymbol{\phi}}^*\{\varphi_n\} = \mathrm{D}^{-2*}(\mathrm{Id} - \mathrm{Proj}_{\mathcal{N}_{\boldsymbol{\phi}}})\{\varphi_n\} = \mathrm{D}^{-2*}\{\varphi_n - \phi_n\} \in C_0(\mathbb{R}) \tag{32}$$

for $n = 1, 2$. With the same arguments as in Step (*ii*) (Riemann-Lebesgue lemma), we ensure that (32) is met by imposing the Fourier-domain condition

$$\frac{\widehat{\phi}_n(\omega) - \widehat{\varphi}_n(\omega)}{\omega^2} \in L_1(\mathbb{R}), \tag{33}$$

which results from the second hypothesis in Definition 10. In effect, the role of $\widehat{\varphi}_n \in \mathcal{S}(\mathbb{R})$ in (33) is to temper the singularity of $1/\omega^2$ at the origin, in reason of the condition $\boldsymbol{p}(\varphi_n - \phi_n) = \mathbf{0}$, which induces a second-order zero in the numerator. This correction does not impact integrability otherwise because of the rapid decay of $\widehat{\varphi}_n$.

Having established that $\mathrm{G}_{\boldsymbol{\phi}}^*\{\tilde{\psi} + \tilde{\phi}\} \in C_0(\mathbb{R})$, we can now check that

$$\|\varphi\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}} = \max(\|\mathrm{G}_{\boldsymbol{\phi}}^*\{\tilde{\psi} + \tilde{\phi}\}\|_\infty, \|\boldsymbol{p}(\tilde{\phi})\|_2) = 0 \Leftrightarrow (\tilde{\psi}, \tilde{\phi}) = (0,0) \Leftrightarrow \varphi = 0,$$

which proves that $\|\cdot\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}}$ is a valid norm over $\mathcal{S}(\mathbb{R}) = \mathcal{S}_{\boldsymbol{p}^\perp}(\mathbb{R}) \oplus \mathcal{N}_{\boldsymbol{\varphi}}$. We then deduce the desired completion result from the BLT theorem by observing that $\mathrm{G}_{\boldsymbol{\phi}}^* : (\tilde{\psi}, \tilde{\phi}) \mapsto \mathrm{G}_{\boldsymbol{\phi}}^*\{\tilde{\psi} + \tilde{\phi}\}$ is bounded from $(\mathcal{S}(\mathbb{R}), \|\cdot\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}})$ to $(C_0(\mathbb{R}), \|\cdot\|_\infty)$. The boundedness of the operator simply follows from the inequality

$$\|\mathrm{G}_{\boldsymbol{\phi}}^*\varphi\|_\infty \le \|\varphi\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}} = \max(\|\mathrm{G}_{\boldsymbol{\phi}}^*\varphi\|_\infty, \|\boldsymbol{p}(\varphi)\|_2) < \infty$$

for any $\varphi \in \mathcal{S}(\mathbb{R})$. ■

By considering the dual form of Property 4 in Theorem 11 (which is a new result, to the best of our knowledge), we obtain an alternative, self-contained definition of our native space as

$$\mathrm{BV}^{(2)}(\mathbb{R}) = \{f \in \mathcal{S}'(\mathbb{R}) : \sup_{\varphi\in\mathcal{S}(\mathbb{R}):\|\varphi\|_{C_{\mathrm{D}^2,\boldsymbol{\phi}}}\le 1} \langle f, \varphi\rangle < \infty\}, \tag{34}$$

which is the direct analog of (7).

Another important observation is that the "canonical" choice $\boldsymbol{\phi} = (\delta, -\delta')$ from (Unser et al., 2017) does not fulfill the second condition in Definition 10 (it actually fails by a tiny margin because $(-\mathrm{i}\omega)$ is only in $L_{1,2+\epsilon}(\mathbb{R})$ for any $\epsilon > 0$). This means that Property 4 does not apply to that particular case, even though the underlying native spaces are hardly distinguishable as sets. The only significant difference is in the specification of the corresponding weak* topology which, as we shall see, is essential to the proof of Lemma 2.

Appendix C. Proof of Theorem 1

(i) Explicit Banach Topology and Identification of the Right Inverse of D^2

The first statement in Theorem 1 results for the application of Proposition 9 with our specific choice of biorthogonal system $\boldsymbol{\phi} = (\phi_1, \phi_2) = \big(\delta, -\delta + \delta(\cdot - 1)\big)$ and $\boldsymbol{p} = (p_1, p_2)$, with $p_1(x) = 1$ and $p_2(x) = x$. This system satisfies the admissibility conditions in Definition 10. In particular, we have that $\langle p_1, \phi_1\rangle = p_1(0) = 1$, $\langle p_1, \phi_2\rangle = -p_1(0) + p_1(1) = 0$, $\langle p_2, \phi_1\rangle = p_2(0) = 0$ and $\langle p_2, \phi_2\rangle = -p_2(0) + p_2(1) = 1$ (biorthogonality). By inserting these functionals in (22), we obtain the formula of the norm for $\mathrm{BV}^{(2)}(\mathbb{R})$ in (11).

The corresponding expression (24) of the kernel of $\mathrm{G}_{\boldsymbol{\phi}}$ is

$$g_{\boldsymbol{\phi}}(x,y) = (x-y)_+ - (1-x)(-y)_+ - x(1-y)_+, \tag{35}$$

whose overall behavior is illustrated in Figure 3. We observe that the functions $y \mapsto g_{\boldsymbol{\phi}}(x,y)$ are continuous, triangle-shaped B-splines with the following characteristics:

- for $x \le 0$: $y \mapsto g_{\boldsymbol{\phi}}(x,y)$ is supported in $[x,1]$ and takes its maximum at $y = 0$;
- for $x \in (0,1)$: $y \mapsto g_{\boldsymbol{\phi}}(x,y)$ is supported in $[0,1]$ and takes its extremum at $y = x$;
- for $x \ge 1$: $y \mapsto g_{\boldsymbol{\phi}}(x,y)$ is supported in $[0,x]$ and takes its maximum at $y = 1$.

(ii) Weak Continuity of Sampling Functionals*

The key here is that $\mathrm{G}_{\boldsymbol{\phi}}^*\{\delta(\cdot - x_m)\}(y) = g_{\boldsymbol{\phi}}(x_m, y)$ where $g_{\boldsymbol{\phi}}(x_m, \cdot)$, as defined by (35), is continuous, bounded and compactly supported, and, hence, vanishing at $\pm\infty$. Consequently, $\delta(\cdot - x_m) = \mathrm{D}^2 v + a_1\phi_1 + a_2\phi_2$ with $v = g_{\boldsymbol{\phi}}(x_m, \cdot) \in C_0(\mathbb{R})$, $a_1 = \langle 1, \delta(\cdot - x_m)\rangle = 1$, and $a_2 = \langle x, \delta(\cdot - x_m)\rangle = x_m$ in accordance with (27) in Theorem 11, which proves that $\delta(\cdot - x_m) \in C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$. This establishes its weak* continuity on $\big(C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})\big)'$ (by Proposition 8).

Based on the observation that $\|g_{\boldsymbol{\phi}}(x_m, \cdot)\|_\infty \le |x_m|$, we then easily estimate the norm of $\delta(\cdot - x_m)$ as

$$\begin{aligned}\|\delta(\cdot - x_m)\|'_{\mathrm{BV}^{(2)}} &= \max\big(\sup_{y\in\mathbb{R}} |g_{\boldsymbol{\phi}}(x_m, y)|, \|\boldsymbol{a}\|_2\big) \\ &\le (1 + |x_m|) < \infty.\end{aligned}$$

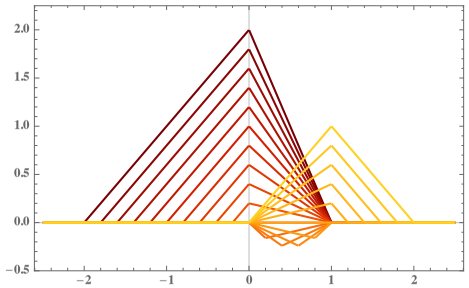


Figure 3: Graphs of the function $y \mapsto g_{\boldsymbol{\phi}}(x, y)$ for a series of values of $x = -2, \ldots, 2$ with step of 0.25. This illustrates the property that $g_{\boldsymbol{\phi}}(x, \cdot) \in C_0(\mathbb{R})$ for any $x \in \mathbb{R}$, which is critical to the proof of Lemma 2. By contrast, the canonical solution of (Unser et al., 2017) with $\boldsymbol{\phi} = (\delta, -\delta')$ would have resulted in a series of piecewice-linear functions with a discontinuous drop to 0 at $x = 0$.

Finally, we recall that the property that two Banach spaces $\mathcal{U}$ and $\mathcal{U}'$ form a dual pair implies that $|\langle u, u' \rangle| \leq \|u\|_{\mathcal{U}} \|u'\|_{\mathcal{U}'}$ for any $u \in \mathcal{U}$ and $u' \in \mathcal{U}'$. Taking $\mathcal{U} = C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ and $u = \delta(\cdot - x_m)$ allows us to translate the above norm estimate into the announced continuity bound.

(iii) Embedding Properties
Property 4 in Theorem 11 tells us that $\mathcal{S}(\mathbb{R}) \hookrightarrow C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R})$ with the embedding being dense. This, together with the observation that $C_{\mathrm{D}^2,\boldsymbol{\phi}}(\mathbb{R}) \hookrightarrow \mathcal{S}'(\mathbb{R}^d)$, implies that $\mathcal{S}(\mathbb{R}) \hookrightarrow \mathrm{BV}^{(2)}(\mathbb{R}) \hookrightarrow \mathcal{S}'(\mathbb{R})$ (by duality) with the outer embedding being dense since $\mathcal{S}(\mathbb{R})$ is itself dense in $\mathcal{S}'(\mathbb{R})$. As for the embedding in $C_{\mathrm{b},1}(\mathbb{R})$, we first invoke Proposition 14 below, which states that $\mathrm{BV}^{(2)}(\mathbb{R}) \hookrightarrow L_{\infty,1}(\mathbb{R})$. Since $C_{\mathrm{b},1}(\mathbb{R})$ is isometrically embedded in $L_{\infty,1}(\mathbb{R})$ and the members of $\mathrm{BV}^{(2)}(\mathbb{R})$ are continuous (because they are differentiable), we then readily deduce that $\mathcal{S}(\mathbb{R}) \hookrightarrow \mathrm{BV}^{(2)}(\mathbb{R}) \hookrightarrow C_{\mathrm{b},1}(\mathbb{R})$.

We note that the proof of the embedding $\mathrm{BV}^{(2)}(\mathbb{R}) \hookrightarrow C_{\mathrm{b},1}(\mathbb{R})$ does not depend upon the specific choice of biorthogonal system $(\boldsymbol{\phi}, \boldsymbol{p})$, as long as the admissibility condition in Definition 10 holds. By contrast, the proof of the weak* continuity of $\delta(\cdot - x_0)$ in Item *(ii)* is specific. While it is sufficient for our purpose, we have recent evidence (work in progress) that the argument is extendable to a broad class of equivalent biorthogonal systems, which makes us believe that there is actually a tight connection between Properties 2 and 3 in Theorem 1. Figure 3 and the argumentation in Step *(ii)* are enlightening in that respect because they make us appreciate the regularization effect of the second component in (24), which counterbalances the growth of the primary term $g(x, y) = (x - y)_+$ (the kernel of the shift-invariant inverse D^{-2*}). The other constraint is that the ϕ_n need to be regular enough (e.g., no worse than a Dirac) to preserve continuity.

Appendix D. Proof of Lemma 2

Proof The lemma is deduced from (Unser et al., 2017, Theorem 4): an abstract optimality result for generalized spline interpolation that holds for an extended class of admissible regularization operators L and for arbitrary linear functionals $(\nu_m : f \mapsto \langle \nu_m, f \rangle)$, subject to the weak*-continuity requirement. The relevant version of the result for functions $f : \mathbb{R} \to \mathbb{R}$ is restated here in the explicit form of Theorem 15.

The maximal polynomial rate of growth (n_0) of functions is controlled via their inclusion in the space

$$L_{\infty,n_0}(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} \ \text{ s.t. } \|f\|_{\infty,n_0} \triangleq \operatorname*{ess\,sup}_{x \in \mathbb{R}} (1 + |x|)^{-n_0} |f(x)| < \infty\}.$$

Definition 13 (Spline-admissible operator) *A linear operator* $\mathrm{L} : \mathcal{M}_{\mathrm{L}}(\mathbb{R}) \to \mathcal{M}(\mathbb{R})$, *where* $\mathcal{M}_{\mathrm{L}}(\mathbb{R}) \supseteq \mathcal{S}(\mathbb{R})$ *is an appropriate subspace of* $\mathcal{S}'(\mathbb{R})$, *is called* spline-admissible *if*

1. *it is shift-invariant;*
2. *there exists a function* $\rho_{\mathrm{L}} : \mathbb{R} \to \mathbb{R}$ *of slow growth (the Green's function of* L*) such that* $\mathrm{L}\rho_{\mathrm{L}} = \delta$, *where* δ *is the Dirac impulse. The rate of polynomial growth of* ρ_{L} *is* $n_0 = \inf\{n \in \mathbb{N} : \rho_{\mathrm{L}} \in L_{\infty,n}(\mathbb{R})\}$.
3. *the (growth-restricted) null space of* L,
$$\mathcal{N}_{\mathrm{L}} = \{q \in L_{\infty,n_0}(\mathbb{R}) : \mathrm{L}q = 0\},$$
has the finite dimension $N_0 \geq 0$.

The native space of L, $\mathcal{M}_{\mathrm{L}}(\mathbb{R})$, is then identified as

$$\mathcal{M}_{\mathrm{L}}(\mathbb{R}) = \{f \in L_{\infty,n_0}(\mathbb{R}) : \|\mathrm{L}f\|_{\mathcal{M}} < \infty\}. \tag{36}$$

In addition, it is assumed that $\mathcal{M}_{\mathrm{L}}(\mathbb{R})$ is equipped with an appropriate Banach topology which gives a concrete meaning to the underlying notion of (weak*-) continuity.

As expected, the operator $\mathrm{L} = \mathrm{D}^2$ is spline-admissible: Its causal Green's function is $\rho_{\mathrm{D}^2}(x) = (x)_+$ (ReLU), which exhibits the algebraic rate of growth $n_0 = 1$, while its null space $\mathcal{N}_{\mathrm{D}^2} = \mathrm{span}\{p_1, p_2\}$ with $p_1(x) = 1$ and $p_2(x) = x$ is finite-dimensional with $N_0 = 2$. These are precisely the basis functions associated with L that appear in (12).

We now show that the slow-growth condition with $n_0 = 1$ is implicit in the specification of $\mathrm{BV}^{(2)}(\mathbb{R})$ given by (9) and/or Proposition 9 so that our definition of the native space is consistent with (36), with $\mathrm{BV}^{(2)}(\mathbb{R}) = \mathcal{M}_{\mathrm{D}^2}(\mathbb{R})$.

Proposition 14 *With the choice of topology specified in Theorem 1,* $\mathrm{BV}^{(2)}(\mathbb{R}) \hookrightarrow L_{\infty,1}(\mathbb{R})$, *while*

$$f \in \mathrm{BV}^{(2)}(\mathbb{R}) \quad \Leftrightarrow \quad \mathrm{TV}^{(2)}(f) = \sup_{\|\varphi\|_\infty \leq 1 : \varphi \in \mathcal{S}(\mathbb{R})} \langle f, \mathrm{D}^2 \varphi \rangle = \|\mathrm{D}^2 f\|_{\mathcal{M}} < \infty.$$

Proof The key is the bound $\|g_{\boldsymbol{\phi}}(x,\cdot)\|_\infty \leq |x|$ for any $x \in \mathbb{R}$ (see Figure 3 and accompanying explanations), which implies that

$$C_{\boldsymbol{\phi}} = \operatorname*{ess\,sup}_{x,y\in\mathbb{R}} (1+|x|)^{-1}|g_{\boldsymbol{\phi}}(x,y)| < \infty.$$

This ensures the continuity of the operator $\mathrm{G}_{\boldsymbol{\phi}} : \mathcal{M}(\mathbb{R}) \to L_{\infty,1}(\mathbb{R})$ with $\|\mathrm{G}_{\boldsymbol{\phi}}\| = C_{\boldsymbol{\phi}}$ by (Unser et al., 2017, Theorem 3). Next, we use the property that any $f \in \mathrm{BV}^{(2)}(\mathbb{R})$ admits the unique decomposition $f = \mathrm{G}_{\boldsymbol{\phi}} w + p$ with $w = \mathrm{L}f \in \mathcal{M}(\mathbb{R})$ and $p = \sum_{n=1}^{2} \langle \phi_n, f\rangle p_n \in \mathcal{N}_{\boldsymbol{p}}$, so that

$$\begin{aligned}
\|f\|_{\infty,1} &\leq \|\mathrm{G}_{\boldsymbol{\phi}} w\|_{\infty,1} + \|p\|_{\infty,1} \\
&\leq C_{\boldsymbol{\phi}} \|w\|_{\mathcal{M}} + \sum_{n=1}^{2} |\langle \phi_n, f\rangle| \, \|p_n\|_{\infty,1} \\
&\leq C_{\boldsymbol{\phi}} \|\mathrm{L}f\|_{\mathcal{M}} + \|\boldsymbol{p}(f)\|_2 \sum_{n=1}^{2} \|p_n\|_{\infty,1} \\
&\leq \left(C_{\boldsymbol{\phi}} + \sum_{n=1}^{2} \|p_n\|_{\infty,1} \right) \|f\|_{\mathrm{BV}^{(2)}},
\end{aligned}$$

which proves that $\mathrm{BV}^{(2)}(\mathbb{R})$ is continuously embedded in $L_{\infty,1}(\mathbb{R})$. The reason for using the dual definition of the $\mathrm{TV}^{(2)}$ semi-norm in the last statement of the proposition is that the formula remains valid for any $f \in \mathcal{S}'(\mathbb{R})$ with $\mathrm{TV}^{(2)}(f) = \infty \Leftrightarrow f \notin \mathrm{BV}^{(2)}(\mathbb{R})$. Likewise, $\mathrm{TV}^{(2)}(f) = 0 \Leftrightarrow f \in \mathcal{N}_{\boldsymbol{p}}$. ■

Theorem 15 (Generalized spline interpolant) *Let us assume that the following conditions are met:*

1. *The operator* $\mathrm{L} : \mathcal{M}_{\mathrm{L}}(\mathbb{R}) \to \mathcal{M}(\mathbb{R})$ *is spline-admissible in the sense of Definition 13.*
2. *The linear measurement operator* $\boldsymbol{\nu} : f \mapsto \boldsymbol{\nu}(f) = \big(\langle \nu_1, f\rangle, \ldots, \langle \nu_M, f\rangle\big)$ *maps* $\mathcal{M}_{\mathrm{L}}(\mathbb{R}^d) \to \mathbb{R}^M$ *and is weak*-continuous on* $\mathcal{M}_{\mathrm{L}}(\mathbb{R}^d) = \big(C_{\mathrm{L}}(\mathbb{R}^d)\big)'$.
3. *The recovery problem is well-posed over the null space of* L*:* $\boldsymbol{\nu}(q_1) = \boldsymbol{\nu}(q_2) \Leftrightarrow q_1 = q_2$, *for any* $q_1, q_2 \in \mathcal{N}_{\mathrm{L}}$.

Then, the extremal points of the (feasible) generalized interpolation problem

$$\beta = \min_{f \in \mathcal{M}_{\mathrm{L}}(\mathbb{R})} \|\mathrm{L}f\|_{\mathcal{M}} \quad s.t. \quad \boldsymbol{\nu}(f) = \mathbf{y} \tag{37}$$

are necessarily nonuniform L*-splines of the form*

$$s(x) = \sum_{n=1}^{N_0} b_n p_n(x) + \sum_{k=1}^{K} a_k \rho_{\mathrm{L}}(x - \tau_k) \tag{38}$$

with parameters $\mathbf{b} = (b_1, \ldots, b_{N_0}) \in \mathbb{R}^{N_0}$, $K \leq (M - N_0)$ *(effective number of knots),* $\{\tau_k\}_{k=1}^K$ *with* $\tau_k \in \mathbb{R}$, *and* $\mathbf{a} = (a_1, \ldots, a_K) \in \mathbb{R}^K$. *Here,* $\{p_n\}_{n=1}^{N_0}$ *is a basis of* $\mathcal{N}_{\mathrm{L}}$ *and* $\mathrm{L}\rho_{\mathrm{L}} = \delta$ *so that* $\beta = \|\mathrm{L}s\|_{\mathcal{M}} = \sum_{k=1}^{K} |a_k| = \|\mathbf{a}\|_1$. *The full solution set of* (37) *is the weak*-closed convex hull of those extremal points.*

Hence, we only need to ensure that the underlying mathematical hypotheses are met for the spline-admissible operator $\mathrm{L} = \mathrm{D}^2$ and $\nu_m = \delta(\cdot - x_m)$.

- Weak* continuity of sampling functionals. This is taken care by the second statement in Theorem 1, the proof of which is quite involved because $\mathrm{BV}^{(2)}(\mathbb{R})$ is non-reflexive.
- Well-posedness of reconstruction for $f \in \mathcal{N}_{\mathrm{D}^2} = \mathrm{span}\{1, x\}$. It is well-known that the classical linear-regression problem

$$\mathbf{b} = \arg\min_{b_1,b_2} \sum_{m=1}^{M} |y_m - (b_1 + b_2 x_m)|^2$$

  is well-posed and has a unique solution if and only if $S = \{x_m\}_{m=1}^M$ contains at least two distinct points, say, $x_1 \neq x_2$, which takes care of the final hypothesis in Theorem 15.

■

Acknowlegdments

The research was partially supported by the Swiss National Science Foundation under Grant 200020-184646. The author is thankful to Julien Fageot, Shayan Aziznejad, Anais Badoual, Kyong Hwan Jin, and Harshit Gupta for helpful interactions.

References

Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. Learning activation functions to improve deep neural networks. In *Proc. Int. Conf. Learn. Representations, arXiv:1412.6830*, 2015.

Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *preprint arXiv:1611.01491*, 2016.

Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

Bastian Bohn, Michael Griebel, and Christian Rieger. A representer theorem for deep kernel learning. *arXiv:1709.10441v3*, 2018.

Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

Carl de Boor and Robert E. Lynch. On splines and their minimum properties. *Journal of Mathematics and Mechanics*, 15(6):953–969, 1966.

David L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006.

Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

S. D. Fisher and J. W. Jerome. Spline solutions to L_1 extremal problems in one and several variables. *Journal of Approximation Theory*, 13(1):73–83, 1975.

Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *Proceedings of Machine Learning Research*, 28(3):1319–1327, 2013.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*, volume 1. MIT press Cambridge, 2016.

Stefano Guarnieri, Francesco Piazza, and Aurelio Uncini. Multilayer feedforward networks with adaptive spline activation function. *IEEE Transactions on Neural Networks*, 10(3): 672–683, 1999.

Harshit Gupta, Julien Fageot, and Michael Unser. Continuous-domain solutions of linear inverse problems with Tikhonov *versus* generalized TV regularization. *IEEE Transactions on Signal Processing*, 66(17):4670–4684, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-Rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Thomas Hofmann, Bernhart Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

Le Hou, Dimitris Samaras, Tahsin Kurc, Yi Gao, and Joel Saltz. ConvNets with smooth adaptive activation functions for regression. In *Artificial Intelligence and Statistics*, pages 430–439, 2017.

George Kimeldorf and Grace Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

Stephen H. Lane, Marshall Flax, David Handelman, and Jack Gelfand. Multi-layer perceptrons with B-spline receptive field functions. In *Advances in Neural Information Processing Systems*, pages 684–692, 1991.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

Enno Mammen and Sara van de Geer. Locally adaptive regression splines. *Annals of Statistics*, 25(1):387–413, 1997.

Charles A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.

Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

Guido F. Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2924–2932, 2014.

Tomaso Poggio and Federico Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247(4945):978–982, 1990.

Tomaso Poggio and Steve Smale. The mathematics of learning: Dealing with data. *Notices of the AMS*, 50(5):537–544, 2003.

Tomaso Poggio, Lorenzo Rosasco, Amnon Shashua, Nadav Cohen, and Fabio Anselmi. Notes on hierarchical splines, DCLNs and i-theory. Technical report, Center for Brains, Minds and Machines (CBMM), 2015.

P.M. Prenter. *Splines and Variational Methods*. Wiley, New York, 1975.

Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics. Vol. 1: Functional Analysis*. Academic Press, 1980.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 9351, pages 234–241. Springer, LNCS, 2015.

Walter Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, 3rd edition, 1987.

Walter Rudin. *Functional Analysis*. McGraw-Hill, New York, 2nd edition, 1991. McGraw-Hill Series in Higher Mathematics.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

I.J. Schoenberg. Spline functions and the problem of graduation. *Proceedings of the National Academy of Sciences*, 52(4):947–950, 1964.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

Bernhard Schölkopf, Kah-Kay Sung, Chris J. C. Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45 (11):2758–2765, 1997.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In David Helmbold and Bob Williamson, editors, *Computational Learning Theory*, pages 416–426. Springer Berlin Heidelberg, 2001.

Larry L. Schumaker. *Spline Functions: Basic Theory*. Wiley, New York, 1981.

Laurent Schwartz. *Théorie des Distributions*. Hermann, Paris, 1966.

Gilbert Strang. The functions of deep learning. *SIAM News*, 51(10):1,4, 2018.

J.M. Tarela and M.V. Martinez. Region configurations for realizability of lattice piecewise-linear models. *Mathematical and Computer Modelling*, 30(11):17–27, 1999.

Michael Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, 1999.

Michael Unser, Julien Fageot, and Harshit Gupta. Representer theorems for sparsity-promoting ℓ_1 regularization. *IEEE Transactions on Information Theory*, 62(9):5167–5180, 2016.

Michael Unser, Julien Fageot, and John Paul Ward. Splines are universal solutions of linear inverse problems with generalized-TV regularization. *SIAM Review*, 59(4):769–793, 2017.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

Lorenzo Vecci, Francesco Piazza, and Aurelio Uncini. Learning and approximation capabilities of adaptive spline activation function neural networks. *Neural Networks*, 11(2): 259–270, 1998.

Grace Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

Shuning Wang and Xusheng Sun. Generalization of hinging hyperplanes. *IEEE Transactions on Information Theory*, 51(12):4425–4431, 2005.

Holger Wendland. *Scattered Data Approximations*. Cambridge University Press, 2005.

Journal of Machine Learning Research 20 (2019) 1-40 Submitted 7/18; Revised 4/19; Published 6/19

An Efficient Two Step Algorithm for High Dimensional Change Point Regression Models Without Grid Search

Abhishek Kaul ABHISHEK.KAUL@WSU.EDU
Department of Mathematics and Statistics
Washington State University
Pullman, WA 99164, USA.

Venkata K. Jandhyala JANDHYALA@WSU.EDU
Department of Mathematics and Statistics
Washington State University
Pullman, WA 99164, USA.

Stergios B. Fotopoulos FOTOPO@WSU.EDU
Department of Finance and Management Science
Washington State University
Pullman, WA 99164, USA.

Editor: Ryan Tibshirani

Abstract

We propose a two step algorithm based on ℓ_1/ℓ_0 regularization for the detection and estimation of parameters of a high dimensional change point regression model and provide the corresponding rates of convergence for the change point as well as the regression parameter estimates. Importantly, the computational cost of our estimator is only $2\cdot\text{Lasso}(n,p)$, where $\text{Lasso}(n,p)$ represents the computational burden of one Lasso optimization in a model of size (n,p). In comparison, existing grid search based approaches to this problem require a computational cost of at least $n\cdot\text{Lasso}(n,p)$ optimizations. Additionally, the proposed method is shown to be able to consistently detect the case of 'no change', i.e., where no finite change point exists in the model. We allow the true change point parameter τ_0 to possibly move to the boundaries of its parametric space, and the jump size $\|\beta_0-\gamma_0\|_2$ to possibly diverge as n increases. We then characterize the corresponding effects on the rates of convergence of the change point and regression estimates. In particular, we show that, while an increasing jump size may have a beneficial effect on the change point estimate, however the optimal rate of regression parameter estimates are preserved only upto a certain rate of the increasing jump size. This behavior in the rate of regression parameter estimates is unique to high dimensional change point regression models only. Simulations are performed to empirically evaluate performance of the proposed estimators. The methodology is applied to community level socio-economic data of the U.S., collected from the 1990 U.S. census and other sources.

Keywords: Change point regression, High dimensional models, ℓ_1,ℓ_0 regularization, Rate of convergence, Two phase regression

©2019 Abhishek Kaul, Venkata K. Jandhyala and Stergios B. Fotopoulos.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-460.html.

1. Introduction

Regression models are fundamental to supervised learning and statistical modelling of data collected from scientific phenomena. While applying regression models, one often assumes the regression parameters to be stable over time. However, this assumption may be rigid and may not hold in several environmental, biological and economic models, particularly when data is collected over an extended period of time. There are several approaches to model this dynamic phenomenon in regression parameters. One approach is to let the parameters change at certain unknown time points of the sampling period (Hinkley (1970), Hinkley (1972), Jandhyala and MacNeill (1997), Bai (1997), Jandhyala and Fotopoulos (1999), Fotopoulos et al. (2010) and Jandhyala et al. (2013)). Another closely related approach is to formulate the change point based on one or more covariate thresholds (Hinkley (1969), Koul and Qian (2002) and Koul et al. (2003)). In the literature, it is common to broadly call both as change point regression models. Such dynamic models have been found to have wide ranging applications in all areas of scientific inquiry (Reeves et al. (2007), Lund et al. (2007), and Liu et al. (2013)).

Technological advances in the past two decades have led to the wide availability of large scale/high dimensional data sets in several areas of applications such as genomics, social networking, empirical economics, finance etc. This has led to rapid development of high dimensional statistical methods. A large body of literature has now been developed pertaining to the study of regression models capable of allowing a vastly larger number of parameters p than the sample size n. One of the most successful methods for analysing high dimensional regression models is the Lasso, which is based on the least squares loss and ℓ_1 regularization (Tibshirani (1996)). Innumerable investigations have since been carried out to study the behavior of the Lasso estimator and its various modifications in many different settings (see e.g., Zou (2006), Zhao and Yu (2006), Bickel et al. (2009), Belloni et al. (2011), Belloni et al. (2017a), Kaul (2014), Kaul and Koul (2015), and the references therein). For a general overview on the developments of Lasso and its variants we refer to the monograph of Bühlmann and Van De Geer (2011) and the review article of Tibshirani (2011). All aforementioned articles provide results in a regression setting where the parameters are dynamically stable. In the recent past, work has also been carried out in the context of high dimensional change point models in an 'only means' setup, where change occurs in only the mean of time ordered independent random vectors, with the dimension of the observation vector being larger than the number of observations (Cho and Fryzlewicz (2015), Fryzlewicz (2014), and Wang and Samworth (2018) among others). Here the change is characterized in the sense of a dynamic mean vector. Another context in which high dimensional change point models have been investigated is that of a dynamic covariance structure which is related to the study of evolving networks (Gibberd and Roy (2017), and Atchade and Bybee (2017)). In contrast, change point methods for high dimensional linear regression models have received much less attention and only a select few articles have considered this problem in the recent literature (Ciuperca (2014), Zhang et al. (2015), Leonardi and Bühlmann (2016), Lee et al. (2016), and Lee et al. (2018)).

In this paper, we consider a high dimensional linear regression model with a potential change point,

$$y_i = x_i^T\beta_0\mathbf{1}[w_i\le\tau_0] + x_i^T\gamma_0\mathbf{1}[w_i>\tau_0] + \varepsilon_i, \quad i=1,...,n. \tag{1.1}$$

The observed variables in model (1.1) are, $y_i \in \mathbb{R}$, the p-dimensional predictors $x_i \in \mathbb{R}^p$, and change inducing variable $w_i \in \mathbb{R}$, $i = 1, .., n$. The unknown parameters of interest are the regression parameters $\beta_0, \gamma_0 \in \mathbb{R}^p$, and the change point $\tau_0 \in \bar{\mathbb{R}}^\star := \mathbb{R} \cup \{-\infty\}$. The change point τ_0 represents a threshold value of the variable w subsequent to which the regression parameter changes from its initial value β_0 to a new value γ_0. Note that, the 'no change' case is allowed by the model (1.1), since we allow $\tau_0 = -\infty$, in its parametric space. In this case, model (1.1) reduces to an ordinary high dimensional linear regression model. The parametric space $\bar{\mathbb{R}}^\star$ of τ is restricted to only contain $-\infty$, and not ∞, since both of these points characterize the 'no change' scenario and are unidentifiable from each other. This differs from the usual characterization of the 'no change' case, which is typically defined by $\beta_0 = \gamma_0$ and $\tau_0 \in \mathbb{R}$, for e.g. in Lee et al. (2016) and Lee et al. (2018). However it should be understood that this difference is only notational and both are characterizing the same null model. It should also be noted that the change point $\tau_0 \in \bar{\mathbb{R}}^*$, may itself depend on n, i.e., as the sample size increases, the change point may shift its location. However, for clarity of exposition, this dependence is notionally suppressed in the rest of this article. Furthermore, we let $p >> n$, so that model (1.1) corresponds to a high dimensional setting. Also, consistent with current literature, we assume that only a total of s components of β_0 and γ_0 are non-zero, i.e., $\|\beta_0\|_0 + \|\gamma_0\|_0 \leq s$, where $s < n$.

Recently, models similar to (1.1) have been studied by Ciuperca (2014), Zhang et al. (2015), Leonardi and Bühlmann (2016), Lee et al. (2016) and Lee et al. (2018). Similar to model (1.1), Lee et al. (2016) and Lee et al. (2018) consider a high dimensional model with only a single unknown change point, whereas, Zhang et al. (2015), and Leonardi and Bühlmann (2016) consider a model where multiple change points may be present in the model. The articles Zhang et al. (2015) and Ciuperca (2014) consider a multiple change point setting where the dimension p of the regression parameters is fixed. The common thread in these articles is to provide regularized estimators for the parameters β, γ, τ and study their rates of convergence under different norms. In an earlier work, Wu (2008) provided an information-based criterion for carrying out change point analysis and variable selection in the fixed p setting; this methodology, however is not extendable to the high dimensional case. While these articles make important contributions to this fast emerging area, many aspects of this problem remain to be understood. For example, existing methods may be unable to satisfactorily detect the 'no change' case, these estimation methods may be computationally challenging to implement, and the underlying technical assumptions required for their theoretical validity may be restrictive.

The most commonly applied approach to estimating parameters of models such as (1.1) with a single change point is to consider,

$$(\hat{\beta}, \hat{\gamma}, \hat{\tau}) = \underset{\beta, \gamma \in \mathbb{R}^p, \tau \in \mathcal{T}}{\arg\min} \left\{ \frac{1}{n} \sum_{i=1}^{n} \text{loss}(\text{data}, \beta, \gamma, \tau) + \text{pen}(\beta, \gamma, \tau) \right\}, \tag{1.2}$$

where loss(data, β, γ, τ) is an appropriately chosen loss function, and pen(β, γ, τ) is a suitably defined penalty function on the parameters β, γ, τ (e.g., Lee et al. (2016), Lee et al. (2018) and (5.1)). Here $\mathcal{T}$ is a restriction on the parametric space of the change point parameter τ. The loss function in (1.2) is nonconvex and consequently a direct optimization of (1.2) is typically computationally infeasible. To circumvent this difficulty, the space $\mathcal{T}$

is usually broken into a grid of points $\mathcal{T}^*$, and $\hat{\beta}(\tau), \hat{\gamma}(\tau)$ are computed for each $\tau \in \mathcal{T}^*$. The estimate $\hat{\tau}$ of the change point τ_0 is then obtained as that $\tau \in \mathcal{T}^*$ which minimizes the objective function in (1.2) over $\hat{\beta}(\tau), \hat{\gamma}(\tau)$. When the loss function is least squares and the penalty is of an ℓ_1-type, the computational cost of the above grid search is $|\mathcal{T}^*|\text{Lasso}(n, p)$, where $|\mathcal{T}^*|$ is typically of order n. Note that this grid search mechanism becomes computationally intensive as n, p increase. In the case of multiple change points, Zhang et al. (2015), and Leonardi and Bühlmann (2016) provide dynamic programming approaches that can estimate the number and locations of change points with the same $n\text{Lasso}(n, p)$ computational cost.

In this article we develop a two step algorithm for detection and estimation of parameters of model (1.1), so that a full grid search is avoided even as the near optimal rates of all parameter estimates are preserved. The idea for developing such an algorithm originates from the following simple and yet surprising numerical observation. Suppose we first choose virtually any initial value $\tau^{(0)} \in \text{Supp}(w)$, separated from its boundaries and then compute regression coefficients $\hat{\beta}^{(0)}$, $\hat{\gamma}^{(0)}$ on the binary partition $\{i;\, i \in \{1, .., n\},\ w_i \leq \tau^{(0)}\}$ and $\{i;\, i \in \{1, .., n\},\ w_i > \tau^{(0)}\}$ respectively. Then a single update of the change point estimate obtained by optimization of the least squares loss over the change point parameter, using the previously obtained regression parameter estimates $\hat{\beta}^{(0)}$, $\hat{\gamma}^{(0)}$, yields a very precise estimate of the unknown change point, where the precision of this estimate is indistinguishable from existing full grid search approaches in any uniform sense. This simple numerical observation is very surprising, since it suggests that any initial $\tau^{(0)}$ which carries even a 'fractional amount of information' on the unknown τ_0 (this notion is described precisely later in Section 2), can be utilized to obtain an updated estimate $\hat{\tau}^{(1)}$ in a single step, which lies in a near optimal neighborhood of τ_0. In other words, the single step update process pulls in the initial guess $\tau^{(0)}$ from a much wider (nearly arbitrary) neighborhood of τ_0, to a near optimal neighborhood of τ_0. The usefulness of this process is immediate, as it removes the necessity of a grid search. The main contribution of this article is to develop a mathematical treatment of this two step algorithm. In doing so we also allow the possibility of 'no change' in the model (1.1).

More precisely, in this article we propose estimators based on ℓ_1/ℓ_0 regularization for the parameters τ_0, β_0, and γ_0 of model (1.1). The proposed methodology completely avoids a grid search approach for locating the unknown change point, consequently has a computational cost of only $2\text{Lasso}(n, p)$, significantly below the $n\text{Lasso}(n, p)$ cost of existing methods. A second important novelty of the proposed method, is its ability to detect the 'no change' case, which is achieved by a ℓ_0 regularization in the change point estimator. From a technical perspective, the rates of convergence associated with the proposed estimators are such that they are optimal for the regression parameter estimates and match the best rate of convergence available in the literature for estimating the change point. Before we describe our proposed methodology in Section 2, we outline below the notations used in this paper.

Notation: Throughout the paper, for any vector $\delta \in \mathbb{R}^p$, $\|\delta\|_0$ represents the number of non-zero components in δ. The norms $\|\delta\|_1$ and $\|\delta\|_2$ represent the standard 1-norm and Euclidean norm, respectively. The norm $\|\delta\|_\infty$ represents the sup norm, i.e., the maximum of absolute values of all elements. For any set of indices $T \subseteq \{1, ..., p\}$, let $\delta_T = (\delta_j)_{j \in T}$ represent a sub-vector of δ containing components corresponding to the indices in T. Also,

we let $|T|$ represent the cardinality of the set T. The notation $\mathbf{1}[\cdot]$ represents the usual indicator function. We denote by $a \wedge b = \min\{a, b\}$, and $a \vee b = \max\{a, b\}$, for any $a, b \in \mathbb{R}$. In the following, let $\mathrm{Supp}(w)$ represent the support of the distribution of w and $\Phi(\cdot)$ be its cdf. Also denote by $\Phi_{\min}(\tau_0) = \Phi(\tau_0) \wedge (1 - \Phi(\tau_0))$. We shall use the following notation to represent generic constants that may be different from one term to the next. For example, $0 < c_u < \infty$ represent universal constants, whereas $0 < c_m < \infty$ are constants that depend on model parameters such as variance parameters of underlying distributions. The generic constants $0 < c_1, c_2 < \infty$ are used to denote constants that may depend on both c_u, and c_m. Lastly, we shall denote by $\bar{\mathbb{R}}^\star := \mathbb{R} \cup \{-\infty\}$, as the extended Euclidean space with negative infinity included. In the following, without loss of generality we assume that $\mathrm{Supp}(w) = \mathbb{R}$.

2. Methodology and Related Work

We begin this section by describing the proposed methodology for the detection and estimation of parameters of model (1.1). For this purpose we require the following notation. Let for any $\tau \in \bar{\mathbb{R}}^\star$, $\beta, \gamma \in \mathbb{R}^p$,

$$Q(\tau, \beta, \gamma) \quad = \quad \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2\mathbf{1}[w_i \le \tau] + \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\gamma)^2\mathbf{1}[w_i > \tau].^{a} \tag{2.1}$$

Additionally define the following Condition I that shall serve as an initializing condition required for the construction of our proposed algorithm.

Condition I: Let $u_n^{(0)}$ be a non-negative sequence defined as,

$$u_n^{(0)} = 1 \wedge c_u\Big(\frac{s\log p}{nl_n^2}\Big)^{\frac{1}{k}}, \quad \text{for any constants, } k \in [1, \infty), \text{ and } c_u > 0,^{b} \tag{2.2}$$

where $0 < l_n \le 1/2$ is any non-negative sequence. Then, assume that the initializer $\tau^{(0)}$ satisfies,

$$\Phi_{\min}(\tau^{(0)}) \ge c_u l_n, \quad \text{and} \quad |\Phi(\tau^{(0)}) - \Phi(\tau_0)| \le u_n^{(0)}. \tag{2.3}$$

In the above Condition I, the sequence l_n and the constant k are arbitrary, subject to satisfying Condition A(iii) to follow. The rate of the sequence l_n shall control the ability of Algorithm 1 to detect a finite change point near the boundaries of $\mathbb{R}$. Specifically Algorithm 1 shall be able to detect a finite change point (when it exists), such that $\Phi_{\min}(\tau_0)$ is of order at least that of l_n. In the case where it is assumed that, either $\Phi(\tau_0) = 0$ or $\Phi_{\min}(\tau_0) \ge c_u > 0$, then we can set $l_n \ge c_u$. Here, Algorithm 1 will be able to distinguish between the two cases, whether (a) there is no change point, $\Phi(\tau_0) = 0$ or (b) there is a finite change point $\tau_0 \in \mathbb{R}$ such that $\Phi(\tau_0)$ is bounded below.[c]

Then, the two step algorithm which we propose to obtain change point and regression coefficient estimates is described in Algorithm 1 below.

a. Here, define $\mathbf{1}[w_i \le \tau] = 0$, for $\tau = -\infty$.

b. Note here that the constant k is arbitrary, hence it can itself depend on initial $\tau^{(0)}$, i.e., the farther the guess $\tau^{(0)}$ is from τ_0 the larger k can be chosen in order to satisfy Condition I.

c. The quantities l_n and k are only required for analysis of Algorithm 1. These quantities play no role in its implementation.

Algorithm 1: Detection and estimation of change point and regression parameters

Step 0 (Initialize): Choose any initial value $\tau^{(0)} \in \mathbb{R}$ satisfying Condition I. Compute the initial regression parameter estimates,

$$\big(\hat{\beta}^{(0)}, \hat{\gamma}^{(0)}\big) = \underset{\beta,\gamma\in\mathbb{R}^p}{\arg\min}\Big\{Q(\tau^{(0)}, \beta, \gamma) + \lambda_1\|(\beta^T, \gamma^T)^T\|_1\Big\}, \quad \lambda_1 > 0.$$

Step 1: Update $\tau^{(0)}$ to obtain the change point estimate $\hat{\tau}^{(1)}$ where [d],

$$\hat{\tau}^{(1)} = \underset{\tau\in\bar{\mathbb{R}}^\star}{\arg\min}\Big\{Q(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) + \mu\|\Phi(\tau)\|_0\Big\}, \quad \mu > 0. \tag{2.4}$$

Step 2: Update $(\hat{\beta}^{(0)}, \hat{\gamma}^{(0)})$ to obtain regression parameter estimates $(\hat{\beta}^{(1)}, \hat{\gamma}^{(1)})$ where,

$$\big(\hat{\beta}^{(1)}, \hat{\gamma}^{(1)}\big) = \underset{\beta,\gamma\in\mathbb{R}^p}{\arg\min}\Big\{Q(\hat{\tau}^{(1)}, \beta, \gamma) + \lambda_2\|(\beta^T, \gamma^T)^T\|_1\Big\}, \quad \lambda_2 > 0.$$

A first concern that may arise to reader regarding **Step 0** of Algorithm 1 pertains to the initializing conditions in (2.3) of Condition I. The first of these conditions is clearly innocuous, all it requires is the initial user chosen $\tau^{(0)}$ to be marginally away from the boundaries of $\mathbb{R}$. The second condition in (2.3) requires that the initial value $\tau^{(0)}$ be in an $u_n^{(0)}$-neighborhood of τ_0. While at first, this might come across as a limitation of the algorithm, however the following discussion shall show how broad this $u_n^{(0)}$-neighborhood truly can be. First note that the constant $k \in [1, \infty)$ is arbitrarily large, subject to Condition A(iii), i.e., this condition is adaptable to the user chosen value of the initializer $\tau^{(0)}$. In other words, the farther the user chosen $\tau^{(0)}$ is from the true change point τ_0, the larger the value of k can be, in order to satisfy this condition. Additionally, note that the largest possible distance (in the cdf scale) between any two $\tau_1, \tau_2 \in \mathbb{R}$, is such that $|\Phi(\tau_1) - \Phi(\tau_2)| \le 1$. Now for $c_u = 1$, consider first the disallowed case of $k = \infty$, then the initial condition is trivially satisfied, since $|\Phi(\tau^{(0)}) - \Phi(\tau_0)| \le 1$. Thus, virtually any initial value in the parametric space of τ_0, separated from its boundaries, will satisfy the required initial condition for a large enough $k \in [1, \infty)$, thereby illustrating that this initial condition is infact very mild. Remarkably, one of our main results shows that, under suitable conditions, the updated change point estimate $\hat{\tau}^{(1)}$ of **Step 1** of Algorithm 1, will satisfy optimal error bounds, irrespective of the value of k. Simply stated, the update in **Step 1** sharpens the initial change point guess from any arbitrary fractional rate to a near optimal rate. The condition (2.3), also provides a precise description of the notion of 'fractional information' mentioned in the introduction section. The sequence $u_n^{(0)}$ forms a metric measuring the amount of information in the guess $\tau^{(0)}$ about τ_0, and the existence of a finite $k < \infty$ provides a way of saying that the guess $\tau^{(0)}$ possesses some fractional amount of information on τ_0.

To numerically illustrate this surprising phenomenon, in Section 5 we use the 'no information' initializer $\tau^{(0)} = w^{(0.5)}$, i.e. the 50th percentile of w.[e] Note that this choice is

d. Note that while the initializing $\tau^{(0)}$ in **Step 0** is chosen in $\mathbb{R}$, however the optimization in **Step 1** is performed over the extended Euclidean space $\bar{\mathbb{R}}^\star = \mathbb{R} \cup \{-\infty\}$.

e. The 50th percentile is the best 'no information' guess, since it is the empirical guess that is equidistant to the ends of the support of w.

the most sensible value of the initializer in the absence of any information about τ_0. These numerical results provide strong evidence to support Algorithm 1 by showing that the precision of the estimates obtained from the proposed method are infact indistinguishable from existing grid search type approaches, and are obtained with a small fraction of the computational burden. Another equivalent way of viewing the above discussion is that, if we pick two distinct initializers $\tau_1^{(0)}$ and $\tau_2^{(0)}$ carrying some fractional information about τ_0, i.e., they satisfy the initializing condition for some $1 \leq k_1 < k_2 < \infty$, respectively ($\tau_1^{(0)}$ is closer to τ_0 than $\tau_2^{(0)}$), then, the corresponding updated change point estimates $\hat{\tau}_1^{(1)}$, $\hat{\tau}_2^{(1)}$ will both be in a near optimal neighborhood of τ_0. This basically implies that the quality of the guess does not influence the updated estimate in its eventual rate of convergence. This observation is also numerically illustrated in Section 5.

Finally, we also mention that a slight relaxation of Condition I is also possible, specifically, the sequence $u_n^{(0)}$ in this condition can instead be relaxed to $u_n^{(0)} = 1 \wedge c_u(\frac{1}{n})^{1/k}$, for any constants $k \in [1, \infty)$ and any $c_u > 0$, while preserving all results to follow. However, Condition I in its current form provides a coherence with other noise bounds that show up in the statement and proof of our results, and thus is a notationally convenient representation. Furthermore, this relaxation of Condition I shall not yield error bounds that are any sharper than those to follow.

A second concern that may arise to the reader regarding implementation of Algorithm 1 is the feasibility of implementing **Step 1**. At first, this optimization seems intractable owing to its nonsmooth, nonconvex (with no apparent convex relaxations) construction. However, upon closer inspection, it is observed that the loss function $Q(\,\cdot\,, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)})$ in **Step 1** is a step function with step changes occurring at any point on the one-dimensional grid $(-\infty, w_1, w_2, ..., w_n)^T$. Secondly, the ℓ_0 term in the objective function only depends on whether $\Phi(\tau)$ is zero or non zero. This implies that the distance between any two τ_1 and τ_2 does not influence the value of the ℓ_0 norm (note that this will not be the case if instead an ℓ_1 norm is used). These two observations together imply that any global optimum achieved in the extended Euclidean space $\bar{\mathbb{R}}^\star$ will also be attained at some point on the finite grid $(-\infty, w_1, w_2, ..., w_n)^T$. An illustration of this step behavior is provided in Figure 1.

A final concern in implementing **Step 1** is that it requires knowledge of the distribution function $\Phi(\cdot)$, which is typically unknown. This concern is also easily overcome upon observing that the objective function in **Step 1** is a step function over the grid $(-\infty, w_1, w_2, ..., w_n)^T$. Specifically, on this grid, the term $\|\Phi(\tau)\|_0 = \|\tau\|_0^*$, where $\|\tau\|_0^* = 1$, if $\tau \in \{w_1, ..., w_n\}$ and $\|\tau\|_0^* = 0$, if $\tau = -\infty$.

In view of the above discussion, **Step 1** of Algorithm 1 can be replaced by the following optimization,

$$\hat{\tau}^{(1)} = \underset{\tau \in \{-\infty\} \cup \{w_1, .., w_n\}}{\arg\min} \Big\{ Q(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) + \mu \|\tau\|_0^* \Big\}, \quad \mu > 0. \tag{2.5}$$

Thus, the optimization (2.4) in **Step 1** of Algorithm 1 is reduced to the optimization (2.5), which can be easily solved in negligible time by explicitly computing n values of the objective function and then locating the minimum.

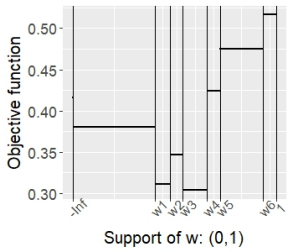


Figure 1: Step behavior of the function $Q(\cdot\,, \hat{\beta}, \hat{\gamma}) + \mu\|\Phi(\tau)\|_0$, with $\mu = 0.1$, evaluated over grid of points $\{-\infty\} \cup \{0, 0.01, ..., 1\}$. Here $w_i \sim \mathcal{U}(0,1)$, $n = 6$, $\tau_0 = 0.25$, $p = 3$, $\beta_0 = (1,0,0)^T$, $\gamma_0 = (0,1,0)^T$, and we use $\hat{\beta} = (0.41, 0, 0)^T$, $\hat{\gamma} = (0.13, 0.92, 0)^T$. The realizations $\{w_1, .., w_n\}$ have been sorted (ascending) in the illustration. Observe that step changes occur at $-\{\infty\} \cup \{w_1, w_2, ..., w_n\}$.

Another note of interest is the convenience of separability in computing the optimizers $\hat{\beta}, \hat{\gamma}$ in **Step 0** and **Step 2**, i.e., for any fixed τ, we can obtain

$$\hat{\beta}(\tau) = \underset{\beta \in \mathbb{R}^p}{\arg\min} \Big\{ \frac{1}{n} \sum_{i;\, w_i \leq \tau} (y_i - x_i^T \beta)^2 + \lambda_1 \|\beta\|_1 \Big\}, \tag{2.6}$$

$$\hat{\gamma}(\tau) = \underset{\beta \in \mathbb{R}^p}{\arg\min} \Big\{ \frac{1}{n} \sum_{i;\, w_i > \tau} (y_i - x_i^T \gamma)^2 + \lambda_1 \|\gamma\|_1 \Big\}. \tag{2.7}$$

These are ordinary Lasso optimizations and can be carried out by any one of the several methods available in the literature. Some of these methods include, coordinate or gradient descent algorithms (see, e.g. Hastie et al. (2015)), or via interior point methods for linear optimization under second order conic constraints (see, e.g., Koenker and Mizera (2014)).

The main results of this article establish selection consistency of the unknown change point and provide finite sample bounds for the error in estimates obtained from Algorithm 1 under suitable conditions. Let $\xi_n := \|\beta_0 - \gamma_0\|_2$ be the jump size between the pre and post regression parameters. Then the specific results we derive are,

$$(i) \quad \text{If } \Phi(\tau_0) = 0, \text{ then } \Phi(\hat{\tau}^{(1)}) = 0, \tag{2.8}$$

$$(ii) \quad \text{If } \Phi_{\min}(\tau_0) \geq c_u l_n, \text{ then } |\Phi(\hat{\tau}^{(1)}) - \Phi(\tau_0)| \leq t_n := c_u c_m \max\Big\{ \frac{s \log p}{n}, \frac{1}{(1 \vee \xi_n^2) l_n^2} \frac{s \log p}{n} \Big\},$$

$$(iii) \quad \big\|\hat{\beta}^{(1)} - \beta_0\big\|_q \leq c_u c_m s^{1/q} \frac{1}{\Phi_{\min}(\tau_0)} \max\Big\{ \sqrt{\frac{\log p}{n}},\ \xi_n t_n \Big\}, \quad q = 1, 2,$$

$$(iv) \quad \big\|\hat{\gamma}^{(1)} - \gamma_0\big\|_q \leq c_u c_m s^{1/q} \frac{1}{\Phi_{\min}(\tau_0)} \max\Big\{ \sqrt{\frac{\log p}{n}},\ \xi_n t_n \Big\}, \quad q = 1, 2,$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$, for n sufficiently large.

These and other related results about estimates from Algorithm 1 are covered in Sections 3 and 4. The sharpness and/or near optimality of the above bounds may be observed from

the following special case. Upon letting $\xi_n s\sqrt{\log p/n} \to 0$, and $l_n \geq c_u$, in (2.8), the last three results of (2.8) reduce to,

$$(i) \quad \text{If } \Phi_{\min}(\tau_0) \geq c_u l_n, \text{ then } |\Phi(\hat{\tau}^{(1)}) - \Phi(\tau_0)| \leq c_u c_m \frac{s \log p}{n}, \tag{2.9}$$

$$(ii) \quad \left\|\hat{\beta}^{(1)} - \beta_0\right\|_q \leq c_u c_m s^{\frac{1}{q}} \sqrt{\frac{\log p}{n}}, \qquad q = 1, 2,$$

$$(iii) \quad \left\|\hat{\gamma}^{(1)} - \gamma_0\right\|_q \leq c_u c_m s^{\frac{1}{q}} \sqrt{\frac{\log p}{n}}, \qquad q = 1, 2,$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$, for n sufficiently large.

In an ordinary high dimensional linear regression model without change points, it has been shown that the optimal rate of convergence for regression estimates is $\sqrt{s \log p/n}$ under the ℓ_2 norm (see, e.g.,Ye and Zhang (2010), Raskutti et al. (2011), and Belloni et al. (2017b)). This implies that the rate of convergence of the regression estimates from Algorithm 1 (which stops after one iteration) cannot be uniformly improved upon by carrying out further iterations (over subgaussian distributions). Also, the rate of convergence of the change point estimate in (2.9) is the fastest available rate in the literature. We shall now state the conditions under which the results of this article are derived.

Condition A (assumptions on model parameters):

(i) Let $S = S_1 \cup S_2$, where $S_1 = \{j; \beta_{0j} \neq 0\}$ and $S_2 = \{j; \gamma_{0j} \neq 0\}$. Then for some $s = s_n \geq 1$, we assume that $|S| \leq s$.

(ii) The model dimensions s, p, n, satisfy $s \log p/n \to 0$. Additionally, the sequence l_n of Condition I satisfies $s \log p/n l_n^2 \to 0$.

(iii) If a finite change point exists, i.e., $\Phi_{\min}(\tau_0) > 0$, then the sequence l_n and constant $k \in [1, \infty)$ of Condition I satisfy

$$\frac{s}{l_n^2}\Big(\frac{s \log p}{n l_n^2}\Big)^{\frac{1}{k}} \to 0.$$

Additionally, in this case $\Phi_{\min}(\tau_0) \geq c_u l_n$.

(iv) If $\Phi_{\min}(\tau_0) > 0$, then the jump size is bounded below by a constant, i.e, $\xi_n := \|\beta_0 - \gamma_0\|_2 > c > 0$.

Condition B (assumptions on model distributions):

(i) The vectors $x_i = (x_{i1}, ..., x_{ip})^T$, $i = 1, .., n$, are i.i.d subgaussian[f] with mean vector zero, and variance parameter $\sigma_x^2 \leq C$. Furthermore, the covariance matrix $\Sigma := E x_i x_i^T$ has bounded eigenvalues, i.e., $0 < \kappa \leq \text{mineigen}(\Sigma) < \text{maxeigen}(\Sigma) \leq \phi < \infty$.

(ii) The errors ε_i's are i.i.d. subgaussian with mean zero and variance parameter $\sigma_\varepsilon^2 \leq C$.

(iii) The variables w_i, $i = 1, ..., n$ are i.i.d r.v.'s (continuous or discrete), with its cdf $\Phi(a) = P(w_i \leq a)$, $a \in \mathbb{R}$.

f. Recall that for $\alpha > 0$, the random variable η is said to be α-subgaussian if, for all $t \in \mathbb{R}$, $E[\exp(t\eta)] \leq \exp(\alpha^2 t^2/2)$. Similarly, a random vector $\xi \in \mathbb{R}^p$ is said to be α-subgaussian if the inner products $\langle \xi, v \rangle$ are α-subgaussian for any $v \in \mathbb{R}^p$ with $\|v\|_2 = 1$.

(iv) The r.v.'s x_i, w_i, ε_i are independent of each other.

Conditions A(i) and A(ii) together form the usual sparsity assumption of high dimensional models. Conditions A(ii) and A(iii) are both restrictions on the model dimensions and in fact A(ii) is implied by A(iii) when it is applicable. However, both conditions are stated here since some of our results in Sections 3 and 4, hold under the weaker Condition A(ii). The Condition A(iii) is on the model parameters and also related to the initial condition of Condition I, via the sequence l_n and the constant $k \in [1, \infty)$. This condition assumes the only additional control on how large a constant k and how small the sequence l_n, can be tolerated by Algorithm 1, given the model dimensions. Heuristically, this condition ensures that the fractional information possessed by the initial guess $\tau^{(0)}$, is not dominated by the noise induced in the linear system due to its large dimensions. Note that Condition A(iii) is only assumed if a change point exists, i.e., $\Phi_{\min}(\tau_0) > 0$. In the case of 'no change' in model (1.1), i.e., $\Phi(\tau_0) = 0$, any initial value $\tau^{(0)}$ satisfying $\Phi(\tau^{(0)}) \geq c_u l_n$, i.e., separated from the boundaries of $\mathbb{R}$, can be used to initialize Algorithm 1. A secondary purpose that Condition A(iii) serves is to ensure that if a finite change point exists, then, to keep $\Phi_{\min}(\tau_0) \geq c_u l_n$ away from the boundaries of $(0, 1)$, whenever a finite change point exists in the model. Note that, this condition does not assume lower boundedness of $\Phi_{\min}(\tau_0)$ as is commonly the case in the literature, since the sequence l_n may converge to zero. Finally, Condition A(iv) requires that if a finite change point exists, then the corresponding jump size ξ_n is bounded below. We also mention here that we do not make any assumptions on the upper bound of ξ_n, and this jump size is allowed to possibly diverge with n.

The subgaussian assumptions in Conditions B(i) and B(ii) are now standard in high dimensional linear regression models and are known to accommodate a large class of random designs. In ordinary high dimensional linear regression, these assumptions are used to establish well behaved restricted eigenvalues of the Gram matrix $\sum x_i x_i^T/n$ (Raskutti et al. (2010), and Rudelson and Zhou (2012)), which are in turn used to derive convergence rates of ℓ_1 regularized estimators (Bickel et al. (2009), and several others). These conditions play a similar role in our change point setup.

One main advantage of the proposed Algorithm 1 over existing methods is its ability to provide near optimal estimates without a grid search. As mentioned earlier in the article, the computational cost of Algorithm 1 is 2Lasso(n, p), significantly below the nLasso(n, p) cost of existing methods and is thus scalable to deal with large data. A novelty of Algorithm 1 in comparison to those proposed in Leonardi and Bühlmann (2016), Lee et al. (2016) and Lee et al. (2018) is its ability to detect the case where $\Phi(\tau_0) = 0$. This is relevant since it removes the necessity to pre-test for the existence of a change point. In contrast, while the methods of Lee et al. (2016), and Lee et al. (2018) are implementable in the case of no change point, they are however unable to detect the absence of the change point. Instead, in this case of $\Phi(\tau_0) = 0$, these methods return a valid $2p$ dimensional estimate $(\hat{\gamma}^T, \hat{\alpha}^T)^T$, where $\alpha_0 = \beta_0 - \gamma_0$, that can be used for predictive purposes using the model (1.1). Note that, the ability to detect the absence of a change point is a stronger statement and may provide additional relevant information, while also preserving the interpretable p dimensional linear regression model in the case where $\Phi(\tau_0) = 0$.

The organization of the remainder of this article is as follows. Section 3 develops preliminary results required for analysis of estimates given by Algorithm 1 and Section 4 provides the main results regarding estimates obtained from Algorithm 1. Proofs of all results are

given in Appendix A, while Appendix B consists of some relevant auxiliary results from the literature, stated without proofs. The performance of Algorithm 1 is empirically evaluated in Section 5. In this numerical section, the implementation of Algorithm 1 assumes no prior information of the unknown change point τ_0, additionally we numerically illustrate that the quality of the initial guess has no discernible impact on the final estimates and finally, we also show that the precision of the proposed estimates is indistinguishable from grid search approaches. Section 6 consists of an application of the proposed methodology to socio-economic data of U.S. collected from the 1990 U.S. census, and other sources.

3. Preliminary Results

In this section we present preliminary results that are important for stating and proving our main results in Section 4. First, for any fixed τ, we define

$$\zeta_i(\tau) = \begin{cases} \mathbf{1}[\tau_0 < w_i \leq \tau], & \text{if } \tau > \tau_0 \\ \mathbf{1}[\tau \leq w_i < \tau_0], & \text{if } \tau \leq \tau_0. \end{cases}$$

Then clearly, for any fixed $\tau \in \mathbb{R}$, $E\zeta_i(\tau) := \Phi^*(\tau_0, \tau) = |\Phi(\tau) - \Phi(\tau_0)|$. We shall now state a key result that uniformly controls (over τ) the quantity $n^{-1}\sum_{i=1}^n \zeta_i(\tau)$.

Lemma 3.1 *Let u_n, and v_n be any non-negative sequences such that $v_n \geq c\log p/n$, $c > 0$ and let $\mathcal{T}(\tau_0, u_n) = \{\tau : \Phi^*(\tau_0, \tau) \leq u_n\}$ be a u_n-neighborhood of τ_0. Then under Condition B(iii), we have,*

$$(i) \sup_{\tau \in \mathcal{T}(\tau_0, u_n)} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau) \leq c_u \max\Big\{\frac{\log p}{n}, u_n\Big\}, \qquad (ii) \inf_{\substack{\tau \in \mathbb{R}; \\ \Phi^*(\tau_0,\tau) \geq v_n}} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau) \geq c_u v_n,$$

with probability at least $1 - c_1\exp(-c_2\log p)$.

To proceed further, define for any τ the following set of random indices,

$$n_w := n_w(\tau_0, \tau) = \begin{cases} i \in \{1, ..., n\}; & \tau_0 < w_i \leq \tau, \quad \text{if } \tau \geq \tau_0, \\ i \in \{1, ..., n\}; & \tau \leq w_i < \tau_0, \quad \text{if } \tau \leq \tau_0. \end{cases} \tag{3.1}$$

Note that the cardinality of the random set n_w is precisely the stochastic term controlled in Lemma 3.1, i.e., $|n_w| = \sum_{i=1}^n \zeta_i(\tau)$. This relation serves to provide bounds on several other stochastic terms considered in subsequent lemmas. The relationship between the cardinality of the random index set n_w and the r.v.'s $\zeta_i(\tau)$, $i = 1, ..., n$ has also been used by Kaul et al. (2017) in the context of graphical models with missing data.

Lemma 3.2 *Let u_n be any non-negative sequence and let n_w be the random set of indices as defined in (3.1). Also, let $\mathcal{T}(\tau_0, u_n)$ be a u_n-neighborhood of τ_0 as defined in Lemma 3.1.*

Then under Condition B, we have for any fixed $\delta \in \mathbb{R}^p$ that,

$$(i) \quad \sup_{\tau \in \mathcal{T}(\tau_0, u_n)} \Big\|\frac{1}{n}\sum_{i \in n_w} \delta^T x_i x_i^T\Big\|_\infty \leq c_u c_{m1}\|\delta\|_2 \max\Big\{\frac{\log p}{n}, u_n\Big\},$$

$$(ii) \quad \sup_{\tau \in \mathcal{T}(\tau_0, u_n)} \frac{1}{n}\sum_{i \in n_w} \delta^T x_i x_i^T \delta \leq c_u c_{m1}\|\delta\|_2^2 \max\Big\{\frac{\log p}{n}, u_n\Big\},$$

$$(iii) \quad \sup_{\tau \in \mathcal{T}(\tau_0, u_n)} \frac{1}{n}\Big\|\sum_{i \in n_w} \varepsilon_i x_i^T\Big\|_\infty \leq c_u c_{m2}\sqrt{\frac{\log p}{n}} \max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\},$$

with probability at least $1 - c_1\exp(-c_2\log p)$. Here $c_u > 0$ is a universal constant, and $c_{m1} = (\phi + \sigma_x + \sigma_x^2)$, $c_{m2} = (\sqrt{\sigma_\varepsilon\sigma_x} + \sigma_\varepsilon\sigma_x)$ are model constants.

Finally in order to obtain the desired error bounds (2.8) and (2.9) we require restricted eigenvalue conditions on the gram matrix $\sum_{i=1}^n x_i x_i^T$. For any deterministic set $S \subset \{1, 2, ..., p\}$, define the collection $\mathbb{A}$ as,

$$\mathbb{A} = \Big\{\delta \in \mathbb{R}^p;\ \|\delta_{S^c}\|_1 \leq 3\|\delta_S\|_1,\Big\}. \tag{3.2}$$

Then, Bickel et al. (2009) define the lower restricted eigenvalue condition as,

$$\inf_{\delta \in \mathbb{A}} \frac{1}{n}\sum_{i=1}^n \delta^T x_i x_i^T \delta \geq c_u \kappa \|\delta\|_2^2, \quad \text{for some constant } \kappa > 0. \tag{3.3}$$

Other slightly weaker versions of this condition are also available in the literature such as the compatibility condition of Bühlmann and Van De Geer (2011), and the ℓ_q sensitivity of Gautier and Tsybakov (2011). In the setup of common random designs, it is also well established that condition (3.3) holds with probability converging to 1, see for e.g. Raskutti et al. (2010), and Rudelson and Zhou (2012), for Gaussian designs. In the subgaussian case, the plausibility of this condition is a consequence of a general result stated as Lemma B.2 in Appendix B. Under our high dimensional change point setup, we shall require versions of the restricted eigenvalue condition (3.3). In the following lemma, we shall show that all required conditions hold with probability converging to 1. Among other arguments, the proof of these conditions shall rely on Lemma B.2. In Lemma 3.3 below, the collection $\mathbb{A}$ in (3.2) applies for the set S in Condition A.

Lemma 3.3 (Restricted Eigenvalue Conditions): *Let u_n, and v_n be any non-negative sequences such that $v_n \geq c\log p/n$, $c > 0$. Let $\mathcal{T}(\tau_0, u_n)$ be as in Lemma 3.1 and the set $\mathbb{A}$ as defined in (3.2) for S given in Condition A. Furthermore, define the set $\mathbb{A}_2 = \Big\{\delta \in \mathbb{R}^p; \|\delta_{S^c}\|_1 \leq 3\|\delta_S\|_1 + 3\|\beta_0 - \gamma_0\|_1\Big\}$, and let any $\tau \in \mathbb{R}$ be such that $\Phi_{\min}^{-1}(\tau)s\log p/n = o(1)$. Then under Conditions A(i), A(ii), and B, and for n sufficiently large, the following re-*

stricted eigenvalue conditions hold with probability at least $1 - c_1 \exp(-c_2 \log p)$,

$$(i) \quad \inf_{\delta \in \mathbb{A}} \frac{1}{n} \sum_{i; w_i \le \tau} \delta^T x_i x_i^T \delta \ge c_u \kappa \Phi(\tau) \|\delta\|_2^2,$$

$$(ii) \quad \inf_{\delta \in \mathbb{A}} \frac{1}{n} \sum_{i; w_i > \tau} \delta^T x_i x_i^T \delta \ge c_u \kappa (1 - \Phi(\tau)) \|\delta\|_2^2,$$

$$(iii) \quad \sup_{\tau \in \mathcal{T}(\tau_0, u_n)} \sup_{\delta \in \mathbb{A}} \frac{1}{n} \sum_{i \in n_w} \delta^T x_i x_i^T \delta \le c_u c_m \|\delta\|_2^2 \max\Big\{\frac{s \log p}{n}, u_n\Big\},$$

$$(iv) \quad \inf_{\substack{\tau \in \mathbb{R}; \\ \Phi^*(\tau_0, \tau) \ge v_n}} \inf_{\delta \in \mathbb{A}_2} \frac{1}{n} \sum_{i \in n_w} \delta^T x_i x_i^T \delta \ge c_u c_m v_n \|\delta\|_2^2 - c_u c_m \frac{s \log p}{n} \Big(\|\delta\|_2^2 + \xi_n^2\Big).$$

Before moving on to state our main results in the next section, we make the following remark regarding the role of the set $\mathbb{A}_2$.

Remark 3.1 Note that if $\beta - \beta_0 \in \mathbb{A}$, i.e., $\|\beta_{S^c} - \beta_{0S^c}\|_1 \le 3\|\beta_S - \beta_{0S}\|_1$, then for $\delta = \beta_0 - \gamma_0 + \beta - \beta_0$, we have,

$$\|\delta_{S^c}\|_1 \le \|\beta_{S^c} - \beta_{0S^c}\|_1 \le 3\|\beta_S - \beta_{0S}\|_1 \le 3\|\delta_S\|_1 + 3\|\beta_0 - \gamma_0\|_1.$$

Thus $\beta - \beta_0 \in \mathbb{A}$, implies the vector $\delta \in \mathbb{A}_2$. This relation is useful in proving Lemma 4.1 and Theorem 4.2 of the next section.

4. Main Results

We are now ready to state our first main result pertaining to the rate of convergence of the regression estimates obtained from (2.6) and (2.7) when τ(possibly random) is in a u_n-neighborhood of τ_0.

Theorem 4.1 *Suppose Conditions A(i), A(ii), and B hold, and consider any* $\tau \in \mathbb{R}$, *satisfying* $\Phi_{\min}^{-1}(\tau) s \log p/n = o(1)$. *Let* $\hat{\beta}(\tau)$ *and* $\hat{\gamma}(\tau)$ *be solutions to (2.6) and (2.7). Then,*
(i) When $\Phi(\tau_0) = 0$ *and* $\lambda_1 \ge c_u c_m \sqrt{\log p/n}$, *for* n *sufficiently large, we have for* $q = 1, 2$,

$$\|\hat{\beta}(\tau) - \gamma_0\|_q \le c_u c_m \frac{1}{\Phi_{\min}(\tau)} s^{1/q} \sqrt{\frac{\log p}{n}},$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. *The same bound holds for* $\|\hat{\gamma}(\tau) - \gamma_0\|_q$, $q = 1, 2$.
(ii) When $\Phi_{\min}(\tau_0) > 0$, *let* u_n *be any non-negative sequence satisfying* $u_n = o\big(\Phi_{\min}(\tau_0)\big)$. *Also, let* $\mathcal{T}(\tau_0, u_n)$ *be as defined in Lemma 3.1 and* $\lambda_1 = c_u c_m \max\{\sqrt{\log p/n},\ \xi_n u_n\}$. *Then for* n *sufficiently large, and* $q = 1, 2$, *the following uniform bound holds,*

$$\sup_{\tau \in \mathcal{T}(\tau_0, u_n)} \|\hat{\beta}(\tau) - \beta_0\|_q \le c_u c_m \frac{1}{\Phi_{\min}(\tau_0)} s^{1/q} \max\Big\{\sqrt{\frac{\log p}{n}}, \|\beta_0 - \gamma_0\|_2 u_n\Big\},$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. *The same uniform upper bound also holds for* $\sup_{\tau \in \mathcal{T}(\tau_0, u_n)} \|\hat{\gamma}(\tau) - \gamma_0\|_q$, $q = 1, 2$.

Remark 4.1 As a direct consequence of Theorem 4.1, we obtain the rates of convergence of the regression estimates $\hat{\beta}^{(0)}, \hat{\gamma}^{(0)}$ from **Step 0** of **Algorithm 1**. Specifically, under the conditions of Theorem 4.1,
(i) When $\Phi_{\min}(\tau_0) = 0$,

$$\|\hat{\beta}^{(0)} - \gamma_0\|_q \le c_u c_m \frac{1}{\Phi_{\min}(\tau^{(0)})} s^{1/q} \sqrt{\frac{\log p}{n}}, \qquad q = 1, 2,$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. The same bound holds for $\|\hat{\gamma}^{(0)} - \gamma_0\|_q$, $q = 1, 2$.
(ii) When $\Phi_{\min}(\tau_0) > 0$, and $\Big|\Phi(\tau^{(0)}) - \Phi(\tau_0)\Big| \le u_n^{(0)}$, we have,

$$\|\hat{\beta}^{(0)} - \beta_0\|_q \le c_u c_m \frac{1}{\Phi_{\min}(\tau_0)} s^{1/q} \max\Big\{\sqrt{\frac{\log p}{n}},\ \xi_n u_n^{(0)}\Big\}, \qquad q = 1, 2, \tag{4.1}$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. The same bound holds for $\|\hat{\gamma}^{(0)} - \gamma_0\|_q$, $q = 1, 2$. In this case, since $s^{1/q} \xi_n u_n^{(0)} \big/ \Phi(\tau_0)$ may diverge, these estimates are not guaranteed to be consistent. Nevertheless, (i) and (ii) above play an important role in deriving convergence rates of estimators from subsequent steps of Algorithm 1.

We now turn our attention to establishing selection and estimation results for estimates obtained from **Step 1** and **Step 2** of Algorithm 1. To achieve this goal, we require the following notations. For any $\tau \in \mathbb{R}$, $\beta, \gamma \in \mathbb{R}^p$, let

$$\begin{aligned} R_n(\tau, \beta, \gamma) &= Q(\tau, \beta, \gamma) - Q(\tau_0, \beta, \gamma). \\ S_n(\tau, \beta, \gamma) &= R_n(\tau, \beta, \gamma) + \mu\big(\|\Phi(\tau)\|_0 - \|\Phi(\tau_0)\|_0\big) \end{aligned}$$

Also, for any non-negative u_n, and v_n, define the collection

$$\mathcal{H}(u_n, v_n) = \big\{\tau \in \mathbb{R};\ v_n \le |\Phi(\tau) - \Phi(\tau_0)| \le u_n\big\}$$

Additionally, for any non-negative sequence u_n, we also define the function,

$$F(u_n) = \begin{cases} 0 & \text{if } u_n/\Phi_{\min}(\tau_0) \to 0 \\ 1 & \text{otherwise} \end{cases}. \tag{4.2}$$

Finally, in the following, we denote by $r_n := \max\big\{\sqrt{s \log p/n}, \sqrt{s} \xi_n u_n^{(0)}\big\}/\Phi_{\min}(\tau_0)$, in the case where $\Phi_{\min}(\tau_0) > 0$. Notice that r_n is the ℓ_2 rate of estimation error provided in Part(ii) of Remark 4.1. The following lemma provides a uniform lower bound of the expression $S_n(\tau, \beta, \gamma)$, over the collection $\mathcal{H}(u_n, v_n)$, that holds with high probability. This result shall lie at the heart of the argument used to obtain the main results of this article.

Lemma 4.1 *Suppose conditions A and B hold and let* u_n *be any non negative sequence. Also, let* $\hat{\beta}^{(0)}$, $\hat{\gamma}^{(0)}$ *be estimates from* **Step 0** *of Algorithm 1. Then,*

(i) When $\Phi(\tau_0) = 0$, for any $v_n > 0$, we have

$$\inf_{\tau \in \mathcal{H}(1, v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \geq \mu - c_u c_m \frac{s \log p}{n \Phi^2_{\min}(\tau^{(0)})}$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$.

(ii) When $\Phi_{\min}(\tau_0) > 0$, for any $v_n \geq c \log p/n$, $c > 0$, we have,

$$\inf_{\tau \in \mathcal{H}(u_n, v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \geq \xi_n^2 \Big(c_u c_m v_n - c_u c_m \frac{s \log p}{n} - \frac{c_u c_m}{1 \vee \xi_n} \sqrt{\frac{s \log p}{n}} \max\Big\{ \sqrt{\frac{\log p}{n}}, \sqrt{u_n} \Big\}$$
$$- c_u c_m \frac{r_n^2}{1 \vee \xi_n^2} \max\Big\{ \frac{s \log p}{n}, u_n \Big\} - \frac{c_u \mu}{1 \vee \xi_n^2} F(u_n) \Big).$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$.

Our main result on rate of convergence of the estimates obtained from **Algorithm 1** is stated in Theorem 4.2 below. While the complete proof of the theorem is given in the appendix, here we provide a sketch of the main idea behind the proof. We show that, for an appropriately chosen regularizer μ, for any $v_n > 0$ (in the case where $\Phi(\tau_0) = 0$), or for any non-negative sequence v_n slower in rate than those given in (2.8) (in the case where $\Phi_{\min}(\tau_0) > 0$), we shall show that,

$$\inf_{\tau;\, v_n \leq \Phi^*(\tau_0, \tau) \leq 1} S_n\big(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}\big) > 0, \text{ for } n \text{ sufficiently large.}$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. Upon noting that the global optimizer $\hat{\tau}^{(1)}$ by definition satisfies $S_n(\hat{\tau}^{(1)}, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \leq 0$, we would have shown that the corresponding global optimizer $\hat{\tau}^{(1)}$ satisfies the relations given in (2.8). Along the way a sequence of recursions are required in order to sequentially sharpen the bound for the change point estimate. Supportive arguments are also required to show that the eventual bound is satisfied with probability at least $1 - c_1 \exp(-c_2 \log p)$. In this process, Remark .1 is quite helpful.

Theorem 4.2 *Suppose Conditions A and B hold and choose $\lambda_1 = c_u c_m \max\{\sqrt{\log p/n}, \xi_n u_n^{(0)}\}$, and $\mu = c_u c_m \big(s \log p / n l_n^2\big)^{1/k^*}$, where $k^* = \max\{k, 2\}$. Then for n sufficiently large, the optimizer $\hat{\tau}^{(1)}$ of* **Step 1** *of Algorithm 1 satisfies the following relations.*

(i) When $\Phi(\tau_0) = 0$, then $\Phi(\hat{\tau}^{(1)}) = 0$, with probability at least $1 - c_1 \exp(-c_2 \log p)$.

(ii) When $\Phi_{\min}(\tau_0) \geq c_u l_n$, then,

$$|\Phi(\hat{\tau}^{(1)}) - \Phi(\tau_0)| \leq t_n := c_u c_m \max\Big\{ \frac{s \log p}{n}, \frac{1}{(1 \vee \xi_n^2) l_n^2} \frac{s \log p}{n} \Big\},$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$.

The usefulness of Theorem 4.2 is apparent. Despite initializing Algorithm 1 with a $\tau^{(0)}$, which is in an $(s \log p / n l_n^2)^{1/k}$ neighborhood of τ_0, for an nearly arbitrary $k \in [1, \infty)$. (any initial value that posses 'fractional information' of τ_0), the updated change point $\hat{\tau}^{(1)}$ lies in a near optimal neighborhood of τ_0, irrespective of the value of k (irrespective of the precision of the initial guess). The following theorem provides the rates of convergence of the regression coefficient estimates $\hat{\beta}^{(1)}$ and $\hat{\gamma}^{(1)}$ obtained from **Step 2** of Algorithm 1.

Theorem 4.3 *Suppose the model (1.1) in the case where a finite change point exists, i.e., $\Phi_{\min}(\tau_0) > 0$. Assume the conditions of Theorem 4.2 and choose $\lambda_2 = c_u c_m \max\{\sqrt{\log p/n}, \xi_n t_n\}$, where t_n is as defined in Theorem 4.2. Then, the estimates $\hat{\beta}^{(1)}$ and $\hat{\gamma}^{(1)}$ of* **Step 2** *of Algorithm 1 satisfy,*

$$(i) \quad \|\hat{\beta}^{(1)} - \beta_0\|_q \leq c_u c_m s^{1/q} \frac{1}{\Phi_{\min}(\tau_0)} \max\Big\{ \sqrt{\frac{\log p}{n}},\ \xi_n t_n \Big\}, \quad q = 1, 2,$$

$$(ii) \quad \|\hat{\gamma}^{(1)} - \gamma_0\|_q \leq c_u c_m s^{1/q} \frac{1}{\Phi_{\min}(\tau_0)} \max\Big\{ \sqrt{\frac{\log p}{n}},\ \xi_n t_n \Big\}, \quad q = 1, 2,$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$.

Remark 4.2 (Interpretation of the rates of Theorem 4.2 and Theorem 4.3) Note that under conditions of Theorem 4.2, and additionally assuming that $\xi_n l_n \geq c_1 > 0$, we have,

$$|\Phi(\hat{\tau}^{(1)}) - \Phi(\tau_0)| \leq c_u c_m \frac{s \log p}{n},$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. This observation provides an intuitive statement, saying that an increasing jump size ξ_n can compensate for the location of the unknown change point moving toward the boundaries of $\mathbb{R}$, in effect allowing $\hat{\tau}^{(1)}$ of Algorithm 1 to approximate the unknown change point (if it exists) at a near optimal rate. In this case, under conditions of Theorem 4.3, the regression estimates of **Step 2** become,

$$\|\hat{\beta}^{(1)} - \beta_0\|_q \leq c_u c_m s^{1/q} \frac{1}{\Phi_{\min}(\tau_0)} \max\Big\{ \sqrt{\frac{\log p}{n}},\ \xi_n \frac{s \log p}{n} \Big\}, \quad q = 1, 2, \tag{4.3}$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. The same bound also holds for $\|\hat{\gamma}^{(1)} - \gamma_0\|_q$, $q = 1, 2$, with the same probability. The bound in the statement (4.3) again provides an interesting observation, where an increasing jump is leading to potentially counteract the precision of the regression estimate $\hat{\beta}^{(1)}$. First note that, the bound in (4.3) can tolerate an increasing jump size ξ_n such that $\xi_n s \sqrt{\log p/n} \to 0$, while preserving optimality of the rate of convergence, i.e, yielding, $\|\hat{\beta}^{(1)} - \beta_0\|_2 \leq c_u c_m \Phi^{-1}(\tau_0) \sqrt{s \log p/n}$, with high probability. It is only when ξ_n increases faster than $\sqrt{n/s^2 \log p}$ that it begins to harm the rate of convergence. This observation is surprising in that it suggests that an increasing jump size always benefits the change point estimate, whereas it benefits the regression coefficient estimates only when the jump size is increasing upto a certain rate. To the best of our knowledge, such a characterization of the effect of the jump size on parameter estimates, which holds only in the high dimensional case, has not been provided in the literature. The illustration in Figure 2 provides an intuitive understanding of this behavior,

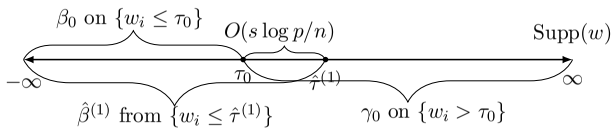


Figure 2: Illustration of counteracting effect of jump size ξ_n on regression coefficient estimates

From Figure 2, observe that for any finite jump size, the best approximation that our analysis can provide is wherein the error is of order $s \log p/n$.[g] Now the regression estimates of **Step 2** are computed based on the binary partition yielded by the change point estimate $\hat{\tau}^{(1)}$ of **Step 1**. Consequently, the data based on which the regression estimate $\hat{\beta}^{(1)}$ of β_0 is obtained, may be corrupted by as much as a fraction $O(s \log p/n)$ of observations where the true regression coefficient is γ_0. Thereby, the higher the jump size ξ_n, the more impact this small corruption will have on the estimate $\hat{\beta}^{(1)}$. The same argument also holds for the other binary partition. This provides an explanation of the rates observed in Theorem 4.3.

5. Implementation and Numerical Results

The three main objectives of this empirical study are, (i) to evaluate the overall performance of Algorithm 1, i.e., its ability to consistently estimate β_0, γ_0, and a finite τ_0, and compare its performance to a full grid search approach, (ii) to numerically support the theoretically claimed statement, that the estimate $\hat{\tau}^{(1)}$ is insensitive to the quality of the initial guess $\tau^{(0)}$, and (iii) to evaluate the numerical performance of Algorithm 1 in detecting the 'no change' case, i.e., when $\Phi(\tau_0) = 0$.

5.1. Simulation setup

We consider the data generating process (1.1) where ε_i, w_i and x_i are drawn independently satisfying $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, $w_i \sim \mathcal{U}(0,1)$,[h] and $x_i \sim \mathcal{N}(0, \Sigma)$. Here, Σ is a $p \times p$ matrix with elements $\Sigma_{ij} = \rho^{|i-j|}$, $i, j = 1, ..., p$. We set, $\sigma_\varepsilon = 1$ and $\rho = 0.5$. The regression parameters of the model are set to be $\beta_0 = (1, 1, 1, 1, ..., 0)^T_{p\times 1}$, and $\gamma_0 = (0_{1\times 4}, 1, 1, 1, 1, 0, ..., 0)^T_{p\times 1}$. The metrics of interest are bias and mean squared error of various estimates: For numerical experiments where $\Phi_{\min}(\tau_0) > 0$, $\text{bias}(\hat{\beta}) = \|\text{E}(\hat{\beta} - \beta_0)\|_2$, $\text{bias}(\hat{\gamma}) = \|\text{E}(\hat{\gamma} - \gamma_0)\|_2$, $\text{bias}(\hat{\tau}) = |\text{E}(\hat{\tau} - \tau_0)|$, $\text{mse}(\hat{\beta}) = \|\text{E}(\hat{\beta} - \beta_0)^2\|_2$, $\text{mse}(\hat{\gamma}) = \|\text{E}(\hat{\gamma} - \gamma_0)^2\|_2$, $\text{mse}(\hat{\tau}) = \text{E}(\hat{\tau} - \tau_0)^2$, $\text{mse}(\Phi(\hat{\tau})) = \text{E}(\Phi(\hat{\tau}) - \Phi(\tau_0))^2$. For numerical experiments where $\Phi(\tau_0) = 0$, we report $PrM = E(\mathbf{1}[\hat{\tau}^{(1)} = -\infty])$, i.e., the proportion of times where the 'no change' model is correctly identified. We shall report monte carlo approximations of these metrics based on 100 replications for each combination of model parameters. In the simulations where a finite change point, i.e., $0 < \tau_0 < 1$ is misidentified as 'no change point', i.e., $\hat{\tau}^{(1)} = -\infty$, (observed to occur sometimes when τ_0 is near the boundaries of $(0, 1)$), we do the following operation to maintain fairness of comparisons of the above metrics. In case where $\tau_0 < 0.5$ and $\hat{\tau}^{(1)} = -\infty$, then we set $\hat{\tau}^{(1)} = 0$, $\hat{\beta}^{(1)} = 0_{p\times 1}$, and when $\tau_0 > 0.5$ and $\hat{\tau}^{(1)} = -\infty$, we set $\hat{\tau}^{(1)} = 1$, $\hat{\gamma}^{(1)} = 0_{p\times 1}$. Finally, we also report the metric *time*: the average (over replications) computation time [i]. All computations are performed in the software R, R Core Team (2017). All lasso optimizations are performed with the R package 'glmnet', developed by Friedman et al. (2010). We perform two sets of simulations for all combinations of the parameters $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$. In the first simulation, we consider finite change points, with $\tau_0 \in \{0.169, 0.264, ..., 0.831\}$, (Equally spaced grid of 8 points between 0.169 to 0.831). This is referred to as **Simulation A** in the following. The second simulation considers the case of 'no change' in the model (1.1), i.e., $\tau_0 = 0$. This simulation is referred

g. This rate matches the fastest available in the literature
h. Since $w_i \sim \mathcal{U}(0,1)$, hence $\Phi(\tau) = \tau$, $\tau \in (0,1)$.
i. CPU: Intel Xeon E5-2609 v3 @ 1.9 GHz, RAM: 128 GB

to as **Simulation B** in the following. Due to the absence of any comparative method that is able to detect the 'no change' case (to the best of our knowledge), we report only the results of our method for this simulation. Note that for each fixed p, the total number of model parameters to be estimated is $2p + 1$.

Choice of tuning parameters: The regularizer λ_1 and λ_2 of the Lasso optimizations of **Step 0** and **Step 2** of Algorithm 1 are chosen via a 5-fold cross validation, which is performed internally by the R package 'glmnet'. The regularizer μ of **Step 1** of Algorithm 1 is chosen via the classical BIC criteria. Specifically, $\hat{\tau}(\mu)$ is computed over a grid of values of μ. Then, the value of μ of that minimizes the criteria,

$$\text{BIC}(\mu) = \log\Big(Q\big(\hat{\tau}(\mu), \hat{\beta}(\mu), \hat{\gamma}(\mu)\big)\Big) + \frac{\log n}{n}\|\Phi\big(\hat{\tau}(\mu)\big)\|_0,$$

is chosen. Here $Q(\cdot, \cdot, \cdot)$ is the least squares loss, as defined in (2.1), and $\hat{\beta}(\mu)$, $\hat{\gamma}(\mu)$ represent regression coefficient estimates obtained on the binary partition given by $\hat{\tau}(\mu)$.

In the following, we consider two schemes to choose the initializer $\tau^{(0)}$ of **Algorithm 1**. The first is to set to $w^{(0.5)}$, i.e., the 0.5^{th} empirical quantile of $w = (w_1, .., w_n)^T$. This is done to make the initializer equidistant from the two extremes of the support of w. Note that, in the absence of any information on the unknown τ_0, the choice $\tau^{(0)} = w^{(0.5)}$ is a sensible choice for the initializer. This approach is represented as '**Algorithm 1A**'. As a second scheme, we choose the initializer $\tau^{(0)}$ by setting it to one of values $\{w^{(m)};\ m = 0.25, 0.50, 0.75\}$, where $w^{(m)}$ represents the mth empirical quantile of $w = (w_1, ..., w_n)^T$. This is done by first computing $\hat{\beta}(\tau)$ and $\hat{\gamma}(\tau)$ in (2.6) and (2.7) for each $\tau = w^{(0.25)}, w^{(0.50)}, w^{(0.75)}$, and finally selecting $\tau^{(0)}$ as the value that minimizes the least squares loss over these three choices. Note that the latter approach has an additional computational burden of two Lasso(n, p) optimizations in comparison to the former. This approach is represented as '**Algorithm 1B**'. Clearly, the initializer in Algorithm 1B will be a closer value to the unknown τ_0 in comparison to the initializer of Algorithm 1A. This shall also help us numerically support our theoretical finding that Algorithm 1 is insensitive to the 'quality' of the initializer. Finally, we also implement the full grid search approach of Lee et al. (2016) in order to serve as a benchmark to compare the performance of the proposed estimates and also to illustrate the dramatic gains in computation time provided by our method. This approach is referred to as **Full grid search** in the following. For completeness, the **Full grid search** estimator of Lee et al. (2016) is described in the notation of this article in the following. The article of Lee et al. (2016) assumes the model $y_i = x_i\delta_0 + x_i^T\eta_0\mathbf{1}[w_i \le \tau_0]$, which is equivalent to the model (1.1) when $\delta_0 = \gamma_0$ and $\eta_0 = \beta_0 - \gamma_0$. Now, let $\tilde{x}_i(\tau) = \big(x_i^T, x_i^T\mathbf{1}(w_i \le \tau)\big)^T_{2p\times 1}$, and the parameter $\alpha = (\gamma, \beta - \gamma)$, where β_0, γ_0 are the true parameter coefficients of the model (1.1), then

$$\begin{aligned}
\hat{\alpha}(\tau) &= \underset{\alpha\in\mathbb{R}^{2p}}{\arg\min}\Big\{\frac{1}{n}\sum_{i=1}^n (y_i - \tilde{x}_i^T(\tau)\alpha)^2 + \lambda\|D(\tau)\alpha\|_1\Big\}, \quad \text{for each } \tau \in \mathcal{T}^*, \\
\hat{\tau} &= \underset{\tau\in\mathcal{T}^*}{\arg\min}\Big\{\frac{1}{n}\sum_{i=1}^n \big(y_i - \tilde{x}_i^T(\tau)\hat{\alpha}(\tau)\big)^2 + \lambda\|D(\tau)\hat{\alpha}(\tau)\|_1\Big\} \qquad (5.1)
\end{aligned}$$

Table 1: Numerical results of **Algorithm 1A and 1B**, and **Full grid search** for $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = 0.169$.

Method	n	p	bias($\hat{\beta}$)	bias($\hat{\gamma}$)	bias($\hat{\tau}$)	mse($\hat{\beta}$)	mse($\hat{\gamma}$)	mse($\hat{\tau}$)	time
Algorithm 1A	150	25	0.6595	0.3026	0.0031	0.4564	0.0858	0.0003	0.5792
	150	150	1.5638	0.3530	0.0067	1.4932	0.1044	0.0006	0.8684
	150	250	1.4271	0.3965	0.0193	1.2918	0.1336	0.0021	0.9606
	250	25	0.4354	0.2444	0.0006	0.2135	0.0498	0.0001	0.6068
	250	150	0.5694	0.2717	0.0045	0.2877	0.0599	0.0001	1.3090
	250	250	0.6600	0.2958	0.0015	0.3836	0.0650	0.0001	1.3468
	350	25	0.4193	0.2188	0.0068	0.1758	0.0369	0.0001	0.6515
	350	150	0.5285	0.2362	0.0023	0.2325	0.0415	0.0000	1.5462
	350	250	0.8964	0.2504	0.0028	0.6499	0.0449	0.0001	2.6849
Algorithm 1B	150	25	0.6758	0.2918	0.0030	0.4724	0.0827	0.0002	0.9387
	150	150	1.5803	0.3880	0.0063	1.5061	0.1372	0.0111	1.3351
	150	250	1.4343	0.4167	0.0002	1.3142	0.1648	0.0136	1.4999
	250	25	0.4040	0.2370	0.0020	0.1964	0.0481	0.0001	0.9684
	250	150	0.5666	0.2645	0.0036	0.2779	0.0564	0.0001	2.2686
	250	250	0.6779	0.2961	0.0023	0.4021	0.0648	0.0001	2.1075
	350	25	0.4034	0.2154	0.0081	0.1661	0.0358	0.0001	1.0306
	350	150	0.5209	0.2293	0.0034	0.2228	0.0401	0.0001	2.4129
	350	250	0.8761	0.2459	0.0041	0.6393	0.0442	0.0001	3.5468
Full grid search	150	25	0.7450	0.2240	0.0193	0.5763	0.0612	0.0007	12.7566
	150	150	2.0766	0.4225	0.0531	1.9157	0.1313	0.0068	25.8863
	150	250	2.0300	0.4732	0.0287	1.8209	0.1650	0.0057	31.5806
	250	25	0.5764	0.1750	0.0098	0.3359	0.0341	0.0002	21.0164
	250	150	1.1066	0.2894	0.0023	0.7863	0.0640	0.0002	59.7864
	250	250	1.2875	0.3318	0.0127	0.9927	0.0758	0.0012	88.9717
	350	25	0.5036	0.1588	0.0005	0.2270	0.0238	0.0001	29.4869
	350	150	1.0025	0.2201	0.0057	0.6845	0.0373	0.0006	113.4715
	350	250	1.6075	0.2627	0.0063	1.3631	0.0477	0.0002	144.6366

where $D(\tau) = \text{diag}\{\|\tilde{x}^{(j)}(\tau)\|_n,\ j = 1, ..., 2p\}$, with $\tilde{x}^{(j)}(\tau)$ representing the jth column of the design matrix $\tilde{x}(\tau) = \left(\tilde{x}_1(\tau), ..., \tilde{x}_n(\tau)\right)^T_{n\times 2p}$. In implementation of this estimator, the search space of the change point is restricted to $\tau \in \mathcal{T}^* = \{w_1, ..., w_n\} \cap (0.1, 0.9)$.

5.2. Results and discussion

The bias and mean squared error (mse) of estimates obtained from **Algorithms 1A, 1B**, and **Full grid search** for **Simulation A**, for all combinations of $n \in \{150, 200, 250\}$, $p \in \{25, 150, 250\}$, and $\Phi(\tau_0) \in \{0.169, 0.67\}$ are presented in Table 1 and Table 2. All results provided are truncated at 10^{-4}. Complete results of the simulation study including all cases for $\tau_0 \in \{0.169, 0.264, ..., 0.831\}$, are available in the supplementary materials of this article. To aid in interpretation, the results on bias are illustrated through Figures 3, 4, 5, 6 and 7. In particular, Figure 3 illustrates bias associated with the change point estimate, Figure 4 and Figure 5 illustrate the bias in $\hat{\beta}^{(1)}$, and $\hat{\gamma}^{(1)}$ respectively. In Figure 6 we illustrate the consistency of the proposed methodology and finally, in Figure 7 we depict the average computation time for the methods implemented in this simulation study. The results of **Simulation B** are reported in Table 3. This table reports the proportion of times the 'no change' model is correctly identified via the metric PrM as described above.

Simulation A: Two important observations from the bias results for the change point estimate depicted in Figures 3 are, first, the proposed **Algorithms 1A and 1B** are indis-

Table 2: Numerical results of **Algorithm 1A and 1B**, and **Full grid search** for $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = 0.642$.

Method	n	p	bias($\hat{\beta}$)	bias($\hat{\gamma}$)	bias($\hat{\tau}$)	mse($\hat{\beta}$)	mse($\hat{\gamma}$)	mse($\hat{\tau}$)	time
Algorithm 1A	150	25	0.3170	0.4765	0.0065	0.1041	0.2308	0.0003	0.5855
	150	150	0.3900	0.5719	0.0039	0.1309	0.2819	0.0002	0.7854
	150	250	0.4191	0.5668	0.0046	0.1510	0.2756	0.0004	0.8583
	250	25	0.2470	0.3375	0.0088	0.0548	0.1114	0.0002	0.6177
	250	150	0.2926	0.4248	0.0007	0.0658	0.1486	0.0001	1.1309
	250	250	0.3142	0.4436	0.0039	0.0770	0.1614	0.0001	1.3242
	350	25	0.2277	0.3098	0.0041	0.0429	0.0858	0.0001	0.6748
	350	150	0.2501	0.3573	0.0018	0.0497	0.1012	0.0000	1.6129
	350	250	0.2861	0.3861	0.0018	0.0627	0.1135	0.0001	1.8991
Algorithm 1B	150	25	0.3148	0.4833	0.0064	0.1059	0.2364	0.0002	0.9449
	150	150	0.3826	0.5667	0.0060	0.1275	0.2737	0.0002	1.3024
	150	250	0.4138	0.5617	0.0039	0.1489	0.2724	0.0004	1.4660
	250	25	0.2512	0.3312	0.0121	0.0558	0.1114	0.0003	0.9661
	250	150	0.3029	0.4213	0.0005	0.0683	0.1470	0.0001	2.3502
	250	250	0.3147	0.4274	0.0056	0.0773	0.1490	0.0001	2.2163
	350	25	0.2318	0.3071	0.0043	0.0436	0.0847	0.0001	1.0086
	350	150	0.2525	0.3472	0.0016	0.0502	0.0958	0.0000	2.5538
	350	250	0.2815	0.3766	0.0015	0.0617	0.1087	0.0000	3.4451
Full grid search	150	25	0.2506	0.5229	0.0012	0.0983	0.2482	0.0002	12.9812
	150	150	0.5061	0.9313	0.0012	0.2037	0.5588	0.0009	31.2470
	150	250	0.6217	1.0329	0.0065	0.2572	0.6412	0.0008	35.3190
	250	25	0.2000	0.4208	0.0005	0.0505	0.1477	0.0003	21.5832
	250	150	0.3397	0.7712	0.0116	0.0909	0.3870	0.0003	67.6660
	250	250	0.4072	0.8228	0.0005	0.1229	0.4251	0.0001	105.1549
	350	25	0.1674	0.4083	0.0045	0.0353	0.1284	0.0001	30.2175
	350	150	0.2769	0.5949	0.0032	0.0640	0.2360	0.0001	126.2810
	350	250	0.3299	0.6896	0.0030	0.0841	0.3031	0.0001	200.2775

Table 3: Numerical results of **Simulation B**, where the underlying model is $y_i = x_i^T\gamma_0 + \varepsilon_i$, i.e., the 'no change' case where $\Phi(\tau_0) = 0$. The metric PrM is reported for each combination of n, p.

n	Algorithm 1A			Algorithm 1B		
	$p = 25$	$p = 150$	$p = 250$	$p = 25$	$p = 150$	$p = 250$
150	0.76	0.87	0.88	0.76	0.87	0.88
250	0.80	0.93	0.91	0.80	0.93	0.91
350	0.85	0.93	0.94	0.85	0.93	0.94

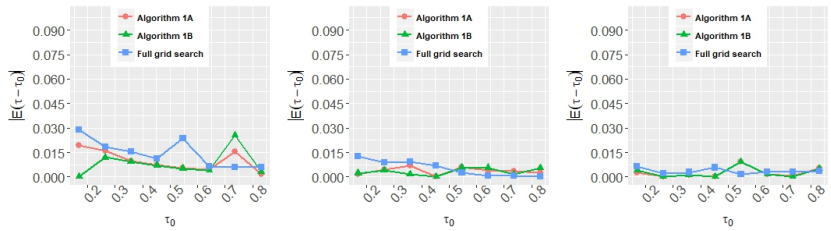


Figure 3: Comparison of bias($\hat{\tau}$) for **Algorithm 1A and 1B** and **Full grid search** across values of τ_0 for $p = 250$. Left panel: $n = 150$, Center panel: $n = 250$, Right panel: $n = 350$.

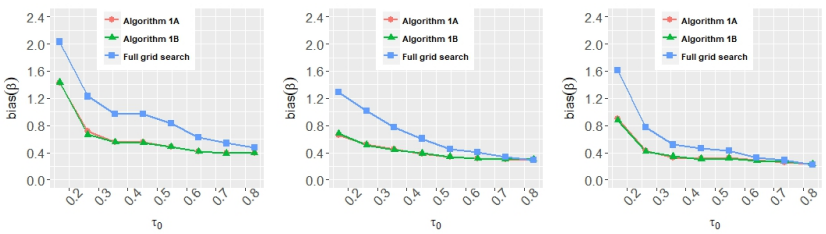


Figure 4: Comparison of bias($\hat{\beta}$) for **Algorithm 1A and 1B**, and **Full grid search** across values of τ_0 for $p = 250$. Left panel: $n = 150$, Center panel: $n = 250$, Right panel: $n = 350$.

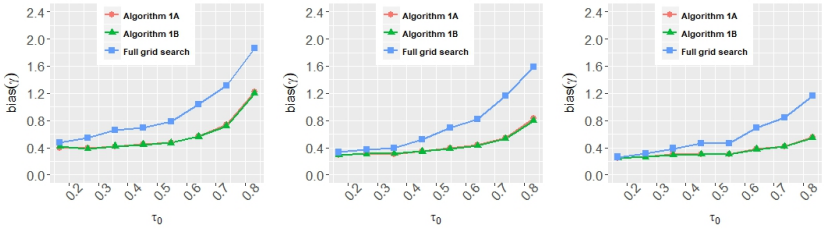


Figure 5: Comparison of bias($\hat{\gamma}$) for **Algorithm 1A and 1B**, and **Full grid search** across values of τ_0 for $p = 250$. Left panel: $n = 150$, Center panel: $n = 250$, Right panel: $n = 350$.

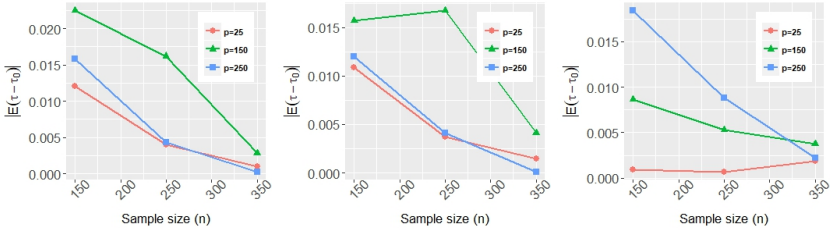


Figure 6: Illustration of consistency of implemented methods with $\tau_0 = 0.264$. Left panel: **Algorithm 1A**, Center panel: **Algorithm 1B**, and Right panel: **Full grid search**

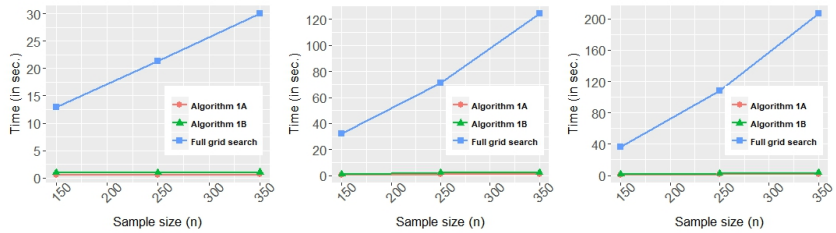


Figure 7: Comparison of computation time (in seconds) for **Algorithm 1A and 1B**, and **Full grid search** across values of n for $\tau_0 = 0.547$ Left panel: $p = 25$, Center panel: $p = 150$, Right panel: $p = 250$. Note that, these times are computed as averages over 100 replications of each method running in parallel over 12 cores. Running a single instance of any method is two to three times faster. Reported computation times include time taken to choose tuning parameters.

tinguishable from the **Full grid search** approach of Lee et al. (2016). Second, it is also observed that the proposed **Algorithms 1A and 1B** are indistinguishable from each other in terms of the bias in the change point estimate. Recall that, Algorithm 1B was designed in a way so that the starting value is always closer to τ_0 in comparison to **Algorithm 1A**. Despite a better initial value, no uniform improvement is observed in Algorithm 1B. This supports our theoretical result, that the quality of the initial value does not impact Algorithm 1, and it yields near optimal estimates with any initializing value containing any fractional information on the unknown change point. The bias results for the regression coefficient estimates depicted in Figures 4 and Figure 5 suggest that the proposed methodology yields a uniformly lower bias at all considered cases of τ_0. One possible reason for this behavior is that the design variable in our methodology are constructed as $z_i = (z_i^{(1)T}, z_i^{(2)T})^T$ where $z_i^{(1)} = x_i \mathbf{1}[w_i \leq \tau_0]$, and $z_i^{(2)} = x_i \mathbf{1}[w_i \leq \tau_0]$, which are orthogonal to each other, in contrast, the design variables in the methodology of Lee et al. (2016), the design variables are constructed as $z_i^{(1)} = x_i$ and $z_i^{(2)} = \mathbf{1}[w_i \leq \tau_0]$, which may be highly correlated. It is also clear from Figure 6 that the in bias in change point estimates from **Algorithm 1A and 1B** and **Full grid search** progressively shrinks with increasing values of n, thereby illustrating the consistency of the proposed methodology. Finally, in Figure 7 we illustrate the dramatic differences in the overall computation times in the implementation of the compared approaches. In the largest considered data set, the average time for computation of **Algorithm 1A and 1B** was $\approx$ 3seconds, as opposed to the full grid search which required $\approx$ 200seconds to implement. Note that the reported computation times include the time taken for choosing all required tuning parameters for each method.

Simulation B: The results of **Simulation B** reported in Table 3 are in accordance with expectations. The proposed methods are able to detect the 'no change' scenario with $\approx$ 85% accuracy in all considered cases. Selection consistency is also observed, i.e. the proportion of correct identifications is seen to increase with n. Finally, both **Algorithm**

1A and 1B are seen to provide the exact same results, which is again not surprising since the only difference in these two methods is the choice of the initial value.

6. Application

In this section, we apply our proposed methodology to the 'Communities and Crime' data set of Redmond and Baveja (2002), available publicly at `https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`. This data contains: (i) socio-economic data at a community level from across the entire United states, and is collected from the 1990 US Census, (ii) law enforcement data from the 1990 US Law Enforcement Management and Administrative Statistics survey, and (iii) crime data from the 1995 FBI Uniform Crime. The full data set contains 1994 observations and 128 variables. The dependent variable of interest is the total number of violent crimes per one hundred thousand population, which is calculated using the population and the sum of the crime variables that are considered violent crimes: murder, rape, robbery, and assault. The remaining variables are quantitative measurements on socio-economic variables such as the median (community level) income per household, percentage of people aged 16 and over who are employed, percentage of households with public assistance, percent of population who have immigrated within the last 10 years, amongst many others. This data was recently analyzed by Leonardi and Bühlmann (2016) for detecting and identifying change points in covariates when the change point(s) are modeled over locations. In this study we are interested in identifying changes in covariates when the change occurs through a change inducing variable. Specifically, we consider tow cases, (1) when the change inducing variable is assumed to be the population for the community, and (2) when the change inducing variable is assumed to be the median household income for the community, in an effort to investigate whether violent crime at a community level is influenced by distinct socio-economic factors below and above a certain threshold of population or the median household income, and also to estimate the threshold level at which such a transition occurs.

The full data set consists of $n = 1994$ observations and $p = 128$ variables, which have been normalized to $[0, 1]$ scale. The normalization process is described in the webpage whose link has been provided at the beginning of this section. This normalized data is pre-processed by deleting observations with any missing values, and by eliminating predictors that are highly correlated with other predictor variables. After the pre-processing, we obtain a filtered data set with $n = 319$ communities. The remaining data is then mean centered and scaled columnwise in order to remove the need for an intercept term in the regression, mainly to be consistent with model (1.1). Finally, predictor variables having a significant correlation with the change inducing variable have also been dropped from the analysis. This process yields a refined data set with $p = 75$ predictor variables (excluding the change inducing variable) in the case where the change inducing variable is 'population' and $p = 77$ in the case where the change inducing variable is 'median income'

We apply the proposed Algorithm 1 to the data under consideration with the initializer chosen as the 50th percentile of the change inducing variable, i.e, Algorithm 1A described in Section 5. The regularizer's λ_1, λ_2 and μ are chosen via cross validation and the classical BIC criteria respectively, as described in Section 5. Table 4 summarizes estimation and variable selection results for the regression coefficients of the assumed model (1.1), in the

Table 4: Summary of analysis of 'Communities and Crime' data set. Change inducing variable (w): population, model size: $n = 319$, $p = 75$. Estimated change point is $\hat{\tau}^{(1)} = 0.24$, which is the 73rd percentile of the population variable. The table lists all estimated non-zero regression coefficients truncated at 10^{-4}.

Variable	**Description**	**Coefficient ($\hat{\beta}_{\hat{S}}$) (pre change)**	**Coefficient ($\hat{\gamma}_{\hat{S}}$) (post change)**
racepctblack	% of population that is african american	0.0322	0.0000
racePctWhite	% of population that is caucasian	-0.1844	0.0000
pctWWage	% of households with wage or salary income in 1989	-0.0505	0.0000
pctWInvInc	% of households with investment / rent income in 1989	-0.0909	0.0000
PctPopUnderPov	% of people under the poverty level	0.0686	0.0000
PctEmploy	% of people 16 and over who are employed	-0.0069	0.0000
PctIlleg	% of kids born to never married	0.3105	0.1734
PctHousLess3BR	% of housing units with less than 3 bedrooms	0.0199	0.0000
PctHousOccup	% of housing occupied	-0.0359	0.0000
PctVacantBoarded	% of vacant housing that is boarded up	0.0000	0.0743
NumStreet	# of homeless people counted in the street	0.0000	0.1597
LemasSwFTFieldOps	# of sworn full time police officers in field operations	0.0000	-0.0229
PolicReqPerOffic	total requests for police per police officer (0/1)	0.0229	0.0000
LemasGangUnitDeploy	gang unit deployed	0.0196	0.0000

Table 5: Summary of analysis of 'Communities and Crime' data set. Change inducing variable (w): median income, model size: $n = 319$, $p = 77$. Estimated change point is $\hat{\tau}^{(1)} = -\infty$, i.e., no change detected in the model w.r.t. w. The table lists all estimated non-zero regression coefficients truncated at 10^{-4}.

Variable	Description	Coefficient ($\hat{\gamma}_{\hat{S}}$)
racepctblack	% of population that is african american	0.0207
racePctWhite	% of population that is caucasian	-0.1863
pctWWage	% of households with wage or salary income in 1989	-0.0245
pctWInvInc	% of households with investment / rent income in 1989	-0.1115
PctIlleg	% of kids born to never married	0.3010
PctHousLess3BR	% of housing units with less than 3 bedrooms	0.0230
HousVacant	# of vacant households	0.0045
PctHousOccup	% of housing occupied	-0.0352
PctVacantBoarded	% of vacant housing that is boarded up	0.0132
NumStreet	# of homeless people counted in the street	0.0919
PolicCars	# of police cars	0.0178

case where the change inducing variable is 'population' and Table 5 summarizes the results of the case where the change inducing variable is the 'median income'. In the first case with the change inducing variable as 'population', we find a change point estimate $\hat{\tau}^{(1)} = 0.23$ which is the 73rd percentile of the population variable. A noteworthy observation in this case about the estimated pre and post coefficients $\hat{\beta}^{(1)}$, $\hat{\gamma}^{(1)}$ from Table 4 is the near disjoint nature of the features influencing violent crime across the threshold $\hat{\tau}^{(1)}$ of the population variable. In the second case, where the change inducing variable is 'median income' the proposed method detects 'no change' in the model, i.e., yields a ordinary linear regression model for this case.

Acknowledgement

We thank three anonymous referee's for insighfut comments and suggestions that led to a significant improvement to the results and presentation of the manuscript.

Appendix A: Proofs

Proof of Lemma 3.1: Let $\tau_1 > \tau_0$ be a boundary point on the right of τ_0, such that $\Phi^*(\tau_0, \tau_1) = u_n$. Then recall that

$$\zeta_i(\tau_1) = \mathbf{1}[\tau_0 < w_i \leq \tau_1], \qquad \Phi^*(\tau_0, \tau) = \Phi(\tau_1) - \Phi(\tau_0).$$

Also, note that $p_n := E\zeta_i(\tau_1) = \Phi^*(\tau_0, \tau_1)$. Since ζ_i, $i = 1, ..., n$ are Bernoulli r.v.'s, for any $s > 0$, the moment generating function is given by $E(\exp(s\zeta_i)) = q_n + p_n \exp(s)$, where $q_n = 1 - p_n$. Applying the Chernoff Inequality, we obtain,

$$P\big(\sum_{i=1}^n \zeta_i(\tau_1) > t + np_n\big) = P\big(e^{\sum_{i=1}^n s\zeta_i(\tau_1)} > e^{(st+snp_n)}\big) \leq e^{-s(t+np_n)}[q_n + p_n e^s]^n.$$

Now in order to show,

$$P\Big(\frac{1}{n}\sum_{i=1}^n \zeta_i(\tau_1) \leq c_u \max\Big\{\frac{\log p}{n}, u_n\Big\}\Big) \quad \geq \quad 1 - c_1 \exp(-c_2 \log p). \tag{A.1}$$

We divide the argument into two cases. First, for any arbitrary constant $c_u > 0$, we let $\Phi^*(\tau_0, \tau_1) \geq c_u \log p/n$, upon choosing $t = n\Phi^*(\tau_0, \tau_1)$ we obtain,

$$P\big(\sum_{i=1}^n \zeta_i(\tau_1) > 2n\Phi^*(\tau_0, \tau_1)\big) \leq e^{[-2sn\Phi^*(\tau_0,\tau_1)]}[1 + (\Phi^*(\tau_0, \tau_1))(e^s - 1)]^n.$$

Using the deterministic inequality $(1 + x)^k \leq \exp(kx)$, for any $k, x > 0$, we obtain that

$$P\big(\sum_{i=1}^n \zeta_i(\tau_1) > 2n\Phi^*(\tau_0, \tau_1)\big) \leq e^{-2sn\Phi^*(\tau_0,\tau_1)} e^{(e^s-1)n\Phi^*(\tau_0,\tau_1)} \leq e^{-c_2 \log p}.$$

The inequality to the right follows by choosing $s = \log 2$, which maximizes the function $f(s) = 2s - e^s + 1$ and provides a positive value at the maximum, and by using the restriction $\Phi^*(\tau_0, \tau) \geq c_u \log p/n$. Next we let $\Phi^*(\tau_0, \tau_1) < c_u \log p/n$. Here choose $t = c_u \log p$ to obtain,

$$P\big(\sum_{i=1}^n \zeta_i(\tau_1) > c_u \log p + n\Phi^*(\tau_0, \tau_1)\big) \leq e^{[-sc_u \log p - sn\Phi^*(\tau_0,\tau_1)]}[1 + (\Phi^*(\tau_0, \tau_1))(e^s - 1)]^n. \tag{A.2}$$

Calling upon the inequality $(1 + x)^k \leq \exp(kx)$, for any $k, x > 0$, we can bound the RHS of (A.2) from above by $\exp\big[-sc_u \log p + (e^s - s - 1)\log p\big]$. Now $s = \log(1 + c_u)$ provides a positive value at the maximum, since it maximizes $f(s) = (1 + c_u)s - e^s + 1$. Then for any $c_u > 0$, we obtain,

$$P\big(\sum_{i=1}^n \zeta_i(\tau_1) > c_u \log p + n\Phi^*(\tau_0, \tau_1)\big) \quad \leq \quad e^{-c_2 \log p}.$$

Upon combining both cases, (A.1) follows by noting $\Phi^\star(\tau_0, \tau_1) = u_n$.

Now repeating the same argument for a fixed boundary point τ_2 on the left of τ_0, such that $\Phi(\tau_0) - \Phi(\tau_2) = u_n$, and applying a union bound we obtain,

$$P\Big(\max_{\tau\in\{\tau_1,\tau_2\}} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau) \leq c_u \max\Big\{\frac{\log p}{n}, u_n\Big\}\Big) \geq 1 - c_1 \exp(-c_2 \log p). \tag{A.3}$$

It remains to show that (A.1) holds uniformly over $\mathcal{T}(\tau_0, u_n)$. For this, we begin by noting that for any $\tau \in \mathcal{T}(\tau_0, u_n)$, where $\tau > \tau_0$ we have $\zeta_i(\tau) = \mathbf{1}\big[w_i \in (\tau_0, \tau]\big] \leq \mathbf{1}\big[w_i \in (\tau_0, \tau_1]\big]$. Similarly for any $\tau \in \mathcal{T}(\tau_0, u_n)$ where $\tau < \tau_0$ we have $\zeta_i(\tau) \leq \mathbf{1}\big[w_i \in [\tau_2, \tau_0)\big]$. Thus

$$\sup_{\tau\in\mathcal{T}(\tau_0,u_n)} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau) \leq \max_{\tau\in\{\tau_1,\tau_2\}} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau). \tag{A.4}$$

Part (i) of this lemma follows by combining (A.4) with the bound in (A.3).

To prove Part (ii) we use a lower bound for sums of non-negative r.v.s' stated in Lemma B.3. This result was originally proved by Maurer (2003). For a fixed right boundary point $\tau_1 > \tau_0$ such that $\Phi(\tau_1) - \Phi(\tau_0) = v_n$, set $t = v_n/2$ in Lemma B.3. Then we have

$$P\Big(\frac{1}{n}\sum_{i=1}^n \zeta_i(\tau_1) \leq \frac{v_n}{2}\Big) \leq \exp\Big(-nv_n\Big) \leq c_1 \exp(-c_2 \log p),$$

where the last inequality follows from $v_n \geq c_u \log p/n$. We obtain the same bound applying a similar argument for the left boundary point $\tau_2 < \tau_0$ such that $\Phi(\tau_0) - \Phi(\tau_2) = v_n$. Now applying an elementary union bound we obtain

$$P\Big(\min_{\tau\in\{\tau_1,\tau_2\}} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau) \geq c_u v_n\Big) \geq 1 - c_1 \exp(-c_2 \log p). \tag{A.5}$$

Finally to obtain uniformity over $\tau \in \big\{\tau;\, \Phi^*(\tau_0, \tau) \geq v_n\big\}$ note that for $\tau > \tau_0$, we have $\zeta_i(\tau) = \mathbf{1}\big[w_i \in (\tau_0, \tau]\big] \geq \mathbf{1}\big[w_i \in (\tau_0, \tau_1]\big]$ and for any $\tau < \tau_0$, we have $\zeta_i(\tau) = \mathbf{1}\big[w_i \in [\tau, \tau_0)\big] \geq \mathbf{1}\big[w_i \in [\tau_2, \tau_0)\big]$. This implies that

$$\inf_{\{\tau;\, \Phi^*(\tau_0,\tau)\geq v_n\}} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau) \geq \min_{\tau\in\{\tau_1,\tau_2\}} \frac{1}{n}\sum_{i=1}^n \zeta_i(\tau). \tag{A.6}$$

Part(ii) follows by combining (A.5) and (A.6). This complete the proof of Lemma 3.1. □

Proof of Lemma 3.2: We begin with the proof of Part (i). Note that the RHS of the inequality in Part (i) is normalized by the ℓ_2 norm of δ. Hence, without loss of generality we can assume $\|\delta\|_2 = 1$. Now, the proof of this lemma relies on $|n_w| = \sum_{i=1}^n \zeta_i(\tau)$, where $\zeta_i(\tau)$ are as defined for Lemma 3.1. Note that if $|n_w| = 0$ then Lemma 3.2 holds trivially with probability 1, thus without loss of generality we shall assume that $|n_w| > 0$. Now, for any fixed $\tau \in \mathcal{T}(\tau_0, u_n)$, we have

$$\Big\|\frac{1}{n}\sum_{i\in n_w} \delta^T x_i x_i^T\Big\|_\infty \leq \frac{|n_w|}{n}\Big\|\frac{1}{|n_w|}\sum_{i\in n_w} \delta^T x_i x_i^T\Big\|_\infty \tag{A.7}$$

The second key observation is that under Condition A(iv) and by properties of conditional expectations (see e.g. Lemma B.4), the conditional probability $P_w(\cdot) = P(\cdot|w)$ can be bounded by treating w as a constant. Thus,

$$P_w\Big(\Big\|\frac{\sum_{i\in n_w}\delta^T x_i x_i^T}{|n_w|} - \delta^T\Sigma\Big\|_\infty > t\Big) \leq 6p\exp(-c_u|n_w|\min\{\frac{t^2}{\sigma_x^4}, \frac{t}{\sigma_x^2}\})$$

where the above probability bound is obtained by an application of Part (ii) of Lemma 14 of Loh and Wainwright (2012): supplementary materials. This lemma is reproduced as Lemma B.1 in the Appendix. Now choosing $t = c_u \max\Big\{\sigma_x^2\sqrt{\frac{\log p}{|n_w|}}, \sigma_x\frac{\log p}{|n_w|}\Big\}$ we obtain,

$$P_w\left(\Big\|\frac{\sum_{i\in n_w}\delta^T x_i x_i^T}{|n_w|}\Big\|_\infty \leq \|\delta^T\Sigma\|_\infty + c_u\max\Big\{\sigma_x^2\sqrt{\frac{\log p}{|n_w|}}, \sigma_x\frac{\log p}{|n_w|}\Big\}\right) \geq 1 - c_1\exp(-c_2\log p). \quad (A.8)$$

The result in (A.8) together with (A.7) yields,

$$P_w\left(\Big\|\frac{1}{n}\sum_{i\in n_w}\delta^T x_i x_i^T\Big\|_\infty \leq \frac{|n_w|}{n}\|\delta^T\Sigma\|_\infty + \frac{|n_w|}{n}c_u\max\Big\{\sigma_x^2\sqrt{\frac{\log p}{|n_w|}}, \sigma_x\frac{\log p}{|n_w|}\Big\}\right) \geq 1 - c_1\exp(-c_2\log p). \quad (A.9)$$

Taking expectations on both sides of the inequality (A.9) and observing that the RHS of the conditional probability (A.9) is free of w, we obtain,

$$P\left(\Big\|\frac{1}{n}\sum_{i\in n_w}\delta^T x_i x_i^T\Big\|_\infty \leq \frac{|n_w|}{n}\|\delta^T\Sigma\|_\infty + \frac{|n_w|}{n}c_u\max\Big\{\sigma_x^2\sqrt{\frac{\log p}{|n_w|}}, \sigma_x\frac{\log p}{|n_w|}\Big\}\right) \geq 1 - c_1\exp(-c_2\log p) \quad (A.10)$$

On the other hand, we have by the result of Lemma 3.1 that with probability at least $1 - c_1\exp(-c_2\log p)$ that $\sup_{\tau\in\mathcal{T}}|n_w|/n \leq c_u\max\{\log p/n, u_n\}$. Also, it is straightforward to see that $\|\delta^T\Sigma\|_\infty \leq c_u\phi$, for some constant $c_u > 0$. Thus with the same probability we have the bound,

$$\sup_{\tau\in\mathcal{T}(\tau_0,u_n)}\frac{|n_w|}{n}\|\delta^T\Sigma\|_\infty \leq c_u\phi\max\Big\{\frac{\log p}{n}, u_n\Big\}. \quad (A.11)$$

By applying Part (i) of Lemma 3.1 we also have the following bound with probability at least $1 - c_1\exp(-c_2\log p)$,

$$\sup_{\tau\in\mathcal{T}(\tau_0,u_n)}\frac{|n_w|}{n}\sqrt{\frac{\log p}{|n_w|}} \leq c_u\sqrt{\frac{\log p}{n}}\max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\} \leq c_u\max\Big\{\frac{\log p}{n}, u_n\Big\}. \quad (A.12)$$

The final inequality follows upon noting that if $\sqrt{\log p/n}\sqrt{u_n} \geq u_n$ then $u_n \leq \log p/n$. Finally also note that $\sup_{\tau\in\mathcal{T}}(|n_w|/n)(\log p/|n_w|) \leq \log p/n$. Part (i) of the lemma follows

by combining these results together with the bounds (A.11) and (A.12) in (A.10). The proofs of Part (ii) and Part (iii) are similar and are thus omitted. □

Proof of Lemma 3.3 To prove Part (i), first define $z_i = x_i\mathbf{1}[w_i \leq \tau]$. Clearly z_i is also subgaussian with the same variance parameter as x_i's, i.e., σ_x^2. Furthermore, since by assumption $\Phi(\tau) > 0$, thus $\Sigma_z = Ez_iz_i^T = \Phi(\tau)\Sigma_x$, which implies that $\lambda_{\min}(\Sigma_z) = \Phi(\tau)\lambda_{\min}(\Sigma_x) \geq \Phi(\tau)\kappa$. Similarly $\lambda_{\max}(\Sigma_z) \leq \Phi(\tau)\phi$. Now applying Lemma B.2 we obtain

$$\frac{1}{n}\sum_{\{i;w_i\leq\tau\}}\delta^T x_i x_i^T\delta = \frac{1}{n}\sum_{i=1}^n\delta^T z_i z_i^T\delta \geq c_u\kappa\Phi(\tau)\|\delta\|_2^2 - c_u\frac{1}{\Phi(\tau)}\frac{\log p}{n}\|\delta\|_1^2, \quad (A.13)$$

with probability at least $1 - c_1\exp(-c_2\log p)$. Since $\delta \in \mathbb{A}$, it is straightforward to see that $\|\delta\|_1^2 \leq c_u s\|\delta\|_2^2$. This together with Condition A(ii) yields Part (i). The proof of Part (ii) is quite similar. To prove Part (iii), we shall invoke the arguments seen in the proof of Lemma 3.2. Consider,

$$\sup_{\tau\in\mathcal{T}(\tau_0,u_n)}\sup_{\delta\in\mathbb{A}}\frac{1}{n}\sum_{i\in n_w}\delta^T x_i x_i^T\delta = \sup_{\tau\in\mathcal{T}(\tau_0,u_n)}\frac{|n_w|}{n}\sup_{\delta\in\mathbb{A}}\frac{1}{|n_w|}\sum_{i\in n_w}\delta^T x_i x_i^T\delta \quad (A.14)$$

Let $P_w(\cdot)$ denote the conditional probability $P(\cdot|w)$, where $w = (w_1, ..., w_n)^T$. Then using Lemma B.2 we have

$$P_w\left(\sup_{\delta\in\mathbb{A}}\frac{1}{|n_w|}\sum_{i\in n_w}\delta^T x_i x_i^T\delta \leq \frac{3\phi}{2}\|\delta\|_2^2 + c_u c_m\frac{\log p}{|n_w|}\|\delta\|_1^2\right) \geq 1 - c_1\exp(-c_2\log p). \quad (A.15)$$

Noting that the above probability on the RHS of (A.15) is free of w, taking expectations on both sides we obtain,

$$P\left(\sup_{\delta\in\mathbb{A}}\frac{1}{|n_w|}\sum_{i\in n_w}\delta^T x_i x_i^T\delta \leq \frac{3\phi}{2}\|\delta\|_2^2 + c_u c_m\frac{\log p}{|n_w|}\|\delta\|_1^2\right) \geq 1 - c_1\exp(-c_2\log p). \quad (A.16)$$

Recall from Lemma 3.1 that $\sup_{\tau\in\mathcal{T}(\tau_0,u_n)}|n_w|/n \leq c_u\max\{\log p/n, u_n\}$, with probability at least $1 - c_1\exp(-c_2\log p)$. Combining this result with (A.16) and substituting into (A.14) we obtain

$$\begin{aligned}\sup_{\tau\in\mathcal{T}(\tau_0,u_n)}\sup_{\delta\in\mathbb{A}}\frac{1}{n}\sum_{i\in n_w}\delta^T x_i x_i^T\delta &\leq c_u\phi\|\delta\|_2^2\max\Big\{\frac{\log p}{n}, u_n\Big\} + c_u c_m\frac{s\log p}{n}\|\delta\|_2^2 \\ &\leq c_u c_m\|\delta\|_2^2\max\Big\{\frac{s\log p}{n}, u_n\Big\}\end{aligned}$$

with probability at least $1 - c_1\exp(-c_2\log p)$. This completes the proof of Part (iii). Proof of Part (iv) is based on similar arguments. First applying the same conditional argument as above, Part (i) of Lemma B.2 yields,

$$P\left(\inf_{\delta\in\mathbb{A}_2}\frac{1}{|n_w|}\sum_{i\in n_w}\delta^T x_i x_i^T\delta \geq \frac{\kappa}{2}\|\delta\|_2^2 - c_u c_m\frac{\log p}{|n_w|}\|\delta\|_1^2\right) \geq 1 - c_1\exp(-c_2\log p). \quad (A.17)$$

Since $v_n \geq c \log p/n$, Part (ii) of Lemma 3.1 gives,

$$\inf_{\substack{\tau\in\mathbb{R};\\ \Phi(\tau_0,\tau)\geq v_n}} \inf_{\delta\in\mathbb{A}_2} \frac{1}{n}\sum_{i\in n_w} \delta^T x_i x_i^T \delta \geq c_u\kappa\|\delta\|_2^2 v_n - c_u c_m \frac{\log p}{n}\|\delta\|_1^2, \tag{A.18}$$

with probability at least $1 - c_1\exp(-c_2 \log p)$. By the definition of the set $\mathbb{A}_2$ together with the fact that $\|\beta_0 - \gamma_0\|_0 \leq s$, we also have

$$\|\delta\|_1^2 \leq c_u s(\|\delta\|_2^2 + \|\beta_0 - \gamma_0\|_2^2). \tag{A.19}$$

Finally, substituting (A.19) in (A.18) we obtain

$$\inf_{\substack{\tau\in\mathbb{R};\\ \Phi(\tau_0,\tau)\geq v_n}} \inf_{\delta\in\mathbb{A}_2} \frac{1}{n}\sum_{i\in n_w} \delta^T x_i x_i^T \delta \geq c_u\kappa\|\delta\|_2^2 v_n - c_u c_m \frac{s\log p}{n}\|\delta\|_2^2 - c_u c_m \frac{s\log p}{n}\|\beta_0 - \gamma_0\|_2^2$$

with probability at least $1 - c_1\exp(-c_2 \log p)$. This completes the proof of the lemma. □

Proof of Theorem 4.1: To prove part (i), first note that when $\Phi_{\min}(\tau_0) = 0$, the model (1.1) reduces to an ordinary linear regression model with regression coefficient γ_0. Thus for any $\tau \in \mathbb{R}$, the estimates $\hat{\beta}(\tau)$ and $\hat{\gamma}(\tau)$ are ordinary Lasso estimates on the binary partitioned data (y_i, z_i), where $z_i = x_i\mathbf{1}[w_i \leq \tau]$, and $z_i = x_i\mathbf{1}[w_i > \tau]$, respectively. Also note that by assumption $\Phi_{\min}^{-1}(\tau)s\log p/n = o(1)$, thus the restricted eigenvalue condition of Part (i) and Part (ii) of Lemma 3.3 are applicable. The remaining arguments to prove the desired bounds are the same as typically used to derive bounds for Lasso estimates, such as those given in Chapter 6 of Bühlmann and Van De Geer (2011), these arguments are also similar to those to follow for the proof of Part (ii) and are thus omitted.

For the proof of Part (ii) where $\Phi_{\min}(\tau_0) > 0$, we only prove the uniform bound for the error in estimate $\|\hat{\beta}(\tau) - \beta_0\|_q$. The proof for $\|\hat{\gamma}(\tau) - \gamma_0\|_q$ is nearly identical. First, for any $\tau \in \mathcal{T}(\tau_0, u_n)$, note that by Lemma 3.2, we have,

$$\begin{aligned}
\sup_{\tau\in\mathcal{T}(\tau_0,u_n)} \Big\|\frac{1}{n}\sum_{i;w_i\leq\tau}\varepsilon_i x_i^T\Big\|_\infty &\leq \|\frac{1}{n}\sum_{i;w_i\leq\tau_0}\varepsilon_i x_i^T\| + \sup_{\tau\in\mathcal{T}(\tau_0,u_n)}\Big\|\frac{1}{n}\sum_{i\in n_w}\varepsilon_i x_i^T\Big\|_\infty \\
&\leq c_u c_m\sqrt{\frac{\log p}{n}} + c_u c_m\sqrt{\frac{\log p}{n}}\max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\} \\
&\leq c_u c_m\sqrt{\frac{\log p}{n}}
\end{aligned} \tag{A.20}$$

with probability at least $1 - c_1\exp(-c_2\log p)$. Also we have for any $\beta \in \mathbb{R}^p$ and $\tau \in \mathbb{R}$,

$$\begin{aligned}
\frac{1}{n}\sum_{i;w_i\leq\tau}(y_i - x_i^T\beta)^2 &= \frac{1}{n}\sum_{i;w_i\leq\tau}(y_i - x_i^T\beta_0 - x_i^T(\beta-\beta_0))^2 \\
&= \frac{1}{n}\sum_{i;w_i\leq\tau}\tilde{\varepsilon}_i^2 - \frac{2}{n}\sum_{i;w_i\leq\tau}\tilde{\varepsilon}_i x_i^T(\beta-\beta_0) + \frac{1}{n}\sum_{i;w_i\leq\tau}\|x_i^T(\beta-\beta_0)\|_2^2.
\end{aligned} \tag{A.21}$$

Here $\tilde{\varepsilon}_i = \varepsilon_i$, for $i \in \{i; w_i \leq \tau_0\}$ and $\tilde{\varepsilon}_i = \varepsilon_i - x_i^T(\beta_0 - \gamma_0)$ for $i \in \{i; w_i > \tau_0\}$. Now by the definition of $\hat{\beta}(\tau)$, it follows that

$$\frac{1}{n}\sum_{i;w_i\leq\tau}(y_i - x_i^T\hat{\beta}(\tau))^2 + \lambda_1\|\hat{\beta}(\tau)\|_1 \leq \frac{1}{n}\sum_{i;w_i\leq\tau}(y_i - x_i^T\beta_0)^2 + \lambda_1\|\beta_0\|_1. \tag{A.22}$$

Applying (A.21) in (A.22) and carrying out some algebraic operations we get

$$\begin{aligned}
&\frac{1}{n}\sum_{i;w_i\leq\tau}\|x_i^T(\hat{\beta}(\tau)-\beta_0)\|_2^2 + \lambda_1\|\hat{\beta}(\tau)\|_1 \\
&\leq \lambda_1\|\beta_0\|_1 + \Big|\frac{2}{n}\sum_{i;w_i\leq\tau}\varepsilon_i x_i(\hat{\beta}(\tau)-\beta_0)\Big| + \Big|\frac{2}{n}\sum_{\tau_0<w_i\leq\tau}(\beta_0-\gamma_0)x_i x_i^T(\hat{\beta}(\tau)-\beta_0)\Big| \\
&\leq \Big\|\frac{2}{n}\sum_{i;w_i\leq\tau}\varepsilon_i x_i\Big\|_\infty\|\hat{\beta}(\tau)-\beta_0\|_1 + \Big\|\frac{2}{n}\sum_{i\in n_w}(\beta_0-\gamma_0)^T x_i x_i^T\Big\|_\infty\|\hat{\beta}(\tau)-\beta_0\|_1 + \lambda_1\|\beta_0\|_1 \\
&\leq c_u c_m\|\beta_0-\gamma_0\|_2\max\Big\{\frac{\log p}{n}, u_n\Big\}\|\hat{\beta}(\tau)-\beta_0\|_1 + c_u c_m\max\sqrt{\frac{\log p}{n}}\|\hat{\beta}(\tau)-\beta_0\|_1 + \lambda_1\|\beta_0\|_1 \\
&\leq \lambda\|\hat{\beta}(\tau)-\beta_0\|_1 + \lambda_1\|\beta_0\|_1.
\end{aligned} \tag{A.23}$$

Here $\lambda = c_u c_m \max\Big\{\sqrt{\log p/n},\ \|\beta_0-\gamma_0\|_2 u_n\Big\}$. The first term of the second to last inequality follows from (A.20) and the second term from Part (i) of Lemma 3.2. The bound (A.23) holds uniformly over $\tau \in \mathcal{T}(\tau_0, u_n)$ with probability at least $1 - c_1\exp(-c_2\log p)$. Observe that the first term on the LHS of inequalities (A.23) is nonnegative, therefore $\lambda_1\|\hat{\beta}\|_1 \leq \lambda\|\hat{\beta}-\beta_0\|_1 + \lambda_1\|\beta_0\|$. Choosing $\lambda_1 \geq 2\lambda$ leads to the inequality $\|\hat{\beta}_{S^c}\|_1 \leq 3\|\hat{\beta}_S - \hat{\beta}_{0S}\|_1$, by elementary triangle inequalities, see for e.g. Lemma 6.3 of Bühlmann and Van De Geer (2011). Thus $\delta = \hat{\beta} - \beta_0 \in \mathbb{A}$ and thus the first three inequalities of Lemma 3.3 are now applicable. From (A.23) we obtain,

$$\begin{aligned}
\frac{2}{n}\sum_{i;w_i\leq\tau_0}\|x_i^T(\hat{\beta}(\tau)-\beta_0)\|_2^2 - \frac{2}{n}\sum_{i\in n_w}\|x_i^T(\hat{\beta}(\tau)-\beta_0)\|_2^2 &\leq 3\lambda_1\|\hat{\beta}(\tau)-\beta_0\|_1 \\
&\leq 3\sqrt{s}\lambda_1\|\hat{\beta}(\tau)-\beta_0\|_2
\end{aligned} \tag{A.24}$$

Bounding the terms on the LHS of (A.24) by applying Part (i) and (iii) of Lemma 3.3 together with the assumption $u_n = o(\Phi(\tau_0))$, yields

$$c_u c_m\Phi(\tau_0)\|\hat{\beta}-\beta_0\|_2^2 \leq 3\sqrt{s}\lambda_1\|\hat{\beta}-\beta_0\|_2$$

This directly implies $\|\hat{\beta}(\tau)-\beta_0\|_2 \leq c_u c_m\sqrt{s}\lambda_1$. The ℓ_1 bound $\|\hat{\beta}(\tau)-\beta_0\|_1 \leq \sqrt{s}\|\hat{\beta}-\beta_0\|_2$, follows from the previously shown result that $\hat{\beta}(\tau)-\beta_0 \in \mathbb{A}$. To complete the proof of Part (ii), note that all bounds in the above arguments hold uniformly over $\mathcal{T}(\tau_0, u_n)$, consequently the final bound holds uniformly over $\mathcal{T}(\tau_0, u_n)$. □

Proof of Lemma 4.1: We begin with proving Part (i), where $\Phi(\tau_0) = 0$, in this case, for any $\tau \in \mathbb{R}$, we have

$$\begin{aligned} nR_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) &= \sum_{i;w_i \le \tau} (y_i - x_i^T \hat{\beta}^{(0)})^2 + \sum_{i;w_i > \tau} (y_i - x_i^T \hat{\gamma}^{(0)})^2 - \sum_{i;w_i > \tau_0} (y_i - x_i^T \hat{\gamma}^{(0)})^2 \\ &= \sum_{i;w_i \le \tau} (y_i - x_i^T \hat{\beta}^{(0)})^2 - \sum_{i;\tau_0 < w_i \le \tau} (y_i - x_i^T \hat{\gamma}^{(0)})^2 \\ &= \sum_{i;w_i \le \tau} (\hat{\beta}^{(0)} - \gamma_0)^T x_i x_i^T (\hat{\beta}^{(0)} - \gamma_0) - \sum_{i;\tau_0 < w_i \le \tau} (\hat{\gamma}^{(0)} - \gamma_0)^T x_i x_i^T (\hat{\gamma}^{(0)} - \gamma_0) \\ &\quad -2 \sum_{i;w_i \le \tau} \varepsilon_i x_i^T (\hat{\beta}^{(0)} - \gamma_0) + 2 \sum_{i;w_i \le \tau} \varepsilon_i x_i^T (\hat{\gamma}^{(0)} - \gamma_0) \\ &\ge -\sum_{i \in n_w} (\hat{\gamma}^{(0)} - \gamma_0)^T x_i x_i^T (\hat{\gamma}^{(0)} - \gamma_0) - 2 \sum_{i;w_i \le \tau} \varepsilon_i x_i^T (\hat{\beta}^{(0)} - \gamma_0) \\ &\quad +2 \sum_{i;w_i \le \tau} \varepsilon_i x_i^T (\hat{\gamma}^{(0)} - \gamma_0) \end{aligned} \tag{A.25}$$

Now, by the result in Remark 4.1, we have that $\|\hat{\beta}^{(0)} - \gamma_0\|_2, \|\hat{\gamma}^{(0)} - \gamma_0\| \le c_u c_m \sqrt{s \log p / n \Phi^2_{\min}(\tau^{(0)})}$, with probability at least $1 - c_1 \exp(-c_2 \log p)$. Additionally from the proof of Theorem 4.1, it has also been shown that $\hat{\beta}^{(0)} - \gamma_0$ and $\hat{\gamma}^{(0)} - \gamma_0$ lie in the set $\mathcal{A}$ of (3.2), with the same probability. Thus the bounds of Lemma 3.2 are applicable. Substituting these bounds in (A.25), for any $v_n > 0$, we obtain uniformly over $\mathcal{H}(1, v_n)$ that,

$$\inf_{\tau \in \mathcal{H}(u_n, v_n)} R_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \ge -c_u c_m \frac{s \log p}{n \Phi^2_{\min}(\tau^{(0)})} - c_u c_m \frac{s \log p}{n \Phi_{\min}(\tau^{(0)})}$$

Finally, recall that $S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) = R_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) + \mu\big(\|\Phi(\tau)\|_0 - \|\Phi(\tau_0)\|_0\big)$, and since in this case $\|\Phi(\tau)\|_0 = 0$, and for any $\tau \in \mathcal{H}(1, v_n)$, we have $\|\Phi(\tau)\|_0 = 1$ (since $v_n > 0$), hence the statement of Part (i) follows directly.

To prove Part (ii), where $\Phi(\tau_0) > 0$, we divide the argument into two cases. First consider the case where $\tau \in \mathcal{H}(u_n, v_n)$, with $\tau \ge \tau_0$, here,

$$\begin{aligned} nR_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) &= nQ(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) - nQ(\tau_0, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \\ &= \sum_{i \in \tau_0 < w_i \le \tau} (y_i - x_i^T \beta)^2 - \sum_{i \in \tau_0 < w_i \le \tau} (y_i - x_i^T \gamma)^2 \end{aligned} \tag{A.26}$$

Recall by construction of model (1.1), $\varepsilon_i = y_i - x_i^T \gamma_0$, for $i; w_i > \tau_0$. Using this relation in (A.26) and performing some algebraic manipulation we have that

$$\begin{aligned} R_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) &= \frac{1}{n} \sum_{i \in n_w} (\hat{\beta}^{(0)} - \gamma_0)^T x_i x_i^T (\hat{\beta}^{(0)} - \gamma_0) - \frac{1}{n} \sum_{i \in n_w} (\hat{\gamma}^{(0)} - \gamma_0)^T x_i x_i^T (\hat{\gamma}^{(0)} - \gamma_0) \\ &\quad -\frac{2}{n} \sum_{i \in n_w} \varepsilon_i x_i^T (\hat{\beta}^{(0)} - \gamma_0) + \frac{2}{n} \sum_{i \in n_w} \varepsilon_i x_i^T (\hat{\gamma}^{(0)} - \gamma_0) \\ &:= T1 + T2 + T3 + T4 \end{aligned} \tag{A.27}$$

Substituting bounds for term $(T1)$-$(T4)$ given in Lemma .1 (stated after this proof), we obtain,

$$\begin{aligned} \inf_{\tau \in \mathcal{H}(u_n, v_n)} R_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) &\ge c_u c_m \xi_n^2 v_n - c_u c_m \xi_n^2 \frac{s \log p}{n} \\ &\quad - c_u c_m \xi_n \sqrt{\frac{s \log p}{n}} \max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\} - c_u c_m r_n^2 \max\Big\{\frac{s \log p}{n}, u_n\Big\}. \end{aligned}$$

Now, note that for any $\tau \in \mathbb{R}$, we have that $\|\Phi(\tau)\|_0 - \|\Phi(\tau_0)\|_0 \le 1$. Also, when $u_n / \Phi(\tau_0) \to 0$, for any $\tau \in \mathcal{H}(u_n, v_n)$, the quantities $\Phi(\tau_0)$ and $\Phi(\tau)$ will have the same sign, consequently $\|\Phi(\tau)\|_0 - \|\Phi(\tau_0)\|_0 = 0$. In effect, we have for any $\tau \in \mathcal{H}(u_n, v_n)$, that, $\|\Phi(\tau)\|_0 - \|\Phi(\tau_0)\|_0 \le F(u_n)$. Using this relation in the definition of $S(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)})$, together with the assumption that $\xi_n > c_u$ we obtain,

$$\begin{aligned} \inf_{\tau \in \mathcal{H}(u_n, v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \ge \xi_n^2 \Big(& c_u c_m v_n - c_u c_m \frac{s \log p}{n} - \frac{c_u c_m}{1 \vee \xi_n} \sqrt{\frac{\log p}{n}} \max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\} \\ & - c_u c_m \frac{r_n^2}{1 \vee \xi_n^2} \max\Big\{\frac{s \log p}{n}, u_n\Big\} - \frac{c_u \mu}{1 \vee \xi_n^2} F(u_n) \Big). \end{aligned}$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. This completes the proof of this lemma. □

Lemma .1 *Suppose the conditions of Lemma 4.1 and let the terms $T1$, $T2$, $T3$ and $T4$ be as defined in (A.27). Then for n sufficiently large, we have the following bounds,*

$$\begin{aligned} &(i) && \inf_{\tau \in \mathcal{H}(u_n, v_n)} |T1| \ge c_u c_m \xi_n^2 v_n - \xi_n^2 \frac{s \log p}{n} \\ &(ii) && \sup_{\tau \in \mathcal{H}(u_n, v_n)} |T2| \le c_u c_m r_n^2 \max\Big\{\frac{s \log p}{n}, u_n\Big\} \\ &(iii) && \sup_{\tau \in \mathcal{H}(u_n, v_n)} |T3| \le c_u c_m \xi_n \sqrt{\frac{s \log p}{n}} \max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\} \\ &(iv) && \sup_{\tau \in \mathcal{H}(u_n, v_n)} |T4| \le c_u c_m r_n \sqrt{\frac{s \log p}{n}} \max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\} \end{aligned}$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$.

Proof of Lemma .1: Consider the term $T1 = n^{-1} \sum_{i \in n_w} (\hat{\beta}^{(0)} - \gamma_0)^T x_i x_i^T (\hat{\beta}^{(0)} - \gamma_0)$. First, recall from the proof of Theorem 4.1 that $\hat{\beta}^{(0)} - \beta_0 \in \mathbb{A}$, with probability at least $1 - c_1 \exp(-c_2 \log p)$. Thus, as described in Remark 3.1, we have that $\delta = \hat{\beta}^{(0)} - \gamma_0 = \hat{\beta}^{(0)} - \beta_0 + \beta_0 - \gamma_0 \in \mathbb{A}_2$ with the same probability. Now applying Part (iv) of Lemma 3.3 we obtain,

$$\inf_{\tau \in \mathcal{H}(u_n, v_n)} \frac{1}{n} \sum_{i \in n_w} \delta^T x_i x_i^T \delta \ge c_u c_m v_n \|\delta\|_2^2 - c_u c_m \frac{s \log p}{n} \big(\|\delta\|_2^2 + \xi_n^2\big)$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$. Applying the algebraic inequality, $\|\delta_1 + \delta_2\|_2^2 \ge \|\delta_1\|_2^2 + \|\delta_2\|_2^2 - 2\|\delta_1\|_2 \|\delta_2\|_2$ which is applicable for any $\delta_1, \delta_2 \in \mathbb{R}^p$, we obtain,

$\|\hat{\beta}^{(0)} - \beta_0 + \beta_0 - \gamma_0\|_2^2 \geq r_n^2 + \xi_n^2 - 2r_n\xi_n$. Now, by definition of r_n, we also have that $r_n = o(1)\xi_n$, thus $\|\hat{\beta}^{(0)} - \beta_0 + \beta_0 - \gamma_0\|_2^2 \geq c_u\xi_n^2$, for n large. Similarly, using the inequality $\|\delta_1+\delta_2\|_2^2 \leq \|\delta_1\|_2^2+\|\delta_2\|_2^2+2\|\delta_1\|_2\|\delta_2\|_2$, we can show that for n large, $\|\hat{\beta}^{(0)}-\beta_0+\beta_0-\gamma_0\|_2^2 \leq c_u\xi_n^2$. Substituting these bounds back in (A.28) we obtain the result of Part (i). Next consider Part (ii), where we have $T2 = n^{-1}\sum_{i\in n_w}(\hat{\gamma}^{(0)} - \gamma_0)^T x_i x_i^T(\hat{\gamma}^{(0)} - \gamma_0)$. Recall from the proof of Theorem 4.1, $\hat{\gamma}^{(0)} - \gamma_0 \in \mathbb{A}$. Now applying Part (iii) of Lemma 3.3 we obtain,

$$\sup_{\tau\in\mathcal{H}(u_n,v_n)} \frac{1}{n}\sum_{i\in n_w}(\hat{\gamma}^{(0)} - \gamma_0)^T x_i x_i^T(\hat{\gamma}^{(0)} - \gamma_0) \leq c_u c_m r_n^2 \max\Big\{\frac{s\log p}{n},\, u_n\Big\}$$

with probability at least $1 - c_1\exp(-c_2\log p)$. This proves Part (ii). The proof of Part (iii) follows by an application of Part (iii) of Lemma 3.2, i.e.,

$$\begin{aligned}\sup_{\tau\in\mathcal{H}(u_n,v_n)} |\frac{1}{n}\sum_{i\in n_w}\varepsilon_i x_i^T(\hat{\beta} - \gamma_0)| &\leq \sup_{\tau\in\mathcal{H}(u_n,v_n)} \|\frac{1}{n}\sum_{i\in n_w}\varepsilon_i x_i^T\|_\infty \|(\hat{\beta}^{(0)} - \gamma_0)\|_1 \\ &\leq \sup_{\tau\in\mathcal{H}(u_n,v_n)} \|\frac{1}{n}\sum_{i\in n_w}\varepsilon_i x_i^T\|_\infty \sqrt{s}\big(\|(\hat{\beta}^{(0)} - \beta_0)\|_2 + \|\beta_0 - \gamma_0\|_1\big) \\ &\leq c_u c_m r_n\sqrt{\frac{s\log p}{n}}\max\Big\{\sqrt{\frac{\log p}{n}},\, \sqrt{u_n}\Big\} + c_u c_m \xi_n\sqrt{\frac{s\log p}{n}}\max\Big\{\sqrt{\frac{\log p}{n}},\, \sqrt{u_n}\Big\}\end{aligned}$$

with probability at least $1 - c_1\exp(-c_2\log p)$. This proves Part (iii). The proof of Part (iv) is very similar and is thus omitted. □

Proof of Theorem 4.2: First consider Part (i), where $\Phi(\tau_0) = 0$. Applying Part (i) of Lemma 4.1 for any $v_n > 0$, we have,

$$\inf_{\tau\in\mathcal{H}(1,v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \geq \mu - c_u c_m \frac{s\log p}{n\Phi_{\min}^2(\tau^{(0)})}$$

with probability at least $1-c_1\exp(-c_2\log p)$. Recall the choice of $\mu = c_u c_m\big(s\log p/nl_n^2\big)^{1/k^*}$, and the initializing condition $\Phi_{\min}(\tau^{(0)}) \geq c_u l_n$. Consequently, $\inf_{\tau\in\mathcal{H}(1,v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) > 0$, for n sufficiently large, with the same probability. This implies that $\hat{\tau}^{(1)} \notin \mathcal{H}(1, v_n)$ for any $v_n > 0$. Thereby proving that $\Phi(\hat{\tau}^{(1)}) = 0$ is the only remaining possibility with the same probability. This completes the proof of Part (i).

To prove Part (ii), for any $v_n \geq s\log p/n$, we apply Part (ii) of Lemma 4.1 on the set $\mathcal{H}(1, v_n)$, to obtain,

$$\begin{aligned}\inf_{\tau\in\mathcal{H}(1,v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \geq \xi_n^2\Big(&c_u c_m v_n - c_u c_m\frac{s\log p}{n} - \frac{c_u c_m}{1\vee\xi_n}\sqrt{\frac{s\log p}{n}} \\ &- c_u c_m\frac{r_n^2}{1\vee\xi_n^2} - \frac{c_u\mu}{1\vee\xi_n^2}\Big),\end{aligned}$$

with probability at least $1 - c_1\exp(-c_2\log p)$. Note that, by Condition A(iii) we have that $(s/l_n^2)u_n^{(0)} = o(1)$. Then, upon choosing,

$$v_n \geq v_n^* \;:=\; c_u c_m \max\Big\{\frac{s\log p}{n},\, \frac{1}{1\vee\xi_n}\Big(\frac{s\log p}{nl_n^2}\Big)^{1/k^*}\Big\}$$

for some $c_u > 0$, we have that $\inf_{\tau\in\mathcal{H}(1,v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) > 0$, for n sufficiently large. This follows by the choice of $\mu = (s\log p/nl_n^2)^{1/k^*}$, and by $r_n^2/\xi_n^2 < v_n^*$, which in turn follows from Condition A(iii). This implies that $\hat{\tau}^{(1)} \notin \mathcal{H}(1, v_n^*)$, i.e., $|\Phi(\hat{\tau}^{(1)}) - \Phi(\tau_0)| \leq v_n^*$ with probability at least $1 - c_1\exp(-c_2\log p)$. Note that if $\frac{1}{1\vee\xi_n}\Big(\frac{s\log p}{nl_n^2}\Big)^{1/k^*} \leq \frac{s\log p}{n}$, then the result is already proved. Else, reset $u_n = v_n^*$ and reapply the above argument for any $v_n \geq s\log p/n$, to obtain,

$$\begin{aligned}\inf_{\tau\in\mathcal{H}(u_n,v_n)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) \geq \xi_n^2\Big(&c_u c_m v_n - c_u c_m\frac{s\log p}{n} - \frac{c_u c_m}{1\vee\xi_n}\sqrt{\frac{s\log p}{n}}\max\Big\{\sqrt{\frac{\log p}{n}}, \sqrt{u_n}\Big\} \\ &- c_u c_m\frac{r_n^2}{\xi_n^2}\max\Big\{\frac{s\log p}{n},\, u_n\Big\}\Big) \quad .\end{aligned}$$

Here, the term $F(u_n) = 0$, since for $u_n = v_n^*$, the sign of $\Phi(\hat{\tau}^{(1)})$ is the same as that of $\Phi(\tau_0)$, for n large. Now, upon choosing,

$$v_n \geq v_n^* := c_u c_m \max\Big\{\frac{s\log p}{n},\, \frac{1}{1\vee\xi_n^{1+\frac{1}{2k^*}}}\Big(\frac{s\log p}{nl_n^2}\Big)^{a_2}\Big\}, \text{ with, } a_2 = \min\Big\{\frac{1}{2} + \frac{1}{2k^*}, \frac{1}{k^*} + \frac{1}{k^*}\Big\},$$

we obtain that for n large, $\inf_{\tau\in\mathcal{H}(u_n,v_n^*)} S_n(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) > 0$, with probability at least $1 - c_1\exp(-c_2\log p)$. Consequently $\hat{\tau}^{(1)} \notin \mathcal{H}(u_n, v_n^*)$, i.e., $|\Phi(\hat{\tau}^{(1)}) - \Phi(\tau_0)| \leq v_n^*$. Note that, by using the above recursive argument, we have tightened the desired rate at each step. As seen earlier, if the second term of the maximum expression is smaller than the first, then the proof is done. Else, continuing these recursions, by resetting u_n to the bound of the previous recursion, and applying Part (ii) of Lemma 4.1, we can obtain for the mth recursion that

$$|\Phi(\hat{\tau}^{(1)}) - \Phi(\tau_0)| \leq c_u c_m \max\Big\{\frac{s\log p}{n},\, \frac{1}{1\vee\xi_n^{b_m}}\Big(\frac{s\log p}{n}\Big)^{a_m}\Big\}, \quad \text{where,}$$

$$a_m = \min\Big\{\frac{1}{2} + \frac{a_{m-1}}{2}, \frac{1}{k^*} + a_{m-1}\Big\}, \quad \text{and} \quad b_m = 1 + b_{m-1}/2,$$

with $a_1 = b_1 = 1/k^*$. Note that, despite the recursions in the above argument, the probability of the bound obtained after every recursion is maintained to be at least $1 - c_1\exp(-c_2\log p)$, this follows from Remark .1. To finish the proof, note that $k^* \in [2, 3]$, $a_m = 1/2 + a_{m-1}/2$, $\forall m$ and when $k^* > 3$, $a_m = 1/2 + a_{m-1}/2$, for m large enough. Finally, if we continue the above recursions an infinite number of times we obtain $a_\infty = \sum_{m=1}^\infty 1/2^m = 1$, and $b_\infty = 1 + \sum_{m=1}^\infty 1/2^m = 2$. This finishes the proof of this theorem. □.

Remark .1 *(Observation utilized in the proof of Theorem 4.2):* The proof of Theorem 4.2 relies on a recursive application of Lemma 4.1, this in turn requires a recursive application of the bounds of Lemma .1, where the probability of all bounds holding simultaneously at each recursion being at least $1 - c_1\exp(-c_2\log p)$. Despite these recursions (potentially infinite) the result from the final recursion continues to hold with probability at least $1 - c_1\exp(-c_2\log p)$. To see this, let $u_n \to 0$ be any positive sequence and let $\{a_j\} \to a_\infty$, $j \to \infty$, $0 < a_j \leq 1$, be any strictly increasing sequence over $j = 1, 2, \ldots$. Then define sequences $u_n^j = u_n^{a_j}$, $j = 1, 2\ldots$. Here note that $u_n^{j+1} = o(u_n^j)$, $j = 1, \ldots$, i.e., each

sequence converges to zero faster than the preceding one. Let $\mathcal{E}_{u^1}, \mathcal{E}_{u^2}...$ be events, each with probability $1 - c_1 \exp(-c_2 \log p)$, on which the upper bounds of Lemma .1 hold for each $u_n^1, u_n^2, ...$ respectively. Clearly, on the intersection of events $\mathcal{E}_{u^1} \cap \mathcal{E}_{u^2} \cap ....$, all upper bounds of Lemma .1 hold simultaneously over any sequence u_n^j, $j = 1, ..., \infty$ Now, note that by the construction of these sequences, and that these are all upper bounds, the following containment holds $\mathcal{E}_{u^1} \supseteq \mathcal{E}_{u^2} \supseteq ... \supseteq \mathcal{E}_{u^\infty}$. This implies that on the event $\mathcal{E}_{u^\infty}$ all bounds of Lemma .1 hold simultaneously for any sequence $\{u_n^j\}$, $j = 1, ..., \infty$. Here $\mathcal{E}_{u^\infty}$ represents the set corresponding to the sequence $u_n^\infty = u_n^{a_\infty}$. Also, by a single application of Lemma .1, $P(\mathcal{E}_{u^\infty}) \geq 1 - c_1 \exp(-c_2 \log p)$. The same argument can be made for the lower bound of Lemma .1, with the direction of the containment switched.

Proof of Theorem 4.3: Recall that the result of Part (ii) of Theorem 4.1 is a uniform result over the set $\mathcal{T}(\tau_0, u_n)$. The proof of this theorem is now a direct application of Part (ii) of Theorem 4.1, since by the result of Theorem 4.2, we have that $\hat{\tau}^{(1)} \in \mathcal{T}(\tau_0, t_n)$, with probability at least $1 - c_1 \exp(-c_2 \log p)$. □

Appendix B: Auxiliary lemma's

Here we restate without proof the technical lemma's from the literature which have been used in the analysis presented in this manuscript.

Lemma B.1 *If $X \in \mathbb{R}^{n \times p_1}$ is a zero mean subgaussian matrix with parameters (Σ_x, σ_x^2), then for any fixed (unit) vector in $v \in \mathbb{R}^{p_1}$, we have*

$$(i)\ P\Big(\Big|\|Xv\|_2^2 - E\|Xv\|_2^2\Big| \geq nt\Big) \leq \exp\Big(-cn \min\Big\{\frac{t^2}{\sigma_x^4}, \frac{t}{\sigma_x^2}\Big\}\Big)$$

Moreover, if $Y \in \mathbb{R}^{n \times p_2}$ is a zero mean subgaussian matrix with parameters (Σ_y, σ_y^2), then

$$(ii)\ P\Big(\|\frac{Y^T X}{n} - \mathrm{cov}(y_i, x_i)\|_\infty \geq t\Big) \leq 6 p_1 p_2 \exp\Big(-cn \min\Big\{\frac{t^2}{\sigma_x^2 \sigma_y^2}, \frac{t}{\sigma_x \sigma_y}\Big\}\Big)$$

where x_i, y_i are the ith rows of X and Y respectively. In particular, if $n \geq c \log p$, then

$$(iii)\ P\Big(\|\frac{Y^T X}{n} - \mathrm{cov}(y_i, x_i)\|_\infty \geq c\sigma_x \sigma_y \sqrt{\frac{\log p}{n}}\Big) \leq c_1 \exp(-c_2 \log p).$$

This lemma provides tail bounds on subexponential r.v.'s and is as stated in Lemma 14 of Loh and Wainwright (2012): supplementary materials. The first part of this lemma is a restatement of Proposition 5.16 of Vershynin (2010) and the other two part are derived via algebraic manipulations of the product under consideration. The following is another useful result from Loh and Wainwright (2012) which provides control on restricted eigenvalues of the gram matrix.

Lemma B.2 *Let $z_i \in \mathbb{R}^p$, $i = 1, ..., n$ be i.i.d subgaussian random vectors with variance parameter σ_z^2 and covariance $\Sigma_z = E z_i z_i^T$. Also, let $\lambda_{\min}(\Sigma_z)$ and $\lambda_{\max}(\Sigma_z)$ be the minimum*

and maximum eigenvalues of the covariance matrix respectively. Then,

$$(i)\quad \frac{1}{n}\sum_{i=1}^n \delta^T z_i z_i^T \delta \geq \frac{\lambda_{\min}(\Sigma_z)}{2}\|\delta\|_2^2 - c_u \lambda_{\min}(\Sigma_z) \max\Big\{\frac{\sigma_z^4}{\lambda_{\min}^2(\Sigma_z)}, 1\Big\}\frac{\log p}{n}\|\delta\|_1^2, \quad \forall \delta \in \mathbb{R}^p,$$

$$(ii)\quad \frac{1}{n}\sum_{i=1}^n \delta^T z_i z_i^T \delta \leq \frac{3\lambda_{\max}(\Sigma_z)}{2}\|\delta\|_2^2 + c_u \lambda_{\min}(\Sigma_z) \max\Big\{\frac{\sigma_z^4}{\lambda_{\min}^2(\Sigma_z)}, 1\Big\}\frac{\log p}{n}\|\delta\|_1^2, \quad \forall \delta \in \mathbb{R}^p,$$

with probability at least $1 - c_1 \exp(-c_2 \log p)$.

A proof of this result in an errors-in-variables setting result can be found in the supplementary material of Loh and Wainwright (2012), However Lemma B.2 can be seen to follow as a special case (substitute $\sigma_w = 0$ in Lemma 1 of Loh and Wainwright (2012): supplementary materials).

Lemma B.3 *Let the $\{X_i\}_{i=1}^m$ be independent random variables, $EX_i^2 < \infty$, $X_i \geq 0$. Set $S = \sum_{i=1}^n X_i$ and let $t > 0$. Then*

$$P\Big(ES - S \geq t\Big) \leq \exp\Big(\frac{-t^2}{2\sum_{i=1}^n EX_i^2}\Big)$$

This result is as stated in Theorem 1 of Maurer (2003), it provides a lower bound on a sum of positive independent r.v.'s.

Lemma B.4 *Suppose X and Y are independent random variables. Let ϕ be a function with $E|\phi(X,Y)| < \infty$ and let $g(x) = E\phi(x, Y)$, then*

$$E\big(\phi(X,Y)|X\big) = g(X)$$

This is an elementary result on conditional expectations. A straightforward proof can be found in Example 1.5. page 222, Durrett (2010).

References

Yves Atchade and Leland Bybee. A scalable algorithm for gaussian graphical models with change-points. *arXiv preprint arXiv:1707.04306*, 2017.

Jushan Bai. Estimation of a change point in multiple regression models. *Review of Economics and Statistics*, 79(4):551–563, 1997.

Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

Alexandre Belloni, Victor Chernozhukov, Abhishek Kaul, Mathieu Rosenbaum, and Alexandre B Tsybakov. Pivotal estimation via self-normalization for high-dimensional linear models with error in variables. *arXiv preprint arXiv:1708.08353*, 2017a.

Alexandre Belloni, Mathieu Rosenbaum, and Alexandre B Tsybakov. Linear and conic programming estimators in high dimensional errors-in-variables models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):939–956, 2017b.

Peter J Bickel, Yaacov Ritov, Alexandre B Tsybakov, et al. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

Haeran Cho and Piotr Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(2):475–507, 2015.

Gabriela Ciuperca. Model selection by lasso methods in a change-point model. *Statistical Papers*, 55(2):349–374, 2014.

Rick Durrett. *Probability: theory and examples*. Cambridge university press, 2010.

Stergios B Fotopoulos, Venkata K Jandhyala, Elena Khapalova, et al. Exact asymptotic distribution of change-point mle for change in the mean of gaussian sequences. *The Annals of Applied Statistics*, 4(2):1081–1104, 2010.

Jerome H Friedman, TJ Hastie, and RJ Tibshirani. glmnet: lasso and elastic-net regularized generalized linear models, 2010b. *URL http://CRAN. R-project. org/package= glmnet. R package version*, pages 1–1, 2010.

Piotr Fryzlewicz. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.

Eric Gautier and Alexandre Tsybakov. High-dimensional instrumental variables regression and confidence sets. *arXiv preprint arXiv:1105.2454*, 2011.

Alex J Gibberd and Sandipan Roy. Multiple changepoint estimation in high-dimensional gaussian graphical models. *arXiv preprint arXiv:1712.05786*, 2017.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

David V Hinkley. Inference about the intersection in two-phase regression. *Biometrika*, 56 (3):495–504, 1969.

David V Hinkley. Inference about the change-point in a sequence of random variables. *Biometrika*, 1970.

David V Hinkley. Time-ordered classification. *Biometrika*, 59(3):509–523, 1972.

Venkata Jandhyala, Stergios Fotopoulos, Ian MacNeill, and Pengyu Liu. Inference for single and multiple change-points in time series. *Journal of Time Series Analysis*, 34(4):423–446, 2013.

Venkata K Jandhyala and Stergios B Fotopoulos. Capturing the distributional behaviour of the maximum likelihood estimator of a changepoint. *Biometrika*, 86(1):129–140, 1999.

Venkata K Jandhyala and Ian B MacNeill. Iterated partial sum sequences of regression residuals and tests for changepoints with continuity constraints. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(1):147–156, 1997.

Abhishek Kaul. Lasso with long memory regression errors. *Journal of Statistical Planning and Inference*, 153:11–26, 2014.

Abhishek Kaul and Hira L Koul. Weighted 1-penalized corrected quantile regression for high dimensional measurement error models. *Journal of Multivariate Analysis*, 140:72–91, 2015.

Abhishek Kaul, Ori Davidov, and Shyamal D Peddada. Structural zeros in high-dimensional data with applications to microbiome studies. *Biostatistics*, 18(3):422–433, 2017.

Roger Koenker and Ivan Mizera. Convex optimization in r. *Journal of Statistical Software*, 60(5):1–23, 2014.

Hira L Koul and Lianfen Qian. Asymptotics of maximum likelihood estimator in a two-phase linear regression model. *Journal of Statistical Planning and Inference*, 108(1-2): 99–119, 2002.

Hira L Koul, Lianfen Qian, and Donatas Surgailis. Asymptotics of m-estimators in two-phase linear regression models. *Stochastic Processes and their Applications*, 103(1):123–154, 2003.

Sokbae Lee, Myung Hwan Seo, and Youngki Shin. The lasso for high dimensional regression with a possible change point. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):193–210, 2016.

Sokbae Lee, Yuan Liao, Myung Hwan Seo, and Youngki Shin. Oracle estimation of a change point in high-dimensional quantile regression. *Journal of the American Statistical Association*, 0(0):1–11, 2018. doi: 10.1080/01621459.2017.1319840. URL `https://doi.org/10.1080/01621459.2017.1319840`.

Florencia Leonardi and Peter Bühlmann. Computationally efficient change point detection for high-dimensional regression. *arXiv preprint arXiv:1601.03704*, 2016.

Biao Liu, Carl D Morrison, Candace S Johnson, Donald L Trump, Maochun Qin, Jeffrey C Conroy, Jianmin Wang, and Song Liu. Computational methods for detecting copy number variations in cancer genome using next generation sequencing: principles and challenges. *Oncotarget*, 4(11):1868, 2013.

Po-Ling Loh and Martin J. Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *Ann. Statist.*, 40(3):1637–1664, 06 2012. doi: 10.1214/12-AOS1018. URL `https://doi.org/10.1214/12-AOS1018`.

Robert Lund, Xiaolan L Wang, Qi Qi Lu, Jaxk Reeves, Colin Gallagher, and Yang Feng. Changepoint detection in periodic and autocorrelated time series. *Journal of Climate*, 20 (20):5178–5190, 2007.

Andreas Maurer. A bound on the deviation probability for sums of non-negative random variables. *J. Inequalities in Pure and Applied Mathematics*, 4(1):15, 2003.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research*, 11(Aug):2241–2259, 2010.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over l_q-balls. *IEEE transactions on information theory*, 57 (10):6976–6994, 2011.

Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46(6):900–915, 2007.

Mark Rudelson and Shuheng Zhou. Reconstruction from anisotropic random measurements. In *Conference on Learning Theory*, pages 10–1, 2012.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Tengyao Wang and Richard J Samworth. High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):57–83, 2018.

Yuehua Wu. Simultaneous change point analysis and variable selection in a regression problem. *Journal of Multivariate Analysis*, 99(9):2154–2171, 2008.

Fei Ye and Cun-Hui Zhang. Rate minimaxity of the lasso and dantzig selector for the lq loss in lr balls. *Journal of Machine Learning Research*, 11(Dec):3519–3540, 2010.

Bingwen Zhang, Jun Geng, and Lifeng Lai. Multiple change-points estimation in linear regression models via sparse group lasso. *IEEE Trans. Signal Processing*, 63(9):2209–2224, 2015.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Journal of Machine Learning Research 20 (2019) 1-49 Submitted 11/18; Published 7/19

Measuring the Effects of Data Parallelism on Neural Network Training

Christopher J. Shallue* SHALLUE@GOOGLE.COM
Jaehoon Lee*† JAEHLEE@GOOGLE.COM
Joseph Antognini† JOE.ANTOGNINI@GMAIL.COM
Jascha Sohl-Dickstein JASCHASD@GOOGLE.COM
Roy Frostig FROSTIG@GOOGLE.COM
George E. Dahl GDAHL@GOOGLE.COM

Google Brain
1600 Amphiteatre Parkway
Mountain View, CA, 94043, USA

Editor: Rob Fergus

Abstract

Recent hardware developments have dramatically increased the scale of data parallelism available for neural network training. Among the simplest ways to harness next-generation hardware is to increase the batch size in standard mini-batch neural network training algorithms. In this work, we aim to experimentally characterize the effects of increasing the batch size on training time, as measured by the number of steps necessary to reach a goal out-of-sample error. We study how this relationship varies with the training algorithm, model, and data set, and find extremely large variation between workloads. Along the way, we show that disagreements in the literature on how batch size affects model quality can largely be explained by differences in metaparameter tuning and compute budgets at different batch sizes. We find no evidence that larger batch sizes degrade out-of-sample performance. Finally, we discuss the implications of our results on efforts to train neural networks much faster in the future. Our experimental data is publicly available as a database of 71,638,836 loss measurements taken over the course of training for 168,160 individual models across 35 workloads.

Keywords: neural networks, stochastic gradient descent, data parallelism, batch size, deep learning

1. Introduction

Neural networks have become highly effective at a wide variety of prediction tasks, including image classification, machine translation, and speech recognition. The dramatic improvements in predictive performance over the past decade have partly been driven by advances in hardware for neural network training, which have enabled larger models to be trained on larger datasets than ever before. However, although modern GPUs and custom

∗. Both authors contributed equally.
†. Work done as a member of the Google AI Residency program (g.co/airesidency).

©2019 Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-789.html.

accelerators have made training neural networks orders of magnitude faster, training time still limits both the predictive performance of these techniques and how widely they can be applied. For many important problems, the best models are still improving at the end of training because practitioners cannot afford to wait until the performance saturates. In extreme cases, training must end before completing a single pass over the data (e.g. Anil et al., 2018). Techniques that speed up neural network training can significantly benefit many important application areas. Faster training can facilitate dramatic improvements in model quality by allowing practitioners to train on more data (Hestness et al., 2017), and by decreasing the experiment iteration time, allowing researchers to try new ideas and configurations more rapidly. Faster training can also allow neural networks to be deployed in settings where models have to be updated frequently, for instance when new models have to be produced when training data get added or removed.

Data parallelism is a straightforward and popular way to accelerate neural network training. For our purposes, data parallelism refers to distributing training examples across multiple processors to compute gradient updates (or higher-order derivative information) and then aggregating these locally computed updates. As long as the training objective decomposes into a sum over training examples, data parallelism is model-agnostic and applicable to any neural network architecture. In contrast, the maximum degree of *model parallelism* (distributing parameters and computation across different processors for the same training examples) depends on the model size and structure. Although data parallelism can be simpler to implement, ultimately, large scale systems should consider all types of parallelism at their disposal. In this work, we focus on the costs and benefits of data parallelism in the synchronous training setting.

Hardware development is trending towards increasing capacity for data parallelism in neural network training. Specialized systems using GPUs or custom ASICs (e.g. Jouppi et al., 2017) combined with high-performance interconnect technology are unlocking unprecedented scales of data parallelism where the costs and benefits have not yet been well studied. On the one hand, if data parallelism can provide a significant speedup at the limits of today's systems, we should build much bigger systems. On the other hand, if additional data parallelism comes with minimal benefits or significant costs, we might consider designing systems to maximize serial execution speed, exploit other types of parallelism, or even prioritize separate design goals such as power use or cost.

There is considerable debate in the literature about the costs and benefits of data parallelism in neural network training and several papers take seemingly contradictory positions. Some authors contend that large-scale data parallelism is harmful in a variety of ways, while others contend that it is beneficial. The range of conjectures, suggestive empirical results, and folk knowledge seems to cover most of the available hypothesis space. Answering these questions definitively has only recently become important (as increasing amounts of data parallelism have become practical), so it is perhaps unsurprising that the literature remains equivocal, especially in the absence of sufficiently comprehensive experimental data.

In this work, we attempt to provide the most rigorous and extensive experimental study on the effects of data parallelism on neural network training to date. In order to achieve this goal, we consider realistic workloads up to the current limits of data parallelism. We try to avoid making assumptions about how the optimal metaparameters vary as a function of batch size. Finally, in order to guide future work, we consider any remaining limitations in

our methodology, and we discuss what we see as the most interesting unanswered questions that arise from our experiments.

1.1. Scope

We restrict our attention to variants of mini-batch stochastic gradient descent (SGD), which are the dominant algorithms for training neural networks. These algorithms iteratively update the model's parameters using an estimate of the gradient of the training objective. The gradient is estimated at each step using a different subset, or *(mini-) batch*, of training examples. See Section 2.2 for a more detailed description of these algorithms. A data-parallel implementation computes gradients for different training examples in each batch in parallel, and so, in the context of mini-batch SGD and its variants, we equate the batch size with the amount of data parallelism.[1] We restrict our attention to synchronous SGD because of its popularity and advantages over asynchronous SGD (Chen et al., 2016).

Practitioners are primarily concerned with out-of-sample error and the cost they pay to achieve that error. Cost can be measured in a variety of ways, including training time and hardware costs. Training time can be decomposed into number of steps multiplied by average time per step, and hardware cost into number of steps multiplied by average hardware cost per step. The per-step time and hardware costs depend on the practitioner's hardware, but the number of training steps is hardware-agnostic and can be used to compute the total costs for any hardware given its per-step costs. Furthermore, in an idealized data-parallel system where the communication overhead between processors is negligible, training time depends only on the number of training steps (and not the batch size) because the time per step is independent of the number of examples processed. Indeed, this scenario is realistic today in systems like TPU pods[2], where there are a range of batch sizes for which the time per step is almost constant. Since we are primarily concerned with training time, we focus on number of training steps as our main measure of training cost.

An alternative hardware-agnostic measure of training cost is the number of training examples processed, or equivalently the number of passes (*epochs*) over the training data. This measure is suitable when the per-step costs are proportional to the number of examples processed (e.g. hardware costs proportional to the number of floating point operations). However, the number of epochs is not a suitable measure of training time in a data-parallel system—it is possible to reduce training time by using a larger batch size and processing *more* epochs of training data, provided the number of training steps decreases.

In light of practitioners' primary concerns of out-of-sample error and the resources needed to achieve it, we believe the following questions are the most important to study to understand the costs and benefits of data parallelism with mini-batch SGD and its variants:

1. What is the relationship between batch size and number of training steps to reach a goal out-of-sample error?
2. What governs this relationship?
3. Do large batch sizes incur a cost in out-of-sample error?

1. Mini-batch SGD can be implemented in a variety of ways, including data-serially, but a data-parallel implementation is always possible given appropriate hardware.
2. `https://www.blog.google/products/google-cloud/google-cloud-offer-tpus-machine-learning/`.

1.2. Contributions of This Work

1. We show that the relationship between batch size and number of training steps to reach a goal out-of-sample error has the same characteristic form across six different families of neural network, three training algorithms, and seven data sets.

   Specifically, for each workload (model, training algorithm, and data set), increasing the batch size initially decreases the required number of training steps proportionally, but eventually there are diminishing returns until finally increasing the batch size no longer changes the required number of training steps. To the best of our knowledge, we are the first to experimentally validate this relationship across models, training algorithms, and data sets while independently tuning the learning rate, momentum, and learning rate schedule (where applicable) for each batch size. Unlike prior work that made strong assumptions about these metaparameters, our results reveal a universal relationship that holds across all workloads we considered, across different error goals, and when considering either training error or out-of-sample error.

2. We show that the maximum useful batch size varies significantly between workloads and depends on properties of the model, training algorithm, and data set. Specifically, we show that:

   (a) SGD with momentum (as well as Nesterov momentum) can make use of much larger batch sizes than plain SGD, suggesting future work to study the batch size scaling properties of other algorithms.

   (b) Some models allow training to scale to much larger batch sizes than others. We include experimental data on the relationship between various model properties and the maximum useful batch size, demonstrating that the relationship is not as simple as one might hope from previous work (e.g. wider models do *not* always scale better to larger batch sizes).

   (c) The effect of the data set on the maximum useful batch size tends to be smaller than the effects of the model and training algorithm, and does not depend on data set size in a consistent way.

3. We show that the optimal values of training metaparameters do not consistently follow any simple relationships with the batch size. In particular, popular learning rate heuristics—such as linearly scaling the learning rate with the batch size— do not hold across all problems or across all batch sizes.

4. Finally, by reviewing the specifics of the experimental protocols used in prior work, we at least partially reconcile conflicting stances in the literature on whether increasing the batch size degrades model quality. Specifically, we show that assumptions about computational budgets and the procedures for selecting metaparameters at different batch sizes can explain many of the disagreements in the literature. We find no evidence that increasing the batch size necessarily degrades model quality, but additional regularization techniques may become important at larger batch sizes.

1.3. Experimental Data

We release our raw experimental data for any further analysis by the research community.[3] Our database contains 454 combinations of workload (model, data set, training algorithm) and batch size, each of which is associated with a metaparameter search space and a set of models trained with different configurations sampled from the search space. In total, our data contains 71,638,836 loss measurements taken over the course of training for 168,160 individual models. Together, these measurements make up the training curves of all of the individual models we trained, and can be used to reproduce all plots in this paper.[4]

2. Setup and Background

In this section we set up the basic definitions and background concepts used throughout the paper.

2.1. Learning

A *data distribution* is a probability distribution $\mathcal{D}$ over a data domain $\mathcal{Z}$. For example, we might consider a supervised learning task over a domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the set of 32-by-32-pixel color images and $\mathcal{Y}$ is the set of possible labels denoting what appears in the image. A *training set* $z_1, \ldots, z_n \in \mathcal{Z}$ is a collection of *examples* from the data domain, conventionally assumed to be drawn i.i.d. from the data distribution $\mathcal{D}$.

A machine learning *model* is a function that, given *parameters* θ from some set $\Theta \subset \mathbb{R}^d$, and given a data point $z \in \mathcal{Z}$, produces a prediction whose quality is measured by a differentiable non-negative scalar-valued loss function.[5] We denote by $\ell(\theta; z)$ the loss of a prediction made by the model, under parameters θ, on the data point z. We denote by L the *out-of-sample loss* or *expected loss*:

$$L(\theta) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}} \left[\ell(\theta; z) \right], \tag{1}$$

and by $\hat{L}$ the *empirical average loss* under a data set $S = (z_1, \ldots, z_n)$:

$$\hat{L}(\theta; S) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta; z_i). \tag{2}$$

When S is the training set, we call $\hat{L}$ the *average training loss*. We will say that the data source $\mathcal{D}$, loss ℓ, and model with parameter set Θ together specify a learning *task*, in which our aim is to find parameters θ that achieve low out-of-sample loss (Equation 1), while given access only to n training examples. A common approach is to find parameters of low average training loss (Equation 2) as an estimate of the out-of-sample loss (Shalev-Shwartz and Ben-David, 2014).

When minimizing average training loss $\hat{L}$, it is common to add regularization penalties to the objective function. For a differentiable penalty $R : \Theta \to \mathbb{R}_+$, regularization weight

3. `https://github.com/google-research/google-research/tree/master/batch_science`
4. `https://colab.research.google.com/github/google-research/google-research/blob/master/batch_science/reproduce_paper_plots.ipynb`
5. Technically, the loss need only be sub-differentiable. Extending our setup to this end is straightforward.

$\lambda > 0$, and training set S, the training objective might be

$$J(\theta) = \hat{L}(\theta; S) + \lambda R(\theta). \tag{3}$$

In practice, we often approach a task by replacing its loss with another that is more amenable to training. For instance, in supervised classification, we might be tasked with learning under the 0/1 loss, which is an indicator of whether a prediction is correct (e.g. matches a ground-truth label), but we train by considering instead a surrogate loss (e.g. the logistic loss) that is more amenable to continuous optimization. When the surrogate loss bounds the original, achieving low loss under the surrogate implies low loss under the original. To distinguish the two, we say *error* to describe the original loss (e.g. 0/1), and we save *loss* to refer to the surrogate used in training.

2.2. Algorithms

The dominant algorithms for training neural networks are based on *mini-batch stochastic gradient descent* (SGD, Robbins and Monro, 1951; Kiefer et al., 1952; Rumelhart et al., 1986; Bottou and Bousquet, 2008; LeCun et al., 2015). Given an initial point $\theta_0 \in \Theta$, mini-batch SGD attempts to decrease the objective J via the sequence of iterates

$$\theta_t \leftarrow \theta_{t-1} - \eta_t g(\theta_{t-1}; B_t),$$

where each B_t is a random subset of training examples, the sequence $\{\eta_t\}$ of positive scalars is called the *learning rate*, and where, for any $\theta \in \Theta$ and $B \subset S$,

$$g(\theta; B) = \frac{1}{|B|} \sum_{z \in B} \nabla \ell(\theta; z) + \lambda \nabla R(\theta). \tag{4}$$

When the examples B are a uniformly random subset of training examples, $g(\theta; B)$ forms an unbiased estimate of the gradient of the objective J that we call a *stochastic gradient*. In our larger-scale experiments, when we sample subsequent batches B_t, we actually follow the common practice of cycling through permutations of the training set (Shamir, 2016). The result of mini-batch SGD can be any of the iterates θ_t for which we estimate that $L(\theta_t)$ is low using a validation data set.

Variants of SGD commonly used with neural networks include SGD with momentum (Polyak, 1964; Rumelhart et al., 1986; Sutskever et al., 2013), Nesterov momentum (Nesterov, 1983; Sutskever et al., 2013), RMSProp (Hinton et al., 2012), and Adam (Kingma and Ba, 2015). All of these optimization procedures, or *optimizers*, interact with the training examples only by repeatedly computing stochastic gradients (Equation 4), so they support the same notion of batch size that we equate with the scale of data parallelism. In this work, we focus on the SGD, SGD with momentum, and Nesterov momentum optimizers. The latter two optimizers are configured by a learning rate $\{\eta_t\}$ and a scalar $\gamma \in (0, 1)$ that we call *momentum*. They define the iterates[6]

SGD with momentum	Nesterov momentum
$v_{t+1} \leftarrow \gamma v_t + g(\theta_t; B_t)$	$v_{t+1} \leftarrow \gamma v_t + g(\theta_t; B_t)$
$\theta_{t+1} \leftarrow \theta_t - \eta_t v_{t+1}$	$\theta_{t+1} \leftarrow \theta_t - \eta_t g(\theta_t; B_t) - \eta_t \gamma v_{t+1},$

6. These rules take slightly different forms across the literature and across library implementations. We present and use the update rules from the `MomentumOptimizer` class in TensorFlow (Abadi et al., 2016).

given $v_0 = 0$ and an initial θ_0. Note that plain SGD can be recovered from either optimizer by taking $\gamma = 0$. The outcome of using these optimizers should therefore be no worse if than SGD, in any experiment, the momentum γ is tuned across values including zero.

If we run SGD with momentum under a constant learning rate $\eta_t = \eta$, then, at a given iteration t, the algorithm computes

$$\theta_{t+1} = \theta_t - \eta v_{t+1} = \theta_0 - \eta \sum_{u=0}^{t} v_{u+1} = \theta_0 - \eta \sum_{u=0}^{t} \sum_{s=0}^{u} \gamma^{u-s} g(\theta_s; B_s).$$

For any fixed $\tau \in \{0, \ldots, t\}$, the coefficient accompanying the stochastic gradient $g(\theta_\tau; B_\tau)$ in the above update is $\eta \sum_{u=\tau}^{t} \gamma^{u-\tau}$. We define the *effective learning rate*, η^{eff} as the value of this coefficient at the end of training ($t = T$), in the limit of a large number of training steps ($T \to \infty$, while τ is held fixed):

$$\eta^{\text{eff}} = \lim_{T \to \infty} \sum_{u=\tau}^{T} \eta \gamma^{u-\tau} = \frac{\eta}{1-\gamma}.$$

Put intuitively, η^{eff} captures the contribution of a given mini-batch gradient to the parameter values at the end of training.

2.3. Additional Terminology in Experiments

A *data-parallel implementation* of mini-batch SGD (or one of its variants) computes the summands of Equation 4 in parallel and then synchronizes to coordinate their summation.

The models and algorithms in our experiments are modifiable by what we call *meta-parameters*.[7] These include architectural choices, such as the number of layers in a neural network, and training parameters, such as learning rates $\{\eta_t\}$ and regularization weights λ. When we use the term *model*, we typically assume that all architectural metaparameters have been set. In our experiments, we *tune* the training metaparameters by selecting the values that yield the best performance on a validation set. We use the term *workload* to jointly refer to a data set, model, and training algorithm.

3. Related Work

In this section we review prior work related to our three main questions from Section 1.1. First we review studies that considered the relationship between batch size and number of training steps (Questions 1 and 2), and then we review studies that considered the effects of batch size on solution quality (Question 3).

3.1. Steps to Reach a Desired Out-Of-Sample Error

We broadly categorize the related work on this topic as either analytical or empirical in nature.

7. Sometimes called "hyperparameters," but we prefer a different name so as not to clash with the notion of hyperparameters in Bayesian statistics.

3.1.1. Analytical Studies

Convergence upper bounds from the theory of stochastic (convex) optimization can be specialized to involve terms dependent on batch size, so in this sense they comprise basic related work. These upper bounds arise from worst-case analysis, and moreover make convexity and regularity assumptions that are technically violated in neural network training, so whether they predict the actual observed behavior of our experimental workloads is an empirical question in its own right.

Given a sequence of examples drawn i.i.d. from a data source, an upper bound on the performance of SGD applied to L-Lipschitz convex losses is (Hazan, 2016; Shalev-Shwartz and Ben-David, 2014)

$$J(\theta_T) - J^\star \leq O\left(\sqrt{\frac{L^2}{T}}\right), \tag{5}$$

for any batch size. Here, J is the objective function, $J^\star$ is its value at the global optimum, and θ_T denotes the final output of the algorithm supposing it took T iterations.[8] Meanwhile, when losses are convex and the objective is H-smooth, accelerated parallel mini-batch SGD enjoys the bound (Lan, 2012)

$$J(\theta_T) - J^\star \leq O\left(\frac{H}{T^2} + \sqrt{\frac{L^2}{Tb}}\right), \tag{6}$$

where b is the batch size.

Compared to sequential processing without batching (i.e. a batch size of one), the bounds Equation 5 and Equation 6 offer two extremes, respectively:

1. **No benefit:** Increasing the batch size b does not change the number of steps to convergence, as per Equation 5.

2. **A b-fold benefit:** The term in Equation 6 proportional to $1/\sqrt{Tb}$ dominates the bound. Increasing the batch size b by a multiplicative factor decreases the number of steps T to a given achievable objective value by the same factor.

In other words, under these simplifications, batching cannot hurt the asymptotic guarantees of steps to convergence, but it could be wasteful of examples. The two extremes imply radically different guidance for practitioners, so the critical task of establishing a relationship between batch size and number of training steps remains one to resolve experimentally.

A few recent papers proposed analytical notions of a critical batch size: a point at which a transition occurs from a b-fold benefit to no benefit. Under assumptions including convexity, Ma et al. (2018) derived such a critical batch size, and argued that a batch size of one is optimal for minimizing the number of training epochs required to reach a given target error. Under different assumptions, Yin et al. (2018) established a critical batch size and a pathological loss function that together exhibit a transition from a b-fold benefit to no benefit. Although they ran experiments with neural networks, their experiments were designed to investigate the effect of data redundancy and do not provide enough

8. Not necessarily the Tth iterate, which may differ from θ_T if the algorithm averages its iterates.

information to reveal the empirical relationship between batch size and number of training steps. Focusing on linear least-squares regression, Jain et al. (2018) also derived a threshold batch size in terms of the operator norm of the objective's Hessian and a constant from a fourth-moment bound on example inputs.

To our knowledge, in all previous work that analytically derived a critical batch size, the thresholds defined are either (i) parameter-dependent, or (ii) specific to linear least-squares regression. A critical batch size that depends on model parameters can change over the course of optimization; it is not a problem-wide threshold that can be estimated efficiently *a priori*. Focusing on least-squares has issues as well: while it sheds intuitive light on how batching affects stochastic optimization locally, the quantities defined inherently cannot generalize to the non-linear optimization setting of neural network training, both because the objective's Hessian is not constant across the space of parameters as it is in a quadratic problem, and more broadly because it is unclear whether the Hessian of the objective is still the correct analogue to consider.

3.1.2. Empirical Studies

Wilson and Martinez (2003) investigated the relationship between batch size and training speed for plain mini-batch SGD. They found that a simple fully connected neural network took more epochs to converge with larger batch sizes on a data set of 20,000 examples, and also that using a batch size equal to the size of the training set took more epochs to converge than a batch size of one on several small data sets of size ≤ 600. However, their experimental protocol and assumptions limit the conclusions we can draw from their results. One issue is that training time was measured to different out-of-sample errors for different batch sizes on the same data set. To compare training speed fairly, the error goal should be fixed across all training runs being compared. Additionally, only four learning rates were tried for each data set, but quite often the best learning rate was at one of the two extremes and it appeared that a better learning rate might be found outside of the four possibilities allowed. Finally, despite the contention of the authors, their results do not imply slower training with larger batch sizes in data-parallel training: for the most part, their larger batch size experiments took fewer training steps than the corresponding batch size one experiments.

In the last few years, increasingly specialized computing systems have spurred practitioners to try much larger batch sizes than ever before, while increasingly promising results have driven hardware designers to create systems capable of even more data parallelism. Chen et al. (2016) used a pool of synchronized worker machines to increase the effective batch size of mini-batch SGD. They demonstrated speedups in both wall time and steps to convergence for an Inception model (Szegedy et al., 2016) on ImageNet (Russakovsky et al., 2015) by scaling the effective batch size from 1,600 to 6,400. More recently, Goyal et al. (2017) showed that the number of training epochs could be held constant across a range of batch sizes to achieve the same validation error for ResNet-50 (He et al., 2016a) on ImageNet. Holding the number of training epochs constant is equivalent to scaling the number of training steps inversely with the batch size, and this reduction in training steps with increasing batch size produced nearly proportional wall time speedups on their hardware. Although this hints at a b-fold benefit regime in which increasing the batch size reduces the

number of training steps by the same factor, the authors did not attempt to minimize the number of training steps (or epochs) required to reach the goal at each batch size separately. It is unclear whether any of the batch sizes that achieved the goal could do so in fewer steps than given, or how many steps the other batch sizes would have needed to achieve the same error goal.

Two studies performed concurrently with this work also investigated the relationship between batch size and training speed for neural networks. Chen et al. (2018) provide experimental evidence of a problem-dependent critical batch size after which a b-fold benefit is no longer achieved for plain mini-batch SGD. They contend that wider and shallower networks have larger critical batch sizes, and while their empirical results are equivocal for this particular claim, they show that the threshold batch size can depend on aspects of both the data set and the model. Additionally, Golmant et al. (2018) studied how three previously proposed heuristics for adjusting the learning rate as a function of batch size (linear scaling, square root scaling, and no scaling) affect the number of training steps required to reach a particular result. They found that if the learning rate is tuned for the the smallest batch size only, all three of these common scaling techniques break down for larger batch sizes and result in either (i) divergent training, or (ii) training that cannot reach the error goal within a fixed number of training epochs. They also describe a basic relationship between batch size and training steps to a fixed error goal, which is comprised of three regions: b-fold benefit initially, then diminishing returns, and finally no benefit for all batch sizes greater than a maximum useful batch size. However, their results are inconclusive because (i) not all model and data set pairs exhibit this basic relationship, (ii) it does not appear consistently across error goals, and (iii) the relationship is primarily evident in training error but not out-of-sample error. These inconsistent results may be due to suboptimal pre-determined learning rates arising from the scaling rules, especially at larger batch sizes. Finally, they also found that the maximum useful batch size depends on aspects of the model and the data set type, but not on the data set size. Since all their experiments use plain mini-batch SGD, their results are unable to reveal any effects from the choice of optimizer and might not generalize to other popular optimizers, such as SGD with momentum.

3.2. Solution Quality

The literature contains some seemingly conflicting claims about the effects of batch size on solution quality (out-of-sample error at the conclusion of training). Primarily, the debate centers on whether increasing the batch size incurs a cost in solution quality. Keskar et al. (2017) argue that large batch[9] training converges to so-called "sharp" minima with worse generalization properties. However, Dinh et al. (2017) show that a minimum with favorable generalization properties can be made, through reparameterization, arbitrarily sharp in the same sense. Le Cun et al. (1998) suggest that a batch size of one can result in better solutions because the noisier updates allow for the possibility of escaping from local minima in a descent algorithm. However, they also note that we usually stop training long before

9. The term "large batch" is inherently ambiguous, and in this case accompanies experiments in Keskar et al. (2017) that only compare two absolute batch sizes per data set, rather than charting out a curve to its apparent extremes.

reaching any sort of critical point. Hoffer et al. (2017) argue that increasing the batch size need not degrade out-of-sample error at all, assuming training has gone on long enough. Goyal et al. (2017), among others, tested batch sizes larger than those used in Keskar et al. (2017) without noticing any reduction in solution quality. Still, their results with yet larger batch sizes do not rule out the existence of a more sudden degradation once the batch size is large enough. Meanwhile, Goodfellow et al. (2016) state that small batches can provide a regularization effect such that they result in the best observed out-of-sample error, although in this case other regularization techniques might serve equally well.

Alas, the best possible out-of-sample error for a particular model and data set cannot be measured unconditionally due to practical limits on wall time and hardware resources, as well as practical limits on our ability to tune optimization metaparameters (e.g. the learning rate). An empirical study can only hope to measure solution quality subject to the budgets allowed for each model experiment, potentially with caveats due to limitations of the specific procedures for selecting the metaparameters. To the best of our knowledge, all published results handle the training budget issue in exactly one of three ways: by ignoring budgets (train to convergence, which is not always possible); by using a step budget (restrict the number of gradient descent updates performed); or by using an epoch budget (restrict number of training examples processed).[10] Furthermore, while some published results tune the learning rate anew for each batch size, others tune for only a single batch size and use a preordained heuristic to set the learning rate for the remaining batch sizes (the most common heuristics are constant, square root, and linear learning rate scaling rules). Tuning metaparameters at a single batch size and then heuristically adjusting them for others could clearly create a systematic advantage for trials at batch sizes near to the one tuned. All in all, the conclusions we can draw from previous studies depend on the budgets they assume and on how they select metaparameters across batch sizes. The following subsections attempt an investigation of their experimental procedures to this end.

3.2.1. Studies That Ignore Budgets

All studies in this section compared solution quality for different batch sizes after deeming their models to have converged. They determined training stopping time by using either manual inspection, convergence heuristics, or fixed compute budgets that they considered large enough to guarantee convergence.[11]

Keskar et al. (2017) trained several neural network architectures on MNIST and CIFAR-10, each with two batch sizes, using the Adam optimizer and without changing the learning rate between batch sizes. They found that the larger batch size consistently achieved worse out-of-sample error after training error had ceased to improve. However, all models used batch normalization (Ioffe and Szegedy, 2015) and presumably computed the batch normalization statistics using the full batch size. For a fair comparison between batch sizes, batch normalization statistics should be computed over the same number of examples or else the training objective differs between batch sizes (Goyal et al., 2017). Indeed, Hoffer et al. (2017) found that computing batch normalization statistics over larger batches can degrade solution quality, which suggests an alternative explanation for the results of Keskar et al. (2017). Moreover, Keskar et al. (2017) reported that data augmentation eliminated the difference in solution quality between small and large batch experiments.

Smith and Le (2018) trained a small neural network on just 1,000 examples sampled from MNIST with two different batch sizes, using SGD with momentum and without changing the learning rate between batch sizes. They observed that the larger batch size overfit more than the small batch size resulting in worse out-of-sample error, but this gap was mitigated by applying L_2 regularization (Smith and Le, 2018, figures 3 and 8). They also compared a wider range of batch sizes in experiments that either (i) used a step budget without changing the learning rate for each batch size (Smith and Le, 2018, figures 4 and 6), or (ii) varied the learning rate and used a step budget that was a function of the learning rate (Smith and Le, 2018, figure 5). Instead, we focus on the case where the learning rate and batch size are chosen independently.

Breuel (2015a,b) trained a variety of neural network architectures on MNIST with a range of batch sizes, using the SGD and SGD with momentum optimizers with a range of learning rates and momentum values. They found that batch size had no effect on solution quality for LSTM networks (Breuel, 2015a), but found that larger batch sizes achieved worse solutions for fully connected and convolutional networks, and that the scale of the effect depended on the activation function in the hidden and output layers (Breuel, 2015b).

Finally, Chen et al. (2016) observed no difference in solution quality when scaling the batch size from 1,600 to 6,400 for an Inception model on ImageNet when using the RMSProp optimizer and a heuristic to set the learning rate for each batch size.

3.2.2. Studies with Step Budgets

Hoffer et al. (2017) trained neural networks with two different batch sizes on several image data sets. They found that, by computing batch normalization statistics over a fixed number of examples per iteration ("ghost batch normalization"), and by scaling the learning rate with the square root of the batch size instead of some other heuristic, the solution quality arising from the larger batch size was as good as or better than the smaller batch size. However, the largest batch size used was 4,096, which does not rule out an effect appearing at still larger batch sizes, as suggested by the work of Goyal et al. (2017). Moreover, it remains open whether their proposed learning rate heuristic extends to arbitrarily large batch sizes, or whether it eventually breaks down for batch sizes sufficiently far from the base batch size.

3.2.3. Studies with Epoch Budgets

An epoch budget corresponds to fixing the total number of per-example gradient computations, but, in an idealized data-parallel implementation of SGD, it also corresponds to a step (or even wall time) budget that scales inversely with the batch size. With an epoch budget, a larger batch size can only achieve the same solution quality as a smaller batch

10. Of course, there are budgets in between an epoch budget and a step budget that might allow the possibility of trading off time, computation, and/or solution quality. For example, it may be possible to increase the number of training epochs and still take fewer steps to reach the same quality solution. However, we are not aware of work that emphasizes these budgets.
11. As discussed further in Section 4.8, we find that millions of training steps for small batch sizes, or thousands of epochs for large batch sizes, are required to saturate performance even for data sets as small and simple as MNIST. In our experiments, this corresponded to more than 25 hours of wall-time for each metaparameter configuration.

size if it achieves perfect scaling efficiency (a b-fold reduction in steps from increasing the batch size, as described in Section 3.1.1).

Masters and Luschi (2018) show that after a critical batch size depending on the model and data set, solution quality degrades with increasing batch size *when using a fixed epoch budget*. Their results effectively show a limited region of b-fold benefit for those model and data set pairs when trained with SGD, although they did not investigate whether this critical batch size depends on the optimizer used, and they did not consider more than one epoch budget for each problem. We reproduced a subset of their experiments and discuss them in Section 5.

Goyal et al. (2017) recently popularized a linear learning rate scaling heuristic for training the ResNet-50 model using different batch sizes. Using this heuristic, a 90 epoch budget, and SGD with momentum without adjusting or tuning the momentum, they increased the batch size from 64 to 8,192 with no loss in accuracy. However, their learning rate heuristic broke down for even larger batch sizes. Inspired by these results, a sequence of follow-up studies applied additional techniques to further increase the batch size while still achieving the same accuracy and using the same 90 epoch budget. These follow-on studies (Codreanu et al., 2017; You et al., 2017; Akiba et al., 2017) confirm that the best solution quality for a given batch size will also depend on the exact optimization techniques used.

There are several additional papers (Lin et al., 2018; Devarakonda et al., 2017; Golmant et al., 2018) with experiments relevant to solution quality that used an epoch budget, tuned the learning rate for the smallest batch size, and then used a heuristic to choose the learning rate for all larger batch sizes. For instance, Devarakonda et al. (2017) and Lin et al. (2018) used linear learning rate scaling and Golmant et al. (2018) tried constant, square root, and linear learning rate scaling heuristics. All of them concluded that small batch sizes have superior solution quality to large batch sizes with a fixed epoch budget, for various notions of "small" and "large." This could just as easily be an artifact of the learning rate heuristics, and a possible alternative conclusion is that these heuristics are limited (as heuristics often are).

4. Experiments and Results

The primary quantity we measure is the number of steps needed to first reach a desired out-of-sample error, or *steps to result*. To measure steps to result, we used seven image and text data sets with training set sizes ranging from 45,000 to 26 billion examples. Table 1 summarizes these data sets and Appendix A provides the full details. We chose six families of neural network to train on these data sets. For MNIST and Fashion MNIST, we chose a simple fully connected neural network and a simple convolutional neural network (CNN). For CIFAR-10, we chose the ResNet-8 model without batch normalization, partly to compare our results to Masters and Luschi (2018), and partly to have a version of ResNet without batch normalization. For ImageNet, we chose ResNet-50, which uses batch normalization and residual connections, and VGG-11, which uses neither. For Open Images, we chose ResNet-50. For LM1B, we chose the Transformer model and an LSTM model. For Common Crawl, we chose the Transformer model. Table 2 summarizes these models and Appendix B provides the full details.

Data Set	Type	Task	Size	Evaluation Metric
MNIST	Image	Classification	55,000	Classification error
Fashion MNIST	Image	Classification	55,000	Classification error
CIFAR-10	Image	Classification	45,000	Classification error
ImageNet	Image	Classification	1,281,167	Classification error
Open Images	Image	Classification (multi-label)	4,526,492	Average precision
LM1B	Text	Language modeling	30,301,028	Cross entropy error
Common Crawl	Text	Language modeling	∼25.8 billion	Cross entropy error

Table 1: Summary of data sets. Size refers to the number of examples in the training set, which we measure in sentences for text data sets. See Appendix A for full details.

Model Class	Sizes	Optimizers	Data Sets	Learning rate schedule
Fully Connected	Various	SGD	MNIST	Constant
Simple CNN	Base Narrow Wide	SGD Momentum Nesterov mom.	MNIST Fashion MNIST	Constant
ResNet	ResNet-8	SGD Nesterov mom.	CIFAR-10	Linear decay
	ResNet-50	Nesterov mom.	ImageNet Open Images	Linear decay
VGG	VGG-11	Nesterov mom.	ImageNet	Linear decay
Transformer	Base Narrow and shallow Shallow Wide	SGD Momentum Nesterov mom.	LM1B Common crawl	Constant
LSTM	—	Nesterov mom.	LM1B	Constant

Table 2: Summary of models. See Appendix B for full details.

Measuring steps to result requires a particular value of out-of-sample error to be chosen as the goal. Ideally, we would select the best achievable error for each task and model, but since validation error is noisy, the best error is sometimes obtained unreliably. Moreover, for some workloads, the validation error continues to improve steadily beyond the maximum practical training time. Therefore, we generally tried to select the best validation error that we could achieve reliably within a practical training time.

Table 2 also shows the learning rate schedule we used for each model and data set. Learning rate schedules are often used to accelerate neural network training, but finding the best schedule is an optimization problem in its own right (Wu et al., 2018). Instead, researchers typically choose from a range of common learning rate functions based on validation performance and individual preference. While most schedules decay the learning rate monotonically over training, some researchers also "warm-up" the learning rate at the start of training (e.g. He et al., 2016a), particularly when training with large batch sizes (Goyal et al., 2017). We ran experiments with both constant learning rates and with learning rate decay. We used decay for ResNet-8, ResNet-50, and VGG-11, which significantly reduced

training time for those models. We selected our decay function by running an extensive set of experiments with ResNet-50 on ImageNet (see Appendix C for details). We chose linear decay because it performed at least as well as all other schedules we tried, while also being the simplest and requiring only two additional metaparameters. In experiments that used linear decay, we specified metaparameters (η_0, α, T) such that the learning rate decayed linearly from η_0 to $\eta_T = \alpha\eta_0$. That is, the learning rate at step t is given by

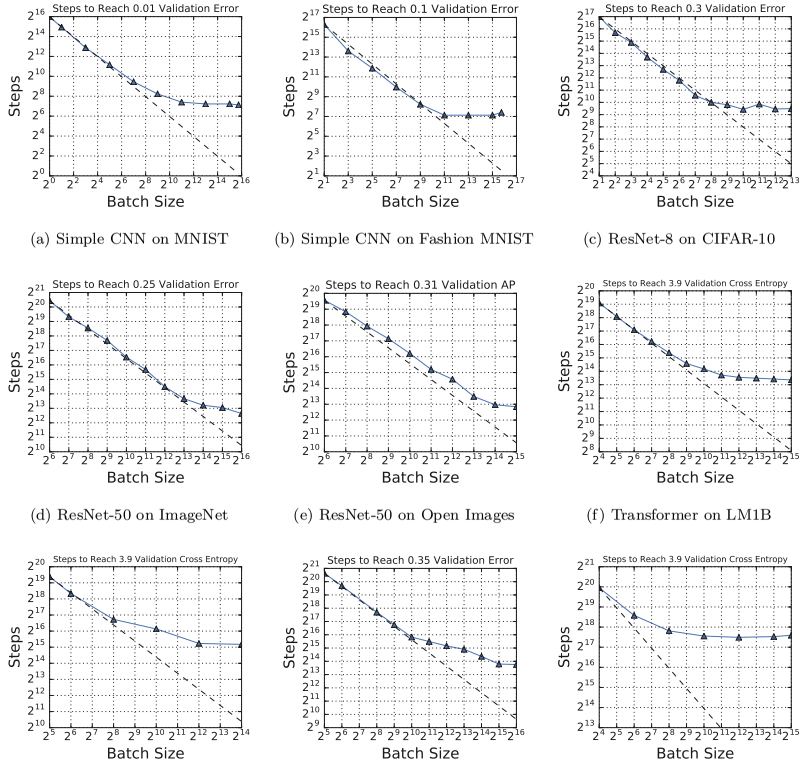
$$\eta_t = \begin{cases} \eta_0 - (1-\alpha)\eta_0 \frac{t}{T} & \text{if } t \leq T, \\ \alpha\eta_0 & \text{if } t > T. \end{cases}$$

Steps to result depends on the training metaparameters, and, for a given task and model, each batch size might have a different metaparameter configuration that minimizes steps to result. In all experiments, we independently tuned the metaparameters at each batch size, including the initial learning rate η_0 and, when learning rate decay was used, the decay schedule (α, T). Also, unless otherwise specified, we used the Nesterov momentum optimizer (Sutskever et al., 2013) and tuned the momentum γ.[12] Tuning anew for each batch size is extremely important since otherwise we would not be measuring steps to result as a function of batch size, rather we would be measuring steps to result as a function of batch size and the specific values of the learning rate and other metaparameters. We used quasi-random search (Bousquet et al., 2017) to tune the metaparameters with equal budgets of non-divergent[13] trials for different batch sizes. We selected metaparameter search spaces by hand based on preliminary experiments. The exact number of non-divergent trials needed to produce stable results depends on the search space, but 100 trials seemed to suffice in our experiments.[14] If the optimal trial occurred near the boundary of the search space, or if the goal validation error was not achieved within the search space, we repeated the search with a new search space. We measured steps to result for each batch size by selecting the metaparameter trial that reached the goal validation error in the fewest number of steps.

4.1. Steps to Result Depends on Batch Size in a Similar Way Across Problems

To get a sense of the basic empirical relationship, we measured the number of steps required to reach a goal validation error as a function of batch size across several different data sets and models (Figure 1). In all cases, as the batch size grows, there is an initial period of **perfect scaling** (b-fold benefit, indicated with a dashed line on the plots) where the steps needed to achieve the error goal halves for each doubling of the batch size. However, for all problems, this is followed by a region of **diminishing returns** that eventually leads to a regime of **maximal data parallelism** where additional parallelism provides no benefit whatsoever. In other words, for any given problem and without making strong assumptions about learning rates or other optimizer parameters, we can achieve both extremes suggested by theory (see Section 3.1.1). *A priori*, it is not obvious that every workload in our experiments should exhibit perfect scaling at the smallest batch sizes instead of immediately showing diminishing returns.

12. For LSTM for LM1B, we used a fixed value of $\gamma = 0.99$. We chose this value based on initial experiments and validated that tuning γ did not significantly affect the results for batch sizes 256, 1,024, or 4,096.
13. We discarded trials with a divergent training loss, which occurred when the learning rate was too high.
14. We used 100 non-divergent trials for all experiments except Transformer Shallow on LM1B with SGD, Transformer on Common Crawl, and LSTM on LM1B, for which we used 50 trials each.

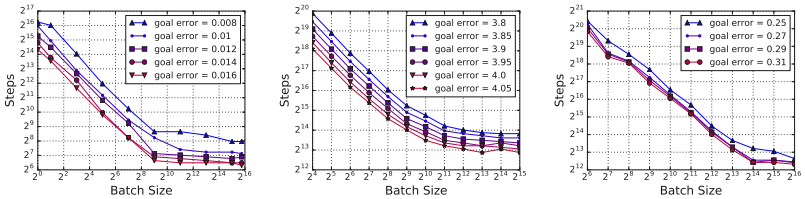


(a) Simple CNN on MNIST (b) Simple CNN on Fashion MNIST (c) ResNet-8 on CIFAR-10

(d) ResNet-50 on ImageNet (e) ResNet-50 on Open Images (f) Transformer on LM1B

(g) Transformer on Common Crawl (h) VGG-11 on ImageNet (i) LSTM on LM1B

Figure 1: **The relationship between steps to result and batch size has the same characteristic form for all problems.** In all cases, as the batch size grows, there is an initial period of **perfect scaling** (indicated with a dashed line) where the steps needed to achieve the error goal halves for each doubling of the batch size. Then there is a region of **diminishing returns** that eventually leads to a region of **maximal data parallelism** where additional parallelism provides no benefit whatsoever. AP denotes average precision (see Appendix A).



(a) Simple CNN on MNIST (b) Transformer on LM1B (c) ResNet-50 on ImageNet

Figure 2: **Steps-to-result plots have a similar form for different (nearby) performance goals.** The transition points between the three regions (perfect scaling, diminishing returns, and maximal data parallelism) are nearly the same.

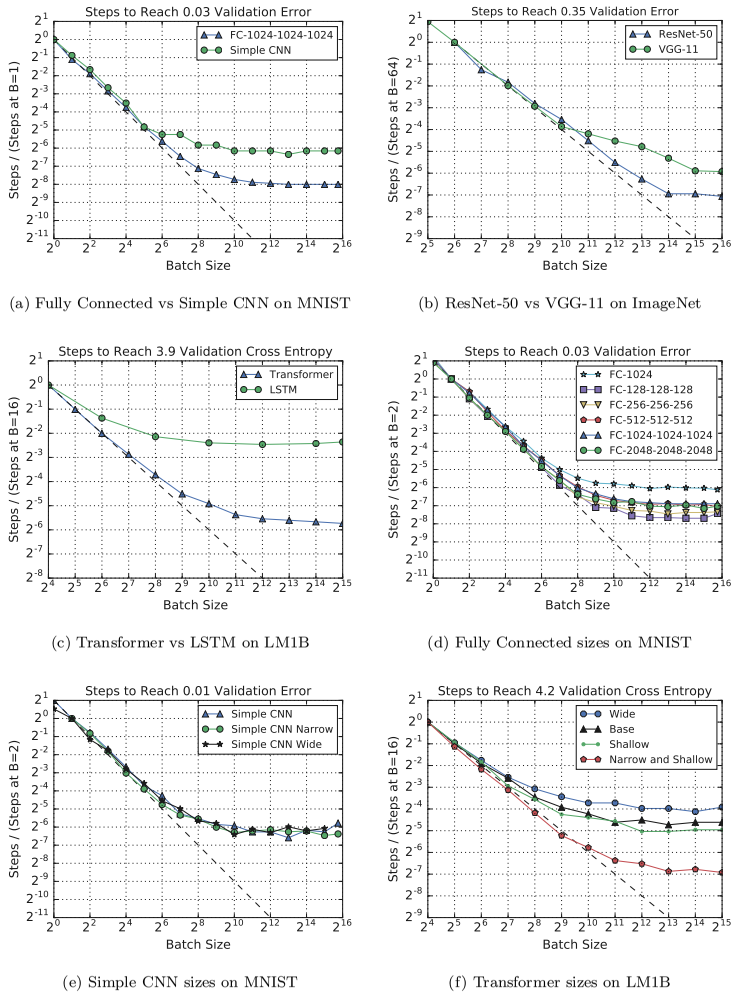
4.2. Validating Our Measurement Protocol

If the curves in Figure 1 were very sensitive to the goal validation error, then measuring the steps needed to reach our particular choice of the goal would not be a meaningful proxy for training speed. For small changes in the goal validation error, we do not care about vertical shifts as long as the transition points between the three scaling regions remain relatively unchanged. Figure 2 shows that varying the error goal only vertically shifts the steps-to-result curve, at least for modest variations centered around a good absolute validation error. Furthermore, although we ultimately care about out-of-sample error, if our plots looked very different when measuring the steps needed to reach a particular *training* error, then we would need to include both curves when presenting our results. However, switching to training error does not change the plots much at all (see Figure 12 in the Appendix).

Our experiments depend on extensive metaparameter tuning for the learning rate, momentum, and, where applicable, the learning rate schedule. For each experiment, we verified our metaparameter search space by checking that the optimal trial was not too close to a boundary of the space. See Figures 13 and 14 in the Appendix for examples of how we verified our search spaces.

4.3. Some Models Can Exploit Much Larger Batch Sizes Than Others

We investigated whether some models can make more use of larger batches than others by experimenting with different models while keeping the data set and optimizer fixed. We explored this question in two ways: (i) by testing completely different model architectures on the same data set, and (ii) by varying the size (width and depth) of a model within a particular model family. Since the absolute number of steps needed to reach a goal validation error depends on the model, the steps to result vs. batch size curves for each model generally appear at different vertical offsets from each other. Since we primarily care about the locations of the perfect scaling, diminishing returns, and maximal data parallelism regions, we normalized the y-axis of each plot by dividing by the number of steps needed to reach the goal for a particular batch size and data set. This normalization corresponds to a vertical shift of each curve (on log-scale plots), and makes it easier to compare different models. Appendix D contains all plots in this section without the y-axis normalized.



(a) Fully Connected vs Simple CNN on MNIST (b) ResNet-50 vs VGG-11 on ImageNet

(c) Transformer vs LSTM on LM1B (d) Fully Connected sizes on MNIST

(e) Simple CNN sizes on MNIST (f) Transformer sizes on LM1B

Figure 3: **Some models can exploit much larger batch sizes than others.** Figures 3a-3c show that some model architectures can exploit much larger batch sizes than others on the same data set. Figures 3d-3f show that varying the depth and width can affect a model's ability to exploit larger batches, but not necessarily in a consistent way across different model architectures. All MNIST models in this Figure used plain mini-batch SGD, while all other models used Nesterov momentum. The goal validation error for each plot was chosen to allow all model variants to achieve that error. Figure 15 in the Appendix contains these plots without the y-axis normalized.

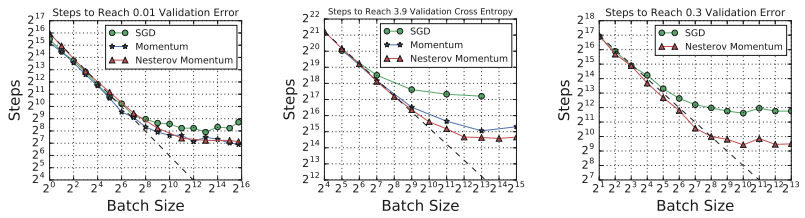
Figures 3a–3c show that the model architecture significantly affects the relationship between batch size and the number of steps needed to reach a goal validation error. In Figure 3a, the curve for the Fully Connected model flattens later than for the Simple CNN model on MNIST (although in this case the Simple CNN model can ultimately achieve better performance than the Fully Connected model). In Figure 3b, the curve for ResNet-50 flattens much later than the curve for VGG-11, indicating that ResNet-50 can make better use of large batch sizes on this data set. Unlike ResNet-50, VGG-11 does not use batch normalization or residual connections. Figure 3c shows that Transformer can make better use of large batch sizes than LSTM on LM1B.

Figures 3d–3f show that varying the depth and width can affect a model's ability to exploit larger batches, but not necessarily in a consistent way across different model architectures. In Figure 3d, the regions of perfect scaling, diminishing returns, and maximum useful batch size do not change much when the width is varied for the Fully Connected model on MNIST, although the shallower model seems less able to exploit larger batches than the deeper models. This contrasts with the findings of Chen et al. (2018), although they changed width and depth simultaneously while keeping the number of parameters fixed. For Simple CNN on MNIST, the relationship between batch size and steps to a goal validation error seems not to depend on width at all (Figure 15e in the Appendix shows that the curves are the same even when the y-axis is not normalized). However, in Figure 3f, the curves for *narrower* Transformer models on LM1B flatten later than for wider Transformer models, while the depth seems to have less of an effect. Thus, reducing width appears to allow Transformer to make more use of larger batch sizes on LM1B.

4.4. Momentum Extends Perfect Scaling to Larger Batch Sizes, but Matches Plain SGD at Small Batch Sizes

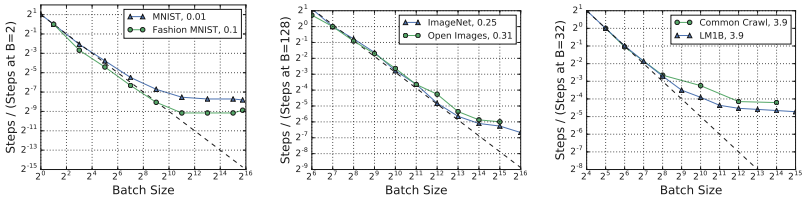
We investigated whether some optimizers can make better use of larger batches than others by experimenting with plain SGD, SGD with momentum, and Nesterov momentum on the same model and data set. Since plain SGD is a special case of both Nesterov momentum and SGD with momentum (with $\gamma = 0$ in each case), and since we tune γ in all experiments, we expect that experiments with either of these optimizers should do no worse than plain SGD at any batch size. However, it is not clear *a priori* whether momentum optimizers should outperform SGD, either by taking fewer training steps or by extending the perfect scaling region to larger batch sizes.

Figure 4 shows that Nesterov momentum and SGD with momentum can both extend the perfect scaling region beyond that achieved by SGD, and thus can significantly reduce the number of training steps required to reach a goal validation error at larger batch sizes. However, at batch sizes small enough that all optimizers are within their perfect scaling region, momentum optimizers perform identically to SGD without momentum. Though initially surprising, this identical performance at small batch sizes is consistent with observations made in Kidambi et al. (2018). In our experiments, we did not see a large difference between Nesterov momentum and SGD with momentum—Nesterov momentum appears to scale slightly better for Transformer on LM1B, but both perform about equally well for Simple CNN on MNIST.



(a) Simple CNN on MNIST (b) Transformer Shallow on LM1B (c) ResNet-8 on CIFAR-10

Figure 4: **Momentum extends perfect scaling to larger batch sizes, but matches plain SGD at small batch sizes.** Nesterov momentum and SGD with momentum can both significantly reduce the absolute number of training steps to reach a goal validation error, and also significantly extend the perfect scaling region and thus better exploit larger batches than plain mini-batch SGD.



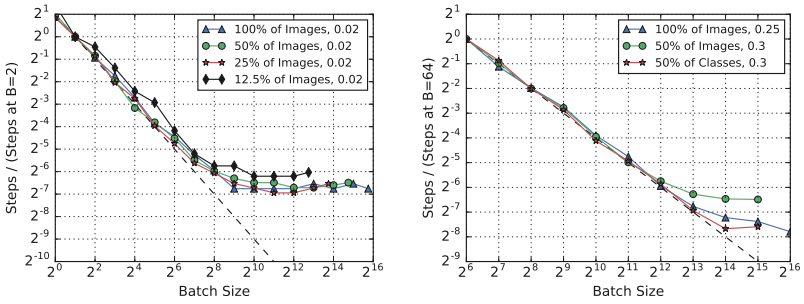
(a) Simple CNN on different data sets (b) ResNet-50 on different data sets (c) Transformer on different data sets

Figure 5: **The data set can influence the maximum useful batch size**. For the data sets shown in this plot, these differences are not simply as straightforward as larger data sets making larger batch sizes more valuable. Appendix A.2 describes the evaluation metric used for each data set, and the plot legends show the goal metric value for each task. Figure 16 in the Appendix contains these plots without the y-axis normalized.

4.5. The Data Set Matters, at Least Somewhat

We investigated whether properties of the data set make some problems able to exploit larger batch sizes than others by experimenting with different data sets while keeping the model and optimizer fixed. We approached this in two ways: (i) by testing the same model on completely different data sets, and (ii) by testing the same model on different subsets of the same data set. We normalized the y-axis of all plots in this section in the same way as Section 4.3. Appendix D contains all plots in this section without the y-axis normalized.

Figure 5 shows that changing the data set can affect the relationship between batch size and the number of steps needed to reach a goal validation error. Figure 5a shows that Fashion MNIST deviates from perfect scaling at a slightly larger batch size than MNIST for the Simple CNN model. Figure 5b shows that ImageNet and Open Images are extremely similar in how well ResNet-50 can make use of larger batch sizes, although, if anything, ImageNet might make slightly better use of larger batch sizes. Figure 5c shows that LM1B scales slightly better with increasing batch size than Common Crawl for Transformer. Since



(a) Simple CNN on MNIST subsets

(b) ResNet-50 on ImageNet subsets

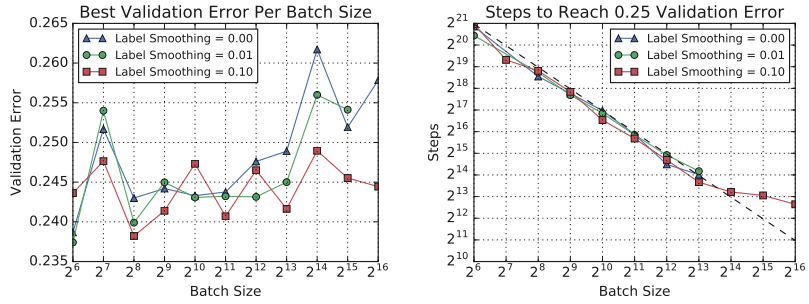
Figure 6: **Investigating the effect of data set size.** At least for MNIST, any effect of subset size on the maximum useful batch size is extremely small or nonexistent. For ImageNet, the random subset of half the images deviates from perfect scaling sooner than the full data set, but the curve for the subset with half the classes is very close to the curve for the full data set and, if anything, deviates from perfect scaling later. Appendix A.2 describes the evaluation metric used for each data set, and the plot legends show the goal metric value for each task. Figure 17 in the Appendix contains these plots without the y-axis normalized.

Fashion MNIST is the same size as MNIST, Open Images is larger than ImageNet, and Common Crawl is far larger than LM1B, these differences are not simply as straightforward as larger data sets making larger batch sizes more valuable.

To disentangle the effects from changes to the distribution and changes to the number of examples, we generated steps to result vs batch size plots for different random subsets of MNIST (Figure 6a) and ImageNet (Figure 6b). For MNIST, we selected subsets of different sizes, while for ImageNet, we selected a random subset of half the images and a similar sized subset that only includes images from half of the classes. At least on MNIST, any effect on the maximum useful batch size is extremely small or nonexistent. For ImageNet, Figure 6b shows that the random subset of half the images deviates from perfect scaling sooner than the full data set, but the curve for the subset with half the classes is very close to the curve for the full data set and, if anything, deviates from perfect scaling later, even though it contains roughly the same number of images as the random subset.

4.6. Regularization Can Be More Helpful at Some Batch Sizes Than Others

We used label smoothing (Szegedy et al., 2016) to regularize training in our experiments with ResNet-50 on ImageNet. Without label smoothing, we could not achieve our goal validation error rate of 0.25 with batch sizes greater than 2^{14} within our training budget. With a fixed compute budget for each batch size, label smoothing improved the error by as much as one percentage point at large batch sizes, while having no apparent effect at small batch sizes (Figure 7a). Meanwhile, if multiple choices for the label smoothing metaparameter achieved the goal within the training budget, then label smoothing did not change the number of steps needed (Figure 7b).



(a) Label smoothing benefits larger batch sizes, but has no apparent effect for smaller batch sizes.

(b) Label smoothing has no apparent effect on training speed, provided the goal error is achieved.

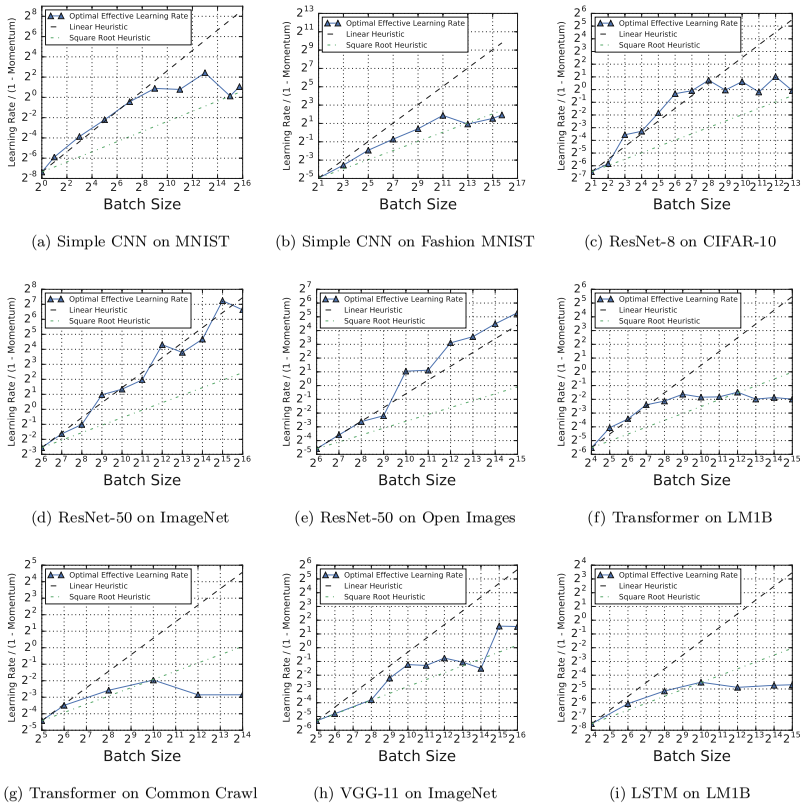
Figure 7: **Regularization can be more helpful at some batch sizes than others.** Plots are for ResNet-50 on ImageNet. Each point corresponds to a different metaparameter tuning trial, so the learning rate, Nesterov momentum, and learning rate schedule are independently chosen for each point. The training budget is fixed for each batch size, but varies between batch sizes.

We confirmed that label smoothing reduced overfitting at large batch sizes for ResNet-50 on ImageNet (see Figure 18 in the Appendix). This is consistent with the idea that noise from small batch training is a form of implicit regularization (e.g. Goodfellow et al., 2016). However, although our results show that *other forms of regularization can serve in place of this noise*, it might be difficult to select and tune other forms of regularization for large batch sizes. For example, we unsuccessfully tried to control overfitting with larger batch sizes by increasing the L_2 weight penalty and by applying additive Gaussian gradient noise before we obtained good results with label smoothing.

Finally, we also tried label smoothing with Simple CNN on MNIST and Fashion MNIST, and found that it generally helped *all* batch sizes, with no consistent trend of helping smaller or larger batch sizes more (see Figure 19 in the Appendix), perhaps because these data sets are sufficiently small and simple that overfitting is an issue at all batch sizes.

4.7. The Best Learning Rate and Momentum Vary with Batch Size

Across all problems we considered, the effective learning rate (η^{eff}; see Section 2.2) that minimized the number of training steps to a goal validation error tended to increase with increasing batch size (Figure 8). However, it did not always follow either a linear or square root scaling heuristic, despite the popularity of these rules of thumb. In some cases, the optimal effective learning rate even decreased for larger batch sizes. We also found that the best effective learning rate should be chosen by jointly tuning the learning rate and momentum, rather than tuning only the learning rate. For example, the optimal way to scale the effective learning rate for Transformer was to increase the momentum while decreasing the learning rate or holding it constant (see Figures 21 and 22 in the Appendix). This is a refinement to past prescriptions that only change the learning rate while keeping the momentum fixed.

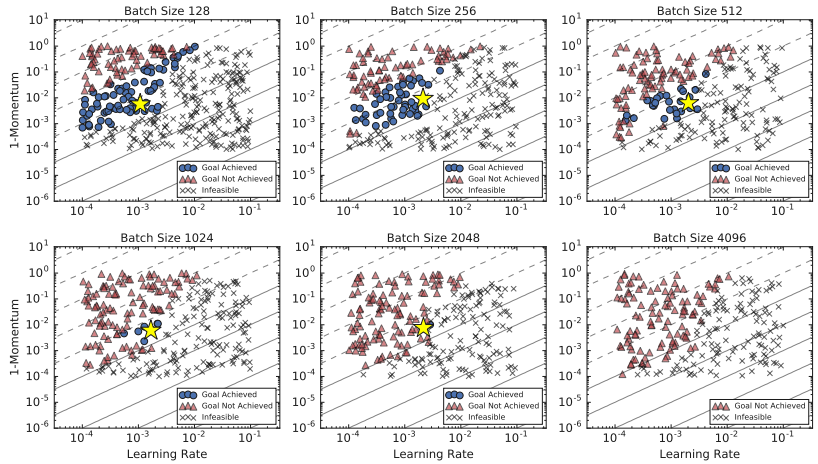


(a) Simple CNN on MNIST (b) Simple CNN on Fashion MNIST (c) ResNet-8 on CIFAR-10

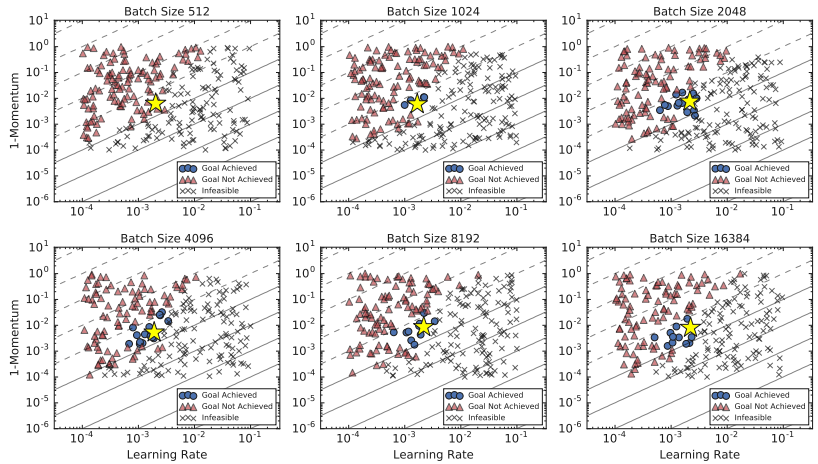
(d) ResNet-50 on ImageNet (e) ResNet-50 on Open Images (f) Transformer on LM1B

(g) Transformer on Common Crawl (h) VGG-11 on ImageNet (i) LSTM on LM1B

Figure 8: **Optimal effective learning rates do not always follow linear or square root scaling heuristics.** Effective learning rates correspond to the trial that reached the goal validation error in the fewest training steps (see Figure 1). For models that used learning rate decay schedules (ResNet-8, ResNet-50, VGG-11), plots are based on the initial learning rate. See Figures 21 and 22 in the Appendix for separate plots of the optimal learning rate and momentum.



(a) Transformer on LM1B with a training budget of one epoch.



(b) Transformer on LM1B with a training budget of 25,000 steps.

Figure 9: **With increasing batch size, the region in metaparameter space corresponding to rapid training in terms of epochs becomes smaller, while the region in metaparameter space corresponding to rapid training in terms of step-count grows larger.** Yellow stars are the trials that achieved the goal in the fewest number of steps. Contours indicate the effective learning rate $\eta^{\text{eff}} = \frac{\eta}{1-\gamma}$. Infeasible trials are those that resulted in divergent training.

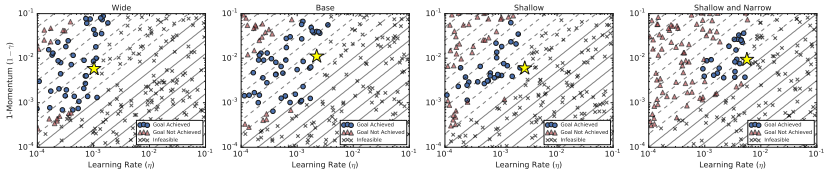


Figure 10: **Smaller models have larger stable learning rates for Transformer on LM1B.** Plots are for different sizes of Transformer on LM1B with a batch size of 1024, a goal validation cross entropy error of 4.2, and a training budget of 50,000 steps. Contours indicate the effective learning rate $\eta^{\text{eff}} = \frac{\eta}{1-\gamma}$. Infeasible trials are those that resulted in divergent training.

We further investigated the relationship between learning rate, momentum, and training speed by examining our metaparameter search spaces for different batch sizes and model sizes. For this analysis, we used Transformer on LM1B with Nesterov momentum because the metaparameter search spaces are consistent between all batch and model sizes, and can be easily visualized because they consist only of the constant learning rate η and the momentum γ. We observe the following behaviors:

- With increasing batch size, the region in metaparameter space corresponding to rapid training in terms of epochs becomes smaller (Figure 9a, consistent with the findings of Breuel, 2015b), while the region in metaparameter space corresponding to rapid training in terms of step-count grows larger (Figure 9b, although it eventually plateaus for batch sizes in the maximal data parallelism regime). Thus, with a fixed error goal and in a setting where training epochs are constrained (e.g. a compute budget), it may become more challenging to choose good values for the metaparameters with increasing batch size. Conversely, with a fixed error goal and in a setting where training steps are constrained (e.g. a wall-time budget), it may become easier to choose good values for the metaparameters with increasing batch size.

- The metaparameters yielding the fastest training are typically on the edge of the feasible region of the search space (Figure 9). In other words, small changes in the optimal metaparameters might make training diverge. This behavior may pose a challenge for metaparameter optimization techniques, such as Gaussian Process approaches, that assume a smooth relationship between metaparameter values and model performance. It could motivate techniques such as learning rate warm-up that enable stability at larger eventual learning rates, since the maximum stable learning rate depends on the current model parameters. We did not observe the same behavior for ResNet-50 on ImageNet. Figure 20 in the Appendix shows the results for a range of effective learning rates near the optimum for ResNet-50 on ImageNet and Transformer on LM1B.

- Smaller models have larger stable learning rates (Figure 10). This is consistent with recent work predicting that the largest stable learning rate is inversely proportional to layer width (Karakida et al., 2018).

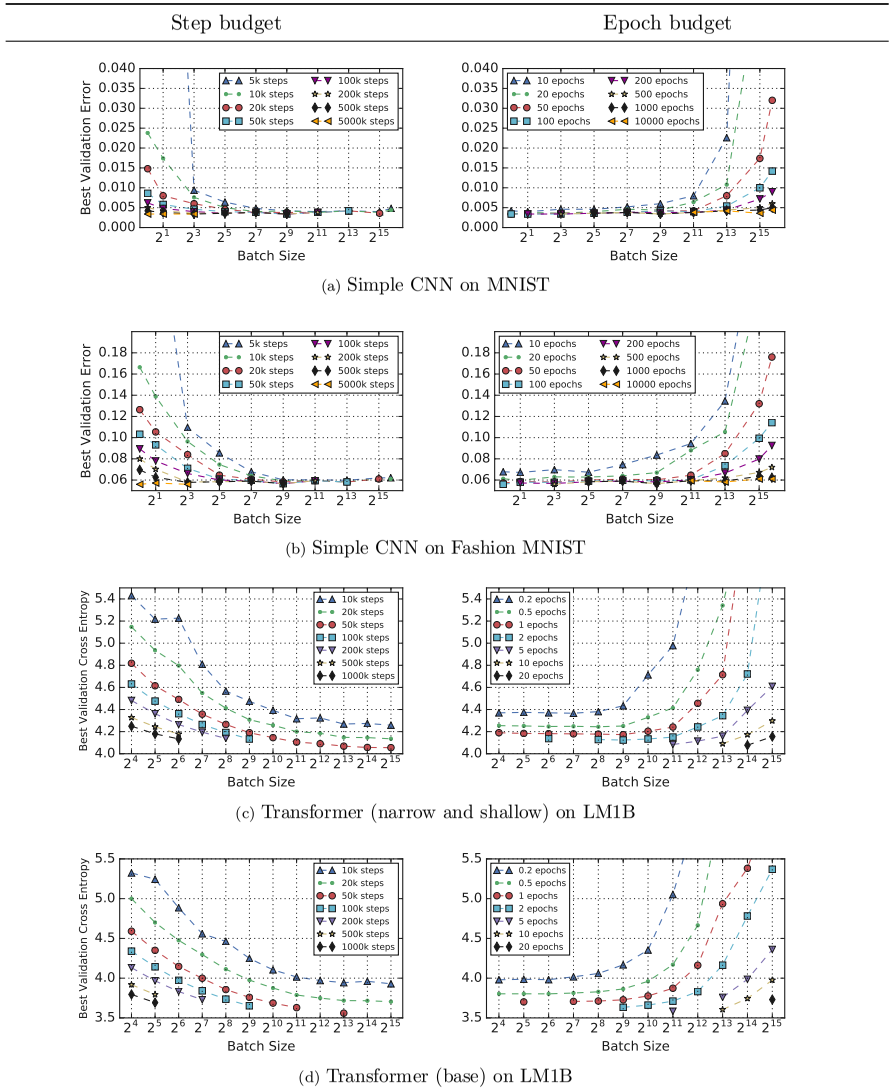
4.8. Solution Quality Depends on Compute Budget More Than Batch Size

We investigated the relationship between batch size and out-of-sample error for Simple CNN on MNIST and Fashion MNIST, and for two sizes of Transformer on LM1B. For each task, we ran a quasi-random metaparameter search over the constant learning rate η and Nesterov momentum γ. For MNIST and Fashion MNIST, we also added label smoothing and searched over the label smoothing parameter in $\{0, 0.1\}$ to mitigate any confounding effects of overfitting (see Section 4.6). We ran 100 metaparameter trials for each batch size with a large practical wall-time budget.

To disentangle the effects of the batch size from the compute budget, we compared batch sizes subject to budgets of either training steps or training epochs. For each batch size and compute budget, we found the model checkpoint that achieved the best validation accuracy across all metaparameter trials, and across all training steps that fell within the compute budget. Figure 11 shows the validation error for these best-validation-error checkpoints, as a function of batch size, for a range of compute budgets. We observe that, subject to a budget on training steps, larger batch sizes achieve better out-of-sample error than smaller batch sizes, but subject to a budget on training epochs, smaller batch sizes achieve better out-of-sample error than larger batch sizes. These observations are likely explained by the observations that, for a fixed number of training steps, larger batch sizes train on more data, while for a fixed number of epochs, smaller batch sizes perform more training steps.

The workloads in Figure 11 represent two distinct modes of neural network training. For the small MNIST and Fashion MNIST data sets, we used training budgets that would saturate (or almost saturate) performance at each batch size. In other words, out-of-sample error cannot be improved by simply increasing the budget, with caveats due to practical limitations on our ability to find optimal values for the metaparameters. Figures 11a and 11b show that differences in maximum performance between batch sizes on these data sets are very small (see Figures 23 and 24 in the Appendix for zoomed versions of these plots). We cannot rule out that any differences at this magnitude are due to noise from metaparameter choices and training stochasticity. Thus, for these workloads at least, the effect of batch size on solution quality is either very small or nonexistent. On the other hand, we cannot saturate performance with Transformer on LM1B within a practical training time. In this case, Figures 11c and 11d show that the best error is simply achieved by the largest compute budget.

Taken together, these observations suggest that in practice *the relevant question is not which batch size leads to the best performance, but rather how compute budget varies as a function of batch size.* Although we tried our best to saturate performance with MNIST and Fashion MNIST, we found that it took millions of training steps for small batch sizes, and thousands of epochs for large batch sizes, even for data sets as small and simple as these. Indeed, despite sampling 100 metaparameter configurations per batch size and training for up to 25 hours per configuration, it is still not certain whether we truly saturated performance at the smallest and largest batch sizes (see Figures 23 and 24 in the Appendix). Thus, the regime of saturated performance is of limited practical concern for most workloads—the compute budget required to saturate performance is likely beyond what a practitioner would typically use. For realistic workloads, practitioners should be most concerned with identifying the batch size at which they can most efficiently apply their compute.



(a) Simple CNN on MNIST

(b) Simple CNN on Fashion MNIST

(c) Transformer (narrow and shallow) on LM1B

(d) Transformer (base) on LM1B

Figure 11: **Validation error depends on compute budget more than batch size.** Plots show the best validation error subject to budgets of training steps (left column) or training epochs (right column). Step budgets favor large batch sizes, while epoch budgets favor small batch sizes.

5. Discussion

Our goals in measuring the effects of data parallelism on neural network training were twofold: first, we hoped to produce actionable advice for practitioners, and second, we hoped to understand the utility of building systems capable of very high degrees of data parallelism. Our results indicate that, for idealized data parallel hardware, there is a universal relationship between training time and batch size, but there is dramatic variation in how well different workloads can make use of larger batch sizes. Across all our experiments, increasing the batch size initially reduced the number of training steps needed proportionally. However, depending on the workload, this perfect scaling regime ended anywhere from a batch size of 2^4 to a batch size of 2^{13}. As batch size increases beyond the perfect scaling regime, there are diminishing returns (where increasing the batch size by a factor of k only reduces the number of training steps needed by a factor less than k) that end with a maximum useful batch size (where increasing the batch size no longer changes the number of training steps needed). Once again, the maximum useful batch size is extremely problem-dependent and varied between roughly 2^9 and 2^{16} in our experiments. Other workloads may have the region of perfect scaling end at batch sizes even smaller or larger than the range we observed, as well as having even smaller or larger maximum useful batch sizes.

On the one hand, the possibility that perfect scaling can extend to batch sizes beyond 2^{13} for some workloads is good news for practitioners because it suggests that efficient data-parallel systems can provide extremely large speedups for neural network training. On the other hand, the wide variation in scaling behavior across workloads is bad news because any given workload might have a maximum useful batch size well below the limits of our hardware. Moreover, for a new workload, measuring the training steps needed as a function of batch size and confirming the boundaries of the three basic scaling regimes requires expensive experiments. In this work, we have only described how to retrospectively predict the scaling behavior by tuning the optimization metaparameters for every batch size. Although Golmant et al. (2018) also described the same basic scaling behavior we found, in their experiments the relationship did not appear consistently across problems, across error goals, or in out-of-sample error. In light of our own results, the heuristics they assumed for adjusting the learning rate as a function of batch size are the likely cause of these inconsistencies, but this explanation only drives home the inconvenience of having to carefully tune at every new batch size. We were unable to find reliable support for any of the previously proposed heuristics for adjusting the learning rate as a function of batch size. Thus we are forced to recommend that practitioners tune all optimization parameters anew when they change the batch size or they risk masking the true behavior of the training procedure.

If the scaling behavior of workloads with respect to batch size has a simple dependence on properties of the workload, then we might be able to predict the limits of perfect scaling (or the maximum useful batch size) before running extensive experiments. We could then prioritize workloads to run on specialized hardware or decide whether gaining access to specialized hardware would be useful for a given workload of interest. On the one hand, our results are bad news for practitioners because they show that accurate scaling predictions must depend on a combination of non-obvious properties of the model, optimizer, and data set. On the other hand, we have a lot of control over the choice of model and optimizer

and there is some indication that they might be responsible for the largest portion of the variation between workloads. Our results comparing SGD and SGD with momentum (or Nesterov momentum) show that, at least for the problems we tried, momentum can extend perfect scaling to much larger batch sizes, offering clear guidance for practitioners. Other optimizers, such as KFAC (Martens and Grosse, 2015; Grosse and Martens, 2016; Ba et al., 2017), or optimization techniques designed specifically for massively data parallel systems (e.g. Li et al., 2014), might allow perfect scaling to extend much further. Intuitively, it seems plausible that optimizers that estimate local curvature information might be able to benefit more from large batches than optimizers that only use gradients.

Although the model seems to have a large effect on the maximum useful batch size and the limit of perfect scaling, our results do not give definitive answers on exactly how to design models that scale better for a given optimizer and data set. Even when we kept the model family fixed, we observed somewhat inconsistent results from changing the model width and depth. Chen et al. (2018) suggested that wider models can exploit larger batch sizes than narrower models, but their theoretical arguments only apply to linear networks and fully connected networks with a single hidden layer. In contrast, we found that *narrower* variants of the Transformer model scaled better to larger batch sizes, although it is unclear if the same notion of "width" transfers between different types of neural networks.

Unlike the model and optimizer, we generally have much less control over the data set. Unfortunately, properties of the data set also affect how well training scales in practice. Our results are equivocal on whether the number of training examples has any effect, but changing the data set entirely can certainly change the scaling behavior with respect to batch size.

Finally, our results at least partially reconcile conflicting stances in the literature on whether increasing the batch size degrades model quality. Our experiments show that:

1. Any study that only tunes the learning rate for one batch size and then uses a heuristic to choose the learning rate for other batch sizes (Goyal et al., 2017; Keskar et al., 2017; Hoffer et al., 2017; Lin et al., 2018; Devarakonda et al., 2017; Golmant et al., 2018) gives a systematic advantage to the batch size used in tuning (as well as nearby batch sizes). Our results did not show a simple relationship between the optimal learning rate and batch size that scales indefinitely (see Figures 8 and 21), so the use of simple heuristics for batch sizes sufficiently far from the base batch size could very well explain the degraded solutions and divergent training reported in prior work. Similarly, the optimal values of other metaparameters, such as the momentum and learning rate decay schedule, should not be assumed to remain constant or scale in a simple way as the batch size increases.

2. Assuming an epoch budget when comparing solution quality between batch sizes (Masters and Luschi, 2018; Goyal et al., 2017; Lin et al., 2018; Devarakonda et al., 2017), in effect, limits an investigation to the perfect scaling region of the steps to result vs batch size curve (see Figure 1). This budget favors smaller batch sizes because they will perform more optimizer steps for the same number of training examples (see Section 4.8). Certainly, there are situations where an epoch budget is appropriate, but there may exist budgets just outside the perfect scaling region that can achieve the same quality solution, and those budgets may still represent a significant reduction in the number of training steps required. Moreover, even for a fixed model and data set, simply changing the optimizer can significantly extend the perfect scaling regime to larger batch sizes. For example, Masters and Luschi (2018) found that test performance of ResNet-8 (without batch normalization) on CIFAR-10 with a fixed epoch budget degraded after batch size 16, but considered only plain mini-batch SGD. Our experiments confirmed that perfect scaling ends at batch size 16 with plain mini-batch SGD, but using Nesterov momentum extends the perfect scaling regime to batch size 256 (see Figure 1c).

3. Assuming a step budget when comparing solution quality between batch sizes (Hoffer et al., 2017) might favor larger batch sizes because they will see more training examples for the same number of gradient updates (see Section 4.8). A step budget is likely sufficient for a larger batch size to reach *at least* the same performance as a smaller batch size: we never saw the number of steps to reach a goal validation error increase when the batch size was increased (see Figure 1).

4. Increasing the batch size reduces noise in the gradient estimates (see Equation 4). However, the noise in updates due to small batches might, in some cases, provide a helpful regularization effect (Goodfellow et al., 2016; Smith and Le, 2018). Thankfully, other regularization techniques, such as label smoothing, can replace this effect (see Section 4.6). Others have also used regularization techniques, such as data augmentation (Keskar et al., 2017) and L_2 regularization (Smith and Le, 2018), to eliminate the "generalization gap" between two batch sizes.

5. Finally, although we do not believe there is an inherent degradation in solution quality associated with increasing the batch size, depending on the compute budget, it may become increasingly difficult to find good values for the metaparameters with larger batch sizes. Specifically, increasing the batch size may shrink the region in metaparameter space corresponding to rapid training in terms of epochs (see Figure 9a), as previously reported by Breuel (2015b). On the other hand, increasing the batch size may increase the region in metaparameter space corresponding to rapid training in terms of steps (see Figure 9b).

5.1. Limitations of our experimental protocol

When interpreting our results, one should keep in mind any limitations of our experimental protocol. We do not believe any of these limitations are debilitating, and we hope that describing these potential areas of concern will spur methodological innovation in future work.

Firstly, we were unable to avoid some amount of human judgment when tuning metaparameters. Although we did not tune metaparameters by hand, we specified the search spaces for automatic tuning by hand and they may not have been equally appropriate for all batch sizes, despite our best efforts. We are most confident in our search spaces that tuned the fewest metaparameters (such as in our experiments that only tuned learning rate and momentum). We found it quite difficult to be confident that our tuning was sufficient when we searched over learning rate decay schedules; readers should be aware that the steps to result measurement is generally quite sensitive to the learning rate schedule. Thus, we

may not have sampled enough trials at some batch sizes or, nearly equivalently, our search spaces may have been too wide at some batch sizes. Even though we verified that the best trial was not on the boundary of the search space, this by no means guarantees that we found the globally optimal metaparameters.

Smaller batch sizes typically had more opportunities to measure validation error and, when validation error was noisy, got more chances to sample a lucky validation error. Batch sizes (usually larger ones) that did not reach the goal validation error using the first search space used revised search spaces that gave them an extra bite of the apple, so to speak.

Finally, our analysis does not consider how robustly we can reach a goal error rate. For instance, we did not distinguish between batch sizes where all 100 trials achieved the goal validation error and batch sizes where only one of the 100 trials achieved the goal. The maximum or minimum value over a set of trials is not usually a very robust statistic, but something like the 50th percentile trial mostly reveals information about the search space. We tried to strike a balance between studying realistic workloads and being able to repeat our experiments so many times that these uncertainty questions became trivial. Ultimately, we opted to study realistic workloads and simply report results for the optimal trials.

6. Conclusions and Future Work

Increasing the batch size is a simple way to produce valuable speedups across a range of workloads, but, for all workloads we tried, the benefits diminished well within the limits of current hardware. Unfortunately, blindly increasing the batch size to the hardware limit will not produce a large speedup for all workloads. However, our results suggest that some optimization algorithms may be able to consistently extend perfect scaling across many models and data sets. Future work should perform our same measurements with other optimizers, beyond the closely-related ones we tried, to see if any existing optimizer extends perfect scaling across many problems. Alternatively, if we only need speedups for specific, high-value problems, we could also consider designing models that extend perfect scaling to much larger batch sizes. However, unlike the optimizer, practitioners are likely to tailor their model architectures to the specific problems at hand. Therefore, instead of searching for model architectures that happen to scale extremely well, future work should try to uncover general principles for designing models that can scale perfectly to larger batch sizes. Even if such principles remain elusive, we would still benefit from methods to prospectively predict the scaling behavior of a given workload without requiring careful metaparameter tuning at several different batch sizes. Finally, the deep learning community can always benefit from methodical experiments designed to test hypotheses, characterize phenomena, and reduce confusion, to balance more exploratory work designed to generate new ideas for algorithms and models.

Acknowledgements

We thank Tomer Koren for helpful discussions. We also thank Justin Gilmer and Simon Kornblith for helpful suggestions and comments on the manuscript. Finally, we thank Matt J. Johnson for lending us some computing resources.

Appendix A. Data Set Details

This section contains details of the data sets summarized in Table 1.

A.1. Data Set Descriptions and Pre-Processing

MNIST (LeCun et al., 1998) is a classic handwritten digit image classification data set with 10 mutually exclusive classes. We split the original training set into 55,000 training images and 5,000 validation images, and used the official test set of 10,000 images. We did not use data augmentation.

Fashion MNIST (Xiao et al., 2017) is another reasonably simple image classification data set with 10 mutually exclusive classes. It was designed as a drop-in replacement for MNIST. We split the original training set into 55,000 training images and 5,000 validation images, and used the official test set of 10,000 images. We did not use data augmentation.

CIFAR-10 (Krizhevsky, 2009) is an image classification data set of 32×32 color images with 10 mutually exclusive classes. We split the original training set into 45,000 training images and 5,000 validation images. We used the official test set of 10,000 images. We pre-processed each image by subtracting the average value across all pixels and channels and dividing by the standard deviation.[15] We did not use data augmentation.

ImageNet (Russakovsky et al., 2015) is an image classification data set with 1,000 mutually exclusive classes. We split the official training set into 1,281,167 training images and 50,045 test images, and used the official validation set of 50,000 images. We pre-processed the images and performed data augmentation in a similar way to Simonyan and Zisserman (2014). Specifically, at training time, we sampled a random integer $S \in [256, 512]$, performed an aspect-preserving resize so that the smallest side had length S, and took a random crop of size $(224, 224)$. We randomly reflected the images horizonally, but unlike Simonyan and Zisserman (2014), we did not distort the colors. At evaluation time, we performed an aspect-preserving resize so that the smallest side had length 256, and took a central crop of size $(224, 224)$. In both training and evaluation, we then subtracted the global mean RGB value from each pixel using the values computed by Simonyan and Zisserman (2014).[16]

Open Images v4 (Krasin et al., 2017) is a data set of 9 million images that are annotated with image-level labels and object bounding boxes.[17] The image labels were generated by a computer vision model and then verified as either *positive* or *negative* labels by human annotators. We only considered the 7,186 "trainable" classes with at least 100 human-annotated positives in the training set. We filtered the official subsets by keeping only images with at least one positive trainable label, which produced training, validation and test sets of size 4,526,492; 41,225; and 124,293 images, respectively. On average, each image in the training set has 2.9 human-annotated positive labels, while each image in the validation and test sets have 8.4 human-annotated positive labels. We only considered the human-annotated positives and assumed all other classes were negative. We pre-processed the images and performed data augmentation identically to ImageNet.

15. We used the TensorFlow op `tf.image.per_image_standardization`.
16. See `https://gist.github.com/ksimonyan/211839e770f7b538e2d8#description` for the mean RGB values used.
17. Available at `https://storage.googleapis.com/openimages/web/index.html`.

LM1B (Chelba et al., 2014) is a text data set of English news articles.[18] We used the official training set and created validation and test sets using files `news.en.heldout-00000-of-00050` and `news.en.heldout-00001-of-00050`, respectively. These splits contain 30,301,028; 6,075; and 6,206 sentences, respectively. We used an invertable word tokenizer to split the text into sub-word tokens with a vocabulary of size 32,000.[19] On average, the training set contains around 20 tokens per sentence and the validation and test sets contain around 29 tokens per sentence. At training time, we clipped long sentences to the first 64 tokens, which affected only about 2% of sentences. We did not clip long sentences at evaluation time. The maximum sentence across the validation and test sets has 476 tokens.

Common Crawl is a repository of web data containing over 3 billion web pages.[20] We filtered and processed the data set identically to Anil et al. (2018).[21] The vocabulary contains 24,006 sub-word tokens. We randomly partitioned the sentences into a training set (99.98%) and a holdout set (0.02%). Our training set contains ∼25.8 billion sentences. We used the first 6,075 sentences of the holdout set as our validation set, which is the same number of sentences in our LM1B validation set. Some sentences are tens of thousands of tokens long. To maintain consistency with our LM1B processing, we clipped sentences to 64 tokens at training time and 476 at evaluation time.

A.2. Evaluation Metrics

We use **classification error** for MNIST, Fashion MNIST, CIFAR-10, and ImageNet. To compute this metric, we consider the model's classification for each image to be the class it assigns the highest probability. Then

$$\text{classification error} = \frac{\#\text{ incorrect classifications}}{\#\text{ classifications}}.$$

We use class-agnostic **average precision** (AP) for Open Images. To compute this metric, we first rank each image-class pair by the predicted likelihood of the class being a true positive for that image. Then

$$AP = \frac{1}{w}\sum_{k=1}^{nm} \text{Precision}(k) \cdot \text{Relevance}(k), \tag{7}$$

where Precision(k) is the precision when considering the top k image-class pairs, Relevance(k) is an indicator function equal to 1 if the kth image-class pair is a verified positive and 0 otherwise, n is the number of images in the validation set, m is the number of classes, and w is the number of positive labels. Average precision was proposed for Open Images by Veit et al. (2017). Due to false negatives in the validation set, Veit et al. (2017) only computed AP over the the human-annotated classes in each image. However, on average, each image in the validation set only has 8.4 positive and 4 negative human-annotated classes, so each image is only evaluated over ∼12 classes out of 7,186. This yields misleadingly high values of AP. Instead, we compute AP over all classes in each image, which may underestimate the true AP due to false negatives in the validation set, but is more indicative of the true performance in our experience. We compute AP using an efficient approximation of the area under the discrete precision-recall curve.[22]

We use average per-token **cross entropy error** for LM1B and Common Crawl. For a single sentence $s = (w_1, ..., w_m)$, let $p(w_j|w_1, ..., w_{j-1})$ denote the model's predicted probability of the token w_j given all prior tokens in the sentence. Thus, the predicted log-probability of s is $\log p(s) = \sum_{j=1}^{m} \log p(w_j|w_1, ..., w_{j-1})$. We compute the average per-token cross entropy error over a data set $\{s_1, ..., s_n\}$ as

$$\text{cross entropy error} = \frac{\sum_{i=1}^{n} \log p(s_n)}{\sum_{i=1}^{n} \text{len}(s_n)},$$

where len(s) denotes the number of tokens in s. This is the logarithm of the per-token perplexity.

Appendix B. Model Details

In this section we give the architectural details of the models summarized in Table 2. In addition to the descriptions below, each model has a task-specific output layer. Models trained on MNIST, Fashion MNIST, CIFAR-10, and ImageNet (classification with mutually exclusive labels) use a softmax output layer to model the probability distribution over classes. Models trained on Open Images (classification with multiple labels per image) use a sigmoid output layer to model the probability of each class. Models trained on LM1B and Common Crawl (language modeling) use a softmax output layer to model the probability of the next word in a sentence given all prior words in the sentence.

Fully Connected is a fully connected neural network with ReLU activation function. Hidden layers use dropout with probability 0.4 during training. We vary the number of layers and number of units per layer in different experiments to investigate the impact of model size. We use the notation FC-N_1-...-N_k to denote a fully connected neural network with k hidden layers and N_i units in the ith layer.

Simple CNN consists of 2 convolutional layers with max-pooling followed by 1 fully connected hidden layer. The convolutional layers use 5×5 filters with stride length 1, "same" padding (Goodfellow et al., 2016), and ReLU activation function. Max pooling uses 2×2 windows with stride length 2. The fully connected layer uses dropout with probability 0.4 during training. We used three different model sizes: **base** has 32 and 64 filters in the convolutional layers and 1,024 units in the fully connected layer; **narrow** has 16 and 32 filters in the convolutional layers and 512 units in the fully connected layer; and **wide** has 64 and 128 filters in the convolutional layers and 2,048 units in the fully connected layer. We used the base model unless otherwise specified.

18. Available at `http://www.statmt.org/lm-benchmark/`.
19. The code for processing the raw data and generating the vocabulary is available at `https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/data_generators/lm1b.py`
20. Available at `http://commoncrawl.org/2017/07/june-2017-crawl-archive-now-available/`.
21. See `https://github.com/google-research/google-research/tree/master/codistillation` for document IDs.
22. Equation 7 can be interpreted as a right Riemann sum of the discrete precision-recall curve $\{(r_i, p_i)|i = 1, ..., w\}$, where $r_i = i/w$ and p_i is the maximum precision among all values of precision with recall r_i (each value of recall may correspond to different values of precision at different classification thresholds). We use the TensorFlow op `tf.metrics.auc` with `curve="PR"`, `num_thresholds=200`, and `summation_method="careful_interpolation"`.

ResNet-8 consists of 7 convolutional layers with residual connections followed by 1 fully connected hidden layer. We used the model described in section 4.2 of He et al. (2016a) with $n = 1$, but with the improved residual block described by He et al. (2016b). We removed batch normalization, which is consistent with Masters and Luschi (2018).

ResNet-50 consists of 49 convolutional layers with residual connections followed by 1 fully connected hidden layer. We used the model described in section 4.1 of He et al. (2016a), but with the improved residual block described by (He et al., 2016b). We replaced batch normalization (Ioffe and Szegedy, 2015) with ghost batch normalization to keep the training objective fixed between batch sizes and to avoid possible negative effects from computing batch normalization statistics over a large number of examples (Hoffer et al., 2017). We used a ghost batch size of 32 for all experiments. We also applied label smoothing (Szegedy et al., 2016) to regularize the model at training time, which was helpful for larger batch sizes. The label smoothing coefficient was a metaparameter that we tuned in our experiments.

VGG-11 consists of 8 convolutional layers followed by 3 fully connected hidden layers. We used the model referred to as "model A" by Simonyan and Zisserman (2014).

LSTM is a one hidden-layer LSTM model (Hochreiter and Schmidhuber, 1997). It is a simpler variant of the LSTM-2048-512 model described by Jozefowicz et al. (2016), with 1,024 embedding dimensions, 2,048 hidden units, and 512 projection dimensions. We did not use bias parameters in the output layer because we found this improved performance in our preliminary experiments.

Transformer is a self-attention model that was originally presented for machine translation (Vaswani et al., 2017). We used it as an autoregressive language model by applying the decoder directly to the sequence of word embeddings for each sentence. We used four different sizes: the **base** model described by Vaswani et al. (2017); a **shallow** model that is identical to the base model except with only two hidden layers instead of six; a **narrow and shallow** model that is identical to the shallow model except with half as many hidden units and attention heads as well as half the filter size; and a **wide** model that is identical to the base model except with double the number of hidden units and attention heads as well as double the filter size. We used the base model unless otherwise specified.

Appendix C. Learning Rate Schedules

We chose our learning rate schedule by experimenting with a variety of different schedules for ResNet-50 on ImageNet. For each schedule, we specified the following metaparameters:

- η_0: initial learning rate
- α: decay factor ($\alpha > 0$)
- T: number of training steps until the learning rate decays from η_0 to $\alpha\eta_0$

Each schedule corresponds to a decay function $d(t)$, such that the learning rate at training step t is

$$\eta(t) = \begin{cases} d(t) \cdot \eta_0 & \text{if } t \leq T, \\ \alpha\eta_0 & \text{if } t > T. \end{cases}$$

We experimented with the following decay functions:

- **Constant**: $d(t) = 1$
- **Linear**: $d(t) = 1 - (1 - \alpha)\frac{t}{T}$
- **Cosine** (Loshchilov and Hutter, 2017): $d(t) = \alpha + \frac{(1-\alpha)}{2}\left(1 + \cos \pi \frac{t}{T}\right)$
- **Exponential Polynomial**: $d(t) = \alpha + (1 - \alpha)\left(1 - \frac{t}{T}\right)^{\lambda}$, where $\lambda > 0$
- **Inverse Exponential Polynomial**: $d(t) = \frac{\alpha}{\alpha + (1-\alpha)\left(\frac{t}{T}\right)^{\lambda}}$, where $\lambda > 0$
- **Exponential**: $d(t) = \alpha^{t/T}$

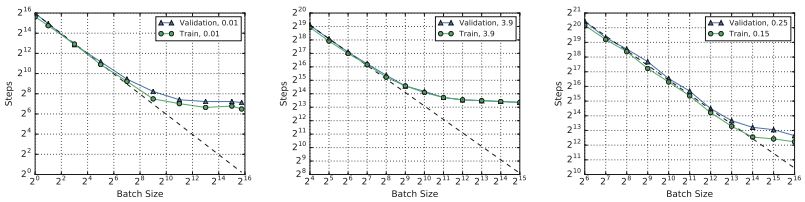
We also tried piecewise linear learning rate schedules. These schedules are specified by a sequence of pairs $\{(t_0, \eta_0), ..., (t_k, \eta_k)\}$, with $0 = t_0 < t_1 ... < t_k$, such that the learning rate at training step t is

$$\eta(t) = \begin{cases} \eta_i + \frac{\eta_{i+1} - \eta_i}{t_{i+1} - t_i}(t - t_i) & \text{if } t_i \leq t < t_{i+1}, \\ \eta_k & \text{if } t \geq t_k. \end{cases}$$

The schedules used by both He et al. (2016a) (piecewise constant) and Goyal et al. (2017) (linear warm-up followed by piecewise constant) for ResNet-50 on ImageNet can both be expressed as piecewise linear.

We ran experiments with ResNet-50 on ImageNet, using Nesterov momentum with batch size 1,024 for 150,000 training steps, while tuning the momentum and all metaparameters governing the learning rate schedule. We used quasi-random metaparameter search as described in Section 4. For piecewise linear schedules, we tried 1, 3, and 5 decay events. We found that it was possible to get good results with several of the schedules we tried, and it is likely that other schedules would also work well. Ultimately, we chose linear decay because it performed at least well as all other schedules we tried, while also being the simplest and requiring only two additional metaparameters.

Appendix D. Additional Plots



(a) Simple CNN on MNIST (b) Transformer on LM1B (c) ResNet-50 on ImageNet

Figure 12: **Steps to result on the training set is almost the same as on the validation set.** The evaluation metrics are described in Appendix A.2. Error goals are specified in the plot legends.

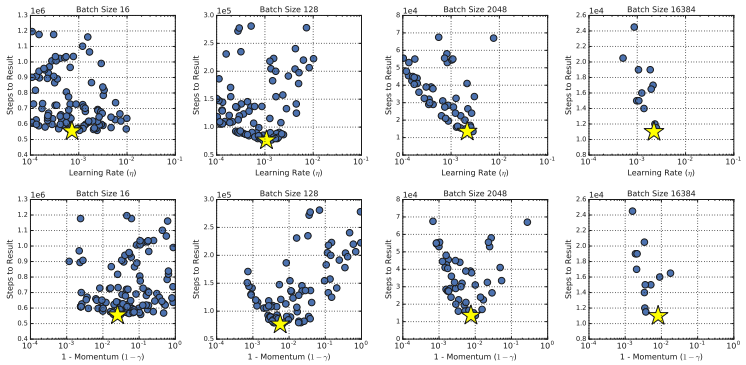


Figure 13: **Validating metaparameter search spaces for Transformer on LM1B.** Rows correspond to the metaparameters we tuned (learning rate η and momentum γ) and columns correspond to different batch sizes. The x-axis is the search range that was sampled by the quasi-random search algorithm. Blue dots represent trials that reached the goal of 3.9 validation cross entropy error, and yellow stars correspond to trials that achieved the goal in the fewest steps. We deem these search spaces appropriate because the yellow stars are not on the boundaries.

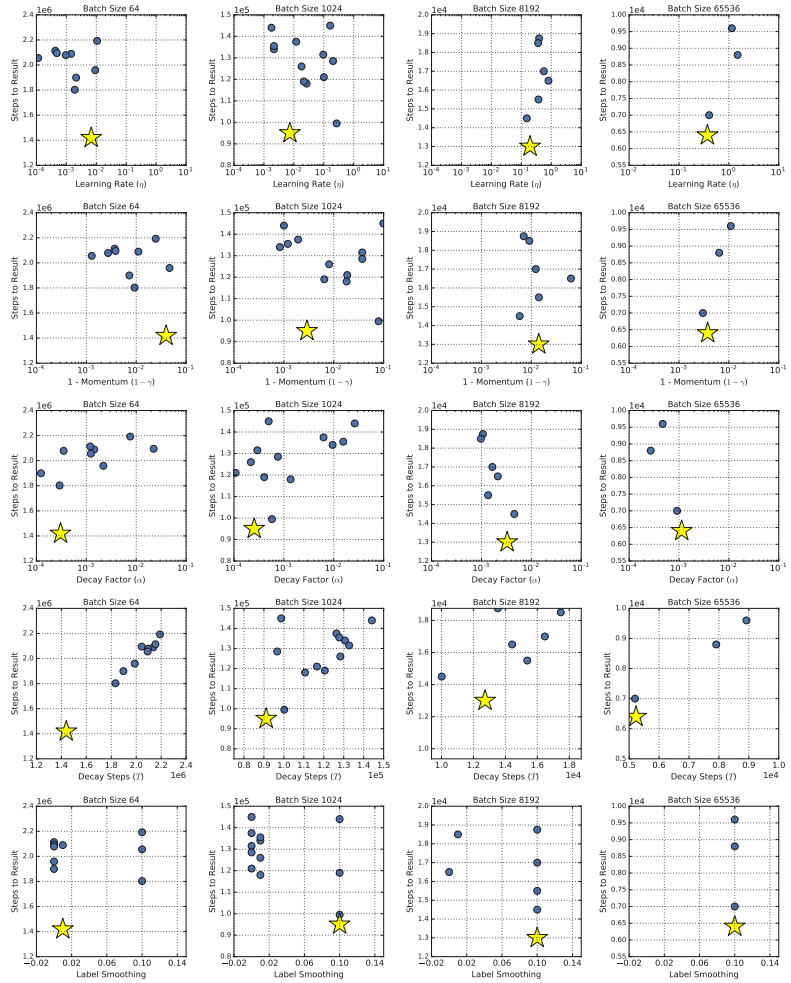
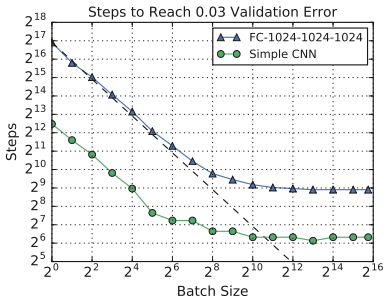
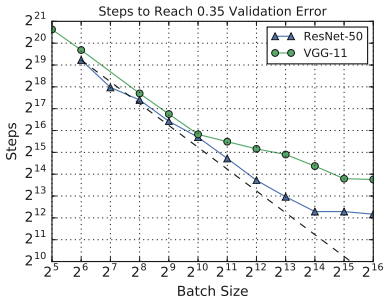


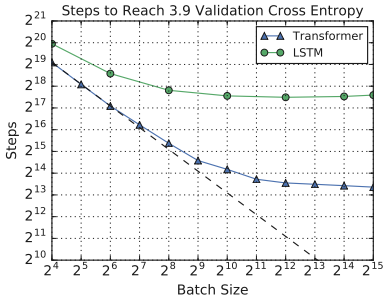
Figure 14: **Validating metaparameter search spaces for ResNet-50 on ImageNet.** Rows correspond to the metaparameters we tuned (initial learning rate η_0, momentum γ, learning rate decay parameters α, T, and label smoothing parameter) and columns correspond to different batch sizes. For all parameters except the label smoothing parameter, the x-axis is the search range sampled by the quasi-random search algorithm. The label smoothing parameter was sampled uniformly in $\{0, 0.01, 0.1\}$ for $b \leq 2^{14}$ and $\{0, 0.1\}$ for $b > 2^{14}$. Blue dots represent trials that reached the goal validation error rate of 0.25, and yellow stars correspond to trials that achieved the goal in the fewest steps. We deem these search spaces appropriate because the yellow stars are not on the boundaries.



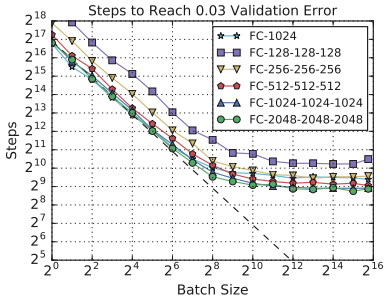
(a) Fully Connected vs Simple CNN on MNIST



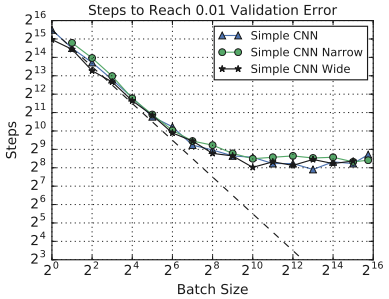
(b) ResNet-50 vs VGG-11 on ImageNet



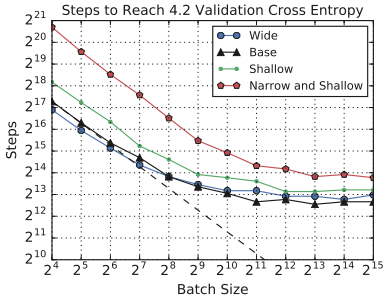
(c) Transformer vs LSTM on LM1B



(d) Fully Connected sizes on MNIST

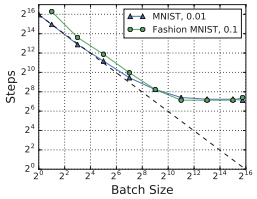


(e) Simple CNN sizes on MNIST

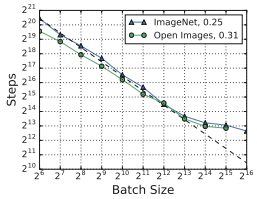


(f) Transformer sizes on LM1B

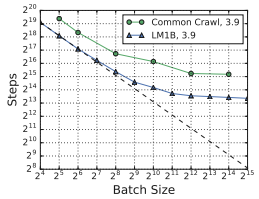
Figure 15: **Figure 3 without the y-axis normalized.**



(a) Simple CNN on different data sets

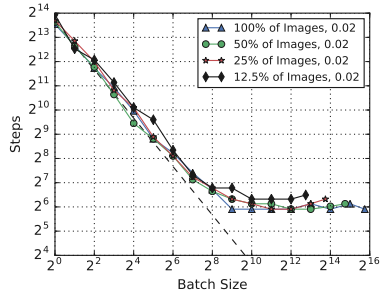


(b) ResNet-50 on different data sets

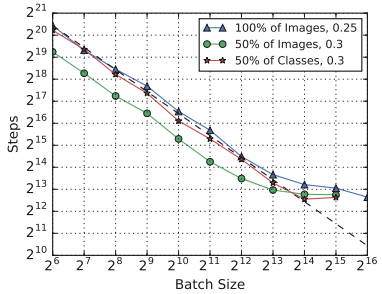


(c) Transformer on different data sets

Figure 16: **Figure 5 without the y-axis normalized.**

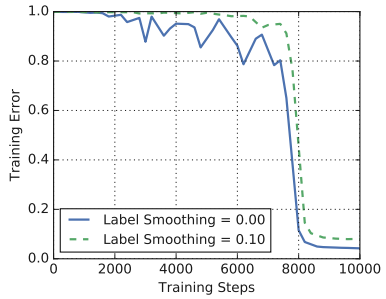


(a) Simple CNN on MNIST subsets



(b) ResNet-50 on ImageNet subsets

Figure 17: **Figure 6 without the y-axis normalized.**



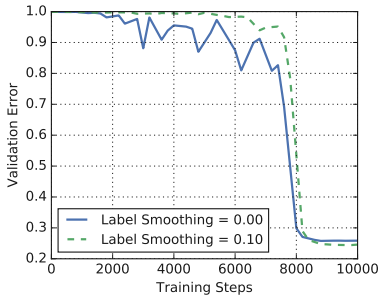
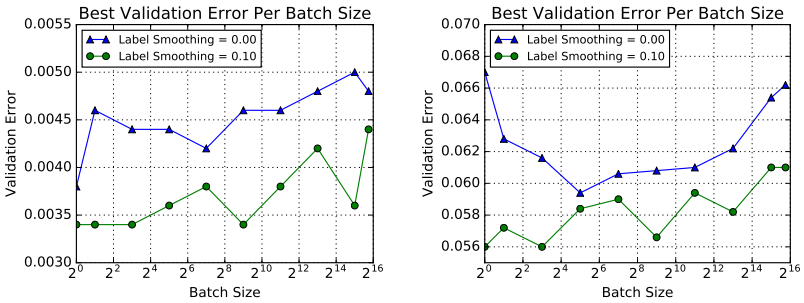
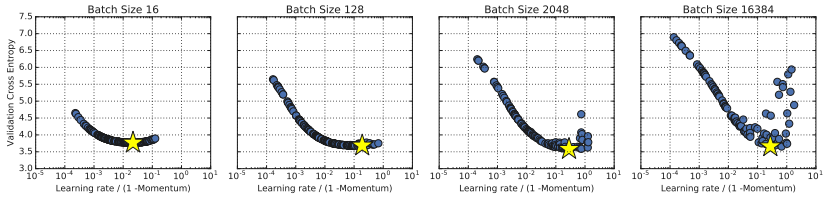


Figure 18: **Label smoothing reduces overfitting at large batch sizes.** Plots are training curves for the two best models with and without label smoothing for ResNet-50 on ImageNet with batch size 2^{16}. The two models correspond to different metaparameter tuning trials, so the learning rate, Nesterov momentum, and learning rate schedule were independently chosen for each trial. The two trials shown are those that reached the highest validation error at any point during training, for label smoothing equal to 0 and 0.1 respectively.

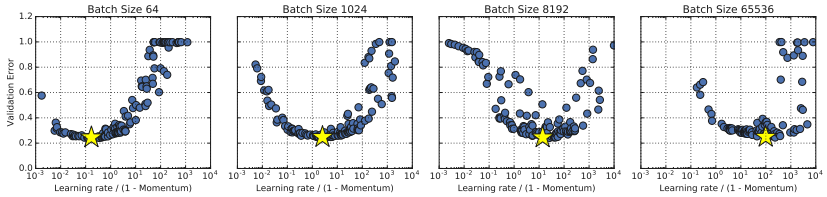


(a) Simple CNN on MNIST (b) Simple CNN on Fashion MNIST

Figure 19: **Label smoothing helps all batch sizes for Simple CNN on MNIST and Fashion MNIST.** There is no consistent trend of label smoothing helping smaller or larger batch sizes more. Each point corresponds to a different metaparameter tuning trial, so the learning rate, Nesterov momentum, and learning rate schedule are independently chosen for each point. The training budget is fixed for each batch size, but varies between batch sizes.

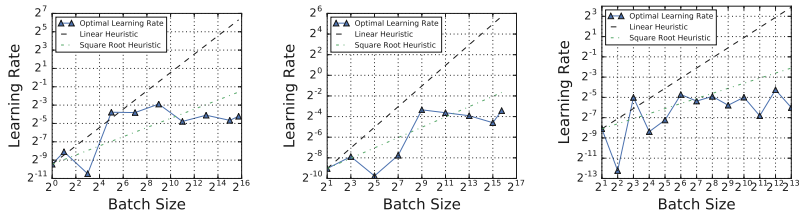


(a) Transformer on LM1B

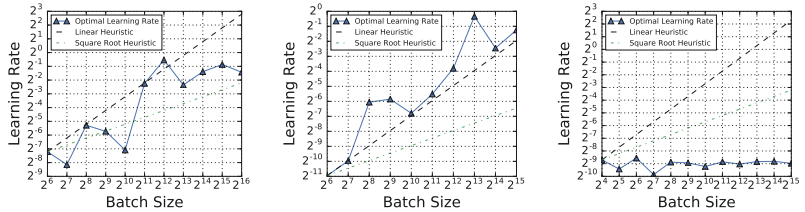


(b) ResNet-50 on ImageNet

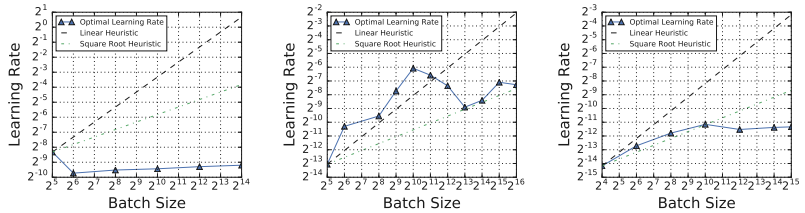
Figure 20: **Validation error vs effective learning rate.** Training budgets are consistent for each batch size, but not between batch sizes. These plots are projections of the entire metaparameter search space, which is 2-dimensional for Transformer on LM1B (see Figure 13) and 5-dimensional for ResNet-50 on ImageNet (see Figure 14).



(a) Simple CNN on MNIST (b) Simple CNN on Fashion MNIST (c) ResNet-8 on CIFAR-10

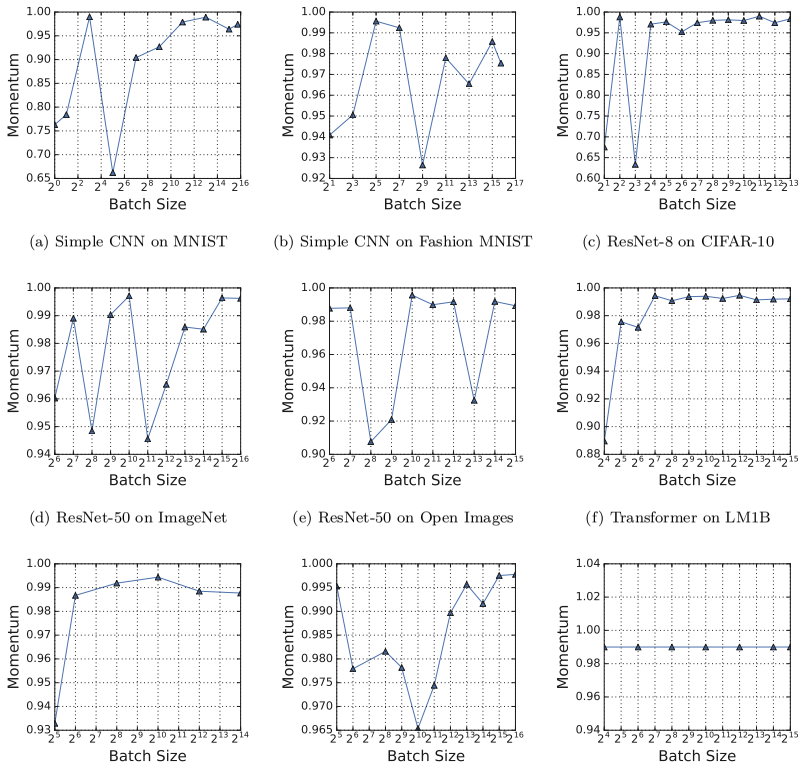


(d) ResNet-50 on ImageNet (e) ResNet-50 on Open Images (f) Transformer on LM1B



(g) Transformer on Common Crawl (h) VGG-11 on ImageNet (i) LSTM on LM1B

Figure 21: **Optimal learning rates do not always follow linear or square root scaling heuristics.** Learning rates correspond to the trial that reached the goal validation error in the fewest training steps (see Figure 1). For models using learning rate decay schedules (ResNet-8, ResNet-50, VGG-11), plots are based on the initial learning rate. See Figure 22 for the corresponding plot of optimal momentum, and Figure 8 for the corresponding plot of effective learning rate.

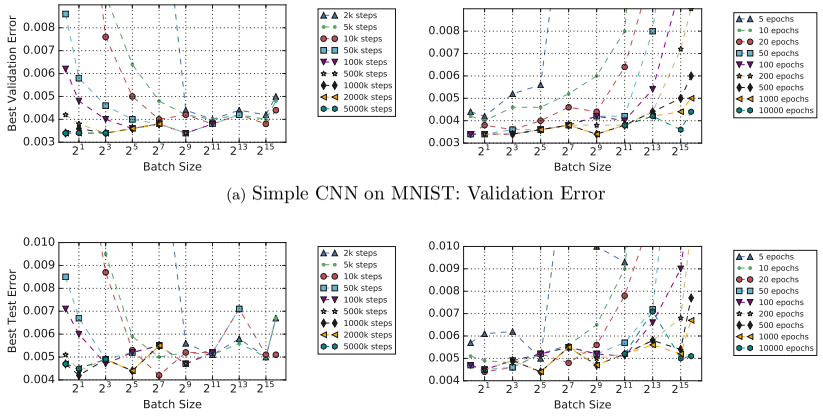


(a) Simple CNN on MNIST (b) Simple CNN on Fashion MNIST (c) ResNet-8 on CIFAR-10

(d) ResNet-50 on ImageNet (e) ResNet-50 on Open Images (f) Transformer on LM1B

(g) Transformer on Common Crawl (h) VGG-11 on ImageNet (i) LSTM on LM1B*

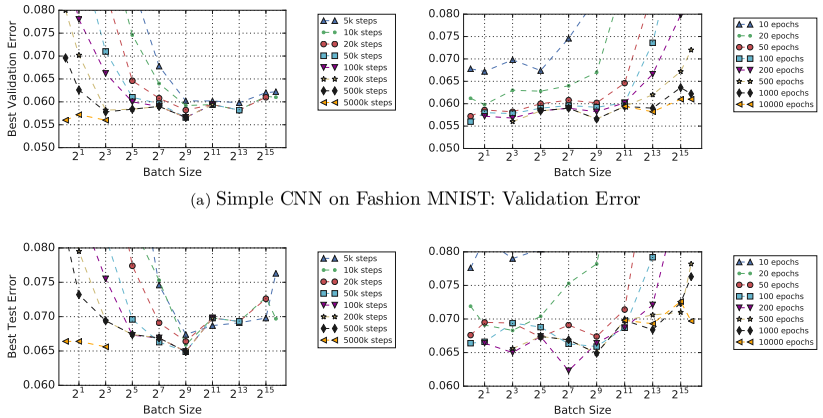
Figure 22: **Optimal momentum has no consistent relationship with batch size.** Momentum corresponds to the trial that reached the goal validation error in the fewest training steps (see Figure 1). See Figure 21 for the corresponding plot of optimal learning rate, and Figure 8 for the corresponding plot of effective learning rate. *For LSTM on LM1B, we only tuned η with fixed $\gamma = 0.99$.



(a) Simple CNN on MNIST: Validation Error

(b) Simple CNN on MNIST: Test Error

Figure 23: **Zoomed version of Figure 11a.**



(a) Simple CNN on Fashion MNIST: Validation Error

(b) Simple CNN on Fashion MNIST: Test Error

Figure 24: **Zoomed version of Figure 11b.**

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: a system for large-scale machine learning. In *Conference on Operating Systems Design and Implementation*, volume 16, pages 265–283. USENIX, 2016.

Takuya Akiba, Shuji Suzuki, and Keisuke Fukuda. Extremely large minibatch SGD: Training ResNet-50 on ImageNet in 15 minutes. *arXiv preprint arXiv:1711.04325*, 2017.

Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E. Dahl, and Geoffrey E. Hinton. Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rkr1UDeC-`.

Jimmy Ba, Roger Grosse, and James Martens. Distributed second-order optimization using Kronecker-factored approximations. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=SkkTMpjex`.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

Olivier Bousquet, Sylvain Gelly, Karol Kurach, Olivier Teytaud, and Damien Vincent. Critical hyper-parameters: No random, no cry. *arXiv preprint arXiv:1706.03200*, 2017.

Thomas M Breuel. Benchmarking of LSTM networks. *arXiv preprint arXiv:1508.02774*, 2015a.

Thomas M Breuel. The effects of hyperparameters on SGD training of neural networks. *arXiv preprint arXiv:1508.02788*, 2015b.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Conference of the International Speech Communication Association*, 2014.

Jianmin Chen, Xinghao Pan, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. Revisiting distributed synchronous SGD. In *International Conference on Learning Representations Workshop Track*, 2016. URL `https://openreview.net/forum?id=D1VDZ5kMAu5jEJ1zfEWL`.

Lingjiao Chen, Hongyi Wang, Jinman Zhao, Dimitris Papailiopoulos, and Paraschos Koutris. The effect of network width on the performance of large-batch training. *arXiv preprint arXiv:1806.03791*, 2018.

Valeriu Codreanu, Damian Podareanu, and Vikram Saletore. Scale out for large minibatch SGD: Residual network training on ImageNet-1K with improved accuracy and reduced time to train. *arXiv preprint arXiv:1711.04291*, 2017.

Aditya Devarakonda, Maxim Naumov, and Michael Garland. AdaBatch: Adaptive batch sizes for training deep neural networks. *arXiv preprint arXiv:1712.02029*, 2017.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028, 2017.

Noah Golmant, Nikita Vemuri, Zhewei Yao, Vladimir Feinberg, Amir Gholami, Kai Rothauge, Michael W Mahoney, and Joseph Gonzalez. On the computational inefficiency of large batch sizes for stochastic gradient descent. *arXiv preprint arXiv:1811.12941*, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL `http://www.deeplearningbook.org`.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Roger Grosse and James Martens. A Kronecker-factored approximate Fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582, 2016.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016b.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning, lecture 6a: overview of mini-batch gradient descent, 2012. URL `https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, pages 1731–1741, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

Prateek Jain, Sham M. Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Parallelizing stochastic gradient descent for least squares regression: Mini-batching, averaging, and model misspecification. *Journal of Machine Learning Research*, 18(223):1–42, 2018. URL `http://jmlr.org/papers/v18/16-595.html`.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *International Symposium on Computer Architecture*, pages 1–12. IEEE, 2017.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal statistics of Fisher information in deep neural networks: Mean field approach. *arXiv preprint arXiv:1806.01316*, 2018.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=H1oyRlYgg`.

Rahul Kidambi, Praneeth Netrapalli, Prateek Jain, and Sham M. Kakade. On the insufficiency of existing momentum schemes for stochastic optimization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJTutzbA-`.

Jack Kiefer, Jacob Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. OpenImages: A public dataset for large-scale multi-label and multi-class image classification., 2017. URL `https://storage.googleapis.com/openimages/web/index.html`.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Yann Le Cun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks, Tricks of the Trade*, Lecture Notes in Computer Science LNCS 1524. Springer Verlag, 1998. URL `http://leon.bottou.org/papers/lecun-98x`.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database, 1998. URL `http://yann.lecun.com/exdb/mnist`.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola. Efficient mini-batch training for stochastic optimization. In *International Conference on Knowledge Discovery and Data Mining*, pages 661–670. ACM, 2014.

Tao Lin, Sebastian U Stich, and Martin Jaggi. Don't use large mini-batches, use local SGD. *arXiv preprint arXiv:1808.07217*, 2018.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Skq89Scxx`.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *International Conference on Machine Learning*, pages 3331–3340, 2018.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417, 2015.

Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In *Doklady AN USSR*, volume 269, pages 543–547, 1983.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From foundations to algorithms*. Cambridge University Press, 2014. URL `https://books.google.com/books?id=OE9etAEACAAJ`.

Ohad Shamir. Without-replacement sampling for stochastic gradient methods. In *Advances in Neural Information Processing Systems*, pages 46–54, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Samuel L. Smith and Quoc V. Le. A Bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BJij4yg0Z`.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pages 1139–1147, 2013.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *Conference on Computer Vision and Pattern Recognition*, pages 2818–2826. IEEE, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Conference on Computer Vision and Pattern Recognition*, pages 6575–6583. IEEE, 2017.

D Randall Wilson and Tony R Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10):1429–1451, 2003.

Yuhuai Wu, Mengye Ren, Renjie Liao, and Roger Grosse. Understanding short-horizon bias in stochastic meta-optimization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H1MczcgR-`.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. Gradient diversity: a key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, 2018. URL `http://proceedings.mlr.press/v84/yin18a.html`.

Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, and Kurt Keutzer. ImageNet training in minutes. *arXiv preprint arXiv:1709.05011*, 2017.

Journal of Machine Learning Research 20 (2019) 1-41 Submitted 11/18; Revised 4/19; Published 6/19

Distributed Inference for Linear Support Vector Machine

Xiaozhou Wang WANGXIAOZHOU@SJTU.EDU.CN
School of Mathematical Sciences
Shanghai Jiao Tong University, Shanghai, 200240, China

Zhuoyi Yang ZYANG@STERN.NYU.EDU
Stern School of Business
New York University, New York, NY 10012, USA

Xi Chen XCHEN3@STERN.NYU.EDU
Stern School of Business
New York University, New York, NY 10012, USA

Weidong Liu WEIDONGL@SJTU.EDU.CN
School of Mathematical Sciences and MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University, Shanghai, 200240, China

Editor: Qiang Liu

Abstract

The growing size of modern data brings many new challenges to existing statistical inference methodologies and theories, and calls for the development of distributed inferential approaches. This paper studies distributed inference for linear support vector machine (SVM) for the binary classification task. Despite a vast literature on SVM, much less is known about the inferential properties of SVM, especially in a distributed setting. In this paper, we propose a multi-round distributed linear-type (MDL) estimator for conducting inference for linear SVM. The proposed estimator is computationally efficient. In particular, it only requires an initial SVM estimator and then successively refines the estimator by solving simple weighted least squares problem. Theoretically, we establish the Bahadur representation of the estimator. Based on the representation, the asymptotic normality is further derived, which shows that the MDL estimator achieves the optimal statistical efficiency, i.e., the same efficiency as the classical linear SVM applying to the entire data set in a single machine setup. Moreover, our asymptotic result avoids the condition on the number of machines or data batches, which is commonly assumed in distributed estimation literature, and allows the case of diverging dimension. We provide simulation studies to demonstrate the performance of the proposed MDL estimator.

Keywords: Linear support vector machine, distributed inference, Bahadur representation, asymptotic theory

1. Introduction

The development of modern technology has enabled data collection of unprecedented size. Very large-scale data sets, such as collections of images, text, transactional data, sensor network data, are becoming prevailing, with examples ranging from digitalized books and newspapers, to collections of images on Instagram, to data generated by large-scale networks of sensing devices or mobile robots. The scale of these data brings new challenges to

©2019 Xiaozhou Wang, Zhuoyi Yang, Xi Chen and Weidong Liu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-B801.html.

traditional statistical estimation and inference methods, particularly in terms of memory restriction and computation time. For example, a large text corpus easily exceeds the memory limitation and thus cannot be loaded into memory all at once. In a sensor network, the data are collected by each sensor in a distributed manner. It will incur an excessively high communication cost if we transfer all the data into a center for processing, and moreover, the center might not have enough memory to store all the data collected from different sensors. In addition to memory constraints, these large-scale data sets also pose challenges in computation. It will be computationally very expensive to directly apply an off-the-shelf optimization solver for computing the maximum likelihood estimator (or empirical risk minimizer) on the entire data set. These challenges call for new statistical inference approaches that are able to not only handle large-scale data sets efficiently, but also achieve the same statistical efficiency as classical approaches.

In this paper, we study the problem of distributed inference for linear support vector machine (SVM). SVM, introduced by Cortes and Vapnik (1995), has been one of the most popular classifiers in statistical machine learning, which finds a wide range of applications in image analysis, medicine, finance, and other domains. Due to the importance of SVM, various parallel SVM algorithms have been proposed in machine learning literature; see, e.g., Graf et al. (2005); Forero et al. (2010); Zhu et al. (2008); Hsieh et al. (2014) and an overview in Wang and Zhou (2012). However, these algorithms mainly focus on addressing the computational issue for SVM, i.e., developing a parallel optimization procedure to minimize the objective function of SVM that is defined on given finite samples. In contrast, our paper aims to address the statistical inference problem, which is fundamentally different. More precisely, the task of distributed inference is to construct an estimator for the *population risk minimizer* in a distributed setting and to characterize its *asymptotic behavior* (e.g., establishing its limiting distribution).

As the size of data becomes increasingly large, distributed inference has received a lot of attentions and algorithms have been proposed for various problems (please see the related work Section 2 and references therein for more details). However, the problem of SVM possesses its own unique challenges in distributed inference. First, SVM is a classification problem that involves binary outputs $\{-1, 1\}$. Thus, as compared to regression problems, the noise structure in SVM is different and more complicated, which brings new technical challenges. We will elaborate this point with more details in Remark 1. Second, the hinge loss in SVM is non-smooth. Third, instead of considering the fixed dimension p as in many existing theories on asymptotic properties of SVM parameters (see, e.g., Lin, 1999; Zhang, 2004; Blanchard et al., 2008; Koo et al., 2008), we aim to study the diverging p case, i.e., $p \to \infty$ as the sample size $n \to \infty$.

To address aforementioned challenges, we focus ourselves on the distributed inference for linear SVM, as the first step to the study of distributed inference for more general SVM.[1] Our goal is three-fold:

1. Our result relies on the Bahadur representation of the linear SVM estimator (see, e.g., Koo et al., 2008). For general SVM, to the best of our knowledge, the Bahadur representation in a single machine setting is still open, which has to be developed before investigating distributed inference for general SVM. Thus, we leave this for future investigation.

1. The obtained estimator should achieve the same statistical efficiency as merging all the data together. That is, the distributed inference should not lose any statistical efficiency as compared to the "oracle" single machine setting.

2. We aim to avoid any condition on the number of machines (or the number of data batches). Although this condition is widely assumed in distributed inference literature (see Lian and Fan, 2017 and Section 2 for more details), removing such a condition will make the results more useful in cases when the size of the entire data set is much larger than the memory size or in applications of sensor networks with a large number of sensors.

3. The proposed algorithm should be computationally efficient.

To simultaneously achieve these three goals, we develop a multi-round distributed linear-type (MDL) estimator for linear SVM. In particular, by smoothing the hinge loss using a special kernel smoothing technique adopted from the quantile regression literature (Horowitz, 1998; Pang et al., 2012; Chen et al., 2018), we first introduce a linear-type estimator in a single machine setup. Our linear-type estimator requires a consistent initial SVM estimator that can be easily obtained by solving SVM on one local machine. Given the initial estimator $\widetilde{\boldsymbol{\beta}}_0$, the linear-type estimator has a simple and explicit formula that greatly facilitates the distributed computing. Roughly speaking, given n samples $(y_i, \boldsymbol{X}_i)$ for $i = 1, \ldots, n$, our linear-type estimator takes the form of "weighted least squares":

$$\widetilde{\boldsymbol{\beta}} = \underbrace{\Big[\frac{1}{n}\sum_{i=1}^{n} u_i(y_i, \boldsymbol{X}_i, \widetilde{\boldsymbol{\beta}}_0)\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}\Big]}_{A_1}{}^{-1}\underbrace{\Big\{\frac{1}{n}\sum_{i=1}^{n} v_i(y_i, \boldsymbol{X}_i, \widetilde{\boldsymbol{\beta}}_0)y_i\boldsymbol{X}_i - \boldsymbol{w}(\widetilde{\boldsymbol{\beta}}_0)\Big\}}_{A_2}, \tag{1}$$

where the term A_1 is a weighted gram matrix and $u_i(y_i, \boldsymbol{X}_i, \widetilde{\boldsymbol{\beta}}_0) \in \mathbb{R}$ is the weight that only depends on the i-th data $(y_i, \boldsymbol{X}_i)$ and $\widetilde{\boldsymbol{\beta}}_0$. In the vector A_2, $\boldsymbol{w}(\widetilde{\boldsymbol{\beta}}_0)$ is a fixed vector that only depends on $\widetilde{\boldsymbol{\beta}}_0$ and $v_i(y_i, \boldsymbol{X}_i, \widetilde{\boldsymbol{\beta}}_0) \in \mathbb{R}$ is the weight that only depends on $(y_i, \boldsymbol{X}_i, \widetilde{\boldsymbol{\beta}}_0)$. The formula in (1) has a similar structure as weighted least squares, and thus can be easily computed in a distributed environment (noting that each term in A_1 and A_2 only involves the i-th data point $(y_i, \boldsymbol{X}_i)$ and there is no interaction term in Equation 1). In addition, the linear-type estimator in (1) can be efficiently computed by solving a linear equation system (instead of computing matrix inversion explicitly), which is computationally more attractive than solving the non-smooth optimization in the original linear SVM formulation.

The linear-type estimator can easily refine itself by using the $\widetilde{\boldsymbol{\beta}}$ on the left hand side of (1) as the initial estimator. In other words, we can obtain a new linear-type estimator by recomputing the right hand side of (1) using $\widetilde{\boldsymbol{\beta}}$ as the initial estimator. By successively refining the initial estimator for q rounds/iterations, we could obtain the final multi-round distributed linear-type (MDL) estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$. The estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$ not only has its advantage in terms of computation in a distributed environment, but also has describable statistical properties. In particular, with a small number q, the estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$ is able to achieve the optimal statistical efficiency, that is, the same efficiency as the classical linear SVM estimator computed on the entire data set. To establish the limiting distribution and statistical efficiency results, we first develop the Bahadur representation of our MDL estimator of SVM

(see Theorem 1). Then the asymptotic normality follows immediately from the Bahadur representation. It is worthwhile noting that the Bahadur representation (see, e.g., Bahadur, 1966; Koenker and Bassett Jr, 1978; Chaudhuri, 1991) provides an important characterization of the asymptotic behavior of an estimator. For the original linear SVM formulation, Koo et al. (2008) first established the Bahadur representation. In this paper, we establish the Bahadur representation of our multi-round distributed linear-type estimator.

Finally, it is worthwhile noting that our algorithm is similar to a recently developed algorithm for distributed quantile regression (Chen et al., 2018), where both algorithms rely on a kernel smoothing technique and linear-type estimators. However, the technique for establishing the theoretical property for linear SVM is quite different from that for quantile regression. The difference and new technical challenges in linear SVM will be illustrated in Remark 1 (see Section 3).

The rest of the paper is organized as follows. In Section 2, we provide a brief overview of related works. Section 3 first introduces the problem setup and then describes the proposed linear-type estimator and MDL estimator for linear SVM. In Section 4, the main theoretical results are given. Section 5 provides the simulation studies to illustrate the performance of MDL estimator of SVM. Conclusions and future works are given in Section 6. We provide the proofs of our theoretical results in Appendix A.

2. Related Works

In distributed inference literature, the divide-and-conquer (DC) approach is one of the most popular approaches and has been applied to a wide range of statistical problems. In the standard DC framework, the entire data set of n i.i.d. samples is evenly split into N batches or distributed on N local machines. Each machine computes a local estimator using the $m = n/N$ local samples. Then, the final estimator is obtained by averaging local estimators. The performance of the DC approach (or its variants) has been investigated on many statistical problems, such as density parameter estimation (Li et al., 2013), kernel ridge regression (Zhang et al., 2015), high-dimensional linear regression (Lee et al., 2017) and generalized linear models (Chen and Xie, 2014; Battey et al., 2018), semi-parametric partial linear models (Zhao et al., 2016), quantile regression (Volgushev et al., 2017; Chen et al., 2018), principal component analysis (Fan et al., 2017), one-step estimator (Huang and Huo, 2015), high-dimensional SVM (Lian and Fan, 2017), M-estimators with cubic rate (Shi et al., 2017), and some non-standard problems where rates of convergence are slower than $n^{1/2}$ and limit distributions are non-Gaussian (Banerjee et al., 2018). On one hand, the DC approach enjoys low communication cost since it only requires one-shot communication (i.e., taking the average of local estimators). On the other hand, almost all the existing work on DC approaches requires a constraint on the number of machines. The main reason is that the averaging only reduces the variance but not the bias of each local estimator. To make the variance the dominating term in the final estimator constructed by taking averaging, the constraint on the number of machines is unavoidable. In particular, in the DC approach for linear SVM in Lian and Fan (2017), the number of machines N has to satisfy the condition $N \leq (n/\log(p))^{1/3}$ (see Remark 1 in Lian and Fan, 2017). As a comparison, our MDL estimator that involves multi-round aggregations successfully eliminates this condition on the number of machines.

In fact, to relax this constraint, several multi-round distributed methods have been recently developed (see Wang et al., 2017; Jordan et al., 2018). In particular, the key idea behind these methods is to approximate the Newton step by using the local Hessian matrix computed on a local machine. However, to compute the local Hessian matrix, their methods require the second-order differentiability on the loss function and thus are not applicable to problems involving non-smooth loss such as SVM.

The second line of the related research is the support vector machine (SVM). Since it was proposed by Cortes and Vapnik (1995), there is a large body of literature on SVM from both machine learning and statistics community. The readers might refer to the books (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002; Steinwart and Christmann, 2008) for a comprehensive review of SVM. In this section, we briefly mention a few relevant works on the statistical properties of linear SVM. In particular, the Bayes risk consistency and the rate of convergence of SVM have been extensively investigated (see, e.g., Lin, 1999; Zhang, 2004; Blanchard et al., 2008; Bartlett et al., 2006). These works mainly concern the asymptotic risk. For the asymptotic properties of underlying coefficients, Koo et al. (2008) first established the Bahadur representation of linear SVM under the fixed p setting. Jiang et al. (2008) proposed interval estimators for the prediction error for general SVM. For the large p case, there are two common settings. One assumes that p grows to infinity at a slower rate than (or linear in) the sample size n but without any sparsity assumption. Our paper also belongs to this setup. Under this setup, Huang (2017) investigated the angle between the normal direction vectors of SVM separating hyperplane and corresponding Bayes optimal separating hyperplane under spiked population models. Another line of research considers high-dimensional SVM under a certain sparsity assumption on underlying coefficients. Under this setup, Peng et al. (2016) established the error bound in L_1 norm. Zhang et al. (2016a) and Zhang et al. (2016b) investigated the variable selection problem in linear SVM.

3. Methodology

3.1. Preliminaries

In a standard binary classification problem setting, we consider a pair of random variables $\{\boldsymbol{X}, Y\}$ with $\boldsymbol{X} \in \mathcal{X} \subseteq \mathbb{R}^p$ and $Y \in \{-1, 1\}$. The marginal distribution of Y is given by $\mathbb{P}(Y = 1) = \pi_+$ and $\mathbb{P}(Y = -1) = \pi_-$ where $\pi_+, \pi_- > 0$ and $\pi_+ + \pi_- = 1$. We assume that the random vector $\boldsymbol{X}$ has a continuous distribution on $\mathcal{X}$ given Y. Let $\{\boldsymbol{X}_i, y_i\}_{i=1,\dots,n}$ be i.i.d. samples drawn from the joint distribution of random variables $\{\boldsymbol{X}, Y\}$. In the linear classification problem, a hyperplane is defined by $\beta_0 + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta} = 0$ with $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_p)^{\mathrm{T}}$. Define $\widetilde{\boldsymbol{X}} = (1, X_1, ..., X_p)^{\mathrm{T}}$ and the coefficient vector $\widetilde{\boldsymbol{\beta}} = (\beta_0, \beta_1, ..., \beta_p)^{\mathrm{T}}$. For convenience purpose we also define $l(\boldsymbol{X}; \widetilde{\boldsymbol{\beta}}) = \beta_0 + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta} = \widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}$. In this paper we consider the standard non-separable SVM formulation, which takes the following form

$$f_{\lambda,n}(\widetilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n}\Big(1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}})\Big)_+ + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2, \tag{2}$$

$$\widetilde{\boldsymbol{\beta}}_{\text{SVM all}} = \underset{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}}{\arg\min}\, f_{\lambda,n}(\widetilde{\boldsymbol{\beta}}). \tag{3}$$

Here $(u)_+ = \max(u, 0)$ is the hinge loss, $\lambda > 0$ is the regularization parameter and $\|\cdot\|_2$ denotes the Euclidean norm of a vector. We note that we do not penalize the first coordinate β_0 and thus the regularization is only imposed on $\boldsymbol{\beta}$ instead of $\widetilde{\boldsymbol{\beta}}$. Throughout this paper, for any $(p+1)$-dimensional parameter vector $\widetilde{\boldsymbol{\alpha}}$, we will use $\boldsymbol{\alpha}$ to denote the subvector of $\widetilde{\boldsymbol{\alpha}}$ without the first coordinate, and only $\boldsymbol{\alpha}$ will appear in the regularization term.

The corresponding population loss function is defined as

$$L(\widetilde{\boldsymbol{\beta}}) = \mathbb{E}[1 - Yl(\boldsymbol{X}; \widetilde{\boldsymbol{\beta}})]_+.$$

We denote the minimizer for the population loss by

$$\widetilde{\boldsymbol{\beta}}^* = \underset{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}}{\arg\min}\, \mathbb{E}[1 - Yl(\boldsymbol{X}; \widetilde{\boldsymbol{\beta}})]_+. \tag{4}$$

Koo et al. (2008) proved that under some mild conditions (see Koo et al., 2008 Theorem 1,2), there exists a unique minimizer for (4) and it is nonzero (i.e., $\widetilde{\boldsymbol{\beta}}^* \neq \mathbf{0}$). We assume that these conditions hold throughout the paper. The minimizer $\widetilde{\boldsymbol{\beta}}^*$ of the population loss function will serve as the "true parameter" in our estimation problem and the goal is to construct an estimator and make inference of $\widetilde{\boldsymbol{\beta}}^*$. We further define some useful quantities as follows:

$$\epsilon = 1 - Yl(\boldsymbol{X}, \widetilde{\boldsymbol{\beta}}^*), \text{ and } \epsilon_i = 1 - y_i l(\boldsymbol{X}_i, \widetilde{\boldsymbol{\beta}}^*).$$

The reason why we use the notation ϵ is because it plays a similar role in the theoretical analysis as the noise term in a standard regression problem. However, as we will show in Section 3 and 4, the behavior of ϵ is quite different from the noise in a classical regression setting since it does not have a continuous density function (see Remark 1). Next, denote by $\delta(\cdot)$ the Dirac delta function, we define

$$\begin{aligned} \boldsymbol{S}(\widetilde{\boldsymbol{\beta}}) &= -\mathbb{E}[I\{1 - Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}} \geq 0\}Y\widetilde{\boldsymbol{X}}], \\ \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}) &= \mathbb{E}[\delta(1 - Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}})\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\mathrm{T}}], \end{aligned} \tag{5}$$

where $I\{\cdot\}$ is the indicator function.

The quantities $\boldsymbol{S}(\widetilde{\boldsymbol{\beta}})$ and $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}})$ can be viewed as the gradient and Hessian matrix of $L(\widetilde{\boldsymbol{\beta}})$ and we assume that the smallest eigenvalue of $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)$ is bounded away from 0. In fact these assumptions can be verified under some regular conditions (see Koo et al., 2008 Lemma 2, Lemma 3 and Lemma 5 for details) and are common in SVM literature (e.g., Zhang et al., 2016b Condition 2 and 6).

3.2. A Linear-type Estimator for SVM

In this section, we first propose a linear-type estimator for SVM on a single machine which can be later extended to a distributed algorithm. The main challenge in solving the optimization problem in (2) is that the objective function is non-differentiable due to the appearance of hinge loss. Motivated by a smoothing technique from quantile regression literature (see, e.g., Chen et al., 2018; Horowitz, 1998; Pang et al., 2012), we consider a smooth function $H(\cdot)$ satisfying $H(u) = 1$ if $u \geq 1$ and $H(u) = 0$ if $u \leq -1$. We replace the hinge loss with its smooth approximation $K_h(u) = uH(\frac{u}{h})$, where h is the bandwidth. As

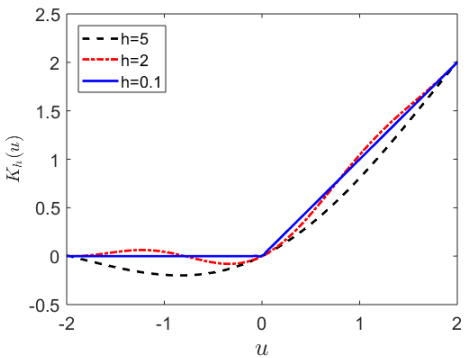


Figure 1: An example of the smoothed hinge loss function K_h with different bandwidth h. See Section 5 for details in the construction of $H(\cdot)$.

the bandwidth $h \to 0$, $H(\frac{u}{h})$ and $\frac{1}{h}H'(\frac{u}{h})$ approaches the indicator function $I\{u \geq 0\}$ and Dirac delta function $\delta(u)$ respectively, and $K_h(u)$ approximates the hinge loss $\max(u, 0)$ (see Figure 1 for an example of K_h with different bandwidths). To motivate our linear-type estimator, we first consider the following estimator with the non-smooth hinge loss in linear SVM replaced by its smooth approximation:

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}_h &= \arg\min_{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}} \frac{1}{n}\sum_{i=1}^{n}\left[1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}})\right] H\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}})}{h}\right) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 \\ &= \arg\min_{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}} \frac{1}{n}\sum_{i=1}^{n} K_h(1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}})) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2.\end{aligned} \tag{6}$$

Since the objective function is differentiable and $\frac{dK_h(x)}{dx} = H(x/h) + \frac{x}{h}H'(x/h)$, by the first order condition (i.e., setting the derivative of the objective function in (6) to zero), $\widetilde{\boldsymbol{\beta}}_h$ satisfies

$$\frac{1}{n}\sum_{i=1}^{n}(-y_i\widetilde{\boldsymbol{X}}_i)\left[H\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_h)}{h}\right) + \frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_h)}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_h)}{h}\right)\right] + \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}_h\end{pmatrix} = 0.$$

We first rearrange the equation and express $\widetilde{\boldsymbol{\beta}}_h$ by

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}_h = &\left[\frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{X}}_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\frac{1}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_h)}{h}\right)\right]^{-1} \\ &\times\left\{\frac{1}{n}\sum_{i=1}^{n}y_i\widetilde{\boldsymbol{X}}_i\left[H\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_h)}{h}\right) + \frac{1}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_h)}{h}\right)\right] - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}_h\end{pmatrix}\right\}.\end{aligned} \tag{7}$$

This fixed-point form formula for $\widetilde{\boldsymbol{\beta}}_h$ cannot be solved explicitly since $\widetilde{\boldsymbol{\beta}}_h$ appears on both sides of (7). Nevertheless, $\widetilde{\boldsymbol{\beta}}_h$ is not our final estimator and is mainly introduced to motivate our estimator. The key idea is to replace $\widetilde{\boldsymbol{\beta}}_h$ on the right hand side of (7) by a consistent initial estimator $\widetilde{\boldsymbol{\beta}}_0$ (e.g., $\widetilde{\boldsymbol{\beta}}_0$ can be constructed by solving a linear SVM on a small batch of samples). Then, we obtain the following linear-type estimator for $\widetilde{\boldsymbol{\beta}}^*$:

$$\begin{aligned}\widetilde{\boldsymbol{\beta}} = &\left[\frac{1}{n}\sum_{i=1}^{n}\widetilde{\boldsymbol{X}}_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\frac{1}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_0)}{h}\right)\right]^{-1} \\ &\times\left\{\frac{1}{n}\sum_{i=1}^{n}y_i\widetilde{\boldsymbol{X}}_i\left[H\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_0)}{h}\right) + \frac{1}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_0)}{h}\right)\right] - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}_0\end{pmatrix}\right\}.\end{aligned} \tag{8}$$

Notice that (8) has a similar structure as weighted least squares (see the explanations in the paragraph below (1) in the introduction). As shown in the following section, this weighted least squares formulation can be computed efficiently in a distributed setting.

3.3. Multi-Round Distributed Linear-type (MDL) Estimator

It is important to notice that given the initial estimator $\widetilde{\boldsymbol{\beta}}_0$, the linear-type estimator in (8) only involves summation of matrices and vectors computed for each individual data point. Therefore based on (8), we will construct a multi-round distributed linear-type estimator (MDL estimator) that can be efficiently implemented in a distributed setting.

First, let us assume that the total data indices $\{1, ..., n\}$ are divided into N subsets $\{\mathcal{H}_1, ..., \mathcal{H}_N\}$ with equal size $m = n/N$. Denote by $\mathcal{D}_k = \{(\boldsymbol{X}_i, y_i) : i \in \mathcal{H}_k\}$ the data in the k-th local machine. In order to compute $\widetilde{\boldsymbol{\beta}}$, for each batch of data $\mathcal{D}_k$ for $k = 1, ..., N$, we define the following quantities

$$\begin{aligned}\boldsymbol{U}_k &= \frac{1}{n}\sum_{i \in \mathcal{H}_k} y_i\widetilde{\boldsymbol{X}}_i\left[H\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_0)}{h}\right) + \frac{1}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_0)}{h}\right)\right], \\ \boldsymbol{V}_k &= \frac{1}{n}\sum_{i \in \mathcal{H}_k}\widetilde{\boldsymbol{X}}_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\frac{1}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}}_0)}{h}\right).\end{aligned} \tag{9}$$

Given $\widetilde{\boldsymbol{\beta}}_0$, the quantities $\boldsymbol{U}_k, \boldsymbol{V}_k$ can be computed independently in each machine and only $(\boldsymbol{U}_k, \boldsymbol{V}_k)$ has to be stored and transferred to the central machine. Then after receiving $(\boldsymbol{U}_k, \boldsymbol{V}_k)$ from all the machines, the central machine can aggregate the data and compute the estimator by

$$\widetilde{\boldsymbol{\beta}}^{(1)} = \left(\sum_{k=1}^{N}\boldsymbol{V}_k\right)^{-1}\left(\sum_{k=1}^{N}\boldsymbol{U}_k - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}_0\end{pmatrix}\right).$$

Then $\widetilde{\boldsymbol{\beta}}^{(1)}$ can be sent to all the machines to repeat the whole process to construct $\widetilde{\boldsymbol{\beta}}^{(2)}$ using $\widetilde{\boldsymbol{\beta}}^{(1)}$ as the new initial estimator. The algorithm is repeated q times for a pre-specified q (see Equation 21 for details), and $\widetilde{\boldsymbol{\beta}}^{(q)}$ is taken to be the final estimator (see Algorithm 1 for details). We name this estimator as the multi-round distributed linear-type (MDL) estimator.

Algorithm 1 Multi-round distributed linear-type estimator (MDL) for SVM

Input: Samples stored in the machines $\{\mathcal{D}_1, ..., \mathcal{D}_N\}$, the number of iterations q, smooth function H, bandwidths $\{h_1, ..., h_q\}$ and regularization parameter λ.

for $g = 1, \ldots, q$ **do**
  if $g = 1$ **then**
    Compute the initial estimator based on $\mathcal{D}_1$:

$$\widetilde{\boldsymbol{\beta}}_0 = \underset{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{m} \sum_{i \in \mathcal{H}_1} \left(1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}})\right)_+ .$$

  else
    $\widetilde{\boldsymbol{\beta}}_0 = \widetilde{\boldsymbol{\beta}}^{(g-1)}$
  end if
  $\widetilde{\boldsymbol{\beta}}_0$ is transferred to all the local machines.
  for $k = 1, \ldots, N$ **do**
    Compute $(\boldsymbol{U}_k, \boldsymbol{V}_k)$ according to (9) with data in $\mathcal{D}_k$ using the bandwidth h_g.
    Transfer $(\boldsymbol{U}_k, \boldsymbol{V}_k)$ to the central machine.
  end for
  The central machine computes the estimator $\widetilde{\boldsymbol{\beta}}^{(g)}$ by

$$\widetilde{\boldsymbol{\beta}}^{(g)} = \left(\sum_{k=1}^{N} \boldsymbol{V}_k\right)^{-1} \left(\sum_{k=1}^{N} \boldsymbol{U}_k - \lambda \begin{pmatrix} 0 \\ \boldsymbol{\beta}_0 \end{pmatrix}\right). \tag{10}$$

end for

Output: The final MDL estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$.

We notice that instead of computing matrix inversion $\left(\sum_{k=1}^{N} \boldsymbol{V}_k\right)^{-1}$ in every iteration which has a computation cost $O(p^3)$, one only needs to solve a linear system in (10). Linear system has been studied in numeric optimization for several decades and many efficient algorithms have been developed, such as conjugate gradient method (Hestenes and Stiefel, 1952). We also notice that we only have to solve a single optimization problem on one local machine to compute the initial estimator. Then at each iteration, only matrix multiplication and summation needs to be computed locally which makes the algorithm computationally efficient. It is worthwhile noticing that according to Theorem 2 in Section 4, under some mild conditions, if we choose $h := h_g = \max\left(\lambda, \sqrt{p/n}, (p/m)^{2^{g-2}}\right)$ for $1 \le g \le q$, the MDL estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$ achieves optimal statistical efficiency as long as q satisfies (21), which is usually a small number. Therefore, a few rounds of iterations would guarantee good performance for the MDL estimator.

For the choice of the initial estimator in the first iteration, we propose to construct it by solving the original SVM optimization (2) only on a small batch of samples (e.g., the samples on the first machine $\mathcal{D}_1$). The estimator $\widetilde{\boldsymbol{\beta}}_0$ is only a crude estimator for $\widetilde{\boldsymbol{\beta}}^*$, but we will prove later that it is enough for the algorithm to produce an estimator with

optimal statistical efficiency under some regularity conditions. In particular, if we compute the initial estimator in the first round on the first batch of data, we will solve the following optimization problem

$$\widetilde{\boldsymbol{\beta}}_0 = \underset{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{m} \sum_{i \in \mathcal{H}_1} \left(1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}})\right)_+ .$$

Then we have the following proposition from Zhang et al. (2016b).

Proposition 1 (Zhang et al. (2016b)) *Under conditions (C1)-(C6) in Zhang et al. (2016b), we have*

$$\|\widetilde{\boldsymbol{\beta}}_0 - \widetilde{\boldsymbol{\beta}}^*\|_2 = O_{\mathbb{P}}(\sqrt{p/m}).$$

According to our Theorem 1, the initial estimator $\widetilde{\boldsymbol{\beta}}_0$ needs to satisfy $\|\widetilde{\boldsymbol{\beta}}_0 - \widetilde{\boldsymbol{\beta}}^*\|_2 = O_{\mathbb{P}}(\sqrt{p/m})$, and therefore the estimator computed on the first machine is a valid initial estimator. On the other hand, one can always use different approaches to construct the initial estimator $\widetilde{\boldsymbol{\beta}}_0$ as long as it is a consistent estimator.

Remark 1 We note that although Algorithm 1 has a similar form as the DC-LEQR estimator for quantile regression (QR) in Chen et al. (2018), the structures of the SVM and QR problems are fundamentally different and thus the theoretical development for establishing the Bahadur representations for SVM is more challenging. To see that, let us recall the quantile regression model:

$$Y = \widetilde{\boldsymbol{X}}^{\mathrm{T}} \widetilde{\boldsymbol{\beta}}^* + \epsilon, \tag{11}$$

where ϵ is the unobserved random noise satisfying $\Pr(\epsilon \le 0 | \widetilde{\boldsymbol{X}}) = \tau$ and τ is known as the quantile level. The asymptotic results of QR estimators heavily rely on the Lipschitz continuity assumption on the conditional density $f(\epsilon | \widetilde{\boldsymbol{X}})$ of ϵ given $\widetilde{\boldsymbol{X}}$, which has been assumed in almost all existing literature. In the SVM problem, the quantity $\epsilon := 1 - Y \widetilde{\boldsymbol{X}}^{\mathrm{T}} \widetilde{\boldsymbol{\beta}}^*$ plays a similar role as the noise in a regression problem. However, since $Y \in \{-1, 1\}$ is binary, the conditional distribution $f(\epsilon | \widetilde{\boldsymbol{X}})$ becomes a two-point distribution, which no longer has a density function. To address this challenge and derive the asymptotic behavior of SVM, we directly work on the joint distribution of ϵ and $\widetilde{\boldsymbol{X}}$. As the dimension of $\widetilde{\boldsymbol{X}}$ (i.e., $p+1$) can go to infinity, we use a slicing technique by considering the one-dimensional marginal distribution of $\widetilde{\boldsymbol{X}}$ (see Condition (C2) and proof of Theorem 1 for more details).

3.4. Communication-Efficient Implementation

In this section, we discuss a communication-efficient implementation of the proposed MDL estimator. Note that in Algorithm 1, each local machine transmits a $(p+1)$-by-$(p+1)$ matrix $\boldsymbol{V}_k$ to the central machine at each iteration. In fact, the communication of $(p+1)$-by-$(p+1)$ matrices can be avoided by using the approximate Newton method (see, e.g., Shamir et al., 2014; Wang et al., 2017; Jordan et al., 2018). Instead of transmitting the Hessian matrix $\boldsymbol{V}_k$, we will only use the local Hessian matrix $\boldsymbol{V}_1$ computed on the first machine. More specifically, the estimator $\widetilde{\boldsymbol{\beta}}^{(g)}$ in (10) essentially solves the minimization problem

$$\underset{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{2} \widetilde{\boldsymbol{\beta}}^{\mathrm{T}} \left(\sum_{k=1}^{N} \boldsymbol{V}_k\right) \widetilde{\boldsymbol{\beta}} - \widetilde{\boldsymbol{\beta}}^{\mathrm{T}} \left(\sum_{k=1}^{N} \boldsymbol{U}_k - \lambda \begin{pmatrix} 0 \\ \boldsymbol{\beta}_0 \end{pmatrix}\right). \tag{12}$$

The approximate Newton method uses the following iterations to solve the above minimization problem:

$$\widetilde{\boldsymbol{\beta}}^{(g,t)} = \widetilde{\boldsymbol{\beta}}^{(g,t-1)} - \left(N\widehat{\boldsymbol{V}}_1\right)^{-1}\left(\sum_{k=1}^{N}\left(\boldsymbol{V}_k\widetilde{\boldsymbol{\beta}}^{(g,t-1)} - \boldsymbol{U}_k\right) + \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}_0\end{pmatrix}\right), \quad \widetilde{\boldsymbol{\beta}}^{(g,0)} = \widetilde{\boldsymbol{\beta}}_0, \tag{13}$$

as an inner iterative procedure, which approximately solves the equation (10). In (13), we let $\widehat{\boldsymbol{V}}_1 = \boldsymbol{V}_1$ in (9) with the bandwidth $h = \sqrt{p/m}$. The matrix $N\widehat{\boldsymbol{V}}_1$ is used to approximate the Hessian matrix $\sum_{k=1}^{N}\boldsymbol{V}_k$ in (12). To compute the minimizer of (13), we note that the matrix $\widehat{\boldsymbol{V}}_1$ only involves the data on the first machine, and thus there is no need to communicate $(p+1)$-by-$(p+1)$ matrices to compute (13). In fact, each local machine only transmits a $(p+1)$-by-1 vector $\boldsymbol{V}_k\widetilde{\boldsymbol{\beta}}^{(g,t-1)} - \boldsymbol{U}_k$ to the central machine. We present the entire communication-efficient implementation in Algorithm 2.

Recall that $\widetilde{\boldsymbol{\beta}}^{(g)}$ is the estimator defined in (10). It is easy to show that $\widetilde{\boldsymbol{\beta}}^{(g,t)}$ in (13) converges to $\widetilde{\boldsymbol{\beta}}^{(g)}$ at a super-linear rate:

$$\|\widetilde{\boldsymbol{\beta}}^{(g,t)} - \widetilde{\boldsymbol{\beta}}^{(g)}\|_2 \le \left\|I - \widehat{\boldsymbol{V}}_1^{-1}\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{V}_k\right\|\|\widetilde{\boldsymbol{\beta}}^{(g,t-1)} - \widetilde{\boldsymbol{\beta}}^{(g)}\|_2,$$

where $\|\cdot\|$ denotes the spectral norm of a matrix. By repeatedly applying the argument, we have the following proposition, whose proof is relegated to Appendix A.3.

Proposition 2 *Assume that the conditions of Theorem 1 in Section 4 hold. Suppose that $p = O(m^{\nu})$ for some $0 < \nu < 1$ and $n = O(m^{A})$ for some $A > 0$. We have*

$$\|\widetilde{\boldsymbol{\beta}}^{(g,t)} - \widetilde{\boldsymbol{\beta}}^{(g)}\|_2 = O_{\mathbb{P}}(m^{-\delta t}) \tag{14}$$

for some constant $\delta > 0$, where δ and $O_{\mathbb{P}}(1)$ do not depend on t.

Proposition 2 and the convergence rate of $\widetilde{\boldsymbol{\beta}}^{(g)}$ (see Theorem 1) imply that the inner procedure takes at most constant-valued iterations to achieve the same convergence rate as $\widetilde{\boldsymbol{\beta}}^{(g)}$. Therefore the theoretical results of the MDL estimator in Algorithm 1 (see Theorem 1 and 2 in Section 4) still hold for Algorithm 2. In summary, as compared to Algorithm 1, Algorithm 2 only requires $O(p)$ communication cost for each local machine. Therefore, Algorithm 2 is communicationally more efficient when p is large.

4. Theoretical Results

In this section, we give a Bahadur representation of the MDL estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$ and establish its asymptotic normality result. From (8), the difference between the MDL estimator and the true coefficient can be written as

$$\widetilde{\boldsymbol{\beta}} - \widetilde{\boldsymbol{\beta}}^{*} = \boldsymbol{D}_{n,h}^{-1}\left(\boldsymbol{A}_{n,h} - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}_0\end{pmatrix}\right), \tag{15}$$

where $\boldsymbol{A}_{n,h} = \boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}_0)$, $\boldsymbol{D}_{n,h} = \boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}_0)$, and for any $\widetilde{\boldsymbol{\alpha}}$, $\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\alpha}})$ and $\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\alpha}})$ are defined as follows,

Algorithm 2 Communication-efficient MDL for SVM

Input: Samples stored in the machines $\{\mathcal{D}_1, ..., \mathcal{D}_N\}$, the number of outer iterations q, the number of inner iterations T, smooth function H, bandwidths $\{h_1, ..., h_q\}$ and regularization parameter λ.

for $g = 1, \ldots, q$ **do**
  if $g = 1$ **then**
    Compute the initial estimator based on $\mathcal{D}_1$:

$$\widetilde{\boldsymbol{\beta}}_0 = \arg\min_{\widetilde{\boldsymbol{\beta}}\in\mathbb{R}^{p+1}} \frac{1}{m}\sum_{i\in\mathcal{H}_1}\left(1 - y_i l(\boldsymbol{X}_i;\widetilde{\boldsymbol{\beta}})\right)_+.$$

  else
    $\widetilde{\boldsymbol{\beta}}_0 = \widetilde{\boldsymbol{\beta}}^{(g-1)}$
  end if
  Let $\widetilde{\boldsymbol{\beta}}^{(g,0)} = \widetilde{\boldsymbol{\beta}}_0$.
  for $t = 1, \ldots, T$ **do**
    $\widetilde{\boldsymbol{\beta}}^{(g,t-1)}$ is transferred to all the local machines.
    for $k = 1, \ldots, N$ **do**
      Compute $\boldsymbol{V}_k\widetilde{\boldsymbol{\beta}}^{(g,t-1)} - \boldsymbol{U}_k$ and transfer it to the central machine.
    end for
    The central machine computes $\widetilde{\boldsymbol{\beta}}^{(g,t)}$ by

$$\widetilde{\boldsymbol{\beta}}^{(g,t)} = \widetilde{\boldsymbol{\beta}}^{(g,t-1)} - \left(N\widehat{\boldsymbol{V}}_1\right)^{-1}\left(\sum_{k=1}^{N}\left(\boldsymbol{V}_k\widetilde{\boldsymbol{\beta}}^{(g,t-1)} - \boldsymbol{U}_k\right) + \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}_0\end{pmatrix}\right),$$

    where $\widehat{\boldsymbol{V}}_1$ is defined as $\boldsymbol{V}_1$ in (9) but with the bandwidth $h = \sqrt{p/m}$.
  end for
  Let $\widetilde{\boldsymbol{\beta}}^{(g)} = \widetilde{\boldsymbol{\beta}}^{(g,T)}$.
end for

Output: The final MDL estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$.

$$\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\alpha}}) = \frac{1}{n}\sum_{i=1}^{n} y_i\widetilde{\boldsymbol{X}}_i\left[H\left(\frac{1 - y_i l(\boldsymbol{X}_i;\widetilde{\boldsymbol{\alpha}})}{h}\right) + \frac{1 - y_i l(\boldsymbol{X}_i;\widetilde{\boldsymbol{\beta}}^{*})}{h}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i;\widetilde{\boldsymbol{\alpha}})}{h}\right)\right],$$

$$\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\alpha}}) = \frac{1}{nh}\sum_{i=1}^{n}\widetilde{\boldsymbol{X}}_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}H'\left(\frac{1 - y_i l(\boldsymbol{X}_i;\widetilde{\boldsymbol{\alpha}})}{h}\right).$$

For a good initial estimator $\widetilde{\boldsymbol{\beta}}_0$ which is close to $\widetilde{\boldsymbol{\beta}}^*$, the quantities $\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}_0)$ and $\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}_0)$ are close to $\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}^*)$ and $\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}^*)$. Recall that $\epsilon_i = 1 - y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*$ and we have

$$\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}^*) = \frac{1}{n}\sum_{i=1}^{n} y_i\widetilde{\boldsymbol{X}}_i\left[H\left(\frac{\epsilon_i}{h}\right) + \frac{\epsilon_i}{h}H'\left(\frac{\epsilon_i}{h}\right)\right],$$

$$\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}^*) = \frac{1}{nh}\sum_{i=1}^{n} \widetilde{\boldsymbol{X}}_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}} H'\left(\frac{\epsilon_i}{h}\right).$$

When h is close to zero, the term $H\left(\frac{\epsilon_i}{h}\right)+\frac{\epsilon_i}{h}H'\left(\frac{\epsilon_i}{h}\right)$ in parenthesis of $\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}^*)$ approximates $I\{\epsilon_i \geq 0\}$. Therefore, $\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}^*)$ will be close to $\frac{1}{n}\sum_{i=1}^{n} y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i \geq 0\}$. Moreover, since $\frac{1}{h}H'(\cdot/h)$ approximates Dirac delta function as $h \to 0$, $\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}^*)$ approaches

$$\boldsymbol{D}_n(\widetilde{\boldsymbol{\beta}}^*) = \frac{1}{n}\sum_{i=1}^{n}[\delta(\epsilon_i)\widetilde{\boldsymbol{X}}_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}].$$

When n is large, $\boldsymbol{D}_n(\widetilde{\boldsymbol{\beta}}^*)$ will be close to its corresponding population quantity $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*) = \mathbb{E}[\delta(\epsilon)\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\mathrm{T}}]$ defined in (5).

According to the above argument, when $\widetilde{\boldsymbol{\beta}}_0$ is close to $\widetilde{\boldsymbol{\beta}}^*$, $\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}_0)$ and $\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}_0)$ approximate $\frac{1}{n}\sum_{i=1}^{n} y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i \geq 0\}$ and $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)$, respectively. Therefore, by (15), we would expect $\widetilde{\boldsymbol{\beta}} - \widetilde{\boldsymbol{\beta}}^*$ to be close to the following quantity,

$$\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i \geq 0\} - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}^*\end{pmatrix}\right). \tag{16}$$

We will see later that (16) is exactly the main term of the Bahadur representation of the estimator. Next, we formalize these statements and present the asymptotic properties of $\boldsymbol{A}_{n,h}$ and $\boldsymbol{D}_{n,h}$ in Proposition 3 and 4. The asymptotic properties of the MDL estimator will be provided in Theorem 1. To this end, we first introduce some notations and assumptions for the theoretical result.

Recall that $\boldsymbol{\beta}^* = (\beta_1^*, ..., \beta_p^*)^{\mathrm{T}}$ and for $\boldsymbol{X} = (X_1, ..., X_p)^{\mathrm{T}}$, let $\boldsymbol{X}_{-s}$ be a $(p-1)$-dimensional vector with X_s removed from $\boldsymbol{X}$. Similar notations are used for $\boldsymbol{\beta}$. Since we assumed that $\widetilde{\boldsymbol{\beta}}^* \neq \boldsymbol{0}$, without loss of generality, we assume $\beta_1^* \neq 0$ and its absolute value is lower bounded by some constant $c > 0$ (i.e., $|\beta_1^*| \geq c$). Let f and g be the density functions of $\boldsymbol{X}$ when $Y = 1$ and $Y = -1$ respectively. Let $f(x|\boldsymbol{X}_{-1})$ be the conditional density function of X_1 given $(X_2, ..., X_p)^{\mathrm{T}}$ and $f_{-1}(\boldsymbol{x}_{-1})$ be the joint density of $(X_2, ..., X_p)^{\mathrm{T}}$. Similar notations are used for $g(\cdot)$.

We state some regularity conditions to facilitate theoretical development of asymptotic properties of $\boldsymbol{A}_{n,h}$ and $\boldsymbol{D}_{n,h}$.

(C0) There exists a unique nonzero minimizer $\widetilde{\boldsymbol{\beta}}^*$ for (4) with $\boldsymbol{S}(\widetilde{\boldsymbol{\beta}}^*) = 0$, and $c \leq \lambda_{\min}(\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)) \leq \lambda_{\max}(\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)) \leq c^{-1}$ for some constant $c > 0$.

(C1) $|\beta_1^*| \geq c$ and $\|\widetilde{\boldsymbol{\beta}}^*\|_2 \leq C$ for some constants $c, C > 0$.

(C2) Assume that $\sup_{x\in\mathbb{R}}|f(x|\boldsymbol{X}_{-1})| \leq C$, $\sup_{x\in\mathbb{R}}|f'(x|\boldsymbol{X}_{-1})| \leq C$, $\sup_{x\in\mathbb{R}}|xf'(x|\boldsymbol{X}_{-1})| \leq C$, $\sup_{x\in\mathbb{R}}|xf(x|\boldsymbol{X}_{-1})| \leq C$, $\sup_{x\in\mathbb{R}}|x^2f(x|\boldsymbol{X}_{-1})| \leq C$ and $\sup_{x\in\mathbb{R}}|x^2f'(x|\boldsymbol{X}_{-1})| \leq C$ for some constant $C > 0$. Also assume $\int_{\mathbb{R}}|x|f(x|\boldsymbol{X}_{-1})dx < \infty$. Similar assumptions are made for $g(\cdot)$.

(C3) Assume that $p = o(nh/\log n)$ and $\sup_{\|\boldsymbol{v}\|_2\leq 1}\mathbb{E}\exp(t_0|\boldsymbol{v}^{\mathrm{T}}\boldsymbol{X}|^2) \leq C$ for some $t_0 > 0$ and $C > 0$.

(C4) The smoothing function $H(x)$ satisfies $H(x) = 1$ if $x \geq 1$ and $H(x) = 0$ if $x \leq -1$, and also assume that H is twice differentiable and $H^{(2)}$ is bounded. Moreover, assume that $h = o(1)$.

As we discussed in Section 3.1, condition (C0) is a standard assumption which can be implied by some mild conditions (see Koo et al., 2008 (A1)-(A4)). Conditions (C1) is a mild condition on the boundness of $\widetilde{\boldsymbol{\beta}}^*$. Condition (C2) is a regularity condition on the conditional density of f and g, and it is satisfied by commonly used density functions, e.g., Gaussian distribution and uniform distribution. Condition (C3) is a sub-Gaussian condition on $\boldsymbol{X}$. Condition (C4) is a smoothness condition on the smooth function $H(\cdot)$ and can be easily satisfied by a properly chosen $H(\cdot)$ (e.g., see an example in Section 5).

Under the above conditions, we give Proposition 3 and Proposition 4 for the asymptotic behavior of $\boldsymbol{A}_{n,h}$ and $\boldsymbol{D}_{n,h}$, respectively. Recall that $\epsilon_i = 1 - y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*$ and we have the following propositions. The proofs of all results in this section are relegated to Appendix A.2.

Proposition 3 *Under conditions (C0)-(C4), assume that we have an initial estimator $\widetilde{\boldsymbol{\beta}}_0$ with $\|\widetilde{\boldsymbol{\beta}}_0 - \widetilde{\boldsymbol{\beta}}^*\|_2 = O_{\mathbb{P}}(a_n)$, where a_n is the convergence rate of the initial estimator. We choose the bandwidth such that $a_n = O(h)$, then we have*

$$\left\|\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\beta}}_0) - \frac{1}{n}\sum_{i=1}^{n} y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i \geq 0\}\right\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{ph\log n}{n}} + a_n^2 + h^2\right).$$

Proposition 4 *Suppose the same conditions in Propositions 3 hold, we have*

$$\left\|\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}_0) - \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)\right\| = O_{\mathbb{P}}\left(\sqrt{\frac{p\log n}{nh}} + a_n + h\right).$$

According to the above propositions, with some algebraic manipulations and condition (C0), we have

$$\widetilde{\boldsymbol{\beta}} - \widetilde{\boldsymbol{\beta}}^* = \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i \geq 0\} - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}^*\end{pmatrix}\right) + \boldsymbol{r}_n, \tag{17}$$

with

$$\|\boldsymbol{r}_n\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{p^2\log n}{n^2h}} + \sqrt{\frac{ph\log n}{n}} + a_n^2 + h^2\right). \tag{18}$$

By appropriately choosing the bandwidth h such that it shrinks with a_n at the same rate (see Theorem 1), a_n^2 becomes the dominating term on the right hand side of (18). This implies that by taking one round of refinement, the L_2 norm of $\widetilde{\boldsymbol{\beta}}-\widetilde{\boldsymbol{\beta}}^*$ improves from $O_{\mathbb{P}}(a_n)$ to $O_{\mathbb{P}}(a_n^2)$ (note that $\|\widetilde{\boldsymbol{\beta}}_0-\widetilde{\boldsymbol{\beta}}^*\|_2=O_{\mathbb{P}}(a_n)$, see Proposition 3). Therefore by recursively applying the argument in (17) and setting the obtained estimator as the new initial estimator $\widetilde{\boldsymbol{\beta}}_0$, the algorithm iteratively refines the estimator $\widetilde{\boldsymbol{\beta}}$. This gives the Bahadur representation of our MDL estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$ for q rounds of refinements (see Algorithm 1).

Theorem 1 *Under conditions (C0)-(C4), assume that the initial estimator $\widetilde{\boldsymbol{\beta}}_0$ satisfies $\|\widetilde{\boldsymbol{\beta}}_0-\widetilde{\boldsymbol{\beta}}^*\|_2=O_{\mathbb{P}}(\sqrt{p/m})$. Also, assume $p=O(m/(\log n)^2)$ and $\lambda=O(1/\log n)$. For a given integer $q\geq 1$, let the bandwidth in the g-th iteration be*

$$h:=h_g=\max(\lambda,\sqrt{p/n},(p/m)^{2^{g-2}})$$

for $1\leq g\leq q$. Then we have

$$\widetilde{\boldsymbol{\beta}}^{(q)}-\widetilde{\boldsymbol{\beta}}^*=\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\left(\frac{1}{n}\sum_{i=1}^n y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i\geq 0\}-\lambda\begin{pmatrix}0\\ \boldsymbol{\beta}^*\end{pmatrix}\right)+\boldsymbol{r}_n, \tag{19}$$

with

$$\|\boldsymbol{r}_n\|_2=O_{\mathbb{P}}\left(\sqrt{\frac{ph_q\log n}{n}}+\left(\frac{p}{m}\right)^{2^{q-1}}+\lambda^2\right). \tag{20}$$

It is worthwhile noting that the choice of bandwidth h_g in Theorem 1 is up to a constant. One can choose $h_g=C_0\max(\sqrt{p/n},(p/m)^{2^{g-2}})$ for a constant $C_0>0$ in practice and Theorem 1 still holds. We omit the constant C_0 for simplicity of the statement (i.e., setting $C_0=1$). We notice that the algorithm is not sensitive to the choice of C_0. Even with a suboptimal constant C_0, the algorithm still shows good performance with a few more rounds of iterations (i.e., using a larger q). Please see Section 5 for a simulation study that shows the insensitivity to the scaling constant.

According to our choice of h_q, we can see that as long as the number of iterations satisfies

$$q\geq 1+\log_2\left(\frac{\log n-\log p}{\log m-\log p}\right), \tag{21}$$

the bandwidth is $h_q=\sqrt{p/n}$. Then by (20), the Bahadur remainder term $\boldsymbol{r}_n$ becomes

$$\|\boldsymbol{r}_n\|_2=O_{\mathbb{P}}((p/n)^{3/4}(\log n)^{1/2}+\lambda^2). \tag{22}$$

When $\lambda\geq\sqrt{p/n}$, the convergence rate $\widetilde{\boldsymbol{\beta}}^{(q)}-\widetilde{\boldsymbol{\beta}}^*$ in (19) is dominated by λ. On the other hand, if $\lambda=O(\sqrt{p/n})$, then $\widetilde{\boldsymbol{\beta}}^{(q)}-\widetilde{\boldsymbol{\beta}}^*$ achieves the optimal rate $O_{\mathbb{P}}(\sqrt{p/n})$.

Remark 2 (The conditions on p and $\|\widetilde{\boldsymbol{\beta}}_0-\widetilde{\boldsymbol{\beta}}^*\|_2$) In this paper, we assume that the initializer is computed on the first machine with the convergence rate $\|\widetilde{\boldsymbol{\beta}}_0-\widetilde{\boldsymbol{\beta}}^*\|_2=O_{\mathbb{P}}(\sqrt{p/m})$. We require $p=O(m/(\log n)^2)$, which not only guarantees the consistency

of the estimator, but also provides us with a concise rate in the Bahadur remainder term (see Equation 20).

In fact, the assumption is not necessary if we assume that there is an initializer $\widetilde{\boldsymbol{\beta}}_0$ that satisfies $\|\widetilde{\boldsymbol{\beta}}_0-\widetilde{\boldsymbol{\beta}}^*\|_2=O_{\mathbb{P}}(n^{-\delta})$ for some constant $\delta>0$. Let $h_g=\max(\lambda,\sqrt{p/n},n^{-2^{g-1}\delta})$ and assume that conditions (C0)-(C4) hold. By the proof of Theorem 1, we have

$$\|\boldsymbol{r}_n\|_2=O_{\mathbb{P}}\left(\sqrt{\frac{ph_q\log n}{n}}+n^{-2^q\delta}+\lambda^2\right).$$

As long as the number of iterations q satisfies

$$q\geq\log_2\left(\frac{\log n-\log p}{\delta\log n}\right), \tag{23}$$

which is usually a small number in practice, we still obtain the optimal rate of the Bahadur remainder term in (22).

Remark 3 (Choice of the batch size m) The data batch size m balances the tradeoff between communication cost and computation cost. More specifically, when the batch size m is large, the convergence rate of the initial estimator is faster. Then, the required number of iterations becomes smaller (see Equation 21), which leads to a smaller communication cost. On the other hand, for a large batch size m, the computation cost of the initial estimator is large. Moreover, the computation time of $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ on each local machine also grows linearly in m. When the batch size m is small, the computation of the initial estimator becomes faster but it requires more iterations to achieve the same performance.

In practice, when the data is collected by multiple machines, the batch size will naturally be the storage size of each local machine. If we are allowed to specify m, we first need to make sure that m should be large enough so that the initial estimator is consistent. Moreover, since the communication is usually the bottleneck in distributed computing, it is desirable to choose m to be as large as possible to reach the capacity/memory limit of each local machine. This will provide a faster convergence rate of the initial estimator, and thus leads to a smaller number of iterations.

Remark 4 (Unbalanced batch size case) When the sample sizes on local machines are not balanced, we will choose the machine with the largest local sample size as the first machine to compute the initial estimator. This will provide us an initial estimator with faster convergence rate. As compared to the balanced case, our MDL estimator will require a smaller number of iterations to achieve the optimal statistical efficiency. It is worth noting that after the initial estimator is given, the MDL estimator in Algorithm 1 does not depend on the sample size on each local machine.

Define $\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)=\mathbb{E}[\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\mathrm{T}}I\{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*\geq 0\}]$. By applying the central limit theorem to (19), we have the following result on the asymptotic distribution of $\widetilde{\boldsymbol{\beta}}^{(q)}-\widetilde{\boldsymbol{\beta}}^*$.

Theorem 2 *Suppose that all the conditions of Theorem 1 hold with $h = h_g$ and $\lambda = o(n^{-1/2})$. Further, assume that $n = O(m^A)$ for some constant $A \geq 1$, $p = o(\min\{n^{1/3}/(\log n)^{2/3}, m^\nu\})$ for some $0 < \nu < 1$ and q satisfies (21). For any nonzero $\tilde{\boldsymbol{v}} \in \mathbb{R}^{p+1}$, we have as $n, p \to \infty$,*

$$\frac{n^{1/2}\tilde{\boldsymbol{v}}^{\mathrm{T}}(\widetilde{\boldsymbol{\beta}}^{(q)} - \widetilde{\boldsymbol{\beta}}^*)}{\sqrt{\tilde{\boldsymbol{v}}^{\mathrm{T}}\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\tilde{\boldsymbol{v}}}} \to \mathcal{N}(0,1).$$

Please see Appendix A.2 for the proofs of Theorem 1 and Theorem 2. We impose the conditions $n = O(m^A)$ and $p = o(m^\nu)$ for some constants $A \geq 1$ and $\nu \in (0,1)$ in order to ensure the right hand side of (21) is bounded by a constant, which implies that we only need to perform a constant number of iterations even when $n, m \to \infty$.

We introduce the vector $\tilde{\boldsymbol{v}}$ since we consider the diverging p regime and thus the dimension of the "sandwich matrix" $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}$ is growing in p. Therefore, it is notationally convenient to introduce an arbitrary vector $\tilde{\boldsymbol{v}}$ to make the limiting variance $\tilde{\boldsymbol{v}}^{\mathrm{T}}\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\tilde{\boldsymbol{v}}$ a positive real number. Also note that the conditions $p = o(n^{1/3}/(\log n)^{2/3})$ guarantees that the remainder term (22) satisfies $\|\boldsymbol{r}_n\|_2 = o_{\mathbb{P}}(n^{-1/2})$, which enables the application of the central limit theorem.

It is also important to note that the asymptotic variance $\tilde{\boldsymbol{v}}^{\mathrm{T}}\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\tilde{\boldsymbol{v}}$ in Theorem 2 matches the optimal asymptotic variance of $\widetilde{\boldsymbol{\beta}}_{\text{SVM all}}$ in (3), which is directly computed on all samples (see Theorem 2 in Koo et al., 2008). This result shows that the MDL estimator $\widetilde{\boldsymbol{\beta}}^{(q)}$ does not lose any statistical efficiency as compared to the linear SVM in a single machine setup. By contrast, the naïve divide-and-conquer approach requires the number of local machines N to satisfy the condition $N \leq (n/\log(p))^{1/3}$ (see Remark 1 in Lian and Fan, 2017). When this condition fails, the asymptotic normality of the estimator no longer holds. The MDL estimator removes the restriction on the number of machines but requires more communications overhead. In particular, the total communication cost for the MDL approach is $O(p^2Nq)$ (or $O(pNqT)$ for Algorithm 2), as compared to the one-shot communication $O(pN)$ in the naïve divide-and-conquer approach. It is worth noting that, under the assumptions $n = O(m^A)$ and $p = O(m^\nu)$ for some constants $A > 0$ and $0 < \nu < 1$ (see Proposition 2), a constant number of iterations (i.e., $qT = O(1)$) is enough to achieve the optimal rate.

We note that to construct the confidence interval of $\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*$ based on Theorem 2, we need consistent estimators of $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)$ and $\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)$. Since $\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)$ is defined as an expectation, it is natural to estimate it by its empirical version $\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\beta}}^{(q)})$. Moreover, by Proposition 4, we can estimate $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)$ by $\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}^{(q-1)}) = N^{-1}\sum_{k=1}^N \boldsymbol{V}_k$. Since $\boldsymbol{D}_{n,h}(\widetilde{\boldsymbol{\beta}}^{(q-1)})$ has already been obtained in the algorithm in the last iteration, we don't need extra computation. Given the nominal coverage probability $1-\rho_0$, the confidence interval for $\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*$ is given by

$$\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^{(q)} \pm n^{-1/2} z_{\rho_0/2}\sqrt{\tilde{\boldsymbol{v}}^{\mathrm{T}}\widehat{\boldsymbol{D}}(\widetilde{\boldsymbol{\beta}}^{(q-1)})^{-1}\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\beta}}^{(q)})\widehat{\boldsymbol{D}}(\widetilde{\boldsymbol{\beta}}^{(q-1)})^{-1}\tilde{\boldsymbol{v}}}, \tag{24}$$

where $z_{\rho_0/2}$ is the $1-\rho_0/2$ quantile of the standard normal distribution. For a fixed vector $\tilde{\boldsymbol{v}}$, denote $\widehat{\sigma}_{n,q} = \sqrt{\tilde{\boldsymbol{v}}^{\mathrm{T}}\widehat{\boldsymbol{D}}(\widetilde{\boldsymbol{\beta}}^{(q-1)})^{-1}\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\beta}}^{(q)})\widehat{\boldsymbol{D}}(\widetilde{\boldsymbol{\beta}}^{(q-1)})^{-1}\tilde{\boldsymbol{v}}}$. We have the following theorem for the asymptotic validity for the constructed confidence interval.

Theorem 3 (Plug-in estimation of the confidence interval) *Under the conditions of Theorem 2, for any nonzero $\tilde{\boldsymbol{v}} \in \mathbb{R}^{p+1}$, we have as $n, p \to \infty$,*

$$\mathbb{P}\left(\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^{(q)} - n^{-1/2}z_{\rho_0/2}\widehat{\sigma}_{n,q} \leq \tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^* \leq \tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^{(q)} + n^{-1/2}z_{\rho_0/2}\widehat{\sigma}_{n,q}\right) \to 1-\rho_0.$$

In the proof, we first show that $\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\beta}}^{(q)})$ is a consistent estimator of $\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)$. The detailed proof of Theorem 3 is relegated to Appendix A.2.

Remark 5 (Kernel SVM) It is worthwhile to note that the proposed distributed algorithm can also be used in solving nonlinear SVM by using feature mapping approximation techniques. In the general SVM formulation, the objective function is defined as follows:

$$\min_{\widetilde{\boldsymbol{\beta}}=(\beta_0,\boldsymbol{\beta})} \sum_{i=1}^n \left(1 - y_i(\phi(\boldsymbol{X}_i)^{\mathrm{T}}\boldsymbol{\beta} + \beta_0)\right)_+ + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2, \tag{25}$$

where the function ϕ is the feature mapping function which maps $\boldsymbol{X}_i$ to a high or even infinite dimensional space. The function $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ defined by $K(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x})^{\mathrm{T}}\phi(\boldsymbol{z})$ is called the kernel function associated with the feature mapping ϕ. With kernel mapping approximation, we construct a low dimensional feature mapping approximation $\psi : \mathbb{R}^p \to \mathbb{R}^d$ such that $\psi(\boldsymbol{x})^{\mathrm{T}}\psi(\boldsymbol{z}) \approx \phi(\boldsymbol{x})^{\mathrm{T}}\phi(\boldsymbol{z}) = K(\boldsymbol{x}, \boldsymbol{z})$. Then the original nonlinear SVM problem (25) can be approximated by

$$\min_{\widetilde{\boldsymbol{\beta}}\in\mathbb{R}^{d+1}} \sum_{i=1}^n \left(1 - y_i(\psi(\boldsymbol{X}_i)^{\mathrm{T}}\boldsymbol{\beta} + \beta_0)\right)_+ + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2. \tag{26}$$

Several feature mapping approximation methods have been developed for kernels with some nice properties (see, e.g., Rahimi and Recht, 2008; Lee and Wright, 2011; Vedaldi and Zisserman, 2012), and it is also shown that the approximation error $|\psi(\boldsymbol{x})^{\mathrm{T}}\psi(\boldsymbol{z}) - K(\boldsymbol{x}, \boldsymbol{z})|$ is small under some regularity conditions. We note that we should use a data-independent feature mapping approximation where ψ only depends on the kernel function K. This ensures that ψ can be directly computed without loading data, which enables efficient algorithm in a distributed setting. For instance, for the RBF kernel, which is defined as $K_{RBF}(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\sigma\|\boldsymbol{x} - \boldsymbol{z}\|_2^2\right)$, Rahimi and Recht (2008) proposed a data-independent approximation ψ as

$$\psi(\boldsymbol{X}) = \sqrt{\frac{2}{d}}[\cos(\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{X} + \omega_1), ..., \cos(\boldsymbol{v}_d^{\mathrm{T}}\boldsymbol{X} + \omega_d)]^{\mathrm{T}},$$

where $\boldsymbol{v}_1, ..., \boldsymbol{v}_d \in \mathbb{R}^p$ are i.i.d. samples from a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{0}, 2\sigma\mathbf{I})$ and $\omega_1, ..., \omega_d$ are i.i.d. samples from the uniform distribution on $[0, 2\pi]$.

Remark 6 (High-dimensional extension) We note that it is possible to extend the proposed MDL estimator to the high-dimensional case. In particular, with the smoothing technique used in our paper, we have the following smoothed loss function,

$$L_n(\widetilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^n K_h(1 - y_i l(\boldsymbol{X}_i; \widetilde{\boldsymbol{\beta}})) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2,$$

where $K_h(u) = uH(\frac{u}{h})$ is the smoothed hinge loss. Since the loss function $L_n(\widetilde{\boldsymbol{\beta}})$ is second-order differentiable, we can adopt the regularized approximate Newton method (see, e.g., Jordan et al., 2018; Wang et al., 2017). More specifically, the regularized approximate Newton method considers the following estimator,

$$\widetilde{\boldsymbol{\beta}}_1 = \underset{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}}{\arg\min}\{L_1(\widetilde{\boldsymbol{\beta}}) - \widetilde{\boldsymbol{\beta}}^{\mathrm{T}}(\nabla L_1(\widetilde{\boldsymbol{\beta}}_0) - \nabla L_n(\widetilde{\boldsymbol{\beta}}_0)) + \frac{\tilde{\lambda}}{2}\|\boldsymbol{\beta}\|_1\}, \tag{27}$$

where $\widetilde{\boldsymbol{\beta}}_0$ is an initial estimator and $L_1(\widetilde{\boldsymbol{\beta}})$ is the smoothed loss function computed using the data on the first machine. It can be extended to an iterative algorithm by repeatedly updating the estimator using (27). However, there are two technical challenges. First, the smoothed loss function becomes non-convex. Second, it is unclear how to choose the bandwidth h such that it shrinks properly with the number of iterations. We leave these two technical questions and the extension to the high-dimensional case for future investigation.

5. Simulation Studies

In this section, we provide a simulation experiment to illustrate the performance of the proposed distributed SVM algorithm. The data is generated from the following model

$$P(Y_i = 1) = p_+, \quad P(Y_i = -1) = p_- = 1 - p_+,$$

$$\boldsymbol{X}_i = Y_i\mathbf{1} + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \quad i = 1, 2, ..., n,$$

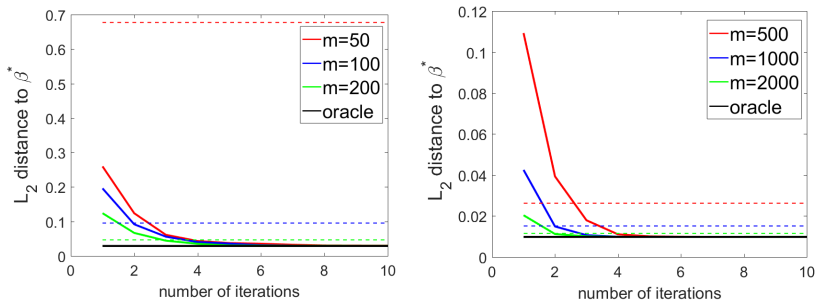
where $\mathbf{1}$ is the all-one vector $(1, 1, ..., 1)^{\mathrm{T}} \in \mathbb{R}^p$ and the triplets $(Y_i, \boldsymbol{X}_i, \boldsymbol{\epsilon}_i)$ are drawn independently. We set $\sigma = \sqrt{p}$ throughout the simulation study. In order to directly compare the proposed estimator to other estimators, we follow the simulation study setting in Koo et al. (2008) and consider the optimization problem without penalty term, i.e., $\lambda = 0$. We set $p_+ = p_- = \frac{1}{2}$, i.e., the data is generated from the two classes with equal probability. Note that we can explicitly solve the true coefficient $\widetilde{\boldsymbol{\beta}}^*$ by the following claim whose proof is relegated to Appendix A.3.

Claim 1 *The true coefficient vector is $\widetilde{\boldsymbol{\beta}}^* = \frac{1}{a}(0, 1, ..., 1)^{\mathrm{T}} \in \mathbb{R}^{p+1}$, where a is the solution to $\int_{-\infty}^{a} \phi_1(x)xdx = 0$ and $\phi_1(x)$ is the p.d.f. of the distribution $\mathcal{N}(p, \sigma^2 p)$.*

We use the integral of a kernel function as the smoothing function:

$$H(v) = \begin{cases} 0 & \text{if } v \le -1, \\ \frac{1}{2} + \frac{15}{16}\left(v - \frac{2}{3}v^3 + \frac{1}{5}v^5\right) & \text{if } |v| < 1, \\ 1 & \text{if } v \ge 1. \end{cases}$$

The initial estimator $\widetilde{\boldsymbol{\beta}}_0$ is computed by directly solving the convex optimization problem (2) with only the samples in the first machine, and the iterative distributed algorithm is then applied to data in all the machines. We consider the naïve divide-and-conquer (Naïve-DC) approach which simply computes the solution of the optimization problem on every single machine and combines all the solutions by taking the average. The oracle estimator is defined by (3) which directly solves the optimization with data from all machines. The



(a) $p = 4$, $n = 10^4$ case (b) $p = 20$, $n = 10^6$ case

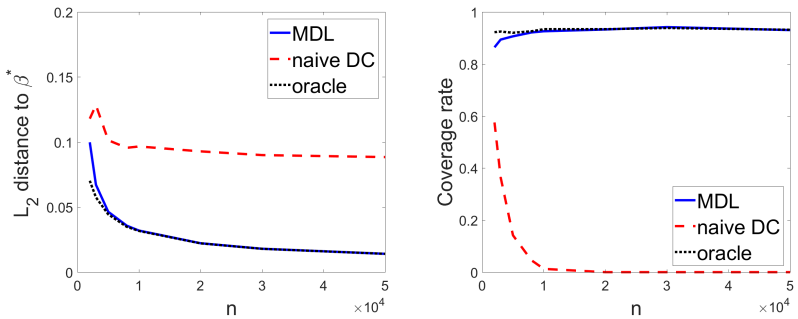
Figure 2: L_2 error of three estimators with different number of iterations q. The dashed horizontal lines show the performance of the Naïve-DC for different values of m and the solid lines show the performance of the MDL estimators (for different m) and the oracle estimator.

confidence intervals are constructed for $\tilde{\boldsymbol{v}}_0^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*$ with all these three estimators, where $\tilde{\boldsymbol{v}}_0 = (p+1)^{-1/2}\mathbf{1}_{p+1}$ and the nominal coverage probability $1 - \rho_0$ is set to 95%. We use (24) to construct the confidence interval and we also use the same interval length for all the three estimators. We compare both the L_2 distance between the estimator and the true coefficient $\widetilde{\boldsymbol{\beta}}^*$ and the empirical coverage rate for all the three estimators.

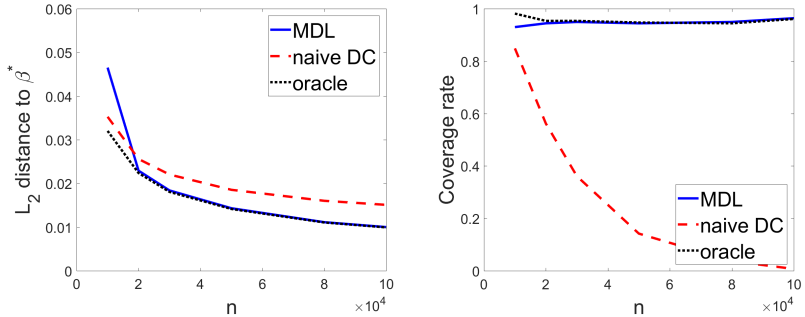
5.1. L_2 Error and Empirical Coverage Rate

We first investigate how the L_2 error of our proposed estimator improves with the number of aggregations. We consider two settings: the number of samples $n = 10^4$, dimension $p = 4$, batch size $m \in \{50, 100, 200\}$ and $n = 10^6$, $p = 20$, $m \in \{500, 1000, 2000\}$. We set the max number of iterations as 10 and plot the L_2 error at each iteration. We also plot the L_2 error of Naïve-DC estimator (the dashed line) and the oracle estimator (the black line) as horizontal lines for comparison. All the results reported are the average of 1000 independent runs. From Figure 2 we can see that the error of proposed MDL estimator decreases quickly with the number of iterations. After 5 rounds of aggregations, the MDL estimator performs better than the Naïve-DC approach and it almost achieves the same L_2 error as the oracle estimator.

Next, we experiment on how the performance of the estimators changes with the total number of data points n while the number of data that each machine can store is fixed. We consider two settings where the machine capacity $m = 100$ and 1000, the number of iterations $q = 10$ and dimension $p = 4$ and 20, and we plot the L_2 error and empirical coverage rate for all the three estimator against the sample size n.



(a) $p = 4$, $m = 100$ case

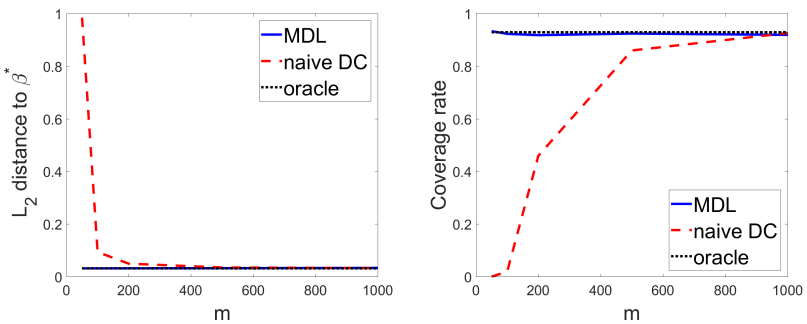


(b) $p = 20$, $m = 1000$ case

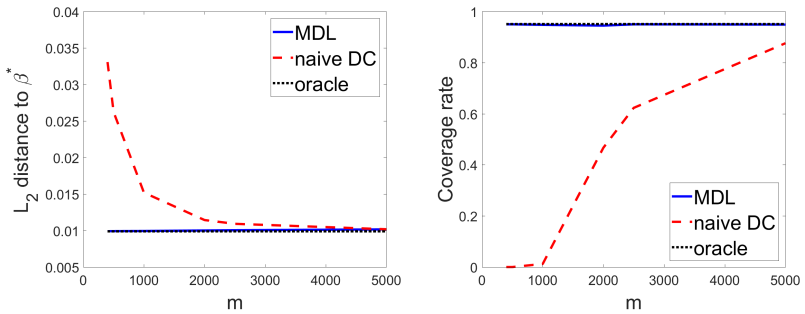
Figure 3: L_2 error and coverage rate of the MDL estimator with different total sample size n (the number of iterations $q = 10$).

From Figure 3 we can observe that the L_2 error of the oracle estimator decreases as n increases, but the Naïve-DC estimator clearly fails to converge to the true estimator which is essentially due to the fact that the bias of the Naïve-DC estimator does not decrease with n. However, the proposed MDL estimator converges to the true coefficient with almost the identical rate as the oracle estimator. We also notice that the coverage rate of the MDL estimator is quite close to that of the oracle estimator which is close to the nominal coverage probability 95%, while the coverage rate of the Naïve-DC estimator quickly decreases and drops to zero when n increases.

The next experiment shows how the L_2 error and the coverage rate change with different machine capacity m with fixed sample size n. Two parameter settings are considered where the sample size $n = 10^5, 10^6$, dimension $p = 4, 20$, and the number of iterations is $q = 10$.



(a) $p = 4$, $n = 10^5$ case



(b) $p = 20$, $n = 10^6$ case

Figure 4: L_2 error and coverage rate of the MDL estimator with different batch size m (the number of iterations $q = 10$).

The results are shown in Figure 4. From Figure 4 we can see that when the machine capacity gets small, the L_2 error of the Naïve-DC estimator increases drastically and it fails when $m \leq 100$ in the $n = 10^5$ case and $m \leq 400$ in the $n = 10^6$ case. On the contrary, the MDL estimator is quite robust even when the machine capacity is small. Moreover, the empirical coverage rate for the Naïve-DC estimator is small and only approaches 95% when m is sufficiently large, while the coverage rate for the proposed MDL estimator is close to the oracle estimator which is close to the nominal coverage probability 95%.

5.2. Bias and Variance Analysis

In Table 1 and Table 2, we report the bias and variance analysis for the MDL, Naïve-DC and oracle estimator. In Table 1, we fix two settings of sample size n and dimension p

(n,p)	m	MDL		Naïve-DC		Oracle	
		bias^2 ($\times 10^{-4}$)	var ($\times 10^{-4}$)	bias^2 ($\times 10^{-4}$)	var ($\times 10^{-4}$)	bias^2 ($\times 10^{-4}$)	var ($\times 10^{-4}$)
$(10^4, 4)$	100	0.004	3.906	59.329	4.457	0.000	2.275
	200	0.002	2.516	10.678	2.950	0.000	2.275
	500	0.005	2.459	1.393	2.581	0.000	2.275
	1000	0.006	2.608	0.304	2.420	0.000	2.275
$(10^5, 20)$	400	0.000	0.076	9.759	0.085	0.000	0.058
	500	0.000	0.059	5.351	0.080	0.000	0.058
	1000	0.000	0.059	1.140	0.069	0.000	0.058
	2000	0.000	0.060	0.261	0.063	0.000	0.058
	2500	0.000	0.060	0.168	0.062	0.000	0.058
	5000	0.000	0.061	0.041	0.058	0.000	0.058

Table 1: Bias and variance analysis of MDL, Naïve-DC and oracle estimator with different batch size m when number of aggregations $q = 6$.

and investigate how the bias and variance of $\tilde{\boldsymbol{v}}_0^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}$ change with the batch size m for each estimator. As we can see from Table 1, the variance of both the MDL and Naïve-DC estimators is close to the oracle estimator. However, when the batch size m gets relatively small, the bias term of the Naïve-DC estimator goes large, and the squared bias quickly exceeds the variance term, which aligns with the discussion in Section 2. On the other hand, the bias of the MDL estimator stays small and is quite close to the bias of the oracle estimator.

Similarly, in Table 2 we fix two settings of m and p and vary the sample size n. We observe that the variance of all the three estimators reduces as the sample size n grows large. However, in both settings the squared bias of the Naïve-DC estimator does not improve as n increases which also illustrates why the central limit theorem fails for the Naïve-DC estimator. On the other hand, the squared bias of the MDL estimator is close to that of the oracle estimator as n gets large.

5.3. The Performance under Large n and p

In this section, we investigate the performance of the MDL estimator for varying dimension p. In Table 3, we choose a large sample size $n = 10^6$, and vary the dimension p and the batch size m. We report the L_2 error of the MDL estimator with different number of iterations. From the result we can see that our proposed estimator maintains good performance under large scale settings. The L_2 error of the MDL estimator becomes small in all settings and stays stable when the number of iterations q is slightly larger (e.g., $q \geq 4$).

(m,p)	n ($\times 10^3$)	MDL		Naïve-DC		Oracle	
		bias^2 ($\times 10^{-4}$)	var ($\times 10^{-4}$)	bias^2 ($\times 10^{-4}$)	var ($\times 10^{-4}$)	bias^2 ($\times 10^{-4}$)	var ($\times 10^{-4}$)
(100,4)	2	1.782	55.412	58.377	24.912	0.017	13.382
	3	0.221	28.275	60.877	17.707	0.046	8.557
	5	0.120	12.694	60.696	10.267	0.020	5.016
	8	0.005	4.056	58.806	6.213	0.028	3.244
	10	0.008	3.054	62.461	4.919	0.025	2.568
	20	0.006	1.400	61.609	2.589	0.005	1.325
	30	0.000	1.611	60.540	1.754	0.002	0.773
(1000,20)	20	0.002	0.355	1.075	0.334	0.001	0.294
	30	0.001	0.207	1.148	0.236	0.001	0.194
	50	0.001	0.122	1.147	0.144	0.001	0.121
	80	0.000	0.072	1.118	0.082	0.000	0.072
	100	0.000	0.054	1.090	0.060	0.000	0.053

Table 2: Bias and variance analysis of MDL, Naïve-DC and oracle estimator with different sample size n when number of aggregations $q = 6$.

5.4. Sensitivity Analysis of the Bandwidth Constant C_0

Finally, we report the simulation study to show that the algorithm is not sensitive to the choice of C_0 in bandwidth h_g where $h_g = C_0 \max(\sqrt{p/n}, (p/m)^{2^{g-2}})$. We set $n = 10^4$, $p = 4$ with $m \in \{50, 100\}$ and $n = 10^5$, $p = 20$ with $m \in \{500, 1000\}$. The constant C_0 is selected from $\{0.5, 1, 2, 5, 10\}$. We plot the L_2 error of the MDL estimator at each iteration step with different choices of C_0. We also plot the L_2 error of the Naïve-DC estimator (the dashed line) and the oracle estimator (the black line) as horizontal lines for comparison. Figure 5 shows that the proposed estimator exhibits good performance for all choices of C_0 after a few rounds of iterations and finally achieves the L_2 errors which are close to the L_2 error of the oracle estimator.

6. Conclusions and Future Works

In this paper, we propose a multi-round distributed linear-type (MDL) estimator for conducting inference for linear support vector machine with a large sample size n and a growing dimension p. The proposed method only needs to calculate the SVM estimator on a small batch of data as an initial estimator, and all the remaining works are simple matrix operations. Our approach is not only computationally efficient but also achieves the same

(n,p)	m	MDL ($\times 10^{-2}$)				
		$q=2$	$q=4$	$q=6$	$q=8$	$q=10$
$(10^6, 50)$	1000	3.223	0.313	0.294	0.293	0.293
	2000	0.601	0.290	0.291	0.291	0.291
	2500	0.436	0.292	0.291	0.291	0.291
$(10^6, 100)$	2000	2.330	0.372	0.337	0.338	0.338
	2500	1.473	0.339	0.338	0.338	0.338
	5000	0.436	0.340	0.338	0.338	0.338
$(10^6, 200)$	2500	3.182	0.540	0.342	0.324	0.324
	4000	1.139	0.351	0.320	0.324	0.324
	5000	0.865	0.336	0.324	0.324	0.324
$(10^6, 500)$	6250	3.502	0.823	0.315	0.324	0.320
	8000	1.547	0.309	0.332	0.321	0.321
	10000	1.342	0.319	0.312	0.320	0.321

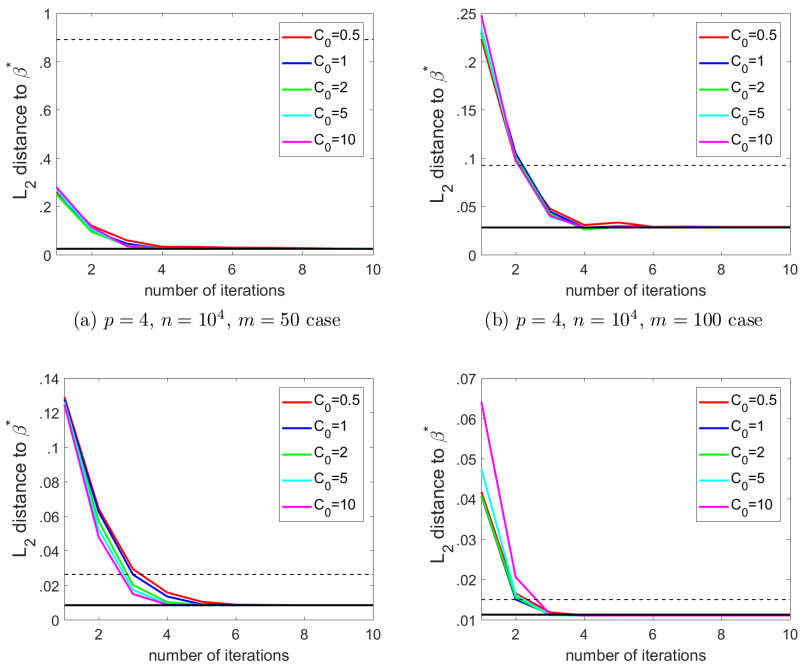
Table 3: Comparison of the L_2 error under different dimensionality p and batch size m. The sample size is fixed to $n = 10^6$, and the number of iterations $q = 10$.

statistical efficiency as the classical linear SVM estimator using all the data. In our theoretical results in Theorem 1, the term $(\frac{p}{m})^{2^{q-1}}$ corresponds to the convergence rate of the bias. An interesting theoretical open problem is that whether the rate of the bias is optimal. Note that according to Lemma 1, the expectation of the bias is bounded by a_n^2 (with the choice of bandwidth $h = a_n$). We conjecture that the rate of the bias a_n^2 is optimal, but we leave this conjecture for future investigation.

This work only serves as the first step towards distributed inference for SVM, which is an important area that bridges statistics and machine learning. In the future, we would like to further establish unified computational approaches and theoretical tools for statistical inference for other types of SVM problems, such as L_q-penalized SVM (see ,e.g., Liu et al. (2007)), high-dimensional SVM (see, Peng et al. (2016); Zhang et al. (2016b)), and more general kernel-based SVM.

Acknowledgments

Xiaozhou Wang and Weidong Liu are supported by NSFC, Grant No. 11825104, 11431006 and 11690013, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, Youth Talent Support Program, 973 Program



(a) $p = 4$, $n = 10^4$, $m = 50$ case (b) $p = 4$, $n = 10^4$, $m = 100$ case

(c) $p = 20$, $n = 10^5$, $m = 500$ case (d) $p = 20$, $n = 10^5$, $m = 1000$ case

Figure 5: L_2 error of the MDL estimator with different C_0. Dashed lines show the performance of the Naïve-DC approach and the black solid line shows the performance of the oracle estimator. Other colored lines show the performance of the MDL estimator with different choices of constants in the bandwidth.

(2015CB856004), and a grant from Australian Research Council. Zhuoyi Yang and Xi Chen are supported by NSF Award (IIS-1845444), Alibaba Innovation Research Award, and Bloomberg Data Science Research Grant.

Appendix A. Proofs for Results

In this appendix, we provide the proofs of the results.

A.1. Technical Lemmas

Before proving the theorems and propositions, we first introduce three technical lemmas, which will be used in our proof.

Lemma 1 *Suppose that conditions (C0)-(C4) hold. For any $\tilde{\boldsymbol{v}} \in \mathbb{R}^{p+1}$ with $\|\tilde{\boldsymbol{v}}\|_2 = 1$, we have*

$$\mathbb{E}\left\{Y\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{X}}\left(H\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)+\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*}{h}H'\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right)\right\} = O(h^2+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2^2),$$

uniformly in $\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2 \le a_n$ with any $a_n \to 0$.

Proof of Lemma 1. Without loss of generality, assume that $\beta_1^* \ge c$. Then $\alpha_1 \ge c/2$. For any $\boldsymbol{v} \in \mathbb{R}^p$,

$$\begin{aligned}&\mathbb{E}\left\{Y(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{X})H\left(\frac{1-Y(\alpha_0+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha})}{h}\right)\right\}\\ =&\pi_+\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})H\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}-\pi_-\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})H\left(\frac{1+\alpha_0+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)g(\boldsymbol{x})d\boldsymbol{x}.\end{aligned}$$

We have

$$\begin{aligned}&\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})H\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}\\ =&\int_{\mathbb{R}^{p-1}}\int_{\mathbb{R}}(v_0+v_1x_1+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})H\left(\frac{1-\alpha_0-x_1\alpha_1-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}}{h}\right)f(x_1,\boldsymbol{x}_{-1})dx_1d\boldsymbol{x}_{-1}\\ =&-\frac{h}{\alpha_1}\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{\mathbb{R}}\left(v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right)\\ &\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H(y)dyd\boldsymbol{x}_{-1}.\end{aligned}$$

Define $G(t|\boldsymbol{x}_{-1}) = \int_{-\infty}^t xf(x|\boldsymbol{x}_{-1})dx$. Since $\int_{\mathbb{R}}|x|f(x|\boldsymbol{x}_{-1})dx < \infty$, we have $G(-\infty|\boldsymbol{x}_{-1}) = 0$. Then,

$$\begin{aligned}&-\int_{\mathbb{R}}v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H(y)dy\\ =&\frac{\alpha_1}{h}v_1\int_{\mathbb{R}}H(y)dG\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)\\ =&-\frac{\alpha_1}{h}v_1\int_{-1}^1G\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dy\\ =&-\frac{\alpha_1}{h}v_1G\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)\\ &-\frac{\alpha_1}{h}v_1\int_{-1}^1\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)H'(y)dy\\ &+O(1)\frac{\alpha_1}{h}v_1\int_{-1}^1\Delta^2(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy,\end{aligned}$$

where

$$\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y) = \frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}-\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*},$$

and the inequality $|xf(x|\boldsymbol{x}_{-1})-yf(y|\boldsymbol{x}_{-1})| \le C|x-y|$ followed from Condition (C2). Also,

$$\begin{aligned}&-\int_{\mathbb{R}}(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H(y)dy\\ =&-\frac{\alpha_1}{h}\int_{-1}^1(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})F\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dy\\ =&-\frac{\alpha_1}{h}(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})F\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)\\ &-\frac{\alpha_1}{h}\int_{-1}^1(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)H'(y)dy\\ &+O(1)\frac{\alpha_1}{h}\int_{-1}^1|v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}|\Delta^2(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy.\end{aligned}$$

Next we consider

$$\begin{aligned}&\mathbb{E}\left\{Y(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{X})\frac{1-Y(\beta_0^*+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}^*)}{h}H'\left(\frac{1-Y(\alpha_0+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha})}{h}\right)\right\}\\ =&\pi_+\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})\frac{1-\beta_0^*-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\beta}^*}{h}H'\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}\\ &-\pi_-\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})\frac{1+\beta_0^*+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\beta}^*}{h}H'\left(\frac{1+\alpha_0+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)g(\boldsymbol{x})d\boldsymbol{x}.\end{aligned}$$

We have

$$\begin{aligned}
&\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})\frac{1-\beta_0^*-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\beta}^*}{h}H'\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}\\
=&-\frac{1}{\alpha_1}\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{\mathbb{R}}\left(v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right)\\
&\times\left(1-\beta_0^*-\beta_1^*\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*\right)\\
&\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dyd\boldsymbol{x}_{-1}.
\end{aligned}$$

Note that

$$\begin{aligned}
&-\frac{1}{\alpha_1}\int_{\mathbb{R}}\left(v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right)\\
&\qquad\times\left(1-\beta_0^*-\beta_1^*\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*\right)\\
&\qquad\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
=&\frac{\beta_1^*}{\alpha_1}\int_{\mathbb{R}}\left(v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right)\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)\\
&\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
=&\int_{\mathbb{R}}\left(v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right)\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)\\
&\times f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
&+O(1)\int_{\mathbb{R}}\left|v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right|\Delta^2(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy\\
&+O(1)\frac{|\beta_1^*-\alpha_1|}{\alpha_1}\int_{\mathbb{R}}\left|v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right|\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy\\
=&\int_{\mathbb{R}}\left(v_0+v_1\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right)\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
&+O(1)\int_{\mathbb{R}}\left|v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right|\Delta^2(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy\\
&+O(1)\frac{|\beta_1^*-\alpha_1|}{\alpha_1}\int_{\mathbb{R}}\left|v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right|\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy\\
&+O(1)\int_{\mathbb{R}}\Delta^2(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy.
\end{aligned}$$

Note that

$$|\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|\le C\left(h+|\boldsymbol{x}_{-1}^{\mathrm{T}}(\boldsymbol{\alpha}_{-1}-\boldsymbol{\beta}_{-1}^*)|+|1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*||\alpha_1-\beta_1^*|+|\alpha_0-\beta_0^*|\right).$$

So we have

$$\begin{aligned}
&\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})H\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}\\
&\qquad+\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})\frac{1-\beta_0^*-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\beta}^*}{h}H'\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}\\
=&-v_1\int_{\mathbb{R}^{p-1}}G\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)f_{-1}(\boldsymbol{x}_{-1})d\boldsymbol{x}_{-1}\\
&-\int_{\mathbb{R}^{p-1}}(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})F\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)f_{-1}(\boldsymbol{x}_{-1})d\boldsymbol{x}_{-1}+O(h^2+\|\boldsymbol{\alpha}-\boldsymbol{\beta}^*\|_2^2).
\end{aligned}$$

Note that

$$v_1\int_{\mathbb{R}^{p-1}}G\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)f_{-1}(\boldsymbol{x}_{-1})d\boldsymbol{x}_{-1}=\mathbb{E}[v_1YX_1I\{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*\ge 0\}|Y=1],$$

and

$$\begin{aligned}
&\int_{\mathbb{R}^{p-1}}(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})F\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)f_{-1}(\boldsymbol{x}_{-1})d\boldsymbol{x}_{-1}\\
=&\mathbb{E}[Y(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{X}_{-1})I\{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*\ge 0\}|Y=1].
\end{aligned}$$

So

$$\begin{aligned}
&\mathbb{E}\left\{Y\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{X}}\left(H\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)+\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*}{h}H'\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right)\right\}\\
=&-\mathbb{E}[\tilde{\boldsymbol{v}}^{\mathrm{T}}Y\widetilde{\boldsymbol{X}}I\{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*\ge 0\}]+O(h^2+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2^2)\\
=&O(h^2+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2^2).
\end{aligned}$$

■

Lemma 2 *Suppose that conditions (C0)-(C4) hold. For any $\tilde{\boldsymbol{v}}\in\mathbb{R}^{p+1}$ with $\|\tilde{\boldsymbol{v}}\|_2=1$, we have*

$$\mathbb{E}\left\{\frac{1}{h}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{X}})^2H'\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right\}=\tilde{\boldsymbol{v}}^{\mathrm{T}}\mathbb{E}\left[\delta(1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*)\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\mathrm{T}}\right]\tilde{\boldsymbol{v}}+O(h+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2),$$

uniformly in $\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2\le a_n$ with any $a_n\to 0$.

Proof of Lemma 2. Without loss of generality, assume that $\beta_1^*\ge c$. Then $\alpha_1\ge c/2$. For any $\boldsymbol{v}\in\mathbb{R}^p$,

$$\begin{aligned}
&\mathbb{E}\left\{\frac{1}{h}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{X}})^2H'\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right\}\\
=&\pi_+\int_{\mathbb{R}^p}\frac{1}{h}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}})^2H'\left(\frac{1-\tilde{\boldsymbol{x}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}+\pi_-\int_{\mathbb{R}^p}\frac{1}{h}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}})^2H'\left(\frac{1+\tilde{\boldsymbol{x}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)g(\boldsymbol{x})d\boldsymbol{x}.
\end{aligned}$$

We have

$$
\begin{aligned}
&\frac{1}{h}\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})^2H'\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}\\
=&\frac{1}{h}\int_{\mathbb{R}^{p-1}}\int_{\mathbb{R}}(v_0+v_1x_1+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2H'\left(\frac{1-\alpha_0-x_1\alpha_1-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}}{h}\right)f(x_1,\boldsymbol{x}_{-1})dx_1d\boldsymbol{x}_{-1}\\
=&-\frac{1}{\alpha_1}\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{-1}^{1}(v_0+v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2\\
&\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dyd\boldsymbol{x}_{-1}.
\end{aligned}
$$

Note that

$$
\begin{aligned}
&\int_{-1}^{1}(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
=&\int_{-1}^{1}(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
&+O(1)\int_{-1}^{1}(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy.
\end{aligned}
$$

According to Condition (C2), we have

$$
\begin{aligned}
&\int_{-1}^{1}2(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})v_1\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
=&\int_{-1}^{1}2(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})v_1\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
&+O(1)\int_{-1}^{1}2(v_0+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})v_1\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy,
\end{aligned}
$$

and

$$
\begin{aligned}
&\int_{-1}^{1}v_1^2\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}\right)^2f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
=&\int_{-1}^{1}v_1^2\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}\right)^2f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)H'(y)dy\\
&+O(1)\int_{-1}^{1}v_1^2\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dy.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\frac{1}{h}\int_{\mathbb{R}^p}(v_0+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x})^2H'\left(\frac{1-\alpha_0-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\alpha}}{h}\right)f(\boldsymbol{x})d\boldsymbol{x}\\
=&-\frac{1}{\beta_1^*}\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{-1}^{1}(v_0+v_1\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2\\
&\times f\left(\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}|\boldsymbol{x}_{-1}\right)H'(y)dyd\boldsymbol{x}_{-1}\\
&+O(1)\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{-1}^{1}(v_0+v_1+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)|H'(y)|dyd\boldsymbol{x}_{-1}\\
&+O(1)\frac{|\alpha_1-\beta_1^*|}{\alpha_1\beta_1^*}\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{-1}^{1}(v_0+v_1\frac{1-\beta_0^*-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*}{\beta_1^*}+\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1})^2|H'(y)|dyd\boldsymbol{x}_{-1}\\
=&\tilde{\boldsymbol{v}}^{\mathrm{T}}\mathbb{E}\left[\delta(1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*)\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\mathrm{T}}|Y=1\right]\tilde{\boldsymbol{v}}+O(h+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2).
\end{aligned}
$$

Then we get

$$
\mathbb{E}\left\{\frac{1}{h}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{X}})^2H'\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right\}=\tilde{\boldsymbol{v}}^{\mathrm{T}}\mathbb{E}\left[\delta(1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*)\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\mathrm{T}}\right]\tilde{\boldsymbol{v}}+O(h+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2).
$$

■

Define $K(\widetilde{\boldsymbol{X}},\widetilde{\boldsymbol{\theta}})=|1+\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\theta}}|$ and

$$
H_h(\widetilde{\boldsymbol{\alpha}})=H\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)-I\{\epsilon\geq 0\}+\frac{\epsilon}{h}H'\left(\frac{1-Y\widetilde{\boldsymbol{X}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right).
$$

Lemma 3 *Suppose that conditions (C0)-(C4) hold. For some $t>0$ and any $\tilde{\boldsymbol{v}},\widetilde{\boldsymbol{\theta}}\in\mathbb{R}^{p+1}$ with $\|\tilde{\boldsymbol{v}}\|_2=1$ and $\|\widetilde{\boldsymbol{\theta}}\|_2=1$, we have*

$$
\mathbb{E}\{\tilde{\boldsymbol{v}}^{\mathrm{T}}Y\widetilde{\boldsymbol{X}}H_h(\widetilde{\boldsymbol{\alpha}})\}^2\exp(t|\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{X}}|K(\widetilde{\boldsymbol{X}},\widetilde{\boldsymbol{\theta}}))=O(h+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2^2/h),
$$

uniformly in $\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2\leq a_n$ with any $a_n\to 0$.

Proof of Lemma 3. We have

$$
\begin{aligned}
&\int_{\mathbb{R}^p}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}})^2\exp(t|\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}|K(\tilde{\boldsymbol{x}},\widetilde{\boldsymbol{\theta}}))\left[H\left(\frac{1-\tilde{\boldsymbol{x}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)-I\{\epsilon\geq 0\}\right]^2f(\boldsymbol{x})d\boldsymbol{x}\\
=&-\frac{h}{\alpha_1}\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{\mathbb{R}}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*)^2\exp(t|\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*|K(\tilde{\boldsymbol{x}}^*,\widetilde{\boldsymbol{\theta}}))\\
&\times\left[H(y)-I\left\{1-\beta_0^*-\beta_1^*\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*\geq 0\right\}\right]^2\\
&\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)dyd\boldsymbol{x}_{-1},
\end{aligned}
$$

where $I\left\{1-\beta_0^*-\beta_1^*\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\beta}_{-1}^*\geq 0\right\}=I\{y\geq\frac{\alpha_1}{h}\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},0)\}$ and $\tilde{\boldsymbol{x}}^*$ denotes $\tilde{\boldsymbol{x}}$ with x_1 being replaced by $\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}$. According to Condition (C2), (C3) and (C4), note that

$$\begin{aligned}
&\int_{\mathbb{R}}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*)^2\exp(t|\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*|K(\tilde{\boldsymbol{x}}^*,\widetilde{\boldsymbol{\theta}}))\left[H(y)-I\{y\geq\frac{\alpha_1}{h}\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},0)\}\right]^2\\
&\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)dy\\
\leq& C\left(\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}\right)^2\exp(t'|\boldsymbol{v}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}|^2+t'(1+|\boldsymbol{\theta}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}|+|\boldsymbol{\alpha}_{-1}^{\mathrm{T}}\boldsymbol{x}_{-1}|+|\boldsymbol{\beta}_{-1}^{*\mathrm{T}}\boldsymbol{x}_{-1}|)^2)\\
&\times\left(1+\left|\frac{\alpha_1}{h}\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},0)\right|\right),
\end{aligned}$$

for some $t'>0$. Therefore, by (C3)

$$\int_{\mathbb{R}^p}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}})^2\exp(t|\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}|K(\tilde{\boldsymbol{x}},\widetilde{\boldsymbol{\theta}}))\left[H\left(\frac{1-\tilde{\boldsymbol{x}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)-I\{\epsilon\geq 0\}\right]^2 f(\boldsymbol{x})d\boldsymbol{x}\leq O(h+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2).$$

On the other hand, we can easily prove that

$$\begin{aligned}
&\int_{\mathbb{R}^p}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}})^2\exp(t|\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}|K(\tilde{\boldsymbol{x}},\widetilde{\boldsymbol{\theta}}))\left[\frac{\epsilon}{h}H'\left(\frac{1-\tilde{\boldsymbol{x}}^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right]^2 f(\boldsymbol{x})d\boldsymbol{x}\\
=&-\frac{h\beta_1^{*2}}{\alpha_1}\int_{\mathbb{R}^{p-1}}f_{-1}(\boldsymbol{x}_{-1})\int_{-1}^{1}(\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*)^2\exp(t|\tilde{\boldsymbol{v}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*|K(\tilde{\boldsymbol{x}}^*,\widetilde{\boldsymbol{\theta}}))\left[\frac{\Delta(\boldsymbol{\alpha},\boldsymbol{\beta}^*,\boldsymbol{x}_{-1},y)}{h}H'(y)\right]^2\\
&\times f\left(\frac{1-\alpha_0-\boldsymbol{x}_{-1}^{\mathrm{T}}\boldsymbol{\alpha}_{-1}-hy}{\alpha_1}|\boldsymbol{x}_{-1}\right)dyd\boldsymbol{x}_{-1}\\
\leq& O(h+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2^2/h).
\end{aligned}$$

Now we complete the proof of the lemma. ■

A.2. Proofs of the Main Results

After introducing and proving the above three lemmas, we begin to prove Proposition 3 and 4, Theorem 1 and 2.

Proof of Proposition 3. Recall that $\epsilon_i=1-y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*$. Define $\Delta(\widetilde{\boldsymbol{\alpha}})=\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*$, and

$$\begin{aligned}
\boldsymbol{C}_{n,h}(\widetilde{\boldsymbol{\alpha}})&=\boldsymbol{A}_{n,h}(\widetilde{\boldsymbol{\alpha}})-\frac{1}{n}\sum_{i=1}^n y_i\widetilde{\boldsymbol{X}}_iI\{\epsilon_i\geq 0\}\\
&=\frac{1}{n}\sum_{i=1}^n y_i\widetilde{\boldsymbol{X}}_i\left[H\left(\frac{1-y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)-I\{\epsilon_i\geq 0\}+\frac{1-y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}}^*}{h}H'\left(\frac{1-y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right].
\end{aligned}$$

Let $\boldsymbol{C}_{n,h}=\boldsymbol{C}_{n,h}(\widetilde{\boldsymbol{\beta}}^*)$. Note that $\|\boldsymbol{C}_{n,h}\|_2=\sup_{\tilde{\boldsymbol{v}}\in\mathbb{R}^{(p+1)},\|\tilde{\boldsymbol{v}}\|_2=1}|\tilde{\boldsymbol{v}}^{\mathrm{T}}\boldsymbol{C}_{n,h}|$.

Let $S_{1/2}^p$ be a 1/2 net of the unit sphere S^p in the Euclidean distance in $\mathbb{R}^{p+1}$. According to the proof of Lemma 3 in Cai et al. (2010), we have d_{p+1}:=Card$(S_{1/2}^p)\leq 5^{p+1}$. Let

$\tilde{\boldsymbol{v}}_1,...,\tilde{\boldsymbol{v}}_{d_{p+1}}$ be the centers of the d_{p+1} elements in the net. Therefore for any $\tilde{\boldsymbol{v}}$ in S^p, we have $\|\tilde{\boldsymbol{v}}-\tilde{\boldsymbol{v}}_j\|_2\leq 1/2$ for some j. Therefore, $\|\boldsymbol{C}_{n,h}\|_2\leq 2\sup_{j\leq d_{p+1}}|\tilde{\boldsymbol{v}}_j^{\mathrm{T}}\boldsymbol{C}_{n,h}|$. It is easy to see that for any $M>0$, there exists a set of points in $\mathbb{R}^{p+1}$, $\{\widetilde{\boldsymbol{\alpha}}_k,1\leq k\leq s_{p+1}\}$ with $s_{p+1}\leq n^{M(p+1)}$, such that for any $\widetilde{\boldsymbol{\alpha}}$ in the ball $\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2\leq a_n$, we have $\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\alpha}}_k\|_2\leq 2\sqrt{p+1}a_n/n^M$ for some $1\leq k\leq s_{p+1}$ and $\|\widetilde{\boldsymbol{\alpha}}_k-\widetilde{\boldsymbol{\beta}}^*\|_2\leq a_n$.

Define

$$\begin{aligned}
C_{n,h,j}(\widetilde{\boldsymbol{\alpha}})&=\frac{1}{n}\sum_{i=1}^n\tilde{\boldsymbol{v}}_j^{\mathrm{T}}y_i\widetilde{\boldsymbol{X}}_i\left[H\left(\frac{1-y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)-I\{\epsilon_i\geq 0\}+\frac{\epsilon_i}{h}H'\left(\frac{1-y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right)\right]\\
&\triangleq\frac{1}{n}\sum_{i=1}^n\tilde{\boldsymbol{v}}_j^{\mathrm{T}}y_i\widetilde{\boldsymbol{X}}_iH_{h,i}(\widetilde{\boldsymbol{\alpha}}).
\end{aligned}$$

According to the proof of Proposition 4.1 in Chen et al. (2018), it is enough to show that

$$\sup_j\sup_k|C_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k)|=O_{\mathbb{P}}\left(\sqrt{\frac{ph\log n}{n}}+a_n^2+h^2\right).$$

Since H and $xH'(x)$ are bounded, it is easy to see that

$$|H_{h,i}(\widetilde{\boldsymbol{\alpha}})|\leq C(1+|\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}(\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*)|/\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2)=:K(\widetilde{\boldsymbol{X}}_i,\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}}^*).$$

By Lemma 3, we have for some $t>0$,

$$\mathbb{E}(\tilde{\boldsymbol{v}}_j^{\mathrm{T}}y_i\widetilde{\boldsymbol{X}}_iH_{h,i}(\widetilde{\boldsymbol{\alpha}}))^2\exp(t|\tilde{\boldsymbol{v}}_j^{\mathrm{T}}\widetilde{\boldsymbol{X}}_i|K(\widetilde{\boldsymbol{X}}_i,\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}}^*))\leq Ch(1+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2/h+\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2^2/h^2).$$

By $\sqrt{p\log n}=o(\sqrt{nh})$ and Lemma 1 in Cai and Liu (2011), we can get for any $\gamma>0$, there exists a constant C such that

$$\sup_j\sup_k\mathbb{P}\left(|C_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k)-\mathbb{E}C_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k)|\geq C\sqrt{\frac{ph\log n}{n}}\right)=O(n^{-\gamma p}).$$

The remaining work is to give a bound for $\mathbb{E}C_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k)$. According to Lemma 1, we know that $\mathbb{E}C_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k)=O(h^2+\|\widetilde{\boldsymbol{\alpha}}_k-\widetilde{\boldsymbol{\beta}}^*\|_2^2)$. Hence, $\sup_j\sup_k|\mathbb{E}C_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k)|=O(h^2+\|\widetilde{\boldsymbol{\alpha}}_k-\widetilde{\boldsymbol{\beta}}^*\|_2^2)$. Combining with the above analysis, the proof is completed. ■

Proof of Proposition 4. For simplicity, denote $\boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)$ by $\boldsymbol{D}$. According to the proof of Lemma 3 in Cai et al. (2010), for $\boldsymbol{D}_{n,h}$ we have

$$\|\boldsymbol{D}_{n,h}-\boldsymbol{D}\|\leq 10\sup_{j\leq b_{p+1}}|\tilde{\boldsymbol{v}}_j^{\mathrm{T}}(\boldsymbol{D}_{n,h}-\boldsymbol{D})\tilde{\boldsymbol{v}}_j|.$$

where $\tilde{\boldsymbol{v}}_j$, $1\leq j\leq b_{p+1}$, are some non-random vectors with $\|\tilde{\boldsymbol{v}}_j\|_2=1$ and $b_{p+1}\leq 5^{p+1}$. Define

$$D_{n,h,j}(\widetilde{\boldsymbol{\alpha}})=\frac{1}{nh}\sum_{i=1}^n(\tilde{\boldsymbol{v}}_j^{\mathrm{T}}\widetilde{\boldsymbol{X}}_i)^2H'\left(\frac{1-y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}}{h}\right).$$

When $\|\widetilde{\boldsymbol{\beta}}_0 - \widetilde{\boldsymbol{\beta}}^*\|_2 \le a_n$, then

$$\sup_{j\le b_{p+1}} |\tilde{\boldsymbol{v}}_j^{\mathrm{T}}(\boldsymbol{D}_{n,h} - \boldsymbol{D})\tilde{\boldsymbol{v}}_j| \le \sup_{j\le b_{p+1}} \sup_{\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2\le a_n} |D_{n,h,j}(\widetilde{\boldsymbol{\alpha}}) - \tilde{\boldsymbol{v}}_j^{\mathrm{T}}\boldsymbol{D}\tilde{\boldsymbol{v}}_j|.$$

As the proof of Lemma 3, we obtain that

$$\mathbb{E}\left[\tilde{\boldsymbol{v}}^{\mathrm{T}}\widetilde{\boldsymbol{x}}_i H'\left(\frac{\epsilon_i - y_i\widetilde{\boldsymbol{x}}_i^{\mathrm{T}}\Delta(\widetilde{\boldsymbol{\alpha}})}{h}\right)\right]^2 = O(h).$$

According to the proof of Proposition 4.2 in Chen et al. (2018), $D_{n,h,j}$ satisfies

$$\sup_j \sup_k |D_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k) - \mathbb{E}D_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k)| = O_{\mathbb{P}}\left(\sqrt{\frac{p\log n}{nh}}\right).$$

The remaining work is to give a bound for $\mathbb{E}D_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k) - \tilde{\boldsymbol{v}}_j^{\mathrm{T}}\boldsymbol{D}\tilde{\boldsymbol{v}}_j$. From Lemma 2, we obtain that

$$\mathbb{E}D_{n,h,j}(\widetilde{\boldsymbol{\alpha}}) - \tilde{\boldsymbol{v}}_j^{\mathrm{T}}\boldsymbol{D}\tilde{\boldsymbol{v}}_j = O(h + \|\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\beta}}^*\|_2).$$

Hence, $\sup_j \sup_k |\mathbb{E}D_{n,h,j}(\widetilde{\boldsymbol{\alpha}}_k) - \tilde{\boldsymbol{v}}_j^{\mathrm{T}}\boldsymbol{D}\tilde{\boldsymbol{v}}_j| = O(h + \|\widetilde{\boldsymbol{\alpha}}_k - \widetilde{\boldsymbol{\beta}}^*\|_2)$. Combining with the above analysis, the proof of the proposition is completed. ■

Proof of Theorem 1 and 2. We first assume that $\|\widetilde{\boldsymbol{\beta}}_0 - \widetilde{\boldsymbol{\beta}}^*\| = O_{\mathbb{P}}(a_n)$ with $a_n = o(1)$ and $a_n = O(h)$. For independent random vectors $\{(y_i, \widetilde{\boldsymbol{X}}_i), i = 1, ..., n\}$ with $\sup_j \mathbb{E}|X_j|^3 = O(1)$, we can see that

$$\|\frac{1}{n}\sum_{i=1}^n y_i\widetilde{\boldsymbol{X}}_i I(\epsilon_i \ge 0)\|_2 = O_{\mathbb{P}}(\sqrt{p/n}).$$

Note that $\|\widetilde{\boldsymbol{\beta}}_0 - \widetilde{\boldsymbol{\beta}}^*\| = o_{\mathbb{P}}(1)$ and $\|\widetilde{\boldsymbol{\beta}}^*\|_2 \le C$. From the above result and Proposition 3 and 4 we know that for the estimator $\widetilde{\boldsymbol{\beta}}$ and the true parameter $\widetilde{\boldsymbol{\beta}}^*$,

$$\widetilde{\boldsymbol{\beta}} - \widetilde{\boldsymbol{\beta}}^* = \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\left(\frac{1}{n}\sum_{i=1}^n y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i \ge 0\} - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}^*\end{pmatrix} + \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}^* - \boldsymbol{\beta}_0\end{pmatrix}\right) + \boldsymbol{r}_n.$$

with

$$\|\boldsymbol{r}_n\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{p^2\log n}{n^2 h}} + \sqrt{\frac{p\log n}{nh}}\lambda + \sqrt{\frac{ph\log n}{n}} + \lambda h + h^2\right).$$

Since $\lambda \le h$ and $\|\widetilde{\boldsymbol{\beta}}_0 - \widetilde{\boldsymbol{\beta}}^*\| = O_{\mathbb{P}}(a_n)$, we have

$$\widetilde{\boldsymbol{\beta}} - \widetilde{\boldsymbol{\beta}}^* = \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1}\left(\frac{1}{n}\sum_{i=1}^n y_i\widetilde{\boldsymbol{X}}_i I\{\epsilon_i \ge 0\} - \lambda\begin{pmatrix}0\\ \boldsymbol{\beta}^*\end{pmatrix}\right) + \boldsymbol{r}_n.$$

with

$$\|\boldsymbol{r}_n\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{p^2\log n}{n^2 h}} + \sqrt{\frac{ph\log n}{n}} + h^2\right). \tag{28}$$

Note that $h_g \ge \sqrt{p/n}$, then $\sqrt{\frac{p^2\log n}{n^2 h_g}} \le \sqrt{\frac{ph_g\log n}{n}}$.

For $q = 1$, it is easy to see that Theorem 1 holds. Suppose the theorem holds for $q = g-1$ with $g \ge 2$. Note that $p = O(m/(\log n)^2)$ and $\lambda = O(1/\log n)$, then $h_{g-1} = O(1/\log n)$ and we have $\sqrt{ph_{(g-1)}(\log n)/n} = O(\sqrt{p/n})$. Then we have $a_n = \max\{\lambda^2, \sqrt{p/n}, (p/m)^{2^{g-2}}\} = O(h_g)$ for $q = g$ with initial estimator $\widehat{\beta}_0 = \widehat{\beta}^{(g-1)}$. Now we complete the proof of Theorem 1 by (28). Theorem 2 follows directly from Theorem 1 and the Lindeberg-Feller central limit theorem.

■

Proof of Theorem 3. To prove Theorem 3, we first introduce the following lemma, which shows that $\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\beta}}^{(q)})$ is a consistent estimator of $\boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)$.

Lemma 4 *Under the conditions of Theorem 2 and $a_n \to 0$, we have*

$$\sup_{\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2\le a_n} \|\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\alpha}}) - \boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)\| = o_{\mathbb{P}}(1).$$

Proof of Lemma 4. Let $\tilde{\boldsymbol{v}}_j$, $1 \le j \le b_{p+1}$ be defined as in the proof Proposition 4. Define

$$G_{n,j}(\widetilde{\boldsymbol{\alpha}}) = \frac{1}{n}\sum_{i=1}^n (\tilde{\boldsymbol{v}}_j^{\mathrm{T}}\widetilde{\boldsymbol{X}}_i)^2 I\{1 - Y\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}} \ge 0\}.$$

By (C3) and Lemma 1 in Cai and Liu (2011), we can show that, for any $\gamma > 0$, there exists a constant $C > 0$ such that

$$\max_j \sup_{\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\beta}}^*\|_2\le a_n} \mathbb{P}\left(\left|G_{n,j}(\widetilde{\boldsymbol{\alpha}}) - \mathbb{E}G_{n,j}(\widetilde{\boldsymbol{\alpha}})\right| \ge C\sqrt{\frac{p\log n}{n}}\right) = O(n^{-\gamma p}).$$

Let $\widetilde{\boldsymbol{\alpha}}_k$, $1 \le k \le s_p$, be defined as in the proof of Proposition 3. Therefore

$$\max_{1\le j\le b_{p+1}} \max_{1\le k\le s_p} \left|G_{n,j}(\widetilde{\boldsymbol{\alpha}}_k) - \mathbb{E}G_{n,j}(\widetilde{\boldsymbol{\alpha}}_k)\right| = O_{\mathbb{P}}\left(\sqrt{\frac{p\log n}{n}}\right). \tag{29}$$

Put $t_M = 2\sqrt{p+1}a_n/n^M$. In the following, we show that

$$\mathbb{P}\left(\max_j \max_k \sup_{\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\alpha}}_k\|_2\le t_M} \left|G_{n,j}(\widetilde{\boldsymbol{\alpha}}) - G_{n,j}(\widetilde{\boldsymbol{\alpha}}_k)\right| \ge C\sqrt{\frac{p\log n}{n}}\right) = o(1) \tag{30}$$

and

$$\max_j \max_k \sup_{\|\widetilde{\boldsymbol{\alpha}}-\widetilde{\boldsymbol{\alpha}}_k\|_2\le t_M} \left|\mathbb{E}G_{n,j}(\widetilde{\boldsymbol{\alpha}}) - \mathbb{E}G_{n,j}(\widetilde{\boldsymbol{\alpha}}_k)\right| \le C\sqrt{\frac{p\log n}{n}}. \tag{31}$$

Note that for $\|\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\alpha}}_k\|_2 \le t_M$,

$$\left|I\{1 - y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}} \ge 0\} - I\{1 - y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}_k \ge 0\}\right| \le I\{-t_M\|\widetilde{\boldsymbol{X}}_i\|_2 \le 1 - y_i\widetilde{\boldsymbol{X}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\alpha}}_k \le t_M\|\widetilde{\boldsymbol{X}}_i\|_2\}$$

$$\leq \quad I\{-t_M n \leq 1 - y_i \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\alpha}}_k \leq t_M n\} + I\{\|\widetilde{\boldsymbol{X}}_i\|_2 \geq n\}.$$

By (C3), we have

$$\mathbb{P}(\max_{1\leq i\leq n} \|\widetilde{\boldsymbol{X}}_i\|_2 \geq n) = o(1). \tag{32}$$

Define

$$Z_{i,j,k}(\widetilde{\boldsymbol{\alpha}}_k) = (\tilde{\boldsymbol{v}}_j^{\mathrm{T}} \widetilde{\boldsymbol{X}}_i)^2 I\{-t_M n \leq 1 - y_i \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\alpha}}_k \leq t_M n\}.$$

By (C2) and (C3), we have

$$\mathbb{E} Z_{i,j,k}(\widetilde{\boldsymbol{\alpha}}_k) \leq \sqrt{\mathbb{E}(\tilde{\boldsymbol{v}}_j^{\mathrm{T}} \widetilde{\boldsymbol{X}}_i)^4} \sqrt{\mathbb{P}(-t_M n \leq 1 - y_i \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\alpha}}_k \leq t_M n)} = O(\sqrt{t_M n}) \tag{33}$$

and

$$\mathbb{E}(Z_{i,j,k}(\widetilde{\boldsymbol{\alpha}}_k))^2 \exp(t_0 Z_{i,j,k}(\widetilde{\boldsymbol{\alpha}}_k)) = O(\sqrt{t_M n}).$$

Note that

$$\sup_{\|\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\alpha}}_k\|_2 \leq t_M} \left| G_{n,j}(\widetilde{\boldsymbol{\alpha}}) - G_{n,j}(\widetilde{\boldsymbol{\alpha}}_k) \right| I\{\max_{1\leq i\leq n} \|\widetilde{\boldsymbol{X}}_i\|_2 < n\} \leq \frac{1}{n} \sum_{i=1}^n Z_{i,j,k}(\widetilde{\boldsymbol{\alpha}}_k). \tag{34}$$

By (C3) and Lemma 1 in Cai and Liu (2011), for any $\gamma > 0$, there exists a constant $C > 0$ such that

$$\max_{j,k} \mathbb{P}\left(\left| \frac{1}{n} \sum_{i=1}^n (Z_{i,j,k}(\widetilde{\boldsymbol{\alpha}}_k) - \mathbb{E} Z_{i,j,k}(\widetilde{\boldsymbol{\alpha}}_k)) \right| \geq C \sqrt{\frac{p \log n}{n}} \right) = O(n^{-\gamma p}).$$

By (32)-(34), we can see that (30) and (31) hold.

By (29), (30) and the definition of $\tilde{\boldsymbol{v}}_j$, we have

$$\sup_{\|\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\beta}}^*\|_2 \leq a_n} \|\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\alpha}}) - \boldsymbol{G}(\widetilde{\boldsymbol{\alpha}})\| = O_{\mathbb{P}}\Big(\sqrt{\frac{p \log n}{n}}\Big).$$

Moreover, we have

$$\begin{aligned} \left| I\{1 - y_i \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\alpha}} \geq 0\} - I\{1 - y_i \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\beta}}^* \geq 0\} \right| \quad \leq \quad & I\{-\sqrt{a_n} \leq 1 - y_i \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\alpha}}_k \leq \sqrt{a_n}\} \\ & + I\{|\widetilde{\boldsymbol{X}}_i^{\mathrm{T}} (\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\beta}}^*)| / \|\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\beta}}^*\|_2 \geq a_n^{-1/2}\}. \end{aligned}$$

By this inequality, (C2) and (C3), it is easy to show that

$$\mathbb{E}(\tilde{\boldsymbol{v}}^{\mathrm{T}} \widetilde{\boldsymbol{X}}_i)^2 I\{1 - Y \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\alpha}} \geq 0\} - \mathbb{E}(\tilde{\boldsymbol{v}}^{\mathrm{T}} \widetilde{\boldsymbol{X}}_i)^2 I\{1 - Y \widetilde{\boldsymbol{X}}_i^{\mathrm{T}} \widetilde{\boldsymbol{\beta}}^* \geq 0\} = o(1)$$

uniformly in $\|\boldsymbol{v}\|_2 = 1$ and $\|\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\beta}}^*\|_2 \leq a_n$. This implies that $\sup_{\|\widetilde{\boldsymbol{\alpha}} - \widetilde{\boldsymbol{\beta}}^*\|_2 \leq a_n} \|\boldsymbol{G}(\widetilde{\boldsymbol{\alpha}}) - \boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*)\| = o_{\mathbb{P}}(1)$. ■

Now we prove Theorem 3. Without loss of generality, we can assume that $\|\tilde{\boldsymbol{v}}\|_2 = 1$. By the consistency of $\widehat{\boldsymbol{D}}(\widetilde{\boldsymbol{\beta}}^{(q-1)})$ and $\widehat{\boldsymbol{G}}(\widetilde{\boldsymbol{\beta}}^{(q)})$ (see Lemma 4), we have

$$\widehat{\sigma}_{n,q} \to \sqrt{\tilde{\boldsymbol{v}}^{\mathrm{T}} \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1} \boldsymbol{G}(\widetilde{\boldsymbol{\beta}}^*) \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*)^{-1} \tilde{\boldsymbol{v}}}$$

as $n, p \to \infty$. This implies the theorem. ■

A.3. Proof of Auxiliary Results

Proof of Proposition 2. By Proposition 4, in the g-th iteration, we have

$$\left\| N^{-1} \sum_{k=1}^N \boldsymbol{V}_k - \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*) \right\| = O_{\mathbb{P}}\left(\sqrt{\frac{p \log n}{n h_g}} + h_g \right).$$

Therefore, it suffices to show that

$$\left\| \widehat{\boldsymbol{V}}_1 - \boldsymbol{D}(\widetilde{\boldsymbol{\beta}}^*) \right\| = O_{\mathbb{P}}\left(m^{-\delta} \right)$$

for some $\delta > 0$. With the notation in the proof of Proposition 4, we have

$$\sup_j \sup_k |D_{m,h,j}(\widetilde{\boldsymbol{\alpha}}_k) - \mathbb{E} D_{m,h,j}(\widetilde{\boldsymbol{\alpha}}_k)| = O_{\mathbb{P}}\left(\sqrt{\frac{p \log n}{m h}} \right)$$

with $h = \sqrt{p/m}$. Also, $|\mathbb{E} D_{m,h,j}(\widetilde{\boldsymbol{\alpha}}_k) - \tilde{\boldsymbol{v}}_j^{\mathrm{T}} \boldsymbol{D} \tilde{\boldsymbol{v}}_j| = O(h + \|\widetilde{\boldsymbol{\alpha}}_k - \widetilde{\boldsymbol{\beta}}^*\|_2)$ uniformly in j, k. This completes the proof as $p = O(m^\gamma)$ for some $0 < \gamma < 1$. ■

Proof of Claim 1. Let us define $\bar{\boldsymbol{\epsilon}} = Y \boldsymbol{\epsilon} = (\bar{\epsilon}^1, \ldots, \bar{\epsilon}^p)$. It is easy to show it follows normal distribution $\mathcal{N}(0, \sigma^2 \mathbf{I})$. By the construction of $\boldsymbol{X}$ (i.e., $\boldsymbol{X} = Y \mathbf{1} + \boldsymbol{\epsilon}$), we have $Y \boldsymbol{X} = Y^2 \mathbf{1} + Y \boldsymbol{\epsilon} = \mathbf{1} + \bar{\boldsymbol{\epsilon}}$ and $Y \widetilde{\boldsymbol{X}} = \binom{Y}{\mathbf{1} + \bar{\boldsymbol{\epsilon}}}$. Recall that $S(\widetilde{\boldsymbol{\beta}}^*) = -\mathbb{E}[I\{1 - Y \widetilde{\boldsymbol{X}}^{\mathrm{T}} \widetilde{\boldsymbol{\beta}}^* \geq 0\} Y \widetilde{\boldsymbol{X}}]$. Therefore we have

$$S(\widetilde{\boldsymbol{\beta}}^*) = -\mathbb{E}\left[I\left\{ 1 - \frac{1}{a} (\mathbf{1} + \bar{\boldsymbol{\epsilon}})^{\mathrm{T}} \mathbf{1} \geq 0 \right\} \begin{pmatrix} Y \\ \mathbf{1} + \bar{\boldsymbol{\epsilon}} \end{pmatrix} \right] = -\mathbb{E}\left[I\{a \geq p + \mathbf{1}^{\mathrm{T}} \bar{\boldsymbol{\epsilon}}\} \begin{pmatrix} Y \\ \mathbf{1} + \bar{\boldsymbol{\epsilon}} \end{pmatrix} \right].$$

In order to show that $S(\widetilde{\boldsymbol{\beta}}^*) = 0$, we only need to show that $\mathbb{E}\left[I\{a \geq p + \mathbf{1}^{\mathrm{T}} \bar{\boldsymbol{\epsilon}}\} Y \right] = 0$ and $\mathbb{E}\left[I\{a \geq p + \mathbf{1}^{\mathrm{T}} \bar{\boldsymbol{\epsilon}}\} (1 + \bar{\epsilon}^i) \right] = 0$ for $i = 1, \ldots, p$. The first equation holds because Y is independent of $\bar{\boldsymbol{\epsilon}}$ and $\mathbb{E}[Y] = p_+ - p_- = 0$. To show the second equation, we note that for any $i, j \in \{1, \ldots, p\}$, we have $\mathbb{E}[I\{a \geq p + \mathbf{1}^{\mathrm{T}} \bar{\boldsymbol{\epsilon}}\}(1 + \bar{\epsilon}^j)] = \mathbb{E}[I\{a \geq p + \mathbf{1}^{\mathrm{T}} \bar{\boldsymbol{\epsilon}}\}(1 + \bar{\epsilon}^i)]$ by distributional symmetry of $\bar{\epsilon}^i$ and $\bar{\epsilon}^j$. Therefore it is enough to show that

$$\sum_{j=1}^p \mathbb{E}[I\{a \geq p + \mathbf{1}^{\mathrm{T}} \bar{\boldsymbol{\epsilon}}\}(1 + \bar{\epsilon}^j)] = \mathbb{E}[I\{\nu \leq a\} \nu] = 0,$$

where $\nu = p + \mathbf{1}^{\mathrm{T}} \bar{\boldsymbol{\epsilon}}$. Recall that a satisfies $\int_{-\infty}^a \phi_1(x) x dx = 0$ where $\phi_1(x)$ is the p.d.f. of the distribution $\mathcal{N}(p, \sigma^2 p)$. Since ν follows the normal distribution $\mathcal{N}(p, \sigma^2 p)$, we have $E[I\{\nu \leq a\} \nu] = 0$. Therefore we have shown that $S(\widetilde{\boldsymbol{\beta}}^*) = 0$. By the convexity of the loss function and uniqueness of the minimizer, we have proved that $\widetilde{\boldsymbol{\beta}}^*$ is the true coefficient under the given setting. ■

References

R Raj Bahadur. A note on quantiles in large samples. *The Annals of Mathematical Statistics*, 37(3):577–580, 1966.

Moulinath Banerjee, Cecile Durot, and Bodhisattva Sen. Divide and conquer in non-standard problems and the super-efficiency phenomenon. *Ann. Statist.*, (To appear), 2018.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *J. Amer. Statist. Assoc.*, 101(473):138–156, 2006.

Heather Battey, Jianqing Fan, Han Liu, Junwei Lu, and Ziwei Zhu. Distributed estimation and inference with statistical guarantees. *Ann. Statist.*, (To appear), 2018.

Gilles Blanchard, Olivier Bousquet, and Pascal Massart. Statistical performance of support vector machines. *Ann. Statist.*, 36(2):489–531, 2008.

T Tony Cai, Cun-Hui Zhang, and Harrison H Zhou. Optimal rates of convergence for covariance matrix estimation. *Ann. Statist.*, 38(4):2118–2144, 2010.

Tony Cai and Weidong Liu. Adaptive thresholding for sparse covariance matrix estimation. *J. Amer. Statist. Assoc.*, 106(494):672–684, 2011.

Probal Chaudhuri. Nonparametric estimates of regression quantiles and their local bahadur representation. *Ann. Statist.*, 19(2):760–777, 1991.

Xi Chen, Weidong Liu, and Yichen Zhang. Quantile regression under memory constraint. *arXiv preprint arXiv:1810.08264*, 2018.

Xueying Chen and Min-ge Xie. A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, 24(4):1655–1684, October 2014.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge university press, 2000.

Jianqing Fan, Dong Wang, Kaizheng Wang, and Ziwei Zhu. Distributed estimation of principal eigenspaces. *arXiv preprint arXiv:1702.06488*, 2017.

Pedro A Forero, Alfonso Cano, and Georgios B Giannakis. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11:1663–1707, 2010.

Hans P Graf, Eric Cosatto, Leon Bottou, Igor Dourdanovic, and Vladimir Vapnik. Parallel support vector machines: the cascade SVM. In *Proceedings of the Advances in Neural Information Processing Systems*, 2005.

Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of Conjugate Gradients for Solving Linear Systems*, volume 49. NBS Washington, DC, 1952.

Joel L Horowitz. Bootstrap methods for median regression models. *Econometrica*, pages 1327–1351, 1998.

Cho-Jui Hsieh, Si Si, and Inderjit Dhillon. A divide-and-conquer solver for kernel support vector machines. In *Proceedings of the International Conference on Machine Learning*, 2014.

Cheng Huang and Xiaoming Huo. A distributed one-step estimator. *arXiv preprint arXiv:1511.01443v2*, 2015.

Hanwen Huang. Asymptotic behavior of support vector machine for spiked population model. *Journal of Machine Learning Research*, 18(45):1–21, 2017.

Bo Jiang, Xuegong Zhang, and Tianxi Cai. Estimating the confidence interval for prediction errors of support vector machine classifiers. *Journal of Machine Learning Research*, 9: 521–540, 2008.

Michael I Jordan, Jason D Lee, and Yun Yang. Communication-efficient distributed statistical inference. *J. Amer. Statist. Assoc.*, (To appear), 2018.

Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 33–50, 1978.

Ja-Yong Koo, Yoonkyung Lee, Yuwon Kim, and Changyi Park. A bahadur representation of the linear support vector machine. *Journal of Machine Learning Research*, 9:1343–1368, 2008.

Jason D Lee, Qiang Liu, Yuekai Sun, and Jonathan E Taylor. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(5):1–30, 2017.

Sangkyun Lee and Stephen J Wright. Approximate stochastic subgradient estimation training for support vector machines. *arXiv preprint arXiv:1111.0432*, 2011.

Runze Li, Dennis KJ Lin, and Bing Li. Statistical inference in massive data sets. *Appl. Stoch. Model Bus.*, 29(5):399–409, 2013.

Heng Lian and Zengyan Fan. Divide-and-conquer for debiased ℓ_1-norm support vector machine in ultra-high dimensions. *Journal of Machine Learning Research*, 18(1):6691–6716, 2017.

Yi Lin. Some asymptotic properties of the support vector machine. Technical report, University of Wisconsin-Madison, 1999.

Yufeng Liu, Hao Helen Zhang, Cheolwoo Park, and Jeongyoun Ahn. Support vector machines with adaptive lq penalty. *Computational Statistics & Data Analysis*, 51(12):6380–6394, 2007.

Lei Pang, Wenbin Lu, and Huixia Judy Wang. Variance estimation in censored quantile regression via induced smoothing. *Computational Statistics & Data Analysis*, 56(4):785–796, 2012.

Bo Peng, Lan Wang, and Yichao Wu. An error bound for ℓ_1-norm support vector machine coefficients in ultra-high dimension. *Journal of Machine Learning Research*, 17(236):1–26, 2016.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2008.

Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the International Conference on Machine Learning*, 2014.

Chengchun Shi, Wenbin Lu, and Rui Song. A massive data framework for m-estimators with cubic-rate. *J. Amer. Statist. Assoc.*, (To appear), 2017.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–492, 2012.

Stanislav Volgushev, Shih-Kang Chao, and Guang Cheng. Distributed inference for quantile regression processes. *arXiv preprint arXiv:1701.06088*, 2017.

Dongli Wang and Yan Zhou. Distributed support vector machines: an overview. In *IEEE Control and Decision Conference (CCDC)*, 2012.

Jialei Wang, Mladen Kolar, Nathan Srebro, and Tong Zhang. Efficient distributed learning with sparsity. In *Proceedings of the International Conference on Machine Learning*, 2017.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.*, 32(1):56–85, 2004.

Xiang Zhang, Yichao Wu, Lan Wang, and Runze Li. A consistent information criterion for support vector machines in diverging model spaces. *Journal of Machine Learning Research*, 17(1):466–491, 2016a.

Xiang Zhang, Yichao Wu, Lan Wang, and Runze Li. Variable selection for support vector machines in moderately high dimensions. *J. Roy. Statist. Soc. Ser. B*, 78(1):53–76, 2016b.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015.

Tianqi Zhao, Guang Cheng, and Han Liu. A partially linear framework for massive heterogeneous data. *Ann. Statist.*, 44(4):1400–1437, 2016.

Kaihua Zhu, Hao Wang, Hongjie Bai, Jian Li, Zhihuan Qiu, Hang Cui, and Edward Y Chang. Parallelizing support vector machines on distributed computers. In *Proceedings of the Advances in Neural Information Processing Systems*, 2008.

Journal of Machine Learning Research 20 (2019) 1-34 Submitted 1/19; Published 6/19

Sharp Restricted Isometry Bounds for the Inexistence of Spurious Local Minima in Nonconvex Matrix Recovery

Richard Y. Zhang RYZ@ILLINOIS.EDU
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
306 N Wright St, Urbana, IL 61801, USA

Somayeh Sojoudi SOJOUDI@BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720, USA

Javad Lavaei LAVAEI@BERKELEY.EDU
Department of Industrial Engineering and Operations Research
University of California, Berkeley
Berkeley, CA 94720, USA

Editor: Sanjiv Kumar

Abstract

Nonconvex matrix recovery is known to contain *no spurious local minima* under a restricted isometry property (RIP) with a sufficiently small RIP constant δ. If δ is too large, however, then counterexamples containing spurious local minima are known to exist. In this paper, we introduce a proof technique that is capable of establishing *sharp* thresholds on δ to guarantee the inexistence of spurious local minima. Using the technique, we prove that in the case of a rank-1 ground truth, an RIP constant of $\delta < 1/2$ is both necessary and sufficient for exact recovery from any arbitrary initial point (such as a random point). We also prove a local recovery result: given an initial point x_0 satisfying $f(x_0) \leq (1-\delta)^2 f(0)$, any descent algorithm that converges to second-order optimality guarantees exact recovery.

Keywords: matrix factorization, nonconvex optimization, Restricted Isometry Property, matrix sensing, spurious local minima

1. Introduction

The *low-rank matrix recovery* problem seeks to recover an unknown $n \times n$ ground truth matrix $M^\star$ of low-rank $r \ll n$ from m linear measurements of $M^\star$. The problem naturally arises in recommendation systems (Rennie and Srebro, 2005) and clustering algorithms (Amit et al., 2007)—often under the names of matrix completion and matrix sensing—and also finds engineering applications in phase retrieval (Candes et al., 2013) and power system state estimation (Zhang et al., 2018b).

In the symmetric, noiseless variant of low-rank matrix recovery, the ground truth $M^\star$ is taken to be positive semidefinite (denoted as $M^\star \succeq 0$), and the m linear measurements are made without error, as in

$$b \equiv \mathcal{A}(M) \quad \text{where} \quad \mathcal{A}(M) = \begin{bmatrix} \langle A_1, M \rangle & \cdots & \langle A_m, M \rangle \end{bmatrix}^T. \tag{1}$$

©2019 Richard Y. Zhang, Somayeh Sojoudi, and Javad Lavaei.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-020.html.

To recover $M^\star$ from b, the standard approach in the machine learning community is to factor a candidate M into its low-rank factors XX^T, and to solve a nonlinear least-squares problem on X using a local search algorithm (usually stochastic gradient descent):

$$\underset{X \in \mathbb{R}^{n \times r}}{\text{minimize}} \quad f(X) \equiv \|\mathcal{A}(XX^T) - b\|^2. \tag{2}$$

The function f is nonconvex, so a "greedy" local search algorithm can become stuck at a spurious local minimum, especially if a random initial point is used. Despite this apparent risk of failure, the nonconvex approach remains both widely popular as well as highly effective in practice.

Recently, Bhojanapalli et al. (2016b) provided a rigorous theoretical justification for the empirical success of local search on problem (2). Specifically, they showed that the problem contains *no spurious local minima* under the assumption that $\mathcal{A}$ satisfies the *restricted isometry property* (RIP) of Recht et al. (2010) with a sufficiently small constant. The nonconvex problem is easily solved using local search algorithms because every local minimum is also a global minimum.

Definition 1 (Restricted Isometry Property) *The linear map $\mathcal{A} : \mathbb{R}^{n \times n} \to \mathbb{R}^m$ is said to satisfy δ-RIP if there is constant $\delta \in [0, 1)$ such that*

$$(1-\delta)\|M\|_F^2 \leq \|\mathcal{A}(M)\|^2 \leq (1+\delta)\|M\|_F^2 \tag{3}$$

holds for all $M \in \mathbb{R}^{n \times n}$ satisfying $\operatorname{rank}(M) \leq 2r$.

Theorem 2 (Bhojanapalli et al., 2016b; Ge et al., 2017) *Let $\mathcal{A}$ satisfy δ-RIP with $\delta < 1/5$. Then, (2) has no spurious local minima:*

$$\nabla f(X) = 0, \quad \nabla^2 f(X) \succeq 0 \quad \Longleftrightarrow \quad XX^T = M^\star.$$

Hence, any algorithm that converges to a second-order critical point is guaranteed to recover $M^\star$ exactly.

While Theorem 2 says that an RIP constant of $\delta < 1/5$ is *sufficient* for exact recovery, Zhang et al. (2018a) proved that $\delta < 1/2$ is *necessary*. Specifically, they gave a counterexample satisfying 1/2-RIP that causes randomized stochastic gradient descent to fail 12% of the time. A number of previous authors have attempted to close the gap between sufficiency and necessity, including Bhojanapalli et al. (2016b); Ge et al. (2017); Park et al. (2017); Zhang et al. (2018a); Zhu et al. (2018). In this paper, we prove that in the rank-1 case, an RIP constant of $\delta < 1/2$ is both *necessary* and *sufficient* for exact recovery.

Once the RIP constant exceeds $\delta \geq 1/2$, global guarantees are no longer possible. Zhang et al. (2018a) proved that counterexamples exist *generically*: almost every choice of $x, z \in \mathbb{R}^n$ generates an instance of nonconvex recovery satisfying RIP with x as a spurious local minimum and $M^\star = zz^T$ as ground truth. In practice, local search may continue to work well, often with a 100% success rate as if spurious local minima do not exist. However, the inexistence of spurious local minima can no longer be assured.

Instead, we turn our attention to local guarantees, based on good initial guesses that often arise from domain expertise, or even chosen randomly. Given an initial point x_0

satisfies $f(x_0) \leq (1-\delta)^2\|M^\star\|_F^2$ where δ is the RIP constant and $M^\star = zz^T$ is a rank-1 ground truth, we prove that a *descent* algorithm that converges to second-order optimality is guaranteed to recover the ground truth. Examples of such algorithms include randomized full-batch gradient descent (Jin et al., 2017) and trust-region methods (Conn et al., 2000; Nesterov and Polyak, 2006).

2. Main Results

Our main contribution in this paper is a proof technique capable of establishing RIP thresholds that are both *necessary* and *sufficient* for exact recovery. The key idea is to disprove the counterfactual. To prove for some $\lambda \in [0,1)$ that "λ-RIP implies no spurious local minima", we instead establish the inexistence of a counterexample that admits a spurious local minimum despite satisfying λ-RIP. In particular, if $\delta^\star$ is the *smallest* RIP constant associated with a counterexample, then any $\lambda < \delta^\star$ cannot admit a counterexample (or it would contradict the definition of $\delta^\star$ as the smallest RIP constant). Accordingly, $\delta^\star$ is precisely the sharp threshold needed to yield a necessary and sufficient recovery guarantee.

The main difficulty with the above line of reasoning is the need to optimize over the set of counterexamples. Indeed, verifying RIP for a fixed operator $\mathcal{A}$ is already NP-hard in general (Tillmann and Pfetsch, 2014), so it is reasonable to expect that optimizing over the set of RIP operators is at least NP-hard. Surprisingly, this is not the case. Consider finding the smallest RIP constant associated with a counterexample with *fixed* ground truth $M^\star = ZZ^T$ and *fixed* spurious point X:

$$\begin{aligned} \delta(X,Z) \quad \equiv \quad & \underset{\mathcal{A}}{\text{minimum}} \quad \delta \\ & \text{subject to} \quad f(X) = \frac{1}{2}\|\mathcal{A}(XX^T - ZZ^T)\|^2 \\ & \qquad\qquad \nabla f(X) = 0, \quad \nabla^2 f(X) \succeq 0 \\ & \qquad\qquad \mathcal{A} \text{ satisfies } \delta\text{-RIP}. \end{aligned} \tag{4}$$

In Section 5, we reformulate problem (4) into a *convex* linear matrix inequality (LMI) optimization, and prove that the reformulation is *exact* (Theorem 8). Accordingly, we can evaluate $\delta(X,Z)$ to arbitrary precision in polynomial time by solving an LMI using an interior-point method.

In the rank $r=1$ case, the LMI reformulation is sufficiently simple that it can be relaxed and then solved in closed-form (Theorem 12). This yields a lower-bound $\delta_{\text{lb}}(x,z) \leq \delta(x,z)$ that we optimize over all spurious choices of $x \in \mathbb{R}^n$ to prove that $\delta^\star \geq 1/2$. Given that $\delta^\star \leq 1/2$ due to the counterexample of Zhang et al. (2018a), we must actually have $\delta^\star = 1/2$.

Theorem 3 (Global guarantee) *Let $r = \text{rank}\,(M^\star) = 1$, let $\mathcal{A}$ satisfy δ-RIP, and define $f(x) = \|\mathcal{A}(xx^T - M^\star)\|^2$.*

- *If $\delta < 1/2$, then f has no spurious local minima:*

$$\nabla f(x) = 0, \quad \nabla^2 f(x) \succeq 0 \quad \Longleftrightarrow \quad xx^T = M^\star.$$

- *If $\delta \geq 1/2$, then there exists a counterexample $\mathcal{A}^\star$ satisfying δ-RIP, but whose $f^\star(x) = \|\mathcal{A}^\star(xx^T - M^\star)\|^2$ admits a spurious point $x \in \mathbb{R}^n$ satisfying:*

$$\|x\|^2 = \frac{1}{2}\|M^\star\|_F, \qquad f(x) = \frac{3}{4}\|M^\star\|_F^2, \qquad \nabla f(x) = 0, \qquad \nabla^2 f(x) \succeq 8\,xx^T.$$

We can also optimize $\delta_{\text{lb}}(x,z)$ over spurious choices $x \in \mathbb{R}^n$ within an ϵ-neighborhood of the ground truth. The resulting guarantee is applicable to much larger RIP constants δ, including those arbitrarily close to one.

Theorem 4 (Local guarantee) *Let $r = \text{rank}\,(M^\star) = 1$, and let $\mathcal{A}$ satisfy δ-RIP. If*

$$\delta < \left(1 - \frac{\epsilon^2}{2(1-\epsilon)}\right)^{1/2} \qquad \text{where } 0 \leq \epsilon \leq \frac{\sqrt{5}-1}{2}$$

then $f(x) = \|\mathcal{A}(xx^T - M^\star)\|^2$ has no spurious local minima within an ϵ-neighborhood of the solution:

$$\nabla f(x) = 0, \quad \nabla^2 f(x) \succeq 0, \quad \|xx^T - M^\star\|_F \leq \epsilon\|M^\star\|_F \quad \Longleftrightarrow \quad xx^T = M^\star.$$

Theorem 4 gives an RIP-based exact recovery guarantee for *descent* algorithms, such as randomized full-batch gradient descent (Jin et al., 2017) and trust-region methods (Conn et al., 2000; Nesterov and Polyak, 2006), that generate a sequence of iterates $x_1, x_2, \ldots, x_k$ from an initial guess x_0 with each iterate no worse than the one before:

$$f(x_k) \leq \cdots \leq f(x_2) \leq f(x_1) \leq f(x_0). \tag{5}$$

Heuristically, it also applies to nondescent algorithms, like stochastic gradient descent and Nesterov's accelerated gradient descent, under the mild assumption that the final iterate x_k is no worse than the initial guess x_0, as in $f(x_k) \leq f(x_0)$.

Corollary 5 *Let $r = \text{rank}\,(M^\star) = 1$, and let $\mathcal{A}$ satisfy δ-RIP. If $x_0 \in \mathbb{R}^n$ satisfies*

$$f(x_0) < (1-\delta)\epsilon^2 f(0) \qquad \text{where } \epsilon = \min\left\{\sqrt{1-\delta^2}, (\sqrt{5}-1)/2\right\},$$

where $f(x) = \|\mathcal{A}(xx^T - M^\star)\|^2$, then the sublevel set defined by x_0 contains no spurious local minima:

$$\nabla f(x) = 0, \quad \nabla^2 f(x) \succeq 0, \quad f(x) \leq f(x_0) \quad \Longleftrightarrow \quad xx^T = M^\star.$$

When the RIP constant satisfies $\delta \geq 0.787$, Corollary 5 guarantees exact recovery from an initial point x_0 satisfying $f(x_0) < (1-\delta)^2 f(0)$. In practice, such an x_0 can often be found using a spectral initializer (Keshavan et al., 2010a; Jain et al., 2013; Netrapalli et al., 2013; Candes et al., 2015; Chen and Candes, 2015). If δ is not too close to one, then even a random point may suffice with a reasonable probability (see the related discussion by Goldstein and Studer (2018)).

In the rank-r case with $r > 1$, our proof technique continues to work, but $\delta(X,Z)$ becomes very challenging to solve in closed-form. The exact RIP threshold $\delta^\star$ requires minimizing $\delta(X,Z)$ over *all pairs* of spurious X and ground truth Z, so the lack of a closed-form solution would be a significant impediment to further progress. Nevertheless, we can probe at an upper-bound on $\delta^\star$ by heuristically optimizing over X and Z, in each case evaluating $\delta(X,Z)$ numerically using an interior-point method. Doing this in Section 8, we obtain empirical evidence that higher-rank have larger RIP thresholds, and so are in a sense "easier" to solve.

3. Related work

3.1. No spurious local minima in matrix completion

Exact recovery guarantees like Theorem 2 have also been established for "harder" choices of $\mathcal{A}$ that do not satisfy RIP over its entire domain. In particular, the matrix completion problem has sparse measurement matrices $A_1, \ldots, A_m$, with each containing just a single nonzero element. In this case, the RIP-like condition $\|\mathcal{A}(M)\|^2 \approx \|M\|_F^2$ holds only when M is both low-rank and sufficiently dense; see the discussion by Candès and Recht (2009). Nevertheless, Ge et al. (2016) proved a similar result to Theorem 2 by adding a regularizing term to the objective.

Our recovery results are developed for the classical form of RIP—a much stronger notion than the RIP-like condition satisfied by matrix completion. Intuitively, if exact recovery cannot be guaranteed under standard RIP, then exact recovery under a weaker notion would seem unlikely. It remains future work to make this argument precise, and to extend our proof technique to these "harder" choices of $\mathcal{A}$.

3.2. Noisy measurements and nonsymmetric ground truth

Recovery guarantees for the noisy and/or nonsymmetric variants of nonconvex matrix recovery typically require a smaller RIP constant than the symmetric, noiseless case. For example, Bhojanapalli et al. (2016b) proved that the symmetric, zero-mean, σ^2-variance Gaussian noise case requires a rank-$4r$ RIP constant of $\delta < 1/10$ to recover a σ-accurate solution X satisfying $\|XX^T - M^\star\|_F \leq 20\sigma\sqrt{(\log n)/m}$. Also, Ge et al. (2017) proved that the nonsymmetric, noiseless case requires a rank-$2r$ RIP constant $\delta < 1/10$ for exact recovery. By comparison, the symmetric, noiseless case requires only a rank-$2r$ RIP constant of $\delta < 1/5$ for exact recovery.

The main goal of this paper is to develop a proof technique capable of establishing *sharp* RIP thresholds for exact recovery. As such, we have focused our attention on the symmetric, noiseless case. While our technique can be easily modified to accommodate for the nonsymmetric, noisy case, the *sharpness* of the technique (via Theorem 8) may be lost. Whether an exact convex reformulation exists for the nonsymmetric, noisy case is an open question, and the subject of important future work.

3.3. Approximate second-order points and strict saddles

Existing "no spurious local minima" results (Bhojanapalli et al., 2016b; Ge et al., 2017) guarantee that satisfying second-order optimality to ϵ-accuracy will yield a point within an ϵ-neighborhood of the solution:

$$\|\nabla f(X)\| \leq C_1\epsilon, \qquad \nabla^2 f(X) \succeq -C_2\sqrt{\epsilon}I \qquad \Longleftrightarrow \qquad \|XX^T - M^\star\|_F \leq \epsilon.$$

Such a condition is often known as "strict saddle" (Ge et al., 2015). The associated constants $C_1, C_2 > 0$ determine the rate at which gradient methods can converge to an ϵ-accurate solution (Du et al., 2017; Jin et al., 2017).

The proof technique presented in this paper can be extended in a straightforward way to the strict saddle condition. Specifically, we replace all instances of $\nabla f(X) = 0$, $\nabla^2 f(X) \succeq 0$, and $XX^T \neq M^\star$ with $\|\nabla f(X)\| \leq C_1\epsilon$, $\nabla^2 f(X) \succeq C_2\sqrt{\epsilon}I$, and $\|XX^T - M^\star\| > \epsilon$ in

Section 5, and derive a suitable version of Theorem 8. However, the resulting reformulation can no longer be solved in closed form, so it becomes difficult to extend the guarantees in Theorem 3 and Theorem 4. Nevertheless, quantifying its asymptotic behavior may yield valuable insights in understanding the optimization landscape.

3.4. Special initialization schemes

Our local recovery result is reminiscent of classic exact recovery results based on placing an initial point sufficiently close to the global optimum. Most algorithms use the spectral initializer to chose the initial point (Keshavan et al., 2010a,b; Jain et al., 2013; Netrapalli et al., 2013; Candes et al., 2015; Chen and Candes, 2015; Zheng and Lafferty, 2015; Zhao et al., 2015; Bhojanapalli et al., 2016a; Sun and Luo, 2016; Sanghavi et al., 2017; Park et al., 2018), although other initializers have also been proposed (Wang et al., 2018; Chen et al., 2018; Mondelli and Montanari, 2018). Our result differs from prior work in being completely agnostic to the specific application and the initializer. First, it requires only a suboptimality bound $f(x_0) \leq (1-\delta)^2 f(0)$ to be satisfied by the initial point x_0. Second, its sole parameter is the RIP constant δ, so issues of sample complexity are implicitly resolved in a universal way for different measurement ensembles. On the other hand, the result is not directly applicable to problems that only approximately satisfy RIP, including matrix completion.

3.5. Comparison to convex recovery

Classical theory for the low-rank matrix recovery problem is based on a quadratic lift: replacing XX^T in (2) by a convex term $M \succeq 0$, and augmenting the objective with a trace penalty $\lambda \cdot \mathrm{tr}(M)$ to induce a low-rank solution (Candès and Recht, 2009; Recht et al., 2010; Candès and Tao, 2010; Candes and Plan, 2011; Candes et al., 2013). The convex approach also enjoys RIP-based exact recovery guarantees: in the noiseless case, Cai and Zhang (2013) proved that $\delta \leq 1/2$ is sufficient, while the counterexample of Wang and Li (2013) shows that $\delta \leq 1/\sqrt{2}$ is necessary. While convex recovery may be able to solve problems with larger RIP constants than nonconvex recovery, it is also considerably more expensive. In practice, convex recovery is seldom used for large-scale datasets with n on the order of thousands to millions.

Recently, several authors have proposed *non-lifting* convex relaxations, motivated by the desire to avoid squaring the number of variables in the classic quadratic lift. In particular, we mention the PhaseMax method studied by Bahmani and Romberg (2017) and Goldstein and Studer (2018), which avoids the need to square the number of variables when both the measurement matrices $A_1, \ldots, A_m$ and the ground truth $M^\star$ are rank-1. These methods also require a good initial guess as an input, and so are in a sense very similar to nonconvex recovery.

4. Preliminaries

4.1. Notation

Lower-case letters are vectors and upper-case letters are matrices. The sets $\mathbb{R}^{n\times n} \supset \mathbb{S}^n$ are the space of $n \times n$ real matrices and real symmetric matrices, and $\langle X, Y\rangle \equiv \mathrm{tr}(X^T Y)$ and

$\|X\|_F^2 \equiv \langle X, X\rangle$ are the Frobenius inner product and norm. We write $M \succeq 0$ (resp. $M \succ 0$) to mean that M is positive semidefinite (resp. positive definite), and $M \succeq S$ to denote $M - S \succeq 0$ (resp. $M \succ S$ to denote $M - S \succ 0$).

Throughout the paper, we use $X \in \mathbb{R}^{n\times r}$ (resp. $x \in \mathbb{R}^n$) to refer to any candidate point, and $M^\star = ZZ^T$ (resp. $M^\star = zz^T$) or to refer to a rank-r (resp. rank-1) factorization of the ground truth $M^\star$. The vector $\mathbf{e}$ and matrix $\mathbf{X}$ are defined in (11). We also denote the optimal value of the nonconvex problem (15) as $\delta(X, Z)$, and later show it to be equal to the optimal value of the convex problem (21) denoted as $\mathrm{LMI}(X, Z)$.

4.2. Basic definitions

The **vectorization** operator stacks the columns of an $m \times n$ matrix A into a single column vector:

$$\mathrm{vec}\,(A) = \begin{bmatrix} A_{1,1} & \cdots & A_{m,1} & A_{1,2} & \cdots & A_{m,2} & \cdots & A_{1,n} & \cdots & A_{m,n}\end{bmatrix}^T.$$

It defines an isometry between the $m \times n$ matrices A, B and their mn underlying degrees of freedom $\mathrm{vec}\,(A), \mathrm{vec}\,(B)$:

$$\langle A, B\rangle \equiv \mathrm{tr}(A^T B) = \mathrm{vec}\,(A)^T \mathrm{vec}\,(B) \equiv \langle \mathrm{vec}\,(A), \mathrm{vec}\,(B)\rangle.$$

The **matricization** operator is the inverse of vectorization, meaning that $A = \mathrm{mat}(a)$ if and only if $a = \mathrm{vec}\,(A)$.

The **Kronecker product** between the $m \times n$ matrix A and the $p \times q$ matrix B is the $mp \times pq$ matrix defined

$$A \otimes B = \begin{bmatrix} A_{1,1}B & \cdots & A_{1,n}B \\ \vdots & \ddots & \vdots \\ A_{m,1}B & \cdots & A_{m,n}B \end{bmatrix}$$

to satisfy the Kronecker identity

$$\mathrm{vec}\,(AXB^T) = (B \otimes A)\,\mathrm{vec}\,(X).$$

The **orthogonal basis** of a given $m \times n$ matrix A (with $m \geq n$) is a matrix $P = \mathrm{orth}(A)$ comprising $\mathrm{rank}\,(A)$ orthonormal columns of length-m that span $\mathrm{range}(A)$:

$$P = \mathrm{orth}(A) \qquad \Longleftrightarrow \qquad PP^T A = A, \qquad P^T P = I_{\mathrm{rank}\,(A)}.$$

We can compute P using either a rank-revealing QR factorization (Chan, 1987) or a (thin) singular value decomposition (Golub and Van Loan, 1996, p. 254) in $O(mn^2)$ time and $O(mn)$ memory.

4.3. Global optimality and local optimality

Given a choice of $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ and the rank-r ground truth $M^\star \succeq 0$, we define the nonconvex objective

$$f : \mathbb{R}^{n\times r} \to \mathbb{R} \qquad \text{such that} \qquad f(X) = \frac{1}{2}\|\mathcal{A}(XX^T - M^\star)\|^2. \tag{6}$$

If the point X attains $f(X) = 0$, then we call it a *globally minimum*; otherwise, we call it a *spurious* point. If $\mathcal{A}$ satisfies δ-RIP, then X is a global minimum if and only if $XX^T = M^\star$ (Recht et al., 2010, Theorem 3.2).

The point X is said to be a *local minimum* if $f(X) \leq f(X')$ holds for all X' within a local neighborhood of X. If X is a local minimum, then it must satisfy the second-order *necessary* condition for local optimality:

$$\nabla f(X) = 0, \qquad \nabla^2 f(X) \succeq 0. \tag{7}$$

Conversely, a point X satisfying (7) is called a *second-order critical point*, and can be either a local minimum or a saddle point. It is worth emphasizing that local search algorithms can only guarantee convergence to a second-order critical point, and not necessarily a local minimum; see Ge et al. (2015); Lee et al. (2016); Jin et al. (2017); Du et al. (2017) for the literature on gradient methods, and Conn et al. (2000); Nesterov and Polyak (2006); Cartis et al. (2012); Boumal et al. (2018) for the literature on trust-region methods.

If a point X satisfies the second-order *sufficient* condition for local optimality (with $\mu > 0$):

$$\nabla f(X) = 0, \qquad \nabla^2 f(X) \succeq \mu I \tag{8}$$

then it is guaranteed to be a local minimum. However, it is also possible for X to be a local minimum without satisfying (8). Indeed, certifying X to be a local minimum is NP-hard in the worst case (Murty and Kabadi, 1987). Hence, the finite gap between necessary and sufficient conditions for local optimality reflects the inherent hardness of the problem.

4.4. Explicit expressions for $\nabla f(X)$ and $\nabla^2 f(X)$

Define $f(X)$ as the nonlinear least-squares objective shown in (6). While not immediately obvious, both the gradient $\nabla f(X)$ and the Hessian $\nabla^2 f(X)$ are *linear* with respect to the the *kernel* operator $\mathcal{H} \equiv \mathcal{A}^T\mathcal{A}$. To show this, we define the matrix representation of the operator $\mathcal{A}$

$$\mathbf{A} = \begin{bmatrix}\mathrm{vec}\,(A_1) & \mathrm{vec}\,(A_2) & \cdots & \mathrm{vec}\,(A_m)\end{bmatrix}^T, \tag{9}$$

which satisfies

$$\mathcal{A}(M) = \begin{bmatrix}\langle A_1, M\rangle \\ \vdots \\ \langle A_m, M\rangle\end{bmatrix} = \begin{bmatrix}\mathrm{vec}\,(A_1)^T\mathrm{vec}\,(M) \\ \vdots \\ \mathrm{vec}\,(A_m)^T\mathrm{vec}\,(M)\end{bmatrix} = \begin{bmatrix}\mathrm{vec}\,(A_1)^T \\ \vdots \\ \mathrm{vec}\,(A_m)^T\end{bmatrix}\mathrm{vec}\,(M) = \mathbf{A}\,\mathrm{vec}\,(M).$$

Then, some linear algebra reveals

$$f(X) = \frac{1}{2}\mathbf{e}^T\mathbf{A}^T\mathbf{A}\mathbf{e}, \tag{10a}$$

$$\nabla f(X) = \mathbf{X}^T\mathbf{A}^T\mathbf{A}\mathbf{e}, \tag{10b}$$

$$\nabla^2 f(X) = 2\cdot[I_r \otimes \mathrm{mat}(\mathbf{A}^T\mathbf{A}\mathbf{e})] + \mathbf{X}^T\mathbf{A}^T\mathbf{A}\mathbf{X}, \tag{10c}$$

where $\mathbf{e}$ and $\mathbf{X}$ are defined with respect to X and $M^\star$ to satisfy

$$\mathbf{e} = \mathrm{vec}\,(XX^T - M^\star), \tag{11a}$$

$$\mathbf{X}\,\mathrm{vec}\,(U) = \mathrm{vec}\,(XU^T + UX^T) \qquad \forall U \in \mathbb{R}^{n\times r}. \tag{11b}$$

(Note that $\mathbf{X}$ is simply the Jacobian of $\mathbf{e}$ with respect to X.) Clearly, $f(X)$, $\nabla f(X)$, and $\nabla^2 f(X)$ are all linear with respect to $\mathbf{H} = \mathbf{A}^T\mathbf{A}$. In turn, $\mathbf{H}$ is simply the matrix representation of the kernel operator $\mathcal{H}$.

As an immediate consequence noted by Zhang et al. (2018a), both the second-order necessary condition (7) and the second-order sufficient condition (8) for local optimality are *linear matrix inequalities* (LMIs) with respect to $\mathbf{H}$. In particular, this means that finding an instance of (2) with a fixed $M^\star$ as the ground truth and X as a spurious local minimum is a *convex* optimization problem:

$$\begin{aligned} &\text{find } \mathcal{A} \\ &\text{such that } f(X) = \frac{1}{2}\|\mathcal{A}(XX^T - M^\star)\|^2, \\ &\qquad \nabla f(X) = 0, \\ &\qquad \nabla^2 f(X) \succeq \mu I. \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} &\text{find } \mathbf{H} \succeq 0 \\ &\text{such that } \mathbf{X}^T\mathbf{H}\mathbf{e} = 0, \\ &\qquad 2 \cdot [I_r \otimes \mathrm{mat}(\mathbf{H}\mathbf{e})] \\ &\qquad\quad + \mathbf{X}^T\mathbf{H}\mathbf{X} \succeq \mu I. \end{aligned} \tag{12}$$

Given a feasible point $\mathbf{H}$, we compute an $\mathbf{A}$ satisfying $\mathbf{H} = \mathbf{A}^T\mathbf{A}$ using Cholesky factorization or an eigendecomposition. Then, matricizing each row of $\mathbf{A}$ recovers the matrices $A_1, \ldots, A_m$ implementing a feasible choice of $\mathcal{A}$.

5. Main idea: The inexistence of counterexamples

At the heart of this work is a simple argument by the inexistence of counterexamples. To illustrate the idea, consider making the following claim for a fixed choice of $\lambda \in [0, 1)$ and $X, Z \in \mathbb{R}^{n\times r}$:

$$\begin{aligned} &\text{If } \mathcal{A} \text{ satisfies } \lambda\text{-RIP, then } X \text{ is } \textit{not} \text{ a spurious second-order} \\ &\quad \text{critical point for the nonconvex recovery of } M^\star = ZZ^T. \end{aligned} \tag{13}$$

The claim is refuted by a counterexample: an instance of (2) satisfying λ-RIP with ground truth $M^\star = ZZ^T$ and spurious local minimum X. The problem of finding a counterexample is a nonconvex feasibility problem:

$$\begin{aligned} \text{find} \quad & \mathcal{A} \\ \text{such that} \quad & f(X) = \frac{1}{2}\|\mathcal{A}(XX^T - ZZ^T)\|^2 \\ & \nabla f(X) = 0, \quad \nabla^2 f(X) \succeq 0 \\ & \mathcal{A} \text{ satisfies } \delta\text{-RIP}. \end{aligned} \tag{14}$$

If problem (14) is *feasible* for $\delta = \lambda$, then any feasible point is a counterexample that refutes the claim (13). However, if problem (14) is *infeasible* for $\delta = \lambda$, then counterexamples do not exist, so we must accept the claim (13) at face value. In other words, *the inexistence of counterexamples is proof for the original claim.*

The same argument can be posed in an optimization form. Instead of finding any arbitrary counterexample, we will look for the counterexample with the *smallest* RIP constant

$$\begin{aligned} \delta(X, Z) \quad \equiv \quad & \underset{\mathcal{A}}{\text{minimum}} \quad \delta \\ & \text{subject to} \quad f(X) = \frac{1}{2}\|\mathcal{A}(XX^T - ZZ^T)\|^2 \\ & \qquad \nabla f(X) = 0, \quad \nabla^2 f(X) \succeq 0 \\ & \qquad \mathcal{A} \text{ satisfies } \delta\text{-RIP}. \end{aligned} \tag{15}$$

Suppose that problem (15) attains its minimum at $\mathcal{A}^\star$. If $\lambda \geq \delta(X, Z)$, then the minimizer $\mathcal{A}^\star$ is a counterexample that refutes the claim (13). On the other hand, if $\lambda < \delta(X, Z)$, then problem (14) is infeasible for $\delta = \lambda$, so counterexamples do not exist, so the claim (13) must be true.

Repeating these arguments over all choices of X and Z yields the following global recovery guarantee.

Lemma 6 (Sharp global guarantee) *Suppose that problem (15) attains its minimum of $\delta(X, Z)$. Define $\delta^\star$ as in*

$$\delta^\star \quad \equiv \quad \underset{X,Z\in\mathbb{R}^{n\times r}}{\textit{infimum}} \quad \delta(X, Z) \quad \textit{subject to} \quad XX^T \neq ZZ^T. \tag{16}$$

If $\mathcal{A}$ satisfies λ-RIP with $\lambda < \delta^\star$, then $f(X) = \|\mathcal{A}(XX^T - M^\star)\|^2$ with ground truth $M^\star \succeq 0$ and $\mathrm{rank}\,(M^\star) \leq r$ satisfies:

$$\nabla f(X) = 0, \quad \nabla^2 f(X) \succeq 0 \quad \Longleftrightarrow \quad XX^T = M^\star. \tag{17}$$

Moreover, if there exist $X^\star, Z^\star$ such that $\delta^\star = \delta(X^\star, Z^\star)$, then the threshold $\delta^\star$ is sharp.

Proof To prove (17), we simply prove the claim (13) for $\lambda < \delta^\star$ and every possible choice of $X, Z \in \mathbb{R}^{n\times r}$. Indeed, if $XX^T = ZZ^T$, then X is not a spurious point (as it is a global minimum), whereas if $XX^T \neq ZZ^T$, then $\lambda < \delta^\star \leq \delta(X, Z)$ proves the inexistence of a counterexample. Sharpness follows because the minimum $\delta^\star = \delta(X^\star, Z^\star)$ is attained by the minimizer $\mathcal{A}^\star$ that refutes the claim (13) for all $\lambda \geq \delta^\star$ and $X = X^\star$ and $Z = Z^\star$. ■

Repeating the same arguments over an ϵ-local neighborhood of the ground truth yields the following *local* recovery guarantee.

Lemma 7 (Sharp local guarantee) *Suppose that problem (15) attains its minimum of $\delta(X, Z)$. Given $\epsilon > 0$, define $\delta^\star(\epsilon)$ as in*

$$\delta^\star(\epsilon) \quad \equiv \quad \underset{X,Z\in\mathbb{R}^{n\times r}}{\textit{infimum}} \quad \delta(X, Z) \quad \textit{subject to} \quad XX^T \neq ZZ^T, \ \|XX^T - ZZ^T\|_F \leq \epsilon\|ZZ^T\|_F. \tag{18}$$

If $\mathcal{A}$ satisfies λ-RIP with $\lambda < \delta^\star(\epsilon)$, then $f(X) = \|\mathcal{A}(XX^T - M^\star)\|^2$ with ground truth $M^\star \succeq 0$ and $\mathrm{rank}\,(M^\star) \leq r$ satisfies:

$$\nabla f(X) = 0, \quad \nabla^2 f(X) \succeq 0, \quad \|XX^T - ZZ^T\|_F \leq \epsilon\|ZZ^T\|_F \Longleftrightarrow \quad XX^T = M^\star. \tag{19}$$

Moreover, if there exist $X^\star, Z^\star$ such that $\delta^\star = \delta(X^\star, Z^\star)$, then the threshold $\delta^\star$ is sharp.

Our main difficulty with Lemma 6 and Lemma 7 is the evaluation of $\delta(X,Z)$. Indeed, verifying δ-RIP for a fixed $\mathcal{A}$ is already NP-hard in general (Tillmann and Pfetsch, 2014), so it is reasonable to expect that solving an optimization problem (15) with a δ-RIP constraint would be at least NP-hard. Instead, Zhang et al. (2018a) suggests replacing the δ-RIP constraint with a convex *sufficient* condition, obtained by enforcing the RIP inequality (3) over *all* $n \times n$ matrices (and not just rank-$2r$ matrices):

$$(1-\delta)\|M\|_F^2 \le \|\mathcal{A}(M)\|^2 \le (1+\delta)\|M\|_F^2 \qquad \forall M \in \mathbb{R}^{n\times n}. \tag{20}$$

The resulting problem is a linear matrix inequality (LMI) optimization over the kernel operator $\mathcal{H} = \mathcal{A}^T\mathcal{A}$ that yields an upper-bound on $\delta(X,Z)$:

$$\begin{aligned} \mathrm{LMI}(X,Z) \quad \equiv \quad & \underset{\mathcal{H}=\mathcal{A}^T\mathcal{A}}{\text{minimum}} \quad \delta \\ & \text{subject to} \quad f(X) = \frac{1}{2}\|\mathcal{A}(XX^T - ZZ^T)\|^2 \\ & \qquad \nabla f(X) = 0, \quad \nabla^2 f(X) \succeq 0 \\ & \qquad (1-\delta)I \preceq \mathcal{A}^T\mathcal{A} \preceq (1+\delta)I \end{aligned} \tag{21}$$

Surprisingly, the upper-bound is *tight*—problem (21) is actually an *exact reformulation* of problem (15).

Theorem 8 (Exact convex reformulation) *Given $X, Z \in \mathbb{R}^{n\times r}$, we have $\delta(X,Z) = \mathrm{LMI}(X,Z)$ with both problems attaining their minima. Moreover, every minimizer $\mathcal{H}^\star$ for the latter problem is related to a minimizer $\mathcal{A}^\star$ for the former problem via $\mathcal{H}^\star = (\mathcal{A}^\star)^T\mathcal{A}^\star$.*

Theorem 8 is the key insight that allows us to establish our main results. When rank $r = 1$, the LMI is sufficiently simple that it can be suitably relaxed and solved in closed-form, as we will soon show in Section 7. But even when $r > 1$, the LMI can still be solved numerically using an interior-point method. This allows us to perform numerical experiments to probe at the true value of $\delta^\star$ and $\delta^\star(\epsilon)$, even when analytical arguments are not available.

Section 5.1 below gives a proof of Theorem 8. A key step of the proof is to establish the following equivalence:

$$\mathrm{LMI}(X,Z) = \mathrm{LMI}(P^TX, P^TZ) \text{ where } P = \mathrm{orth}([X,Z]). \tag{22}$$

For small values of the rank $r \ll n$, equation (22) also yields an efficient algorithm for evaluating $\mathrm{LMI}(X,Z)$ in *linear* time: compute P, P^TX, and P^TZ, and then evaluate $\mathrm{LMI}(P^TX, P^TZ)$. Moreover, the associated minimizer $\mathcal{A}^\star$ can also be efficiently recovered. These practical aspects are discussed in detail in Section 5.2.

5.1. Proof of Theorem 8

Given $X, Z \in \mathbb{R}^{n\times r}$, we define $e \in \mathbb{R}^{n^2}$ and $\mathbf{X} \in \mathbb{R}^{n^2\times nr}$ to satisfy equation (11) with respect to X and $M^\star = ZZ^T$. Then, problem (21) can be explicitly written as

$$\begin{aligned} \mathrm{LMI}(X,Z) \quad = \quad & \underset{\delta,\mathbf{H}}{\text{minimum}} \quad \delta \\ & \text{subject to} \quad \mathbf{X}^T\mathbf{H}\mathbf{e} = 0, \\ & \qquad 2\cdot[I_r \otimes \mathrm{mat}(\mathbf{H}\mathbf{e})] + \mathbf{X}^T\mathbf{H}\mathbf{X} \succeq 0, \\ & \qquad (1-\delta)I \preceq \mathbf{H} \preceq (1+\delta)I, \end{aligned} \tag{23}$$

with Lagrangian dual

$$\begin{aligned} \underset{y,U_1,U_2,V}{\text{maximize}} \quad & \mathrm{tr}(U_1 - U_2) \\ \text{subject to} \quad & \mathrm{tr}(U_1 + U_2) = 1, \\ & \sum_{j=1}^{r} (\mathbf{X}y - \mathrm{vec}\,(V_{j,j}))\mathbf{e}^T + \mathbf{e}(\mathbf{X}y - \mathrm{vec}\,(V_{j,j}))^T \\ & \qquad - \mathbf{X}V\mathbf{X}^T = U_1 - U_2, \\ & V = \begin{bmatrix} V_{1,1} & \cdots & V_{r,1} \\ \vdots & \ddots & \vdots \\ V_{r,1}^T & \cdots & V_{r,r} \end{bmatrix} \succeq 0, \quad U_1 \succeq 0, \quad U_2 \succeq 0. \end{aligned} \tag{24}$$

The dual problem admits a strictly feasible point (for sufficiently small $\epsilon > 0$, set $y = 0$, $V = \epsilon I$, $U_1 = \eta I - \epsilon W$, and $U_2 = \eta \cdot I + \epsilon W$ where $2\eta = n^{-2}$ and $2W = r[\mathrm{vec}\,(I)\mathbf{e}^T + \mathbf{e}\mathrm{vec}\,(I)^T] - \mathbf{X}\mathbf{X}^T$) and the primal problem is bounded (the constraints imply $\delta \ge 0$). Hence, Slater's condition is satisfied, strong duality holds, and the primal problem attains its optimal value at a minimizer.

It turns out that both the minimizer and the minimum are invariant under an orthogonal projection.

Lemma 9 (Orthogonal projection) *Given $X, Z \in \mathbb{R}^{n\times r}$, let $P \in \mathbb{R}^{n\times q}$ with $q \le n$ satisfy*

$$P^TP = I_q, \qquad PP^TX = X, \qquad PP^TZ = Z.$$

Let $(\hat{\delta}, \hat{\mathbf{H}})$ be a minimizer for $\mathrm{LMI}(P^TX, P^TZ)$. Then, $(\delta, \mathbf{H})$ is a minimizer for $\mathrm{LMI}(X,Z)$, where $\mathbf{P} = P \otimes P$ and

$$\delta = \hat{\delta}, \qquad \mathbf{H} = \mathbf{P}\hat{\mathbf{H}}\mathbf{P}^T + (I - \mathbf{P}\mathbf{P}^T).$$

Proof Choose arbitrarily small $\epsilon > 0$. Strong duality guarantees the existence of a dual feasible point $(\hat{y}, \hat{U}_1, \hat{U}_2, \hat{V})$ with duality gap ϵ. This is a certificate that proves $(\hat{\delta}, \hat{\mathbf{H}})$ to be ϵ-suboptimal for $\mathrm{LMI}(P^TX, P^TZ)$. We can mechanically verify that $(\delta, \mathbf{H})$ is primal feasible and that (y, U_1, U_2, V) is dual feasible, where

$$y = (I_r \otimes P)\hat{y}, \qquad U_1 = \mathbf{P}\hat{U}_1\mathbf{P}^T, \qquad U_2 = \mathbf{P}\hat{U}_2\mathbf{P}^T, \qquad V = (I_r \otimes P)\hat{V}(I_r \otimes P)^T.$$

Then, (y, U_1, U_2, V) is a certificate that proves $(\delta, \mathbf{H})$ to be ϵ-suboptimal for $\mathrm{LMI}(X,Z)$, since

$$\delta - \mathrm{tr}(U_1 - U_2) = \hat{\delta} - \mathrm{tr}(\hat{U}_1 - \hat{U}_2) = \epsilon.$$

Given that ϵ-suboptimal certificates exist for all $\epsilon > 0$, the point $(\delta, \mathbf{H})$ must actually be optimal. The details for verifying primal and dual feasibility are straightforward but tedious; they are included in Appendix A for completeness. ■

Recall that we developed an upper-bound $\mathrm{LMI}(X,Z)$ on $\delta(X,Z)$ by replacing δ-RIP with a convex *sufficient* condition (20). The same idea can also be used to produce a lower-bound. Specifically, we replace the δ-RIP constraint with a convex *necessary* condition, obtained

by enforcing the RIP inequality (3) over *a subset of* rank-$2r$ matrices (instead of over all rank-$2r$ matrices):

$$(1-\delta)\|PYP^T\|_F^2 \le \|\mathcal{A}(PYP^T)\|^2 \le (1+\delta)\|PYP^T\|_F^2 \qquad \forall Y \in \mathbb{R}^{d\times d} \tag{25}$$

where P is a *fixed* $n \times d$ matrix with $d \le 2r$. The resulting problem is also convex (we write $\mathbf{P} = P \otimes P$)

$$\begin{aligned} \delta(X,Z) \quad \ge \quad & \text{minimize} && \delta \\ & \text{subject to} && \mathbf{X}^T\mathbf{H}\mathbf{e} = 0, \\ & && 2\cdot[I_r \otimes \mathrm{mat}(\mathbf{H}\mathbf{e})] + \mathbf{X}^T\mathbf{H}\mathbf{X} \succeq 0, \\ & && (1-\delta)\mathbf{P}^T\mathbf{P} \preceq \mathbf{P}^T\mathbf{H}\mathbf{P} \preceq (1+\delta)\mathbf{P}^T\mathbf{P} \end{aligned} \tag{26}$$

with Lagrangian dual

$$\begin{aligned} \underset{y,U_1,U_2,V}{\text{maximize}} \quad & \mathrm{tr}[\mathbf{P}(U_1-U_2)\mathbf{P}^T] \\ \text{subject to} \quad & \mathrm{tr}[\mathbf{P}(U_1+U_2)\mathbf{P}^T] = 1, \\ & \sum_{j=1}^{r}(\mathbf{X}y - \mathrm{vec}\,(V_{j,j}))\mathbf{e}^T + \mathbf{e}(\mathbf{X}y - \mathrm{vec}\,(V_{j,j}))^T \\ & \qquad - \mathbf{X}V\mathbf{X}^T = \mathbf{P}(U_1-U_2)\mathbf{P}^T, \\ & V = \begin{bmatrix} V_{1,1} & \cdots & V_{r,1} \\ \vdots & \ddots & \vdots \\ V_{r,1}^T & \cdots & V_{r,r} \end{bmatrix} \succeq 0, \quad U_1 \succeq 0, \quad U_2 \succeq 0. \end{aligned} \tag{27}$$

It turns out that for the specific choice of $P = \mathrm{orth}([X,Z])$, the lower-bound in (26) *coincides* with the upper-bound in (23).

Lemma 10 (Tightness) *Define* $P = \mathrm{orth}([X,Z])$. *Let* $(\hat{\delta}, \hat{\mathbf{H}})$ *be a minimizer for* $\mathrm{LMI}(P^TX, P^TZ)$. *Then,* $(\delta, \mathbf{H})$ *is a minimizer for problem (26), where* $\mathbf{P} = P \otimes P$ *and*

$$\delta = \hat{\delta}, \qquad\qquad \mathbf{H} = \mathbf{P}\hat{\mathbf{H}}\mathbf{P}^T.$$

Proof The proof is almost identical to that of Lemma 9. Again, choose arbitrarily small $\epsilon > 0$. Let $(\hat{y}, \hat{U}_1, \hat{U}_2, \hat{V})$ be a dual feasible point for $\mathrm{LMI}(P^TX, P^TZ)$ with duality gap ϵ. Then, (y, U_1, U_2, V) where

$$y = (I_r \otimes P)\hat{y}, \qquad U_1 = \hat{U}_1, \qquad U_2 = \hat{U}_2, \qquad V = (I_r \otimes P)\hat{V}(I_r \otimes P)^T$$

is a certificate that proves $(\delta, \mathbf{H})$ to be ϵ-suboptimal for problem (26). The details for verifying primal and dual feasibility are included in Appendix B. ■

Putting the upper- and lower-bounds together then yields a short proof of Theorem 8.

Proof [Proof of Theorem 8] Denote $\delta_{\mathrm{ub}} = \mathrm{LMI}(X,Z)$ as the optimal value to the upper-bound problem (23) and $\mathcal{H}^\star$ as the corresponding minimizer. (The minimizer $\mathcal{H}^\star$ always exists due to the boundedness of the primal problem and the existence of a strictly feasible

Algorithm 1 Efficient algorithm for $\delta(X,Z)$ and $\mathcal{A}^\star$.

Input. Choices of $X, Z \in \mathbb{R}^{n\times r}$.
Output. The value $\hat{\delta} = \delta(X,Z)$ and the corresponding minimizer $\mathcal{A}^\star$ (if desired).
Algorithm.

1. Compute $P = \mathrm{orth}([X,Z]) \in \mathbb{R}^{n\times d}$ and project $\hat{X} = P^TX$ and $\hat{Z} = P^TZ$.
2. Solve $\hat{\delta} = \mathrm{LMI}(\hat{X}, \hat{Z})$ using an interior-point method to obtain minimizer $\hat{\mathbf{H}}$. **Output** $\hat{\delta}$.
3. Compute the orthogonal complement $P_\perp = \mathrm{orth}(I - PP^T) \in \mathbb{R}^{n\times(n-d)}$.
4. Factor $\hat{\mathbf{H}} = \hat{\mathbf{A}}^T\hat{\mathbf{A}}$ using (dense) Cholesky factorization.
5. Analytically factor $(\mathbf{A}^\star)^T\mathbf{A}^\star = \mathbf{H}^\star = \mathbf{P}\hat{\mathbf{H}}\mathbf{P}^T + (I - \mathbf{P}\mathbf{P}^T)$ using the formula
$$(\mathbf{A}^\star)^T = \begin{bmatrix}(P\otimes P)\hat{\mathbf{A}}^T & P\otimes P_\perp & P_\perp \otimes P & P_\perp \otimes P_\perp\end{bmatrix}$$
while using the Kronecker identity $(P\otimes P)\mathrm{vec}\,(U) = \mathrm{vec}\,(PUP^T)$ to evaluate each column of $(P\otimes P)\hat{\mathbf{A}}^T$.
6. Recover the matrices $A_1^\star, \ldots, A_m^\star$ associated with the minimizer $\mathcal{A}^\star$ by matricizing each row of $\mathbf{A}^\star$. **Output** $\mathcal{A}^\star$.

point in the dual problem.) Denote δ_{lb} as the optimal value to the lower-bound problem (26). For $P = \mathrm{orth}([X,Z])$, the sequence of inclusions

$$\{PYP^T : Y \in \mathbb{R}^{d\times d}\} \subseteq \{M \in \mathbb{R}^{n\times n} : \mathrm{rank}\,(M) \le 2r\} \subseteq \mathbb{R}^{n\times n},$$

implies $\delta_{\mathrm{lb}} \le \delta(X,Z) \le \delta_{\mathrm{ub}}$. However, by Lemma 9 and Lemma 10, we actually have $\delta_{\mathrm{ub}} = \delta_{\mathrm{lb}} = \mathrm{LMI}(P^TX, P^TZ)$, and hence $\delta_{\mathrm{lb}} = \delta(X,Z) = \delta_{\mathrm{ub}}$. Finally, the minimizer $\mathcal{H}^\star$ factors into $(\mathcal{A}^\star)^T\mathcal{A}^\star$, where $\mathcal{A}^\star$ satisfies the sufficient condition (20), and hence also δ-RIP. ■

5.2. Efficient evaluation of $\delta(X,Z)$ and $\mathcal{A}^\star$

We now turn to the practical problem of evaluating $\delta(X,Z)$ and the associated minimizer $\mathcal{A}^\star$ using a numerical algorithm. While its exact reformulation $\mathrm{LMI}(X,Z) = \delta(X,Z)$ is indeed convex, naï¿œvely solving it using an interior-point solution can require up to $O(n^{13})$ time and $O(n^8)$ memory (as it requires solving an order-n^2 semidefinite program). In our experiments, the largest instances of (21) that we could accommodate using the state-of-the-art solver MOSEK (Andersen and Andersen, 2000) had dimensions no greater than $n \le 12$.

Instead, we can efficiently evaluate $\delta(X,Z)$ using Algorithm 1. When the rank $r \ll n$ is small, the algorithm evaluates $\delta(X,Z)$ in linear $O(n)$ time and memory, and if desired, also recovers the minimizer $\mathcal{A}^\star$ in $O(n^4)$ time and memory. In practice, our numerical experiments were able to accommodate for rank as large as $r \le 10$.

Proposition 11 *Algorithm 1 correctly outputs the minimum value $\hat{\delta} = \delta(X, Z)$ and the minimizer $\mathcal{A}^\star$. Moreover, Steps 1-2 for $\hat{\delta}$ use*

$$O(nr^2 + r^{13}\log(1/\epsilon)) \text{ time and } O(nr + r^8) \text{ memory,} \tag{28}$$

while Steps 3-6 for $\mathcal{A}^\star$ use

$$O(n^4 + n^2r^3 + nr^4 + r^6) \text{ time and } O(n^4) \text{ memory.} \tag{29}$$

Proof We begin by verifying correctness. The fact that the minimum value $\delta(X, Z) = \mathrm{LMI}(P^TX, P^TZ)$ follows from Theorem 8 and Lemma 9. To prove correctness for the minimizer $\mathcal{A}^\star$, we recall that Algorithm 1 defines $P_\perp \in \mathbb{R}^{n\times(n-d)}$ as the orthogonal complement of $P \in \mathbb{R}^{n\times d}$, and note that

$$\begin{aligned}
&PP^T \otimes P_\perp P_\perp^T + P_\perp P_\perp^T \otimes PP^T + P_\perp P_\perp^T \otimes P_\perp P_\perp^T \\
=&(PP^T + P_\perp P_\perp^T) \otimes (PP^T + P_\perp P_\perp^T) - PP^T \otimes PP^T \\
=&I - \mathbf{P}\mathbf{P}^T,
\end{aligned}$$

where $\mathbf{P} = P \otimes P$. Hence, Algorithm 1 produces the minimizer $\mathbf{H}^\star = \mathbf{P}\hat{\mathbf{H}}\mathbf{P}^T + (I - \mathbf{P}\mathbf{P}^T)$ for $\mathrm{LMI}(X, Z)$ in Lemma 9 as desired.

Now, let us quantify complexity. Note that $d \le 2r = O(r)$ by construction. Step 1 takes $O(nr^2)$ time and $O(nr)$ memory. Step 2 requires solving an order $\theta = O(r^2)$ semidefinite program in $O(\theta^{6.5}\log(1/\epsilon)) = O(r^{13}\log(1/\epsilon))$ time and $O(\theta^4) = O(r^8)$ memory. Stopping here yields (28). Step 3 uses $O(n^3 + n^2r)$ time and $O(n^2)$ memory. Step 4 uses $O(r^6)$ time and $O(r^4)$ memory. Step 5 performs $O(r^2)$ matrix-vector products each costing $O(nr^2+n^2r)$ time and $O(n^4)$ memory, and then filling the rest of $\mathbf{A}$ in $O(n^4)$ time and memory. Step 6 costs $O(n^4)$ time and memory. Summing the terms and substituting $O(n^4 + r^4) = O(n^4)$ in the memory complexity yields the desired figures. ■

6. Counterexample with $\delta = 1/2$ for the rank-1 problem

In this section, we use a *family* of counterexamples to prove that δ-RIP with $\delta < 1/2$ is necessary for the exact recovery of *any* arbitrary rank-1 ground truth $M^\star = zz^T$ (and not just the 2×2 ground truth studied by Zhang et al. (2018a)). Specifically, we state a choice of $\mathcal{A}^\star$ that satisfies 1/2-RIP but whose $f^\star(x) = \|\mathcal{A}^\star(xx^T - M^\star)\|^2$ admits a spurious second-order point.

Example 1 *Given rank-1 ground truth $M^\star = zz^T \neq 0$, define a set of orthonormal vectors $u_1, u_2, \ldots, u_n \in \mathbb{R}^n$ with $u_1 = z/\|z\|$, and define $m = n^2$ measurement matrices $A_1, A_2, \ldots, A_m$, with*

$$A_1 = u_1u_1^T + \frac{1}{2}u_2u_2^T, \qquad A_2 = \frac{\sqrt{3}}{2}(u_1u_2^T + u_2u_1^T),$$
$$A_{n+1} = \frac{1}{\sqrt{2}}(u_1u_2^T - u_2u_1^T), \qquad A_{n+2} = \frac{\sqrt{3}}{2}u_2u_2^T,$$

and the remaining $n^2 - 4$ measurement matrices sequentially assigned as

$$A_k = u_iu_j^T, \quad k = i + n\cdot(j-1), \qquad \forall (i,j) \in \{1,2,\ldots,n\}^2\backslash\{1,2\}^2.$$

Then, the associated operator $\mathcal{A}^\star$ satisfies 1/2-RIP:

$$\left(1 - \frac{1}{2}\right)\|M\|_F^2 \le \|\mathcal{A}^\star(M)\|^2 \le \left(1 + \frac{1}{2}\right)\|M\|_F^2 \qquad \forall M \in \mathbb{R}^{n\times n},$$

but the corresponding $f^\star(x) \equiv \|\mathcal{A}^\star(xx^T - M^\star)\|^2$ admits $x = (\|z\|/\sqrt{2})\,u_2$ as a spurious second-order critical point:

$$f^\star(x) = \frac{3}{4}\|M^\star\|_F^2, \qquad \nabla f^\star(x) = 0, \qquad \nabla^2 f^\star(x) \succeq 8xx^T.$$

We derived Example 1 by numerically solving $\delta(x, z)$ with any x satisfying $x^Tz = 0$ and $\|x\| = \|z\|/\sqrt{2}$ using Algorithm 1. The 1/2-RIP counterexample of Zhang et al. (2018a) arises as the instance of Example 1 associated with the 2×2 ground truth $\hat{z}\hat{z}^T$ and $\hat{z} = (1, 0)$:

$$\hat{A}_1 = \begin{bmatrix}1 & 0\\ 0 & 1/2\end{bmatrix}, \quad \hat{A}_2 = \begin{bmatrix}0 & \sqrt{3}/2\\ \sqrt{3}/2 & 0\end{bmatrix}, \quad \hat{A}_3 = \begin{bmatrix}0 & -1/\sqrt{2}\\ 1/\sqrt{2} & 0\end{bmatrix}, \quad \hat{A}_4 = \begin{bmatrix}0 & 0\\ 0 & \sqrt{3}/2\end{bmatrix}.$$

The associated operator $\hat{\mathcal{A}} : \mathbb{S}^2 \to \mathbb{R}^4$ is invertible and satisfies 1/2-RIP, but $\hat{x} = (0, 1/\sqrt{2})$ is a spurious second-order point:

$$\hat{f}(\hat{x}) \equiv \|\hat{\mathcal{A}}(\hat{x}\hat{x}^T - \hat{z}\hat{z}^T)\|^2 = \frac{3}{4}, \qquad \nabla\hat{f}(\hat{x}) = 0, \qquad \nabla^2\hat{f}(\hat{x}) = \begin{bmatrix}0 & 0\\ 0 & 4\end{bmatrix}.$$

We can verify the correctness of Example 1 for a general rank-1 ground truth by reducing it down to this specific 2×2 example.

Proof [Proof of correctness for Example 1] We can mechanically verify Example 1 to be correct with ground truth $\hat{z}\hat{z}^T$ and $\hat{z} = (1, 0)$. Denote $\hat{\mathcal{A}}$, $\hat{f}(\hat{x}) = \|\hat{\mathcal{A}}(\hat{x}\hat{x}^T - \hat{z}\hat{z}^T)\|^2$, and $\hat{x} = (0, 1/\sqrt{2})$ as the corresponding minimizer, nonconvex objective, and spurious second-order critical point.

For a general rank-1 ground truth $M^\star = zz^T$, recall that we have defined a set of orthonormal vectors $u_1, u_2, \ldots, u_n \in \mathbb{R}^n$ with $u_1 = z/\|z\|$. Then, setting $P = [u_1, u_2]$ and $P_\perp = [u_3, \ldots, u_n]$ shows that the matrix version of $\mathcal{A}^\star$ can be permuted row-wise to satisfy

$$(\mathbf{A}^\star)^T = \begin{bmatrix}(P\otimes P)\hat{\mathbf{A}}^T & P\otimes P_\perp & P_\perp\otimes P & P_\perp\otimes P_\perp\end{bmatrix}$$

where $\hat{\mathbf{A}}$ is the matrix version of $\hat{\mathcal{A}}$. Repeating the proof of Proposition 11 shows that

$$(\mathbf{A}^\star)^T\mathbf{A}^\star = \mathbf{P}\hat{\mathbf{A}}^T\hat{\mathbf{A}}\mathbf{P} + (I - \mathbf{P}\mathbf{P}^T)$$

where $\mathbf{P} = P \otimes P$, and so $\mathcal{A}^\star$ also satisfies 1/2-RIP. Moreover, this implies that

$$f^\star(x) \equiv \|\mathcal{A}^\star(xx^T - zz^T)\|^2 = \|z\|^4\hat{f}(P^Tx/\|z\|) + (\|x\|^4 - \|P^Tx\|^4).$$

Differentiating yields the following at $x = (\|z\|/\sqrt{2})u_2$:

$$\begin{aligned} f^\star(x) &= \|z\|^4 \hat{f}(\hat{x}) &&= (3/4)\|z\|^4, \\ \nabla f^\star(x) &= \|z\|^3 P \nabla \hat{f}(\hat{x}) &&= 0, \\ \nabla^2 f^\star(x) &= \|z\|^2 P \nabla^2 \hat{f}(\hat{x}) P^T + 2\|x\|^2 (I - PP^T) &&\succeq 4\|z\|^2 u_2 u_2^T. \end{aligned}$$

■

7. Closed-form lower-bound for the rank-1 problem

It turns out that the LMI problem (21) in the rank-1 case is sufficiently simple to be suitably relaxed and then solved in closed-form. Our main result in this section is the following lower-bound on $\delta(x, z) = \mathrm{LMI}(x, z)$.

Theorem 12 (Closed-form lower-bound) *Let $x, z \in \mathbb{R}^n$ be arbitrary nonzero vectors, and define their length ratio ρ and incidence angle ϕ:*

$$\rho \equiv \frac{\|x\|}{\|z\|}, \qquad \phi \equiv \arccos\left(\frac{x^T z}{\|x\|\|z\|}\right). \tag{30}$$

Define the following two scalars with respect to ρ and ϕ:

$$\alpha = \frac{\sin^2\phi}{\sqrt{(\rho^2-1)^2 + 2\rho^2\sin^2\phi}}, \qquad \beta = \frac{\rho^2}{\sqrt{(\rho^2-1)^2 + 2\rho^2\sin^2\phi}}.$$

Then, we have $\delta(x, z) \geq \delta_{\mathrm{lb}}(x, z)$, where

$$\delta_{\mathrm{lb}}(x, z) \equiv \begin{cases} \sqrt{1-\alpha^2} & \text{if } \beta \geq \dfrac{\alpha}{1+\sqrt{1-\alpha^2}}, \quad (31) \\ \dfrac{1-2\alpha\beta+\beta^2}{1-\beta^2} & \text{if } \beta \leq \dfrac{\alpha}{1+\sqrt{1-\alpha^2}}. \quad (32) \end{cases}$$

The rank-1 global and local recovery guarantees follow quickly from this theorem, as shown below.

Proof [Proof of Theorem 3] The existence of Example 1 already proves that

$$\delta^\star = \min_{x,z\in\mathbb{R}^n} \delta(x, z) \leq 1/2. \tag{33}$$

Below, we will show that $\delta_{\mathrm{lb}}(x, z)$ attains its minimum of $1/2$ at any x satisfying $x^T z = 0$ and $\|x\|/\|z\| = 1/\sqrt{2}$, as in

$$1/2 = \min_{x,z\in\mathbb{R}^n} \delta_{\mathrm{lb}}(x, z) \leq \delta^\star. \tag{34}$$

Substituting $\delta^\star = 1/2$ into Lemma 6 then completes the proof of our global recovery guarantee in Theorem 3.

To prove (34), we begin by optimizing $\delta_{\mathrm{lb}}(x, z)$ over the region $\beta \geq \alpha/(1+\sqrt{1-\alpha^2})$ using equation (31), and find that the minimum value is attained along the boundary

$$\beta = \frac{\alpha}{1+\sqrt{1-\alpha^2}} = \frac{1-\sqrt{1-\alpha^2}}{\alpha}.$$

Note that the two equations (31) and (32) coincide at this boundary:

$$\left(\frac{1-2\alpha\beta+\beta^2}{1-\beta^2}\right)\left(\frac{\alpha/\beta}{\alpha/\beta}\right) = \frac{(1+\sqrt{1-\alpha^2}) - 2\alpha^2 + (1-\sqrt{1-\alpha^2})}{(1+\sqrt{1-\alpha^2}) - (1-\sqrt{1-\alpha^2})} = \sqrt{1-\alpha^2}.$$

Now, we optimize $\delta_{\mathrm{lb}}(x, z)$ over the region $\beta \leq \alpha/(1+\sqrt{1-\alpha^2})$ using equation (32). First, substituting the definitions of α and β yields

$$\delta_{\mathrm{lb}}(x, z) = \frac{1-2\alpha\beta+\beta^2}{1-\beta^2} = \frac{(\rho^4+1-2\rho^2\cos^2\phi) - 2\rho^2\sin^2\phi + \rho^4}{(\rho^4+1-2\rho^2\cos^2\phi) - \rho^4} = \frac{(\rho^2-1)^2+\rho^4}{1-2\rho^2\cos^2\phi}.$$

This expression is minimized at $\phi = \pm\pi/2$ and $\rho = 1/\sqrt{2}$, with a minimum value of $1/2$. The corresponding point $\alpha = 2/\sqrt{5}$ and $\beta = \alpha/4$ lies in the strict interior $\beta < \alpha/(1+\sqrt{1-\alpha^2})$. This point must be the global minimum, because it dominates the boundary $\beta = \alpha/(1+\sqrt{1-\alpha^2})$, which in turn dominates the other region $\beta > \alpha/(1+\sqrt{1-\alpha^2})$. ■

Proof [Proof of Theorem 4] We will optimize over an ϵ-neighborhood of the ground truth and show that

$$\left(1 - \frac{\epsilon^2}{2(1-\epsilon)}\right)^{1/2} \leq \min_{x,z\in\mathbb{R}^n} \{\delta_{\mathrm{lb}}(x, z) : \|xx^T - zz^T\|_F \leq \epsilon\|zz^T\|_F\} \leq \delta^\star(\epsilon). \tag{35}$$

Substituting this lower-bound on $\delta^\star(\epsilon)$ into Lemma 7 then completes the proof of our local recovery guarantee in Theorem 4.

To obtain (35), we first note that the ϵ-neighborhood constraint implies the following

$$\|xx^T - zz^T\|_F \leq \epsilon\|zz^T\|_F \qquad \Longleftrightarrow \qquad (\rho^2-1)^2 + 2\rho^2\sin^2\phi \leq \epsilon^2.$$

This in turn implies $\epsilon^2 \geq (\rho^2-1)^2$ and $\epsilon^2 \geq [(\rho^2-1)^2 + 2\rho^2]\sin^2\phi$, and hence

$$1-\epsilon \leq \rho^2 \leq 1+\epsilon, \qquad \sin^2\phi \leq \epsilon^2.$$

We wish to derive a threshold $\hat{\epsilon}$ such that if $\epsilon \leq \hat{\epsilon}$, then

$$\frac{\beta}{\alpha} = \frac{\rho^2}{\sin^2\phi} \geq \frac{1-\epsilon}{\epsilon^2} \geq 1 \geq \frac{1}{1+\sqrt{1-\alpha^2}},$$

and so $\delta_{\mathrm{lb}}(x, z) = \sqrt{1-\alpha^2}$ as dictated entirely by equation (31). Clearly, this requires solving the quadratic equation $(1-\hat{\epsilon}) = \hat{\epsilon}^2$ for the positive root at $\hat{\epsilon} = (-1+\sqrt{5})/2 \geq 0.618$. Now, we upper-bound α^2 to lower-bound $\sqrt{1-\alpha^2}$:

$$\begin{aligned} \alpha^2 &= \frac{\sin^4\phi}{(\rho^2-1)^2 + 2\rho^2\sin^2\phi} \leq \frac{\sin^4\phi}{[(\rho^2-1)^2+2\rho^2]\sin^2\phi} = \frac{\sin^2\phi}{\rho^4+1} \\ &\leq \frac{\epsilon^2}{(1-\epsilon)^2+1} = \frac{\epsilon^2}{2-2\epsilon+\epsilon^2} \leq \frac{\epsilon^2}{2(1-\epsilon)} \end{aligned}$$

and so

$$\delta(x,z) \geq \delta_{\text{lb}}(x,z) = \sqrt{1-\alpha^2} \geq \sqrt{1-\frac{\epsilon^2}{2(1-\epsilon)}}.$$

■

Proof [Proof of Corollary 5] Under δ-RIP, a point with a small residual must also have a small error:

$$(1-\delta)\|xx^T - M^\star\|_F^2 \leq f(x) \leq (1-\delta)\epsilon^2\|M^\star\|_F^2. \tag{36}$$

In particular, any point in the level set $f(x) \leq f(x_0)$ must also lie in the ϵ-neighborhood:

$$f(x) \leq f(x_0) < (1-\delta)\epsilon^2 f(0) \qquad \Longrightarrow \qquad \|xx^T - M^\star\|_F \leq \epsilon\|M^\star\|_F.$$

Additionally, note that

$$\epsilon^2 \leq 1-\delta^2, \quad \epsilon^2 \leq \frac{\sqrt{5}-1}{2} \qquad \Longrightarrow \qquad \delta \leq \sqrt{1-\frac{\epsilon^2}{2(1-\epsilon)}}$$

because $2(1-\epsilon) \leq 1$. The result then follows by applying Theorem 4. ■

The rest of this section is devoted to proving Theorem 12. We begin by providing a few important lemmas in Section 7.1, and then move to the proof itself in Section 7.2.

7.1. Technical lemmas

Given $M \in \mathbb{S}^n$ with eigendecomposition $M = \sum_{i=1}^m \lambda_i v_i v_i^T$, we define its projection onto the semidefinite cone as the following

$$[M]_+ \equiv \arg\min_{S\succeq 0} \|M-S\|_F^2 = \sum_{i=1}^n \max\{\lambda_i, 0\} v_i v_i^T.$$

For notational convenience, we also define a complement projection

$$[M]_- \equiv [-M]_+ = [M]_+ - M,$$

thereby allowing us to decompose every M into a positive and a negative part as in

$$M = [M]_+ - [M]_- \quad \text{where} \quad [M]_+ \succeq 0, \quad [M]_- \succeq 0.$$

Lemma 13 *Given $M \in \mathbb{S}^n$ with $\operatorname{tr}(M) \geq 0$, the following problem*

$$\underset{\alpha,U,V}{\text{minimize}} \quad \operatorname{tr}(V) \quad \text{subject to} \quad \operatorname{tr}(U) = 1, \quad \alpha M = U - V, \quad U, V \succeq 0$$

has minimizer

$$\alpha^\star = 1/\operatorname{tr}([M]_+), \qquad U^\star = \alpha^\star \cdot [M]_+, \qquad V^\star = \alpha^\star \cdot [M]_-.$$

Proof Write $p^\star$ as the optimal value. Then,

$$\begin{aligned}
p^\star =& \max_\beta \min_{\substack{\alpha\in\mathbb{R}\\ U,V\succeq 0}} \{\operatorname{tr}(V) + \beta\cdot[\operatorname{tr}(U)-1] : \alpha M = U - V\} \\
=& \max_{\beta\geq 0}\min_{\alpha\in\mathbb{R}}\{-\beta + \min_{U,V\succeq 0}\{\operatorname{tr}(V) + \beta\cdot\operatorname{tr}(U) : \alpha M = U - V\}\} \\
=& \max_{\beta\geq 0}\min_{\alpha\in\mathbb{R}}\{-\beta + \alpha\cdot[\operatorname{tr}([M]_-) + \beta\cdot\operatorname{tr}([M]_+)]\} \\
=& \max_{\beta\geq 0}\{-\beta : \operatorname{tr}([M]_-) + \beta\cdot\operatorname{tr}([M]_+) = 0\} \\
=& \operatorname{tr}([M]_-)/\operatorname{tr}([M]_+) = \operatorname{tr}(V^\star).
\end{aligned}$$

The first line converts an equality constraint into a Lagrangian. The second line isolates the optimization over $U, V \succeq 0$ with $\beta \geq 0$, noting that $\beta < 0$ would yield $\operatorname{tr}(U) \to \infty$. The third line solves the minimization over $U, V \succeq 0$ in closed-form. The fourth line views α as a Lagrange multiplier. ■

For symmetric indefinite matrices of a particular rank-2 form, the positive and negative eigenvalues can be computed in closed-form.

Lemma 14 *Given $a, b \in \mathbb{R}^n$, the matrix $M = ab^T + ba^T$ has eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ where:*

$$\lambda_i = \begin{cases} +\|a\|\|b\|(1+\cos\theta) & i = 1 \\ -\|a\|\|b\|(1-\cos\theta) & i = n \\ 0 & \textit{otherwise,} \end{cases}$$

and $\theta \equiv \arccos\left(\frac{a^Tb}{\|a\|\|b\|}\right)$ is the angle between a and b.

Proof Without loss of generality, assume that $\|a\| = \|b\| = 1$. (Otherwise, rescale $\hat{a} = a/\|a\|$, $\hat{b} = b/\|b\|$ and write $M = \|a\|\|b\| \cdot (\hat{a}\hat{b}^T + \hat{b}\hat{a}^T)$.) Decompose b into a tangent and normal component with respect to a, as in

$$b = a\underbrace{a^Tb}_{\cos\theta} + \underbrace{(I - aa^T)b}_{c\sin\theta} = a\cos\theta + c\sin\theta,$$

where c is a unit normal vector with $\|c\| = 1$ and $a^Tc = 0$. This allows us to write

$$ab^T + ba^T = \begin{bmatrix} a & c \end{bmatrix} \begin{bmatrix} 2\cos\theta & \sin\theta \\ \sin\theta & 0 \end{bmatrix} \begin{bmatrix} a & c \end{bmatrix}^T$$

and hence M is spectrally similar a 2×2 matrix with eigenvalues $\cos\theta \pm \sqrt{\cos^2\theta + \sin^2\theta}$. ■

Given $x, z \in \mathbb{R}$, recall that $\mathbf{e}$ and $\mathbf{X}$ are implicitly defined in (11) to satisfy

$$\mathbf{e} = \operatorname{vec}(xx^T - zz^T), \qquad \mathbf{X}y = \operatorname{vec}(xy^T + yx^T) \qquad \forall y \in \mathbb{R}^n.$$

Let us give a preferred orthogonal basis to study these two objects. We define $v_1 = x/\|x\|$ in the direction of x. Then, we decompose z into a tangent and normal component with respect to v_1, as in

$$z = v_1 \underbrace{v_1^T z}_{\|z\|\cos\phi} + \underbrace{(I - v_1 v_1^T) z}_{v_2 \|z\| \sin\phi} = \|z\| \cdot (v_1 \cos\phi + v_2 \sin\phi). \tag{37}$$

Here, ϕ is the incidence angle between x and z, and v_2 is the associated unit normal vector with $\|v_2\| = 1$ and $v_1^T v_2 = 0$. Using the Gram-Schmidt process, we complete v_1, v_2 with the remaining $n-2$ set of orthonormal unit vectors $v_3, v_4, \ldots, v_n$. This results in a set of right singular vectors for $\mathbf{X}$.

Lemma 15 *The matrix $\mathbf{X} \in \mathbb{R}^{n^2 \times n}$ has singular value decomposition $\mathbf{X} = \sum_{i=1}^n \sigma_i u_i v_i^T$ where v_i are defined as above, and*

$$\sigma_i = \begin{cases} 2\|x\| & i = 1 \\ \sqrt{2}\|x\| & i > 1 \end{cases}, \qquad u_i = \begin{cases} v_1 \otimes v_1 & i = 1 \\ \frac{1}{\sqrt{2}}(v_i \otimes v_1 + v_1 \otimes v_i) & i > 1 \end{cases}.$$

Proof It is easy to verify that

$$\begin{aligned} \mathbf{X}y &= y \otimes x + x \otimes y = \|x\|(y \otimes v_1 + v_1 \otimes y) \\ &= \|x\| \cdot \sum_{i=1}^n (v_i^T y)(v_i \otimes v_1 + v_1 \otimes v_i). \end{aligned}$$

Normalizing the left singular vectors then yields the designed u_i and σ_i. ■

We can also decompose $\mathbf{e}$ into a tangent and normal component with respect to range($\mathbf{X}$) as in

$$\mathbf{e} = \underbrace{\mathbf{X}\mathbf{X}^\dagger \mathbf{e}}_{\hat{e}_1 \|\mathbf{e}\| \cos\theta} + \underbrace{(I - \mathbf{X}\mathbf{X}^\dagger)\mathbf{e}}_{\hat{e}_2 \|\mathbf{e}\| \sin\theta} = \|\mathbf{e}\| \cdot (\hat{e}_1 \cos\theta + \hat{e}_2 \sin\theta) \tag{38}$$

where $\mathbf{X}^\dagger = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the usual pseudoinverse. The following Lemma gives the exact values of $\hat{e}_2$ and $\sin\theta$ (thereby also implicitly giving $\hat{e}_1$ and $\cos\theta$).

Lemma 16 *Define ϕ and v_2 as in (37), we have*

$$(I - \mathbf{X}\mathbf{X}^\dagger)\mathbf{e} = -(v_2 \otimes v_2)(\|z\| \sin\phi)^2$$

and hence $\hat{e}_2 = v_2 \otimes v_2$ and $\sin\theta = (\|z\|\sin\phi)^2/\|\mathbf{e}\|$.

Proof We solve the projection problem

$$\begin{aligned} \|(I - \mathbf{X}\mathbf{X}^\dagger)\mathbf{e}\| &= \min_y \|\mathbf{e} - \mathbf{X}y\| = \|(xx^T - zz^T) - (xy^T + yx^T)\|_F \\ &= \min_{\alpha,\beta} \left\| \begin{bmatrix} \|x\|^2 - \|z\|^2\cos^2\phi & -\|z\|^2 \sin\phi\cos\phi \\ -\|z\|^2\sin\phi\cos\phi & -\|z\|^2\sin^2\phi \end{bmatrix} - \begin{bmatrix} 2\alpha & \beta \\ \beta & 0 \end{bmatrix} \right\| \\ &= \|z\|^2 \sin^2\phi \end{aligned}$$

in which the second line makes a change of bases to v_1 and v_2. Clearly, the minimizer is in the direction of $-v_2 \otimes v_2$. ■

Using these properties of $\mathbf{X}$ and $\mathbf{e}$, we can now solve the following problem in closed-form.

Lemma 17 *Define $\alpha = (\|z\|\sin\phi)^2/\|e\| = \sin\theta$ and $\beta = \|x\|^2/\|e\|$. Then, the following optimization problem*

$$\begin{aligned} \psi(\gamma) \quad \equiv \quad & \underset{y,W}{\text{maximum}} && \mathbf{e}^T[\mathbf{X}y - \text{vec}\,(W)] \\ & \text{subject to} && \|\mathbf{e}\| \cdot \|\mathbf{X}y - \text{vec}\,(W)\| = 1 \\ & && \text{tr}(\mathbf{X}W\mathbf{X}^T) = 2\beta \cdot \gamma \\ & && W \succeq 0 \end{aligned}$$

is feasible if and only if $0 \le \gamma \le 1$ with optimal value

$$\psi(\gamma) = \gamma\alpha + \sqrt{1-\gamma^2}\sqrt{1-\alpha^2}.$$

Proof The case of $\gamma < 0$ is infeasible as it would require $\text{tr}(W) < 0$ with $W \succeq 0$. For $\gamma \ge 0$, we begin by relaxing the norm constraint into an inequality, as in $\|\mathbf{e}\| \cdot \|\mathbf{X}y - \text{vec}\,(W)\| \le 1$. Solving the resulting convex optimization over y with a fixed W yields

$$y^\star = \mathbf{X}^\dagger[\tau \cdot \mathbf{e} + \text{vec}\,(W)], \qquad \|\mathbf{e}\|\|\mathbf{X}y^\star - \text{vec}\,(W)\| = 1, \qquad \tau \ge 0. \tag{39}$$

The problem is feasible if and only if $\|\mathbf{e}\|\|(I - \mathbf{X}\mathbf{X}^\dagger)\text{vec}\,(W)\| \le 1$. Whenever feasible, the relaxation is tight, and equality is attained. The remaining problem over W reads (after some rearranging):

$$\underset{W \succeq 0}{\text{minimize}} \quad \mathbf{e}^T(I - \mathbf{X}\mathbf{X}^\dagger)\text{vec}\,(W) \quad \text{subject to} \quad \langle \mathbf{X}^T\mathbf{X}, W \rangle = 2\beta \cdot \gamma$$

and this reduces to the following using Lemma 15 and Lemma 16:

$$\underset{W \succeq 0}{\text{minimize}} \quad -(\|e\|\sin\theta)\langle v_2 v_2^T, W \rangle \quad \text{subject to} \quad 2\|x\|^2 \langle I + 2v_1 v_1^T, W \rangle = 2\beta \cdot \gamma$$

with minimizer

$$\text{vec}\,(W^\star) = \frac{2\beta \cdot \gamma}{2\|x\|^2}(v_2 \otimes v_2) = \frac{\gamma}{\|e\|}\hat{e}_2.$$

Clearly, we have feasibility $\|e\|\|(I - \mathbf{X}\mathbf{X}^\dagger)\text{vec}\,(W^\star)\| \le 1$ if and only if $\gamma \le 1$. Substituting this particular $W^\star$ into (39) yields

$$\mathbf{X}y^\star - \text{vec}\,(W^\star) = \frac{\sqrt{1-\gamma^2}}{\|e\|}\hat{e}_1 + \frac{\gamma}{\|e\|}\hat{e}_2.$$

Substituting (38) yields

$$\mathbf{e}^T[\mathbf{X}y^\star - \text{vec}\,(W^\star)] = \sqrt{1-\gamma^2}\cos\theta + \gamma\sin\theta$$

as desired. ■

7.2. Proof of Theorem 12

We consider the condition number optimization problem from Zhang et al. (2018a):

$$\eta(x,z) \quad \equiv \quad \max_{\eta,\mathbf{H}} \left\{ \eta \quad : \quad \mathbf{X}^T\mathbf{H}\mathbf{e} = 0, \quad 2\mathrm{mat}(\mathbf{H}\mathbf{e}) + \mathbf{X}^T\mathbf{H}\mathbf{X} \succeq 0, \quad \eta I \preceq \mathbf{H} \preceq I \right\}. \quad (40)$$

Its optimal value satisfies the following identity with respect to our original LMI in (21):

$$\delta(x,z) = \mathrm{LMI}(x,z) = \frac{1-\eta(x,z)}{1+\eta(x,z)} = 1 - \frac{2}{1+1/\eta(x,z)}. \quad (41)$$

The latter equality shows that $\delta(x,z)$ is a decreasing function of $\eta(x,z)$. This allows us to lower-bound $\delta(x,z)$ by upper-bounding $\eta(x,z)$.

Next, we relax (40) to the following problem

$$\eta_{\mathrm{ub}}(x,z) \quad \equiv \quad \max_{\eta,\mathbf{H}} \left\{ \eta \quad : \quad \mathbf{X}^T\mathbf{H}\mathbf{e} = 0, \quad 2\mathrm{mat}(\mathbf{H}\mathbf{e}) + \mathbf{X}^T\mathbf{X} \succeq 0, \quad \eta I \preceq \mathbf{H} \preceq I \right\}. \quad (42)$$

This yields an upper-bound $\eta_{\mathrm{ub}}(x,z)$ on $\eta(x,z)$ because $\mathbf{H} \succeq I$ implies $\mathbf{X}^T\mathbf{X} \succeq \mathbf{X}^T\mathbf{H}\mathbf{X}$. Problem (42) has Lagrangian dual (we write $v = \mathrm{vec}\,(V)$ to simplify notation)

$$\begin{aligned} \underset{y,U_1,U_2,V=\mathrm{mat}(v)}{\text{minimize}} \quad & \mathrm{tr}(U_2) + \langle \mathbf{X}^T\mathbf{X}, V\rangle \quad (43) \\ \text{subject to} \quad & (\mathbf{X}y - v)\mathbf{e}^T + \mathbf{e}(\mathbf{X}y-v)^T = U_1 - U_2 \\ & \mathrm{tr}(U_1) = 1, \quad U_1, U_2, V \succeq 0. \end{aligned}$$

The dual is strictly feasible (for sufficiently small ϵ, set $y = 0$, $V = \epsilon I$, $U_1 = \eta I - \epsilon W$, and $U_2 = \eta \cdot I_{n^2} + \epsilon W$ with suitable η and W), so Slater's condition is satisfied, strong duality holds, and the objectives coincide. We will implicitly solve the primal problem (42) by solving the dual problem (43).

In the case that $x = 0$, problem (43) yields a trivial solution $y = 0$, $V = zz^T/(2\|z\|^4)$, $U_1 = (z\otimes z)(z\otimes z)^T/\|z\|^4$, and $U_2 = 0$ with objective value $\eta_{\mathrm{ub}}(0,z) = 0$.

In the case that $x \neq 0$, we define $\alpha = (\|z\|\sin\phi)/\|\mathbf{e}\|$ and $\beta = \|x\|^2/\|\mathbf{e}\| > 0$ and make a number of reductions on the dual problem (43). First, we use Lemma 13 to optimize over U_1 and U_2 and the length of y to yield

$$\underset{y,V=\mathrm{mat}(v)\succeq 0}{\text{minimize}} \quad \frac{\mathrm{tr}([M]_-) + \langle \mathbf{X}^T\mathbf{X}, V\rangle}{\mathrm{tr}([M]_+)} \quad \text{where} \quad M = (\mathbf{X}y - v)\mathbf{e}^T + \mathbf{e}(\mathbf{X}y-v)^T. \quad (44)$$

Here, we have divided the objective by the constraint $\mathrm{tr}([M]_+) = 1$ noting that the problem is homogenous over y and V. Substituting explicit expressions for the eigenvalues of M in Lemma 14 yields

$$\underset{y,V=\mathrm{mat}(v)\succeq 0}{\text{minimize}} \quad \frac{\langle \mathbf{X}^T\mathbf{X}, V\rangle + \|\mathbf{e}\|\|\mathbf{X}y-v\|(1-\cos\theta)}{\|\mathbf{e}\|\|\mathbf{X}y-v\|(1+\cos\theta)} \quad \text{where} \quad \cos\theta = \frac{\mathbf{e}^T(\mathbf{X}y-v)}{\|\mathbf{e}\|\|\mathbf{X}y-v\|}. \quad (45)$$

This is a multi-objective optimization over two competing trade-offs: minimizing $\langle \mathbf{X}^T\mathbf{X}, V\rangle$ and maximizing $\cos\theta$. To balance these two considerations, we parameterize over a fixed $\gamma = \langle \mathbf{X}^T\mathbf{X}, V\rangle/2\beta$ and use Lemma 17 to maximize $\cos\theta$. The resulting univariate optimization reads

$$\eta_{\mathrm{ub}}(x,z) \quad = \quad \min_{0\le\gamma\le\alpha} \Psi(\gamma) \equiv \frac{2\beta\cdot\gamma + [1-\psi(\gamma)]}{1+\psi(\gamma)} \quad (46)$$

where the function $\psi(\gamma) = \gamma\alpha + \sqrt{1-\gamma^2}\sqrt{1-\alpha^2}$ defined on Lemma 17 takes on the role of the best choice of $\cos\theta$. Here, one limit $\gamma = 0$ sets $\langle \mathbf{X}^T\mathbf{X}, V\rangle = 0$, while the other $\gamma = \alpha$ sets $\cos\theta = 1$. We cannot have $\gamma < 0$ because $V \succeq 0$. Any choice of $\gamma > \alpha$ will be strictly dominated by $\gamma = \alpha$, because $\gamma = \alpha$ already maximizes $\cos\theta$.

The univariate problem (46) is *quasiconvex*. This follows from the *concavity* of $\psi(\gamma)$ over this range:

$$\psi'(\gamma) = \alpha - \gamma\frac{\sqrt{1-\alpha^2}}{\sqrt{1-\gamma^2}}, \qquad \psi''(\gamma) = -\frac{\sqrt{1-\alpha^2}}{\sqrt{1-\gamma^2}} - \gamma^2\frac{\sqrt{1-\alpha^2}}{(1-\gamma^2)^{3/2}}.$$

Hence, the level sets of $\Psi(\gamma) \ge 0$ are convex:

$$\Psi(\gamma) \le c \quad \Longleftrightarrow \quad 2\beta\cdot\gamma + (1-c) \le (1+c)\psi(\gamma).$$

We will proceed to solve the problem in closed-form and obtain

$$\Psi(\gamma^\star) = \min_{0\le\gamma\le\alpha}\Psi(\gamma) \quad = \quad \begin{cases} \dfrac{1-\sqrt{1-\alpha^2}}{1+\sqrt{1-\alpha^2}} & \text{if } \beta \ge \dfrac{\alpha}{1+\sqrt{1-\alpha^2}}, \quad (47) \\ \dfrac{\beta(\beta-\alpha)}{\beta\alpha - 1} & \text{if } \beta \le \dfrac{\alpha}{1+\sqrt{1-\alpha^2}}. \quad (48) \end{cases}$$

Substituting $\Psi(\gamma^\star) = \eta_{\mathrm{ub}}(x,z)$ into $\eta_{\mathrm{ub}}(x,z) \ge \eta(x,z)$ and using $\eta(x,z)$ to lower-bound $\delta(x,z)$ via (41) completes the proof of the lemma. (Note that setting $x = 0$ sets $\beta = 0$ and yields $\Psi(\gamma^\star) = \eta_{\mathrm{ub}}(0,z) = 0$ as desired.)

First, we verify whether the optimal solution $\gamma^\star$ lies on the boundary of the search interval $[0,\alpha]$, that is $\gamma^\star \in \{0,\alpha\}$. Taking derivatives yields

$$\Psi'(\gamma) = \frac{[2\beta - \psi'(\gamma)](1+\psi(\gamma)) - \psi'(\gamma)[2\beta\cdot\gamma + 1 - \psi(\gamma)]}{(1+\psi(\gamma))^2}.$$

For $\gamma = 0$ to be a stationary point, we require $\Psi'(0) \ge 0$, and hence

$$\begin{aligned} [2\beta - \psi'(0)](1+\psi(0)) &\ge \psi'(0)[2\beta\cdot 0 + 1 - \psi(0)] \\ \Longleftrightarrow \qquad \beta &\ge \frac{\alpha}{1+\sqrt{1-\alpha^2}}. \end{aligned}$$

In this case, we have $\Psi(0) = (1-\sqrt{1-\alpha^2})/(1+\sqrt{1-\alpha^2})$, which is the expression in (47). The choice $\gamma = \alpha$ cannot be stationary, because $\Psi'(\alpha) \le 0$ would imply

$$\begin{aligned} [2\beta - \psi'(\alpha)](1+\psi(\alpha)) - \psi'(\alpha)[2\beta\cdot\alpha + 1 - \psi(\alpha)] &\le 0, \\ \Longleftrightarrow \qquad 2\beta(1+1) &\le 0, \end{aligned}$$

which is impossible as we have $\beta = \|x\|^2/\|e\| > 0$ by hypothesis.

Otherwise, the optimal solution $\gamma^\star$ lies in the interior of the search interval $[0, \alpha]$, that is $\gamma^\star \in (0, \alpha)$. In this case, we simply relax the bound constraints on γ and solve the unconstrained problem as a linear fractional conic program

$$\min_{|\gamma|\le 1} \Psi(\gamma) = \min_{\|\xi\|\le 1} \left\{ \frac{1 + (c-d)^T \xi}{1 + d^T \xi} \right\}$$

where

$$c = \begin{bmatrix} 2\beta \\ 0 \end{bmatrix}, \qquad d = \begin{bmatrix} \alpha \\ \sqrt{1-\alpha^2} \end{bmatrix}, \qquad \xi = \begin{bmatrix} \gamma \\ \sqrt{1-\gamma^2} \end{bmatrix}.$$

(Note that the relaxation $\|\xi\| \le 1$ is always tight, because the linear fractional objective is always monotonous with respect to scaling of ξ.) Defining $q = \xi/(1 + d^T\xi)$ and $q_0 = 1/(1 + d^T\xi) \ge 0$ rewrites this as the second-order cone program

$$\Psi(\gamma^\star) = \min_{\|q\|\le q_0} \left\{ \begin{bmatrix} c-d \\ 1 \end{bmatrix}^T \begin{bmatrix} q \\ q_0 \end{bmatrix} : \begin{bmatrix} d \\ 1 \end{bmatrix}^T \begin{bmatrix} q \\ q_0 \end{bmatrix} = 1 \right\}$$

that admits a strictly feasible point $q = 0$ and $q_0 = 1$. Accordingly, the Lagrangian dual has zero duality gap:

$$\Psi(\gamma^\star) = \max_\lambda \{\lambda : \|c - (1+\lambda)d\| \le (1-\lambda)\}.$$

If the maximum $\lambda^\star$ exists, then it must attain the inequality, as in

$$\|c - (1+\lambda^\star)d\|^2 = (1-\lambda^\star)^2.$$

We can simply solve this quadratic equation

$$c^T c - 2c^T d(1+\lambda^\star) + (1+\lambda^\star)^2 d^T d = 4 - 4(\lambda^\star + 1) + (\lambda^\star + 1)^2$$

for the optimal $\lambda^\star = \Psi(\gamma^\star)$. Noting that $d^T d = 1$, we actually have just a single root

$$1 + \lambda^\star = \frac{(c^T c - 4)}{2(c^T d - 2)} = \frac{\beta^2 - 1}{\alpha\beta - 1} = 1 + \frac{\beta(\beta - \alpha)}{\alpha\beta - 1},$$

and this yields the expression (48).

8. Numerical Results

An important advantage of our formulation is that $\delta(X, Z)$ can be evaluated numerically in cases where an exact closed-form solution does not exist (or is too difficult to obtain). In this section, we augment our analysis with a numerical study. In the rank $r = 1$ case, we exhaustively evaluate $\delta(x, z)$ over its two degrees of freedom to gain insight on its behavior, and also to quantify the conservatism of the lower-bound in Theorem 12. In the rank $r \ge 1$ case, we sample $\delta(X, Z)$ uniformly at random over X and Z, in order to understand its distribution and hypothesize on higher-rank versions of our recovery guarantees.

In our experiments, we implement Algorithm 1 in MATLAB. We parse the LMI problem using YALMIP (Lofberg, 2004) and solve it using MOSEK (Andersen and Andersen, 2000). All algorithms parameters (e.g. accuracy, iterations, etc.) are left at their default values.

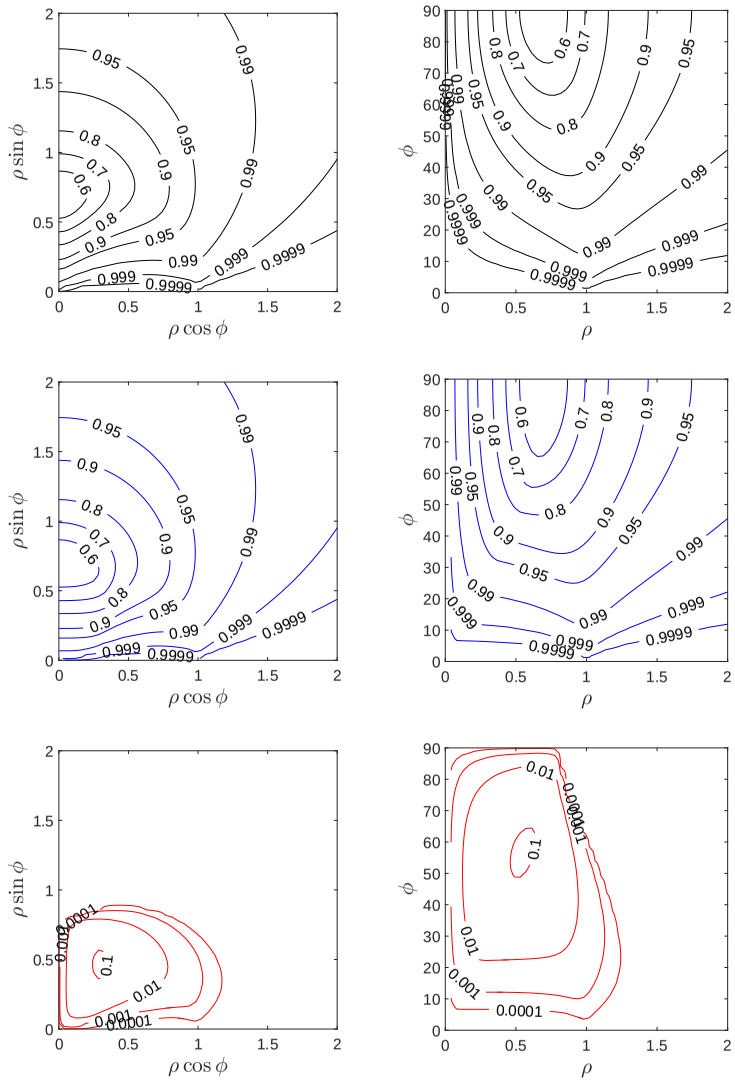


Figure 1: The function $\delta(x, z)$ and its lower bound $\delta_{\text{lb}}(x, z)$ visualized with respect to the length ratio $\rho = \|x\|/\|z\|$ and the incidence angle $\phi = \arccos(x^T z/\|x\|\|z\|)$: **(top)** the function $\delta(x, z)$; **(middle)** the lower-bound $\delta_{\text{lb}}(x, z)$; **(bottom)** the error $\delta(x, z) - \delta_{\text{lb}}(x, z)$; **(left)** rectangular coordinates; **(right)** polar coordinates.

8.1. Visualizing $\delta(x,z)$ and $\delta_{\text{lb}}(x,z)$ for rank $r=1$

Using a suitable orthogonal projector P, we can reduce the function $\delta(x,z)$ down to two underlying degrees of freedom: the length ratio $\rho = \|x\|/\|z\|$ and the incidence angle $\phi = \arccos(x^T z/\|x\|\|z\|)$. First, without loss of generality, we assume that the ground truth $M^\star = zz^T$ has unit norm $\|M^\star\|_F = 1$. (Otherwise, we can suitably rescale all arguments below.) Then, the following projector P satisfies $PP^T x = x$ and $PP^T z = z$ with

$$P = \begin{bmatrix} z & \dfrac{(I - zz^T)x}{\|(I - zz^T)x\|} \end{bmatrix}, \qquad P^T x = \begin{bmatrix} \rho\cos\phi \\ \rho\sin\phi \end{bmatrix}, \qquad P^T z = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{49}$$

Applying this particular P to Lemma 9 yields the following

$$\delta(x,z) = \delta(P^T x, P^T z) = \delta\left(\begin{bmatrix} \rho\cos\phi \\ \rho\sin\phi \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right). \tag{50}$$

In fact, this two-variable function is symmetric over its four rectangular quadrants

$$\delta\left(\begin{bmatrix} \rho\cos\phi \\ \rho\sin\phi \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \delta\left(\begin{bmatrix} \pm\rho\cos\phi \\ \pm\rho\sin\phi \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \tag{51}$$

because either $\pm z$ corresponds to the same ground truth, and because the second column of P can point in either $\pm(I - zz^T)x$.

Accordingly, we can use (50) and (51) to visualize $\delta(x,z)$ as a two-dimensional graph, either in rectangular coordinates over $(\rho\cos\phi, \rho\sin\phi) \in [0, \rho_{\max}]^2$, or in polar coordinates over $(\rho,\phi) \in [0,\rho_{\max}] \times [0, \pi/2]$. Moreover, we can plot our closed-form lower-bound $\delta_{\text{lb}}(x,z)$ on the same axes, in order to quantify its conservatism $\delta(x,z) - \delta_{\text{lb}}(x,z)$.

The top row of Figure 1 plots $\delta(x,z)$ in rectangular and polar coordinates. The plot shows $\delta(x,z)$ as a smooth function with a single basin at $\rho = 1/\sqrt{2}$ and $\phi = 90°$. Outside of a narrow region with $1/2 \le \rho \le 1$ and $\phi \ge 45°$, we have $\delta(x,z) \ge 0.9$. For smaller RIP constants, spurious local minima must appear in a narrow region—they cannot occur arbitrarily anywhere. Excluding this region—as in our local guarantee in Theorem 4—allows much larger RIP constants δ to be accommodated.

The middle and bottom rows of Figure 1 plots $\delta_{\text{lb}}(x,z)$ and $\delta(x,z) - \delta_{\text{lb}}(x,z)$ in rectangular and polar coordinates. The two functions match within 0.01 for either $\rho \ge 1$ or $\phi \le 30°$, and fully concur in the asymptotic limits $\rho \to \{0, +\infty\}$ and $\phi \to \{0°, 90°\}$. The greatest error of around 0.1 occurs at $\rho \approx 0.5$ and $\phi \approx 55°$. We conclude that $\delta_{\text{lb}}(x,z)$ is a high quality approximation for $\delta(x,z)$.

8.2. Distribution of $\delta(X,Z)$ for rank $r \ge 1$

In the high-rank case, a simple characterization of $\delta(X,Z)$ is much more elusive. Given a fixed rank-r ground truth $M^\star$, let its corresponding eigendecomposition be written as $M^\star = V\Lambda V^T$ where $V \in \mathbb{R}^{n\times r}$ is orthogonal and Λ is diagonal. By setting $Z = V\Lambda^{1/2}$ and suitably selecting an orthogonal projector P, it is always possible to satisfy

$$\delta(X,Z) = \delta(P^T X, P^T Z) = \delta\left(\begin{bmatrix} \hat{X}_1 \\ \hat{X}_2 \end{bmatrix}, \begin{bmatrix} \Lambda^{1/2} \\ 0 \end{bmatrix}\right), \tag{52}$$

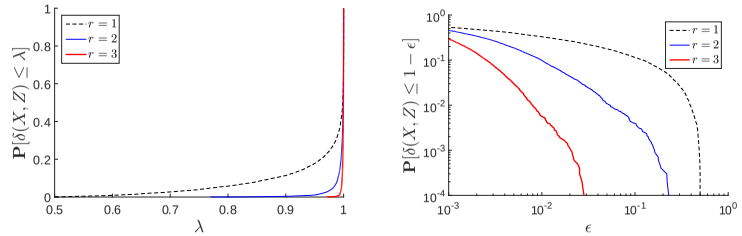


Figure 2: Empirical cumulative distribution of $\delta(X,Z)$ over $N = 10^4$ samples of $X, Z \in \mathbb{R}^{n\times r}$ where $X_{i,j}, Z_{i,i} \sim \text{Gaussian}(0,1)$: **(left)** linear plot of $\mathbf{P}[\delta(X,Z) \le \lambda]$ over $\lambda \in [1/2, 1]$; **(right)** logarithmic plot of $\mathbf{P}[\delta(X,Z) \le 1-\epsilon]$ over the tail $\epsilon \in [10^{-3}, 10^0]$.

where $\hat{X}_1, \hat{X}_2 \in \mathbb{R}^{r\times r}$. While (52) bares superficial similarities to (50), the equation now contains at least $2r^2 + r - 1$ degrees of freedom. Even $r = 2$ results in 9 degrees of freedom, which is too many to visualize.

Instead, we sample $\delta(X,Z)$ uniformly at random over its underlying degrees of freedom. Specifically, we select all elements in $X \in \mathbb{R}^{n\times r}$ and only the diagonal elements of $Z \in \mathbb{R}^{n\times r}$ independently and identically distributed from the standard Gaussian, as in $X_{i,j}, Z_{i,i} \sim \text{Gaussian}(0,1)$. We then use Algorithm 1 to evaluate $\delta(X,Z)$.

Figure 2 plots the empirical cumulative distributions for $r \in \{1,2,3\}$ from $N = 10^4$ samples. We see that each increase in rank r results in a sizable reduction in the distribution tail. The rank $r = 1$ trials yielded $\delta(x,z)$ arbitrarily close to the minimum value of $1/2$, but the rank $r = 2$ trials were only able to find $\delta(X,Z) \approx 0.8$. The rank $r = 3$ trials were even more closely concentrated about one, with the minimum at $\delta(X,Z) \approx 0.97$. These results suggest that higher rank problems are generically easier to solve, because larger RIP constants are sufficient to prevent the points from being spurious local minima. They also suggest that $\delta(X,Z) \ge 1/2$ over *all* rank $r \ge 1$, though this is not guaranteed, because "bad" choices of X, Z can always exist on a lower-dimensional zero-measure set.

9. Conclusions

The low-rank matrix recovery problem is known to contain *no spurious local minima* under a restricted isometry property (RIP) with a sufficiently small RIP constant δ. In this paper, we introduce a proof technique capable of establishing RIP thresholds that are both *necessary* and *sufficient* for exact recovery. Specifically, we define $\delta(X,Z)$ as the *smallest* RIP constant associated with a counterexample with *fixed* ground truth $M^\star = ZZ^T$ and *fixed* spurious point X, and define $\delta^\star = \min_{X,Z} \delta(X,Z)$ as the smallest RIP constant over all counterexamples. Then, δ-RIP low-rank matrix recovery contains no spurious local minima if and only if $\delta < \delta^\star$.

Our key insight is to show that $\delta(X,Z)$ has an *exact* convex reformulation. In the rank-1 case, the resulting problem is sufficiently simple that it can be relaxed and solved in closed-

form. Using this closed-form bound, we prove that $\delta < 1/2$ is both necessary and sufficient for exact recovery from any arbitrary initial point. For larger RIP constants $\delta \geq 1/2$, we show that an initial point x_0 satisfying $f(x_0) \leq (1-\delta)^2 f(0)$ is enough to guarantee exact recovery using a descent algorithm. It is important to emphasize, however, that these sharp results are derived specifically for the rank-1 case.

Acknowledgements

We are grateful to Salar Fattahi for a meticulous reading and detailed comments, and to Salar Fattahi and Cï¿œdric Josz for fruitful discussions. We thank two anonymous reviewers for helpful comments and for pointing out typos. This work was supported by grants from ONR, AFOSR, ARO, and NSF.

Appendix A. Detailed proof of Lemma 9

Given $X, Z \in \mathbb{R}^{n\times r}$, define $\mathbf{e} \in \mathbb{R}^{n^2}$ and $\mathbf{X} \in \mathbb{R}^{n^2\times nr}$ to satisfy the following with respect to X and Z

$$\mathbf{e} = \mathrm{vec}\,(XX^T - ZZ^T), \qquad \mathbf{X}\mathrm{vec}\,(Y) = \mathrm{vec}\,(XY^T + YX^T) \qquad \forall Y \in \mathbb{R}^{n\times r}, \tag{53}$$

Let $P \in \mathbb{R}^{n\times d}$ with $d \leq n$ satisfy

$$P^TP = I_d, \qquad PP^TX = X, \qquad PP^TZ = Z$$

and define $\mathbf{P} = P \otimes P$ and the projections $\hat{X} = P^TX$ and $\hat{Z} = P^TZ$. Define $\hat{\mathbf{e}} \in \mathbb{R}^{d^2}$ and $\hat{\mathbf{X}} \in \mathbb{R}^{d\times dr}$ to satisfy (53) with X, Z replaced by $\hat{X}, \hat{Z}$.

Our goal is to show that

$$\delta = \hat{\delta}, \qquad \mathbf{H} = \mathbf{P}\hat{\mathbf{H}}\mathbf{P}^T + (I - \mathbf{P}\mathbf{P}^T),$$

satisfy the primal feasibility equations

$$\mathbf{X}^T\mathbf{H}\mathbf{e} = 0, \tag{54a}$$
$$2[I_r \otimes \mathrm{mat}(\mathbf{H}\mathbf{e})] + \mathbf{X}^T\mathbf{H}\mathbf{e} \succeq 0, \tag{54b}$$
$$(1-\delta)I \preceq \mathbf{H} \preceq (1+\delta)I, \tag{54c}$$

and that

$$y = (I_r \otimes P)\hat{y}, \qquad U_1 = \mathbf{P}\hat{U}_1\mathbf{P}^T, \qquad U_2 = \mathbf{P}\hat{U}_2\mathbf{P}^T, \qquad V = (I_r \otimes P)\hat{V}(I_r \otimes P)^T$$

satisfy the dual feasibility equations

$$\sum_{j=1}^{r}(\mathbf{X}y - \mathrm{vec}\,(V_{j,j}))\mathbf{e}^T + \mathbf{e}(\mathbf{X}y - \mathrm{vec}\,(V_{j,j}))^T - \mathbf{X}V\mathbf{X}^T = U_1 - U_2, \tag{55a}$$
$$\mathrm{tr}(U_1 + U_2) = 1, \tag{55b}$$
$$V \succeq 0, \quad U_1 \succeq 0, \quad U_2 \succeq 0, \tag{55c}$$

under the hypothesis that $(\hat{\delta}, \hat{\mathbf{H}})$ and $(\hat{y}, \hat{U}_1, \hat{U}_2, \hat{V})$ satisfy (54) and (55) with $\mathbf{e}, \mathbf{X}$ replaced by $\hat{\mathbf{e}}, \hat{\mathbf{X}}$.

We can immediately verify (54c), (55b), and (55c) using the orthogonality of $\mathbf{P}$. To verify the remaining equations, we will use the following identities.

Claim 18 *We have*

$$\mathbf{e} = \mathbf{P}\hat{\mathbf{e}}, \qquad \mathbf{X}(I_r \otimes P) = \mathbf{P}\hat{\mathbf{X}} \qquad \mathbf{P}^T\mathbf{X} = \hat{\mathbf{X}}(I_r \otimes P)^T.$$

Proof For all $Y \in \mathbb{R}^{n\times r}$ and $\hat{Y} \in \mathbb{R}^{d\times r}$, we have

$$\mathbf{e} = \mathrm{vec}\,(XX^T - ZZ^T) = \mathrm{vec}\,[P(\hat{X}\hat{X}^T - \hat{Z}\hat{Z}^T)P^T] = (P \otimes P)\hat{\mathbf{e}},$$
$$\mathbf{X}(I_r \otimes P)\mathrm{vec}\,(\hat{Y}) = \mathbf{X}\mathrm{vec}\,(P\hat{Y}) = \mathrm{vec}\,[P(\hat{X}\hat{Y}^T + \hat{Y}\hat{X}^T)P^T] = \mathbf{P}\hat{\mathbf{X}}\mathrm{vec}\,(\hat{Y}),$$
$$\mathbf{P}^T\mathbf{X}\mathrm{vec}\,(Y) = \mathrm{vec}\,[(P^TX)(P^TY)^T + (P^TY)(P^TX)^T] = \hat{\mathbf{X}}\mathrm{vec}\,(P^TY) = \hat{\mathbf{X}}(I_r \otimes P)^T\mathrm{vec}\,(Y).$$

■

Now, we have (55a) from

$$\mathbf{X}y - \mathrm{vec}\,(V_{j,j}) = \mathbf{X}(I_r \otimes P)\hat{y} - \mathbf{P}\mathrm{vec}\,(\hat{V}_{j,j}) = \mathbf{P}(\hat{\mathbf{X}}\hat{y} - \mathrm{vec}\,(\hat{V}_{j,j})),$$
$$\mathbf{X}V\mathbf{X}^T = \mathbf{X}(I_r \otimes P)\hat{V}(I_r \otimes P)^T\mathbf{X} = \mathbf{P}(\hat{\mathbf{X}}\hat{V}\hat{\mathbf{X}}^T)\mathbf{P}^T.$$

To prove (54a), we use

$$\mathbf{X}^T\mathbf{H}\mathbf{e} = \mathbf{X}^T\mathbf{H}(\mathbf{P}\hat{\mathbf{e}}) = \mathbf{X}^T(\mathbf{P}\hat{\mathbf{H}})\hat{\mathbf{e}} = (I_r \otimes P)\hat{\mathbf{X}}^T\hat{\mathbf{H}}\hat{\mathbf{e}}.$$

Lastly, to prove (54b), we define

$$\mathbf{S} = 2 \cdot [I_r \otimes \mathrm{mat}(\mathbf{H}e)] + \mathbf{X}^T\mathbf{H}\mathbf{X}$$

and $P_\perp$ as the orthogonal complement of P. Then, observe that

$$I_r \otimes \mathrm{mat}(\mathbf{H}e) = I_r \otimes (P\,\mathrm{mat}(\hat{\mathbf{H}}\hat{e})\,P^T) = (I_r \otimes P)(I_r \otimes \mathrm{mat}(\hat{\mathbf{H}}\hat{e}))(I_r \otimes P)^T,$$

and that

$$\mathbf{X}^T\mathbf{H}\mathbf{X}(I_r \otimes P) = \mathbf{X}^T\mathbf{H}(\mathbf{P}\hat{\mathbf{X}}) = \mathbf{X}^T(\mathbf{P}\hat{\mathbf{H}})\hat{\mathbf{X}} = (I_r \otimes P)\hat{\mathbf{X}}^T\hat{\mathbf{H}}\hat{\mathbf{X}}.$$

Hence, we have

$$(I_r \otimes P)^T\mathbf{S}(I_r \otimes P) = 2 \cdot [I_r \otimes \mathrm{mat}(\hat{\mathbf{H}}\hat{e})] + \hat{\mathbf{X}}\hat{\mathbf{H}}\hat{\mathbf{X}} \succeq 0,$$
$$(I_r \otimes P_\perp)^T\mathbf{S}(I_r \otimes P_\perp) = (I_r \otimes P_\perp)^T\mathbf{X}^T\mathbf{H}\mathbf{X}(I_r \otimes P_\perp) \succeq 0,$$
$$(I_r \otimes P_\perp)^T\mathbf{S}(I_r \otimes P) = 0,$$

and this shows that $\mathbf{S} \succeq 0$ as desired.

Appendix B. Detailed proof of Lemma 10

Given $X, Z \in \mathbb{R}^{n\times r}$, let $P = \text{orth}([X, Z])$ and $\mathbf{P} = P \otimes P$. Our goal is to show that

$$\delta = \hat{\delta}, \qquad \mathbf{H} = \mathbf{P}\hat{\mathbf{H}}\mathbf{P}^T$$

satisfy the primal feasibility equations

$$\mathbf{X}^T\mathbf{H}\mathbf{e} = 0, \tag{56a}$$

$$2[I_r \otimes \text{mat}(\mathbf{H}\mathbf{e})] + \mathbf{X}^T\mathbf{H}\mathbf{e} \succeq 0, \tag{56b}$$

$$(1-\delta)I \preceq \mathbf{P}^T\mathbf{H}\mathbf{P} \preceq (1+\delta)I, \tag{56c}$$

and that

$$y = (I_r \otimes P)\hat{y}, \qquad U_1 = \hat{U}_1, \qquad U_2 = \hat{U}_2, \qquad V = (I_r \otimes P)\hat{V}(I_r \otimes P)^T$$

satisfy the dual feasibility equations

$$\sum_{j=1}^{r}(\mathbf{X}y - \text{vec}\,(V_{j,j}))\mathbf{e}^T + \mathbf{e}(\mathbf{X}y - \text{vec}\,(V_{j,j}))^T - \mathbf{X}V\mathbf{X}^T = \mathbf{P}(U_1 - U_2)\mathbf{P}^T, \tag{57a}$$

$$\text{tr}[\mathbf{P}(U_1 + U_2)\mathbf{P}^T] = 1, \tag{57b}$$

$$V \succeq 0, \quad U_1 \succeq 0, \quad U_2 \succeq 0, \tag{57c}$$

under the hypothesis that $(\hat{\delta}, \hat{\mathbf{H}})$ and $(\hat{y}, \hat{U}_1, \hat{U}_2, \hat{V})$ satisfy (54) and (55) with $\mathbf{e}, \mathbf{X}$ replaced by $\hat{\mathbf{e}}, \hat{\mathbf{X}}$. The exact steps for verifying (56) and (57) are identical to the proof of Lemma 9, and are omitted for brevity.

References

Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on Machine learning*, pages 17–24. ACM, 2007.

Erling D Andersen and Knud D Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, pages 197–232. Springer, 2000.

Sohail Bahmani and Justin Romberg. Phase retrieval meets statistical learning theory: A flexible convex relaxation. In *Artificial Intelligence and Statistics*, pages 252–260, 2017.

Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. Dropping convexity for faster semi-definite optimization. In *Conference on Learning Theory*, pages 530–582, 2016a.

Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881, 2016b.

Nicolas Boumal, P-A Absil, and Coralia Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, page drx080, 2018. doi: 10.1093/imanum/drx080. URL `http://dx.doi.org/10.1093/imanum/drx080`.

T Tony Cai and Anru Zhang. Sharp RIP bound for sparse signal and low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 35(1):74–93, 2013.

Emmanuel J Candes and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory*, 57(4):2342–2359, 2011.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Emmanuel J Candes, Thomas Strohmer, and Vladislav Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

Coralia Cartis, Nicholas IM Gould, and Ph L Toint. Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108, 2012.

Tony F Chan. Rank revealing qr factorizations. *Linear algebra and its applications*, 88: 67–82, 1987.

Pengwen Chen, Albert Fannjiang, and Gi-Ren Liu. Phase retrieval with one or two diffraction patterns by alternating projections with the null initialization. *Journal of Fourier Analysis and Applications*, 24(3):719–758, 2018.

Yuxin Chen and Emmanuel Candes. Solving random quadratic systems of equations is nearly as easy as solving linear systems. In *Advances in Neural Information Processing Systems*, pages 739–747, 2015.

Andrew R Conn, Nicholas IM Gould, and Ph L Toint. *Trust region methods*, volume 1. Siam, 2000.

Simon S Du, Chi Jin, Jason D Lee, Michael I Jordan, Aarti Singh, and Barnabas Poczos. Gradient descent can take exponential time to escape saddle points. In *Advances in Neural Information Processing Systems*, pages 1067–1077, 2017.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points–online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242, 2017.

Tom Goldstein and Christoph Studer. Phasemax: Convex phase retrieval via basis pursuit. *IEEE Transactions on Information Theory*, 2018.

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU Press, 3 edition, 1996.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732, 2017.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010a.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010b.

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

Johan Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

Marco Mondelli and Andrea Montanari. Fundamental limits of weak recovery with applications to phase retrieval. *Foundations of Computational Mathematics*, pages 1–71, 2018.

Katta G Murty and Santosh N Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129, 1987.

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. Phase retrieval using alternating minimization. In *Advances in Neural Information Processing Systems*, pages 2796–2804, 2013.

Dohyung Park, Anastasios Kyrillidis, Constantine Carmanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. In *Artificial Intelligence and Statistics*, pages 65–74, 2017.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably. *SIAM Journal on Imaging Sciences*, 11(4):2165–2204, 2018.

Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.

Sujay Sanghavi, Rachel Ward, and Chris D White. The local convexity of solving systems of quadratic equations. *Results in Mathematics*, 71(3-4):569–608, 2017.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

Andreas M Tillmann and Marc E Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2014.

Gang Wang, Georgios B Giannakis, and Yonina C Eldar. Solving systems of random quadratic equations via truncated amplitude flow. *IEEE Transactions on Information Theory*, 64(2):773–794, 2018.

HuiMin Wang and Song Li. The bounds of restricted isometry constants for low rank matrices recovery. *Science China Mathematics*, 56(6):1117–1127, 2013.

Richard Y Zhang, Cédric Josz, Somayeh Sojoudi, and Javad Lavaei. How much restricted isometry is needed in nonconvex matrix recovery? In *Advances in Neural Information Processing Systems*, 2018a. arXiv:1805.10251.

Richard Y Zhang, Javad Lavaei, and Ross Baldick. Spurious critical points in power system state estimation. In *Hawaii International Conference on System Sciences (HICSS)*, 2018b.

Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. In *Advances in Neural Information Processing Systems*, pages 559–567, 2015.

Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems*, pages 109–117, 2015.

Zhihui Zhu, Qiuwei Li, Gongguo Tang, and Michael B Wakin. Global optimality in low-rank matrix optimization. *IEEE Transactions on Signal Processing*, 66(13):3614–3628, 2018.

Journal of Machine Learning Research 20 (2019) 1-58 Submitted 3/19; Revised 6/19; Published 6/19

Learning Attribute Patterns in High-Dimensional Structured Latent Attribute Models

Yuqi Gu YUQIGU@UMICH.EDU
Gongjun Xu GONGJUN@UMICH.EDU
Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA

Editor: Animashree Anandkumar

Abstract

Structured latent attribute models (SLAMs) are a special family of discrete latent variable models widely used in social and biological sciences. This paper considers the problem of learning significant attribute patterns from a SLAM with potentially high-dimensional configurations of the latent attributes. We address the theoretical identifiability issue, propose a penalized likelihood method for the selection of the attribute patterns, and further establish the selection consistency in such an overfitted SLAM with a diverging number of latent patterns. The good performance of the proposed methodology is illustrated by simulation studies and two real datasets in educational assessments.

1. Introduction

Structured Latent Attribute Models (SLAMs) are widely used statistical and machine learning tools in modern social and biological sciences. SLAMs offer a framework to achieve fine-grained inference on individuals' latent attributes based on their observed multivariate responses, and also to obtain the latent subgroups of a population based on the inferred attribute patterns. In practice, each latent attribute is often assumed to be discrete and has particular scientific interpretation, such as mastery or deficiency of some targeted skill in educational assessments (Junker and Sijtsma, 2001; de la Torre, 2011), presence or absence of some underlying mental disorder in psychiatric diagnosis (Templin and Henson, 2006; de la Torre et al., 2018), and the existence or nonexistence of some disease pathogen in subjects' biological samples (Wu et al., 2017). In these scenarios, the framework of SLAMs enables one to simultaneously achieve the machine learning task of clustering, and the scientific purpose of diagnostic inference.

Different from the exploratory nature of traditional latent variable models, SLAMs often have some additional scientific information for model fitting. In particular, the observed variables are assumed to have certain structured dependence on the unobserved latent attributes, where the dependence is introduced through a binary design matrix to respect the scientific context. The rich structure and nice interpretability of SLAMs make them popular in many scientific disciplines, such as cognitive diagnosis in educational assessment (Junker and Sijtsma, 2001; von Davier, 2008; Henson et al., 2009; Rupp et al., 2010; de la Torre, 2011), psychological and psychiatric measurement for diagnosis of mental disorders (Templin and Henson, 2006; de la Torre et al., 2018), and epidemiological and medical studies for scientifically constrained clustering (Wu et al., 2017, 2018).

One challenge in modern applications of SLAMs is that the number of discrete latent attributes could be large, leading to a high-dimensional space for all the possible configurations of the attributes, i.e., a high-dimensional space for latent attribute patterns. In many applications, the number of potential patterns is much larger than the sample size. For scientific interpretability and practical use, it is often assumed that not all the possible attribute patterns exist in the population. Examples with a large number of potential latent patterns and a moderate sample size can be found in educational assessments (Lee et al., 2011; Choi et al., 2015; Yamaguchi and Okada, 2018) and the epidemiological diagnosis of disease etiology (Wu et al., 2017). For instance, Example 1 in Section 2 presents a dataset from Trends in International Mathematics and Science Study (TIMSS), which has 13 binary latent attributes (i.e., $2^{13} = 8192$ possible latent attribute patterns) while only 757 students' responses are observed. In cognitive diagnosis, it is of interest to select the significant attribute patterns among these $2^{13} = 8192$ ones. In such high-dimensional scenarios, existing estimation methods often tend to over select the number of latent patterns, and may not scale to datasets with a huge number of patterns. Moreover, theoretical questions remain open on whether and when the "sparse" latent attribute patterns are identifiable and can be consistently learned from data.

Identifiability of SLAMs has long been an issue in the literature (e.g., von Davier, 2008; DeCarlo, 2011; Maris and Bechger, 2009; von Davier, 2014; Xu and Zhang, 2016). SLAMs can be viewed as a special family of restricted latent class models and their identifiability has a close connection with the study of tensor decompositions, by noting that the probability distribution of a SLAM can be viewed as a mixture of specially structured tensor products. In the literature, it is known that unrestricted latent class models are not identifiable (Gyllenberg et al., 1994). Nonetheless, Carreira-Perpinán and Renals (2000) showed through extensive simulations that they are almost always identifiable, which the authors termed as practical identifiability. Allman et al. (2009) further established *generic* identifiability of various latent variable models, including latent class models. Generic identifiability is weaker than strict identifiability, and it implies that the model parameters are almost surely identifiable with respect to the Lebesgue measure of the parameter space. The study of Allman et al. (2009) is based on an identifiability result of the three-way tensor decomposition in Kruskal (1977). Other analysis of tensor decompositions has also been conducted to study the identifiability of various latent variable models (e.g., Drton et al., 2007; Hsu and Kakade, 2013; Anandkumar et al., 2014; Bhaskara et al., 2014; Anandkumar et al., 2015; Jaffe et al., 2018). However, the structural constraints imposed by the design matrix make these results not directly applicable to SLAMs.

With the aid of the structural constraints, strict identifiability of SLAMs has been obtained under certain conditions on the design matrix (Xu, 2017; Xu and Shang, 2018; Gu and Xu, 2019a,b). However, these works either make the strong assumption that all the possible configurations of the attributes exist in the population with positive probabilities (Xu, 2017; Xu and Shang, 2018), or assume these significant attribute patterns are known *a priori* (Gu and Xu, 2019b). These assumptions are difficult to meet in practice for SLAMs with high-dimensional attributes patterns, and the fundamental learnability issue of the sparse patterns in SLAMs remains unaddressed.

©2019 Yuqi Gu and Gongjun Xu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-197.html.

In terms of estimation, learning sparse attribute patterns from a high-dimensional space is related to learning the significant mixture components in a highly overfitted mixture model. Researchers have shown that the estimation of the mixing distributions in overfitted mixture models is technically challenging and it usually leads to nonstandard convergence rate (e.g., Chen, 1995; Ho and Nguyen, 2016; Heinrich and Kahn, 2018). Estimating the number of components in the mixture model goes beyond only estimating the parameters of a mixture, by learning at least the order of the mixing distribution (Heinrich and Kahn, 2018). This problem was also studied in Rousseau and Mengersen (2011) from a Bayesian perspective; however, the Bayesian estimator in Rousseau and Mengersen (2011) may not guarantee the frequentist selection consistency, as to be shown in Section 3. In the setting of SLAMs with the structural constraints and a large number (larger than sample size) of potential latent attribute patterns, it is not clear how to consistently select the significant patterns.

Our contributions in this paper contain the following aspects. First, we characterize the identifiability requirement needed for a SLAM with an arbitrary subset of attribute patterns to be learnable, and establish mild identifiability conditions. Our new identifiability conditions significantly extends the results of previous works (Xu, 2017; Xu and Shang, 2018) to more general and practical settings. Second, we propose a statistically consistent method to perform attribute pattern selection. In particular, we establish theoretical guarantee for selection consistency in the setting of high dimensional latent patterns, where both the sample size and the number of latent patterns can go to infinity. Our analysis also shows that imposing the popular Dirichlet prior on the population proportions would fail to select the true model consistently, when the convergence rate of the SLAM is slower than the usual root-N rate. As for computation, we develop two approximation algorithms to maximize the penalized likelihood for pattern selection. In addition, we propose a fast screening strategy for SLAMs as a preprocessing step that can scale to a huge number of potential patterns, and establish its sure screening property.

The rest of the paper is organized as follows. Section 2 introduces the general setup of structured latent attribute models and motivates our study. Section 3 investigates the learnability requirement and proposes mild sufficient conditions for learnability. Section 4 proposes the estimation methodology and establishes theoretical guarantee for the proposed methods. Section 5 and Section 6 include simulations and real data analysis, respectively. The proofs of all the theoretical results and additional experimental results are included in the Appendix.

2. Model Setup and Motivation

In this section, we first describe the model setup of SLAMs and present several examples. Then we describe the motivation for our study and introduce the problem of interest.

2.1. Structured Latent Attribute Models and Examples

We first introduce the general setup of SLAMs. Consider a SLAM with J designed items which depend on the K latent attributes of interest. There are two types of subject-specific variables in the model, the observed responses to items $\mathbf{R} = (R_1, \ldots, R_J)$ and the latent attribute pattern $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$, both assumed to be binary vectors in this work. The J-

dimensional vector $\mathbf{R} \in \{0,1\}^J$ denotes the observed binary responses to the set of J items. The K-dimensional vector $\boldsymbol{\alpha} \in \{0,1\}^K$ denotes a profile of existence or non-existence of the K attributes.

A key structure that specifies how the observed responses depend on the latent attributes is called the Q-matrix, which is a $J \times K$ matrix with binary entries. We denote $Q = (q_{j,k})$ and $q_{j,k} \in \{1, 0\}$ reflects whether or not the response to item j has statistical dependence on attribute k. In the context of an educational assessment, $q_{j,k} = 1$ implies the jth test item requires the mastery of the kth skill attribute to answer correctly. We denote the jth row vector of Q by $\boldsymbol{q}_j$, then the K-dimensional binary vector $\boldsymbol{q}_j$ reflects the full attribute requirements of item j. For an attribute pattern $\boldsymbol{\alpha}$, we say $\boldsymbol{\alpha}$ possesses all the required attributes of item j, if $\boldsymbol{\alpha} \succeq \boldsymbol{q}_j$, where $\boldsymbol{\alpha} \succeq \boldsymbol{q}_j$ denotes $\alpha_k \geq q_{j,k}$ for all $k = 1, \ldots, K$. Example 1 below gives an example of the Q-matrix.

Example 1 *Trends in International Mathematics and Science Study (TIMSS) is a large scale cross-country educational assessment. TIMSS evaluates the mathematics and science abilities of fourth and eighth graders every four years since 1995. Researchers have used SLAMs to analyze the TIMSS data (e.g., Lee et al., 2011; Choi et al., 2015; Yamaguchi and Okada, 2018). For example, a 23×13 Q-matrix constructed by mathematics educators was specified for the TIMSS 2003 eighth grade mathematics assessment (Choi et al., 2015). Thirteen attributes ($K = 13$) are identified, which fall in five big categories of skill domains measured by the exam, Number, Algebra, Geometry, Measurement, and Data. Table 1 shows the first and last three rows of the Q-matrix, i.e., $\{\boldsymbol{q}_j : j = 1, 2, 3, 21, 22, 23\}$.*

Item	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}
1	1	0	0	0	0	0	0	0	0	0	1	0	1
2	0	0	0	0	0	1	0	0	0	0	0	0	0
3	0	1	0	0	0	0	1	0	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
21	0	0	0	0	1	0	0	0	0	0	0	0	0
22	0	1	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	1	0	0	0	0	1	0	0	0	0

Table 1: Q-matrix, TIMSS 2003 8th Grade Data

The Q-matrix constrains the model parameters in a certain way to reflect the scientific assumptions. We next introduce the model parameters and how the Q-matrix impose constraints on them in general. Conditional on a subject's latent attribute pattern $\boldsymbol{\alpha} \in \{0,1\}^K$, his/her responses to the J items are assumed to be independent Bernoulli random variables with parameters $\theta_{1,\boldsymbol{\alpha}}, \ldots, \theta_{J,\boldsymbol{\alpha}}$. Specifically, $\theta_{j,\boldsymbol{\alpha}} = \mathbb{P}(R_j = 1 \mid \boldsymbol{\alpha})$ denotes the positive response probability, and is also called an item parameter of item j. We collect all the item parameters in the matrix $\boldsymbol{\Theta} = (\theta_{j,\boldsymbol{\alpha}})$, which has size $J \times 2^K$ with rows indexed by the J items and columns by the 2^K attribute patterns. For pattern $\boldsymbol{\alpha} \in \{0,1\}^K$, we denote its corresponding column vector in $\boldsymbol{\Theta}$ by $\boldsymbol{\Theta}_{\cdot,\boldsymbol{\alpha}}$.

One key assumption in SLAMs is that for a latent attribute pattern $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ and item j, the parameter $\theta_{j,\boldsymbol{\alpha}}$ is only determined by whether $\boldsymbol{\alpha}$ possesses the attributes

in the set $\mathcal{K}_j = \{k \in \{1, \ldots, K\} : q_{j,k} = 1\}$; that is, those attributes related to item j as specified in the Q-matrix. We will sometimes call the attributes in $\mathcal{K}_j$ the *required attributes of item j*. Under this assumption, all latent attribute patterns in the set

$$\mathcal{C}_j = \{\boldsymbol{\alpha} \in \{0,1\}^K : \boldsymbol{\alpha} \succeq \boldsymbol{q}_j\} \tag{1}$$

share the same value of $\theta_{j,\boldsymbol{\alpha}}$; namely,

$$\max_{\boldsymbol{\alpha} \in \mathcal{C}_j} \theta_{j,\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha} \in \mathcal{C}_j} \theta_{j,\boldsymbol{\alpha}} \text{ for any } j \in \{1, \ldots, J\}. \tag{2}$$

We will call the set $\mathcal{C}_j$ a *constraint set*. Thus, the Q-matrix puts constraints on $\boldsymbol{\Theta}$ by forcing certain entries of it to be the same. Different SLAMs model the dependence of $\theta_{j,\boldsymbol{\alpha}}$ on the required attributes in $\mathcal{K}_j$ differently to encode different scientific assumptions; please see Examples 2 and 3.

In addition to (2), another key assumption in SLAMs is the monotonicity assumption that

$$\theta_{j,\boldsymbol{\alpha}} > \theta_{j,\boldsymbol{\alpha}'} \text{ for any } \boldsymbol{\alpha} \in \mathcal{C}_j,\ \boldsymbol{\alpha}' \notin \mathcal{C}_j. \tag{3}$$

Constraint (3) is commonly used in our motivating applications of cognitive diagnosis in educational assessments, where (3) indicates subjects mastering all required attributes of an item are more "capable" of giving a positive response to it (i.e., with a larger Bernoulli parameter $\theta_{j,\boldsymbol{\alpha}}$), than those who lack some required attributes. Nonetheless, our theoretical results of model learnability in Section 3 also applies if (3) is relaxed to

$$\theta_{j,\boldsymbol{\alpha}} \neq \theta_{j,\boldsymbol{\alpha}'} \text{ for any } \boldsymbol{\alpha} \in \mathcal{C}_j,\ \boldsymbol{\alpha}' \notin \mathcal{C}_j.$$

This allows more flexibility in the model assumptions of SLAMs used in other applications.

Next we introduce some popular SLAMs in educational and psychological applications. These models are also called Cognitive Diagnosis Models in the psychometrics literature. The first type of SLAMs have exactly two item parameters associated with each item.

Example 2 (two-parameter SLAM) *The two-parameter SLAM specifies exactly two item parameters for each item j, which we denote by θ_j^+ and θ_j^-, with $\theta_j^+ > \theta_j^-$. The popular Deterministic Input Noisy output "And" gate (DINA) model introduced in Junker and Sijtsma (2001) is a two-parameter SLAM. It specifies the general form of $\theta_{j,\boldsymbol{\alpha}}$ can be rewritten as*

$$\theta_{j,\boldsymbol{\alpha}}^{\textit{two-param.}} = \begin{cases} \theta_j^+, & \textit{if } \boldsymbol{\alpha} \in \mathcal{C}_j, \\ \theta_j^-, & \textit{if } \boldsymbol{\alpha} \notin \mathcal{C}_j. \end{cases}$$

In the application of the two-parameter SLAM in educational assessments, the item parameters θ_j^+ and θ_j^- have the following interpretations. The $1 - \theta_j^+$ is called the slipping parameter, denoting the probability of a "capable" subject slips the correct answer, despite mastering all the required attributes of the test item j; and θ_j^- is called the guessing parameter, denoting the probability of a "non-capable" subject coincidentally giving the correct answer by guessing, despite lacking some required attributes of item j. In this case, the unique item parameters in matrix $\boldsymbol{\Theta}$ reduce to $(\boldsymbol{\theta}^+, \boldsymbol{\theta}^-)$, where $\boldsymbol{\theta}^+ = (\theta_1^+, \ldots, \theta_J^+)^\top$ and $\boldsymbol{\theta}^- = (\theta_1^-, \ldots, \theta_J^-)^\top$. Under the two-parameter SLAM, the constraint set of each item j takes the form of (1) *and satisfies* (2) *and* (3).

Another family of SLAMs are the multi-parameter models, which allow each item to have multiple levels of item parameters.

Example 3 (multi-parameter SLAMs) *Multi-parameter SLAMs can be categorized into two general types, the main-effect models and the all-effect models. The main-effect models assume the main effects of the required attributes in $\mathcal{K}_j$ play a role in distinguishing the item parameters, which can be written as*

$$\theta_{j,\boldsymbol{\alpha}}^{\textit{main-eff}} = f\Big(\beta_{j,0} + \sum\nolimits_{k \in \mathcal{K}_j} \beta_{j,k}\alpha_k\Big), \tag{4}$$

where $f(\cdot)$ is a link function. Different link functions $f(\cdot)$ lead to different models, including the popular reduced Reparameterized Unified Model (reduced-RUM; DiBello et al., 1995) with $f(\cdot)$ being the exponential function, the Linear Logistic Model (LLM; Maris, 1999) with $f(\cdot)$ being the sigmoid function, and the Additive Cognitive Diagnosis Model (ACDM; de la Torre, 2011) with $f(\cdot)$ the identity function.

Another type of multi-parameter SLAMs are the all-effect models. The item parameter of an all-effect model can be written as

$$\theta_{j,\boldsymbol{\alpha}}^{\textit{all-eff}} = f\Big(\sum\nolimits_{S \subseteq \mathcal{K}_j} \beta_{j,S} \prod\nolimits_{k \in S} \alpha_k\Big). \tag{5}$$

When $f(\cdot)$ is the identity function, (5) *gives the Generalized DINA (GDINA) model proposed by de la Torre (2011); and when $f(\cdot)$ is the sigmoid function,* (5) *gives the Log-linear Cognitive Diagnosis Models (LCDMs) proposed by Henson et al. (2009); see also the General Diagnostic Models (GDMs) proposed in von Davier (2008).*

Under the multi-parameter SLAMs, the constraint set of each item j also takes the form of (1). Those attribute patterns in $\mathcal{C}_j$ still share the same value of item parameters by the definition; and what is different from the two-parameter counterpart is that those $\boldsymbol{\alpha}$ not in $\mathcal{C}_j$ can have different levels of item parameters. We next give another example of multi-parameter SLAMs.

Example 4 (Deep Boltzmann Machines) *The Restricted Boltzmann Machine (RBM) (Smolensky, 1986; Goodfellow et al., 2016) is a popular neural network model. RBM is an undirected probabilistic graphical model, with one layer of latent (hidden) binary variables, one layer of observed (visible) binary variables, and a bipartite graph structure between the two layers. We denote variables in the observed layer by $\mathbf{R}$ and variables in the latent layer by $\boldsymbol{\alpha}$, with lengths J and K, respectively. Under an RBM, the probability mass function of $\mathbf{R}$ and $\boldsymbol{\alpha}$ is $\mathbb{P}(\mathbf{R}, \boldsymbol{\alpha}) \propto \exp(-\mathbf{R}^\top \boldsymbol{W}^Q \boldsymbol{\alpha} - \boldsymbol{f}^\top \mathbf{R} - \boldsymbol{b}^\top \boldsymbol{\alpha})$, where $\boldsymbol{f}$, $\boldsymbol{b}$, and $\boldsymbol{W}^Q = (w_{j,k})$ are the parameters. The binary Q-matrix then specifies the sparsity structure in $\boldsymbol{W}^Q$, by constraining $w_{j,k} \neq 0$ only if $q_{j,k} \neq 0$. The Deep Boltzmann Machine (DBM) is a generalization of RBM by allowing multiple latent layers. Consider a DBM with two latent layers $\boldsymbol{\alpha}^{(1)}$ and $\boldsymbol{\alpha}^{(2)}$ of length K_1 and K_2, respectively. The probability mass function of $(\mathbf{R}, \boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)})$ in this DBM can be written as*

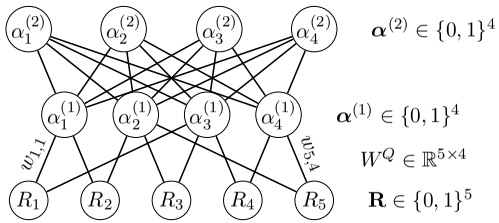
$$\mathbb{P}(\mathbf{R}, \boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}) \propto \exp\Big(-\mathbf{R}^\top \boldsymbol{W}^Q \boldsymbol{\alpha}^{(1)} - (\boldsymbol{\alpha}^{(1)})^\top \boldsymbol{U} \boldsymbol{\alpha}^{(2)} - \boldsymbol{f}^\top \mathbf{R} - \boldsymbol{b}_1^\top \boldsymbol{\alpha}^{(1)} - \boldsymbol{b}_2^\top \boldsymbol{\alpha}^{(2)}\Big), \tag{6}$$

where $\boldsymbol{f} \in \mathbb{R}^J$, $\boldsymbol{b}_i \in \mathbb{R}^{K_i}$ *for* $i = 1, 2$, *and* $\boldsymbol{W}^Q = (w_{j,k}) \in \mathbb{R}^{J \times K_1}$, $\boldsymbol{U} \in \mathbb{R}^{K_1 \times K_2}$ *are model parameters; Figure 1 gives an example of a DBM with a* 5×4 *Q-matrix. For* $\boldsymbol{f} = (f_1, \ldots, f_J)^\top$ *and* $\boldsymbol{\alpha}^{(1)} = (\alpha_1^{(1)}, \ldots, \alpha_{K_1}^{(1)})$, *the conditional distribution of an observed variable* R_j *given the latent variables is*

$$\mathbb{P}(R_j = 1 \mid \boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \cdots) = \mathbb{P}(R_j = 1 \mid \boldsymbol{\alpha}^{(1)}) = \frac{\exp\left(\sum_{k=1}^{K_1} w_{j,k}\alpha_k^{(1)} + f_j\right)}{1 + \exp\left(\sum_{k=1}^{K_1} w_{j,k}\alpha_k^{(1)} + f_j\right)}, \tag{7}$$

where "$\cdots$" represents deeper latent layers that potentially exist in a DBM. Moreover, from (6) *we have* $\mathbb{P}(\mathbf{R} \mid \boldsymbol{\alpha}^{(1)}) = \prod_{j=1}^J \mathbb{P}(R_j \mid \boldsymbol{\alpha}^{(1)})$, *so a DBM satisfies the local independence assumption that the* R_j*'s are conditionally independent given the* $\boldsymbol{\alpha}^{(1)}$. *Therefore, a DBM can be viewed as a multi-parameter main-effect SLAM in* (4) *with a sigmoid link function. Viewing a DBM in this way,* (7) *gives the item parameter* $\theta_{j,\boldsymbol{\alpha}^{(1)}}$, *and the constraint set of each item* j *also takes the form* $\mathcal{C}_j = \{\boldsymbol{\alpha}^{(1)} \in \{0,1\}^{K_1} : \boldsymbol{\alpha}^{(1)} \succeq \boldsymbol{q}_j\}$.

$$Q = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix};$$



$\boldsymbol{\alpha}^{(2)} \in \{0,1\}^4$

$\boldsymbol{\alpha}^{(1)} \in \{0,1\}^4$

$W^Q \in \mathbb{R}^{5\times 4}$

$\mathbf{R} \in \{0,1\}^5$

Figure 1: Deep Boltzmann Machine

2.2. Motivation and Problem

One challenge in modern applications of SLAMs is that the number of potential latent attribute patterns 2^K increases exponentially with K and could be much larger than the sample size N. It is often assumed that a relatively small portion of attribute patterns exist in the population. For instance, Example 1 has $2^K = 2^{13} = 8192$ different configurations of attribute patterns; for the limited sample size 757 there, it is desirable to learn the potentially small set of significant attribute patterns from data.

Another motivation for assuming a small number of attribute patterns exist in the population results from the possible hierarchical structure among the targeted attributes. For instance, in an educational assessment of a set of underlying latent skill attributes, some attributes often serve as prerequisites for some others (Leighton et al., 2004; Templin and Bradshaw, 2014). Specifically, the prerequisite relationship depicts the different level of difficulty of the skill attributes, and also reveals the order in which these skills are learned in the population of students. For instance, if attribute α_1 is a prerequisite for attribute α_2, then the attribute pattern $(\alpha_1 = 0, \alpha_2 = 1)$ does not exist in the population, naturally resulting in a sparsity structure of the existence of attribute patterns. When the number of attributes is large and the underlying hierarchy structure is complex and unknown, it is

desirable to learn the hierarchy of attributes directly from data. In such cases with attribute hierarchy, the number of patterns respecting the hierarchy could be far fewer than 2^K.

The problem of interest is that, given a moderate sample size, how to consistently estimate the small set of latent attribute patterns among all the possible 2^K ones. As discussed in the introduction, in the high-dimensional case when the total number of attribute patterns is large or even larger than the sample size, the questions of when the true model with the significant latent patterns are learnable from data, and how to perform consistent pattern selection, remain open in the literature.

This problem is equivalent to selecting the nonzero elements of the population proportion parameters $\boldsymbol{p} = (p_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \{0,1\}^K)$, where $p_{\boldsymbol{\alpha}}$ denotes the proportion of the subjects with latent pattern $\boldsymbol{\alpha}$ in the population. The $\boldsymbol{p}$ satisfies $p_{\boldsymbol{\alpha}} \in [0,1]$ for all $\boldsymbol{\alpha} \in \{0,1\}^K$ and $\sum_{\boldsymbol{\alpha}\in\{0,1\}^K} p_{\boldsymbol{\alpha}} = 1$. In this work, we will treat the latent attribute patterns $\boldsymbol{\alpha}$ as random variables (random effects). For any subject, his/her attribute pattern is a random vector $\mathbf{A} \in \{0,1\}^K$ that (marginally) follows a categorical distribution with population proportion parameters $\boldsymbol{p} = (p_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \{0,1\}^K)$. One main reason for this random effect assumption is that, when the number of observed variables per subject (i.e., J) does not increase with the sample size N asymptotically, the counterpart fixed effect model can not consistently estimate the model parameters. As a consequence, the fixed effect approach can not give consistent selection of significant attribute patterns. This scenario with relatively small J but larger N and 2^K is commonly seen in the motivating applications in educational and psychological assessments.

We would like to point out that we give the joint distribution of the attributes full flexibility by modeling it as a categorical distribution with $2^K - 1$ free proportion parameters $p_{\boldsymbol{\alpha}}$'s. Modeling in this way allows those "sparse" significant attribute patterns to have arbitrary structures among the 2^K possibilities. On the contrary, any simpler parametric model of the distribution of $\boldsymbol{\alpha}$ with fewer parameters would fail to capture all the possibilities of the attributes' dependency.

Under the introduced notations, the probability mass function of a subject's response vector $\mathbf{R} = (R_1, \ldots, R_J)^\top$ can be written as $\mathbb{P}(\mathbf{R} = \boldsymbol{r} \mid \boldsymbol{\Theta}, \boldsymbol{p}) = \sum_{\boldsymbol{\alpha}\in\{0,1\}^K} p_{\boldsymbol{\alpha}} \prod_{j=1}^J \theta_{j,\boldsymbol{\alpha}}^{r_j}(1-\theta_{j,\boldsymbol{\alpha}})^{1-r_j}$, for $\boldsymbol{r} \in \{0,1\}^J$. Alternatively, the responses can be viewed as a J-th order tensor and the probability mass function of $\mathbf{R}$ can be written as a probability tensor

$$\mathbb{P}(\mathbf{R} \mid \boldsymbol{\Theta}, \boldsymbol{p}) = \sum_{l=1}^{2^K} p_{\boldsymbol{\alpha}_l} \begin{pmatrix} \theta_{1,\boldsymbol{\alpha}_l} \\ 1 - \theta_{1,\boldsymbol{\alpha}_l} \end{pmatrix} \circ \begin{pmatrix} \theta_{2,\boldsymbol{\alpha}_l} \\ 1 - \theta_{2,\boldsymbol{\alpha}_l} \end{pmatrix} \circ \cdots \circ \begin{pmatrix} \theta_{J,\boldsymbol{\alpha}_l} \\ 1 - \theta_{J,\boldsymbol{\alpha}_l} \end{pmatrix}, \tag{8}$$

where "$\circ$" denotes the tensor outer product and $\theta_{j,\boldsymbol{\alpha}}$'s are constrained by (2) and (3).

In the following sections, we first investigate the learnability requirement of learning a SLAM with an arbitrary set of true latent patterns, and provide identifiability conditions in Section 3. Then in Section 4, we propose a penalized likelihood method to select the latent attribute patterns, and establish theoretical guarantee for the proposed method.

3. Learnability Requirement and Conditions

To facilitate the discussion on identifiability of SLAMs, we need to introduce a new notation, the Γ-matrix. We first introduce the $J \times 2^K$ constraint matrix Γ^{all} that is entirely determined

by the Q-matrix. The rows of Γ^{all} are indexed by the J items, and columns by the 2^K latent attribute patterns in $\{0,1\}^K$. The $(j,\boldsymbol{\alpha})$th entry of $\Gamma^{\text{all}}_{j,\boldsymbol{\alpha}}$ is defined as

$$\Gamma^{\text{all}}_{j,\boldsymbol{\alpha}} = I(\boldsymbol{\alpha} \succeq \boldsymbol{q}_j) = I(\boldsymbol{\alpha} \in \mathcal{C}_j), \quad j \in \{1,\ldots,J\},\ \boldsymbol{\alpha} \in \{0,1\}^K, \tag{9}$$

which is a binary indicator of whether attribute pattern $\boldsymbol{\alpha}$ possess all the required attributes of item j. We will also call Γ^{all} the *constraint matrix*, since its entries indicate what latent patterns are constrained to have the highest level of Bernoulli parameters for each item. For example, consider the 2×2 Q-matrix in the following (10). Then its corresponding Γ-matrix Γ^{all} with a saturated set of attribute patterns takes the following form.

$$Q = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \implies \Gamma^{\text{all}} = \begin{array}{c} \begin{array}{cccc} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_3 & \boldsymbol{\alpha}_4 \\ (0,0) & (0,1) & (1,0) & (1,1) \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}. \tag{10}$$

More generally, we generalize the definition of the constraint matrix Γ^{all} in (9) to an arbitrary subset of latent patterns $\mathcal{A} \subseteq \{0,1\}^K$, and an arbitrary set of items $S \subseteq [J]$. For $S \subseteq [J]$ and $\mathcal{A} \subseteq \{0,1\}^K$, we simply denote by $\Gamma^{(S,\mathcal{A})}$ the $|S| \times |\mathcal{A}|$ submatrix of Γ^{all} with row indices from S and column indices from $\mathcal{A}$. When $S = \{1,\ldots,J\}$, we will sometimes just denote $\Gamma^{(S,\mathcal{A})}$ by $\Gamma^{\mathcal{A}}$ for simplicity. Then $\Gamma^{\mathcal{A}}$ itself can be viewed as the constraint matrix for a SLAM with attribute pattern space $\mathcal{A}$, and $\Gamma^{\mathcal{A}}$ directly characterizes how the items constrain the positive response probabilities of latent attribute patterns in $\mathcal{A}$.

Given the Q-matrix, we denote by $\mathcal{A}_0 \subseteq \{0,1\}^K$ the set of true attribute patterns existing in the population, i.e., $\mathcal{A}_0 = \{\boldsymbol{\alpha} \in \{0,1\}^K : p_{\boldsymbol{\alpha}} > 0\}$. In knowledge space theory (Düntsch and Gediga, 1995), the set $\mathcal{A}_0$ of patterns corresponds to the *knowledge structure* of the population. We further denote by $\boldsymbol{\Theta}^{\mathcal{A}_0}$ the item parameter matrix respecting the constraints imposed by $\Gamma^{\mathcal{A}_0}$; specifically, $\boldsymbol{\Theta}^{\mathcal{A}_0} = (\theta_{j,\boldsymbol{\alpha}})$ has the same size as $\Gamma^{\mathcal{A}_0}$, with rows and columns indexed by the J items and the attribute patterns in $\mathcal{A}_0$, respectively. For any positive integer $k \le 2^K$, we let $\mathcal{T}^{k-1}$ be the k-dimensional simplex, i.e., $\mathcal{T}^{k-1} = \{(x_1,x_2,\ldots,x_k) : x_i \ge 0,\ \sum_{i=1}^k x_k = 1\}$. We denote the true proportion parameters by $\boldsymbol{p}^{\mathcal{A}_0} = (p_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}_0) \in \mathcal{T}^{|\mathcal{A}_0|-1}$, then $\boldsymbol{p}^{\mathcal{A}_0} \succ \boldsymbol{0}$ by the definition of $\mathcal{A}_0$.

The following toy example illustrates why we need to establish identifiability guarantee for pattern selection.

Example 5 *Consider the 2×2 Q-matrix together with its corresponding 2×4 Γ-matrix in Equation (10). Consider two attribute pattern sets, the true set $\mathcal{A}_0 = \{\boldsymbol{\alpha}_1 = (0,0), \boldsymbol{\alpha}_2 = (0,1)\}$ and an alternative set $\mathcal{A}_1 = \{\boldsymbol{\alpha}_2 = (0,1), \boldsymbol{\alpha}_3 = (1,0)\}$. Under the two-parameter SLAM, for any valid item parameters $\boldsymbol{\Theta}$ restricted by Γ and any proportion parameters $\boldsymbol{p} = (p_{\boldsymbol{\alpha}_1}, p_{\boldsymbol{\alpha}_2}, p_{\boldsymbol{\alpha}_3}, p_{\boldsymbol{\alpha}_4})$ such that $p_{\boldsymbol{\alpha}_1} = p_{\boldsymbol{\alpha}_3}$, we have $\mathbb{P}(\mathbf{R} = \boldsymbol{r} \mid \boldsymbol{\Theta}^{\mathcal{A}_0}, (p_{\boldsymbol{\alpha}_1}, p_{\boldsymbol{\alpha}_2})) = \mathbb{P}(\mathbf{R} = \boldsymbol{r} \mid \boldsymbol{\Theta}^{\mathcal{A}_1}, (p_{\boldsymbol{\alpha}_3}, p_{\boldsymbol{\alpha}_2}))$. This is because $\Gamma^{\mathcal{A}_0} = \Gamma^{\mathcal{A}_1}$ from (10) and hence $\boldsymbol{\Theta}^{\mathcal{A}_0} = \boldsymbol{\Theta}^{\mathcal{A}_1}$; and also $(p_{\boldsymbol{\alpha}_1}, p_{\boldsymbol{\alpha}_2}) = (p_{\boldsymbol{\alpha}_3}, p_{\boldsymbol{\alpha}_2})$ by our construction that $p_{\boldsymbol{\alpha}_1} = p_{\boldsymbol{\alpha}_3}$. This implies even if one knows exactly there are two latent attribute patterns in the population, one can never tell which two patterns those are based on the likelihood function. In this sense, $\mathcal{A}_0$ is not identifiable, due to the fact that $\Gamma^{\mathcal{A}_0}$ and $\Gamma^{\mathcal{A}_1}$ do not lead to distinguishable distributions of responses under the two-parameter SLAM.*

From the above example, to make sure the set of true attribute patterns $\mathcal{A}_0$ is learnable from the observed multivariate responses, we need the $\Gamma^{\mathcal{A}_0}$-matrix to have certain structures. We state the formal definition of (strict) learnability of $\mathcal{A}_0$.

Definition 1 (strict learnability of $\mathcal{A}_0$) *Given Q, the set $\mathcal{A}_0$ is said to be (strictly) learnable, if for any constraint matrix $\Gamma^{\mathcal{A}}$ of size $J \times |\mathcal{A}|$ with $|\mathcal{A}| \le |\mathcal{A}_0|$, any valid item parameters $\boldsymbol{\Theta}^{\mathcal{A}}$ respecting constraints given by $\Gamma^{\mathcal{A}}$, and any proportion parameters $\boldsymbol{p}^{\mathcal{A}} \in \mathcal{T}^{|\mathcal{A}|-1}$, $\boldsymbol{p}^{\mathcal{A}} \succ \boldsymbol{0}$, the following equality*

$$\mathbb{P}(\mathbf{R} \mid \boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0}) = \mathbb{P}(\mathbf{R} \mid \boldsymbol{\Theta}^{\mathcal{A}}, \boldsymbol{p}^{\mathcal{A}}) \tag{11}$$

implies $\mathcal{A} = \mathcal{A}_0$. Moreover, if (11) implies $(\boldsymbol{\Theta}^{\mathcal{A}}, \boldsymbol{p}^{\mathcal{A}}) = (\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$, then we say the model parameters $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ are (strictly) identifiable.

Next we further introduce some notations and definitions about the constraint matrix Γ and then present the needed identifiability result. Consider an arbitrary subset of items $S \subseteq \{1,\ldots,J\}$. For $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{A}$, we denote $\boldsymbol{\alpha} \succeq_S \boldsymbol{\alpha}'$ under $\Gamma^{\mathcal{A}}$, if for each $j \in S$ there is $\Gamma^{\mathcal{A}}_{j,\boldsymbol{\alpha}} \ge \Gamma^{\mathcal{A}}_{j,\boldsymbol{\alpha}'}$. If viewing $\Gamma_{j,\boldsymbol{\alpha}} = 1$ as $\boldsymbol{\alpha}$ being "capable" of item j, then $\boldsymbol{\alpha} \succeq_S \boldsymbol{\alpha}'$ would mean $\boldsymbol{\alpha}$ is at least as capable as $\boldsymbol{\alpha}'$ of items in set S. Then under Γ, any subset of items S defines a partial order "$\succeq_S$" on the set of latent attribute patterns $\mathcal{A}$. For two item sets S_1 and S_2, we say "$\succeq_{S_1}$" = "$\succeq_{S_2}$" under $\Gamma^{\mathcal{A}}$, if for any $\boldsymbol{\alpha}', \boldsymbol{\alpha} \in \mathcal{A}$, there is $\boldsymbol{\alpha} \succeq_{S_1} \boldsymbol{\alpha}'$ under $\Gamma^{\mathcal{A}}$ if and only if $\boldsymbol{\alpha} \succeq_{S_2} \boldsymbol{\alpha}'$ under $\Gamma^{\mathcal{A}}$. The next theorem gives conditions that ensure the constraint matrix Γ as well as the Γ-constrained model parameters are jointly identifiable.

Theorem 2 (conditions for strict learnability) *Consider a SLAM with an arbitrary set of true attribute patterns $\mathcal{A}_0 \subseteq \{0,1\}^K$, and a corresponding constraint matrix $\Gamma^{\mathcal{A}_0}$. If this true $\Gamma^{\mathcal{A}_0}$ satisfies the following conditions, then $\mathcal{A}_0$ is identifiable.*

A. There exist two disjoint item sets S_1 and S_2, such that $\Gamma^{(S_i,\mathcal{A}_0)}$ has distinct column vectors for $i = 1,2$ and "$\succeq_{S_1} = \succeq_{S_2}$" under $\Gamma^{\mathcal{A}_0}$.

B. For any $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}' \in \mathcal{A}_0$ where $\boldsymbol{\alpha}' \succeq_{S_i} \boldsymbol{\alpha}$ under $\Gamma^{\mathcal{A}_0}$ for $i = 1$ or 2, there exists some $j \in (S_1 \cup S_2)^c$ such that $\Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}} \ne \Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}'}$.

C. Any column vector of $\Gamma^{\mathcal{A}_0}$ is different from any column vector of $\Gamma^{\mathcal{A}_0^c}$, where $\mathcal{A}_0^c = \{0,1\}^K \setminus \mathcal{A}_0$.

Recall that each column in the Γ-matrix corresponds to a latent attribute pattern, then Conditions A and B help ensure the Γ-matrix of the true patterns $\Gamma^{\mathcal{A}_0}$ contains enough information to distinguish between these true patterns. Specifically, Condition A requires $\Gamma^{\mathcal{A}_0}$ to contain two vertically stacked submatrices corresponding to item sets S_1 and S_2, each having distinct columns, i.e., each being able to distinguish between the true patterns; and Condition B requires the remaining submatrix of $\Gamma^{\mathcal{A}_0}$ to distinguish those pairs of true patterns that have some order ($\boldsymbol{\alpha}' \succeq_{S_i} \boldsymbol{\alpha}$) based on the first two item sets S_1 or S_2. Condition C is necessary for identifiability of $\mathcal{A}_0$ by ensuring that any true pattern would have a different column vector in Γ^{all} from that of any false pattern. Condition C is satisfied for any $\mathcal{A}_0 \subseteq \{0,1\}^K$ if the Q-matrix contains an identity submatrix I_K, because such a Q-matrix will give a Γ^{all} that has all the 2^K columns distinct.

We would like to point out that our identifiability conditions in Theorem 2 do not depend on the unknown parameters (e.g., $\boldsymbol{\Theta}$ and $\boldsymbol{p}$), but only rely on the structure of the constraint matrix Γ. The Γ-matrix with respect to the true set of patterns $\mathcal{A}_0$ is the key quantity that defines the latent structure of a SLAM. Generally, it is hard to establish identifiability conditions that only depend on the cardinality of $\mathcal{A}_0$ but not on $\Gamma^{\mathcal{A}_0}$. For instance, in Example 5, the two sets $\mathcal{A}_0$ and $\mathcal{A}_1$ have the same cardinality but can not be distinguished under the conditions there; indeed further conditions on Q (and the resulting Γ) are needed to guarantee identifiability.

The developed identifiability conditions generally apply to any SLAM satisfying the constraints (2) and (3) introduced in Section 2.1. If one makes further assumptions on $\boldsymbol{\Theta}$, such as assuming each item $j \in [J]$ has exactly two item parameters to make it a two-parameter model, then the conditions in Theorem 2 may be further relaxed. For example, in the saturated case with $\mathcal{A}_0 = \{0,1\}^K$, the sufficient identifiability conditions developed in Xu (2017) for a general SLAM require Q to contain two copies of I_K as submatrices, while the necessary and sufficient conditions established in Gu and Xu (2019a) for the two-parameter SLAM require Q to have just one submatrix I_K. We expect that in the current case with an arbitrary $\mathcal{A}_0 \subseteq \{0,1\}^K$, the conditions in Theorem 2 can also be relaxed under the two-parameter model in a technically nontrivial way. For the reason of generality, we focus on SLAMs under the general constraints (2) and (3) in this work.

When the conditions in Theorem 2 are satisfied, $\mathcal{A}_0$ is identifiable; and from Theorem 4.1 in Gu and Xu (2019b), the model parameters $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ associated with $\mathcal{A}_0$ are also identifiable.

Corollary 3 *Under the conditions in Theorem 2, the model parameters $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ associated with $\mathcal{A}_0$ are identifiable.*

Note that the result of Theorem 2 differs from the existing works Xu (2017), Xu and Shang (2018) and Gu and Xu (2019b) in that those works assume $\mathcal{A}_0$ is known *a priori* and study the identifiability of $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$, while in the current work $\mathcal{A}_0$ is unknown and we focus on the identifiability of $\mathcal{A}_0$ itself. This is crucially needed in order to guarantee that we can learn the set of true attribute patterns.

Remark 4 *The identifiability results in Theorem 2 and Corollary 3 are related to the uniqueness of tensor decomposition. As shown in (8), the probability mass function of the multivariate responses of each subject can be viewed as a higher order tensor with constraints on entries of the tensor, and unique decomposition of the tensor correspond to identification of the constraint matrix as well as the model parameters. The identifiability conditions in Theorem 2 are weaker than the general conditions for uniqueness of three-way tensor decomposition in Kruskal (1977), which is a celebrated result in the literature. Kruskal's conditions require the tensor can be decomposed as a Khatri-Rao product of three matrices, two having full-rank and the other having Kruskal rank at least two (Kruskal rank of a matrix is the largest number T such that every set of T columns of it are linearly independent). Consider an example with $J = 5$, $K = 2$, $\mathcal{A}_0 = \{\boldsymbol{\alpha}_2 = (0,1),\ \boldsymbol{\alpha}_3 = (1,0)\}$, and the corresponding $\Gamma^{\mathcal{A}_0}$ in the form of (12). Then we can set $S_1 = \{1,2\}$, $S_2 = \{3,4\}$ and Condition A in Theorem 2 is satisfied. Further, Condition B is also satisfied since $\boldsymbol{\alpha}_2 \not\succeq_{S_i} \boldsymbol{\alpha}_3$ and $\boldsymbol{\alpha}_3 \not\succeq_{S_i} \boldsymbol{\alpha}_2$ under $\Gamma^{\mathcal{A}_0}$. Therefore, Theorem 1 guarantees the set $\mathcal{A}_0$ is identifiable, and further guarantees the parameters $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ are identifiable. On the contrary, results based on Kruskal's conditions for unique three-way tensor decomposition can not guarantee identifiability, because other than two full rank structures given by the items in S_1 and S_2, the remaining item 5 in $(S_1 \cup S_2)^c$ corresponds to a structure with Kruskal rank only one.*

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \hline 1 & 0 \\ 0 & 1 \\ \hline 1 & 1 \end{pmatrix} \implies \Gamma^{\mathcal{A}_0} = \begin{array}{c} \begin{array}{cc} \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_3 \\ (0,1) & (1,0) \end{array} \\ \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hline 1 & 0 \\ 0 & 1 \\ \hline 0 & 0 \end{array}\right) \end{array}. \tag{12}$$

We next discuss two extensions of the developed identifiability theory. First, Theorem 2 guarantees the strict learnability of $\mathcal{A}_0$. Under a multi-parameter SLAM, these conditions can be relaxed if the aim is to obtain the so-called generic joint identifiability of $\mathcal{A}_0$, which means that $\mathcal{A}_0$ is learnable with the true model parameters ranging almost everywhere in the constrained parameter space except a set with Lebesgue measure zero. Specifically, we have the following definition.

Definition 5 (generic learnability of the true model) *Denote the parameter space of $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ constrained by $\Gamma^{\mathcal{A}_0}$ by Ω. We say $\mathcal{A}_0$ is generically identifiable, if there exists a subset $\mathcal{V}$ of Ω that has Lebesgue measure zero, such that for any $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0}) \in \Omega \setminus \mathcal{V}$, Equation (11) implies $\mathcal{A} = \mathcal{A}_0$. Moreover, if for any $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0}) \in \Omega \setminus \mathcal{V}$, Equation (11) implies $(\boldsymbol{\Theta}^{\mathcal{A}}, \boldsymbol{p}^{\mathcal{A}}) = (\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$, we say the model parameters $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ are generically identifiable.*

The generic learnability result is presented in the next theorem.

Theorem 6 (conditions for generic learnability) *Consider a multi-parameter SLAM with the set of true attribute patterns $\mathcal{A}_0$ and the $J \times |\mathcal{A}_0|$ constraint matrix $\Gamma^{\mathcal{A}_0}$. If $\Gamma^{\mathcal{A}_0}$ satisfies Condition C and also the following conditions, then $\mathcal{A}_0$ is generically identifiable.*

- $A^\star$. *There exist two disjoint item sets S_1 and S_2, such that altering some entries from 0 to 1 in $\Gamma^{(S_1 \cup S_2, \mathcal{A}_0)}$ can yield a $\widetilde{\Gamma}^{(S_1 \cup S_2, \mathcal{A}_0)}$ satisfying Condition A. That is, $\widetilde{\Gamma}^{(S_i, \mathcal{A}_0)}$ has distinct columns for $i = 1, 2$ and "$\succeq_{S_1}$" = "$\succeq_{S_2}$" under $\widetilde{\Gamma}^{(S_1 \cup S_2, \mathcal{A}_0)}$.*
- $B^\star$. *For any $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}' \in \mathcal{A}_0$ where $\boldsymbol{\alpha}' \succeq_{S_i} \boldsymbol{\alpha}$ under $\widetilde{\Gamma}^{(S_1 \cup S_2, \mathcal{A}_0)}$ for $i = 1$ or 2, there exists some $j \in (S_1 \cup S_2)^c$ such that $\Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}} \neq \Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}'}$.*

We also have the following corollary, where the identifiability requirements are directly characterized by the structure of the Q-matrix, instead of Γ.

Corollary 7 *If the Q-matrix satisfies the following conditions, then for any true set of attribute patterns $\mathcal{A}_0 \subseteq \{0,1\}^K$ such that $\Gamma^{\mathcal{A}_0}$ satisfies Condition C, the set $\mathcal{A}_0$ is generically identifiable.*

($A^{\star\star}$) The Q contains two $K \times K$ sub-matrices Q_1, Q_2, such that for $i = 1, 2$,

$$Q = \begin{pmatrix} Q_1 \\ Q_2 \\ Q' \end{pmatrix}_{J\times K} \quad ; \quad Q_i = \begin{pmatrix} 1 & * & \dots & * \\ * & 1 & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & 1 \end{pmatrix}_{K\times K}, \quad i = 1, 2, \tag{13}$$

where each '$*$' can be either zero or one.

($B^{\star\star}$) With Q in the form of (13), there is $\sum_{j=2K+1}^{J} q_{j,k} \geq 1$ for each $k \in \{1, \dots, K\}$.

Remark 8 *When the conditions in Theorem 7 are satisfied, $\mathcal{A}_0$ is generically identifiable and from Theorem 4.3 in Gu and Xu (2019b), the model parameters $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ are also generically identifiable. Corollary 7 differs from Theorem 4.3 in Gu and Xu (2019b) in that, here we allow the true set of attribute patterns $\mathcal{A}_0$ to be unknown and arbitrary, and study its identifiability, while Gu and Xu (2019b) assumes $\mathcal{A}_0$ is pre-specified and studies the identifiability of the model parameters $(\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$.*

Remark 9 *Under the conditions for generic identifiability in Theorem 6 or Corollary 7, we can obtain the explicit forms of the measure zero set $\mathcal{V}$ ($\mathcal{V} \subseteq \Omega$) where the non-identifiability may occur. Under either Theorem 6 or Corollary 7, the set $\mathcal{V}$ is characterized by the zero set of certain polynomials about the parameters $(\boldsymbol{\Theta}, \boldsymbol{p})$ (see the proofs for details). The zero set of these polynomials indeed defines a lower-dimensional manifold in the parameter space. Therefore, Theorem 6 and Corollary 7 supplement Theorem 2 by relaxing the original conditions and establishing identifiability when $(\boldsymbol{\Theta}, \boldsymbol{p})$ satisfy certain shape constraints, i.e., $(\boldsymbol{\Theta}, \boldsymbol{p})$ do not fall on that manifold $\mathcal{V}$ in the parameter space.*

The above generic identifiability results of $\mathcal{A}_0$ ensure that nonidentifiability happens only in a measure zero set in the parameter space. Next, we develop a second extension of Theorem 2 for scenarios where nonidentifiability cases occupy a positive measure set in the parameter space. This situation happens when certain latent attribute patterns always have the same item parameters across all the items, i.e., $\boldsymbol{\Theta}_{\cdot,\boldsymbol{\alpha}} = \boldsymbol{\Theta}_{\cdot,\boldsymbol{\alpha}'}$ for some $\boldsymbol{\alpha} \neq \boldsymbol{\alpha}'$. We define $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ to be in the same equivalence class if $\boldsymbol{\Theta}_{\cdot,\boldsymbol{\alpha}} = \boldsymbol{\Theta}_{\cdot,\boldsymbol{\alpha}'}$. For instance, still consider the following 2×2 Q-matrix under the two-parameter SLAM introduced in Example 2,

$$Q = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \tag{14}$$

then attribute patterns $\boldsymbol{\alpha}_1 = (0, 0)$ and $\boldsymbol{\alpha}_3 = (1, 0)$ are equivalent under the two-parameter SLAM, as can be seen from the Γ^{all} in (10). Therefore the two latent patterns $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_3$ are not identifiable, no matter which values the true model parameters take.

In this case where both strict and generic identifiability do not hold, we study the $\boldsymbol{p}$-partial identifiability, a concept introduced in Gu and Xu (2019b). Specifically, when some attribute patterns have the same item parameters across all items, we define the set of these attribute patterns as an equivalence class, and aim to identify the proportion of this equivalence class, instead of the separate proportions of these equivalent patterns, in the population. For instance, in the above example in (14), because $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_3$ are

equivalent, there are three equivalence classes: $\{\boldsymbol{\alpha}_1 = (0, 0), \boldsymbol{\alpha}_3 = (1, 0)\}$, $\{\boldsymbol{\alpha}_2 = (0, 1)\}$, and $\{\boldsymbol{\alpha}_4 = (1, 1)\}$. We denote these three equivalence classes by $[\boldsymbol{\alpha}_1]$ (or $[\boldsymbol{\alpha}_3]$, since $[\boldsymbol{\alpha}_1] = [\boldsymbol{\alpha}_3]$), $[\boldsymbol{\alpha}_2]$ and $[\boldsymbol{\alpha}_4]$, since $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_4$ form a complete set of representatives of the equivalence classes. For any Q, we denote the induced set of equivalence classes by $\mathcal{A}^{\text{equiv}} = \{[\boldsymbol{\alpha}_1], \dots, [\boldsymbol{\alpha}_C]\}$, where $\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_C$ form a complete set of representatives of the equivalence classes. In this case, the pattern selection problem of interest is to learn which equivalence classes in $\mathcal{A}^{\text{equiv}}$ are significant.

For the two-parameter SLAM introduced in Example 2, two attribute patterns $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ are in the same equivalence class if and only if $\Gamma^{\mathcal{A}}_{\cdot,\boldsymbol{\alpha}_1} = \Gamma^{\mathcal{A}}_{\cdot,\boldsymbol{\alpha}_2}$. This is because under the two-parameter SLAM, the Γ-matrix determined by the Q-matrix with $\Gamma_{j,\boldsymbol{\alpha}} = I(\boldsymbol{\alpha} \succeq \boldsymbol{q}_j)$ fully captures the model structure in the sense that $\theta_{j,\boldsymbol{\alpha}} = \theta_j^+ \Gamma_{j,\boldsymbol{\alpha}} + \theta_j^-(1 - \Gamma_{j,\boldsymbol{\alpha}})$. Therefore under a two-parameter SLAM, we can obtain a complete set of representatives of the equivalence classes directly from the $\boldsymbol{q}$-vectors, which are

$$\mathcal{A}_Q = \{\vee_{j\in S}\, \boldsymbol{q}_j : S \subseteq \{1, \dots, J\}\}, \tag{15}$$

where $\vee_{j\in S}\, \boldsymbol{q}_j = (\max_{j\in S} q_{j,1}, \dots, \max_{j\in S} q_{j,K})$. For $S = \varnothing$, we define the vector $\vee_{j\in S}\, \boldsymbol{q}_j$ to be $\mathbf{0}_K$, the all-zero attribute pattern. The reasons for $\mathcal{A}_Q$ being a complete set of representatives are that, first, $\Gamma^{\mathcal{A}_Q}$ has distinct columns and contains all the unique column vectors in Γ^{all}; and second, for any other pattern not in $\mathcal{A}_Q$, there is some pattern in $\mathcal{A}_Q$ such that the two patterns have identical column vectors in Γ^{all}. It is not hard to see that $\mathcal{A}_Q = \{0, 1\}^K$ if and only if the Q-matrix contains a submatrix I_K.

For multi-parameter SLAMs introduced in Example 3, two attribute patterns $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ are in the same equivalence class if $\Gamma_{\cdot,\boldsymbol{\alpha}_1} = \Gamma_{\cdot,\boldsymbol{\alpha}_2} = \mathbf{1}$. This can be seen by considering $\Gamma_{\cdot,\boldsymbol{\alpha}_1} = \Gamma_{\cdot,\boldsymbol{\alpha}_1} \neq \mathbf{1}$, i.e., $\Gamma_{j,\boldsymbol{\alpha}_1} = \Gamma_{j,\boldsymbol{\alpha}_2} = 0$ for some item j. Then different from the two-parameter SLAMs, for such item j, the $\theta_{j,\boldsymbol{\alpha}_1}$ and $\theta_{j,\boldsymbol{\alpha}_2}$ are not always the same by the modeling assumptions of multi-parameter SLAMs. Indeed, under a multi-parameter SLAM, for item j, patterns in the set $\mathcal{A}_0 \setminus \mathcal{C}_j$ can have multiple levels of item parameters.

We have the following corollary of Theorem 2 on identifiability, when certain attribute patterns are not distinguishable. Denote the set of significant equivalence classes by $\mathcal{A}_0^{\text{equiv}} = \{[\boldsymbol{\alpha}_{\ell_1}], \dots, [\boldsymbol{\alpha}_{\ell_m}]\}$, which is a subset of the saturated set $\mathcal{A}^{\text{equiv}} = \{[\boldsymbol{\alpha}_1], \dots, [\boldsymbol{\alpha}_C]\}$. Denote the set of representative patterns of the significant equivalence classes by $\{\boldsymbol{\alpha}_{\ell_1}, \dots, \boldsymbol{\alpha}_{\ell_m}\} = \mathcal{A}^{\text{rep}}$.

Corollary 10 *If the matrix $\Gamma^{\mathcal{A}^{\text{rep}}}$ satisfies Conditions A, B and C, $\mathcal{A}_0^{\text{equiv}}$ is identifiable.*

Remark 11 *Under the two-parameter SLAM with $\mathcal{A}^{equiv} = \{[\boldsymbol{\alpha}_1], \dots, [\boldsymbol{\alpha}_C]\}$, the Γ-matrix $\Gamma^{\{\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_C\}}$ by definition would have distinct column vectors. Therefore any column vector of $\Gamma^{\mathcal{A}^{\text{rep}}}$ in Corollary 10 must be different form any column vector of $\Gamma^{\{\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_C\}\setminus\mathcal{A}^{\text{rep}}}$. In this case, Condition C is automatically satisfied. And in order to identify $\mathcal{A}_0^{\text{equiv}}$, one only needs to check if $\Gamma^{\mathcal{A}^{\text{rep}}}$ satisfies Conditions A and B.*

4. Penalized Likelihood approach to pattern selection

In this section, we first present the method of shrinkage estimation, and then describe a screening approach as a preprocessing step.

4.1. Shrinkage Estimation

The developed identifiability conditions guarantee that the true set of patterns can be distinguished from any alternative set that has not more than $|\mathcal{A}_0|$ patterns, since they would lead to different probability mass functions of the responses. As $\mathcal{A}_0 = \{\boldsymbol{\alpha} \in \{0,1\}^K : p_{\boldsymbol{\alpha}} > 0\}$, we know that learning the significant attribute patterns is equivalent to selecting the nonzero elements of the population proportion vector $\boldsymbol{p}$. In practice, if we directly overfit the data with all the 2^K possible attribute patterns, the corresponding maximum likelihood estimator (MLE) can not correctly recover the sparsity structure of the vector $\boldsymbol{p}$. In this case, we propose to impose some regularization on the proportion parameters $\boldsymbol{p}$, and perform pattern selection through maximizing a penalized likelihood function.

In general, we denote by $\mathcal{A}_{\text{input}}$ the set of candidate attribute patterns given to the shrinkage estimation method as input. If the saturated space of all the possible attribute patterns are considered, then $\mathcal{A}_{\text{input}} = \{0,1\}^K$ and it contains all the 2^K possible configurations of attributes. When $2^K \gg N$, we propose to use a preprocessing step that returns a proper subset $\mathcal{A}_{\text{input}}$ of the saturated set $\{0,1\}^K$ as candidate attribute patterns, and then perform the shrinkage estimation (please see Section 4.2 for the preprocessing procedure).

We first introduce the general data likelihood of a structured latent attribute model. Given a sample of size N, we denote the ith subject's response by $\mathbf{R}_i = (R_{i,1}, \ldots, R_{i,J})^\top$, $i = 1, \ldots, N$. We further use $\mathcal{R}$ to denote the $N \times J$ data matrix $(\mathbf{R}_1^\top, \ldots, \mathbf{R}_N^\top)^\top$. The marginal likelihood can be written as

$$L(\boldsymbol{\Theta}, \boldsymbol{p} \mid \mathcal{R}) = \prod_{i=1}^{N} \Big[\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} p_{\boldsymbol{\alpha}} \prod_{j=1}^{J} \theta_{j,\boldsymbol{\alpha}}^{R_{i,j}} (1 - \theta_{j,\boldsymbol{\alpha}})^{1-R_{i,j}} \Big], \tag{16}$$

where the constraints on $\boldsymbol{\Theta}$ imposed by Q are made implicit. We denote the corresponding log likelihood by $\ell(\boldsymbol{\Theta}, \boldsymbol{p}) = \log L(\boldsymbol{\Theta}, \boldsymbol{p} \mid \mathcal{R})$.

As the proportion parameters $\boldsymbol{p}$ belongs to a simplex, in order to encourage sparsity of $\boldsymbol{p}$, we propose to use a log-type penalty with a tuning parameter $\lambda < 0$. Specifically, we use the following penalized likelihood as the objective function,

$$\ell^\lambda(\boldsymbol{\Theta}, \boldsymbol{p}) \ = \ell(\boldsymbol{\Theta}, \boldsymbol{p}) + \lambda \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} \log_{\rho_N}(p_{\boldsymbol{\alpha}}), \quad \lambda \in (-\infty, 0), \tag{17}$$

where $\log_{\rho_N}(p_{\boldsymbol{\alpha}}) = \log(p_{\boldsymbol{\alpha}}) \cdot I(p_{\boldsymbol{\alpha}} > \rho_N) + \log(\rho_N) \cdot I(p_{\boldsymbol{\alpha}} \le \rho_N)$ and ρ_N is a small threshold parameter that is introduced to circumvent the singularity issue of the log function at zero. Specifically, we take

$$\rho_N \asymp N^{-d} \tag{18}$$

for some constant $d \ge 1$, where for two sequences $\{a_N\}$ and $\{b_N\}$, we denote $a_N \lesssim b_N$ if $a_N = O(b_N)$ and $a_N \asymp b_N$ if $a_N \lesssim b_N$ and $b_N \lesssim a_N$. Any attribute pattern $\boldsymbol{\alpha}$ whose estimated $p_{\boldsymbol{\alpha}} < \rho_N$ will be considered as 0, and hence not selected. The tuning parameter $\lambda \in (-\infty, 0)$ controls the sparsity level of the estimated proportion vector $\boldsymbol{p}$, and a smaller λ leads to a sparser solution (with more estimated proportion $p_{\boldsymbol{\alpha}}$ falling below ρ_N). Given a $\lambda \in (-\infty, 0)$, we denote the estimated set of patterns by $\widehat{\mathcal{A}}^\lambda = \{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}} : \widehat{p}_{\boldsymbol{\alpha}} > \rho_N, (\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{p}}) = \arg\max_{\boldsymbol{\Theta}, \boldsymbol{p}} \ell^\lambda(\boldsymbol{\Theta}, \boldsymbol{p})\}$.

Remark 12 *In the literature, Chen et al. (2001) and Chen et al. (2004) used a similar form of penalty as the summation term in our* (17)*, but instead imposed $\lambda > 0$ to avoid sparse solutions of the proportion parameters. These works used that penalty in order to avoid singularity when performing restricted likelihood ratio test. While our goal here is to encourage sparsity of $\boldsymbol{p}$ so that significant attribute patterns can be selected.*

The formulation of (17) *can also be interpreted in a Bayesian way, where the penalty term regarding the proportions $\boldsymbol{p}$ is the logarithm of the Dirichlet prior density with hyperparameter $\beta = \lambda + 1$ over the proportions. But note that when $\beta < 0$, the penalty term is not a proper prior density. Our later Proposition 15 reveals that, under nonstandard convergence rate of the mixture model, the traditional Bayesian way of imposing a proper Dirichlet prior over proportions is not sufficient for selecting significant attribute patterns consistently. Instead, this classical procedure will yield too many false patterns being selected. Therefore, our novelty of allowing λ in* (17) *to be negative with arbitrarily large magnitude is crucial to selection consistency.*

Other than the nice connection to the Dirichlet prior density in the Bayesian literature, the log-type penalty in (17) *also facilitates the computation based on modified EM and variational EM algorithms, as shown in our Algorithms 1 and 2. For such reasons, this work uses the log-type penalty. There are also alternative ways of imposing penalty on the proportion parameters $\boldsymbol{p}$ that would lead to selection consistency, such as the truncated L_1 penalty used in Shen et al. (2012) for high-dimensional feature selection.*

We denote the MLE obtained from directly maximizing $L(\boldsymbol{\Theta}, \boldsymbol{p} \mid \mathcal{R})$ in (16) by $\widehat{\boldsymbol{\Theta}}$ and $\widehat{\boldsymbol{p}}$, and denote the "oracle" MLE of the parameters obtained by maximizing the likelihood constrained to the true set of attribute patterns by $(\widehat{\boldsymbol{\Theta}}^{\mathcal{A}_0}, \widehat{\boldsymbol{p}}^{\mathcal{A}_0})$. We denote the rate of convergence of $\ell(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{p}})$ to $\ell(\widehat{\boldsymbol{\Theta}}^{\mathcal{A}_0}, \widehat{\boldsymbol{p}}^{\mathcal{A}_0})$ by $\delta \in (0, 1]$, that is,

$$\big[\ell(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{p}}) - \ell(\widehat{\boldsymbol{\Theta}}^{\mathcal{A}_0}, \widehat{\boldsymbol{p}}^{\mathcal{A}_0})\big]/N = O_P(N^{-\delta}). \tag{19}$$

When $\delta = 1$, (19) implies $\ell(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{p}})$ converges with the usual root-N rate, and $\delta < 1$ would imply a slower convergence rate. In the literature, Ho and Nguyen (2016) and Heinrich and Kahn (2018) have studied the technically involved problem of convergence rate of the mixing distribution of certain mixture models, and showed these models may not have the standard root-N rate. As implied by these works, for complicated models like SLAMs, the convergence rate of the mixing distribution is likely to be slower than root-N, so as the convergence rate of $\ell(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{p}})$.

For a set $\mathcal{A}$, denote its cardinality by $|\mathcal{A}|$. We have the following theorem.

Theorem 13 (selection consistency) *Suppose the true constraint matrix $\Gamma^{\mathcal{A}_0}$ associated with $\mathcal{A}_0$ satisfies conditions A, B and C in Theorem 2. The true parameters satisfy*

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}_0} p_{\boldsymbol{\alpha}} > c_0; \quad \theta_{j,\boldsymbol{\alpha}^\star} - \max_{\boldsymbol{\alpha}: \Gamma_{j,\boldsymbol{\alpha}} = 0} \theta_{j,\boldsymbol{\alpha}} \ge c_1, \ \forall\ j = 1, \ldots, J \ \textit{and}\ \boldsymbol{\alpha}^\star \in \mathcal{C}_j, \tag{20}$$

where $c_0, c_1 > 0$ are some constants. Assume $\log|\mathcal{A}_{\text{input}}| = o(N)$ and $|\mathcal{A}_{\text{input}}| \cdot \rho_N = O(N^{-\delta})$. Then there exist a sequence of tuning parameters $\{\lambda_N\}$ satisfying $N^{1-\delta}/|\log \rho_N| \lesssim -\lambda_N \lesssim N/|\log \rho_N|$ such that $\mathbb{P}(\widehat{\mathcal{A}}^{\lambda_N} = \mathcal{A}_0) \to 1$ as $N \to \infty$.

Remark 14 *Together with our identifiability result in Theorem 2, the assumption* (20) *helps distinguish the true patterns from any alternative set of patterns with no larger cardinality, and further helps establish selection consistency. It is possible to further extend the current result and relax the constant lower bound assumption, though identifiability conditions would need to be adapted carefully to the case with a growing number of significant patterns and a shrinking magnitude of the proportions; we leave this for future work.*

The proof of Theorem 13 also reveals that if the convergence rate of U_N are slower than $\sqrt{N}$ with $\delta < 1$ in (19), then the tuning parameter λ in (17) has to satisfy $\lambda < -1$ in order to have pattern selection consistency; otherwise the issue of over selecting exists. Under the Bayesian interpretation as discussed in Remark 12, this result implies that imposing the popular Dirichlet prior with a proper hyperparameter $\beta = \lambda + 1 \in (0, 1)$ is not sufficient for consistent selection of the significant mixture components (i.e., latent attribute patterns). Therefore, the approach proposed by Rousseau and Mengersen (2011) would not yield frequentist selection consistency in this considered scenario. We state this in the following proposition.

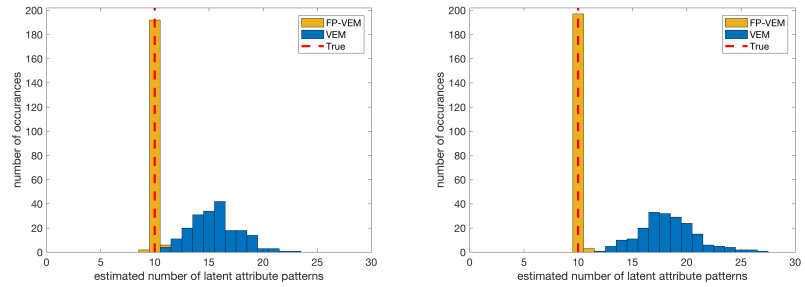
Proposition 15 (selection inconsistency of Dirichlet prior) *Suppose $\delta < 1$ in* (19), *i.e., the rate of convergence of $\ell(\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{p}})$ is slower than the usual $\sqrt{N}$-rate. Then there does not exist a sequence of $\{\lambda_N,\ N = 1, 2, ...\} \subseteq [-1, 0)$ such that $\mathbb{P}(\widehat{\mathcal{A}}^{\lambda_N} = \mathcal{A}_0) \to 1$ as $N \to \infty$.*

Example 6 *To visualize how the numbers of selected patterns differ for our proposed method based on maximizing* (17) *with $\beta = \lambda + 1 \in (-\infty, 1)$, and the variational EM algorithm resulting from imposing a proper Dirichlet prior over the proportions, we conduct a simulation study. In a simulation setting of $K = 10$ and $J = 30$, for each sample size $N = 500$ and* 1000, *we carry out* 200 *independent runs and in each run record the number of selected attribute patterns given by the proposed method, and that by the variational EM algorithm. We plot the histogram corresponding to the proposed method (FP-VEM, see Section 4 for details), together with that corresponding to Variational EM (VEM) with a small Dirichlet parameter $\beta = 0.01$. For both algorithms, we use the same threshold $\rho_N = 1/(2N)$ for selecting attribute patterns in the end of the algorithm, by only keeping patterns whose posterior means exceeds ρ_N. Here we did not plot the results corresponding to VEM with β smaller than* 0.01, *because we found the VEM algorithm with smaller β values can have convergence issues and in many cases it fails to converge but just jumps between several solutions. One can see from Figure 2 that the proposed method selects 10 patterns for most of datasets, which are indeed the 10 true patterns; while VEM over selects the patterns.*

We next propose two algorithms to perform pattern selection, one being a modification of an EM algorithm, and the other being a variational EM algorithm resulting from an alternative formulation of the problem.

4.1.1. Modified EM algorithm.

We first consider using an EM algorithm with a slight modification in the E step to maximize (17). For each subject $i = 1, \ldots, N$, denote his/her latent attribute pattern by $\mathbf{A}_i =$



(a) $K = 10,\ N = 500$ (b) $K = 10,\ N = 1000$

Figure 2: Histograms of estimated number of latent attribute patterns. VEM represents Variational EM with $\beta = \lambda + 1 = 0.01$, and FP-VEM represents the proposed Algorithm 2 in Section 4. The true number of latent attribute patterns is $|\mathcal{A}_0| = 10$.

$(A_{i,1}, \ldots, A_{i,K})$, then $\mathbf{A}_i \in \{0, 1\}^K$. The complete log likelihood corresponding to (17) is

$$\ell^{\lambda}_{\text{comp}}(\boldsymbol{\Theta}, \boldsymbol{p} \mid \mathcal{R}, \mathbf{A}) = \sum_{\boldsymbol{\alpha}_l \in \mathcal{A}_{\text{input}}} \Big(\sum_i I(\mathbf{A}_i = \boldsymbol{\alpha}_l) + \lambda \Big) \log_{\rho_N}(p_{\boldsymbol{\alpha}_l}) \tag{21}$$
$$+ \sum_{\boldsymbol{\alpha}_l \in \mathcal{A}_{\text{input}}} \sum_i I(\mathbf{A}_i = \boldsymbol{\alpha}_l) \sum_j \Big[R_{i,j} \log(\theta_{j,\boldsymbol{\alpha}_l}) + (1 - R_{i,j}) \log(1 - \theta_{j,\boldsymbol{\alpha}_l}) \Big],$$

where $I(\cdot)$ denotes the binary indicator function. Following the standard formulation of the EM algorithm (Dempster et al., 1977), in the E step of the $(t+1)$-th iteration, conditional expectations of $\ell^{\lambda}_{\text{comp}}(\boldsymbol{\Theta}, \boldsymbol{p} \mid \mathcal{R}, \mathbf{A})$ is evaluated with respect to the posterior distribution of latent variables $\mathbf{A}_i$'s given the current iterates of parameters $\boldsymbol{\Theta}^{(t)}$ and $\boldsymbol{p}^{(t)}$. Specifically, in the E step we replace the indicator $I(\mathbf{A}_i = \boldsymbol{\alpha}_l)$ in (21) by the probability $\varphi_{i,l} = \mathbb{P}(\mathbf{A}_i = \boldsymbol{\alpha}_l \mid \boldsymbol{\Theta}^{(t)}, \boldsymbol{p}^{(t)})$; and this is equivalent to updating

$$Q(\boldsymbol{\Theta}, \boldsymbol{p} \mid \boldsymbol{\Theta}^{(t)}, \boldsymbol{p}^{(t)}) := \mathbb{E}\Big[\ell^{\lambda}_{\text{comp}}(\boldsymbol{\Theta}, \boldsymbol{p} \mid \mathcal{R}, \mathbf{A}) \ \Big|\ \boldsymbol{\Theta}^{(t)}, \boldsymbol{p}^{(t)} \Big].$$

In the M step, we update $(\boldsymbol{\Theta}^{(t+1)}, \boldsymbol{p}^{(t+1)}) = \arg\max Q(\boldsymbol{\Theta}, \boldsymbol{p} \mid \boldsymbol{\Theta}^{(t)}, \boldsymbol{p}^{(t)})$. Note that directly using a negative λ in the EM algorithm may yield an invalid E step, due to potentially negative updates for some proportion parameters (e.g., $p_{\boldsymbol{\alpha}}$'s). When this happens, we do a thresholding in the E step as an approximation by replacing the probably negative class potential (Δ_l in Algorithm 1) with a pre-specified small constant $c > 0$. In practice, Algorithm 1's performance appears not sensitive to small values of c, and we take $c = 0.01$ in our numerical experiments; see Appendix B for a sensitivity study of the parameter c.

Remark 16 *Under the two-parameter SLAM, the DINA model in Example 2, or the identity-link multi-parameter all-effect SLAM, the GDINA model in Example 3, the M-step of updating the item parameters $\{\theta_{j,\boldsymbol{\alpha}}\}$'s in Algorithm 1 has closed forms. Specifically, under*

Algorithm 1: PEM: Penalized EM for log-penalty with $\lambda \in (-\infty, 0)$

Data: Q, responses $\mathcal{R}$, and candidate attribute patterns $\mathcal{A}_{\text{input}}$.
Initialize $\boldsymbol{\Delta} = (\Delta_1^{(0)}, \ldots, \Delta_{|\mathcal{A}_{\text{input}}|}^{(0)})$.
while *not converged* **do**
  In the $(t+1)$th iteration,
  for $(i,l) \in [N] \times [|\mathcal{A}_{\text{input}}|]$ **do**

$$\varphi_{i,\boldsymbol{\alpha}_l}^{(t+1)} = \frac{\Delta_l^{(t)} \cdot \exp\left\{\sum_j \left[R_{i,j}\log(\theta_{j,\boldsymbol{\alpha}_l}^{(t)}) + (1-R_{i,j})\log(1-\theta_{j,\boldsymbol{\alpha}_l}^{(t)})\right]\right\}}{\sum_m \Delta_m^{(t)} \cdot \exp\left\{\sum_j \left[R_{i,j}\log(\theta_{j,\boldsymbol{\alpha}_m}^{(t)}) + (1-R_{i,j})\log(1-\theta_{j,\boldsymbol{\alpha}_m}^{(t)})\right]\right\}};$$

  for $l \in [|\mathcal{A}_{\text{input}}|]$ **do**
    $\Delta_l^{(t+1)} = \max\{c,\ \lambda + \sum_{i=1}^N \varphi_{i,\boldsymbol{\alpha}_l}^{(t+1)}\}$; $(c > 0$ is pre-specified$)$;
  $\boldsymbol{p}^{(t+1)} \leftarrow \boldsymbol{\Delta}^{(t+1)}/(\sum_l \Delta_l^{(t+1)})$;
  for $j \in [J]$ **do**
    $\boldsymbol{\Theta}^{(t+1)} = \arg\max_{\boldsymbol{\Theta}} \left\{\sum_{\boldsymbol{\alpha}_l}\sum_i \varphi_{i,\boldsymbol{\alpha}_l}^{(t+1)} \sum_j \left[R_{i,j}\log(\theta_{j,\boldsymbol{\alpha}_l}) + (1-R_{i,j})\log(1-\theta_{j,\boldsymbol{\alpha}_l})\right]\right\}$;
After the total T iterations,
Output: $\{\boldsymbol{\alpha}_l \in \mathcal{A}_{\text{input}} :\ p_{\boldsymbol{\alpha}_l}^{(T)} > \rho_N\}$.

DINA, for any item j the update for the unique parameters (θ_j^+, θ_j^-) takes the form

$$(\theta_j^+)^{(t+1)} = \frac{\sum_i \sum_{\boldsymbol{\alpha}} R_{i,j}\Gamma_{j,\boldsymbol{\alpha}}\varphi_{i,\boldsymbol{\alpha}}^{(t+1)}}{\sum_i \sum_{\boldsymbol{\alpha}} \Gamma_{j,\boldsymbol{\alpha}}\varphi_{i,\boldsymbol{\alpha}}^{(t+1)}}, \quad (\theta_j^-)^{(t+1)} = \frac{\sum_i \sum_{\boldsymbol{\alpha}} R_{i,j}(1-\Gamma_{j,\boldsymbol{\alpha}})\varphi_{i,\boldsymbol{\alpha}}^{(t+1)}}{\sum_i \sum_{\boldsymbol{\alpha}} (1-\Gamma_{j,\boldsymbol{\alpha}})\varphi_{i,\boldsymbol{\alpha}}^{(t+1)}}.$$

Under GDINA, for item j, the update for the unique parameters $\theta_{j,\{k_1,\ldots,k_l\}}$ with $\{k_1,\ldots,k_l\} \subseteq \mathcal{K}_j$ takes the following form,

$$\theta_{j,\{k_1,\ldots,k_l\}}^{(t+1)} = \frac{\sum_i \sum_{\boldsymbol{\alpha}} I(\{k \in \mathcal{K}_j :\ \alpha_k = 1\} = \{k_1,\ldots,k_l\}) R_{i,j}\varphi_{i,\boldsymbol{\alpha}}^{(t+1)}}{\sum_i \sum_{\boldsymbol{\alpha}} I(\{k \in \mathcal{K}_j :\ \alpha_k = 1\} = \{k_1,\ldots,k_l\})\varphi_{i,\boldsymbol{\alpha}}^{(t+1)}}.$$

In addition, when certain latent patterns are not distinguishable as discussed earlier in Corollary 10, we can easily modify Algorithm 1 from selecting attribute patterns to selecting equivalence classes of attribute patterns. For instance, under a two-parameter SLAM, given the row vectors $\{\boldsymbol{q}_j, j \in [J]\}$ of Q, we first obtain the representatives of the Q-induced equivalence classes: $\mathcal{A}_Q = \{\vee_{j \in S}\boldsymbol{q}_j : S \subseteq \{1,\ldots,J\}\}$, then get the ideal response matrix of $\mathcal{A}_Q$, namely $\Gamma(\cdot, \mathcal{A}_Q) = (\gamma_{j,l})_{J \times |\mathcal{A}_Q|}$ where $\gamma_{j,l} = I(\boldsymbol{\alpha}_l \succeq \boldsymbol{q}_j)$ for $\boldsymbol{\alpha}_l \in \mathcal{A}_Q$ and $j \in [J]$. After initializing $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_{|\mathcal{A}_Q|})$, we just follow the same iterative procedure as that of Algorithm 1 for the two-parameter SLAM. In the end of the algorithm, after calculating $\nu_{[\boldsymbol{\alpha}_l]} = \Delta_l/(\sum_m \Delta_m)$, we select those $[\boldsymbol{\alpha}_l]$ with proportion $\nu_{[\boldsymbol{\alpha}_l]}$ above a pre-specified threshold. From the selected equivalence classes of attribute profiles, we can go back to obtain their representatives which are combinations of the $\boldsymbol{q}$-vectors from $\mathcal{A}_Q$ defined in Equation (15).

In practice when applying the PEM algorithm, we recommend using a sequential procedure with a range of λ values $\lambda_1 > \lambda_2 > \cdots > \lambda_B$, where $\lambda_1 > -1$ is close to 0 and λ_B should be less than -1. Specifically, we start with the relatively large λ_1 and use the estimated parameters from PEM with λ_1 as initial values for the next round of PEM with λ_2. We do this sequentially with estimates from PEM with λ_b serving as initializations for PEM with λ_{b+1}. When this sequential procedure ends, we choose the final model from the total number of B estimated ones using certain information criterion.

Given the large model space, we propose to use the Extended Bayesian Information Criterion (EBIC) introduced in Chen and Chen (2008) to select the tuning parameter. Recall that we denote by $\mathcal{A}^\lambda$ the selected set of attribute patterns obtained by maximizing the penalized likelihood function (17) with the specific tuning parameter λ. And we denote the item parameters and proportion parameters defined on this $\mathcal{A}^\lambda$ by $\boldsymbol{\Theta}^{\mathcal{A}^\lambda}$ and $\boldsymbol{p}^{\mathcal{A}^\lambda}$, respectively. The EBIC family have the following information criterion

$$\text{BIC}_\gamma(\mathcal{A}^\lambda) = -2\ell(\boldsymbol{\Theta}^{\mathcal{A}^\lambda}, \boldsymbol{p}^{\mathcal{A}^\lambda}) + |\mathcal{A}^\lambda|\log N + 2\gamma\log\binom{|\mathcal{A}_{\text{input}}|}{|\mathcal{A}^\lambda|},$$

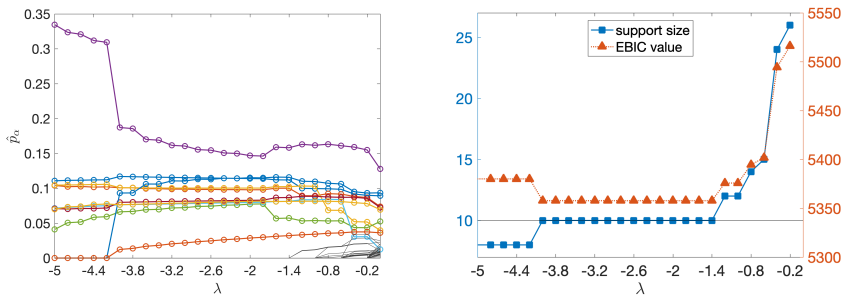
with the EBIC parameter $\gamma \in [0,1]$. A smaller EBIC value implies a more favorable model. Selection consistency of the EBIC for high-dimensional model is established in Theorem 1 of Chen and Chen (2008) for γ greater than a certain threshold. When $\gamma = 0$, EBIC becomes the the classical BIC. Generally, larger γ yields a more parsimonious model. Here we choose $\gamma = 1$, for which the condition in Theorem 1 for selection consistency in Chen and Chen (2008) is satisfied.

Example 7 *Figure 3 presents an illustration of the solution paths of the estimated proportions versus λ based on a simulated dataset with $N = 150$, $K = 10$, and $J = 30$. The Q-matrix $Q = (Q_1^\top, Q_2^\top, Q_3^\top)^\top$ with Q_i in the following form,*

$$Q_1 = \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & 1 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 1 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 1 \end{pmatrix}, \quad Q_3 = \begin{pmatrix} 1 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 1 \end{pmatrix}. \tag{22}$$

When generating the data, 10 attribute patterns are randomly selected from the $2^{10} = 1024$ possible ones as true patterns, and the proportion of each of them is set to be 0.1. The item parameters are set to $1 - \theta_j^+ = \theta_j^- = 0.2$ for each j under a two-parameter SLAM. In the current setting with $K = 10$, we take the set of patterns as input to the PEM algorithm to be $\mathcal{A}_{\text{input}} = \{0,1\}^K$. Figure 3(a) plots the solution paths of the estimated proportions of all the $2^{10} = 1024$ attribute patterns as λ varies in $\{-0.2, -0.4, \cdots, -4.8, -5.0\}$. The 10 true attribute patterns are plotted with colored lines with circles while the remaining $2^{10} - 10$ attribute patterns are plotted with black solid lines. Figure 3(b) plots the estimated support size of $\boldsymbol{p}$ versus λ, and the EBIC value versus λ. We observe that when $\lambda \in [-4.4, -1.4]$, Algorithm 1 selects the correct model with 10 true attribute patterns. This interval of λ corresponds to a "stable window" of the estimation algorithm that gives the correct selection and also has the smallest EBIC value. For this specific dataset, the proposed method along with EBIC succeeds in selecting the true model. Please see Section 5 for more

simulation results which show that the proposed methods combined with EBIC indeed have good performance in general.



(a) solution paths versus λ (b) EBIC values and support sizes versus λ

Figure 3: PEM solution paths and EBIC values in one trial, $N = 150$.

4.1.2. Variational EM algorithm from an alternative formulation.

In the following, we discuss an alternative formulation of the objective function (17) and propose a variational EM algorithm for estimation, by treating the proportion parameters $\boldsymbol{p}$ as latent random variables. As discussed in Remark 12, for the objective function (17) with $\lambda \in (-\infty, -1]$, the penalty term $\prod_{l=1}^{2^K} p_{\boldsymbol{\alpha}_l}^{\lambda}$ does not correspond to a proper Dirichlet distribution density. However, for any arbitrarily small λ value, the objective function (17) can be replaced by the following alternative formulation:

$$\ell_{\text{pseudo}}^{\lambda,\Upsilon}(\boldsymbol{\Theta}, \boldsymbol{p}) = \Upsilon \cdot \ell(\boldsymbol{\Theta}, \boldsymbol{p}) + (\beta - 1) \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} \log_{\rho_N}(p_{\boldsymbol{\alpha}}) \quad \text{for } \beta \in (0,1),\ \Upsilon \in (0,1]. \tag{23}$$

where we introduce a new parameter $\Upsilon \in (0,1]$ and replace λ with $\beta - 1$ to respect the convectional notation of a Dirichlet distribution with hyperparameter $\beta \in (0,1)$ to encourage sparsity. With $\beta \in (0,1)$ and $\Upsilon \in (0,1]$, the ratio $(1-\beta)/\Upsilon$ can be arbitrarily large when Υ is arbitrarily close to zero, therefore making (23) equivalent to (17).

In the new objective function (23), the penalty term $\prod_{l=1}^{2^K} p_{\boldsymbol{\alpha}_l}^{\beta-1}$, $\beta \in (0,1)$, can be viewed as a well-defined Dirichlet density function for the latent variables $\boldsymbol{p}$. In (23), the first term is the logarithm of the likelihood function raised to a fractional power $\Upsilon \in (0,1]$. One intuition behind (23) is that given a moderate sample size and a large number of potential latent patterns, one needs to downweight the influence of the data likelihood and magnify the prior information encoded by the Dirichlet prior, in order to have the sufficient extent of shrinkage. The fractional-powered likelihood multiplied by the Dirichlet density can then be treated as a loss function to minimize. The idea of assigning a fractional power to the likelihood was also used in the Bayesian literature, such as Bissiri et al.

(2016) and Holmes and Walker (2017) for Bayesian learning under model misspecification, and Yang et al. (2019) and Chérief-Abdellatif and Alquier (2018) for variational Bayesian inference. Different from these works, here we use the alternative formulation (23) of the original objective function (17) in order to consistently select the significant latent attribute patterns.

The formulation (23) allows for a variational EM algorithm for obtaining the item parameters $\boldsymbol{\Theta}$ and the posterior means of the latent variables $\boldsymbol{p}$. Here we treat $\boldsymbol{\Theta}$ still as model parameters, then we follow the general derivation of variational algorithms in Blei et al. (2017) to derive Algorithm 2. We denote the digamma function by $\Psi(x) = \frac{d}{dx}\log\Gamma(x)$ for $x \in (0,\infty)$. In particular, the complete log likelihood is

$$\begin{aligned}\ell_{\text{comp}}^{\lambda,\Upsilon}(\boldsymbol{\Theta} \mid \mathcal{R}, \mathbf{A}, \boldsymbol{p}) = &\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} \Big\{\Upsilon \cdot \Big[\sum_i I(\mathbf{A}_i = \boldsymbol{\alpha})\Big] + \beta - 1\Big\} \log_{\rho_N}(p_{\boldsymbol{\alpha}}) \\ &+ \Upsilon \cdot \Big\{\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} \sum_i I(\mathbf{A}_i = \boldsymbol{\alpha}) \sum_j \Big[R_{i,j}\log(\theta_{j,\boldsymbol{\alpha}}) + (1 - R_{i,j})\log(1 - \theta_{j,\boldsymbol{\alpha}})\Big]\Big\}.\end{aligned} \tag{24}$$

In the variational E step, we first obtain the conditional probability of $I(\mathbf{A}_i = \boldsymbol{\alpha}_l)$ for each individual i and each input attribute pattern $\boldsymbol{\alpha}_l$, which we denote by $\varphi_{i,\boldsymbol{\alpha}_l}$. In updating this $\varphi_{i,\boldsymbol{\alpha}_l}$, the variational posterior distribution of the $p_{\boldsymbol{\alpha}}$'s are used, which is still a Dirichlet distribution with mean parameters $(\Delta_1, \ldots, \Delta_{|\mathcal{A}_{\text{input}}|})$ updated in the previous E step (or from initializations if in the first iteration). Then we update the mean parameters for the variational posterior distribution of $p_{\boldsymbol{\alpha}_l}$'s based on the obtained $\varphi_{i,\boldsymbol{\alpha}_l}$, following the conventional derivation in variational inference. After finishing this E step, in the M step we maximize the complete likelihood with respect to $\boldsymbol{\Theta}$, by substituting the $I(\mathbf{A}_i = \boldsymbol{\alpha}_l)$'s with $\varphi_{i,\boldsymbol{\alpha}_l}$'s. Note that taking the derivatives of (24) with respect to $\theta_{j,\boldsymbol{\alpha}_l}$'s does not involve either terms of $p_{\boldsymbol{\alpha}_l}$ or terms of Υ and β, so only $\varphi_{i,\boldsymbol{\alpha}_l}$ are used in the M step for updating $\boldsymbol{\Theta}$. Indeed, the M step of updating $\boldsymbol{\Theta}$ in the current Algorithm 2 takes the same form as that of Algorithm 1.

Similar to Algorithm 1, in the practical use of Algorithm 2 for pattern selection, we recommend using a sequential fitting procedure. For a small fixed $\beta > 0$, we choose a sequence of Υ values $1 > \Upsilon_1 > \Upsilon_2 > \cdots > \Upsilon_B > 0$ where Υ_1 should be close to 1 and Υ_B should be relatively small. In our simulation studies, we found a $\Upsilon_B = 0.3$ is sufficient in most of cases. Then we sequentially run Algorithm 2 for B times with fractional powers $\Upsilon_1, \ldots, \Upsilon_B$ respectively and use estimated parameters from FP-VEM with Υ_b as initial values for FP-VEM with Υ_{b+1}. In the end, we also use EBIC to select the best Υ. Since β and Υ can be viewed as acting together through the term $(1-\beta)/\Upsilon$, in terms of practical parameter tuning, we recommend fixing β to a relatively small value, say $\beta = 0.01$, and let the fractional power $\Upsilon \in (0,1]$ vary to control the sparsity level of the proportion parameters.

4.2. Screening as a preprocessing step when $2^K \gg N$

In many applications of SLAMs, the number of attribute patterns 2^K could be much larger than N. This is especially the case in the application of SLAMs in epidemiological and medical diagnosis (Wu et al., 2017, 2018). In such scenarios, given a sample with size of several thousands or hundreds, it is desirable to develop an efficient screening procedure to

Algorithm 2: FP-VEM: Fractional Power Variational EM for $\Upsilon \in (0,1]$

Data: Q, $\mathcal{R}$, and candidate attribute patterns $\mathcal{A}_{\text{input}}$.
Initialize $\boldsymbol{\Delta} = (\Delta_1^{(0)}, \ldots, \Delta_{|\mathcal{A}_{\text{input}}|}^{(0)}) = (\beta, \ldots, \beta)$.
while *not converged* **do**
  In the $(t+1)$th iteration,
  for $(i,l) \in [N] \times [|\mathcal{A}_{\text{input}}|]$ **do**

$$\varphi_{i,\boldsymbol{\alpha}_l}^{(t+1)} = \frac{\exp\left\{\Psi(\Delta_l^{(t)}) + \Upsilon \cdot \sum_j \left[R_{i,j}\log(\theta_{j,\boldsymbol{\alpha}_l}^{(t)}) + (1-R_{i,j})\log(1-\theta_{j,\boldsymbol{\alpha}_l}^{(t)})\right]\right\}}{\sum_m \exp\left\{\Psi(\Delta_m^{(t)}) + \Upsilon \cdot \sum_j \left[R_{i,j}\log(\theta_{j,\boldsymbol{\alpha}_m}^{(t)}) + (1-R_{i,j})\log(1-\theta_{j,\boldsymbol{\alpha}_m}^{(t)})\right]\right\}};$$

  for $l \in [|\mathcal{A}_{\text{input}}|]$ **do**
    $\Delta_l^{(t+1)} \leftarrow \beta + \Upsilon \times \sum_{i=1}^N \varphi_{i,l}^{(t+1)}$;
  for $j \in [J]$ **do**
    $\boldsymbol{\Theta}^{(t+1)} = \arg\max_{\boldsymbol{\Theta}} \left\{\sum_{\boldsymbol{\alpha}_l}\sum_i \varphi_{i,\boldsymbol{\alpha}_l}^{(t+1)} \sum_j \left[R_{i,j}\log(\theta_{j,\boldsymbol{\alpha}_l}) + (1-R_{i,j})\log(1-\theta_{j,\boldsymbol{\alpha}_l})\right]\right\}$
After the total T iterations,
for $\boldsymbol{\alpha}_l \in \mathcal{A}_{\text{input}}$ **do**
  $p_{\boldsymbol{\alpha}_l} \leftarrow \Delta_l^{(T)}/(\sum_m \Delta_m^{(T)})$.
output: $\{\boldsymbol{\alpha}_l \in \mathcal{A}_{\text{input}} : \ p_{\boldsymbol{\alpha}_l} > \rho_N\}$.

bring down the number of candidate attribute patterns, and then perform the shrinkage estimation.

We next describe our screening approach. Recall that for each subject $i = 1, \ldots, N$, we denote his/her latent attribute pattern by $\mathbf{A}_i = (A_{i,1}, \ldots, A_{i,K}) \in \{0,1\}^K$. In the screening stage we jointly estimate the item parameters $\boldsymbol{\Theta}$ and the $\{\mathbf{A}_i, i \in [N]\}$ to get a rough estimation of each subject i's attribute pattern, and gather all the N estimated attribute profiles as candidate patterns. The estimation of $\boldsymbol{p}$ is postponed to the estimation stage. Under the basic two-parameter SLAM, the complete log likelihood involving the latent variables $\{\mathbf{A}_i, i \in [N]\}$ takes the form

$$\begin{aligned}\ell_{\text{complete}}(\boldsymbol{\Theta}, \mathbf{A}) = &\sum_{i=1}^N \sum_{j=1}^J \Big[R_{i,j}\Big(\prod_k A_{i,k}^{q_{j,k}} \log\theta_j^+ + (1 - \prod_k A_{i,k}^{q_{j,k}})\log\theta_j^-\Big)\Big] \\ &+ (1-R_{i,j})\Big(\prod_k A_{i,k}^{q_{j,k}} \log(1-\theta_j^+) + (1 - \prod_k A_{i,k}^{q_{j,k}})\log(1-\theta_j^-)\Big)\Big].\end{aligned}$$

We next derive an algorithm with a stochastic EM flavor to estimate the posterior mean of each latent variable $A_{i,k}$, denoted by a matrix $(\widehat{a}_{i,k})$ of size $N \times K$, where $\widehat{a}_{i,k} = \mathbb{E}[A_{i,k} \mid \cdot]$. In the end of the algorithm, we obtain the binary matrix W containing the candidate attribute patterns by defining $W = (w_{i,k})_{N\times K}$ with $w_{i,k} = I(\widehat{a}_{i,k} > 1/2)$. In such a screening procedure, we first use the dependency among the K attributes in iterative updates, then partly ignore the dependency in the last step through applying Bayes' rule to each subject

i's each single attribute k. This results in fast and valid screening of attribute patterns. Viewing the ith row vector of W as the estimated attribute pattern of subject i, the unique row vectors in W are the roughly selected attribute patterns output by the screening stage. We denote this set of candidate patterns by $\widehat{\mathcal{A}}_{\text{screen}}$. As long as the screening has the nice property of "no false exclusion", meaning the rows in W contain all the true attribute patterns, then the screening stage is considered successful. The selected candidate patterns are passed along to the shrinkage estimation stage as input patterns.

We say the screening procedure has the *sure screening property* if as N goes to infinity, the probability of all the true attribute patterns included in $\widehat{\mathcal{A}}_{\text{screen}}$ goes to one. The next theorem establishes the sure screening property of the proposed screening procedure.

Theorem 17 (sure screening property) *Suppose the identifiability conditions in Theorem 2 and the constraints* (20) *are satisfied. The screening procedure applied to a SLAM that covers the two-parameter SLAM as a submodel has the sure screening property. Specifically, there exists a constant $\beta_{\min} > 0$ such that $\mathbb{P}(\widehat{\mathcal{A}}_{\text{screen}} \supseteq \mathcal{A}_0) \geq 1 - |\mathcal{A}_0|\exp(-N\beta_{\min}) \to 1$ as $N \to \infty$.*

Theorem 17 shows that the probability of the screening procedure failing to include all true patterns has an exponential decay with the sample size N. We point out that despite having the nice property of sure screening, the screening procedure does not guarantee consistency in selecting exactly the set $\mathcal{A}_0$ of true patterns, if the number of observed variables per subject J is not large enough. Generally speaking, as N goes large but J does not, the set $\widehat{\mathcal{A}}_{\text{screen}}$ will include many false attribute patterns, although it will contain the true set $\mathcal{A}_0$ with probability tending to one. Therefore the shrinkage estimation approach in Section 4.1 is still essential to performing pattern selection.

In Algorithm 3, we present the proposed screening algorithm with stochastic approximations based on a number of M_{eff} Gibbs samples of $\mathbf{A}$ in the E step. Alternatively, we can also use an even faster screening procedure by just updating the conditional probability of each subject possessing each attribute (i.e., the conditional posterior mean of each $A_{i,k}$) in each E step, conditioning on everything else; we term this alternative procedure the variational screening procedure. As stated before, the screening algorithm is derived based on the log-likelihood of the two-parameter SLAM, but can be applied to a multi-parameter SLAM that covers the two-parameter SLAM as a submodel. After the screening stage, the set of attribute patterns as input to the shrinkage Algorithms 1 or 2 is taken as $\mathcal{A}_{\text{input}} = \widehat{\mathcal{A}}_{\text{screen}}$. Screening drastically lowers down the computational cost of the subsequent shrinkage estimation, and the number of candidate patterns fed to the shrinkage stage is kept at the order of N, even if the original number of possible configurations $2^K \gg N$.

Remark 18 *The screening algorithm can be modified to be more conservative in order to reduce the risk of excluding true patterns. In particular, after each stochastic E step in the screening algorithm, based on the current iterate of $\mathbf{A}^{\text{ave}}$ we can obtain a $N \times K$ binary matrix with the (i,k)th entry being $I(A_{i,k}^{\text{ave}}) > 1/2$. The unique row vectors of this intermidiate binary matrix can be viewed as the current candidate latent patterns. To make the screening procedure more conservative, we recommend saving this set of candidate patterns after every M stochastic EM iterations (M is a positive integer), and take the union of these saved sets in the end of the algorithm to form $\widehat{\mathcal{A}}_{screen}$ as the output. We call this*

Algorithm 3: Stochastic Approximation Gibbs Screening

Data: Q, $\mathbf{R}$
Result: Candidate attribute patterns $\widehat{\mathcal{A}}_{\text{screen}}$.
Initialize latent attribute patterns $\mathbf{A} = (A_{i,k})_{N\times K} \in \{0,1\}^{N\times K}$, and $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$.
Set $t = 1$, $\mathbf{A}^{\text{ave}} = \mathbf{0}$, $\mathbf{I}^{\text{ave}} = \mathbf{0}$.
while *not converged* **do**
  $\mathbf{A}^{\text{s}} \leftarrow \mathbf{0}$, $\mathbf{I}^{\text{s}} \leftarrow \mathbf{0}$, $M_{\text{eff}} \leftarrow 0$.
  for $r \in [M_{\max}]$ **do**
    for $(i,k) \in [N]\times[K]$ **do**
      Draw $A_{i,k} \sim$
      $\text{Bernoulli}\Big(\text{logit}^{-1}\Big(\sum_j q_{j,k}\prod_{m\neq k} A_{i,m}^{q_{j,m}}\Big[R_{i,j}\log\frac{\theta_j^+}{\theta_j^-} + (1-R_{i,j})\log\frac{1-\theta_j^+}{1-\theta_j^-}\Big]\Big)\Big)$.
    if $r \geq M_{\max} - M_{\text{eff}}$ **then**
      $\mathbf{A}^{\text{s}} \leftarrow \mathbf{A}^{\text{s}} + \mathbf{A}$, $\mathbf{I}^{\text{s}} \leftarrow \mathbf{I}^{\text{s}} + \Big(\prod_k A_{i,k}^{q_{j,k}}\Big)_{N\times J}$.
  $\mathbf{A}^{\text{ave}} \leftarrow \frac{1}{t}\mathbf{A}^{\text{s}}/M_{\text{eff}} + \Big(1-\frac{1}{t}\Big)\mathbf{A}^{\text{ave}}$, $\mathbf{I}^{\text{ave}} \leftarrow \frac{1}{t}\mathbf{I}^{\text{s}}/M_{\text{eff}} + \Big(1-\frac{1}{t}\Big)\mathbf{I}^{\text{ave}}$, $t = t+1$.
  for $j \in [J]$ **do**
    $\theta_j^+ \leftarrow (\sum_i R_{i,j} I_{i,j}^{\text{ave}})/(\sum_i I_{i,j}^{\text{ave}})$, $\theta_j^- \leftarrow (\sum_i R_{i,j}(1-I_{i,j}^{\text{ave}}))(\sum_i (1-I_{i,j}^{\text{ave}}))$.
for $(i,k) \in [N]\times[K]$ **do**
  $w_{i,k} \leftarrow I(A_{i,k}^{\text{ave}} > \frac{1}{2})$.
Output: include all the unique row vectors of W in the set $\widehat{\mathcal{A}}_{\text{screen}}$.

strategy "screening enhanced by Gibbs exploration", since it takes advantage of the latent patterns that the Gibbs sampling explores along the stochastic EM iterations.

5. Simulation Studies

We next present simulation results with the two-parameter SLAM and the multi-parameter all-effect SLAM, respectively.

Two-parameter SLAM. Consider the two-parameter SLAM with a $3K \times K$ Q-matrix $Q = (Q_1^\top, Q_2^\top, Q_3^\top)^\top$, where the three submatrices Q_1, Q_2 and Q_3 are specified in (22). We consider three dimensions of possible attribute patterns with $2^K = 2^{10}, 2^{15}$, and 2^{20}, three sample sizes with $N = 150, 500$ and 1000, and two different signal levels with true item parameters: $\{\theta_j^+ = 0.8,\ \theta_j^- = 0.2;\ j \in [J]\}$, the relatively weak signals; and $\{\theta_j^+ = 0.9,\ \theta_j^- = 0.1;\ j \in [J]\}$, the relatively strong signals. We randomly generate the set of true attribute patterns $\mathcal{A}_0 \subseteq \{0,1\}^K$ with cardinality $|\mathcal{A}_0| = 10$ and set $p_{\boldsymbol{\alpha}} = 0.1$ for all $\boldsymbol{\alpha} \in \mathcal{A}_0$. In the simulations, for $K = 10$ the $\mathcal{A}_{\text{input}}$ is taken to be $\{0,1\}^K$; while for $K = 15$ and 20, the $\mathcal{A}_{\text{input}}$ is taken to be $\widehat{\mathcal{A}}_{\text{screen}}$, i.e., the set of candidate patterns output by the screening method.

In each scenario we perform 200 independent replications. For shrinkage estimation, we apply the proposed Algorithm 1 "Penalized EM (PEM)" and Algorithm 2 "Fractional Power

Variational EM (FP-VEM)", and also apply the plain EM algorithm with thresholding for comparison. When running PEM we compute a solution path by varying λ in the range of $\lambda \in \{-0.2, -0.4, \ldots, -3.8, -4.0\}$, and select the λ that gives the smallest EBIC. When running FP-VEM we fix $\beta = \lambda + 1 = 0.01$ and compute a solution path by varying Υ in $\{1.0, 0.9, \ldots, 0.4, 0.3\}$ and also select Υ using EBIC. We use the threshold value $\rho_N = 1/(2N)$ for the estimated proportions in the last step for all three shrinkage algorithms to select patterns (other smaller ρ_N values give similar results).

signal strength	2^K	N	1−FDR			TPR		
			EM	Algo. 1	Algo. 2	EM	Algo. 1	Algo. 2
$\theta_j^+ = 0.8$, $\theta_j^- = 0.2$.	2^{10}	150	0.139	0.883	0.896	0.930	0.885	0.895
		500	0.115	0.995	0.992	1.000	1.000	0.999
		1000	0.100	1.000	0.996	1.000	1.000	1.000
	2^{15}	150	0.049	0.523	0.544	0.539	0.530	0.543
		500	0.089	0.924	0.928	0.934	0.930	0.932
		1000	0.078	0.984	0.988	0.991	0.991	0.991
	2^{20}	150	0.019	0.213	0.264	0.270	0.255	0.271
		500	0.019	0.609	0.633	0.636	0.641	0.642
		1000	0.038	0.816	0.848	0.864	0.864	0.863
$\theta_j^+ = 0.9$, $\theta_j^- = 0.1$.	2^{10}	150	0.323	0.909	1.000	1.000	1.000	1.000
		500	0.208	1.000	1.000	1.000	1.000	1.000
		1000	0.167	1.000	1.000	1.000	1.000	1.000
	2^{15}	150	0.317	0.989	0.974	0.993	0.991	0.992
		500	0.220	1.000	0.995	1.000	1.000	1.000
		1000	0.205	1.000	0.994	1.000	1.000	1.000
	2^{20}	150	0.232	0.968	0.941	0.972	0.971	0.970
		500	0.159	1.000	0.999	1.000	1.000	1.000
		1000	0.146	1.000	0.997	1.000	1.000	1.000

Table 2: Pattern selection accuracies for two-parameter SLAM. Tuning parameter $\lambda \in \{-0.2, -0.4, \ldots, -3.8, -4.0\}$ in PEM (Algorithm 1) and $\Upsilon \in \{1.0, 0.9, \ldots, 0.4, 0.3\}$ in FP-VEM (Algorithm 2) are selected based on EBIC.

The simulation results on selection accuracies are presented in Table 2. The "TPR" stands for True Positive Rate, which denotes the proportion of true patterns that are selected. The "1-FDR" stands for "1−False Discovery Rate (FDR)", which denotes the proportion of selected patterns that are true patterns. Table 2 shows the proposed PEM and FP-VEM yield good selection results in various scenarios, while the EM algorithm with direct thresholding at ρ_N suffers from high FDR, i.e., selecting too many non-existing attribute patterns. We would like to point out that the plain VEM as presented in Example 6 is a special case of the proposed FP-VEM, by just taking the fractional power Υ to be $\Upsilon = 1$. So in each simulation run, the result given by VEM is included in the solution path given

by FP-VEM with $\Upsilon \in \{1.0, 0.9, \ldots, 0.4, 0.3\}$, and in the final step EBIC selects the best Υ from the entire solution path. Indeed, in all our simulations about FP-VEM, the result given by $\Upsilon = 1$ is never selected by EBIC, which means the selection result given by plain VEM is never favored over the proposed FP-VEM. We also remark here that the proposed methods are computationally efficient. All the algorithms are implemented in Matlab. In particular, in the case of relatively strong signal with $1 - \theta_j^+ = \theta_j^- = 0.10$, screening and computing an entire solution path for $(2^K, N) = (2^{20}, 1000)$ takes < 2 minutes on average on a laptop with a 2.8 GHz processor, and yields almost perfect pattern selection results, as shown in the last row of Table 2.

We give some discussions on the comparison of the PEM and the FP-VEM algorithms. The estimation accuracies presented in Table 2 generally show the two algorithms have comparable performance on pattern selection. In terms of selecting the tuning parameter, the FP-VEM can be easier to tune because the fractional power Υ is always between 0 and 1, while the PEM algorithm has a negative tuning parameter $\lambda \in (-\infty, 0)$ that can have an arbitrarily large magnitude. Specifically, the scenario of an increasing sparsity corresponds to $\Upsilon \to 0$ and $\lambda \to -\infty$, and when extremal sparsity exists, the FP-VEM needs to choose Υ close to zero with a small magnitude and the PEM needs to choose λ with a large magnitude. Therefore, in such cases the tuning of PEM may take more time, since $\lambda < 0$ needs to be searched over a relatively large interval; an exponential grid search might be of help in this case, while further investigation into how to best specify the grid for searching tuning parameters would be needed. Meanwhile, we find in simulation studies that choosing a small Υ in FP-VEM too close to zero may result in the algorithm to be less stable in some cases. In practice, if the computation time is not a primary concern, we recommend first considering the PEM algorithm for the better stability.

We further conduct a simulation study to investigate how the threshold value ρ_N for the estimated proportions impact the pattern selection results of different methods. In the setting with $1 - \theta_j^+ = \theta_j^- = 0.2$ and $N = 150$ (the same setting as the first line in Table 2), we simulate 200 independent datasets, and apply the proposed PEM (Algorithm 1), FP-VEM (Algorithm 2) and the usual EM algorithm with various thresholds $\rho_N \in \{1/(50N)\} \cup \{i/(2N),\ i = 1, 3, 5, \ldots, 15\}$. Figure 4 plots the average "TPR" and average "1−FDR" versus the threshold values. It can be seen that directly thresholding the MLE of the proportions (corresponding to the thresholding after EM) does not yield good selection results. For a small threshold $\rho_N = 1/(2N)$, the FDR of thresholded EM is quite high. When further decreasing the threshold ρ_N from $1/(2N)$ to $1/(50N)$, the FDR of thresholded EM becomes worse while the proposed methods have stable performance. On the other hand, as the threshold ρ_N increases from $1/(2N)$ to larger values, the TPR of EM quickly decreases. In contrast, the proposed methods PEM and FP-VEM give reasonably good selection results across all the threshold values, and have slightly better performance for smaller thresholds. Even the best selection result given by thresholding EM corresponding to the threshold $\rho_N = 7/(2N)$ is not comparable to those given by the proposed methods.

We next evaluate the performance of the screening procedure. We find that the screening procedure drastically reduces the computational cost in the subsequent shrinkage estimation stage. For instance, in the setting $(N, K) = (150, 15)$ when noise rate is $1 - \theta_j^+ = \theta_j^- = 20\%$, based on 200 runs, the variational screening procedure takes 1.55 seconds on average, and

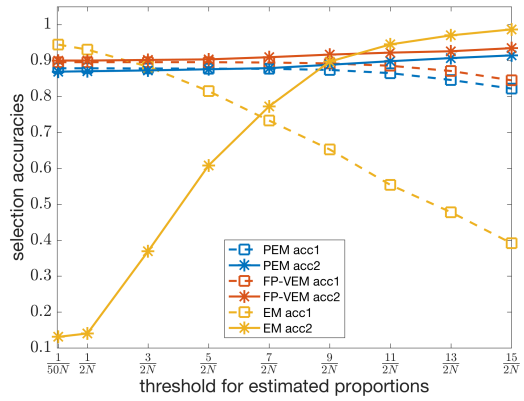
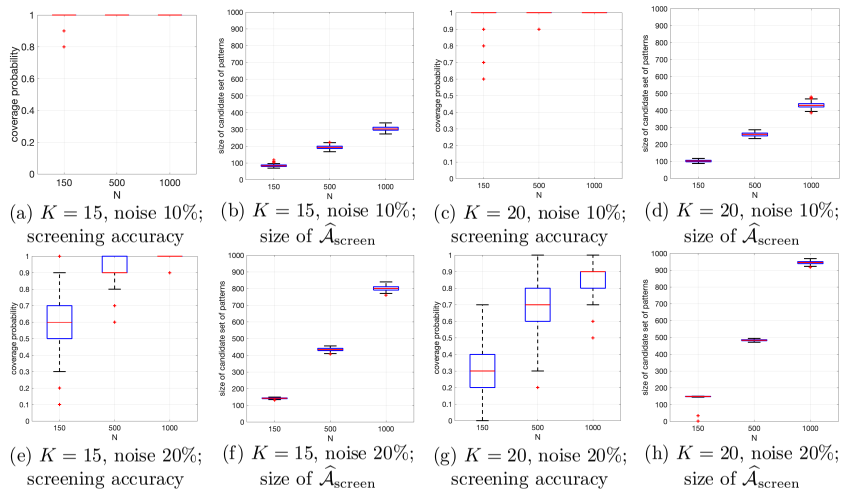


Figure 4: Selection accuracies versus thresholds for the two-parameter SLAM with $1 - \theta_j^+ = \theta_j^- = 0.2$ and $N = 150$. For each method, "acc1" denotes the True Positive Rate (TPR), the proportion of true patterns that are selected; and "acc2" denotes "1−False Discovery Rate (FDR)", the proportion of selected patterns that are true.

the subsequent PEM algorithm takes 6.42 seconds on average; while if no screening is performed, the PEM algorithm takes 7.96×10^3 seconds on average.

As described earlier, the screening is considered successful if all true patterns are included in the candidate set $\widehat{\mathcal{A}}_{\text{screen}}$. Under each simulation scenario in Table 2 corresponding to $K = 15$ or $K = 20$, we record the coverage probabilities of the true patterns for each of 200 runs, where in each run $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} I(\boldsymbol{\alpha} \in \widehat{\mathcal{A}}_{\text{screen}})/|\mathcal{A}_0|$ is recorded as the coverage probability. The boxplots of coverage probabilities under these scenarios are presented in Figure 5(a), (c), (e) and (g). We also record the size of $\widehat{\mathcal{A}}_{\text{screen}}$, i.e., the number of candidate patterns given by the screening procedure in each run, and present their boxplots in Figure 5(b), (d), (f) and (h). The screening procedure generally has good performance. On the other hand, Figure 5(e) and (g) show that for the relatively large noise rate and small sample size, the screening accuracy is not very high.

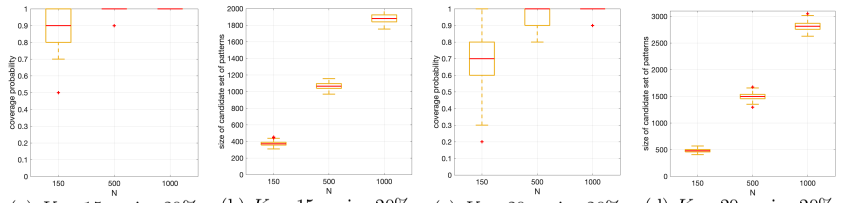
To improve the performance of screening, we apply the strategy of *screening enhanced by Gibbs exploration* described in Remark 18 and take $M = 3$. That is, along the stochastic EM iterations of the screening algorithm, after every three iterations we add the current set of latent patterns to the candidate set $\widehat{\mathcal{A}}_{\text{screen}}$. The resulting screening accuracies and sizes of $\widehat{\mathcal{A}}_{\text{screen}}$ are presented in Figure 6. Compared to the second row of plots in Figure 5, one can clearly see that the enhancing procedure improves the screening accuracy significantly, while the size of $\mathcal{A}_{\text{screen}}$ also increases but still remains quite manageable. Under the noise rate $1 - \theta_j^+ = \theta_j^- = 20\%$, the size of $\mathcal{A}_{\text{screen}}$ is always below N for screening without enhancing, while for screening with enhancing, the size of $\mathcal{A}_{\text{screen}}$ is around $2N$ for $K = 15$ and around $3N$ for $K = 20$. The enhancing by Gibbs exploration would not sacrifice the efficiency of



(a) $K = 15$, noise 10%; screening accuracy (b) $K = 15$, noise 10%; size of $\widehat{\mathcal{A}}_{\text{screen}}$ (c) $K = 20$, noise 10%; screening accuracy (d) $K = 20$, noise 10%; size of $\widehat{\mathcal{A}}_{\text{screen}}$

(e) $K = 15$, noise 20%; screening accuracy (f) $K = 15$, noise 20%; size of $\widehat{\mathcal{A}}_{\text{screen}}$ (g) $K = 20$, noise 20%; screening accuracy (h) $K = 20$, noise 20%; size of $\widehat{\mathcal{A}}_{\text{screen}}$

Figure 5: Screening: plots (a), (c), (e) and (g) are coverage probabilities of the true patterns, from the screening procedure under the two-parameter SLAM; plots (b), (d), (f) and (h) are sizes of $\widehat{\mathcal{A}}_{\text{screen}}$. The "noise" refers to the value of $1 - \theta_j^+ = \theta_j^-$.



(a) $K = 15$, noise 20%; screening accuracy (b) $K = 15$, noise 20%; size of $\widehat{\mathcal{A}}_{\text{screen}}$ (c) $K = 20$, noise 20%; screening accuracy (d) $K = 20$, noise 20%; size of $\widehat{\mathcal{A}}_{\text{screen}}$

Figure 6: Screening enhanced by Gibbs exploration: screening accuracy and size of $\widehat{\mathcal{A}}_{\text{screen}}$. Noise rate is $1 - \theta_j^+ = \theta_j^- = 20\%$.

the screening procedure itself, though it results in a larger set of $\widehat{\mathcal{A}}_{\text{screen}}$ which incurs higher computational cost in the shrinkage stage. In practice, one should leverage this tradeoff according to the sample size. Specifically, when sample size N is small, choosing a more conservative screening procedure (with a smaller integer M) is recommended, because this would increase the screening accuracy without causing much computational burden for the shrinkage algorithm. With the enhanced screening procedure, in the relatively weak signal case $1 - \theta_j^+ = \theta_j^- = 0.2$ and under $(K, N) = (15, 150)$, the two accuracy measures 1−FDR and TPR for the PEM algorithm, become $(0.850, 0.860)$ (previously it was $(0.523, 0.530)$

in Table 2), and those under the FP-VEM algorithm become $(0.839, 0.853)$ (previously $(0.544, 0.543)$ in Table 2). Under $(K, N) = (20, 150)$, the two accuracy measures for the PEM become $(0.608, 0.648)$ (previously $(0.213, 0.255)$ in Table 2) and those for the FP-VEM become $(0.620, 0.634)$ (previously $(0.264, 0.271)$ in Table 2).

Multi-parameter all-effect SLAM. We next consider the multi-parameter all-effect SLAM introduced in (5) in Example 3 with an identity link function $f(\cdot)$. Let the Q-matrix be in the form $Q = (Q_1^\top, Q_2^\top, Q_2^\top)$ with Q_1 and Q_2 specified in (22). Similar to the two-parameter simulation study, we consider three dimensions of possible attribute patterns with $2^K = 2^{10}, 2^{15}$, and 2^{20}, and three sample sizes with $N = 150, 500$ and 1000. For each item, we set the baseline probability, the positive response probability of the all-zero attribute pattern $\boldsymbol{\alpha} = \mathbf{0}_K$, to 0.2 (i.e., $\theta_{j,\mathbf{0}_K} = 0.2$), and the positive response probability of $\boldsymbol{\alpha} = \mathbf{1}_K$ to 0.8 (i.e., $\theta_{j,\mathbf{1}_K} = 0.8$). And we set all the main effects and interaction effects parameters of the item to be equal (i.e., $\beta_{j,S_1} = \beta_{j,S_2}$ for any $\varnothing \neq S_1, S_2 \subset \mathcal{K}_j$ for the β-coefficients in (5)). We randomly generate the set of true attribute patterns, $\mathcal{A}_0 \subseteq \{0,1\}^K$ with cardinality $|\mathcal{A}_0| = 10$ and set $p_{\boldsymbol{\alpha}} = 0.1$ for all $\boldsymbol{\alpha} \in \mathcal{A}_0$.

2^K	N	1−FDR			TPR		
		EM	Algo. 1	Algo. 2	EM	Algo. 1	Algo. 2
2^{10}	150	0.277	0.983	0.953	0.996	0.980	0.974
	500	0.214	0.988	0.976	1.000	1.000	1.000
	1000	0.193	0.992	0.986	1.000	1.000	1.000
2^{15}	150	0.198	0.900	0.893	0.904	0.902	0.902
	500	0.166	0.999	0.997	1.000	1.000	1.000
	1000	0.134	1.000	0.996	1.000	1.000	1.000
2^{20}	150	0.109	0.723	0.741	0.739	0.734	0.743
	500	0.129	0.980	0.981	0.980	0.982	0.983
	1000	0.104	1.000	0.998	1.000	1.000	1.000

Table 3: Pattern selection accuracies for multi-parameter all-effect SLAM. Tuning parameter $\lambda \in \{-0.2, -0.4, \ldots, -4.0\}$ in PEM (Algorithm 1) and $\Upsilon \in \{1.0, 0.9, \ldots, 0.3\}$ in FP-VEM (Algorithm 2) are selected using EBIC. Signal strengths are $\theta_{j,\mathbf{0}_K} = 0.1$, $\theta_{j,\mathbf{1}_K} = 0.9$.

Similar to the observations in Table 2, Table 3 shows that the proposed methods also have good pattern selection performance for the more complicated multi-parameter all-effect model. The approximate screening algorithm based on the likelihood of the two-parameter submodel is quite effective here for obtaining candidate patterns under the multi-parameter model. And similarly to the two-parameter case, the EM algorithm tends to severely overselects the attribute patterns. Please see Appendix B for additional results on the performance of the screening procedure.

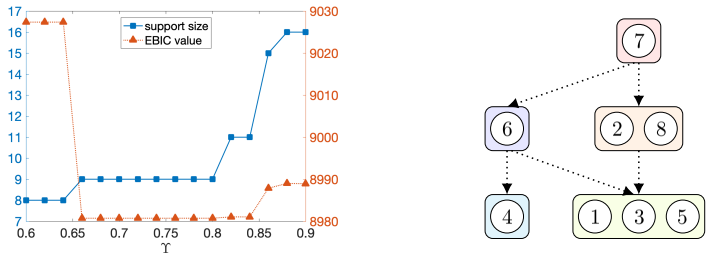
6. Data Analysis

In this section, we apply the proposed methodology to two real world datasets in educational assessments to uncover the knowledge structure of the student population.

Analysis of Fraction Subtraction Data. The fraction subtraction dataset is widely analyzed in the psychometrics literature (de la Torre and Douglas, 2004; DeCarlo, 2011; Henson et al., 2009; de la Torre, 2011). The dataset contains $N = 536$ middle school students' binary (correct or wrong) responses to 20 questions that were designed for the diagnostic assessment of 8 skill attributes related to fraction and subtraction. Table 4 presents the Q-matrix specified in de la Torre and Douglas (2004). The eight attributes are (α_1) Convert a whole number to a fraction; (α_2) Separate a whole number from a fraction; (α_3) Simplify before subtracting; (α_4) Find a common denominator; (α_5) Borrow from whole number part; (α_6) Column borrow to subtract the second numerator from the first; (α_7) Subtract numerators; (α_8) Reduce answers to simplest form.

Item ID	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
1	0	0	0	1	0	1	1	0
2	0	0	0	1	0	0	1	0
3	0	0	0	1	0	0	1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
18	0	1	0	0	1	1	1	0
19	1	1	1	0	1	0	1	0
20	0	1	1	0	1	0	1	0

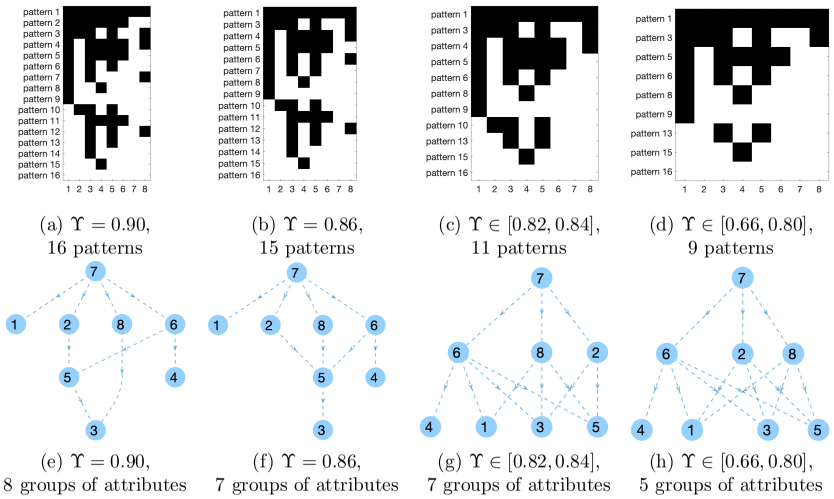
Table 4: Q-matrix, Fraction Subtraction Data



(a) EBIC values and support sizes versus Υ (b) attribute structure selected by EBIC

Figure 7: Results of Fraction Subtraction Data analyzed using two-parameter SLAM.

Many studies in the literature use the two-parameter SLAM to fit the dataset, mostly due to that it is reasonable to assume the required attributes of each item act together to form a "capable" knowledge state and an "incapable" knowledge state. This results in two levels of item parameters for each item. We first use the two-parameter model to



(a) $\Upsilon = 0.90$, 16 patterns (b) $\Upsilon = 0.86$, 15 patterns (c) $\Upsilon \in [0.82, 0.84]$, 11 patterns (d) $\Upsilon \in [0.66, 0.80]$, 9 patterns

(e) $\Upsilon = 0.90$, 8 groups of attributes (f) $\Upsilon = 0.86$, 7 groups of attributes (g) $\Upsilon \in [0.82, 0.84]$, 7 groups of attributes (h) $\Upsilon \in [0.66, 0.80]$, 5 groups of attributes

Figure 8: Fraction Subtraction Data: different sets of estimated patterns (a)–(d) (black for "0" and white for "1") and the corresponding attribute structures (e)–(h) under various Υ's in Algorithm 2. Plot (h) here is equivalent to Figure 7(b).

analyze the data. Given this 20×8 Q-matrix, the number of equivalence classes induced by the Q-matrix $Q_{20\times 8}$ under the two-parameter model is $|\{\vee_{j\in S}\boldsymbol{q}_j : S \subseteq \{1,\ldots,J\}\}| = 58$. We apply Algorithm 2, the FP-VEM algorithm with a sequence of fractional power values $\Upsilon \in \{0.90, 0.89, \cdots, 0.60\}$ and use EBIC to select the tuning parameter Υ while keeping the Dirichlet hyperparameter $\beta = 0.01$. Figure 7(a) plots the EBIC values and the support sizes of $\boldsymbol{p}$, both against the Υ values. It can be seen that $\Upsilon = 0.8$ yields the smallest EBIC value 8.98×10^3, and it is the largest Υ value in the flat window of $[0.66, 0.8]$ that gives 9 equivalence classes of attribute patterns. We also use the multi-parameter all-effect model introduced in Example 3 to fit the dataset. For a range of values of the tuning parameters Υ, the smallest EBIC value is above 1.02×10^4, which is much higher than the smallest EBIC 8.98×10^3 given by the two-parameter model. This also aligns with the results in the literature that the two-parameter model fits the fraction subtraction dataset better than other models (DeCarlo, 2011; de la Torre and Douglas, 2004). Therefore next we only present and discuss the results given by the two-parameter model.

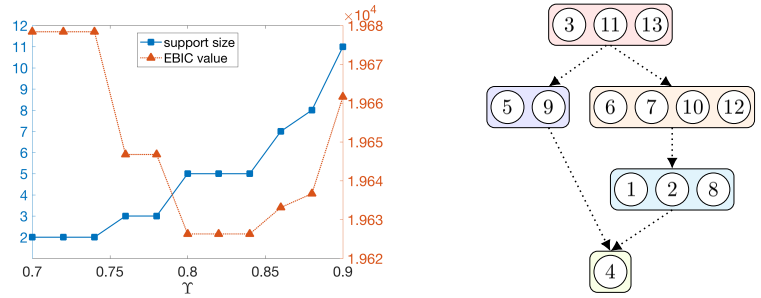
Figure 7(b) plots the attribute structure corresponding to the 9 equivalence classes of attribute patterns selected by EBIC. We obtain this attribute structure using the following procedure. First, we obtain the representatives of these 9 equivalence classes and construct a 9×8 matrix of selected attribute patterns. We denote this 9×8 matrix by $\widehat{\mathbf{A}}$, with each row of $\widehat{\mathbf{A}}$ a 8-dimensional binary vector denoting one selected knowledge state (i.e., attribute pattern). We next examine the partial orders among the columns of this matrix to determine

the relationships among attributes. In particular, if $\widehat{\mathbf{A}}(\cdot, k_1) \succeq \widehat{\mathbf{A}}(\cdot, k_2)$, then attribute k_1 is considered as a prerequisite for attribute k_2. Examining these 9 selected knowledge states, we find that the total number of 8 attributes are separated into 5 groups $G_1 = \{7\}$, $G_2 = \{2, 8\}$, $G_3 = \{6\}$ and $G_4 = \{4\}$ and $G_5 = \{1, 3, 5\}$, such that the attributes in the same group play the same role in clustering the students population into the 9 knowledge states. In particular, based on the observed data, attributes 2 and 8 are equivalent in distinguishing the students population's knowledge states; and so are attributes 1, 3, 5. The estimated prerequisite relationship among these 5 groups is depicted in Figure 7(b). Figure 7(b) implies that attribute (α_7) *Subtract numerators*, is a quite basic skill attribute and serves as prerequisite for all the remaining attributes. This suits the common sense that in the problems about fraction and subtraction, the ability of subtracting integers should be the most basic. Figure 7 also shows that attributes (α_2), (α_6), (α_8) are middle level skills that only has one prerequisite attribute (α_7), and serve as prerequisites for multiple other skills. Finally, the remaining attributes (α_4), (α_1), (α_3) and (α_5) are high level skills in the hierarchical structure. We would like to point out that the directed edges in the attribute hierarchy in Figure 7(b) (and also in the later Figure 9 for the TIMSS dataset) do not necessarily correspond to causal relations between the skill attributes. Instead, the attribute hierarchy results from the learned subset of attribute patterns, and it just reflects the estimated cognitive structure of the students being measured.

For the Fraction Subtraction data, in addition to the attribute structure chosen by EBIC shown in Figure 7(b), we also present those sets of attribute patterns selected by different Υ's in the solution path. The four sets of patterns and their corresponding attribute structures are presented in Figure 8. As shown in Figure 8(a)–(d), the latent patterns selected by a smaller Υ always form a subset of those patterns selected by a larger Υ. Also, the attribute structures selected by different Υ's share some commonalities. Among the second row of Figure 8, plot (h) is equivalent to the attribute structure in Figure 7(b).

Analysis of TIMSS Data. We also apply the proposed method to the TIMSS 2003 8th grade data. The dataset contains $N = 757$ students' responses to $J = 23$ test items, and the Q-matrix is of size 23×13. Under the two-parameter SLAM, the Q-matrix gives $|\{\vee_{j \in S} \boldsymbol{q}_j : S \subseteq \{1, \ldots, J\}\}| = 1625$ equivalence classes. Figure 9 shows the results of fitting the two-parameter SLAM with $\beta = 0.01$. The fractional power Υ selected by EBIC is 0.84 and the corresponding number of equivalence classes is 5. The smallest EBIC value in Figure 9(a) is 1.96×10^4. We remark that we also fit the general multi-parameter all-effect SLAM to the dataset, while the smallest EBIC given by the multi-parameter model is 7.38×10^4, which is much larger than the best EBIC given by the two-parameter SLAM. So we next focus on the results given by the two-parameter SLAM.

Figure 9(b) plots the attribute structure given by the selected 5 knowledge states. The 13 attributes are separated into five groups $G_1 = \{3, 11, 13\}$, $G_2 = \{5, 9\}$, $G_3 = \{6, 7, 10, 12\}$ and $G_4 = \{1, 2, 8\}$ and $G_5 = \{4\}$, such that the attributes in the same group play the same role in clustering the student population into the five knowledge states. The prerequisite relationships among groups of attributes is also shown in Figure 9(b). Attribute (α_3) *compute fluently with multi-digit numbers and find common factors and multiples*, attribute (α_{11}) *compare two fractions with different numerators and different denominators*, attribute (α_{13}) *use equivalent fraction as a strategy to add and subtract fractions*, are the most basic skills



(a) EBIC values and support sizes versus Υ (b) attribute structure selected by EBIC

Figure 9: Results of TIMSS 2003 8th Grade Data analyzed using two-parameter SLAM.

in the attribute hierarchy and serve as the prerequisites for all the remaining attributes. Indeed, these three are basic algorithmic operations needed to solve the mathematical problems in the TIMSS test. In addition to the structure selected by EBIC presented in Figure 9(b), other attribute structures corresponding to different $\Upsilon \in [0.7, 0.9]$ are presented in Figure 14 in Appendix B.

Existing works in the literature analyzing the fraction subtraction data and the TIMSS data either make the assumption that all possible configurations of latent attribute patterns exist in the population or pre-specify the attribute structure based on domain experts' judgements (Su et al., 2013). To our knowledge, there has not been a systematic approach to selecting a potentially small set of latent patterns from a high-dimensional space. For the two real datasets, we also find that the EBIC values of the existing EM algorithm are much larger than the proposed method, as indicated in Figures 7 and 9 when Υ close to 1; thus the proposed method provides a better fit of the two datasets.

7. Discussion

In this paper we propose a penalized likelihood method to learn the attribute patterns in the structured latent attribute models, a special family of discrete latent variable models. We allow the number of latent patterns to go to infinity and perform pattern selection by penalizing the proportion parameters of the latent attribute patterns. The theory of pattern selection consistency is established for the proposed regularized MLE. The nice form of the penalty term facilitates the computation. Two algorithms are developed to solve the optimization problem, one being a modification of the EM algorithm, and the other being a variational EM algorithm that results from an alternative Bayesian formulation of the objective function. The simulation study and real data analysis show the proposed methods have good pattern selection performance.

This work assumes the design matrix Q is prespecified and correct. In practice, if there is reason to suspect that the Q-matrix could be misspecified, then one needs to simultaneously

estimate the Q-matrix and learn the attribute patterns from data. Given fixed number of attribute patterns, previous works including Xu and Shang (2018) and Chen et al. (2018) used the likelihood based methods and the Bayesian methods, respectively, to estimate Q. It is also desirable to develop methods to jointly estimate Q and learn attribute patterns with the existence of large number of attributes. We would like to point out that the identifiability results developed in this work (in Section 3) directly apply to this case, and can guarantee both the design matrix Q and the set of significant attribute patterns are learnable from data.

The learnability theory developed in this paper guarantees one can reliably learn a SLAM with an arbitrary set of attribute patterns from data. As mentioned earlier, SLAMs can be expressed as higher-order probability tensors with special structures. Also, SLAMs share similarities with the restricted Boltzmann machines and the deep Boltzmann machines in terms of the bipartite graph structure among the latent and observed multivariate binary variables. Current techniques for proving identifiability of SLAMs could be adapted to develop theory for uniqueness of structured tensor decompositions and learnability of some more complicated latent variable models. We leave these directions for future study.

Acknowledgement

The authors are grateful to the editors and two reviewers for their helpful and constructive comments. This research was partially supported by National Science Foundation grants SES1659328 and DMS-1712717, and Institute of Education Sciences grant R305D160010. This research was also supported in part through the High Performance Computing clusters provided by Advanced Research Computing at the University of Michigan, Ann Arbor.

Appendix

Appendix A: Technical Proofs

We introduce a useful notation, the T-matrix, before proving the identifiability theory. We consider a marginal probability matrix $T(\Gamma^{\mathcal{A}}, \boldsymbol{\Theta}^{\mathcal{A}})$ of size $2^J \times |\mathcal{A}|$ as follows. When it causes no confusion, we also write $T(\Gamma^{\mathcal{A}}, \boldsymbol{\Theta}^{\mathcal{A}})$ simply as $T(\Gamma, \boldsymbol{\Theta})$. Rows of $T(\Gamma, \boldsymbol{\Theta})$ are indexed by the 2^J possible response patterns $\boldsymbol{r} = (r_1, \ldots, r_J)^\top \in \{0,1\}^J$ and columns of $T(\Gamma, \boldsymbol{\Theta})$ are indexed by latent attribute patterns $\boldsymbol{\alpha} \in \mathcal{A}$, while the $(\boldsymbol{r}, \boldsymbol{\alpha})$th entry of $T(\Gamma, \boldsymbol{\Theta})$, denoted by $T_{\boldsymbol{r},\boldsymbol{\alpha}}(\Gamma, \boldsymbol{\Theta})$, represents the marginal probability that subjects in latent class $\boldsymbol{\alpha}$ provide positive responses to the set of items $\{j : r_j = 1\}$, namely $T_{\boldsymbol{r},\boldsymbol{\alpha}}(\Gamma, \boldsymbol{\Theta}) = P(\boldsymbol{R} \succeq \boldsymbol{r} \mid \boldsymbol{\Theta}, \boldsymbol{\alpha}) = \prod_{j=1}^J \theta_{j,\boldsymbol{\alpha}}^{r_j}$. Denote the $\boldsymbol{\alpha}$th column vector and the $\boldsymbol{r}$th row vector of the T-matrix by $T_{\boldsymbol{\cdot},\boldsymbol{\alpha}}(\Gamma, \boldsymbol{\Theta})$ and $T_{\boldsymbol{r},\boldsymbol{\cdot}}(\Gamma, \boldsymbol{\Theta})$ respectively. Let $\boldsymbol{e}_j$ denote the J-dimensional unit vector with the jth element being one and all the other elements being zero, then any response pattern $\boldsymbol{r}$ can be written as a sum of some $\boldsymbol{e}$-vectors, namely $\boldsymbol{r} = \sum_{j:r_j=1} \boldsymbol{e}_j$. The $\boldsymbol{r}$th element of the 2^J-dimensional vector $T(\Gamma, \boldsymbol{\Theta})\boldsymbol{p}$ is $\{T(\Gamma, \boldsymbol{\Theta})\boldsymbol{p}\}_{\boldsymbol{r}} = T_{\boldsymbol{r},\boldsymbol{\cdot}}(\Gamma, \boldsymbol{\Theta})\boldsymbol{p} = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} T_{\boldsymbol{r},\boldsymbol{\alpha}}(\Gamma, \boldsymbol{\Theta}) p_{\boldsymbol{\alpha}} = P(\boldsymbol{R} \succeq \boldsymbol{r} \mid \Gamma, \boldsymbol{\Theta})$. The T-matrix have some nice algebraic properties that will be useful in later proofs. We state them in the following lemma, the proof of which is similar to that of Proposition 3 in Xu (2017) and hence is omitted.

Lemma 19 *Under a SLAM with constraint matrix Γ, $(\Gamma, \boldsymbol{\Theta}, \boldsymbol{p})$ are jointly identifiable if and only if for any $(\Gamma, \boldsymbol{\Theta}, \boldsymbol{p})$ and $(\bar{\Gamma}, \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{p}})$,*

$$T(\Gamma, \boldsymbol{\Theta})\boldsymbol{p} = T(\bar{\Gamma}, \bar{\boldsymbol{\Theta}})\bar{\boldsymbol{p}} \tag{25}$$

implies $(\Gamma, \boldsymbol{\Theta}, \boldsymbol{p}) = (\bar{\Gamma}, \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{p}})$. For any $\boldsymbol{\theta}^* = (\theta_1, \ldots, \theta_J)^\top \in \mathbb{R}^J$, there exists an invertible matrix $D(\boldsymbol{\theta}^*)$ only depending on $\boldsymbol{\theta}^*$, such that

$$T(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^* \mathbf{1}^\top) = D(\boldsymbol{\theta}^*) T(\Gamma, \boldsymbol{\Theta}),$$

where $\mathbf{1}^\top$ denotes an all-one vector and $\boldsymbol{\theta}^* \mathbf{1}^\top$ is a matrix of same size as $\boldsymbol{\Theta}$.

Proof of Theorem 2 and Corollary 3. We aim to prove that if $\Gamma := \Gamma^{\mathcal{A}_0}$ of size $J \times L_0$ ($L_0 = |\mathcal{A}_0|$) satisfies Conditions A and B, then for any binary matrix $\bar{\Gamma}$ also of size $J \times L_0$, which can be viewed as a constraint matrix imposing restrictions on the parameter space of the $J \times L_0$ item parameter matrix $\bar{\boldsymbol{\Theta}}$, and for any L_0-dimensional vector $\bar{\boldsymbol{p}} := (\bar{p}_1, \ldots, \bar{p}_{L_0})$ with $\bar{p}_l \geq 0$ and $\sum_{l=1}^{L_0} \bar{p}_l = 1$, which can be viewed as a population proportion vector giving proportions of the L_0 latent classes, if

$$T(\Gamma, \boldsymbol{\Theta})\boldsymbol{p} = T(\bar{\Gamma}, \bar{\boldsymbol{\Theta}})\bar{\boldsymbol{p}} \tag{26}$$

holds, then $(\Gamma, \boldsymbol{\Theta}, \boldsymbol{p}) = (\bar{\Gamma}, \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{p}})$ up to a label swapping of the latent classes. If this is proved, then combining Condition C that any column vector of $\Gamma^{\mathcal{A}_0}$ is different from any column vector of $\Gamma^{\mathcal{A}_0^c}$, we would have the conclusion that the identified $\Gamma^{\mathcal{A}_0}$ uniquely maps to the true set of attribute patterns $\mathcal{A}_0$.

We add a remark here that given (26), the columns of the $\bar{\Gamma}$ do not necessarily have the interpretation of representing some K-dimensional binary attribute patterns; instead, these columns just correspond to L_0 latent classes. And after we obtain $(\Gamma, \boldsymbol{\Theta}, \boldsymbol{p}) = (\bar{\Gamma}, \bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{p}})$ up to a label swapping, we would have the conclusion that $\bar{\Gamma}$ equals Γ up to column permutation; Then with Condition C, the $\bar{\Gamma}$ would have the interpretation of being the constraint matrix for the attribute patterns in $\mathcal{A}_0$. Because of this, in the following proof, we sometimes will also ignore the interpretation of the columns of the true $\Gamma^{\mathcal{A}_0}$, and simply denote the columns of it by the column index integer l, i.e., $\Gamma^{\mathcal{A}_0}$ has columns $\Gamma^{\mathcal{A}_0}_{\boldsymbol{\cdot},l}$ for $l = 1, \ldots, L_0$.

For notational simplicity, we denote $\Gamma^{(S_i,\mathcal{A}_0)}$ by Γ^i for $i = 1, 2$ and $\Gamma^{((S_1\cup S_2)^c,\mathcal{A}_0)}$ by Γ^3. We also denote item parameter matrix $\boldsymbol{\Theta}^{(S_1,\mathcal{A}_0)}$, $\boldsymbol{\Theta}^{(S_2,\mathcal{A}_0)}$ and $\boldsymbol{\Theta}^{((S_1\cup S_2)^c,\mathcal{A}_0)}$ by $\boldsymbol{\Theta}^1$, $\boldsymbol{\Theta}^2$ and $\boldsymbol{\Theta}^3$, respectively. So each $\boldsymbol{\Theta}^i$ has the same size as Γ^i and respects the constraints specified by Γ^i. Without loss of generality, suppose Γ takes the form $\Gamma^\top = [(\Gamma^1)^\top, (\Gamma^2)^\top, (\Gamma^3)^\top]$, where each Γ^i is of size $J_i \times L_0$ and $J_1 + J_2 + J_3 = J$. For any item j, by the definition of SLAM we have all those $\boldsymbol{\alpha}$ with $\Gamma^{\mathcal{A}_0}_{j,\boldsymbol{\alpha}} = 1$ have the same highest value of item parameter. For simplicity, we denote this value of the item parameter by $\theta_{j,H}$, where "H" stands for "highest" level item parameter for item j.

We first show $T(\bar{\Gamma}^1, \bar{\boldsymbol{\Theta}}^1)$ and $T(\bar{\Gamma}^2, \bar{\boldsymbol{\Theta}}^2)$ both have full column rank L_0, and that $\bar{p}_l > 0$ for all $l \in \{1, \ldots, L_0\}$. By Proposition 3 in Gu and Xu (2019b), Condition A ensures that $T(\Gamma^1, \boldsymbol{\Theta}^1)$ of size 2^{J_1} and $T(\Gamma^2, \boldsymbol{\Theta}^2)$ of size 2^{J_2} both have full column rank L_0, since Γ^1 and Γ^2 are both separable. Moreover, in the proof of that conclusion, an invertible square matrix W_1 of size $2^{J_1} \times 2^{J_1}$ as well as L_0 response patterns $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{L_0} \in \{0,1\}^L$ were constructed

such that the row vectors in the transformed $W_1 \cdot T(\Gamma^1, \mathbf{\Theta}^1)$, which are indexed by the chosen $\boldsymbol{r}_1, \ \ldots, \ \boldsymbol{r}_{L_0}$, form a $L_0 \times L_0$ lower triangular matrix with nonzero diagonal elements. In other words, in the $2^{J_1} \times L_0$ rectangular matrix $W_1 T(\Gamma^1, \mathbf{\Theta}^1)$, there is a $L_0 \times L_0$ submatrix that is lower triangular and full-rank. For notational simplicity, we denote this submatrix by $\{W_1 T(\Gamma^1, \mathbf{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}}$. Similarly, there exists W_2 and $\boldsymbol{r}'_1, \ \ldots, \ \boldsymbol{r}'_{L_0} \in \{0,1\}^{L_0}$ such that there is a $L \times L$ full-rank submatrix of $W_2 T(\Gamma^2, \mathbf{\Theta}^2)$ with rows indexed by $\boldsymbol{r}'_1, \ \ldots, \ \boldsymbol{r}'_{L_0}$, which we denote by $\{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}$.

Based on the above constructions, there exist two invertible square matrices U_1 and U_2 such that $U_1 \cdot \{W_1 T(\Gamma^1, \mathbf{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}} = I_{L_0}$ and $U_2 \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}} = I_{L_0}$. Denote the C row vectors of U_1 by $\{\boldsymbol{u}_l^\top, \ l \in [L_0]\}$, then we have that for any $l \in [L_0]$,

$$\boldsymbol{u}_l^\top \cdot \{W_1 T(\Gamma^1, \mathbf{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}} = (\mathbf{0}, \underbrace{1}_{\text{column } l}, \mathbf{0}). \tag{27}$$

Next we prove by contradiction that $\{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}$ and $\{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}$ must also be invertible. We focus on $\{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}$ and conclusion for the other is the same. If $\{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}$ does not have full rank, then $U_2 \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}$ also does not have full rank, so there exists a nonzero vector $\boldsymbol{x} = (x_1, \ldots, x_{L_0})$ such that

$$\boldsymbol{x}^\top \cdot U_2 \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}} = \mathbf{0}.$$

Note that $\boldsymbol{x}^\top \cdot U_2 \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}} = \boldsymbol{x}$ from the previous construction of W_2. Since $\boldsymbol{x} \neq \mathbf{0}$, suppose without loss of generality that $x_l \neq 0$ for some l, then we have

$$[\boldsymbol{u}_{\boldsymbol{\alpha}}^\top \cdot \{W_1 T(\Gamma^1, \mathbf{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{x}^\top \cdot U_2 \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \boldsymbol{p} = x_l p_l \neq 0,$$
$$[\boldsymbol{u}_{\boldsymbol{\alpha}}^\top \cdot \{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{x}^\top \cdot U_2 \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \bar{\boldsymbol{p}} = 0,$$

which contradicts (26). Here $\boldsymbol{a} \odot \boldsymbol{b}$ denotes the elementwise product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of the same length. Therefore $\{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}$ must have full rank C, and so as $\{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}$.

Based on the above conclusion, we next show that $\bar{p}_l > 0$ for any $l \in [L_0]$. Suppose this is not true and $\bar{p}_l = 0$ for some l, then there exists a nonzero vector $\boldsymbol{y} = (y_1, \ldots, y_{L_0})^\top$ such that

$$\boldsymbol{y}^\top \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}} = (\mathbf{0}, \underbrace{1}_{\text{column } l}, \mathbf{0}).$$

Since $\{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}$ has full rank and $\boldsymbol{y} \neq \mathbf{0}$, we have $\boldsymbol{y}^\top \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}} \neq \mathbf{0}$. Without loss of generality, suppose the $l^\star$-th column of this product vector is nonzero and denote the nonzero value by $b_{l^\star}$, then using the $\boldsymbol{u}$-vectors constructed previously in (27), we have

$$[\boldsymbol{u}_{\boldsymbol{\alpha}^\star}^\top \cdot \{W_1 T(\Gamma^1, \mathbf{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{y}^\top \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \boldsymbol{p} = b_{l^\star} p_{l^\star} \neq 0,$$
$$[\boldsymbol{u}_{\boldsymbol{\alpha}^\star}^\top \cdot \{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{y}^\top \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \bar{\boldsymbol{p}} = 0,$$

which contradicts (26). This shows that $\bar{p}_l > 0$ must hold for all $l \in [L_0]$.

We next show that for any $j \in (S_1 \cup S_2)^c$ and any $l \in \{1, \ldots, L_0\}$, $\theta_{j,l} = \theta_{j,\sigma(l)}$, where $\sigma(\cdot)$ is a permutation map from $\{1, \ldots, L_0\}$ to $\{1, \ldots, L_0\}$. There must exist a permutation map $\sigma : \{1, \ldots, L\} \to \{1, \ldots, L\}$ such that for each $l \in [L_0]$,

$$\bar{f}_{\sigma(l)} := [\boldsymbol{u}_l^\top \cdot \{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}]_{\sigma(l)} \neq 0.$$

This is because otherwise there would exist $l \in [L_0]$ such that $\{U_1 \cdot T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\cdot,l}$ equals the zero vector, which contradicts the fact that both U_1 and $\{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}$ are invertible matrices. Given the permutation σ, there exists a $L_0 \times L_0$ invertible matrix V with row vectors denoted by $\{\boldsymbol{v}_l, \ l \in [L_0]\}$ such that for each $\boldsymbol{\alpha} \in \mathcal{A}$,

$$\boldsymbol{v}_l^\top \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}} = (\mathbf{0}, \underbrace{1}_{\text{column } \sigma(l)}, \mathbf{0}). \tag{28}$$

Then we have

$$[\boldsymbol{u}_l^\top \cdot \{W_1 T(\Gamma^1, \mathbf{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \boldsymbol{p} = f_l p_l, \tag{29}$$
$$[\boldsymbol{u}_l^\top \cdot \{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{v}_l^\top \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \bar{\boldsymbol{p}} = \bar{f}_{\sigma(l)} \bar{p}_{\sigma(l)} \neq 0, \tag{30}$$

where $f_l = [\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}]_l$. Now we have $f_l p_l = \bar{f}_{\sigma(l)} \bar{p}_{\sigma(l)} \neq 0$. Next further consider an arbitrary item $j \in (S_1 \cup S_2)^c$. Equation (26) indicates that

$$\theta_{j,l} = \frac{T_{\boldsymbol{e}_j,\cdot}(\Gamma, \mathbf{\Theta}) \odot (29)}{(29)} = \frac{T_{\boldsymbol{e}_j,\cdot}(\bar{\Gamma}, \bar{\mathbf{\Theta}}) \odot (30)}{(30)} = \bar{\theta}_{j,\sigma(l)}.$$

We next show that for any $j \in S_1 \cup S_2$ and any $l \in \{1, \ldots, L_0\}$ such that $\Gamma_{j,l} = 1$, $\theta_{j,l} = \theta_{j,H} = \bar{\theta}_{j,\sigma(l)} = \bar{\theta}_{j,H}$. We introduce a lemma before proceeding with the proof.

Lemma 20 *Under the assumptions of Theorem 2, the vectors $\{\boldsymbol{v}_l, \ l \in \mathcal{A}_0\}$ constructed in (28) satisfy that*

$$\{\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}\}_{l'} = 0, \quad \forall \boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l \text{ under } \Gamma^{\mathcal{A}_0}.$$

Proof of Lemma 20 If $\{\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}\}_{l'} = z_{l'} \neq 0$, then similar to (29) and (30) we have

$$[\boldsymbol{u}_{l'}^\top \cdot \{W_1 T(\Gamma^1, \mathbf{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \mathbf{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \boldsymbol{p} = z_{l'} p_{l'} \neq 0,$$
$$[\boldsymbol{u}_{l'}^\top \cdot \{W_1 T(\bar{\Gamma}^1, \bar{\mathbf{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}] \odot [\boldsymbol{v}_{\boldsymbol{\alpha}}^\top \cdot \{W_2 T(\bar{\Gamma}^2, \bar{\mathbf{\Theta}}^2)\}_{\boldsymbol{r}'_{1:L_0}}] \cdot \bar{\boldsymbol{p}} = \bar{f}_{\sigma(l)} \bar{p}_{\sigma(l)},$$

and further we have $\theta_{j,l'} = \bar{\theta}_{j,\sigma(l)} = \theta_{j,l}$ for $j \in (S_1 \cup S_2)^c$, which contradicts condition (C2). This completes the proof of the lemma. □

We proceed with the proof. For any $l \in [L_0]$, define $\boldsymbol{\theta}^* = \sum_{h \in S_1 : \Gamma_{h,l}=1} \theta_{h,1} \boldsymbol{e}_h$. With $\boldsymbol{\theta}^*$, the row vector corresponding to $\boldsymbol{r}^* = \sum_{h \in S_1 : \Gamma_{h,l}=0} \boldsymbol{e}_h$ in the transformed T-matrix satisfies that

$$b_l := T_{\boldsymbol{r}^*,l}(\Gamma^1, \mathbf{\Theta}^1 - \boldsymbol{\theta}^* \mathbf{1}^\top) \neq 0; \tag{31}$$

$$T_{\boldsymbol{r}^*,l'}(\Gamma^1, \boldsymbol{\Theta}^1 - \boldsymbol{\theta}^*\mathbf{1}^\top) = 0, \quad \forall \boldsymbol{\alpha}_{l'} \not\preceq_{S_1} \boldsymbol{\alpha}_l \text{ under } \Gamma^{\mathcal{A}_0}.$$

The proof of Step 2 as well as Lemma 20 ensures

$$\begin{aligned} f_l =& [\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \boldsymbol{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}]_l \neq 0; \\ & [\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \boldsymbol{\Theta}^2)\}_{\boldsymbol{r}'_{1:L_0}}]_{l'} = 0, \quad \forall \boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l \text{ under } \Gamma^{\mathcal{A}_0}. \end{aligned} \tag{32}$$

Consider any $j \in S_1 \cup S_2$ such that $\Gamma_{j,l} = 1$, then obviously $\boldsymbol{e}_j$ is not included in the sum in the previously defined response pattern $\boldsymbol{r}^*$, because $\boldsymbol{r}^*$ only contains those items that $\boldsymbol{\alpha}_l$ is not capable of, i.e., those j s.t. $\Gamma^{\mathcal{A}_0}_{j,l} = 0$. The above two equations (31) and (32) indicate

$$T_{\boldsymbol{r}^*,\cdot}(\Gamma^1, \boldsymbol{\Theta}^1 - \boldsymbol{\theta}^*\mathbf{1}^\top) \odot [\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \boldsymbol{\Theta}^2)\}] = \Big(\mathbf{0}^\top, \underbrace{b_l \cdot f_l}_{\text{column } l}, \mathbf{0}^\top\Big), \tag{33}$$

$$T_{\boldsymbol{r}^*+\boldsymbol{e}_j,\cdot}(\Gamma^1, \boldsymbol{\Theta}^1 - \boldsymbol{\theta}^*\mathbf{1}^\top) \odot [\boldsymbol{v}_l^\top \cdot \{W_2 T(\Gamma^2, \boldsymbol{\Theta}^2)\}] = \Big(\mathbf{0}^\top, \underbrace{\theta_{j,H} \cdot b_l \cdot f_l}_{\text{column } l}, \mathbf{0}^\top\Big). \tag{34}$$

Similarly for $(\bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{p}})$ we have

$$T_{\boldsymbol{r}^*,\cdot}(\bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top) \odot \{\boldsymbol{v}_l^\top \cdot T(\bar{\boldsymbol{\Theta}}^2)\} = \Big(\mathbf{0}^\top, \underbrace{\prod_{h \in S_1 : \Gamma_{h,l}=0} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,H})}_{\text{column } \sigma(l)}, \mathbf{0}^\top\Big), \tag{35}$$

$$T_{\boldsymbol{r}^*+\boldsymbol{e}_j,\cdot}(\bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top) \odot \{\boldsymbol{v}_l^\top \cdot T(\bar{\boldsymbol{\Theta}}^2)\} = \Big(\mathbf{0}^\top, \underbrace{\bar{\theta}_{j,H} \cdot \prod_{h \in S_1 : \Gamma_{h,l}=0} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,H})}_{\text{column } \sigma(l)}, \mathbf{0}^\top\Big). \tag{36}$$

Equation (26) implies $(33) \cdot \boldsymbol{p} = (35) \cdot \bar{\boldsymbol{p}}$. By (26), the above four equations give that

$$\theta_{j,H} = \theta_{j,l} = \frac{(34) \cdot \boldsymbol{p}}{(33) \cdot \boldsymbol{p}} = \frac{(36) \cdot \bar{\boldsymbol{p}}}{(35) \cdot \bar{\boldsymbol{p}}} = \bar{\theta}_{j,\sigma(l)} = \bar{\theta}_{j,H}, \quad \forall j \in S_2.$$

Note that the above equality $\theta_{j,H} = \bar{\theta}_{j,H}$ holds for any l and any item j such that $\Gamma_{j,l} = 1$. Therefore we have shown $\theta_{j,H} = \bar{\theta}_{j,H}$ holds for any $j \in S_1 \cup S_2$.

We next show that for any $j \in S_1 \cup S_2$ and any $l \in \{1, \ldots, L_0\}$ such that $\Gamma_{j,l} = 0$, $\theta_{j,l} = \bar{\theta}_{j,\sigma(l)}$, and show $p_l = \bar{p}_{\sigma(l)}$ for any $l \in \{1, \ldots, L_0\}$. We use an induction method to show for any $l \in [L_0]$,

$$\forall j \in S_1 \cup S_2, \quad \theta_{j,l} = \bar{\theta}_{j,\sigma(l)}, \quad p_l = \bar{p}_{\sigma(l)}. \tag{37}$$

We first introduce the *lexicographic order* between two binary vectors of the same length. For two vectors $\boldsymbol{a} = (a_1, \ldots, a_L)$ and $\boldsymbol{b} = (b_1, \ldots, b_L)$, we say $\boldsymbol{a}$ has smaller lexicographic order than $\boldsymbol{b}$ and denote by $\boldsymbol{a} \prec_{\text{lex}} \boldsymbol{b}$, if either $a_1 < b_1$, or $a_l < b_l$ for some integer $l \leq L$ and $a_m = b_m$ for all $m = 1, \ldots, l-1$. By Condition A, $\Gamma^{(S_i,\mathcal{A}_0)}$ has distinct column vectors

for $i = 1, 2$, so without loss of generality, we can assume the columns of it are sorted in an increasing lexicographic order, i.e.,

$$\Gamma^{(S_1,\mathcal{A}_0)}_{\bullet,1} \prec_{\text{lex}} \cdots \prec_{\text{lex}} \Gamma^{(S_1,\mathcal{A}_0)}_{\bullet,L_0}. \tag{38}$$

Firstly, we prove (37) hold for $l = 1$, where from (38) we have $\Gamma^{(S_1,\mathcal{A}_0)}_{\bullet,1}$ has the smallest lexicographical order among the column vectors of $\Gamma^{(S_1,\mathcal{A}_0)}$. We claim that $\Gamma^{(S_2,\mathcal{A}_0)}_{\bullet,1}$ has the smallest lexicographical order among the column vectors of $\Gamma^{(S_2,\mathcal{A}_0)}$, because otherwise "$\succeq_{S_1} = \succeq_{S_2}$" under $\mathcal{A}_0$ will not hold. For $l = 1$ we define

$$\boldsymbol{\theta}^* = \sum_{h \in S_1 : \Gamma_{h,1}=0} \theta_{h,H} \boldsymbol{e}_h,$$

and consider the row vector of the transformed T-matrix $T(\boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top)$ corresponding to $\boldsymbol{r} = \sum_{h \in S_1 : \Gamma_{h,1}=0} \boldsymbol{e}_h$ has only one potentially nonzero element in the first column, i.e.,

$$T_{\boldsymbol{r},\bullet}(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top) = \Big(\prod_{h \in S_1 : \Gamma_{h,1}=0} (\theta_{h,1} - \theta_{h,H}),\ 0, \ldots, 0\Big)$$

Then similarly for parameters $(\bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{p}})$ we have

$$T_{\boldsymbol{r},\bullet}(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top) = \Big(0, \ldots, 0, \underbrace{\prod_{h \in S_1 : \Gamma_{h,1}=0} (\bar{\theta}_{h,\sigma(1)} - \theta_{h,H})}_{\text{column } \sigma(1)},\ 0, \ldots, 0\Big)$$

and

$$\prod_{h \in S_1 : \Gamma_{h,1}=0} (\theta_{h,1} - \bar{\theta}_{h,H}) \neq 0, \qquad \prod_{h \in S_1 : \Gamma_{h,1}=0} (\bar{\theta}_{h,1} - \theta_{h,H}) \neq 0.$$

Now consider $\theta_{j,1}$ for any $j \in S_2$ and $\Gamma_{j,1} = 0$. The row vectors of $T(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top)$ and $T(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top)$ corresponding to the response pattern $\boldsymbol{r} + \boldsymbol{e}_j$ are

$$T_{\boldsymbol{r}+\boldsymbol{e}_j,\cdot}(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top) = \Big(\prod_{h \in S_1 : \Gamma_{h,1}=0} (\theta_{h,1} - \theta_{h,H}) \cdot \theta_{j,1},\ 0, \ldots, 0\Big), \tag{39}$$

and

$$T_{\boldsymbol{r}+\boldsymbol{e}_j,\cdot}(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top) = \Big(0, \ldots, 0, \underbrace{\prod_{h \in S_1 : \Gamma_{h,1}=0} (\bar{\theta}_{h,\sigma(1)} - \theta_{h,H}) \cdot \bar{\theta}_{j,\sigma(1)}}_{\text{column } \sigma(1)},\ 0, \ldots, 0\Big), \tag{40}$$

respectively. The only potentially nonzero term in the first column of (39) is indeed nonzero, because we have $\theta_{h,1} < \theta_{h,H}$ for $h \in S_1$, $\Gamma_{h,1} = 0$. Now Equation (25) implies that

$$\theta_{j,1} = \frac{T_{\boldsymbol{r}+\boldsymbol{e}_j,\bullet}(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top)\boldsymbol{p}}{T_{\boldsymbol{r},\bullet}(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top)\boldsymbol{p}} = \frac{T_{\boldsymbol{r}+\boldsymbol{e}_j,\bullet}(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top)\bar{\boldsymbol{p}}}{T_{\boldsymbol{r},\bullet}(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top)\bar{\boldsymbol{p}}} = \bar{\theta}_{j,\sigma(1)},$$

for any $j \in S_2$ and $\Gamma_{j,1} = 0$. Similarly we can obtain $\theta_{j,1} = \bar{\theta}_{j,\sigma(1)}$ for any $j \in S_1$ and $\Gamma_{j,\sigma(1)} = 0$.

After obtaining these $\bar{\theta}_{j,\sigma(1)} = \theta_{j,1}$ for $j \in (S_1 \cup S_2)$ and $\Gamma_{j,1} = 0$, the previous equations (39) and (40) just become the following,

$$T_{\boldsymbol{r}+\boldsymbol{e}_j,\cdot}(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top) = \Big(\prod_{h \in S_1 : \Gamma_{h,1}=0} (\theta_{h,1} - \theta_{h,H}) \cdot \theta_{j,1},\ 0, \ldots, 0 \Big), \tag{41}$$

$$T_{\boldsymbol{r}+\boldsymbol{e}_j,\cdot}(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top) = \Big(0, \ldots, 0, \underbrace{\prod_{h \in S_1 : \Gamma_{h,1}=0} (\theta_{h,\sigma(1)} - \theta_{h,H}) \cdot \theta_{j,\sigma(1)}}_{\text{column } \sigma(1)},\ 0, \ldots, 0 \Big). \tag{42}$$

Therefore $(41) \cdot \boldsymbol{p} = (42) \cdot \bar{\boldsymbol{p}}$ just gives $p_1 = \bar{p}_{\sigma(1)}$.

Now as the inductive hypothesis, we assume for an $l \in [L_0]$,

$$\forall \boldsymbol{\alpha}_{l'} \text{ s.t. } \boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l, \quad \forall j \in S_1 \cup S_2, \quad \theta_{j,l'} = \bar{\theta}_{j,\sigma(l')}, \quad p_{l'} = \bar{p}_{\sigma(l')}.$$

Recall that $\boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l$ if and only if $\boldsymbol{\alpha}_{l'} \preceq_{S_2} \boldsymbol{\alpha}_l$ under $\mathcal{A}_0$. Define $\boldsymbol{\theta}^*$ as

$$\boldsymbol{\theta}^* = \sum_{h \in S_1 : \Gamma_{h,l}=0} \theta_{h,H} \boldsymbol{e}_h + \sum_{h \in S_1 : \Gamma_{h,l}=1} \theta_{h,l} \boldsymbol{e}_h,$$

then for $\boldsymbol{r}^* := \sum_{h \in S_1} \boldsymbol{e}_h$ we have

$$\begin{aligned} T_{\boldsymbol{r}^*,\cdot}(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top)\boldsymbol{p} = & \sum_{\boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l} t_{\boldsymbol{r}^*,l'} \cdot p_{l'} \\ & + \prod_{h \in S_1 : \Gamma_{h,l}=0} (\theta_{h,l} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\theta_{h,l} - \theta_{h,1}) \cdot p_l, \end{aligned} \tag{43}$$

$$\begin{aligned} T_{\boldsymbol{r}^*,\cdot}(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top)\bar{\boldsymbol{p}} = & \sum_{\boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l} \bar{t}_{\boldsymbol{r}^*,\sigma(l')} \cdot \bar{p}_{\sigma(l')} \\ & + \prod_{h \in S_1 : \Gamma_{h,l}=0} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,1}) \cdot \bar{p}_{\sigma(l)}, \end{aligned} \tag{44}$$

where the notations $t_{\boldsymbol{r}^*,l'}$ and $\bar{t}_{\boldsymbol{r}^*,l'}$ are defined as

$$\begin{aligned} t_{\boldsymbol{r}^*,l'} &= \prod_{h \in S_1 : \Gamma_{h,l}=0} (\theta_{h,l'} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\theta_{h,l} - \theta_{h,1}), \\ \bar{t}_{\boldsymbol{r}^*,l'} &= \prod_{h \in S_1 : \Gamma_{h,l}=0} (\bar{\theta}_{h,\sigma(l')} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,1}). \end{aligned}$$

Note that by induction assumption we have $\theta_{h,l'} = \bar{\theta}_{h,\sigma(l')}$ for any l' such that $\boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l$ under $\mathcal{A}_0$. This implies $t_{\boldsymbol{r}^*,l'} = \bar{t}_{\boldsymbol{r}^*,\sigma(l')}$ and further implies

$$\sum_{\boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l} t_{\boldsymbol{r}^*,l'} \cdot p_{l'} = \sum_{\boldsymbol{\alpha}_{l'} \preceq_{S_1} \boldsymbol{\alpha}_l} \bar{t}_{\boldsymbol{r}^*,\sigma(l')} \cdot \bar{p}_{\sigma(l')}.$$

So (43) = (44) gives

$$\begin{aligned} & \prod_{h \in S_1 : \Gamma_{h,l}=0} (\theta_{h,l} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\theta_{h,l} - \theta_{h,1}) \cdot p_l \\ = & \prod_{h \in S_1 : \Gamma_{h,l}=0} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,1}) \cdot \bar{p}_{\sigma(l)}, \end{aligned} \tag{45}$$

and the two terms on both hand sides of the above equation are nonzero. Now consider any $j \notin S_1$ and similarly $T_{\boldsymbol{r}^*+\boldsymbol{e}_j,\cdot}(\Gamma, \boldsymbol{\Theta} - \boldsymbol{\theta}^*\mathbf{1}^\top)\boldsymbol{p} = T_{\boldsymbol{r}^*+\boldsymbol{e}_j,\cdot}(\bar{\Gamma}, \bar{\boldsymbol{\Theta}} - \boldsymbol{\theta}^*\mathbf{1}^\top)\bar{\boldsymbol{p}}$ yields

$$\begin{aligned} & \theta_{j,l} \cdot \prod_{h \in S_1 : \Gamma_{h,l}=0} (\theta_{h,l} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\theta_{h,\boldsymbol{\alpha}} - \theta_{h,1}) \cdot p_l \\ = & \bar{\theta}_{j,\sigma(l)} \cdot \prod_{h \in S_1 : \Gamma_{h,l}=0} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,H}) \prod_{h \in S_1 : \Gamma_{h,l}=1} (\bar{\theta}_{h,\sigma(l)} - \theta_{h,1}) \cdot \bar{p}_{\sigma(l)}. \end{aligned} \tag{46}$$

Taking the ratio of the above two equations (46) and (45) gives $\theta_{j,l} = \bar{\theta}_{j,\sigma(l)}, \quad \forall j \notin S_1$. Redefining $\boldsymbol{r}^* := \sum_{h \in S_2} \boldsymbol{e}_h$ similarly as above we have $\theta_{j,l} = \bar{\theta}_{j,\sigma(l)}$ for any $j \in S_1$. Plug $\theta_{j,l} = \bar{\theta}_{j,\sigma(l)}$ for all $j \in S_1$ into (45), then we have $p_l = \bar{p}_{\sigma(l)}$. Now we have shown (37) hold for this particular l. Then the induction argument gives

$$\forall l \in [L_0], \quad \forall j \in S_1 \cup S_2, \quad \theta_{j,l} = \bar{\theta}_{j,\sigma(l)}, \quad p_l = \bar{p}_{\sigma(l)}.$$

Now we have shown for any item j and latent class index l, $\theta_{j,l} = \bar{\theta}_{j,\sigma(l)}$, which we denote by $\bar{\boldsymbol{\Theta}} = \sigma(\boldsymbol{\Theta})$. We claim that this result also indicates that the permutation σ is unique. This is because $U_1 \cdot \{W_1 T(\Gamma^1, \boldsymbol{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}} = I_L$ implies that

$$U_1 \cdot \{W_1 T(\bar{\Gamma}^1, \bar{\boldsymbol{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}} = U_1 \cdot \{W_1 T(\Gamma^1, \boldsymbol{\Theta}^1)\}_{\boldsymbol{r}_{1:L_0}} \cdot \sigma(I_L) = \sigma(I_L),$$

which means given U_1 constructed from $(\Gamma, \boldsymbol{\Theta})$, the form of $U_1 \cdot \{W_1 T(\bar{\Gamma}^1, \bar{\boldsymbol{\Theta}}^1)\}_{\boldsymbol{r}_{1:L_0}}$ explicitly and uniquely determines σ. Now we have shown $\bar{\Gamma} = \Gamma = \Gamma^{\mathcal{A}_0}$ and $(\bar{\boldsymbol{\Theta}}, \bar{\boldsymbol{p}}) = (\boldsymbol{\Theta}, \boldsymbol{p})$ must hold up to the column permutation σ.

As stated in the beginning of the proof, combining Condition C that any column in $\Gamma^{\mathcal{A}_0}$ is different from any column in $\Gamma^{\mathcal{A}_0^c}$, the identification of $\Gamma^{\mathcal{A}_0}$ uniquely identifies the set of true patterns $\mathcal{A}_0$. The proof of both Theorem 2 and Corollary 3 is complete. □

Proof of Theorem 6 and Corollary 7. The following proofs of Theorem 6 and Corollary 7 use a similar proof idea as that of Allman et al. (2009); see also proofs of Theorems 4.2 and 4.3 in Gu and Xu (2019b).

Proof of Theorem 6. We need to introduce the definition of *algebraic variety*, a concept in algebraic geometry. An algebraic variety $\mathcal{V}$ is defined as the simulateneous zero-set of a finite collection of multivariate polynomials $\{f_i\}_{i=1}^n \subseteq \mathbb{R}[x_1, x_2, \ldots, x_d]$, $\mathcal{V} = \mathcal{V}(f_1, \ldots, f_n) = \{\boldsymbol{x} \in \mathbb{R}^d \mid f_i(\boldsymbol{x}) = 0,\ 1 \le i \le n.\}$ An algebraic variety $\mathcal{V}$ is all of $\mathbb{R}^d$ only when all the polynomials defining it are zero polynomials; otherwise, $\mathcal{V}$ is called a *proper subvariety* and is of dimension less than d, hence necessarily of Lebesgue measure zero in $\mathbb{R}^d$. The same

argument holds when $\mathbb{R}^d$ is replaced by the parameter space $\Omega \subseteq \mathbb{R}^d$ that has full dimension in $\mathbb{R}^d$. For the structured latent attribute model, we consider the following parameter space,

$$\Omega = \Big\{(\boldsymbol{\Theta}, \boldsymbol{p}) : \forall\, j, \max_{\boldsymbol{\alpha}:\Gamma_{j,\boldsymbol{\alpha}}=1} \theta_{j,\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}:\Gamma_{j,\boldsymbol{\alpha}}=1} \theta_{j,\boldsymbol{\alpha}} > \theta_{j,\boldsymbol{\alpha}'},\ \forall\, \Gamma_{j,\boldsymbol{\alpha}'} = 0\Big\}.$$

On Ω, altering some entries of zero to one in the Γ-matrix is equivalent to impose more affine constraints on the parameters and force them to be in a subset Ω^* of Ω. Condition $A^\star$ guarantees that, there exists a Ω^* such that Condition A holds for model parameters belonging to this Ω^*, the proof of Theorem 2 gives that the matrix $T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$ has full column rank C for $i = 1, 2$ for $(\boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)}, \boldsymbol{p}^{\mathcal{A}_0}) \in \Omega^*$. Note that the statement that $2^{|S_i|} \times C$ matrix $T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$ has full column rank is equivalent to the statement that the map sending $T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$ to all its $\binom{2^{|S_i|}}{C}$ possible $C \times C$ minors $A^i_1, A^i_2, \ldots, A^i_{2^{|S_i|}}$ yields at least one nonzero minor, where $A^i_1, A^i_2, \ldots, A^i_{2^{|S_i|}}$ are all polynomials of the item parameters $\boldsymbol{\Theta}_{S_i}$. Define

$$\mathcal{V} = \bigcup_{i=1,2} \Big\{ \bigcap_{l=1}^{2^{|S_i|}} \{(\boldsymbol{\Theta}, \boldsymbol{p}) \in \Omega : A^i_l(\boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)}) = 0\} \Big\},$$

then $\mathcal{V}$ is a algebraic variety defined by polynomials of the model parameters. Moreover, $\mathcal{V}$ is a proper subvariety of Ω, since the fact $T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$ has full column rank C for $i = 1, 2$ for one particular set of $(\boldsymbol{\Theta}, \boldsymbol{p}) \in \Omega^*$ ensures that there exists one particular set of model parameters that give nonzero values when plugged into the polynomials defining $\mathcal{V}$. This indicates that the polynomials defining $\mathcal{V}$ are not all zero polynomials on Ω. Then restricting parameters to Ω^* and proceeding in the same steps as the proof of Theorem 2 proves the conclusion of the proposition.

Proof of Corollary 7. Consider a Q-matrix in the form of (13). We denote $S_1 = \{1, \ldots, K\}$, $S_2 = \{K+1, \ldots, 2K\}$ and $S_3 = \{2K+1, \ldots, J\}$, which are item sets corresponding to Q_1, Q_2 and Q', respectively. According to the proof of Theorem 4.3 in Gu and Xu (2019b), since the two submatrices Q_1 and Q_2 have all the diagonal elements equal to one, the $2^K \times 2^K$ T-matrices $T(\Gamma^{(S_1,\text{all})}, \boldsymbol{\Theta}^{(S_1,\text{all})})$ and $T(\Gamma^{(S_2,\text{all})}, \boldsymbol{\Theta}^{(S_2,\text{all})})$ are generically full-rank. Furthermore, the matrix $T(\Gamma^{(S_3,\text{all})}, \boldsymbol{\Theta}^{(S_3,\text{all})}) \cdot \text{Diag}(\boldsymbol{p}^{\text{all}})$ has Kruskal rank at least two. This means generically, any two columns of $T(\Gamma^{(S_3,\text{all})}, \boldsymbol{\Theta}^{(S_3,\text{all})}) \cdot \text{Diag}(\boldsymbol{p}^{\text{all}})$ are linearly independent.

Now consider an arbitrary set of attribute patterns $\mathcal{A}_0 \subseteq \{0, 1\}$, we have the conclusion that $T(\Gamma^{(S_1,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_1,\mathcal{A}_0)})$ and $T(\Gamma^{(S_2,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_2,\mathcal{A}_0)})$ have full column rank generically. This is because for $i = 1, 2$, the $T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$ is just a submatrix of $T(\Gamma^{(S_i,\text{all})}, \boldsymbol{\Theta}^{(S_i,\text{all})})$ whose columns are a subset of different column vectors of the latter matrix. Therefore columns of $T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$ must be linearly independent, and hence the matrix must have full column rank generically. Also, the columns of $T(\Gamma^{(S_3,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_3,\mathcal{A}_0)}) \cdot \text{Diag}(\boldsymbol{p}^{\mathcal{A}_0})$ can also be considered as a subset of different columns of $T(\Gamma^{(S_3,\text{all})}, \boldsymbol{\Theta}^{(S_3,\text{all})}) \cdot \text{Diag}(\boldsymbol{p}^{\text{all}})$ up to a resealing of the columns. Therefore the former matrix must have any two different columns linearly independent generically and hence has Kruskal rank at least two. Now by Kruskal's conditions for unique tensor decomposition, a probability distribution of $\mathbf{R}$ with $T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$, $i = 1, 2, 3$ having the above properties uniquely determines

$T(\Gamma^{(S_i,\mathcal{A}_0)}, \boldsymbol{\Theta}^{(S_i,\mathcal{A}_0)})$ and also $\boldsymbol{p}^{\mathcal{A}_0}$ generically. Therefore $(\Gamma^{\mathcal{A}_0}, \boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ are generically identifiable. Then combined with Condition C, we have the conclusion that $\mathcal{A}_0$ is generically identifiable. This completes the proof of the corollary. □

Proof of Corollary 10. Under our definition of $\mathcal{A}^{\text{rep}}$ and also Condition C, this matrix must have distinct column vectors, and each of its column corresponds to an equivalence class. We define $\boldsymbol{\Theta}^{\text{rep}}$ to be item parameters corresponding to the representative patterns in $\mathcal{A}^{\text{rep}}$. We further define the proportion parameters of the equivalence classes $\boldsymbol{\nu}^{\text{rep}} = (\nu_{[\boldsymbol{\alpha}_{\ell_1}]}, \ldots, \nu_{[\boldsymbol{\alpha}_{\ell_m}]})$, where $\nu_{[\boldsymbol{\alpha}_{\ell_i}]} > 0$ and $\sum_{i=1}^m \nu_{[\boldsymbol{\alpha}_{\ell_i}]} = 1$. Note that each $\nu_{[\boldsymbol{\alpha}_{\ell_i}]}$ is a sum of population proportions of the attribute patterns that are in the same equivalence class of $\boldsymbol{\alpha}_{\ell_i}$. Since $\Gamma^{\mathcal{A}^{\text{rep}}}$ also satisfies Conditions A and B by the assumption of the corollary. So Theorem 2 gives that $\mathcal{A}^{\text{rep}}$ is identifiable. □

Proof of Theorem 13 and Proposition 15. We use $L = |\mathcal{A}_{\text{input}}|$ to denote the number of attribute patterns as input given to the penalized likelihood method, then $L = 2^K$ if there is no screening stage as preprocessing. We denote the true proportion parameters by $\boldsymbol{p} = (p_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}_{\text{input}})$, where $p_{\boldsymbol{\alpha}} \geq 0$ for $\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}$ and $\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} p_{\boldsymbol{\alpha}} = 1$. Denote the number of true attribute patterns by $|\mathcal{A}_0|$. We now consider the following log likelihood with penalty parameter λ_N for some $\gamma > 0$,

$$\begin{aligned}\ell^{\lambda_N}(\boldsymbol{p}, \boldsymbol{\Theta}) =& \frac{1}{N} \sum_{i=1}^N \log \Big\{ \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} p_{\boldsymbol{\alpha}} \prod_{j=1}^J \theta_{j,\boldsymbol{\alpha}}^{R_{i,j}} (1 - \theta_{j,\boldsymbol{\alpha}})^{1-R_{i,j}} \Big\} \\ &+ \frac{\lambda_N}{N} \underbrace{\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} \Big[\log p_{\boldsymbol{\alpha}} \cdot I(p_{\boldsymbol{\alpha}} > \rho_N) + \log \rho_N \cdot I(p_{\boldsymbol{\alpha}} \leq \rho_N) \Big]}_{\log_{\rho_N}(\boldsymbol{p})}.\end{aligned}$$

For a given λ_N, denote the estimated support of the proportion parameters $\widehat{\boldsymbol{p}}$ by $\widehat{\mathcal{A}}$, namely $\widehat{\mathcal{A}} = \{1 \leq l \leq L : \widehat{p}_{\boldsymbol{\alpha}_l} > \rho_N\}$. We denote the true and the estimated $|\mathcal{A}_{\text{input}}|$-dimensional proportions by $\boldsymbol{p}^{\mathcal{A}_0}_{\text{full}} = (p_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}_{\text{input}} : p_{\boldsymbol{\alpha}} > 0$ if and only if $\boldsymbol{\alpha} \in \mathcal{A}_0)$ and $\widehat{\boldsymbol{p}}^{\widehat{\mathcal{A}}}_{\text{full}} = (\widehat{p}_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}_{\text{input}} : \widehat{p}_{\boldsymbol{\alpha}} > \rho_N$ if and only if $\boldsymbol{\alpha} \in \widehat{\mathcal{A}})$. Denote the oracle MLE obtained assuming $\mathcal{A}_0$ is known by $\widehat{\boldsymbol{\Theta}}^0 := \widehat{\boldsymbol{\Theta}}^{\mathcal{A}_0}$ and $\widehat{\boldsymbol{p}}^0 := \widehat{\boldsymbol{p}}^{\mathcal{A}_0}$, and denote $\widehat{\boldsymbol{\eta}}^0 = (\widehat{\boldsymbol{\Theta}}^0, \widehat{\boldsymbol{p}}^0)$. Note that for $\widehat{\mathcal{A}} \neq \mathcal{A}_0$ the event $\{\ell^{\lambda_N}(\widehat{\boldsymbol{\eta}}^{\widehat{\mathcal{A}}}) > \ell^{\lambda_N}(\widehat{\boldsymbol{\eta}}^0)\}$ implies the following event

$$\begin{aligned}&\frac{1}{N} \sum_{i=1}^N \log \left[\frac{\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} \widehat{p}_{\boldsymbol{\alpha}} \prod_j \widehat{\theta}_{j,\boldsymbol{\alpha}}^{R_{i,j}} (1 - \widehat{\theta}_{j,\boldsymbol{\alpha}})^{1-R_{i,j}}}{\sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} \widehat{p}^0_{\boldsymbol{\alpha}} \prod_j (\widehat{\theta}^0_{j,\boldsymbol{\alpha}})^{R_{i,j}} (1 - \widehat{\theta}^0_{j,\boldsymbol{\alpha}})^{1-R_{i,j}}} \right] \\ &> \frac{|\lambda_N|}{N} \Big\{ \log_{\rho_N}(\widehat{\boldsymbol{p}}^{\widehat{\mathcal{A}}}_{\text{full}}) - \log_{\rho_N}(\widehat{\boldsymbol{p}}^{\mathcal{A}_0}_{\text{full}}) \Big\}.\end{aligned} \tag{47}$$

In the case of $|\widehat{\mathcal{A}}| > |\mathcal{A}_0|$ (which we call the overfitted case), the right hand side (RHS) of (47) regarding the difference between the penalty terms has order $O(N^{-1}|\lambda_N| \cdot |\mathcal{A}_0| \cdot |\log \rho_N|)$. In this overfitted case, we now consider the left hand side (LHS) of (47),

$$\text{LHS of (47)} = \frac{1}{N} \sum_{i=1}^N \log \left[\sum_{\boldsymbol{\alpha} \in \mathcal{A}_{\text{input}}} \widehat{p}_{\boldsymbol{\alpha}} \prod_j \widehat{\theta}_{j,\boldsymbol{\alpha}}^{R_{i,j}} (1 - \widehat{\theta}_{j,\boldsymbol{\alpha}})^{1-R_{i,j}} \right]$$

$$-\frac{1}{N}\sum_{i=1}^{N}\log\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}_0}\widehat{p}^0_{\boldsymbol{\alpha}}\prod_j(\widehat{\theta}^0_{j,\boldsymbol{\alpha}})^{R_{i,j}}(1-\widehat{\theta}^0_{j,\boldsymbol{\alpha}})^{1-R_{i,j}}\right]\equiv I_1-I_0,$$

where the I_1 part can be written as

$$\begin{aligned}I_1&=\frac{1}{N}\sum_{i=1}^{N}\log\Big[\sum_{\substack{\boldsymbol{\alpha}\in\mathcal{A}_{\text{input}},\\ \widehat{p}_{\boldsymbol{\alpha}}>\rho_N}}\widehat{p}_{\boldsymbol{\alpha}}\prod_j\widehat{\theta}^{R_{i,j}}_{j,\boldsymbol{\alpha}}(1-\widehat{\theta}_{j,\boldsymbol{\alpha}})^{1-R_{i,j}}\Big]+O(|\mathcal{A}_{\text{input}}|\rho_N)\\&=\frac{1}{N}\sum_{i=1}^{N}\log\Big[\sum_{\substack{\boldsymbol{\alpha}\in\mathcal{A}_{\text{input}},\\ \widehat{p}_{\boldsymbol{\alpha}}>\rho_N}}\widehat{p}_{\boldsymbol{\alpha}}\prod_j\widehat{\theta}^{R_{i,j}}_{j,\boldsymbol{\alpha}}(1-\widehat{\theta}_{j,\boldsymbol{\alpha}})^{1-R_{i,j}}\Big]+O(N^{-\delta}),\end{aligned}\tag{48}$$

where the last equality follows from the assumption $|\mathcal{A}_{\text{input}}|\cdot\rho_N=O(N^{-\delta})$ in the theorem. So we further have the LHS of (47) equal to

$$\begin{aligned}I_1-I_0=&\frac{1}{N}\sum_{i=1}^{N}\log\Big[\sum_{\substack{\boldsymbol{\alpha}\in\mathcal{A}_{\text{input}},\\ \widehat{p}_{\boldsymbol{\alpha}}>\rho_N}}\widehat{p}_{\boldsymbol{\alpha}}\prod_j\widehat{\theta}^{R_{i,j}}_{j,\boldsymbol{\alpha}}(1-\widehat{\theta}_{j,\boldsymbol{\alpha}})^{1-R_{i,j}}\Big]\\&-\frac{1}{N}\sum_{i=1}^{N}\log\Big[\sum_{\boldsymbol{\alpha}\in\mathcal{A}_0}\widehat{p}^0_{\boldsymbol{\alpha}}\prod_j(\widehat{\theta}^0_{j,\boldsymbol{\alpha}})^{R_{i,j}}(1-\widehat{\theta}^0_{j,\boldsymbol{\alpha}})^{1-R_{i,j}}\Big]+O(N^{-\delta}).\end{aligned}$$

Note that other than the last term $O(N^{-\delta})$ in the above display, the difference of the first two terms also has order $O_p(N^{-\delta})$ from assumption (19), so LHS of (47)$=I_1-I_0=O_p(N^{-\delta})$. In order to have selection consistency in the overfitted case, we need the event described in (47) to happen with probability tending to zero, so the $|\lambda_N|$ needs to be sufficiently large such that

$$N^{-\delta}\lesssim O\Big(N^{-1}|\lambda_N|\cdot|\mathcal{A}_0|\cdot|\log\rho_N|\Big).\tag{49}$$

Note that by (18), we have $\rho_N\asymp N^{-d}$ for some $d>0$. So if $\delta<1$, i.e., if the convergence rate is slower than the $\sqrt{N}$ rate, then λ_N must go to negative infinity as N goes to infinity since $\delta<1$. Specifically, we obtain the following lower bound of the magnitude of the penalty parameter λ_N,

$$|\lambda_N|\gtrsim N^{1-\delta}/|\log\rho_N|$$

would suffice for (49) to hold.

We now prove the conclusion of Proposition 15. A further implication of the above discussion is that, with $\rho_N\asymp N^{-d}$ as assumed in (18), just imposing a proper Dirichlet prior with a positive hyperparameter would fail to select the true model consistently. In particular, with a proper Dirichlet prior density with hyperparameter $\beta=\lambda_N+1\in(0,1)$, Equation (49) instead becomes $N^{1-\delta}=o(\log N)$. However, when $0<\delta<1$, $N^{1-\delta}/\log N\to\infty$. So (49) fails to hold, and one can not have consistent selection in the overfitted case. So if we denote the set of attribute patterns estimated by maximizing (17) by $\widehat{\mathcal{A}}^{\lambda}$. Then for any $\{\lambda_N\}\subseteq[-1,0)$, $\mathbb{P}(\widehat{\mathcal{A}}^{\lambda}=\mathcal{A}_0)\not\to 1$ as $N\to\infty$. This proves Proposition 15.

Now we consider the random set $\{\boldsymbol{\alpha}\in\mathcal{A}_{\text{input}}:\widehat{p}_{\boldsymbol{\alpha}}>\rho_N\}=:\widehat{\mathcal{A}}$ appearing in I_1 in (48). With probability tending to one, the cardinality of this set is smaller than $|\mathcal{A}_0|$. This is

because if $|\widehat{\mathcal{A}}|>|\mathcal{A}_0|$, the log-penalty term corresponding to $\widehat{\mathcal{A}}$ would be smaller than that corresponding to $\mathcal{A}_0$ by $N^{-1}|\lambda_N|\cdot|\log\rho_N|$ which has order at least $N^{-\delta}$. Recall that the right hand side of (47) has order $O_P(N^{-\delta})$, which means when $|\widehat{\mathcal{A}}|>|\mathcal{A}_0|$ the extent that the log-penalty part favors the a smaller model $\mathcal{A}_0$ would dominate the extent that the likelihood part favors a larger model $\widehat{\mathcal{A}}$ in the proposed penalized likelihood. Therefore any larger model $\widehat{\mathcal{A}}$ with $|\widehat{\mathcal{A}}|\geq|\mathcal{A}_0|$ would be favored over $\mathcal{A}_0$ with probability tending to zero. Therefore we have the conclusion that $\mathbb{P}(\widehat{\mathcal{A}}\neq\mathcal{A}_0)\not\to 0$ could only happen for $|\widehat{\mathcal{A}}|\leq|\mathcal{A}_0|$. So in the following discussion we will focus on the case where $|\widehat{\mathcal{A}}|\leq|\mathcal{A}_0|$ and prove consistency in this case. Namely, we aim to bound

$$\mathbb{P}\Big(\sup_{|\widehat{\mathcal{A}}|\leq|\mathcal{A}_0|,\widehat{\mathcal{A}}\neq\mathcal{A}_0}[\ell^{\lambda_N}(\boldsymbol{\eta}^{\widehat{\mathcal{A}}})-\ell^{\lambda_N}(\boldsymbol{\eta}^{\mathcal{A}_0})]>0\Big).\tag{50}$$

Next, we consider the upper bound of the magnitude of the penalty term. In order to have selection consistency in the case of $|\widehat{\mathcal{A}}|\leq|\mathcal{A}_0|$ and $\widehat{\mathcal{A}}\neq\mathcal{A}_0$, the log-penalty term can not be too large such that the extent that the penalty part favors a smaller model does not dominate the extent that the likelihood part favors the true model. We follow a similar argument to Shen et al. (2012). Specifically, considering the term $-\epsilon_N^2\to 0$ in the large deviation inequality (53) below; for a small constant $t>\epsilon_N$, we need that the difference of the penalty part of the true and any alternative smaller model to be less than t^2, i.e.,

$$|\lambda_N|\cdot|\mathcal{A}_0|\cdot|\log\rho_N|/N\lesssim t^2,\tag{51}$$

Equation (51) would hold if

$$|\lambda_N|=o(N/|\log\rho_N|).\tag{52}$$

We next show that such λ_N can guarantee selection consistency. So we have a sample-size dependent λ_N that penalizes the overfitted mixture and constrains the support size of the proportion parameters to be less than the true support size $|\mathcal{A}_0|$. As said, with such λ_N it suffices to consider the case $|\widehat{\mathcal{A}}|\leq|\mathcal{A}_0|$.

In order to bound this mis-selection probability, we need to introduce the notion of bracketing Hellinger metric entropy $H(t,\mathcal{B}_{\mathcal{A}})$. Let $h(\boldsymbol{\eta}^{\mathcal{A}},\boldsymbol{\eta}^{\mathcal{A}_0})$ denote the Hellinger distance between the probability mass functions of $\mathbf{R}$ indexed by $\boldsymbol{\eta}^{\mathcal{A}}$ and $\boldsymbol{\eta}^{\mathcal{A}_0}$, i.e.,

$$h(\boldsymbol{\eta}^{\mathcal{A}},\boldsymbol{\eta}^{\mathcal{A}_0})=\Big(\sum_{\boldsymbol{r}\in\{0,1\}^J}\Big[\mathbb{P}(\mathbf{R}=\boldsymbol{r}\mid\boldsymbol{\Theta}^{\mathcal{A}},\boldsymbol{p}^{\mathcal{A}})^{\frac{1}{2}}-\mathbb{P}(\mathbf{R}=\boldsymbol{r}\mid\boldsymbol{\Theta}^{\mathcal{A}_0},\boldsymbol{p}^{\mathcal{A}_0})^{\frac{1}{2}}\Big]\Big)^{\frac{1}{2}}.$$

Consider the local parameter space $\mathcal{B}_{\mathcal{A}}=\{\boldsymbol{\eta}^{\mathcal{A}}=(\boldsymbol{\Theta}^{\mathcal{A}},\boldsymbol{p}^{\mathcal{A}}):|\mathcal{A}|\leq|\mathcal{A}_0|,h^2(\boldsymbol{\eta}^{\mathcal{A}},\boldsymbol{\eta}^{\mathcal{A}_0})\leq 2\epsilon_N^2\}$, the $H(t,\mathcal{B}_{\mathcal{A}})$ is defined as the logarithm of the cardinality of the t-bracketing of $\mathcal{B}_{\mathcal{A}}$ of the smallest size. More specifically, following the definition in Shen et al. (2012), consider a bracket covering $S(t,m)=\{f_1^l,f_1^u,\ldots,f_m^l,f_m^u\}$ satisfying that $\max_{1\leq j\leq m}\|f_j^u-f_j^l\|_2\leq t$ and for any $f\in\mathcal{B}_{\mathcal{A}}$ there is some j such that $f_j^l\leq f\leq f_j^u$ almost surely. Then $H(t,\mathcal{B}_{\mathcal{A}})$ is $\log(\min\{m:S(t,m)\})$. The $H(t,\mathcal{B}_{\mathcal{A}})$ measures the complexity of the local parameter space. The next lemma gives an upper bound for the bracketing Hellinger metric entropy $H(t,\mathcal{B}_{\mathcal{A}})$ for $|\mathcal{A}|\leq|\mathcal{A}_0|$.

Lemma 21 *Denote $N_{[]}(t,\mathcal{B}_{\mathcal{A}})=\exp(H(t,\mathcal{B}_{\mathcal{A}}))$. For the considered structured latent attribute model, denote the item parameter space of the ℓ-th attribute pattern by $\mathcal{F}_\ell$. For $|\mathcal{A}|\leq|\mathcal{A}_0|$ and any $2^{-4}\epsilon<t<\epsilon$, there is $H(t,\mathcal{B}_{\mathcal{A}})\lesssim|\mathcal{A}_0|\log|\mathcal{A}_{\text{input}}|\log(2\epsilon/t)$.*

By the assumption of the theorem there is $\log|\mathcal{A}_{\text{input}}|/N \to 0$, so if we take

$$\epsilon_N = \sqrt{1/N|\mathcal{A}_0|\log|\mathcal{A}_{\text{input}}|},$$

there is $\epsilon_N = o(1)$. We next verify the entropy integral condition in Theorem 1 of Wong and Shen (1995) is satisfied with this ϵ_N, in order to obtain a large deviation inequality to bound the mis-selection probability. With Lemma 21, the integral of bracketing Hellinger metric entropy in the interval $[2^{-8}\epsilon_N^2, \sqrt{2}\epsilon_N]$ satisfies the following inequality

$$\begin{aligned}\int_{2^{-8}\epsilon_N^2}^{\sqrt{2}\epsilon_N} H^{1/2}(t,\mathcal{B}_{\mathcal{A}})dt &\le \int_{2^{-8}\epsilon_N^2}^{\sqrt{2}\epsilon_N} \sqrt{|\mathcal{A}_0|\log|\mathcal{A}_{\text{input}}|\log(2\epsilon_N/t)}dt \\ &= \sqrt{|\mathcal{A}_0|\log|\mathcal{A}_{\text{input}}|}\int_{\sqrt{\log\sqrt{2}}}^{\sqrt{\log\frac{2^9}{\epsilon_N}}} 4\epsilon_N u^2 e^{-u^2}du \\ &= \sqrt{|\mathcal{A}_0|\log|\mathcal{A}_{\text{input}}|}\cdot 2\epsilon_N \underbrace{\int_{\log\sqrt{2}}^{\log\frac{2^9}{\epsilon_N}} \sqrt{u}e^{-u}du}_{\text{bounded as } \epsilon_N\to 0} \lesssim \sqrt{N}\epsilon_N^2.\end{aligned}$$

So the entropy integral condition in Theorem 1 in Wong and Shen (1995) is satisfied and the large deviation inequality there holds. In particular, we have

$$\begin{aligned}&\mathbb{P}\Big(\sup_{h^2(\widehat{\boldsymbol{\eta}}^{\widehat{\mathcal{A}}},\boldsymbol{\eta}^{\mathcal{A}_0})\ge\epsilon_N^2}\Big[\frac{1}{N}\ell(\widehat{\boldsymbol{\eta}}^{\widehat{\mathcal{A}}}) - \frac{1}{N}\ell(\widehat{\boldsymbol{\eta}}^{\mathcal{A}_0})\Big] > -\epsilon_N^2\Big) \\ &\le \mathbb{P}\Big(\sup_{h^2(\widehat{\boldsymbol{\eta}}^{\widehat{\mathcal{A}}},\boldsymbol{\eta}^{\mathcal{A}_0})\ge\epsilon_N^2}\Big[\frac{1}{N}\ell(\widehat{\boldsymbol{\eta}}^{\widehat{\mathcal{A}}}) - \frac{1}{N}\ell(\boldsymbol{\eta}^{\mathcal{A}_0})\Big] > -\epsilon_N^2\Big) \le \exp(-N\epsilon_N^2). \qquad (53)\end{aligned}$$

where $\boldsymbol{\eta}^{\mathcal{A}_0} = (\boldsymbol{\Theta}^{\mathcal{A}_0}, \boldsymbol{p}^{\mathcal{A}_0})$ denote the true parameters. Indeed, Theorem 1 in Wong and Shen (1995) guarantees the inequality (53) holds with ϵ_N replaced by any $t > \epsilon_N = \sqrt{|\mathcal{A}_0|\log|\mathcal{A}_{\text{input}}|/N}$. This large deviation inequality will be used later to bound the mis-selection probability in the case of $|\mathcal{A}| \le |\mathcal{A}_0|$.

We next further look at the Hellinger distance between $\boldsymbol{\eta}^0 := \boldsymbol{\eta}^{\mathcal{A}_0}$ and $\boldsymbol{\eta}^{\mathcal{A}}$ for $|\mathcal{A}| \le |\mathcal{A}_0|$, and investigate how the distance between a set of true patterns $\mathcal{A}_0$ and an alternative set relate to identifiability of $\mathcal{A}_0$.

$$\begin{aligned}&\frac{h^2(\boldsymbol{\eta}^{\mathcal{A}},\boldsymbol{\eta}^{\mathcal{A}_0})}{\max(|\mathcal{A}_0\setminus\mathcal{A}|,1)} \\ &\asymp [\max(|\mathcal{A}_0\setminus\mathcal{A}|,1)]^{-1}\sum_{\boldsymbol{r}\in\{0,1\}^J}\Big[\Big(\sum_{\boldsymbol{\alpha}\in\mathcal{A}}\mathbb{P}(\mathbf{R}=\boldsymbol{r}\mid\boldsymbol{\Theta}^{\mathcal{A}},\mathbf{A}=\boldsymbol{\alpha})p_{\boldsymbol{\alpha}}^{\mathcal{A}}\Big)^{1/2} - \\ &\qquad\Big(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_0}\mathbb{P}(\mathbf{R}=\boldsymbol{r}\mid\boldsymbol{\Theta}^{\mathcal{A}_0},\mathbf{A}=\boldsymbol{\alpha})p_{\boldsymbol{\alpha}}^{\mathcal{A}_0}\Big)^{1/2}\Big]^2 \\ &\asymp [\max(|\mathcal{A}_0\setminus\mathcal{A}|,1)]^{-1}\sum_{\boldsymbol{r}\in\{0,1\}^J}\Big(\sum_{\boldsymbol{\alpha}\in\mathcal{A}}T_{\boldsymbol{r},\boldsymbol{\alpha}}(\boldsymbol{\Theta}^{\mathcal{A}})\,p_{\boldsymbol{\alpha}}^{\mathcal{A}} - \sum_{\boldsymbol{\alpha}\in\mathcal{A}_0}T_{\boldsymbol{r},\boldsymbol{\alpha}}(\boldsymbol{\Theta}^{\mathcal{A}_0})\,p_{\boldsymbol{\alpha}}^{\mathcal{A}_0}\Big)^2\end{aligned}$$

$$= [\max(|\mathcal{A}_0\setminus\mathcal{A}|,1)]^{-1}\|T(\Gamma^{\mathcal{A}},\boldsymbol{\Theta}^{\mathcal{A}})\boldsymbol{p}^{\mathcal{A}} - T(\Gamma^{\mathcal{A}_0},\boldsymbol{\Theta}^{\mathcal{A}_0})\boldsymbol{p}^{\mathcal{A}_0}\|_2^2$$

To proceed with the proof, we need to use Theorem 2 to establish an identifiability argument. Theorem 2 and Corollary 3 state that if the true constraint matrix $\Gamma^{\mathcal{A}_0}$ satisfies conditions A, B and C, then $(\Gamma^{\mathcal{A}_0},\ \boldsymbol{\Theta}^{\mathcal{A}_0},\ \boldsymbol{p}^{\mathcal{A}_0})$ are jointly identifiable. This implies that given the set of true attribute patterns $\mathcal{A}_0$, for any other set $\mathcal{A}\ne\mathcal{A}_0$, $|\mathcal{A}|\le|\mathcal{A}_0|$, and model parameters defined by $\mathcal{A}$ must lead to different $T(\boldsymbol{\Theta}^{\mathcal{A}})\boldsymbol{p}^{\mathcal{A}}$ that is different from $T(\boldsymbol{\Theta}^{\mathcal{A}_0})\boldsymbol{p}^{\mathcal{A}_0}$. Moreover, consider the parameter space $\mathcal{B} = \{(\boldsymbol{\Theta}^{\mathcal{A}},\boldsymbol{p}^{\mathcal{A}}):\ |\mathcal{A}|\le|\mathcal{A}_0|,\ p_{\boldsymbol{\alpha}} > \rho_N\ \forall\boldsymbol{\alpha}\in\mathcal{A}\}$. Then $(\boldsymbol{\Theta}^{\mathcal{A}_0},\boldsymbol{p}^{\mathcal{A}_0})\in\mathcal{B}$ and for any $(\boldsymbol{\Theta}^{\mathcal{A}},\boldsymbol{p}^{\mathcal{A}})\in\mathcal{B}$ with $\mathcal{A}\ne\mathcal{A}_0$, **either** some elements in $\boldsymbol{\Theta}^{\mathcal{A}}$ differs from those in $\boldsymbol{\Theta}^{\mathcal{A}_0}$ by a nonzero constant, **or** some elements in $\boldsymbol{p}^{\mathcal{A}}$ differs from those in $\boldsymbol{p}^{\mathcal{A}_0}$ by a nonzero constant. Since $T^{\mathcal{A}}(\boldsymbol{\Theta})\boldsymbol{p}^{\mathcal{A}}$ is a continuous vector-valued function of the model parameters, we must have $[\max(|\mathcal{A}_0\setminus\mathcal{A}|,1)]^{-1}\|T^{\mathcal{A}}(\boldsymbol{\Theta})\boldsymbol{p}^{\mathcal{A}} - T^{\mathcal{A}_0}(\boldsymbol{\Theta})\boldsymbol{p}^{\mathcal{A}_0}\|_2^2 \ge C_0$ for some $C_0 > 0$. By the conditions of the theorem $\epsilon_N^2 = o(1)$, so we have obtained for some small constant $t > \epsilon_N$,

$$C_{\min}(\boldsymbol{\eta}^0) \equiv \inf_{\boldsymbol{\eta}^{\mathcal{A}}:\,\mathcal{A}\ne\mathcal{A}_0,|\mathcal{A}|\le|\mathcal{A}_0|}\Big\{\frac{h^2(\boldsymbol{\eta}^{\mathcal{A}},\boldsymbol{\eta}^{\mathcal{A}_0})}{\max(|\mathcal{A}_0\setminus\mathcal{A}|,1)}\Big\} \ge C_0 \gtrsim t^2 > \epsilon_N^2. \qquad (54)$$

Finally, with the λ_N of the previously specified order, we use the large deviation inequality (53) and also the (54) to bound the false selection probability (50). The following argument uses a similar proof idea as that of Theorem 1 in Shen et al. (2012) which establishes finite sample mis-selection error bound of the L_0-constrained maximum likelihood estimation. Consider $|\widehat{\mathcal{A}}\cap\mathcal{A}_0| = m \le |\mathcal{A}_0|-1$, by (54) we have $h^2(\boldsymbol{\eta}^{\mathcal{A}},\boldsymbol{\eta}^{\mathcal{A}_0}) \ge (|\mathcal{A}_0|-m)C_{\min}(\boldsymbol{\eta}^0)$. So

$$\begin{aligned}&\mathbb{P}\Big(\sup_{|\widehat{\mathcal{A}}|\le|\mathcal{A}_0|,\widehat{\mathcal{A}}\ne\mathcal{A}_0}\Big[\frac{1}{N}\ell^{\lambda_N}(\boldsymbol{\eta}^{\widehat{\mathcal{A}}}) - \frac{1}{N}\ell^{\lambda_N}(\boldsymbol{\eta}^0)\Big] > 0\Big) \\ &\le \sum_{m=0}^{|\mathcal{A}_0|-1}\sum_{j=1}^{|\mathcal{A}_0|-m}\mathbb{P}\Big(\sup_{\substack{h^2(\boldsymbol{\eta}^{\mathcal{A}},\boldsymbol{\eta}^{\mathcal{A}_0})\ge\\(|\mathcal{A}_0|-m)C_{\min}(\boldsymbol{\eta}^0)}}\frac{1}{N}\Big[\ell(\boldsymbol{\eta}^{\widehat{\mathcal{A}}}) - \ell(\boldsymbol{\eta}^0)\Big] > -\frac{|\lambda_N|\cdot|\mathcal{A}_0|\cdot|\log\rho_N|}{N}\Big) \\ &\le \sum_{m=0}^{|\mathcal{A}_0|-1}\sum_{j=1}^{|\mathcal{A}_0|-m}\mathbb{P}\Big(\sup_{|\widehat{\mathcal{A}}\cap\mathcal{A}_0|=m}\frac{1}{N}\Big[\ell(\boldsymbol{\eta}^{\widehat{\mathcal{A}}}) - \ell(\boldsymbol{\eta}^0)\Big] > -t^2\Big) \quad \text{(by (52))} \\ &\le \sum_{m=0}^{|\mathcal{A}_0|-1}\sum_{j=1}^{|\mathcal{A}_0|-m}\mathbb{P}\Big(\sup_{|\widehat{\mathcal{A}}\cap\mathcal{A}_0|=m}\frac{1}{N}\Big[\ell(\boldsymbol{\eta}^{\widehat{\mathcal{A}}}) - \ell(\boldsymbol{\eta}^0)\Big] > -(|\mathcal{A}_0|-m)C_{\min}(\boldsymbol{\eta}^0)\Big) \quad \text{(by (54))} \\ &\le \sum_{m=0}^{|\mathcal{A}_0|-1}\binom{|\mathcal{A}_0|}{m}\exp\Big(-c_2N(|\mathcal{A}_0|-m)C_{\min}(\boldsymbol{\eta}^0)\Big)\sum_{j=1}^{|\mathcal{A}_0|-m}\binom{|\mathcal{A}_{\text{input}}|-|\mathcal{A}_0|}{j} \quad \text{(by (53))} \\ &\le c_3\exp\Big(-c_2NC_{\min}(\boldsymbol{\eta}^0) + 2\log(|\mathcal{A}_{\text{input}}|+1)\Big),\end{aligned}$$

where the last but one line above uses the large deviation inequality in (53), and c_2, c_3 are some constants. And the last line follows from the calculations in the proof of Theorem 1 in Shen et al. (2012) using some basic inequalities about binomial coefficients. Since $C_{\min}(\boldsymbol{\eta}^0) \ge C_0$, and $\log|\mathcal{A}_{\text{input}}| = o(N)$ by the assumption of the theorem, the right hand

side of the above display goes to zero as $N \to \infty$. Therefore $\mathbb{P}(\widehat{\mathcal{A}}^{\lambda_N} \neq \mathcal{A}_0,\ |\widehat{\mathcal{A}}^{\lambda_N}| \leq |\mathcal{A}_0|) \to 0$ as $N \to \infty$. Combined with the previously shown result $\mathbb{P}(\widehat{\mathcal{A}}^{\lambda_N} \neq \mathcal{A}_0) \not\to 0$ could potentially happen only for $|\widehat{\mathcal{A}}^{\lambda_N}| \leq |\mathcal{A}_0|$, we have the conclusion $\mathbb{P}(\widehat{\mathcal{A}}^{\lambda_N} \neq \mathcal{A}_0) \to 0$ as $N \to \infty$. The proof of the theorem is complete. □

Proof of Theorem 17. Denote $\theta_j^+ = \theta_{j,H}$ and $\theta_j^- = \max_{\boldsymbol{\alpha} \not\succeq \boldsymbol{q}_j} \theta_{j,\boldsymbol{\alpha}}$ for each j. Since the screening algorithm is developed for the two-parameter SLAM introduced in Example 2, for each item j there are exactly two estimated item parameters, and we denote them by $\widehat{\theta}_j^+$ and $\widehat{\theta}_j^-$. We claim that it suffices to prove that for any $\boldsymbol{\alpha} \in \mathcal{A}_0$, there exists a response pattern $\boldsymbol{r}^{\boldsymbol{\alpha}} \in \{0,1\}^J$ such that as $K \to \infty$,

$$\mathbb{P}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}},\ \mathbf{A} = \boldsymbol{\alpha} \mid \boldsymbol{\Theta}) > \mathbb{P}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}},\ \mathbf{A} = \widetilde{\boldsymbol{\alpha}} \mid \boldsymbol{\Theta}), \quad \forall \widetilde{\boldsymbol{\alpha}} \neq \boldsymbol{\alpha}. \tag{55}$$

For $\boldsymbol{\alpha} \in \mathcal{A}_0$, define $\boldsymbol{r}^{\boldsymbol{\alpha}} = (r_1^{\boldsymbol{\alpha}}, \ldots, r_J^{\boldsymbol{\alpha}})$ to be $r_j^{\boldsymbol{\alpha}} = I(\boldsymbol{\alpha} \succeq \boldsymbol{q}_j) = \prod_k \alpha_k^{q_{j,k}}$. For a general structured latent attribute model, consider the joint distribution of observed response vector $\mathbf{R}$ and latent attribute pattern vector $\mathbf{A}$ is

$$\mathbb{P}(\mathbf{R} = \boldsymbol{r},\ \mathbf{A} = \boldsymbol{\alpha} \mid \boldsymbol{\Theta}) = \exp\Big\{\sum_{j=1}^J \Big[r_j\Big(\prod_k \alpha_k^{q_{j,k}} \log\theta_j^+ + (1 - \prod_k \alpha_k^{q_{j,k}}) \log\theta_{j,\boldsymbol{\alpha}}^-\Big) + (1 - r_j)\Big(\prod_k \alpha_k^{q_{j,k}} \log(1-\theta_j^+) + (1 - \prod_k \alpha_k^{q_{j,k}}) \log(1 - \theta_{j,\boldsymbol{\alpha}}^-)\Big)\Big]\Big\}.$$

Therefore

$$\mathbb{P}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}},\ \mathbf{A} = \widetilde{\boldsymbol{\alpha}} \mid \boldsymbol{\Theta}) = \exp\Big\{\sum_{j=1}^J \Big[\prod_k \alpha_k^{q_{j,k}}\Big(\prod_k \widetilde{\alpha}_k^{q_{j,k}} \log\theta_j^+ + (1 - \prod_k \widetilde{\alpha}_k^{q_{j,k}}) \log\theta_{j,\widetilde{\boldsymbol{\alpha}}}^-\Big) + (1 - \prod_k \alpha_k^{q_{j,k}})\Big(\prod_k \widetilde{\alpha}_k^{q_{j,k}} \log(1-\theta_j^+) + (1 - \prod_k \widetilde{\alpha}_k^{q_{j,k}}) \log(1 - \theta_{j,\widetilde{\boldsymbol{\alpha}}}^-)\Big)\Big]\Big\}.$$

$$\mathbb{P}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}},\ \mathbf{A} = \boldsymbol{\alpha} \mid \boldsymbol{\Theta}) = \exp\Big\{\sum_{j=1}^J \Big[\prod_k \alpha_k^{q_{j,k}} \log\theta_j^+ + (1 - \prod_k \alpha_k^{q_{j,k}}) \log(1 - \theta_{j,\boldsymbol{\alpha}}^-)\Big]\Big\}.$$

Then for any $\widetilde{\boldsymbol{\alpha}} \neq \boldsymbol{\alpha}$,

$$\begin{aligned} &\log\mathbb{P}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}},\ \mathbf{A} = \boldsymbol{\alpha} \mid \boldsymbol{\Theta}) - \log\mathbb{P}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}},\ \mathbf{A} = \widetilde{\boldsymbol{\alpha}} \mid \boldsymbol{\Theta}) \\ &\geq \min_{j=1,\ldots,J} \{\log\theta_j^+ - \log\theta_{j,\widetilde{\boldsymbol{\alpha}}}^-,\ \log(1-\theta_{j,\boldsymbol{\alpha}}^+) - \log(1-\theta_j^+)\} \geq d > 0. \end{aligned} \tag{56}$$

That the above probability is bounded away from zero follows from the second part of assumption (20). So the claim (55) is proved. We next bound the probability of failure of including all the true patterns in the screening stage. First, since $\mathbf{A}_1, \ldots, \mathbf{A}_N \overset{i.i.d.}{\sim}$ Multinomial$(N,\ (p_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}_0))$, then $|\{i \in [N] : \mathbf{A}_i = \boldsymbol{\alpha}\}|$ denotes the number of subjects in the random sample whose attribute pattern is $\boldsymbol{\alpha}$. By the concentration inequality of the multinomial distribution, for any $\boldsymbol{\alpha} \in \mathcal{A}_0$,

$$\mathbb{P}\Big(\Big|\{i \in [N] : \mathbf{A}_i = \boldsymbol{\alpha}\}\Big| \geq Np_{\boldsymbol{\alpha}} - 2\sqrt{N}t\Big) \geq 1 - 2^{|\mathcal{A}_0|}\exp(-2t^2), \quad \forall t > 0.$$

Because of (20), we have $Np_{\boldsymbol{\alpha}} \geq Nc_0 \to \infty$ for all $\boldsymbol{\alpha} \in \mathcal{A}_0$. Assume that $\widehat{\theta}_j^+ - \widehat{\theta}_j^- > \delta > 0$ for each $j \in [J]$. This constraint can be incorporated into the screening procedure or checked *a posteiriori* after screening. So with probability at least $1 - 2^{|\mathcal{A}_0|}\exp(-2t^2)$ for a suitable t,

$$\begin{aligned} &\mathbb{P}(\widehat{\mathcal{A}}_{\text{screen}} \not\supseteq \mathcal{A}_0) \leq \sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} \mathbb{P}(\widehat{\mathbf{A}}_i \neq \boldsymbol{\alpha}\ \forall i \in [N] \text{ s.t. } \mathbf{A}_i = \boldsymbol{\alpha}) \\ &\leq \sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} \Big[\mathbb{P}\Big(\mathbf{R}_i = \boldsymbol{r}^{\boldsymbol{\alpha}},\ \exists \widetilde{\boldsymbol{\alpha}} \neq \boldsymbol{\alpha}, \\ &\qquad \widehat{\mathbb{P}}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}}, \mathbf{A} = \boldsymbol{\alpha}) > \widehat{\mathbb{P}}(\mathbf{R} = \boldsymbol{r}^{\boldsymbol{\alpha}}, \mathbf{A} = \widetilde{\boldsymbol{\alpha}}) \Big| \mathbf{A}_i = \boldsymbol{\alpha}\Big)\Big]^{N\left(p_{\boldsymbol{\alpha}} - 2t/\sqrt{N}\right)} \to 0, \end{aligned}$$

as $N \to \infty$. Here $\widehat{\mathbb{P}}$ refers to the probability measure of $\mathbf{R}$ and $\mathbf{A}$ given the estimated item parameters $\widehat{\boldsymbol{\theta}}^+$ and $\widehat{\boldsymbol{\theta}}^-$. This is because the probability inside the bracket in the above expression is strictly less than 1 due to (56); we denote this quantity by C_δ since it depends on δ. Therefore there is

$$\begin{aligned} \mathbb{P}(\widehat{\mathcal{A}}_{\text{screen}} \not\supseteq \mathcal{A}_0) \leq \sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} C_\delta^{N(p_{\boldsymbol{\alpha}} + o(1))} &= \sum_{\boldsymbol{\alpha} \in \mathcal{A}_0} \exp[-N(p_{\boldsymbol{\alpha}} + o(1))\log(1/C_\delta)] \\ &\leq |\mathcal{A}_0|\exp(-N\beta_{\min}), \end{aligned}$$

where $\beta_{\min}$ is a positive constant which can be taken as $c_0/2\log(1/C_\delta)$. The last inequality above results from $p_{\boldsymbol{\alpha}} \geq c_0$ for $\boldsymbol{\alpha} \in \mathcal{A}_0$ in (20) and that $C_\delta < 1$. Now we have obtained $\mathbb{P}(\widehat{\mathcal{A}}_{\text{screen}} \supseteq \mathcal{A}_0) \geq 1 - |\mathcal{A}_0|\exp(-N\beta_{\min})$, so the sure screening property holds and the proof is complete. □

Proof of Lemma 21. Following the proof of Theorem 2 in Genovese and Wasserman (2000), the overall bracketing entropy of the mixture distribution over $|\mathcal{A}|$ mixture components (latent attribute patterns) can be bounded by the entropy of the $|\mathcal{A}| - 1$ dimensional simplex multiplied by the product of the entropy of the item parameter space for each mixture component. Since there are a total number of $\binom{|\mathcal{A}_{\text{input}}|}{|\mathcal{A}|}$ possibilities of choosing $|\mathcal{A}|$ components from $|\mathcal{A}_{\text{input}}|$ ones, we have

$$N_{[]}(t, \mathcal{B}_{\mathcal{A}}) \leq \binom{|\mathcal{A}_{\text{input}}|}{|\mathcal{A}|} N_{[]}(t, \mathcal{T}^{|\mathcal{A}|-1}) \prod_{l=1}^{|\mathcal{A}|} N_{[]}(t/3, \mathcal{F}_l).$$

Next, Lemma 2 in Genovese and Wasserman (2000) gives the following bracketing entropy bound for the simplex, $N_{[]}(t, \mathcal{T}^{|\mathcal{A}|-1}) \leq |\mathcal{A}|(2\pi e)^{|\mathcal{A}|/2}/t^{|\mathcal{A}|-1}$. Since we consider the local parameter space around the true parameters (with squared Hellinger distance between the alternative model and the true model not greater than $2\epsilon^2$), the $1/t$ in the above display can be replaced by ϵ/t. Also, $N_{[]}(t/3, \mathcal{F}_l) \leq C_0\epsilon/t$ since the Hellinger distance is bounded by the L_2 distance and the t-bracketing number under the L_2 norm is bounded by $O(\epsilon/t)$. Therefore we have

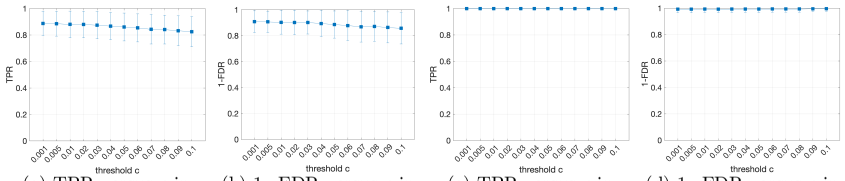
$$\begin{aligned} H(t, \mathcal{B}_{\mathcal{A}}) &\leq \log\Big\{\binom{|\mathcal{A}_{\text{input}}|}{|\mathcal{A}|} \frac{|\mathcal{A}|(2\pi e)^{|\mathcal{A}|/2}(\epsilon)^{|\mathcal{A}|-1}}{t^{|\mathcal{A}|-1}} \Big(\frac{\epsilon}{t}\Big)^{|\mathcal{A}|}\Big\} \\ &\lesssim |\mathcal{A}|\log|\mathcal{A}_{\text{input}}| + \log|\mathcal{A}| + |\mathcal{A}|\log(\epsilon/t) \end{aligned}$$

$$\lesssim |\mathcal{A}_0| \log |\mathcal{A}_{\text{input}}| \log(\epsilon/t).$$

where $|\mathcal{A}| \leq |\mathcal{A}_0|$ and an elementary inequality $\binom{a}{b} \leq a^b$ are used. □

Appendix B: Additional Experimental Results

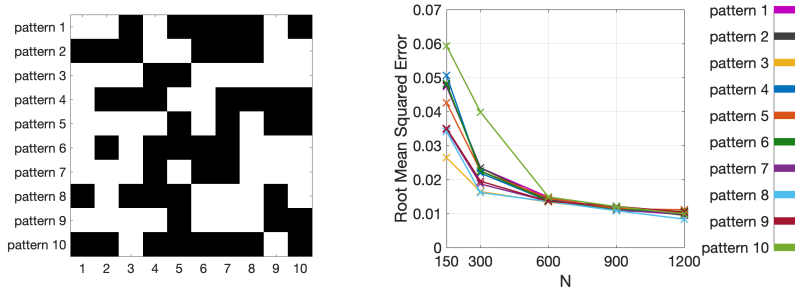
Impact of the value of the pre-specified c in Algorithm 1. In Algorithm 1, there is a pre-specified constant $c > 0$ when updating the Δ_l's. This constant c should be small, ideally close to zero. In all of our experiments in Section 5, we take $c = 0.01$. Next we examine how the value of c impacts the selection result of Algorithm 1. Since the performance of Algorithm 1 is the focus here, we choose the simulation setting with $K = 10$ such that screening can be omitted. Under sample sizes $N = 150$ and $N = 500$, the plots of the two accuracy measures versus c are presented in Figure 10. We observe that the results of Algorithm 1 are generally not that sensitive to the choice of c, though smaller c gives slightly better results for both accuracy measures under a small sample size $N = 150$. For N as large as 500, for all the values of $c \in \{0.001, 0.005\} \cup \{0.01 \times i : i = 1, 2, \ldots, 10\}$, the two accuracy measures are very close to one and do not have much variation. In practice, we recommend fixing c to a value no greater than 0.01.



(a) TPR versus c in Algo. 1, $N = 150$ (b) 1−FDR versus c in Algo. 1, $N = 150$ (c) TPR versus c in Algo. 1, $N = 500$ (d) 1−FDR versus c in Algo. 1, $N = 500$

Figure 10: Performance of Algorithm 1 across various values for threshold c. Setting is $K = 10$ and $1 - \theta_j^+ = \theta_j^- = 0.2$. In each scenario 200 runs are carried out, and the error bar is within one standard deviation of the mean accuracy.

Algorithm 1's performance on estimating the actual proportions of patterns. Other than the two accuracy measures for pattern selection presented in Table 2, we also evaluate how well the algorithms perform on estimating the actual proportions of the latent patterns. We use the simulation setting of the two-parameter SLAM with $K = 10$, $|\mathcal{A}_0| = 10$, $Q = (Q_1^\top, Q_2^\top, Q_3^\top)^\top$, with parameters $p_{\boldsymbol{\alpha}} = 0.1$ for $\boldsymbol{\alpha} \in \mathcal{A}$ and $1 - \theta_j^+ = \theta_j^- = 0.2$. This is the same setting as that of Example 7. We vary the sample size $N \in \{150, 300, 600, 900, 1200\}$ and compute the Root Mean Square Errors (RMSEs) of estimating the true proportions of latent patterns. The randomly generated 10 true patterns in $\mathcal{A}_0$ are presented in Figure 11(a), where each row represents a K-dimensional binary pattern. For each N, the RMSE of each proportion $p_{\boldsymbol{\alpha}}$, $\boldsymbol{\alpha} \in \mathcal{A}_0$ is computed based on 200 runs; and in each run, we first perform pattern selection by using EBIC to choose $\lambda \in \{-0.2 \times i : i = 1, 2, \ldots, 20\}$ in Algorithm 1 and then estimate the proportions based on the selected set of patterns. The results of RMSEs are presented in Figure 11(b). As can be seen from the figure, under a small sample size $N = 150$, the RMSEs of patterns are relatively diverse. In particular, the largest RMSE is around 0.06 and corresponds to pattern 10, $\boldsymbol{\alpha}_{10} = (0010000010)$, which is the pattern consisting of most "0"s; while the smallest RMSE is less than half of the largest and corresponds to pattern 3, $\boldsymbol{\alpha}_3 = (1110011111)$, which is the pattern consisting of most "1"s. Interestingly, this observation implies for a very small sample size and a sparse Q-matrix (each row having at most three entries of "1"s), those attribute patterns possessing fewer attributes are harder to estimate while those possessing more attributes are easier to estimate. While as N increases, the RMSEs of all the proportions decrease and their difference become not discernible. For $N = 1200$, all the RMSEs are around 0.01.

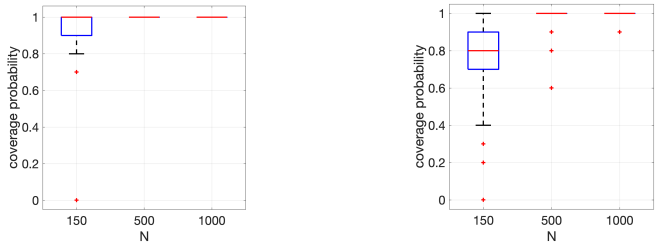


(a) patterns in $\mathcal{A}_0$: white for 1, black for 0 (b) RMSE of $\{p_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}_0\}$ versus N

Figure 11: Root Mean Square Errors (RMSEs) for estimating the true proportions of patterns decrease as sample size N increases. Results are based on 200 runs for each N.

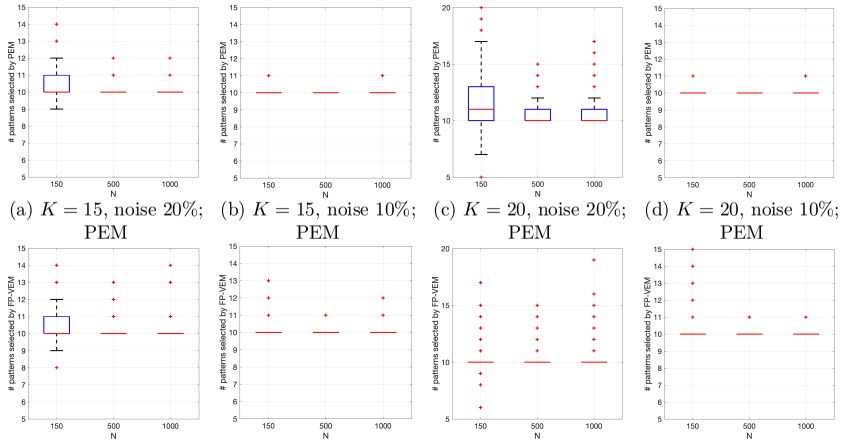
Evaluating the screening procedure under the multi-parameter SLAM. In the multi-parameter setting, we also evaluate the performance of the approximate screening procedure that is developed based on the likelihood of the two-parameter model. The results of the coverage probabilities are presented in Figure 12. The figure shows that despite being an approximate procedure, the screening Algorithm 3 has excellent performance for the multi-parameter SLAM that covers the two-parameter model as a submodel. Specifically, Figure 12 shows that for both $K = 15$ and $K = 20$, the approximate screening procedure almost always has a 100% coverage probability for $N = 500$ and $N = 1000$.

Sizes of the set of finally selected patterns under scenarios in Table 2. We present the results of the number of patterns that are finally selected by the proposed methods, corresponding to simulation scenarios in Table 2. Denote the set of patterns selected by the PEM algorithm and that selected by the FP-VEM algorithm by $\widehat{\mathcal{A}}_{\text{PEM}}$ and $\widehat{\mathcal{A}}_{\text{FP-VEM}}$, respectively. As shown in Figure 13, in the relatively strong signal setting with $1 - \theta_j^+ = \theta_j^- = 10\%$, the sizes of $\widehat{\mathcal{A}}_{\text{PEM}}$ and $\widehat{\mathcal{A}}_{\text{FP-VEM}}$ almost always equal 10, the number of true patterns. Combined with the accuracy measures presented in Table 2 in the main text, in most cases these selected 10 patterns are indeed exactly the true ones in $\mathcal{A}_0$. And in the relatively weak signal setting with $1 - \theta_j^+ = \theta_j^- = 20\%$, the sizes of $\widehat{\mathcal{A}}_{\text{PEM}}$ and $\widehat{\mathcal{A}}_{\text{FP-VEM}}$ can be slightly larger than $|\mathcal{A}_0|$ but still close to it.



(a) $K = 15$, multi-parameter SLAM (b) $K = 20$, multi-parameter SLAM

Figure 12: Coverage probabilities of the true patterns, from the approximate screening procedure under the multi-parameter SLAM. Boxplots are from 200 runs in each scenario.

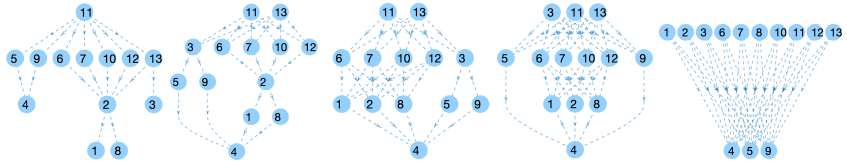


(a) $K = 15$, noise 20%; PEM (b) $K = 15$, noise 10%; PEM (c) $K = 20$, noise 20%; PEM (d) $K = 20$, noise 10%; PEM

(e) $K = 15$, noise 20%; FP-VEM (f) $K = 15$, noise 10%; FP-VEM (g) $K = 20$, noise 20%; FP-VEM (h) $K = 20$, noise 10%; FP-VEM

Figure 13: Sizes of the finally selected patterns $\widehat{\mathcal{A}}_{\text{PEM}}$ and $\widehat{\mathcal{A}}_{\text{FP-VEM}}$ under the two-parameter SLAM. The "noise" refers to the value of $1 - \theta_j^+ = \theta_j^-$. The number of true patterns is $|\mathcal{A}_0| = 10$.

TIMSS Data: Attribute structures corresponding to different Υ's. For the TIMSS data, we obtain those different attribute structures corresponding to different Υ's in the FP-VEM algorithm. The results are presented in Figure 14. Apart from the five structures shown in Figure 14(a)–(e), the two patterns selected when $\Upsilon \in [0.70, 0.74]$ are the all-zero and the all-one patterns, which do not result in any structure among the 13 attributes. Note that the structure in Figure 14(d) is equivalent to the structure selected by EBIC in Figure 9(b).



(a) $\Upsilon = 0.90$ (b) $\Upsilon = 0.88$ (c) $\Upsilon = 0.86$ (d) $\Upsilon \in [0.80, 0.84]$ (e) $\Upsilon \in [0.76, 0.78]$

Figure 14: Different attribute structures corresponding to various Υ's in Algorithm 2. Plot (d) here is equivalent to Figure 9(b), the attribute structure selected by EBIC.

References

Elizabeth S Allman, Catherine Matias, and John A Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37: 3099–3132, 2009.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

Animashree Anandkumar, Daniel Hsu, Majid Janzamin, and Sham Kakade. When are overcomplete topic models identifiable? Uniqueness of tensor tucker decompositions with structured sparsity. *Journal of Machine Learning Research*, 16:2643–2694, 2015.

Aditya Bhaskara, Moses Charikar, and Aravindan Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. In *Conference on Learning Theory*, pages 742–778, 2014.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Miguel A Carreira-Perpinán and Steve Renals. Practical identifiability of finite mixtures of multivariate bernoulli distributions. *Neural Computation*, 12(1):141–152, 2000.

Hanfeng Chen, Jiahua Chen, and John D Kalbfleisch. A modified likelihood ratio test for homogeneity in finite mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):19–29, 2001.

Hanfeng Chen, Jiahua Chen, and John D Kalbfleisch. Testing for a finite mixture model with two components. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):95–115, 2004.

Jiahua Chen. Optimal rate of convergence for finite mixture models. *The Annals of Statistics*, 23:221–233, 1995.

Jiahua Chen and Zehua Chen. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.

Yinghan Chen, Steven Andrew Culpepper, Yuguo Chen, and Jeffrey Douglas. Bayesian estimation of the DINA Q-matrix. *Psychometrika*, 83(1):89–108, 2018.

Badr-Eddine Chérief-Abdellatif and Pierre Alquier. Consistency of variational Bayes inference for estimation and model selection in mixtures. *arXiv preprint arXiv:1805.05054*, 2018.

Kyong Mi Choi, Young-Sun Lee, and Yoon Soo Park. What CDM can tell about what students have learned: An analysis of timss eighth grade mathematics. *Eurasia Journal of Mathematics, Science & Technology Education*, 11(6), 2015.

Jimmy de la Torre. The generalized DINA model framework. *Psychometrika*, 76:179–199, 2011.

Jimmy de la Torre and Jeffrey A Douglas. Higher-order latent trait models for cognitive diagnosis. *Psychometrika*, 69(3):333–353, 2004.

Jimmy de la Torre, L Andries van der Ark, and Gina Rossi. Analysis of clinical data from a cognitive diagnosis modeling framework. *Measurement and Evaluation in Counseling and Development*, 51(4):281–296, 2018.

Lawrence T DeCarlo. On the analysis of fraction subtraction data: The DINA model, classification, latent class sizes, and the Q-matrix. *Applied Psychological Measurement*, 35(1):8–26, 2011.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Louis V DiBello, William F Stout, and Louis A Roussos. Unified cognitive/psychometric diagnostic assessment likelihood-based classification techniques. *Cognitively diagnostic assessment*, pages 361–389, 1995.

Mathias Drton, Bernd Sturmfels, and Seth Sullivant. Algebraic factor analysis: tetrads, pentads and beyond. *Probability Theory and Related Fields*, 138(3-4):463–493, 2007.

Ivo Düntsch and Günther Gediga. Skills and knowledge structures. *British Journal of Mathematical and Statistical Psychology*, 48(1):9–27, 1995.

Christopher R Genovese and Larry Wasserman. Rates of convergence for the Gaussian mixture sieve. *The Annals of Statistics*, 28(4):1105–1127, 2000.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Yuqi Gu and Gongjun Xu. The sufficient and necessary condition for the identifiability and estimability of the DINA model. *Psychometrika*, 84(2):468–483, 2019a.

Yuqi Gu and Gongjun Xu. Partial identifiability of restricted latent class models. *The Annals of Statistics*, forthcoming, 2019b.

Mats Gyllenberg, Timo Koski, Edwin Reilink, and Martin Verlaan. Non-uniqueness in probabilistic numerical identification of bacteria. *Journal of Applied Probability*, 31:542–548, 1994.

Philippe Heinrich and Jonas Kahn. Strong identifiability and optimal minimax rates for finite mixture estimation. *The Annals of Statistics*, 46(6A):2844–2870, 2018.

Robert A. Henson, Jonathan L. Templin, and John T. Willse. Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74:191–210, 2009.

Nhat Ho and XuanLong Nguyen. Convergence rates of parameter estimation for some weakly identifiable finite mixtures. *The Annals of Statistics*, 44(6):2726–2755, 2016.

C.C. Holmes and S.G. Walker. Assigning a value to a power likelihood in a general Bayesian model. *Biometrika*, 104(2):497–503, 2017.

Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 11–20. ACM, 2013.

Ariel Jaffe, Roi Weiss, Shai Carmi, Yuval Kluger, and Boaz Nadler. Learning binary latent variable models: A tensor eigenpair approach. *arXiv preprint arXiv:1802.09656*, 2018.

Brian W. Junker and Klaas Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25:258–272, 2001.

Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

Young-Sun Lee, Yoon Soo Park, and Didem Taylan. A cognitive diagnostic modeling of attribute mastery in Massachusetts, Minnesota, and the US national sample using the TIMSS 2007. *International Journal of Testing*, 11(2):144–177, 2011.

Jacqueline P Leighton, Mark J Gierl, and Stephen M Hunka. The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement*, 41(3):205–237, 2004.

Eric Maris. Estimating multiple classification latent class models. *Psychometrika*, 64(2): 187–212, 1999.

Gunter Maris and Timo M Bechger. Equivalent diagnostic classification models. *Measurement*, 7:41–46, 2009.

Judith Rousseau and Kerrie Mengersen. Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):689–710, 2011.

André A Rupp, Jonathan Templin, and Robert A Henson. *Diagnostic measurement: Theory, methods, and applications*. Guilford Press, 2010.

Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.

P Smolensky. Chapter 6: information processing in dynamical systems: foundations of harmony theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, 1986.

Yu-Law Su, KM Choi, WC Lee, T Choi, and M McAninch. Hierarchical cognitive diagnostic analysis for timss 2003 mathematics. *Centre for Advanced Studies in Measurement and Assessment*, 35:1–71, 2013.

Jonathan Templin and Laine Bradshaw. Hierarchical diagnostic classification models: A family of models for estimating and testing attribute hierarchies. *Psychometrika*, 79(2): 317–339, 2014.

Jonathan L. Templin and Robert A. Henson. Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods*, 11:287–305, 2006.

Matthias von Davier. A general diagnostic model applied to language testing data. *British Journal of Mathematical and Statistical Psychology*, 61:287–307, 2008.

Matthias von Davier. The DINA model as a constrained general diagnostic model: Two variants of a model equivalency. *British Journal of Mathematical and Statistical Psychology*, 67(1):49–71, 2014.

Wing Hung Wong and Xiaotong Shen. Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *The Annals of Statistics*, pages 339–362, 1995.

Zhenke Wu, Maria Deloria-Knoll, and Scott L Zeger. Nested partially latent class models for dependent binary data; estimating disease etiology. *Biostatistics*, 18(2):200–213, 2017.

Zhenke Wu, Livia Casciola-Rosen, Antony Rosen, and Scott L Zeger. A Bayesian approach to restricted latent class models for scientifically-structured clustering of multivariate binary outcomes. *arXiv preprint arXiv:1808.08326*, 2018.

Gongjun Xu. Identifiability of restricted latent class models with binary responses. *The Annals of Statistics*, 45:675–707, 2017.

Gongjun Xu and Zhuoran Shang. Identifying latent structures in restricted latent class models. *Journal of the American Statistical Association*, 113(523):1284–1295, 2018.

Gongjun Xu and Stephanie Zhang. Identifiability of diagnostic classification models. *Psychometrika*, 81:625–649, 2016.

Kazuhiro Yamaguchi and Kensuke Okada. Comparison among cognitive diagnostic models for the timss 2007 fourth grade mathematics assessment. *PLoS ONE*, 12(2):e0188691, 2018.

Yun Yang, Debdeep Pati, and Anirban Bhattacharya. α-variational inference with statistical guarantees. *The Annals of Statistics*, forthcoming, 2019.

Journal of Machine Learning Research 20 (2019) 1-42 Submitted 10/18; Revised 3/19; Published 6/19

Graph Reduction with Spectral and Cut Guarantees

Andreas Loukas ANDREAS.LOUKAS@EPFL.CH
Laboratoire de traitement des signaux 2,
École Polytechnique Fédérale de Lausanne,
CH-1015 Lausanne, Switzerland

Editor: Karsten Borgwardt

Abstract

Can one reduce the size of a graph without significantly altering its basic properties? The graph reduction problem is hereby approached from the perspective of *restricted spectral approximation*, a modification of the spectral similarity measure used for graph sparsification. This choice is motivated by the observation that restricted approximation carries strong *spectral* and *cut* guarantees, and that it implies approximation results for unsupervised learning problems relying on spectral embeddings. The article then focuses on coarsening—the most common type of graph reduction. Sufficient conditions are derived for a small graph to approximate a larger one in the sense of restricted approximation. These findings give rise to algorithms that, compared to both standard and advanced graph reduction methods, find coarse graphs of improved quality, often by a large margin, without sacrificing speed.

Keywords: graph reduction and coarsening, spectral methods, unsupervised learning

1. Introduction

As graphs grow in size, it becomes pertinent to look for generic ways of simplifying their structure while preserving key properties. Simplified graph representations find profound use in the design of approximation algorithms, can facilitate storage and retrieval, and ultimately ease graph data analysis by separating overall trends from details.

There are two main ways to simplify graphs. First, one may reduce the number of edges—a technique commonly referred to as *graph sparsification*. In a series of works, it has been shown that it is possible to find sparse graphs that approximate all pairwise distances (Peleg and Schäffer, 1989), every cut (Karger, 1999), or every eigenvalue (Spielman and Teng, 2011)—respectively referred to as spanners, cut sparsifiers and spectral sparsifiers. Spectral sparsification techniques, in particular, can yield computational benefits whenever the number of edges is the main bottleneck (Batson et al., 2013). Indeed, they form a fundamental component of nearly-linear time algorithms for linear systems involving symmetric diagonally dominant matrices (Koutis et al., 2010; Spielman and Srivastava, 2011), and have found application to machine learning problems involving graph-structured data (Calandriello et al., 2018).

©2019 Loukas.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-680.html.

Alternatively, one may seek to reduce the size of the graph directly, i.e., the number of its vertices N, by some form of vertex selection or re-combination scheme followed by re-wiring. This idea can be traced back to the multigrid literature, that targets the acceleration of finite-element methods using cycles of multi-level coarsening, lifting, and refinement. After being generalized to graphs, reduction methods have become pervasive in computer science and form a key element of modern graph processing pipelines, especially with regards to graph partitioning (Hendrickson and Leland, 1995; Karypis and Kumar, 1998; Kushnir et al., 2006; Dhillon et al., 2007; Wang et al., 2014) and graph visualization (Koren, 2002; Hu, 2005; Walshaw, 2006). In machine learning, reduction methods are used to create multi-scale representations of graph-structured data (Lafon and Lee, 2006; Gavish et al., 2010; Shuman et al., 2016) and as a layer of graph convolutional neural networks (Bruna et al., 2014; Defferrard et al., 2016; Bronstein et al., 2017; Simonovsky and Komodakis, 2017; Ardizzone et al., 2018; Liang et al., 2018). In addition, being shown to solve linear systems in (empirically) linear time (Koutis et al., 2011; Livne and Brandt, 2012) as well as to approximate the Fiedler vector (Urschel et al., 2014; Gandhi, 2016), reduction methods have been used to accelerate inverse problems employing graph-regularization (Hirani et al., 2015; Colley et al., 2017). Some of their main benefits are the ability to deal with sparse graphs (graphs with at most $O(N \log N)$ edges) and to accelerate algorithms whose complexity depends on the number of vertices as well as edges.

In contrast to graph sparsification, there has been only circumstantial theory supporting graph reduction (Moitra, 2011; Dörfler and Bullo, 2013; Loukas and Vandergheynst, 2018). The lack of a concrete understanding of how different reduction choices affect fundamental graph properties is an issue: the significant majority of reduction algorithms in modern graph processing and machine learning pipelines have been designed based on intuition and possess no rigorous justification or provable guarantees.

1.1. A Spectral Perspective

My starting point in this work is *spectral similarity*—a measure that has been proven useful in sparsification for determining how well a graph approximates another one. To render spectral similarity applicable to graphs of different sizes, I generalize it and restrict it over a subspace of size that is at most equal to the size of the reduced graph. I refer to the resulting definition as *restricted spectral approximation*[1] (or restricted approximation for short). Despite being a statement about subspaces, restricted approximation has significant consequences. It is shown that when the subspace in question is a principal eigenspace (this is a feature agnostic choice where one wants to preserve only the graph structure), the eigenvalues and eigenspaces of the reduced graph approximate those of the original large graph. It is then a corollary that (i) if the large graph has a good (normalized) cut so does the smaller one; and (ii) that unsupervised learning algorithms that utilize spectral embeddings, such as spectral clustering (Von Luxburg, 2007) and Laplacian eigenmaps (Belkin and Niyogi, 2003), can also work well when ran on the smaller graph and their solution is lifted. More

1. Though similarly named, the definition of *restricted spectral similarity* previously proposed by Loukas and Vandergheynst (2018) concerns a set of vectors (rather than subspaces) and is significantly weaker than the one examined here.

generally, it is expected that methods relying on specific eigenspaces will exhibit similar behavior when combined with coarsening, e.g., in graph signal regression (Loukas and Perraudin, 2016; Grassi et al., 2017), compression (Shahid et al., 2016), sampling (Puy et al., 2018), or forecasting (Isufi et al., 2018).

The analysis then focuses on *graph coarsening*—a popular type of reduction where, in each level, reduced vertices are formed by contracting disjoint sets of connected vertices (each such set is called a *contraction set*). I derive sufficient conditions for a small coarse graph to approximate a larger graph in the sense of restricted spectral approximation. Crucially, this result holds for any number of levels and is independent of how the subspace is chosen. Though the derived bound is global, a decoupling argument renders it locally separable over levels and contraction sets, facilitating computation. The final bound can be interpreted as measuring the *local variation* over each contraction set, as it involves the maximum variation of vectors supported on each contracted subgraph.

These findings give rise to algorithms for graph coarsening that I refer to as *local variation algorithms*. Each such algorithm starts from a predefined family of candidate contraction sets. Even though any connected set of vertices may form a valid candidate set, I opt for small well-connected sets constructed based on simple rules: in the *edge-based variant* one candidate set is constructed for each of pair of adjacent vertices, whereas in the *neighborhood-based variant* every candidate set contains a vertex along with all adjacent vertices. The algorithm then greedily[2] contracts those sets whose local variation is the smallest. Depending on how the candidate family is constructed, the proposed algorithms obtain different solutions, trading off computational complexity for reduction.

1.2. Theoretical and Practical Implications

This work improves and generalizes upon the state-of-the-art in several ways:

- Instead of directly focusing on specific constructions, a general graph reduction scheme is studied featuring coarsening as a special case. As a consequence, the implications of restricted approximation are proven in a fairly general setting where specifics of the reduction, such as the type of graph representation and the reduction matrices involved, are abstracted.
- Contrary to previous results on the analysis of coarsening (Loukas and Vandergheynst, 2018), the analysis holds for multiple levels of reduction. Given that the majority of coarsening methods reduce the number of vertices by a constant factor at each level, a multi-level approach is necessary to achieve significant reduction. Along that line, the analysis also brings an intuitive insight: rather than taking the standard approach of approximating at each level the graph produced by the previous level, one should strive to preserve the properties of the original graph at every level.

2. Even after decoupling, the problem of candidate set selection is not only NP-hard but also cannot be approximated to a constant factor in polynomial time (by reduction to the maximum-weight independent set problem). For the specific case of edge-based families, where one candidate set is constructed for each pair of adjacent vertices, the greedy iterative contraction can be substituted by more sophisticated procedures accompanied by improved guarantees.

- The proposed local variation algorithms are not heuristically designed, but optimize (an upper bound of) the restricted spectral approximation objective. Despite the breadth of the literature that utilizes some form of graph reduction and coarsening, the overwhelming majority of known methods are heuristics (Safro et al., 2015). A notable exception is Kron reduction (Dörfler and Bullo, 2013), an elegant method that aims to preserve the effective resistance distance. Compared to Kron reduction, the methods proposed here are accompanied by stronger spectral guarantees (i.e., beyond interlacing), do not sacrifice the sparsity of the graph, and can ultimately be more scalable as they do not rely on the Schur complement of the Laplacian matrix.

To demonstrate the practical benefits of local variation methods, the analysis is complemented with numerical results on representative graphs ranging from scale-free to planar graphs. Compared to both standard (Karypis and Kumar, 1998) and advanced reduction methods (Ron et al., 2011; Livne and Brandt, 2012; Shuman et al., 2016), the proposed methods yield small graphs of improved spectral quality, often by a large margin, without being much slower than naive heavy-edge matching. A case in point: when examining how close are the principal eigenvalues of the coarse and original graph for a reduction of 70%, local variation methods attain on average $2.6\times$ smaller error; this gain becomes $3.9\times$ if one does not include Kron reduction in the comparison.

2. Graph Reduction and Coarsening

The following section introduces graph reduction. The exposition starts by considering a general reduction scheme. It is then shown how graph coarsening arises if one additionally imposes few natural restrictions on the interpretability of reduced variables.

2.1. Graph Reduction

Consider a positive semidefinite (PSD) matrix $L \in \mathbb{R}^{N\times N}$ whose sparsity structure captures the connectivity structure of a connected weighted undirected graph $G = (\mathcal{V}, \mathcal{E}, W)$ of $N = |\mathcal{V}|$ vertices and $M = |\mathcal{E}|$ edges. In other words, $L(i,j) \neq 0$ only if e_{ij} is a valid edge between vertices v_i and v_j. Moreover, let x be an arbitrary vector of size N.

I study the following generic reduction scheme:

Scheme 1: Graph reduction

Commence by setting $L_0 = L$ and $x_0 = x$ and proceed according to the following two recursive equations:

$$L_\ell = P_\ell^{\mp} L_{\ell-1} P_\ell^{+} \quad \text{and} \quad x_\ell = P_\ell x_{\ell-1},$$

where $P_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ are matrices with more columns than rows, $\ell = 1, 2, \ldots, c$ is the level of the reduction, symbol $+$ (resp. $\mp$) denotes the pseudoinverse (resp. transposed

Scheme 1: Graph reduction (cont.)

pseudoinverse), and N_ℓ is the dimensionality at level ℓ such that $N_0 = N$ and $N_c = n \ll N$.

Vector x_c is lifted back to $\mathbb{R}^N$ by recursion $\widetilde{x}_{\ell-1} = P_\ell^+ \widetilde{x}_\ell$, where $\widetilde{x}_c = x_c$.

Graph reduction thus involves a sequence of $c+1$ graphs

$$G = G_0 = (\mathcal{V}_0, \mathcal{E}_0, W_0) \quad G_1 = (\mathcal{V}_1, \mathcal{E}_1, W_1) \quad \cdots \quad G_c = (\mathcal{V}_c, \mathcal{E}_c, W_c) \tag{1}$$

of decreasing size $N = N_0 > N_1 > \cdots > N_c = n$, where the sparsity structure of L_ℓ matches that of graph G_ℓ, and each vertex of G_ℓ represents one of more vertices of $G_{\ell-1}$.

The multi-level design allows us to achieve high *dimensionality reduction ratio*

$$r = 1 - \frac{n}{N},$$

even when at each level the ratio $r_\ell = 1 - \frac{N_\ell}{N_{\ell-1}}$ is small. For instance, supposing that $r_\ell \geq \varrho$ for each ℓ, then $c = O(\log(n/N)/\log(1-\varrho))$ levels suffice to reduce the dimension to n.

One may express the reduced quantities in a more compact form:

$$x_c = Px, \quad L_c = P^{\mp} L P^+ \quad \text{and} \quad \widetilde{x} = \Pi x, \tag{2}$$

where $P = P_c \cdots P_1$, $P^+ = P_1^+ \cdots P_c^+$, $\Pi = P^+ P$, and for convenience, I drop zero indices and refer to a lifted vector as $\widetilde{x} (= \widetilde{x}_0)$. The rational of this scheme is that vector $\widetilde{x}$ should be the best approximation of x given P in an ℓ_2-sense, which is a consequence of the following property:

Property 1 Π *is a projection matrix.*

On the other hand, matrix L is reduced such that $x_c^\top L_c x_c = \widetilde{x}^\top L \widetilde{x}$.

Though introduced here for the reduction of sparse PSD matrices representing the sparsity structure of a graph, Scheme 1 can also be applied to any PSD matrix L. This and similar reduction schemes belong to the class of Nyström methods and, to the extent of my knowledge, were first studied in the context of approximate low-rank matrix approximation (Halko et al., 2011; Wang and Zhang, 2013). Despite the common starting point, interpreting L and L_c as sparse "graph matrices", as it is done here, incorporates a graph-theoretic twist to reduction, distinguishing it from previous approaches [3]: the constructions that we will study are eventually more scalable and interpretable as they maintain the graph structure of L after reduction. Obtaining guarantees is also arguably more challenging in this setting, as the involved problems end up being strongly combinatorial.

3. To achieve low-rank approximation, matrix P is usually built by sampling columns of L.

2.2. Properties of Reduced Graphs

Even in the general context where P is an arbitrary $n \times N$ matrix, certain handy properties can be proven about the relation between L_c and L.

To begin with, it is simple to see that the set of positive semidefinite matrices is closed under reduction.

Property 2 *If L is PSD, then so is L_c.*

The proof is elementary: if L is PSD then there exists matrix S such that $L = S^\top S$, implying that $L_c = P^{\mp} L P^+$ can also be written as $L_c = S_c^\top S_c$ if one sets $S_c = SP^+$.

I further consider the spectrum of the two matrices. Sort the eigenvalues of L as $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ and denote by $\widetilde{\lambda}_k$ the k-th largest eigenvalue of L_c and $\widetilde{u}_k$ the associated eigenvector.

It turns out that the eigenvalues $\widetilde{\lambda}$ and λ are interlaced.

Theorem 3 *For any P with full-row rank and $k = 1, \ldots, n$, we have*

$$\gamma_1 \lambda_k \leq \tilde{\lambda}_k \leq \gamma_2 \lambda_{k+N-n}$$

with $\gamma_1 = \lambda_1((PP^\top)^{-1})$ and $\gamma_2 = \lambda_n((PP^\top)^{-1})$, respectively the smallest and largest eigenvalue of $(PP^\top)^{-1}$.

The above result is a generalization of the Cauchy interlacing theorem for the case that $PP^\top \neq I$. It also resembles the interlacing inequalities known for the normalized Laplacian, where the re-normalization is obtained by construction. Chen et al. (2004) showed in Theorem 2.7 of their paper that after contracting $N-n$ edges $\lambda_{k-N+n} \leq \lambda_k \leq \lambda_{k+N-n}$ for $k = 1, 2, \ldots, n$ and with $\lambda_\ell = 0$ when $\ell \leq 0$, resembling the upper bound above. The lower bound is akin to that given by Chung (1997, Lemma 1.15), again for the normalized Laplacian. Also notably, the inequalities are similar to those known for Kron reduction (Dörfler and Bullo, 2013, Lemma 3.6).

Theorem 3 is particularly pessimistic as it has to hold for every possible P and L (subject to γ_1 and γ_2). Much stronger results will be obtained later on by restricting the attention to constructions that satisfy additional properties (see Theorem 13).

One can also say something about the action of L_c on vectors.

Property 4 *For every vector $x \in im(\Pi)$, one has*

$$x_c^\top L_c x_c = x^\top \Pi L \Pi x = x^\top L x \quad \textit{and} \quad \widetilde{x} = \Pi x = x.$$

In other words, reduction maintains the action of L of every vector that lies in the image of Π. Naturally, after lifting the eigenvectors $\widetilde{u}_k$ of L_c are included in this set.

2.3. Coarsening as a Type of Graph Reduction

Coarsening is a type of graph reduction abiding to a set of constraints that render the graph transformation interpretable. More precisely, in coarsening one selects for each level ℓ a surjective (i.e., many-to-one) map $\varphi_\ell : \mathcal{V}_{\ell-1} \to \mathcal{V}_\ell$ between the original vertex set $\mathcal{V}_{\ell-1}$ and the smaller vertex set $\mathcal{V}_\ell$. I refer to the set of vertices $\mathcal{V}_{\ell-1}^{(r)} \subseteq \mathcal{V}_{\ell-1}$ mapped onto the same vertex v'_r of $\mathcal{V}_\ell$ as a *contraction set*:

$$\mathcal{V}_{\ell-1}^{(r)} = \{v \in \mathcal{V}_{\ell-1} : \varphi_\ell(v) = v'_r\}$$

For a graphical depiction of contraction sets, see Figure 1. I also constrain φ_ℓ slightly by requiring that the subgraph of $G_{\ell-1}$ induced by each contraction set $\mathcal{V}_{\ell-1}^{(r)}$ is connected.

It is easy to deduce that contraction sets induce a partitioning of $\mathcal{V}_{\ell-1}$ into N_ℓ subgraphs, each corresponding to a single vertex of $\mathcal{V}_\ell$. Every reduced variable thus corresponds to a small set of adjacent vertices in the original graph and coarsening amounts to a scaling operation. An appropriately constructed coarse graph aims to capture the global problem structure, whereas neglected details can be recovered in a local refinement phase.

Coarsening can be placed in the context of Scheme 1 by restricting each P_ℓ to lie in the family of coarsening matrices, defined next:

Definition 5 (Coarsening matrix) *Matrix $P_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ is a coarsening matrix w.r.t. graph $G_{\ell-1}$ if and only if it satisfies the following two conditions:*

a. It is a surjective mapping of the vertex set, meaning that if $P_\ell(r,i) \neq 0$ then $P_\ell(r',i) = 0$ for every $r' \neq r$.

b. It is locality preserving, equivalently, the subgraph of $G_{\ell-1}$ induced by the non-zero entries of $P_\ell(r,:)$ is connected for each r.

An interesting consequence of this definition is that, in contrast to graph reduction, with coarsening matrices the expensive pseudo-inverse computation can be substituted by simple transposition and re-scaling:

Proposition 6 (Easy inversion) *The pseudo-inverse of a coarsening matrix P_ℓ is given by $P_\ell^+ = P_\ell^\top D_\ell^{-2}$, where D_ℓ is the diagonal matrix with $D_\ell(r,r) = \|P_\ell(r,:)\|_2$.*

Proposition 6 carries two consequences. First, coarsening can be done in linear time. Each coarsening level (both in the forward and backward directions) entails multiplication by a sparse matrix. Furthermore, both P_ℓ and P_ℓ^+ have only $N_{\ell-1}$ non-zero entries meaning that $O(N)$ and $O(M)$ operations suffice to coarsen respectively a vector and a matrix L whose sparsity structure reflects the graph connectivity. In addition, the number of graph edges also decreases at each level. Denoting by μ_ℓ the average number of edges of the graphs induced by contraction sets $\mathcal{V}_{\ell-1}^{(r)}$ for every r, then a quick calculation reveals that the coarsest graph has $m = M - \sum_{\ell=1}^{c} N_\ell \mu_\ell$ edges. If, for instance, at each level all vertices are perfectly contracted into pairs then $\mu_\ell = 2$ and $N_\ell = N/2^\ell$, meaning that $m = M - 2N(1-2^{-c})$.

(a) Graph G (b) Coarse graph G_c

Figure 1: Toy coarsening example. Grey discs denote contraction sets. The first three vertices of G forming contraction set $\mathcal{V}_0^1$ are contracted onto vertex v'_1. All other vertices remain unaffected.

2.4. Laplacian Consistent Coarsening

A further restriction that can be imposed is that coarsening is consistent w.r.t. the Laplacian form. Suppose that L is the combinatorial Laplacian of G defined as

$$L(i,j) = \begin{cases} d_i & \text{if } i = j \\ -w_{ij} & \text{if } e_{ij} \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases}$$

where w_{ij} is the weight associated with edge e_{ij} and d_i the weighted degree of v_i. The following proposition can be proven:

Proposition 7 (Consistency) *Let P be a coarsening matrix w.r.t. a graph with combinatorial Laplacian L. Matrix $L_c = P^\mp L P^+$ is a combinatorial Laplacian if and only if the non-zero entries of P^+ are equally valued.*

It is a corollary of Propositions 6 and 7 that in Laplacian consistent coarsening, for any $v'_r \in \mathcal{V}_\ell$ and $v_i \in \mathcal{V}_{\ell-1}$ matrices $P_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $P_\ell^+ \in \mathbb{R}^{N_{\ell-1} \times N_\ell}$ are given by:

$$P_\ell(r,i) = \begin{cases} \frac{1}{|\mathcal{V}_{\ell-1}^{(r)}|} & \text{if } v_i \in \mathcal{V}_{\ell-1}^{(r)} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad [P_\ell^+](i,r) = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}_{\ell-1}^{(r)} \\ 0 & \text{otherwise,} \end{cases}$$

where the contraction sets $\mathcal{V}_{\ell-1}^{(1)}, \ldots, \mathcal{V}_{\ell-1}^{(N_\ell)}$ were defined in Section 2.3.

2.4.1. A Toy Example

The toy graph shown in Figure 1a whose Laplacian is given by

$$L = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & 2 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

illustrates an example where the gray vertices $\mathcal{V}_0^{(1)} = \{v_1, v_2, v_3\}$ of G are contracted onto vertex v_1', as shown in Figure 1b. The main matrices I have defined are

$$P_1 = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad P_1^+ = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Pi = P_1^+ P_1 = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and coarsening results in

$$L_c = P_1^{\mp} L P_1^+ = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad x_c = P_1 x = \begin{bmatrix} (x(1) + x(2) + x(3))/3 \\ x(4) \\ x(5) \end{bmatrix}.$$

Finally, when lifted x_c becomes

$$\widetilde{x} = P_1^+ x_c = \begin{bmatrix} (x(1) + x(2) + x(3))/3 \\ (x(1) + x(2) + x(3))/3 \\ (x(1) + x(2) + x(3))/3 \\ x(4) \\ x(5) \end{bmatrix}.$$

Since vertices v_4 and v_5 are not affected, the respective contraction sets $\mathcal{V}_0^{(2)}$ and $\mathcal{V}_0^{(3)}$ are singleton sets.

2.5. Properties of Laplacian Consistent Coarsening

Due to its particular construction, Laplacian consistent coarsening is accompanied some interesting properties. I present three in the following:

Cuts. To begin with, weights of edges in G_c correspond to weights of cuts in G.

Property 8 *For any level ℓ, the weight $W_\ell(r, q)$ between vertices $v_r', v_q' \in \mathcal{V}_\ell$ is equal to*

$$W_\ell(r, q) = \sum_{v_i \in \mathcal{S}_\ell^{(r)}} \sum_{v_j \in \mathcal{S}_\ell^{(q)}} w_{ij},$$

where $\mathcal{S}_\ell^{(r)} = \{v_i \in \mathcal{V} : \varphi_\ell \circ \cdots \circ \varphi_1(v_i) = v_r'\} \subset \mathcal{V}$ contains all vertices of G contracted onto $v_r' \in \mathcal{V}_\ell$.

In the toy example, there exists a single edge of unit weight connecting vertices in $\mathcal{V}_0^{(1)}$ and $\mathcal{V}_0^{(2)}$, and as such the weight between v_1' and v_2' is equal to one.

Eigenvalue interlacing. For a single level of Laplacian consistent coarsening, matrix $PP^\top = P_1 P_1^\top$ is given by $\mathrm{diag}(1/|\mathcal{V}_0^{(1)}|, \ldots, 1/|\mathcal{V}_0^{(N_1)}|)$, implying that the multiplicative constants in Theorem 3 are:

$$\gamma_1 = \min_{v_i \in \mathcal{V}} |\mathcal{V}_0^{\varphi_1(v_i)}| \geq 1 \quad \text{and} \quad \gamma_2 = \max_{v_i \in \mathcal{V}} |\mathcal{V}_0^{\varphi_1(v_i)}|.$$

Above, $v_r' = \varphi_1(v_i) \in \mathcal{V}_1$ is the vertex to which v_i is mapped to and the set $\mathcal{V}_0^{\varphi_1(v_i)}$ contains all vertices also contracted to v_r'. Thus in the toy example, $\lambda_k \leq \tilde{\lambda}_k \leq 3\lambda_{k+2}$ for every $k \leq 3$. If multiple levels are utilized these terms become dependent on the sequence of contractions. To obtain a general bound let $\varphi_1^\ell(v_i) = \varphi_\ell \circ \cdots \circ \varphi_1(v_i) \in \mathcal{V}_\ell$ be the vertex onto which $v_i \in \mathcal{V}$ is contracted to in the ℓ-th level.

Property 9 *If L_c is obtained from L by Laplacian consistent coarsening, then*

$$\gamma_1 \geq \min_{v_i \in \mathcal{V}} \prod_{\ell=1}^{c} |\mathcal{V}_{\ell-1}^{\varphi_1^\ell(v_i)}| \geq 1 \quad \text{and} \quad \gamma_2 \leq \max_{v_i \in \mathcal{V}} \prod_{\ell=1}^{c} |\mathcal{V}_{\ell-1}^{\varphi_1^\ell(v_i)}|,$$

with the set $\mathcal{V}_{\ell-1}^{\varphi_1^\ell(v_i)}$ containing all vertices of $\mathcal{V}_{\ell-1}$ that are contracted onto $\varphi_1^\ell(v_i)$.

The proof follows from the diagonal form of $P_\ell \cdots P_1 P_1^\top \cdots P_\ell^\top$ and the special row structure of each P_ℓ for every ℓ, but it's not included for brevity. The dependency of $\tilde{\lambda}_k$ on the size of contraction sets can be removed either by enforcing at each level that all contraction sets have identical size and dividing the graph weights by that size, or by re-normalizing each P_ℓ such that $P_\ell^\top = P_\ell^+$. The latter approach was used by Loukas and Vandergheynst (2018) but is not adopted here as it results in L_c losing its Laplacian form.

Nullspace. Finally, as is desirable, the structure of the nullspace of L is preserved both by coarsening and lifting:

Property 10 *If P is a (multi-level) Laplacian consistent coarsening matrix, then*

$$P 1_N = 1_n \quad \text{and} \quad P^+ 1_n = 1_N,$$

where the subscript indicates the dimensionality of the constant vector.

Thus, we can casually ignore vectors parallel to the constant vector in our analysis.

3. Restricted Notions of Approximation

This section formalizes how should a graph be reduced such that fundamental structural properties (e.g., its spectrum and cuts) are preserved. Inspired by work in graph sparsification, I introduce a measure of approximation that is tailored to graph reduction. The new definition implies strong guarantees about the distance of the original and coarsened spectrum and gives conditions such that the cut structure of a graph is preserved by coarsening.

3.1. Restricted Spectral Approximation

One way to define how close a PSD matrix L is to its reduced counterpart is to establish an isometry guarantee w.r.t. the following induced semi-norms:

$$\|x\|_L = \sqrt{x^\top L x} \quad \text{and} \quad \|x_c\|_{L_c} = \sqrt{x_c^\top L_c x_c}$$

Ideally, one would hope that there exists $\epsilon > 0$ such that

$$(1-\epsilon)\ \|x\|_L \leq \|x_c\|_{L_c} \leq (1+\epsilon)\ \|x\|_L \tag{3}$$

for all $x \in \mathbb{R}^N$.

If the inequalities in (3) hold, matrices L_c and L are called ϵ-similar. The objective of constructing sparse spectrally similar graphs is the main idea of spectral graph sparsifiers, a popular method for accelerating the solution of linear systems involving the Laplacian. In addition, spectral similarity carries several interesting consequences that are of great help in the construction of approximation algorithms: the eigenvalues and eigenvectors of two similar graphs are close and, moreover, all vertex partitions have similar cut size.

In contrast to graph sparsification however, since here the dimension of the space changes it is impossible to satisfy (3) for every $x \in \mathbb{R}^N$ unless one trivially sets $\epsilon = 1$ (this follows by a simple rank argument). To carry out a meaningful analysis, one needs to consider a subspace of dimension $k \leq n$ and aim to approximate the behavior of L solely within it.

With this in mind, I define the following generalization of spectral similarity:

Definition 11 (Restricted spectral approximation) *Let $\mathbf{R}$ be a k-dimensional subspace of $\mathbb{R}^N$. Matrices L_c and L are $(\mathbf{R}, \epsilon)$-similar if there exists an $\epsilon \geq 0$ such that*

$$\|x - \tilde{x}\|_L \leq \epsilon\ \|x\|_L \quad \textit{for all} \quad x \in \mathbf{R},$$

where $\tilde{x} = P^+ P x$.

In addition to the restriction on $\mathbf{R}$, the above definition differs from (3) in how the error is measured. In fact, it asserts a property that is slightly stronger than an approximate isometry w.r.t. a semi-norm within $\mathbf{R}$. The strengthening of the notion of approximation deviates from the restricted spectral similarity property proposed by Loukas and Vandergheynst (2018) and is a key ingredient in obtaining multi-level bounds. Nevertheless, one may recover a restricted spectral similarity-type guarantee as a direct consequence:

Corollary 12 *If L_c and L are $(\mathbf{R}, \epsilon)$-similar, then*

$$(1-\epsilon)\ \|x\|_L \leq \|x_c\|_{L_c} \leq (1+\epsilon)\ \|x\|_L \quad \textit{for all} \quad x \in \mathbf{R}.$$

Proof Let S be defined such that $L = S^\top S$. By the triangle inequality:

$$|\ \|x\|_L - \|x_c\|_{L_c}\ | = |\ \|Sx\|_2 - \left\|SP^+Px\right\|_2\ | \leq \|Sx - SP^+Px\|_2 = \|x - \tilde{x}\|_L \leq \epsilon\ \|x\|_L\ ,$$

which is equivalent to the claimed relation. ■

Clearly, if L_c and L are $(\mathbf{R}, \epsilon)$-similar then they are also $(\mathbf{R}', \epsilon')$-similar, where $\mathbf{R}'$ is any subspace of $\mathbf{R}$ and $\epsilon' \geq \epsilon$. As such, results about large subspaces and small ϵ are the most desirable.

It will be shown in Sections 3.2 and 3.3 that the above definition implies restricted versions of the spectral and cut guarantees provided by spectral similarity. For instance, instead of attempting to approximate the entire spectrum as done by spectral graph sparsifiers, here one can focus on a subset of the spectrum with particular significance.

3.2. Implications for the Graph Spectrum

One of the primary benefits of restricted spectral approximation is that it implies a relation between the spectra of matrices L and L_c that goes beyond interlacing (see Theorem 3).

To this effect, consider the smallest k eigenvalues and corresponding eigenvectors and define the following matrices:

$$U_k \in \mathbb{R}^{N \times k} = [u_1, u_2, \ldots, u_k] \quad \text{and} \quad \Lambda_k = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)$$

As I will show next, ensuring that ϵ in Proposition 17 is small when $\mathbf{R} = \mathbf{U}_k \triangleq \operatorname{span}(U_k)$ suffices to guarantee that the first k eigenvalues and eigenvectors of L and L_c are aligned.

The first result concerns eigenvalues.

Theorem 13 (Eigenvalue approximation) *If L_c and L are $(\mathbf{U}_k, \epsilon_k)$-similar, then*

$$\gamma_1\ \lambda_k \leq \widetilde{\lambda}_k \leq \gamma_2\ \frac{(1+\epsilon_k)^2}{1 - \epsilon_k^2 (\lambda_k / \lambda_2)}\ \lambda_k,$$

whenever $\epsilon_k^2 < \lambda_2 / \lambda_k$.

Crucially, the bound depends on λ_k instead of λ_{k+N-n} and thus can be significantly tighter than the one given by Theorem 3. Noticing that $\epsilon_k \leq \epsilon_{k'}$ whenever $k < k'$, one also deduces that it is stronger for smaller eigenvalues. For $k = 2$ in particular, one has

$$\gamma_1\ \lambda_2 \leq \widetilde{\lambda}_2 \leq \gamma_2\ \frac{(1+\epsilon_2)^2}{1 - \epsilon_2^2}\ \lambda_2,$$

which can be made small by controlling ϵ_2.

I also analyze the angle between principal eigenspaces of L and L_c. I follow Li (1994) and split the eigendecompositions of $L = U \Lambda U^\top$ and $P^\top L_c P = P^\top \widetilde{U} \widetilde{\Lambda} \widetilde{U}^\top P$ as

$$L = (U_k, U_{k^\perp}) \begin{pmatrix} \Lambda_k & \\ & \Lambda_{k^\perp} \end{pmatrix} \begin{pmatrix} U_k^\top \\ U_{k^\perp}^\top \end{pmatrix} \quad P^\top L_c P = (P^\top \widetilde{U}_k, P^\top \widetilde{U}_{k^\perp}) \begin{pmatrix} \widetilde{\Lambda}_k & \\ & \widetilde{\Lambda}_{k^\perp} \end{pmatrix} \begin{pmatrix} \widetilde{U}_k^\top P \\ \widetilde{U}_{k^\perp}^\top P \end{pmatrix},$$

where $\widetilde{\Lambda}_k$ and $\widetilde{U}_k$ are defined analogously to Λ_k and U_k. Davis and Kahan (1970) defined the *canonical angles* between the spaces spanned by U_k and $P^\top \widetilde{U}_k$ as the singlular values of the matrix

$$\Theta(U_k, P^\top \widetilde{U}_k) \triangleq \arccos(U_k^\top P^\top \widetilde{U}_k \widetilde{U}_k^\top P U_k)^{-1/2},$$

see also (Stewart, 1990). The smaller the sinus of the canonical angles are the closer the two subspaces lie. The following theorem reveals a connection between the Frobenius norm of the sinus of the canonical angles and restricted spectral approximation.

Theorem 14 (Eigenspace approximation) *If L_c and L are $(\mathbf{U}_k, \epsilon_k)$-similar then*

$$\left\|\sin\Theta\big(U_k, P^\top \widetilde{U}_k\big)\right\|_F^2 \leq \frac{1}{\lambda_{k+1}-\lambda_k}\left(\sum_{i\leq k}\lambda_i\left(\frac{(1+\epsilon_i)^2}{\gamma_1}-1\right)+\lambda_k\sum_{i\leq k}\epsilon_i\right),$$

Note that the theorem above utilizes all ϵ_i with $i \leq k$, corresponding to the restricted spectral approximation constants for $\mathbf{R} = \mathbf{U}_i$, respectively. However, all these can be trivially relaxed to ϵ_k, since $\epsilon_i \leq \epsilon_k$ for all $i \leq k$.

3.3. Implications for Graph Partitioning

One of the most popular applications of coarsening is to accelerate graph partitioning (Hendrickson and Leland, 1995; Karypis and Kumar, 1998; Kushnir et al., 2006; Dhillon et al., 2007; Wang et al., 2014). In the following, I provide a rigorous justification for this choice by showing that if the (Laplacian consistent) coarsening is done well and G_c contains a good normalized cut, then so will G. For the specific case of spectral clustering, I also provide an explicit bound on the coarse solution quality.

3.3.1. Existence Results

For consistent coarsening, the spectrum approximation results presented previously imply similarities between the cut-structures of G_c and G.

To formalize this intuition, the conductance of any subset $\mathcal{S}$ of $\mathcal{V}$ is defined as

$$\phi(\mathcal{S}) \triangleq \frac{w(\mathcal{S},\bar{\mathcal{S}})}{\min\{w(\mathcal{S}), w(\bar{\mathcal{S}})\}},$$

where $\bar{\mathcal{S}} = \mathcal{V}\setminus\mathcal{S}$ is the complement set, $w(\mathcal{S},\bar{\mathcal{S}}) = \sum_{v_i\in\mathcal{S}, v_j\in\bar{\mathcal{S}}} w_{ij}$ is the weight of the cut and $w(\mathcal{S}) = \sum_{v_i\in\mathcal{S}}\sum_{v_j\in\mathcal{V}} w_{ij}$ is the volume of $\mathcal{S}$.

The *k-conductance* of a graph measures how easy it is to cut it into k disjoint subsets $\mathcal{S}_1,\ldots,\mathcal{S}_k \subset \mathcal{V}$ of balanced volume:

$$\phi_k(G) = \min_{\mathcal{S}_1,\ldots,\mathcal{S}_k}\max_i \phi(\mathcal{S}_i),$$

where the minimum is taken over all possible k-partitionings $(\mathcal{S}_1,\ldots,\mathcal{S}_k)$. The smaller $\phi_k(G)$ is, the easier it is to partition our graph.

As it turns out, restricted spectral approximation can be used to relate the conductance of the original and coarse graphs. To state the result, it will be useful to denote by D the diagonal degree matrix and further to suppose that V_k contains the first k eigenvectors of the normalized Laplacian $L^n = D^{-1/2}LD^{-1/2}$, whose eigenvalues are $0 = \mu_1 \leq \cdots \leq \mu_k$.

Theorem 15 *For any graph G and integer $2 \leq k \leq \lfloor n/2\rfloor$, if L_c and L are $(\mathbf{R}_{2k}, \epsilon_{2k})$-similar combinatorial Laplacian matrices then*

$$\phi_k(G) \leq \phi_k(G_c) = O\left(\sqrt{\frac{\gamma_2\,(1+\epsilon_{2k})^2\xi_k(G)}{1-\epsilon_{2k}^2(\mu_{2k}/\mu_2)}\phi_k(G)}\right)$$

with $\mathbf{R}_{2k} = span(D^{-1/2}V_{2k})$ and $\xi_k(G) \leq \log k$, whenever $\epsilon_{2k}^2 < \mu_2/\mu_{2k}$. If G is planar then $\xi_k(G) \leq 1$ and if G excludes K_h as a minor (but is not planar) then $\xi_k(G) \leq h^4$. For $k = 2$, supposing that L_c and L are $(\mathbf{R}_2, \epsilon_2)$-similar, we additionally have

$$\phi_2(G) \leq \phi_2(G_c) \leq 2\sqrt{\frac{\gamma_2\,(1+\epsilon_2)^2}{1-\epsilon_2^2}\phi_2(G)}.$$

The above theorem gives a non-constructive result: it does not reveal how to find the optimal partitioning, but provides conditions such that the latter is of similar quality in the two graphs.

3.3.2. Spectral Clustering

It is also possible to derive approximation results about the solution quality of unsupervised learning algorithms that utilize the first k eigenvectors in order to partition G. I focus here on spectral clustering. To perform the analysis, let U_k and $P^\top\widetilde{U}_k$ be the spectral embedding of the vertices w.r.t. L and L_c, respectively, and define the optimal partitioning as

$$\mathcal{P}^* = \operatorname*{arg\,min}_{\mathcal{P}=\{\mathcal{S}_1,\ldots,\mathcal{S}_k\}} \mathcal{F}_k(U_k,\mathcal{P}) \quad\text{and}\quad \widetilde{\mathcal{P}}^* = \operatorname*{arg\,min}_{\mathcal{P}=\{\mathcal{S}_1,\ldots,\mathcal{S}_k\}} \mathcal{F}_k(P^\top\widetilde{U}_k,\mathcal{P}), \tag{4}$$

where, for any embedding X, the k-means cost induced by partitioning $\mathcal{V}$ into clusters $\mathcal{S}_1,\ldots,\mathcal{S}_k$ is defined as

$$\mathcal{F}_k(X,\mathcal{P}) \triangleq \sum_{z=1}^{k}\sum_{v_i,v_j\in\mathcal{S}_z}\frac{\|X(i,:)-X(j,:)\|_2^2}{2\,|\mathcal{S}_z|}.$$

One then measures the quality of $\widetilde{\mathcal{P}}^*$ by examining how far the correct minimizer $\mathcal{F}_k(U_k,\mathcal{P}^*)$ is to $\mathcal{F}_k(U_k,\widetilde{\mathcal{P}}^*)$. Boutsidis et al. (2015) noted that if the two quantities are close then, despite the clusters themselves possibly being different, they both feature the same quality w.r.t. the k-means objective.

An end-to-end control of the k-means error is obtained by combining the inequality derived by Loukas and Vandergheynst (2018), based on the works of (Boutsidis et al., 2015; Yu et al., 2014; Martin et al., 2018), $|\mathcal{F}_k(U_k,\mathcal{P}^*)^{1/2} - \mathcal{F}_k(U_k,\widetilde{\mathcal{P}}^*)^{1/2}| \leq 2\sqrt{2}\,\left\|\sin\Theta\big(U_k,P^\top\widetilde{U}_k\big)\right\|_F$ with Theorem 14:

Corollary 16 *If L_c and L are $(\mathbf{U}_k, \epsilon_k)$-similar then*

$$\left(\mathcal{F}_k(U_k, \mathcal{P}^*)^{1/2} - \mathcal{F}_k(U_k, \widetilde{\mathcal{P}}^*)^{1/2}\right)^2 \leq \frac{8}{\lambda_{k+1} - \lambda_k}\left(\sum_{i\leq k}\lambda_i\left(\frac{(1+\epsilon_i)^2}{\gamma_1} - 1\right) + \lambda_k \sum_{i\leq k}\epsilon_i\right).$$

Contrary to previous analysis (Loukas and Vandergheynst, 2018), the approximation result here is applicable to any number of levels and it can be adapted to hold for the eigenvectors of the normalized Laplacian[4]. Nevertheless, it should be stressed that at this point it is an open question whether the above analysis yields benefits over other approaches tailored especially to the acceleration of spectral clustering. A plethora of such specialized algorithms are known (Tremblay and Loukas, 2019)—arguing about the pros and cons of each extends beyond the scope of this work.

4. Graph Coarsening by Local Variation

This section proposes algorithms for Laplacian consistent coarsening. I suppose that L is a combinatorial graph Laplacian and, given a subspace $\mathbf{R}$ and a target graph size n, aim to find an $(\mathbf{R}, \epsilon)$-similar Laplacian L_c of size $n \times n$ with ϵ smaller than some threshold ϵ'.

Local variation algorithms differ only in the type of contraction sets that they consider. For instance, the edge-based local variation algorithm only contracts edges, whereas in the neighborhood-based variant each contraction set is a subset of the neighborhood of a vertex. Otherwise, all local variation algorithms follow the same general methodology and aim to minimize an upper bound of ϵ. To this end, two bounds are exploited: First, L_c is shown to be $(\mathbf{R}, \epsilon)$-similar to L with $\epsilon \leq \prod_\ell(1+\sigma_\ell) - 1$, where the *variation cost* σ_ℓ depends only on previous levels (see Section 4.1). The main difficulty with minimizing σ_ℓ is that it depends on interactions between contraction sets. For this reason, the second bound shows that these interactions can be decoupled by considering each *local variation cost*, i.e., the cost of contracting solely the vertices in $\mathcal{V}_{\ell-1}^{(r)}$, independently on a slightly modified subgraph (see Section 4.2). Having achieved this, Section 4.3 considers ways of efficiently identifying disjoint contraction sets with small local variation cost.

4.1. Decoupling Levels and the Variation Cost

Guaranteeing restricted spectral approximation w.r.t. subspace $\mathbf{R}$ boils down to minimizing at each level ℓ the *variation cost*

$$\sigma_\ell = \|\Pi_\ell^\perp A_{\ell-1}\|_{L_{\ell-1}} = \|S_{\ell-1}\Pi_\ell^\perp A_{\ell-1}\|_2,$$

where $L_{\ell-1} = S_{\ell-1}^\top S_{\ell-1}$ and $\Pi_\ell^\perp = I - P_\ell^+ P_\ell$ is a projection matrix. Matrix $A_{\ell-1}$ captures two types of information:

4. For the normalized Laplacian, one should perform (combinatorial) Laplacian consistent coarsening on a modified eigenspace, as in the proof of Theorem 15.

1. Foremost, it encodes the behavior of the target Laplacian matrix L w.r.t. $\mathbf{R}$. This is clearly seen in the first level, for which one has that $A_0 = VV^\top L^{+1/2}$ with $V \in \mathbb{R}^{N\times k}$ being an orthonormal basis of $\mathbf{R}$.

2. When $\ell > 1$ one needs to consider A_0 in view of the reduction done in previous levels. The necessary modification turns out to be $A_{\ell-1} = B_{\ell-1}(B_{\ell-1}^\top L_{\ell-1} B_{\ell-1})^{+1/2}$, with $B_{\ell-1} = P_{\ell-1}B_{\ell-2} \in \mathbb{R}^{N_{\ell-1}\times N}$ expressed in a recursive manner and $B_0 = A_0$.

The following result makes explicit the connection between ϵ and σ_ℓ.

Proposition 17 *Matrices L_c and L are $(\mathbf{R}, \epsilon)$-similar with $\epsilon \leq \prod_{\ell=1}^c(1+\sigma_\ell) - 1$.*

Crucially, the previous result makes it possible to design a multi-level coarsening greedily, by starting from the first level and optimizing following levels one at a time. Thus, every local variation algorithm operates in the following manner:

Algorithm 1 Multi-level coarsening

input: Combinatorial Laplacian L, threshold ϵ', and target size n.
Set $\ell \leftarrow 0$, $L_\ell \leftarrow L$, and $\epsilon_\ell \leftarrow 0$.
while $N_\ell > n$ and $\epsilon_\ell < \epsilon'$ **do**
  $\ell \leftarrow \ell + 1$
  Coarsen $L_{\ell-1}$ using Algorithm 2 with threshold $\sigma' = \frac{1+\epsilon'}{1+\epsilon_{\ell-1}} - 1$ and target size n. Let L_ℓ be the resulting Laplacian of size N_ℓ with variation cost σ_ℓ.
  $\epsilon_\ell \leftarrow (1+\epsilon_{\ell-1})(1+\sigma_\ell) - 1$.
return L_ℓ

Algorithm 1 returns a Laplacian matrix L_c that is $(\mathbf{R}, \epsilon)$-similar to L with $\epsilon \leq \epsilon_c \leq \epsilon'$, where c is the last level ℓ. On the other hand, setting ϵ' to a large value ensures that the same algorithm always attains the target reduction at the expense of loose restricted approximation guarantees.

Remark. The variation cost simplifies when $\mathbf{R}$ is an eigenspace of L. I demonstrate this for the choice of $\mathbf{U}_k$, though an identical argument can be easily derived for any eigenspace. Denote by Λ the diagonal $N \times N$ eigenvalue matrix placed from top-left to bottom-right in non-decreasing order and by U the respective full eigenvector matrix. Furthermore, let Λ_k be the $k \times k$ sub-matrix of Λ with the smallest k eigenvalues in its diagonal. By the unitary invariance of the spectral norm, it follows that $\sigma_0 = \|\Pi_1^\perp U_k U_k^\top L^{+1/2}\|_{L_0} = \|\Pi_1^\perp U_k U_k^\top L^{+1/2} U\|_{L_0} = \|\Pi_1^\perp U_k U_k^\top U \Lambda^{+1/2}\|_{L_0}$. Simplifying and eliminating zero columns, one may redefine $B_0 = U_k \Lambda_k^{+1/2} \in \mathbb{R}^{N\times k}$, such that once more $\sigma_0 = \|\Pi_1^\perp B_0\|_{L_0}$. This is computationally attractive because now at each level one needs to take the pseudo-inverse-square-root of a $k \times k$ matrix $B_{\ell-1}^\top L_{\ell-1} B_{\ell-1}$, with $k \ll N$.

4.2. Decoupling Contraction Sets and Local Variation

Suppose that $\Pi_{\mathcal{C}}^{\perp}$ is the (complement) projection matrix obtained by contracting solely the vertices in set $\mathcal{C}$, while leaving all other vertices in $\mathcal{V}_{\ell-1}$ untouched:

$$\left[\Pi_{\mathcal{C}}^{\perp}\, x\right](i) = \begin{cases} x(i) - \sum_{v_j \in \mathcal{C}} \frac{x(j)}{|\mathcal{C}|} & \text{if } v_i \in \mathcal{C} \\ 0 & \text{otherwise.} \end{cases}$$

(Here, for convenience, the level index is suppressed.)

Furthermore, let $L_{\mathcal{C}}$ be the $N_{\ell-1} \times N_{\ell-1}$ combinatorial Laplacian whose weight matrix is

$$[W_{\mathcal{C}}]\,(i,j) = \begin{cases} W_{\ell-1}(i,j) & \text{if } v_i, v_j \in \mathcal{C} \\ 2\, W_{\ell-1}(i,j) & \text{if } v_i \in \mathcal{C} \text{ and } v_j \notin \mathcal{C} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

That is, $W_{\mathcal{C}}$ is zero everywhere other than at the edges touching at least one vertex in $\mathcal{C}$. The following proposition shows us how to decouple the contribution of each contraction set to the variation cost.

Proposition 18 *The variation cost is bounded by*

$$\sigma_\ell^2 \leq \sum_{\mathcal{C} \in \mathcal{P}_\ell} \|\Pi_{\mathcal{C}}^{\perp}\, A_{\ell-1}\|_{L_{\mathcal{C}}}^2,$$

where $\mathcal{P}_\ell = \{\mathcal{V}_{\ell-1}^{(1)}, \ldots, \mathcal{V}_{\ell-1}^{(N_\ell)}\}$ *is the family of contraction sets of level* ℓ.

The above argument therefore entails bounding the, difficult to optimize, variation cost as a function of locally computable and independent costs $\|\Pi_{\mathcal{C}}^{\perp}\, A_{\ell-1}\|_{L_{\mathcal{C}}}^2$. The obtained expression is a relaxation, as it assumes that the interaction between contraction sets will be the worst possible. It might be interesting to notice that the quality of the relaxation depends on the weight of the cut between contraction sets. Taking the limit, the inequality converges to an equality as the weight of the cut shrinks. Also of note, the bound becomes tighter the larger the per-level dimensionality reduction requested (the smaller $N_\ell = |\mathcal{P}_\ell|$ is, the fewer inequalities are involved in the derivation).

4.3. Local Variation Coarsening Algorithms

Starting from a *candidate family* $\mathcal{F}_\ell = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \ldots\}$, that is, an appropriately sized family of candidate contraction sets, the strategy will be to search for a small *contraction family* $\mathcal{P}_\ell = \{\mathcal{V}_{\ell-1}^{(1)}, \ldots, \mathcal{V}_{\ell-1}^{(N_\ell)}\}$ with minimal variation cost σ_ℓ ($\mathcal{P}_\ell$ is valid if it partitions $\mathcal{V}_{\ell-1}$ into N_ℓ contraction sets). Every coarse vertex $v_r' \in \mathcal{V}_\ell$ is then formed by contracting the vertices in $\mathcal{V}_{\ell-1}^{(r)}$.

As a thought experiment, suppose that set $\mathcal{C} \in \mathcal{F}_\ell$ is chosen to be part of $\mathcal{P}_\ell$. From the decoupling argument, its contribution to σ_ℓ^2 will be at most $\|\Pi_{\mathcal{C}}^{\perp}\, A_{\ell-1}\|_{L_{\mathcal{C}}}^2$ independently of

how other candidate sets are chosen. Moreover, the selection will reduce $N_{\ell-1}$ by $|\mathcal{C}| - 1$ vertices. Thus, one needs to look for the non-singleton candidate sets $\mathcal{C}$ with cost

$$\text{cost}_\ell(\mathcal{C}) \triangleq \frac{\|\Pi_{\mathcal{C}}^{\perp}\, A_{\ell-1}\|_{L_{\mathcal{C}}}^2}{|\mathcal{C}| - 1} \tag{6}$$

that is as small as possible. I refer to (6) as *local variation cost* because it captures the maximal variation of all signals from an appropriate subspace (implied by $A_{\ell-1}$) with support on $\mathcal{C}$. On the other hand, since any permissible contraction family $\mathcal{P}_\ell$ should be a partitioning of $\mathcal{V}_{\ell-1}$, choosing $\mathcal{C}$ precludes us from selecting any $\mathcal{C}'$ with which it intersects.

Based on this intuition, Algorithm 2 sequentially examines candidate sets from $\mathcal{F}_\ell$, starting from those with minimal cost. To decide whether a candidate set $\mathcal{C}$ will be added to $\mathcal{P}_\ell$ the algorithm asserts that all vertices in $\mathcal{C}$ are unmarked—essentially enforcing that all contraction sets are disjoint. Accordingly, as soon as $\mathcal{C}$ is added to $\mathcal{P}_\ell$, all vertices that are in $\mathcal{C}$ become marked. Candidate sets with marked vertices are pruned ($\mathcal{C}' \leftarrow \mathcal{C} \setminus \mathsf{marked}$) and their cost is updated. The algorithm terminates if either the target reduction is achieved, the error threshold is exceeded, or no candidate sets remain. Even though this remains implicit in the discussion, if at termination $\mathcal{P}_\ell$ does not cover every vertex of $\mathcal{V}_{\ell-1}$, then I compliment it with singleton sets, featuring one vertex each (and zero cost).

Algorithm 2 Single-level coarsening by local variation

input: Combinatorial Laplacian $L_{\ell-1}$, threshold σ', and target size n.
Form the family of candidate sets $\mathcal{F}_\ell = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \ldots\}$ (algorithm-specific step).
$N_\ell \leftarrow N_{\ell-1}$, $\mathsf{marked} \leftarrow \varnothing$, $\sigma_\ell^2 \leftarrow 0$.
Sort $\mathcal{F}_\ell$ in terms of increasing $\text{cost}_\ell(\mathcal{C})$.
while $|\mathcal{F}_\ell| > 0$ and $N_\ell > n$ and $\sigma_\ell \leq \sigma'$ **do**
  Pop the candidate set $\mathcal{C}$ of minimal cost s from $\mathcal{F}_\ell$.
  if all vertices of $\mathcal{C}$ are not marked and $\sigma' \geq \sqrt{\sigma_\ell^2 + (|\mathcal{C}| - 1)s}$ **then**
    $\mathsf{marked} \leftarrow \mathsf{marked} \cup \mathcal{C}$, $\mathcal{P}_\ell \leftarrow \mathcal{P}_\ell \cup \mathcal{C}$, $N_\ell \leftarrow N_\ell - |\mathcal{C}| + 1$, $\sigma_\ell^2 \leftarrow \sigma_\ell^2 + (|\mathcal{C}| - 1)s$
  else
    $\mathcal{C}' \leftarrow \mathcal{C} \setminus \mathsf{marked}$
    if $|\mathcal{C}'| > 1$ **then**
      Compute $\text{cost}_\ell(\mathcal{C}')$ and insert $\mathcal{C}'$ into $\mathcal{F}_\ell$ while keeping the latter sorted.
Form the $N_\ell \times N_{\ell-1}$ coarsening matrix P_ℓ based on $\mathcal{P}_\ell$.
return $L_\ell \leftarrow P_\ell^{\mp} L_{\ell-1} P_\ell^{+}$ and σ_ℓ

Undeniably, Algorithm 2 is only one of the possible ways to select a partitioning of small variation cost. However, this algorithm stands out from other algorithms I experimented with, as it is efficient when the subspace of interest is an eigenspace (e.g., $V = U_k$), k is small, and the families $\mathcal{F}_\ell$ have been selected appropriately. Denote by $\Phi = \max_\ell \sum_{\mathcal{C} \in \mathcal{F}_\ell} |\mathcal{C}|$ the maximum number of vertices in all candidate sets and by $\delta = \max_{\ell,\, \mathcal{C} \in \mathcal{F}_\ell} |\mathcal{C}|$ the cardinality of the maximum candidate set—I refer to these measures as *family weight* and *width*, respectively. Choosing $\mathbf{R} = \mathbf{U}_k$, the computational complexity of Algorithm 2 is $\tilde{O}(ckM + k^2 N + ck^3 + c\,\Phi\left(\min\{k^2\delta + k\delta^2,\ k\delta^2 + \delta^3\} + \log \max_\ell |\mathcal{F}_\ell|\right))$, which up to polylog factors is linear on the number of edges, vertices, and Φ (see Appendix B for details).

If computational complexity is of no concern, one may consider the following two more sophisticated algorithms: *The optimal algorithm.* Given a candidate family, the algorithm that optimally minimizes the sum of local variation costs constructs a graph with one vertex for each subset of a candidate set and adds an edge between every two vertices whose respective sets have a non-empty intersection. It then selects $\mathcal{P}_\ell$ as the maximum independent set of minimal weight (the weight of each vertex is a local variation cost w.r.t a set). Unfortunately, even if the size of this graph is polynomial in N, this problem cannot be solved efficiently, since the minimum-weight independent set problem is NP-hard. Nevertheless, for the specific case where candidate sets correspond to edges the problem simplifies to a minimum-weight matching problem, which can be computed in $O(N_{\ell-1}^3)$ time exactly, whereas a $(2+\delta)$-approximation can be found much faster (Paz and Schwartzman, 2017). *The quadratic variant.* A second possibility is to proceed as with Algorithm 2, but to prune each $\mathcal{C}' \in \mathcal{F}_\ell$ after a set $\mathcal{C}$ is added to $\mathcal{P}_\ell$. The numerical experiments indicated that this additional step may improve the coarsening quality slightly, but it is not recommended for large graphs as it introduces a quadratic dependency of the complexity on N.

4.3.1. Candidate Contraction Families

To keep coarsening efficient, I focus on families of linear weight and almost constant width. Two possibilities are considered:

Edge-based. Here $\mathcal{F}_\ell$ contains one candidate set for each edge of $G_{\ell-1}$. This is a natural choice for coarsening—indeed, most coarsening algorithms in the literature use some form of edge contraction. It is straightforward to see that in this case $\Phi = 2M$ and $\delta = 2$, meaning that the expression of the computational complexity simplifies to $\tilde{O}(ckM + ck^3 + k^2N)$. The drawback of contracting edges is that at each level the graph size can only be reduced by at most a factor of 2, meaning that a large number of levels is necessary to achieve significant reduction[5].

Neighborhood-based. A more attractive choice is to construct one candidate set for the neighborhood of each vertex, including the vertex itself. Denoting by Δ the largest combinatorial degree and since $\Phi = 2M$, the complexity here is bounded by $\tilde{O}(cM(k + \min\{k^2\Delta + k\Delta^2,\ k\Delta^2 + \Delta^3\}) + ck^3 + k^2N)$. The experiments show that the neighborhood-based variant generally achieves better reduction while being marginally slower than the edge-based variant.

As a final remark, when G is dense, the dependency on M can be dropped by sparsifying the graph before using Algorithm 1.

5. Numerical Results

The evaluation was performed on four representative graphs, each exhibiting different structural characteristics:

5. In practice, depending on the graph in question, the per-level reduction ratio r_ℓ is usually between 0.35 and 0.45.

- *Yeast.* Protein-to-protein interaction network in budding yeast, analyzed by Jeong et al. (2001). The network has $N = 1458$ vertices, $M = 1948$ edges, diameter of 19, and degree between 1 and 56.
- *Airfoil.* Finite-element graph obtained by airflow simulation Preis and Diekmann (1997), consisting of $N = 4000$ vertices, $M = 11490$ edges, diameter of 65, and degree between 1 and 9.
- *Minnesota.* Road network with $N = 2642$ vertices, $M = 3304$ edges, diameter of 99, and degree between 1 and 5 (Gleich, 2008).
- *Bunny.* Point cloud consisting of $N = 2503$ vertices, $M = 65490$ edges, diameter of 15, and degree between 13 and 97 (Turk and Levoy, 1994). The point cloud has been sub-sampled from its original size.

I compare to the following methods for multi-level graph reduction:

- *Heavy edge matching.* At each level of the scheme, the contraction family is obtained by computing a maximum-weight matching with the weight of each contraction set (v_i, v_j) calculated as $w_{ij}/\max\{d_i, d_j\}$. In this manner, heavier edges connecting vertices that are well separated from the rest of the graph are contracted first. Heavy edge matching was first introduced in the algebraic multigrid literature and, perhaps due to its simplicity, its variants have been repeatedly used for partitioning (Karypis and Kumar, 1998; Dhillon et al., 2007) and drawing (Walshaw, 2000; Hu, 2005) graphs, as well as more recently in graph convolutional neural networks (Defferrard et al., 2016).
- *Algebraic distance.* This method differs from heavy edge matching in that the weight of each candidate set $(v_i, v_j) \in \mathcal{E}$ is calculated as $(\sum_{q=1}^{Q}(x_q(i) - x_q(j))^2)^{1/2}$, where x_k is an N-dimensional test vector computed by successive sweeps of Jacobi relaxation. The complete method is described by Ron et al. (2011), see also (Chen and Safro, 2011). As recommended by the authors, I performed 20 relaxation sweeps. Further, I set the number of test vectors Q to equal the dimension k of the space I aimed to approximate (a simple rank argument shows that $Q \geq k$ for the test vectors to span the space).
- *Affinity.* This is a vertex proximity heuristic in the spirit of the algebraic distance that was proposed by Livne and Brandt (2012) in the context of their work on the lean algebraic multigrid. As per the author suggestion, the $Q = k$ test vectors are here computed by a single sweep of a Gauss-Seidel iteration.
- *Kron reduction.* At each level, the graph size is reduced by selecting a set of vertices of size $N/2$ (corresponding to the positive entries of the last eigenvector of L) and applying Kron reduction. The method, which was proposed by Shuman et al. (2016), is not strictly a coarsening method as it rewires the vertices of the reduced graph, resulting in significantly denser graphs[6]. Unfortunately, the rewiring step en-

6. As suggested by the authors, the sparsity of reduced graphs can be controlled by spectral sparsification. The sparsification step was not included in the numerical experiments since it often resulted in increased errors.

tails finding the Schur complement of a large Laplacian submatrix and thus generally exhibits $O(N^3)$ complexity, rendering it prohibitive for graphs of more than a few thousand vertices. Despite these drawbacks, the method is quite popular because of its elegant theoretical guarantees (Dörfler and Bullo, 2013).

Depending on how the edge matching is constructed, different variants of edge contraction methods can be implemented. At the two extremes of the complexity spectrum one finds the maximum matching of minimum weight at $O(N^3)$ complexity (Galil, 1986) or greedily constructs a matching by visiting vertices in random order and inducing $O(M)$ overhead (Dhillon et al., 2007).

For consistency, I implemented all edge-based methods by combining Algorithm 2 with an edge-based family and substituting the local variation cost with the (negative) method-specific edge weight. This generally yields matchings of better quality (heavier weight) than visiting vertices in a predefined order, at the expense of the marginally larger $O(M \log M)$ complexity necessary for sorting the edge weights. The choice is also motivated by the observation that the computational bottleneck of (sophisticated) edge contraction methods lies in the edge weight computation. For all experiments, I set $\epsilon' = \infty$ aiming for a fixed reduction rather than a restricted spectral approximation guarantee. The code reproducing the experiments can be accessed online [7].

5.1. Restricted Spectral Approximation

The first experiment tests how well L_c approximates the action of L w.r.t. the subspace U_k of the smallest variation. In other words, for each method, I plot the smallest ϵ such that the following equation holds:

$$\|x - \widetilde{x}\|_L \leq \epsilon \, \|x\|_L \quad \text{for all} \quad x \in \mathbf{U}_k. \tag{7}$$

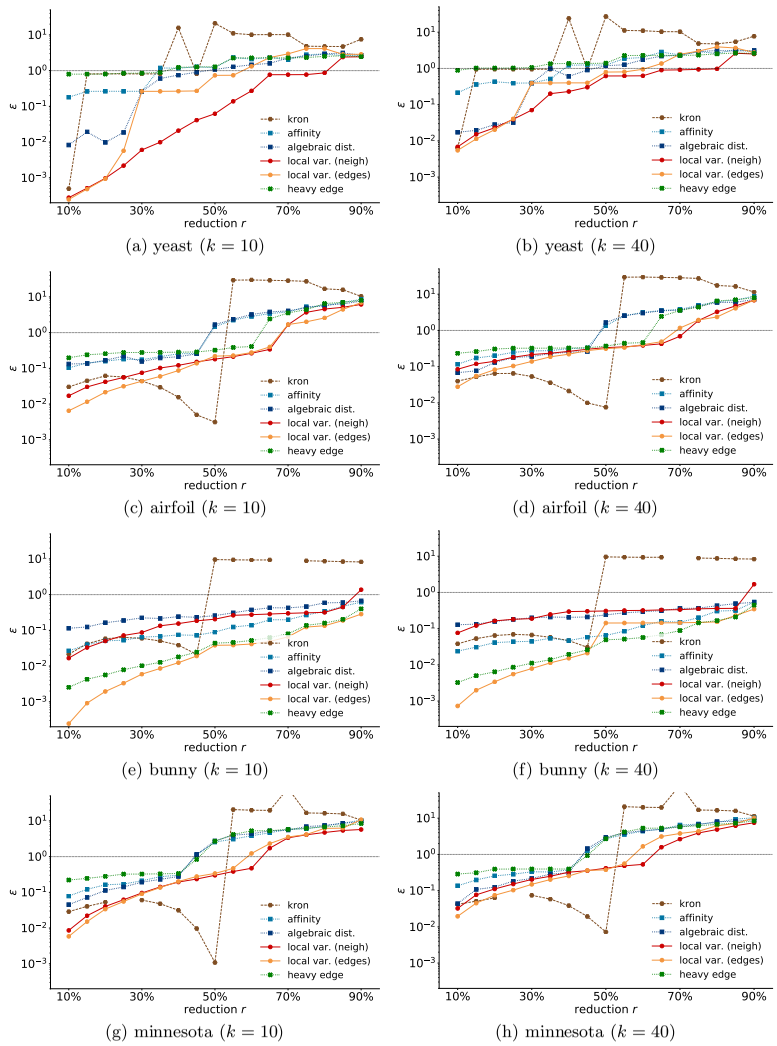
The results are summarized in Figure 2 for two representative subspaces of size $k = 10$ and $k = 40$. With the exception of the Kron reduction that was repeated 10 times, all methods are deterministic and thus were run only once.

Overall, it can be seen that local variation methods outperform other coarsening methods in many problem instances. The gap is particularly prominent for large reductions, where multiple levels are employed. Neighborhood-based contraction yields the best result overall, mainly because it achieves the same reduction in fewer levels. Interestingly, local variation (and coarsening) methods in many cases also outperform Kron reduction, even though the latter is more demanding computationally.

I elaborate further on four points stemming from the results:

In most instances, a reduction of up to 70% is feasible while maintaining a decent approximation. The attained approximation is a function of the graph in question and k. Nevertheless, in almost all experiments, the best coarsening method could reduce the graph size by at least 70% while also ensuring that $\epsilon < 1$ (horizontal black line). This is an

7. github.com/loukasa/graph-coarsening/tree/v1.1 (DOI 175851068)



(a) yeast ($k = 10$) (b) yeast ($k = 40$)
(c) airfoil ($k = 10$) (d) airfoil ($k = 40$)
(e) bunny ($k = 10$) (f) bunny ($k = 40$)
(g) minnesota ($k = 10$) (h) minnesota ($k = 40$)

Figure 2: Quality of approximation comparison for four representative graphs (rows) and two subspace sizes (columns).

encouraging result, illustrating that significant dimensionality reduction is often possible, without sacrificing too much the solution quality.

Following intuition, it is generally harder to approximate subspaces of larger dimension k, but not excessively so. Increasing k from 10 to 40 in most cases increases ϵ only slightly. The only case where the approximation becomes profoundly better with small subspaces is with small reduction ratios r. For instance, coarsening the yeast graph results in an impressive approximation for all $r < 30\%$ when $k = 10$, whereas ϵ increases almost by an order of magnitude when k becomes 40.

Kron reduction is an effective way to half the graph size but can result in poor approximation otherwise. If one is willing to sacrifice in terms of efficiency, Kron reduction effectively reduces the graph size by a factor of two (with the exception of the yeast graph). What might be startling is that the method behaves poorly for different reduction ratios. Three main factors cause this deterioration of performance. First, the sampling set is constructed based on the sign pattern of u_N, i.e., by keeping the vertices v_i for which $u_N(i) \geq 0$, and has cardinality close to $N/2$ (Shuman et al., 2016). Therefore, if at any level one tries to reduce the graph size by less than half, the last eigenvector heuristic cannot be used precisely. Second, numerical instability issues sometimes manifest when r exceeds 50%. I was not able to improve the implementation featured in the PyGSP toolbox, and in my experiments some problem instances could not be solved successfully (hence the missing markers). The third reason is described next.

Coarse levels should aim to approximate the original graph and not the proceeding levels. The conventional approach in multi-level schemes is to aim at each level to approximate as closely as possible the graph of the previous level. This can lead to a sudden increase of error at consecutive levels (e.g. notice the minnesota error as r approaches 50%) as decisions early in the scheme can have a significant impact later on. On the other hand, local variation methods modify the cost function minimized at each level (see Proposition 17), resulting in smoother transitions between levels and tighter approximations at large r.

5.2. Spectrum Approximation

The second part of the experiments examines the coarsening through the lens of spectral graph theory. The premise is that, since the spectrum of the Laplacian distills information about the graph structure, one may interpret the spectral distance as a proxy for the structural similarity of the two graphs. This is by no means a new idea—the Laplacian spectrum is a common ingredient in accessing graph similarity (Wilson and Zhu, 2008).

Tables 1 and 2 report the mean relative eigenvalue error defined as $\frac{1}{k}\sum_{i=1}^{k}\frac{|\widetilde{\lambda}_i-\lambda_i|}{\lambda_i}$ for two representative k, respectively 10 and 40. The results for $k = 80$ were consistent with those presented here, and are not reported here for reasons of brevity.

As expected, the reduction ratio plays a significant role in the closeness of Laplacian spectra. Indeed, for most cases, the eigenvalue error jumps by almost an order of magnitude whenever r increases by 20%. Yet, also in most cases, acceptable errors can be achieved even when

	r	heavy edge	local var. (edges)	local var. (neigh.)	algebraic distance	affinity	Kron reduction
yeast	30%	0.284	0.123	**0.003**	0.126	0.164	0.054
	50%	1.069	0.460	**0.034**	0.759	0.877	1.321
	70%	5.126	3.920	**0.409**	3.395	3.140	1.865
airfoil	30%	0.278	**0.036**	0.065	0.219	0.258	0.345
	50%	0.527	0.201	**0.197**	1.221	1.291	0.900
	70%	3.954	1.042	**0.926**	5.562	5.145	2.027
bunny	30%	0.015	**0.006**	0.061	0.244	0.070	0.335
	50%	0.064	**0.046**	0.190	0.401	0.137	0.801
	70%	0.122	**0.080**	0.323	0.694	0.304	1.812
minnesota	30%	0.332	0.088	**0.078**	0.220	0.295	0.324
	50%	1.363	0.431	**0.310**	2.394	2.676	0.873
	70%	7.452	4.553	**1.892**	8.412	9.354	2.068

Table 1: Mean relative error for the first $k = 10$ eigenvalues, for different graphs, reduction ratios, and coarsening methods.

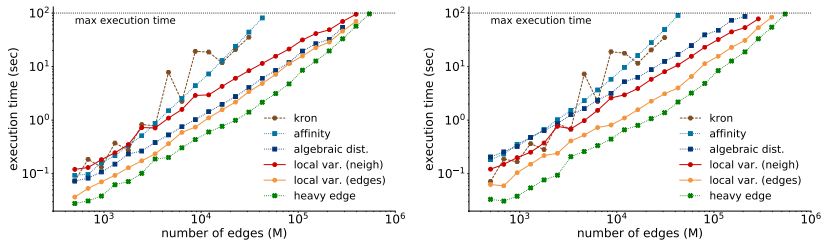
	r	heavy edge	local var. (edges)	local var. (neigh.)	algebraic distance	affinity	Kron reduction
yeast	30%	0.311	0.113	**0.023**	0.113	0.162	0.120
	50%	1.087	0.413	**0.130**	0.522	0.676	1.196
	70%	3.618	2.212	**0.454**	2.387	2.474	1.946
airfoil	30%	0.278	**0.095**	0.181	0.191	0.257	0.366
	50%	0.555	**0.326**	0.349	0.692	0.788	0.955
	70%	2.059	0.905	**0.848**	2.309	2.619	2.141
bunny	30%	0.014	**0.008**	0.085	0.215	0.049	0.294
	50%	0.067	**0.058**	0.181	0.356	0.092	0.660
	70%	0.122	**0.098**	0.299	0.520	0.200	1.192
minnesota	30%	0.358	0.118	**0.115**	0.223	0.308	0.342
	50%	0.967	0.468	**0.383**	1.105	1.296	0.944
	70%	3.588	2.160	**1.610**	4.113	4.133	2.198

Table 2: Mean relative error for the first $k = 40$ eigenvalues, for different graphs, reduction ratios, and coarsening methods.

the coarse graph is as small as one third of the size of the original graph (corresponding to $r = 70\%$).

It might also be interesting to observe that there is a general agreement between the trends reported here and those described in the matrix approximation experiment. In particular, if one sorts the tested methods from best to worse, he/she will obtain an ordering that is generally consistent across the two experiments, with local variation methods giving the best approximation by a significant margin. A case in point: for the maximum ratio, the smallest local variation error is on average 3.5× smaller than the leading state-of-the-art coarsening method. The gain is 2.3× if the Kron reduction is also included in the comparison.

Overall, it can be deduced from these results that local variation methods coarsen a graph in a manner that preserves its spectrum. This is in accordance with the theoretical results. As it was shown by Theorem 13, if L and L_c act similar w.r.t. all vectors in $\mathbf{U}_k$, then their



(a) $k = 10$ (b) $k = 40$

Figure 3: Execution time as a function of the number of edges of the graph.

eigenvalues cannot be far apart. Therefore, by aiming for restricted approximation, local variation methods implicitly also guarantee spectrum approximation.

5.3. Efficiency

The last experiment tests computational efficiency. I adopt a simple approach and aim to coarsen a 10-regular graph of increasing size[8]. I measure the execution time of the six different methods for graph reduction and report the mean over ten iterations while capping computation at 100 seconds.

The results are displayed in Figures 3a and 3b for subspaces of size 10 and 40, respectively. As with most such comparisons, the actual numbers are only indicative and depend on the programming language utilized (Python), processing paradigm (no parallelism was employed), and hardware architecture (2.2GHz CPU)[9].

Focusing on the trends, except for Kron reduction and affinity, most methods scale quasi-linearly with the number of edges. Interestingly, local variation methods are quite competitive and do not sacrifice much as compared to the straightforward heavy edge matching. One can also observe that constructing $\mathcal{F}_{\ell-1}$ based on neighborhoods results in slightly slower computation that with the edge-based method construction because in the latter case the local variation cost can be computed much more efficiently.

6. Conclusions

Graph reduction techniques are commonly used in modern graph processing and machine learning pipelines in order to facilitate the processing and analysis of large graphs. Nevertheless, more often than not these techniques have been designed based on intuition and possess no rigorous justification or provable guarantees.

8. Starting from a ring, this graph is constructed by connecting each vertex with the five neighbors in either side.
9. I expect that a significant speedup can be achieved by compiling the code to native machine instructions as well as by parallelizing the local variation cost computation.

This work considered the graph reduction problem from the perspective of restricted spectral approximation—a modification of the measure utilized for the construction of graph sparsifiers. This measure is especially relevant when restricted to Laplacian eigenspaces of small variation, as it implies strong spectral and cut guarantees. The analysis of restricted spectral approximation has lead to new algorithms for graph coarsening that provably approximate (a portion of) the graph spectrum, as well as the cut structure of a large graph.

Certain important questions remain open at the point of concluding this manuscript. To begin with, I am currently unaware of a rigorous way to determine how much one may benefit from reduction—that is, how small can ϵ be for a specific subspace and target n? In addition, no polynomial-time algorithm for graph coarsening exists that provably approximates the minimal achievable ϵ. Finally, though I lack a formal proof, I also suspect that stronger cut guarantees can be derived from restricted spectral approximation. I argue that the potential of graph reduction cannot be fully realized until these fundamental questions are satisfactorily addressed.

Acknowledgments

I would like to thank the Swiss National Science Foundation (SNSF) for supporting this work in the context of the project "*Deep Learning for Graph-Structured Data*", grant number PZ00P2 179981. I would like to thank Rodrigo C. G. Pena and Nguyen Q. Tran as well as the editor Karsten Borgwardt and the anonymous reviewers for their useful comments.

Appendix A. Deferred Proofs

A.1. Proof of Property 1

I draw up an inductive argument demonstrating that Π is a projection matrix. The *base case*, i.e., that $A_c = P_c^+ P_c$ is a projection matrix follows by the definition of the pseudo-inverse: $A_c A_c = P_c^+ P_c P_c^+ P_c = P_c^+ P_c = A_c$, where one uses the property $P_c = P_c P_c^+ P_c$.

For the *inductive step*, I argue that if $A_{\ell+1}$ is a projection matrix the same holds for

$$A_\ell = P_\ell^+ \, A_{\ell+1} \, P_\ell.$$

To this end, let $P_\ell = U\Sigma V^\top$ be the singular-value decomposition with $\Sigma = (D; 0) \in \mathbb{R}^{N_{\ell+1} \times N_\ell}$ decomposed into the $N_{\ell+1} \times N_{\ell+1}$ diagonal matrix D and the all zero matrix 0. Then

$$A_\ell = V\Sigma^+ U^\top A_{\ell+1} \, U\Sigma V^\top.$$

Recalling that a projection matrix remains projective if it undergoes a similarity transformation, we deduce that $U^\top A_{\ell+1} U$ is also projective and, moreover, if $\Sigma^+ U^\top A_{\ell+1} U\Sigma$ is a projection matrix, so is A_ℓ. However, one may write

$$\Sigma^+ U^\top A_{\ell+1} \, U\Sigma = \begin{pmatrix} D^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^\top A_{\ell+1} \, U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} D^{-1} U^\top A_{\ell+1} \, UD & 0 \\ 0 & 0 \end{pmatrix}.$$

As a block diagonal matrix whose blocks are projective (again $D^{-1}U^\top A_{\ell+1}UD$ is a similarity transformation), $\Sigma^+U^\top A_{\ell+1}U\Sigma$ is also a projection matrix. The proof that Π is a projection matrix concludes by letting the induction unfold backwards from c to 1 and setting $\Pi = A_1$.

A.2. Proof of Theorem 3

The Courant-Fischer min-max theorem for a Hermitian matrix L reads

$$\lambda_k = \min_{\dim(\mathbf{U})=k} \max_{x\in\mathbf{U}} \left\{ \frac{x^\top L x}{x^\top x} \,|\, x \neq 0 \right\}, \tag{8}$$

whereas the same theorem for L_c gives

$$\widetilde{\lambda}_k = \min_{\dim(\mathbf{U}_c)=k} \max_{x_c\in\mathbf{U}_c} \left\{ \frac{x_c^\top L_c x_c}{x_c^\top x_c} \,|\, x_c \neq 0 \right\} = \min_{\dim(\mathbf{U}_c)=k} \max_{Px\in\mathbf{U}_c} \left\{ \frac{x^\top \Pi L \Pi x}{x^\top P^\top P x} \,|\, x \neq 0 \right\},$$

where in the second equality I substitute $L_c = P^{\mp}LP^+$ and $x_c = Px$.

We will need the following result:

Lemma 19 *For any P with full-row rank, the following holds:*

$$\lambda_1(PP^\top)\, x^\top \Pi x \leq x^\top P^\top P x \leq \lambda_n(PP^\top)\, x^\top \Pi x,$$

with $\lambda_1(PP^\top)$ and $\lambda_n(PP^\top)$, respectively the smallest and largest eigenvalues of $PP^\top$.

Proof Set $D = (PP^\top)^+$, which is an $n \times n$ PSD matrix. By the properties of the Moore–Penrose inverse $P^+ = P^\top(PP^\top)^+ = P^\top D$ and therefore $P^\top P = P^\top D D^{-1} P = P^+ D^{-1} P$. Supposing that the eigenvalues of D lie in $[\lambda_{\min}(D), \lambda_{\max}(D)]$ and that P is full row-rank such that $\lambda_{\min}(D) > 0$, one may write

$$\frac{1}{\lambda_{\max}(D)}\, x^\top \Pi x \leq x^\top P^\top P x = x^\top P^+ D^{-1} P x \leq \frac{1}{\lambda_{\min}(D)}\, x^\top \Pi x.$$

To grasp why the aforementioned inequality holds, first use the cyclic property of the trace to obtain

$$x^\top P^+ D^{-1} P x = \mathrm{tr}(x^\top P^+ D^{-1} P x) = \mathrm{tr}(D^{-1} P x x^\top P^+),$$

and further recall that for any symmetric A and PSD matrix B one has $\lambda_{\min}(A)\mathrm{tr}(B) \leq \mathrm{tr}(AB) \leq \lambda_{\max}(A)\mathrm{tr}(B)$, where $\lambda_{\min}(A)$ denotes the smallest eigenvalue of A (and $\lambda_{\max}(A)$ is the largest) (Fang et al., 1994). Set $A = D^{-1}$, which is by assumption symmetric, and $B = Pxx^\top P^+$, which is PSD since its rank is (at most) one with the only (potentially) non-zero eigenvalue exactly $\mathrm{tr}(Pxx^\top P^+) = x^\top P^+ P x = x^\top \Pi x = x^\top \Pi^2 x = \|\Pi x\|_2^2 \geq 0$, where $\Pi = \Pi^2$ due to being a projection matrix (Property 1. The desired inequality then follows since $\lambda_{\min}(D^{-1}) = 1/\lambda_{\max}(D) = \frac{1}{\beta}$, $\lambda_{\max}(D^{-1}) = 1/\lambda_{\min}(D) = \frac{1}{\alpha}$ and once more $\mathrm{tr}(Pxx^\top P^+) = x^\top \Pi x$.

Finally, since P is full row-rank, D is invertible meaning $\lambda_{\min}(D) = 1/\lambda_n(PP^\top)$ and $\lambda_{\max}(D) = 1/\lambda_1(PP^\top)$. ■

From the above, it is deduced that

$$\begin{aligned}\widetilde{\lambda}_k &\geq \min_{\dim(\mathbf{U}_c)=k} \max_{Px\in\mathbf{U}_c} \left\{ \frac{x^\top \Pi L \Pi x}{\lambda_n(PP^\top)\, x^\top \Pi x} \,|\, x \neq 0 \right\} \\ &= \frac{1}{\lambda_n(PP^\top)} \min_{\dim(\mathbf{U})=k, \mathbf{U}\subseteq \mathrm{im}(\Pi)} \max_{x\in\mathbf{U}} \left\{ \frac{x^\top L x}{x^\top x} \,|\, x \neq 0 \right\},\end{aligned}$$

where the equality holds since Π is a projection matrix (see Property 1). Notice how, with the exception of the constraint $x = \Pi x$ and the multiplicative term, the final optimization problem is identical to the one for λ_k, given in (8). As such, the former's solution must be strictly larger (since it is a more constrained problem) and $\widetilde{\lambda}_k \geq \frac{\lambda_k}{\lambda_n(PP^\top)}$.

Analogously, one obtains the lower inequality $\widetilde{\lambda}_{k-(N-n)} \leq \frac{\lambda_k}{\lambda_1(PP^\top)}$ by applying the same argument on matrices $-L$ and $-L_c$ and exploiting that the k-th largest eigenvalue of any matrix M is also the k-th smallest eigenvalue of $-M$.

A.3. Proof of Proposition 6

For notational convenience, I drop the level index supposing that $c = 1$ and thus P is an $n \times N$ coarsening matrix. As we will see in the following, P has rank n and thus to prove that $P^+ = P^\top D^{-2}$, it is sufficient to show that matrix $\Pi = P^\top D^{-2} P$ is a projection matrix of rank n (and thus equal to P^+P). Matrix $\tilde{P} = D^{-1}P$ has the same sparsity structure as P and is thus also a coarsening matrix. W.l.o.g. let the rows of P be sorted based on their support, such that for any two rows $r < r'$ and $P(r,i), P(r',i') \neq 0$ we necessarily have $i < i'$. Furthermore, denote by p_r the vector containing all non-zero entries of $P(r,:)$ such that $\|p_r\|_2 = \|P(r,:)\|_2 = D(r,r)$. Due to the disjoint support of the rows of P and under this particular sorting, matrix Π is block-diagonal. Moreover, each block B_r in its diagonal is a rank 1 projection matrix as $B_r^2 = B_r B_r = \left(p_r D(r,r)^{-2} p_r^\top\right)\left(p_r D(r,r)^{-2} p_r^\top\right) = p_r D(r,r)^{-2} p_r^\top \frac{p_r^\top p_r}{\|p_r\|_2^2} = B_r$. We have thus arrived to the relation $\Pi^2 = \Pi$, which constitutes a necessary and sufficient condition for Π to be a projection matrix. The block-diagonal structure of Π also implies that its rank (as well as that of P) is n.

A.4. Proof of Proposition 7

Let us first remark that, by Proposition 6, $A = P^{\mp}$ is also a coarsening matrix with the same sparsity structure as P.

Necessity. I start by considering the nullspace of $L_c = ALA^\top$ and aim to ensure that it is equal to the span of the constant vector 1, which is a necessity for all combinatorial Laplacian matrices. Since matrix A is full row-rank and L has rank $N-1$, the nullspace of L_c is one dimensional. Therefore, the nullspace is correct as long as $1^\top ALA^\top 1 = 0$, which

happens if either $A^\top 1 = \alpha 1$ for a constant α or $A^\top 1 = 0$. In both cases, $(A^\top 1)(r) = \alpha 1$ for every r. By the definition of A however, we know that its rows have disjoint support and, as such, vector $A^\top 1$ exactly contains the non-zero entries of A. In other words, for the nullspace of L_c to be properly formed, the non-zero entries of A should either all be equal to α (such that $A^\top 1 = \alpha 1$) or zero (in which case $A^\top 1 = 0$). The latter case can clearly be discarded as it would disconnect the graph. We have thus discovered that $ALA^\top$ has a properly formed nullspace if and only if the non-zero entries of A are equal, rendering the latter condition necessary.

Sufficiency. Considering that every Laplacian of M edges can be re-written as $L = S^\top S$, where S is the $M \times N$ incidence matrix, one may confirm that the condition is also sufficient by showing that, for every A with equal non-zero entries, the matrix $S_c = SA^\top$ is an incidence matrix of L_c such that $L_c = S_c^\top S_c$. W.l.o.g., suppose that $\alpha = 1$ ($\alpha^2 L$ is a valid Laplacian for all α). By construction, each row of S_c is $S_c(q,:)^\top = AS(q,:)^\top$. Name as e_{ij} the corresponding edge, such that $S(q,:)^\top = \delta_i - \delta_j$, where δ_i is a dirac centered at vertex v_i. It follows that $S_c(q,:)^\top = A\delta_i - A\delta_j$. Obviously, if none of v_i, v_j are contracted then $S_c(q,:)^\top = \delta_i - \delta_j$, which is a valid row. Moreover, by construction of A, if either of v_i, v_j is contracted (but not both) or if v_i, v_j are contracted into different vertices then both relations $A\delta_i = \delta_i$ and $A\delta_j = \delta_j$ hold and thus once more $S_c(q,:)^\top = \delta_i - \delta_j$ is a valid row. Lastly, if v_i, v_j are contracted into the same vertex then for some r it must be that $A(r,i) = A(r,j)$, whereas $A(r',i) = A(r',j) = 0$ for all $r' \neq r$ and thus $S_c(r,:)^\top = 0$, signifying that the edge is not present. Summarizing, in every case S_c is a valid incidence matrix, rendering the condition also sufficient.

A.5. Proof of Property 8

For any two disjoint subsets $\mathcal{S}_1, \mathcal{S}_2$ of $\mathcal{V}$ denote by $w(\mathcal{S}_1, \mathcal{S}_2) = \sum_{v_i \in \mathcal{S}_1} \sum_{v_j \in \mathcal{S}_j} w_{ij}$ the cut weight in G.

The claim is proven by induction on the number of levels. For the *base case*, set $\ell = 1$ and define $C = P_1^+$ such that $c_r = C(:,r)$ is the indicator vector of the contraction set $\mathcal{V}_0^{(r)}$. It is a consequence of the Laplacian form of L that, for any $v_r, v_q \in \mathcal{V}_1$ with $r \neq q$, we have

$$\begin{aligned} W_1(r,q) = -L_1(r,q) = -c_r^\top L c_q &= \sum_{v_i \neq v_j} w_{ij} c_r(i) c_q(j) + \sum_{v_i} d_i c_r(i) c_q(i) \\ &= \sum_{v_i \in \mathcal{V}_0^{(r)}, v_j \in \mathcal{V}_0^{(q)}} w_{ij} = w(\mathcal{S}_1^{(r)}, \mathcal{S}_1^{(q)}), \end{aligned}$$

where the penultimate step uses $c_r(i)c_q(i) = 0$ since contraction steps are disjoint, and the last step exploits the equivalence $\mathcal{V}_0^{(q)} = \mathcal{S}_1^{(q)}$. For the *inductive step*, consider level $\ell > 1$. Since $L_{\ell-1}$ is a Laplacian matrix, one may employ an identical argument as when $\ell = 1$ to find that the weight between vertices $v_r, v_q \in \mathcal{V}_\ell$ with $r \neq q$ is

$$W_\ell(r,q) = \sum_{v_i \in \mathcal{V}_{\ell-1}^{(r)}, v_j \in \mathcal{V}_{\ell-1}^{(q)}} W_{\ell-1}(i,j).$$

By the induction hypothesis however, it must be $W_{\ell-1}(i,j) = w(\mathcal{S}_{\ell-1}^{(i)}, \mathcal{S}_{\ell-1}^{(j)})$, implying

$$W_\ell(r,q) = \sum_{v_i \in \mathcal{V}_{\ell-1}^{(r)}, v_j \in \mathcal{V}_{\ell-1}^{(q)}} w(\mathcal{S}_{\ell-1}^{(i)}, \mathcal{S}_{\ell-1}^{(j)}) = w(\mathcal{S}_\ell^{(r)}, \mathcal{S}_\ell^{(q)}),$$

with the final equality being true due to the recursive definition of sets $\mathcal{S}_\ell^{(r)}$ and $\mathcal{S}_\ell^{(q)}$, as well as the following property of cuts: for any two sets (call them large sets) and any partition of each into an arbitrary number of subsets, the cut between the large sets is equal to the sum of all cuts between pairs of subsets belonging to different large sets. This completes the proof.

A.6. Proof of Theorem 13

The lower bound is given by Theorem 3. For the upper bound, I reason similarly to the proof of the latter to find:

$$\widetilde{\lambda}_k = \min_{\dim(\mathbf{U}_c)=k} \max_{x_c \in \mathbf{U}_c} \left\{ \frac{x_c^\top L_c x_c}{x_c^\top x_c} \,|\, x_c \neq 0 \right\} \leq \gamma_2 \min_{\dim(\mathbf{U})=k, \mathbf{U} \subseteq \mathrm{im}(\Pi)} \max_{x \in \mathbf{U}} \left\{ \frac{x^\top \Pi L \Pi x}{x^\top \Pi x} | \, x \neq 0 \right\}.$$

Above, the inequality is due to Lemma 19 with $\gamma_2 = 1/\lambda_1(PP^\top)$. Thus, for any matrix V the following inequality holds

$$\widetilde{\lambda}_k \leq \gamma_2 \max_{x \in \mathrm{span}(V)} \left\{ \frac{x^\top \Pi L \Pi x}{x^\top \Pi x} \,|\, x \neq 0 \right\},$$

as long as the image of V is of dimension k and does not intersect the nullspace of Π. Write U_k to denote the $N \times k$ matrix with the k first eigenvectors of L, whose image is of dimension k as needed. Assume for now that the nullspace requirement is also met:

$$\widetilde{\lambda}_k \leq \gamma_2 \max_{x \in \mathrm{span}(U_k)} \left\{ \frac{x^\top \Pi L \Pi x}{x^\top \Pi x} \,|\, x \neq 0 \right\} = \gamma_2 \max_{x \in \mathrm{span}(U_k)} \left\{ \frac{\|S\Pi x\|_2^2}{\|\Pi x\|_2^2} \,|\, x \neq 0 \right\}.$$

It will be convenient to manipulate the square-root of this quantity:

$$\sqrt{\frac{\widetilde{\lambda}_k}{\gamma_2}} \leq \max_{a \in \mathbb{R}^k} \frac{\|S\Pi U_k a\|_2}{\|\Pi U_k a\|_2} = \frac{\|S\Pi U_k\|_2}{\|\Pi U_k\|_2} \leq \frac{\|SU_k\|_2 + \|S\Pi^\perp U_k\|_2}{\|\Pi U_k\|_2} = \frac{\sqrt{\lambda_k} + \|S\Pi^\perp U_k\|_2}{\|\Pi U_k\|_2}, \quad (9)$$

with S defined such that $L = S^\top S$. The norm in the numerator is upper bounded by

$$\begin{aligned} \|S\Pi^\perp U_k\|_2 = \|S\Pi^\perp U_k \Lambda_k^{-1/2} \Lambda_k^{1/2}\|_2 &\leq \|S\Pi^\perp U_k \Lambda_k^{+1/2}\|_2 \|\Lambda_k^{1/2}\|_2 \\ &= \sqrt{\lambda_k}\, \|S\Pi^\perp U_k \Lambda_k^{+1/2}\|_2 = \sqrt{\lambda_k}\, \epsilon_k. \end{aligned}$$

If the last step is not immediately obvious, one can be convinced by first exploiting the unitary-invariance of the spectral norm to write $\|S\Pi^\perp U_k \Lambda_k^{+1/2}\|_2 = \|S\Pi^\perp U_k U_k^\top L^{+1/2}\|_2$, and then confirming in the proof of Proposition 17 that the latter quantity is exactly ϵ_k when $V = U_k$.

The preceding analysis assumed that the image of U_k and the nullspace of Π did not intersect. Since $\Pi^\perp = I - \Pi$ is a complement projection matrix , the previous holds when $\|\Pi^\perp U_k\|_2^2 < 1$. Since $\|\Pi^\perp u_1\| = 0$, one may w.l.o.g. exclude the first eigenvector u_1 from the space of interest. For the remainder of $\mathrm{im}(U_k)$ the following holds:

$$\|\Pi^\perp U_k\|_2^2 = \max_{x \in \mathbf{U}_k \text{ and } x \perp u_1} \frac{\|\Pi^\perp x\|_2^2}{\|x\|_2^2} \leq \frac{1}{\lambda_2} \max_{x \in \mathbf{U}_k \text{ and } x \perp u_1} \frac{\|\Pi^\perp x\|_L^2}{\|x\|_2^2} = \epsilon_k^2 \frac{\lambda_k}{\lambda_2}.$$

Therefore, when $\epsilon_k^2 < \frac{\lambda_2}{\lambda_k}$, the nullspace condition is met. The proof is then concluded by substituting the bound $\|\Pi U_k\|_2^2 = 1 - \|\Pi^\perp U_k\|_2^2 \geq 1 - \epsilon_k^2 \frac{\lambda_k}{\lambda_2}$ in the denominator of (9).

A.7. Proof of Theorem 14

Li (1994) showed that we can express the $\sin\Theta$ as a sum of squared inner products:

$$\left\|\sin\Theta\big(U_k, P^\top \widetilde{U}_k\big)\right\|_F^2 = \left\|\widetilde{U}_{k^\perp}^\top P U_k\right\|_F^2 = \sum_{i \leq k} \sum_{j > k} (\widetilde{u}_j^\top P u_i)^2 \tag{10}$$

If L_c and L are $(\mathbf{U}_i, \epsilon_i)$-similar it follows from Corollary 12 that

$$u_i^\top P^\top L_c P u_i \leq (1 + \epsilon_i)^2 \lambda_i.$$

Summing these inequalities for all $i \leq k$ amounts to

$$\begin{aligned}
\sum_{i \leq k} (1 + \epsilon_i)^2 \lambda_i &\geq \sum_{i \leq k} \sum_{j=1}^{n} \widetilde{\lambda}_j (\widetilde{u}_j^\top P u_i)^2 \\
&\geq \gamma_1 \sum_{i \leq k} \sum_{j=1}^{n} \lambda_j (\widetilde{u}_j^\top P u_i)^2 \\
&= \gamma_1 \sum_{j \leq k} \lambda \sum_{i \leq k} (\widetilde{u}_j^\top P u_i)^2 + \gamma_1 \sum_{j > k} \lambda_j \sum_{i \leq k} (\widetilde{u}_j^\top P u_i)^2.
\end{aligned} \tag{11}$$

where, following Theorem 3, I set $\gamma_1 = 1/\lambda_n(PP^\top)$ such that $\tilde{\lambda}_i \geq \gamma_1 \lambda_i$. Perform the following manipulation:

$$\begin{aligned}
\sum_{j \leq k} \lambda_j \sum_{i \leq k} (\widetilde{u}_j^\top P u_i)^2 \geq \sum_{j \leq k} \lambda_j \sum_{i \leq k} (\widetilde{u}_j^\top P u_i)^2 &= \sum_{j \leq k} \lambda_j \left(1 - \sum_{i > k} (\widetilde{u}_j^\top P u_i)^2\right) \\
&\geq \sum_{j \leq k} \lambda_j - \lambda_k \sum_{i \leq k} \left(\|\Pi^\perp u_i\|_2^2 + \sum_{j \geq k} (\widetilde{u}_j^\top P u_i)^2\right),
\end{aligned}$$

which together with (10) and (11) yields

$$\left\|\sin\Theta\big(U_k, P^\top \widetilde{U}_k\big)\right\|_F^2 \leq \sum_{i \leq k} \frac{(1 + \epsilon_i)^2 \lambda_i / \gamma_1 - \lambda_i + \lambda_k \|\Pi^\perp u_i\|_2^2}{\lambda_{k+1} - \lambda_k} \tag{12}$$

To proceed, I note the following useful inequality:

Lemma 20 *If L and L_c are $(\mathbf{U}_i, \epsilon_i)$-similar, then $\|\Pi^\perp u_i\|_2^2 \leq \epsilon_i$.*

Proof For every u_i one sees that

$$\begin{aligned}
u_i^\top P^\top L_c P u_i = u_i^\top \Pi L \Pi u_i &= u_i^\top (I - \Pi^\perp) L (I - \Pi^\perp) u_i \\
&= u_i^\top L u_i - 2 u_i^\top L \Pi^\perp u_i + u_i^\top \Pi^\perp L \Pi^\perp u_i \\
&= \lambda_i - 2\lambda_i u_i^\top \Pi^\perp u_i + u_i^\top \Pi^\perp L \Pi^\perp u_i
\end{aligned}$$

meaning that

$$\|\Pi^\perp u_i\|_2^2 = \frac{1}{2}\left(1 + \frac{u_i^\top \Pi^\perp L \Pi^\perp u_i}{\lambda_i} - \frac{u_i^\top P^\top L_c P u_i}{\lambda_i}\right) \leq \frac{1}{2}\left(1 + \epsilon_i^2 - (1 - \epsilon_i)^2\right) = \epsilon_i.$$

The last inequality is because, by restricted spectral approximation, we have $u_i^\top \Pi^\perp L \Pi^\perp u_i = \|\Pi^\perp u_i\|_L^2 \leq \epsilon_i^2 \|u_i\|_L^2 = \epsilon_i^2 \lambda_i$ and from Corollary 12:

$$u_i^\top P^\top L_c P u_i = \|P u_i\|_{L_c}^2 \geq (1 - \epsilon_i)^2 \|u_i\|_L^2 = (1 - \epsilon_i)^2 \lambda_i.$$

■

As a consequence, it follows that

$$\left\|\sin\Theta\big(U_k, P^\top \widetilde{U}_k\big)\right\|_F^2 \leq \sum_{i \leq k} \frac{(1 + \epsilon_i)^2 \lambda_i / \gamma_1 - \lambda_i + \lambda_k \epsilon_i}{\lambda_{k+1} - \lambda_k},$$

which after manipulation gives the desired inequality.

A.8. Proof of Theorem 15

The lower bound is a direct consequence of consistent coarsening and holds independently of restricted spectral approximation: for any set $\mathcal{S}_c \subset \mathcal{V}_c$ define $\mathcal{S} \subset \mathcal{V}$ such that $v_i \in \mathcal{S}$ if and only if $\varphi_c \circ \varphi_{c-1} \circ \cdots \varphi_1(v_i) \in \mathcal{S}_c$. Clearly, $w(\mathcal{S}) \geq w_c(\mathcal{S}_c)$, where the subscript c implies that the latter volume is w.r.t. G_c. In addition, by the definition of Laplacian consistent coarsening and since every contraction set belongs either in $\mathcal{S}$ or $\bar{\mathcal{S}}$ (but not in both), it follows that $w(\mathcal{S}, \bar{\mathcal{S}}) = w_c(\mathcal{S}_c, \bar{\mathcal{S}}_c)$. In other words, for every $\mathcal{S}_c$ there exists a set $\mathcal{S}$ such that

$$\phi(\mathcal{S}) = \frac{w(\mathcal{S}, \bar{\mathcal{S}})}{\min\{w(\mathcal{S}), w(\bar{\mathcal{S}})\}} \leq \frac{w_c(\mathcal{S}_c, \bar{\mathcal{S}}_c)}{\min\{w_c(\mathcal{S}_c), w_c(\bar{\mathcal{S}}_c)\}} = \phi_c(\mathcal{S}_c),$$

implying also that the k-conductance of G and G_c are related by $\phi_k(G) \leq \phi_k(G_c)$.

For the upper bound, I exploit the following multi-way Cheeger inequality:

Theorem 21 (Restatement of Theorem 1.2 by Lee et al. (2014)) *For every graph G and every $k \in \mathbb{N}$, we have*

$$\frac{\mu_k}{2} \leq \phi_k(G) = O(\sqrt{\mu_{2k}\, \xi_k(G)}),$$

with $\xi_k(G) \leq \log k$. If G is planar then $\xi_k(G) \leq 1$ and if G excludes K_h as a minor (but is not planar) then $\xi_k(G) \leq h^4$.

Further, in the standard Cheeger inequality (Alon and Milman, 1985; Alon, 1986) $k = 2$ and the upper bound is given by $\sqrt{2\mu_2}$. Note that the eigenvalues mentioned here are those of the normalized Laplacian matrix $L^n = D^{-1/2}LD^{-1/2}$. To this end, suppose that V_{2k} contains the first $2k$ eigenvectors of L^n and fix $\mathbf{R} = \text{span}(D^{-1/2}V_{2k})$. Perform consistent coarsening w.r.t. to the combinatorial Laplacian L (not L^n). Then, by definition, if L_c and L are $(\epsilon_{2k}, \mathbf{R})$-similar then for every $x \in \mathbf{R}$ one gets

$$\|\Pi^{\perp}x\|_L \leq \epsilon_{2k}\|x\|_L.$$

The substitution $y = D^{1/2}x$, such that $y \in \text{span}(V_{2k})$, allows us to transform the semi-norms above into semi-norms concerning L^n as follows:

$$\|x\|_L^2 = x^{\top}Lx = y^{\top}D^{-1/2}LD^{-1/2}y = \|y\|_{L^n}^2$$

and

$$\|\Pi^{\perp}x\|_L = \|D^{-1/2}D^{1/2}\Pi^{\perp}D^{-1/2}D^{1/2}x\|_L = \|D^{1/2}\Pi^{\perp}D^{-1/2}y\|_{L^n} = \|(I - \Pi^n)y\|_{L^n}.$$

Above, $\Pi^n = D^{1/2}\Pi D^{-1/2}$ is the projection matrix (the set of projection matrices is closed under similarity transformations) corresponding to the coarsening matrix $P^n = PD^{-1/2}$, and now $y_c = \Pi^n y$. It follows that, for every $y \in \mathbf{V}_{2k} = \text{span}(V_{2k})$, we have

$$\|y - \Pi^n y\|_{L^n} \leq \epsilon_{2k}\|y\|_{L^n}$$

and thus L_c^n and L^n are $(\mathbf{V}_{2k}, \epsilon_{2k})$-similar.

Combining the multi-way Cheeger inequality with Theorem 13 for L_c^n and L^n one obtains

$$\begin{aligned}
\phi_k^2(G_c) &= O\left(\tilde{\mu}_{2k}\,\xi_k(G)\right) \\
&= O\left(\gamma_2(1+\epsilon_{2k})^2\frac{\mu_2}{\mu_2 - \epsilon_{2k}\mu_{2k}}\mu_{2k}\,\xi_k(G)\right) \\
&= O\left(\frac{\gamma_2\,(1+\epsilon_{2k})^2\xi_k(G)}{1-\epsilon_{2k}^2(\mu_{2k}/\mu_2)}\phi_k(G)\right),
\end{aligned}$$

where the eigenvalues above are those of L^n and the preceeding holds whenever $\epsilon_{2k}^2 < \mu_2/\mu_{2k}$. Further, when $k = 2$ the upper bound simplifies to $\phi_2^2(G_c) \leq \frac{4\,\gamma_2\,(1+\epsilon_2)^2}{1-\epsilon_2^2}\phi_2(G)$.

A.9. Proof of Proposition 17

The following analysis is slightly more general than what is claimed in the statement of Proposition 17: it holds for arbitrary PSD L and L_c (i.e., not necessarily Laplacian matrices) as long as the image $\text{im}(\Pi)$ of the projection matrix Π encloses the nullspace of L. The former trivially holds for Laplacian consistent coarsening, as, by design, one has $\Pi 1 = 1$ (see Section 2.4).

Let $V \in \mathbb{R}^{N\times k}$ be a basis of $\mathbf{R}$. I start by proving that, for any integer $k \leq n$ and for all $x \in \text{span}(V)$ the inequality

$$\|x - \Pi x\|_L \leq \epsilon\,\|x\|_L$$

holds for all $\epsilon \geq \|\Pi^{\perp}B_0\|_L$, where $B_0 = VV^{\top}L^{+1/2}$. I remind the reader that $\|x\|_L = \|Sx\|_2 = \|L^{1/2}x\|_2$ and $\Pi^{\perp} = I - P^{+}P$. Furthermore, since $\text{im}(\Pi)$ necessarily encloses the nullspace $\mathbf{N}$ of L, w.l.o.g., one may assume that $\forall x \in \mathbf{R}$ the action of matrix L is invertible. To see why, note that if $x \in \mathbf{N}$ then $\|x\|_L = 0$ and $\|x - \Pi x\|_L = 0$, meaning that the inequality above is trivially satisfied. I then derive

$$\begin{aligned}
\max_{x\in\mathbf{R}}\frac{\|x-\Pi x\|_L}{\|x\|_L} &= \max_{x\in\mathbf{R}}\frac{\|S\Pi^{\perp}x\|_2}{\|L^{1/2}x\|_2} \\
&= \max_{x\in\mathbf{R}}\frac{\|S\Pi^{\perp}VV^{\top}x\|_2}{\|L^{1/2}x\|_2} \qquad (13)\\
&= \max_{x\in\text{im}(LV)}\frac{\|S\Pi^{\perp}VV^{\top}L^{+1/2}x\|_2}{\|x\|_2} \qquad (14)\\
&\leq \|S\Pi^{\perp}VV^{\top}L^{+1/2}\|_2 = \|\Pi^{\perp}B_0\|_L,
\end{aligned}$$

where equality (13) holds because $VV^{\top}$ is a projection onto $\mathbf{R}$, whereas equality (14) is true since L is invertible within $\mathbf{R}$.

One should also note that, for the specific case where V is an eigenspace of L, $\text{im}(LV) = \mathbf{R}$ and as such $\epsilon = \|\Pi^{\perp}x\|_L/\|x\|_L = \|S\Pi^{\perp}VV^{\top}L^{+1/2}\|_2$ (once more w.l.o.g. the nullspace of L can be ignored).

In addition, as the following technical lemma claims, in a multi-level scheme, any $\|\Pi^{\perp}x\|_L$ can be broken down into the contributions of each level:

Lemma 22 *Define projection matrices $\Pi_\ell = P_\ell^{+}P_\ell$ and $\Pi_\ell^{\perp} = I - \Pi_\ell$. If*

$$\|\Pi_\ell^{\perp}x_{\ell-1}\|_{L_{\ell-1}} \leq \sigma_\ell\,\|x_{\ell-1}\|_{L_{\ell-1}} \quad \textit{at each level } \ell \leq c,$$

then the multi-level error is bounded by

$$\|\Pi^{\perp}x\|_L \leq \left(\sum_{\ell=1}^{c}\sigma_\ell\prod_{q=1}^{\ell-1}(1+\sigma_q)\right)\|x\|_L = \left(\prod_{\ell=1}^{c}(1+\sigma_\ell) - 1\right)\|x\|_L.$$

Proof Recursively apply the following inequality

$$\begin{aligned}
\left\|S_{\ell-1}\Pi_\ell^{\perp}x_{\ell-1}\right\|_2 &\leq \sigma_\ell\,\|S_{\ell-1}x_{\ell-1}\|_2 \\
&= \sigma_\ell\,\|S_{\ell-2}\Pi_{\ell-1}x_{\ell-2}\|_2 \\
&\leq \sigma_\ell\left(\|S_{\ell-2}x_{\ell-2}\|_2 + \left\|S_{\ell-2}\Pi_{\ell-1}^{\perp}x_{\ell-2}\right\|_2\right) \\
&\leq \sigma_\ell\left(\|S_{\ell-2}x_{\ell-2}\|_2 + \sigma_{\ell-1}\,\|S_{\ell-2}x_{\ell-2}\|_2\right) = \sigma_\ell\,(1+\sigma_{\ell-1})\,\|S_{\ell-2}x_{\ell-2}\|_2
\end{aligned}$$

to deduce that

$$\left\|S_{\ell-1}\Pi_\ell^{\perp}x_{\ell-1}\right\|_2 \leq \sigma_\ell\prod_{q=1}^{\ell-1}(1+\sigma_q)\|S_0x_0\|_2 = \sigma_\ell\prod_{q=1}^{\ell-1}(1+\sigma_q)\|x\|_L.$$

The end-to-end error $\|S\Pi^\perp x\|_2$ is controlled with a simple telescopic series argument.

$$\begin{aligned}\|\Pi^\perp x\|_L = \|S_0\Pi^\perp x_0\|_2 &= \|S_0x_0 - S_cx_c\|_2\\ &\le \|S_0x_0 - S_1x_1\|_2 + \|S_1x_1 - S_2x_2\|_2 + \ldots + \|S_{c-1}x_{c-1} - S_cx_c\|_2\\ &= \|S_0\Pi_1^\perp x_1\|_2 + \|S_1\Pi_1^\perp x_1\|_2 + \ldots + \|S_{c-1}\Pi_c^\perp x_{c-1}\|_2\end{aligned}$$

Together, the above two results imply the desired bound. ■

Therefore, to guarantee that in a multi-level scheme

$$\|\Pi^\perp B_0\|_L = \max_{b\in\mathbb{R}^N} \frac{\|S\Pi^\perp B_0\, b\|_2}{\|b\|_2} \le \epsilon,$$

one needs to make sure that, for each level $\ell = 1, \ldots, c$, the following holds:

$$\frac{\|S_{\ell-1}\Pi_\ell^\perp x_{\ell-1}\|_2}{\|S_{\ell-1}x_{\ell-1}\|_2} \le \sigma_\ell, \quad \text{for all} \quad x_{\ell-1} = P_{\ell-1}\cdots P_1B_0\, b$$

By the same argument used for the multi-level error, when $\ell = 1$, we have that $\sigma_1 = \|\Pi_1^\perp B_0\|_{L_0}$. For all other ℓ, set $B_{\ell-1} = P_{\ell-1}\cdots P_1B_0$ and further let $(B_{\ell-1}^\top L_{\ell-1}B_{\ell-1})^{+1/2}$ be the pseudo-inverse of the matrix square-root of the $N\times N$ matrix $B_{\ell-1}^\top L_{\ell-1}B_{\ell-1}$. By the substitution $b = S_{\ell-1}B_{\ell-1}a$, the above can be rewritten as

$$\max_{b\in\mathbb{R}^N} \frac{\|S_{\ell-1}\Pi_\ell^\perp B_{\ell-1}b\|_2}{\|S_{\ell-1}B_{\ell-1}b\|_2} = \max_{b\in\mathbb{R}^N} \frac{\|S_{\ell-1}\Pi_\ell^\perp B_{\ell-1}(B_{\ell-1}^\top L_{\ell-1}B_{\ell-1})^{+1/2}b\|_2}{\|b\|_2}.$$

For $\ell > 1$, therefore $\sigma_\ell = \|\Pi_\ell^\perp A_{\ell-1}\|_{L_{\ell-1}}$ with $A_{\ell-1} = B_{\ell-1}(B_{\ell-1}^\top L_{\ell-1}B_{\ell-1})^{+1/2}$.

A.10. Proof of Proposition 18

For notational simplicity in the context of this proof I drop level indices and assume that only a single coarsening level is used—this is without loss of generality, as an identical argument holds for every level of the scheme.

Consider any x and set $y = \Pi^\perp x$. Furthermore, define for each contraction set the (i) *internal* edge set $\mathcal{E}^{(r)} = \{e_{ij} | v_i, v_j \in \mathcal{V}^{(r)} \text{ and } e_{ij} \in \mathcal{E}_{\ell-1}\}$, and (ii) the *boundary* edge set $\partial\mathcal{E}^{(r)}$, such that $e_{ij} \in \partial\mathcal{E}^{(r)}$ if and only if $v_i \in \mathcal{V}^{(r)}$ and $v_j \notin \mathcal{V}^{(r)}$. It is true that

$$\begin{aligned}\|\Pi^\perp x\|_L^2 &= \sum_{e_{ij}\in\mathcal{E}} w_{ij}(y(i) - y(j))^2\\ &= \sum_{r=1}^{n}\Bigg(\underbrace{\sum_{e_{ij}\in\mathcal{E}^{(r)}} w_{ij}(y(i) - y(j))^2}_{a_r} + \underbrace{\frac{1}{2}\sum_{e_{ij}\in\partial\mathcal{E}^{(r)}} w_{ij}(y(i) - y(j))^2}_{b_r}\Bigg).\end{aligned}$$

In the following, I will express a_r and b_r as a function of the vector $y_r = \Pi^\perp_{\mathcal{V}^{(r)}}x$. Term a_r is luckily independent of any other contraction set:

$$a_r = \sum_{e_{ij}\in\mathcal{E}^{(r)}} w_{ij}(y(i) - y(j))^2 = \sum_{e_{ij}\in\mathcal{E}^{(r)}} w_{ij}(y_r(i) - y_r(j))^2.$$

On the other hand, b_r is smaller than

$$b_r = \sum_{e_{ij}\in\partial\mathcal{E}^{(r)}} w_{ij}(y(i) - y(j))^2 \le 2\sum_{e_{ij}\in\partial\mathcal{E}^{(r)}} w_{ij}(y(i) - 0)^2 + 2\sum_{e_{ij}\in\partial\mathcal{E}^{(r)}} w_{ij}(0 - y(j))^2.$$

Distributing the second quantities, respectively, amongst the contraction sets that include said vertices, one gets

$$\begin{aligned}\|\Pi^\perp x\|_L^2 &\le \sum_{r=1}^{n}\Bigg(\sum_{e_{ij}\in\mathcal{E}^{(r)}} w_{ij}(y_r(i) - y_r(j))^2 + 2\sum_{e_{ij}\in\partial\mathcal{E}^{(r)}} w_{ij}(y(i) - 0)^2\Bigg)\\ &= \sum_{r=1}^{n}\Bigg(\sum_{e_{ij}\in\mathcal{E}^{(r)}} w_{ij}(y_r(i) - y_r(j))^2 + \sum_{e_{ij}\in\partial\mathcal{E}^{(r)}} (2w_{ij})(y_r(i) - y_r(j))^2\Bigg)\\ &= \sum_{r=1}^{n}\|y_r\|^2_{L_{\mathcal{V}^{(r)}}} = \sum_{\mathcal{C}\in\mathcal{P}}\|y_r\|^2_{L_\mathcal{C}}.\end{aligned}$$

The second step above used the fact that $[\Pi^\perp](i) = 0$ for all $v_i \notin \mathcal{C}$. A decoupled bound can then be obtained as follows:

$$\|\Pi^\perp A\|_L^2 = \max_{a\in\mathbb{R}^{k-1}} \frac{\|S\Pi^\perp A\, a\|_2^2}{\|a\|_2^2} \le \sum_{\mathcal{C}\in\mathcal{P}} \max_{a\in\mathbb{R}^{k-1}} \frac{\|\Pi_\mathcal{C}^\perp\, A\, a\|^2_{L_\mathcal{C}}}{\|a\|_2^2} = \sum_{\mathcal{C}\in\mathcal{P}} \|\Pi_\mathcal{C}^\perp\, A\|^2_{L_\mathcal{C}}$$

The final inequality is derived by taking the square-root of the last equation.

Appendix B. Complexity Analysis

The computational complexity of Algorithm 1 depends on the number of nodes N and edges M of G, the number of levels c, the subspace size k, as well as on how the families of candidate sets are formed. To derive worst-case bounds, I denote by $\Phi_\ell = \sum_{\mathcal{C}\in\mathcal{F}_\ell}|\mathcal{C}|$ the number of vertices in all candidate sets and by $\delta = \max_{\ell,\mathcal{C}\in\mathcal{F}_\ell}|\mathcal{C}|$ the cardinality of the maximum candidate set over all levels. Furthermore, I suppose that the per-level reduction ratio r_ℓ is a constant.

I start with some basic observations:

- *Computing $A_0, \ldots, A_{c-1}$ is possible in $\tilde{O}(ckM + k^2N + ck^3)$ operations when $V = U_k$.* Each $A_{\ell-1}$ is computed once for each level. For $\ell = 1$, one needs to approximate the first k eigenpairs of L, which can be achieved in $\tilde{O}(kM)$ operations using inverse iteration as described by Vishnoi et al. (2013). For consecutive levels, forming matrix

$B_{\ell-1}^\top L_{\ell-1} B_{\ell-1}$ takes $O(M_{\ell-1}k + N_{\ell-1}k^2)$ operations, whereas computing the pseudo-square-root $(B_{\ell-1}^\top L_{\ell-1} B_{\ell-1})^{+1/2}$ is possible in $O(k^3)$ operations. Summed up, the costs for all levels amount to $O(k\sum_{\ell=1}^{c} M_{\ell-1} + k^2 \sum_{\ell=2}^{c} N_{\ell-1} + ck^3) = O(ckM + k^2N + ck^3)$, where I used the observation that $\sum_{\ell=2}^{c} N_{\ell-1} = O(N)$.

- *At each level, the cost function is evaluated at most Φ_ℓ times.* One starts by computing the cost of each candidate set in $\mathcal{F}_\ell$. Moreover, every $\mathcal{C}$ added to $\mathcal{P}_\ell$ causes the pruning of at most $\sum_{v_i \in \mathcal{C}} (\phi_i - 1)$ other sets, where ϕ_i is the number of candidate sets that include v_i. Since $\mathcal{P}_\ell$ is a partitioning of $\mathcal{V}_{\ell-1}$, at most $\sum_{\mathcal{C}\in\mathcal{P}_\ell} \sum_{v_i\in\mathcal{C}} (\phi_i - 1) \leq \sum_{v_i \in \mathcal{V}_{\ell-1}} \phi_i - |\mathcal{F}_\ell| = \Phi_\ell - |\mathcal{F}_\ell|$ cost re-evaluation are needed.

- *Given $A_{\ell-1}$, each call of $\text{cost}_\ell(\mathcal{C})$ requires $O(\min\{k^2\delta + k\delta^2,\ k\delta^2 + \delta^3\})$ operations.* The involved matrices themselves can be easily formed since, excluding all-zero rows and columns, both $L_\mathcal{C}$ and $\Pi_\mathcal{C}^\perp$ are $|\mathcal{C}| \times |\mathcal{C}|$ matrices and one can safely restrict $A_{\ell-1}$ to be of size $|\mathcal{C}| \times k$ by deleting all rows that would have been multiplied by zero. Now, by definition, the incidence matrix $S_\mathcal{C}$ of $L_\mathcal{C}$ has at most δ columns and 2δ rows (since one can bundle all boundary weights of a vertex in $\mathcal{C}$ in a single row). Depending on the relative size of k and δ the computation can be performed in two ways:
  - Either one forms the $k \times k$ matrix $A_{\ell-1}^\top \Pi_\mathcal{C}^\perp L_\mathcal{C} \Pi_\mathcal{C}^\perp A_{\ell-1}$ and approximate its spectral norm paying a total of $O(k^2\delta + k\delta^2)$.
  - Otherwise, the $2d \times 2d$ matrix $S_\mathcal{C} \Pi_\mathcal{C}^\perp A_{\ell-1} A_{\ell-1}^\top \Pi_\mathcal{C}^\perp S_\mathcal{C}^\top$ is formed and its norm is computed at a combined cost of $O(\delta^2 k + \delta^3)$.

- *Maintaining $\mathcal{F}_\ell$ sorted incurs $O(\Phi_\ell \log|\mathcal{F}_\ell|)$ cost.* Sorting $\mathcal{F}_\ell$ during initialization entails $O(|\mathcal{F}_\ell| \log|\mathcal{F}_\ell|)$ operations. Inserting each $\mathcal{C}'$ into $\mathcal{F}_\ell$ (see step 12) can be done in $O(\log|\mathcal{F}_\ell|)$ and, moreover, by the same argument used to bound the number of cost evaluations, at most $\Phi_\ell - |\mathcal{F}_\ell|$ such insertions can happen.

- *Other operations carry negligible cost.* In particular, by implementing marked as a binary array, checking if a vertex is marked or not can be done in constant time.

Overall, using Algorithm 2 and for $\mathbf{R} = \mathbf{U}_k$ one can coarsen a graph in $\tilde{O}(ckM + k^2N + ck^3 + \sum_{\ell=1}^{c} \Phi_\ell(\min\{k^2\delta + k\delta^2,\ k\delta^2 + \delta^3\} + \log|\mathcal{F}_\ell|))$ time, where the asymptotic notation hides poly-logarithmic factors.

References

Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

Noga Alon and Vitali D Milman. λ1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W Pellegrini, Ralf S Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

Joshua Batson, Daniel A. Spielman, Nikhil Srivastava, and Shang-Hua Teng. Spectral sparsification of graphs: Theory and algorithms. *Commun. ACM*, 56(8):87–94, 2013. ISSN 0001-0782.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Christos Boutsidis, Prabhanjan Kambadur, and Alex Gittens. Spectral clustering via the power method-provably. In *International Conference on Machine Learning*, pages 40–48, 2015.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. ISSN 1053-5888. doi: 10.1109/MSP.2017.2693418.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

Daniele Calandriello, Ioannis Koutis, Alessandro Lazaric, and Michal Valko. Improved large-scale graph learning through ridge spectral sparsification. In *International Conference on Machine Learning*, 2018.

Guantao Chen, George Davis, Frank Hall, Zhongshan Li, Kinnari Patel, and Michael Stewart. An interlacing result on normalized Laplacians. *SIAM Journal on Discrete Mathematics*, 18(2):353–361, 2004.

Jie Chen and Ilya Safro. Algebraic distance on graphs. *SIAM Journal on Scientific Computing*, 33(6):3468–3490, 2011.

Fan RK Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

Charles Colley, Junyuan Lin, Xiaozhe Hu, and Shuchin Aeron. Algebraic multigrid for least squares problems on graphs with applications to hodgerank. In *Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International*, pages 627–636. IEEE, 2017.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11), 2007.

Florian Dörfler and Francesco Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems*, 60(1):150–163, 2013.

Yuguang Fang, Kenneth A Loparo, and Xiangbo Feng. Inequalities for the trace of matrix product. *IEEE Transactions on Automatic Control*, 39(12):2489–2490, 1994.

Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1):23–38, 1986.

Shivam Gandhi. Improvement of the cascadic multigrid algorithm with a Gauss-Seidel smoother to efficiently compute the Fiedler vector of a graph Laplacian. *arXiv preprint arXiv:1602.04386*, 2016.

Matan Gavish, Boaz Nadler, and Ronald R Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *ICML*, pages 367–374, 2010.

David Gleich. MatlabBGL. a matlab graph library. *Institute for Computational and Mathematical Engineering, Stanford University*, 2008.

Francesco Grassi, Andreas Loukas, Nathanael Perraudin, and Benjamin Ricaud. A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs. *IEEE Transactions on Signal Processing*, 66(3):817–829, 2017.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Bruce Hendrickson and Robert W Leland. A multi-level algorithm for partitioning graphs. *SC*, 95(28):1–14, 1995.

Anil N Hirani, Kaushik Kalyanaraman, and Seth Watts. Graph Laplacians and least squares on graphs. In *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*, pages 812–821. IEEE, 2015.

Yifan Hu. Efficient, high-quality force-directed graph drawing. *Mathematica Journal*, 10 (1):37–71, 2005.

Elvin Isufi, Andreas Loukas, Nathanael Perraudin, and Geert Leus. Forecasting time series with varma recursions on graphs. *arXiv preprint arXiv:1810.08581*, 2018.

Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41, 2001.

David R Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

David Harel Yehuda Koren. A fast multi-scale method for drawing large graphs. *Journal of graph algorithms and applications*, 6(3):179–202, 2002.

I. Koutis, G. L. Miller, and R. Peng. Approaching optimality for solving sdd linear systems. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 235–244, Oct 2010.

Ioannis Koutis, Gary L Miller, and David Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. *Computer Vision and Image Understanding*, 115(12):1638–1646, 2011.

Dan Kushnir, Meirav Galun, and Achi Brandt. Fast multiscale clustering and manifold identification. *Pattern Recognition*, 39(10):1876–1891, 2006.

Stephane Lafon and Ann B Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1393–1403, 2006.

James R Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM (JACM)*, 61(6):37, 2014.

Ren-Cang Li. Relative perturbation theory:(ii) eigenspace variations. Technical report, 1994.

Jiongqian Liang, Saket Gurukar, and Srinivasan Parthasarathy. Mile: A multi-level framework for scalable graph embedding. *arXiv preprint arXiv:1802.09612*, 2018.

Oren E Livne and Achi Brandt. Lean algebraic multigrid (LAMG): Fast graph Laplacian linear solver. *SIAM Journal on Scientific Computing*, 34(4):B499–B522, 2012.

Andreas Loukas and Nathanaël Perraudin. Stationary time-vertex signal processing. *arXiv preprint arXiv:1611.00255*, 2016.

Andreas Loukas and Pierre Vandergheynst. Spectrally approximating large graphs with smaller graphs. In *Interenational Conference on Machine Learning (ICML)*, 2018.

Lionel Martin, Andreas Loukas, and Pierre Vandergheynst. Fast approximate spectral clustering for dynamic networks. In *Interenational Conference on Machine Learning (ICML)*, 2018.

Ankur Moitra. *Vertex sparsification and universal rounding algorithms.* PhD thesis, Massachusetts Institute of Technology, 2011.

Ami Paz and Gregory Schwartzman. A $(2+\epsilon)$-approximation for maximum weight matching in the semi-streaming model. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2153–2161. SIAM, 2017.

David Peleg and Alejandro A Schäffer. Graph spanners. *Journal of graph theory*, 13(1): 99–116, 1989.

Robert Preis and Ralf Diekmann. Party-a software library for graph partitioning. *Advances in Computational Mechanics with Parallel and Distributed Processing*, pages 63–71, 1997.

Gilles Puy, Nicolas Tremblay, Rémi Gribonval, and Pierre Vandergheynst. Random sampling of bandlimited signals on graphs. *Applied and Computational Harmonic Analysis*, 44(2):446–475, 2018.

Dorit Ron, Ilya Safro, and Achi Brandt. Relaxation-based coarsening and multiscale graph organization. *Multiscale Modeling & Simulation*, 9(1):407–423, 2011.

Ilya Safro, Peter Sanders, and Christian Schulz. Advanced coarsening schemes for graph partitioning. *Journal of Experimental Algorithmics (JEA)*, 19:2–2, 2015.

Nauman Shahid, Nathanael Perraudin, Vassilis Kalofolias, Gilles Puy, and Pierre Vandergheynst. Fast robust PCA on graphs. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):740–756, 2016.

David I Shuman, Mohammad Javad Faraji, and Pierre Vandergheynst. A multiscale pyramid transform for graph signals. *IEEE Transactions on Signal Processing*, 64(8):2119–2134, 2016.

Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

Gilbert W Stewart. Matrix perturbation theory. 1990.

Nicolas Tremblay and Andreas Loukas. Approximating spectral clustering via sampling: a review. *CoRR*, abs/1901.10204, 2019.

Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318. ACM, 1994.

John C Urschel, Xiaozhe Hu, Jinchao Xu, and Ludmil T Zikatanov. A cascadic multigrid algorithm for computing the Fiedler vector of graph Laplacians. *arXiv preprint arXiv:1412.0565*, 2014.

Nisheeth K Vishnoi et al. Lx=b. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

Chris Walshaw. A multilevel algorithm for force-directed graph drawing. In *International Symposium on Graph Drawing*, pages 171–182. Springer, 2000.

Chris Walshaw. A multilevel algorithm for force-directed graph-drawing. *Journal of Graph Algorithms and Applications*, 7(3):253–285, 2006.

Lu Wang, Yanghua Xiao, Bin Shao, and Haixun Wang. How to partition a billion-node graph. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 568–579. IEEE, 2014.

Shusen Wang and Zhihua Zhang. Improving cur matrix decomposition and the Nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*, 14(1): 2729–2769, 2013.

Richard C Wilson and Ping Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9):2833–2841, 2008.

Yi Yu, Tengyao Wang, and Richard J Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2014.

Journal of Machine Learning Research 20 (2019) 1-63 Submitted 9/16; Revised 7/19; Published 8/19

Generic Inference in Latent Gaussian Process Models

Edwin V. Bonilla* E.BONILLA@DATA61.CSIRO.AU
Machine Learning Research Group
CSIRO's Data61
Sydney NSW 2015, Australia

Karl Krauth* KARL.KRAUTH@GMAIL.COM
Department of Electrical Engineering and Computer Science
University of California
Berkeley, CA 94720-1776, USA

Amir Dezfouli AKDEZFULI@GMAIL.COM
Machine Learning Research Group
CSIRO's Data61
Sydney NSW 2015, Australia

Editor: Manfred Opper

Abstract

We develop an automated variational method for inference in models with Gaussian process (GP) priors and general likelihoods. The method supports multiple outputs and multiple latent functions and does not require detailed knowledge of the conditional likelihood, only needing its evaluation as a black-box function. Using a mixture of Gaussians as the variational distribution, we show that the evidence lower bound and its gradients can be estimated efficiently using samples from *univariate* Gaussian distributions. Furthermore, the method is scalable to large datasets which is achieved by using an augmented prior via the inducing-variable approach underpinning most sparse GP approximations, along with parallel computation and stochastic optimization. We evaluate our approach quantitatively and qualitatively with experiments on small datasets, medium-scale datasets and large datasets, showing its competitiveness under different likelihood models and sparsity levels. On the large-scale experiments involving prediction of airline delays and classification of handwritten digits, we show that our method is on par with the state-of-the-art hard-coded approaches for scalable GP regression and classification.

Keywords: Gaussian processes, black-box likelihoods, nonlinear likelihoods, scalable inference, variational inference

1. Introduction

Developing fully automated methods for inference in complex probabilistic models has become arguably one of the most exciting areas of research in machine learning, with notable examples in the probabilistic programming community (see e.g. Hoffman and Gelman, 2014; Wood et al., 2014; Goodman et al., 2008; Domingos et al., 2006). Indeed, while these probabilistic programming systems allow for the formulation of very expressive and flexible probabilistic models, building efficient inference methods for reasoning in such systems is a significant challenge.

One particularly interesting setting is when the prior over the latent parameters of the model can be well described with a Gaussian process (GP, Rasmussen and Williams, 2006). GP priors are a popular choice in practical non-parametric Bayesian modeling with perhaps the most straightforward and well-known application being the standard regression model with Gaussian likelihood, for which the posterior can be computed in closed form (see e.g. Rasmussen and Williams, 2006, §2.2).

Interest in GP models stems from their functional non-parametric Bayesian nature and there are at least three critical benefits from adopting such a modeling approach. Firstly, by being a prior over functions, they represent a much more elegant way to address problems such as regression, where it is less natural to think of a prior over parameters such as the weights in a linear model. Secondly, by being Bayesian, they provide us with a principled way to combine our prior beliefs with the observed data in order to predict a full posterior distribution over unknown quantities, which of course is much more informative than a single point prediction. Finally, by being non-parametric, they address the common concern of parametric approaches about how well we can fit the data, since the model space is not constrained to have a parametric form.

1.1. Key Inference Challenges in Gaussian Process Models

Nevertheless, such a principled and flexible approach to machine learning comes at the cost of facing two fundamental probabilistic inference challenges, (i) scalability to large datasets and (ii) dealing with nonlinear likelihoods. With regards to the first challenge, scalability, GP models have been notorious for their poor scalability as a function of the number of training points. Despite ground-breaking work in understanding scalable GPs through the so-called sparse approximations (Quiñonero-Candela and Rasmussen, 2005) and in developing practical inference methods for these models (Titsias, 2009), recent literature on addressing this challenge gives a clear indication that the scalability problem is still a very active area of research, see e.g. Das et al. (2015); Deisenroth and Ng (2015); Hoang et al. (2015); Dong et al. (2017); Hensman et al. (2017); Salimbeni et al. (2018); John and Hensman (2018).

Concerning the second challenge, dealing with nonlinear likelihoods, the main difficulty is that of estimating a posterior over latent functions distributed according to a GP prior, given observations assumed to be drawn from a possibly nonlinear conditional likelihood model. In general, this posterior is analytically intractable and one must resort to approximate methods. These methods can be roughly classified into *stochastic* approaches and *deterministic* approaches. The former, stochastic approaches, are based on sampling algorithms such as Markov Chain Monte Carlo (MCMC, see e.g. Neal, 1993, for an overview) and the latter, deterministic approaches, are based on optimization techniques and include variational inference (VI, Jordan et al., 1998), the Laplace approximation (see e.g. MacKay, 2003, Chapter 27) and expectation propagation (EP, Minka, 2001).

On the one hand, although stochastic techniques such as MCMC provide a flexible framework for sampling from complex posterior distributions of probabilistic models, their generality comes at the expense of a very high computational cost as well as cumbersome con-

*. Joint first author.

©2019 Edwin V. Bonilla, Karl Krauth and Amir Dezfouli. JMLR 20(117):1-63, 2019
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/16-437.html.

vergence analysis. On the other hand, deterministic methods such as variational inference build upon the main insight that optimization is generally easier than integration. Consequently, they estimate a posterior by maximizing a lower bound of the marginal likelihood, the so-called evidence lower bound (ELBO). Variational methods can be considerably faster than MCMC but they lack MCMC's broader applicability, usually requiring mathematical derivations on a model-by-model basis.[1]

1.2. Contributions

In this paper we address the above challenges by developing a scalable automated variational method for inference in models with Gaussian process priors and general likelihoods. This method reduces the overhead of the tedious mathematical derivations traditionally inherent to variational algorithms, allowing their application to a wide range of problems. In particular, we consider models with multiple latent functions, multiple outputs and non-linear likelihoods that satisfy the following properties: (i) factorization across latent functions and (ii) factorization across observations. The former assumes that, when there are more than one latent function, they are generated from independent GPs. The latter assumes that, given the latent functions, the observations are conditionally independent.

Existing GP models, such as regression (Rasmussen and Williams, 2006), binary and multi-class classification (Nickisch and Rasmussen, 2008; Williams and Barber, 1998), warped GPs (Snelson et al., 2003), log Gaussian Cox process (Møller et al., 1998), and multi-output regression (Wilson et al., 2012), all fall into this class of models. Furthermore, our approach goes well beyond standard settings for which elaborate learning machinery has been developed, as we only require access to the likelihood function in a black-box manner. As we shall see below, our inference method can scale up to very large datasets, hence we will refer to it as SAVIGP, which stands for *scalable automated variational inference for Gaussian process* models. The key characteristics of our method and contributions of this work are summarized below.

- *Black-box likelihoods:* As mentioned above, the main contribution of this work is to be able to carry out posterior inference with GP priors and general likelihood models, without knowing the details of the conditional likelihood (or its gradients) and only requiring its evaluation as a black-box function.

- *Scalable inference and stochastic optimization:* By building upon the inducing-variable approach underpinning most sparse approximations to GP models (Quiñonero-Candela and Rasmussen, 2005; Titsias, 2009), the computational complexity of our method is dominated by $\mathcal{O}(M^3)$ operations in time, where $M \ll N$ is the number of inducing variables and N the number of observations. This is in contrast to naive inference in GP models which has a time complexity of $\mathcal{O}(N^3)$. As the resulting ELBO decomposes over the training datapoints, our model is amenable to parallel computation and stochastic optimization. In fact, we provide an implementation that can scale up to a very large number of observations, exploiting stochastic optimization, multi-core architectures and GPU computation.

1. However, we note recent developments such as variational auto-encoders (Kingma and Welling, 2014) and refer the reader to the related work and the discussion (Sections 2.4 and 10, respectively).

- *Joint learning of model parameters:* As our approach is underpinned by variational inference principles, SAVIGP allows for learning of all model parameters, including posterior parameters, inducing inputs, covariance hyperparameters and likelihood parameters, within the same framework via maximization of the evidence lower bound (ELBO).

- *Multiple outputs and multiple latent functions:* SAVIGP is designed to support models with multiple outputs and multiple latent functions, such as in multi-class classification (Williams and Barber, 1998) and non-stationary multi-output regression (Wilson et al., 2012). It does so in a very flexible way, allowing the different GP priors on the latent functions to have different covariance functions and inducing inputs.

- *Flexible posterior:* SAVIGP uses a mixture of Gaussians as the approximating posterior distribution. This is a very general approach as it is well known that, with a sufficient number of components, almost any continuous density can be approximated with arbitrary accuracy (Maz'ya and Schmidt, 1996).

- *Statistical efficiency:* By using knowledge of the approximate posterior and the structure of the GP prior, we exploit the decomposition of the ELBO, into a KL-divergence term and an expected log likelihood term, to provide *statistically* efficient parameter estimates. In particular, we derive an analytical lower bound for the KL-divergence term and we show that, for general black-box likelihood models, the expected log likelihood term and its gradients can be computed efficiently using samples from *univariate* Gaussian distributions.

- *Efficient re-parametrization:* For the case of a single full Gaussian variational posterior, we show that it is possible to re-parametrize the model so that the optimal posterior can be represented using a parametrization that is linear in the number of observations. This parametrization becomes useful for denser models, i.e. for models that have a larger number of inducing variables.

- *Extensive experimentation:* We evaluate SAVIGP with experiments on small datasets, medium-scale datasets and two large datasets. The experiments on small datasets ($N < 1,300$) evaluate the method under different likelihood models and sparsity levels (as determined by the number of inducing variables), including problems such as regression, classification, Log Gaussian Cox processes, and warped GPs (Snelson et al., 2003). We show that SAVIGP can perform as well as hard-coded inference methods under high levels of sparsity. The medium-scale experiments consider binary and multi-class classification using the MNIST dataset ($N = 60,000$) and non-stationary regression under the GPRN model (Wilson et al., 2012) using the SARCOS dataset ($N \approx 45,000$). Besides showing the competitiveness of our model for problems at this scale, we analyze the effect of learning the inducing inputs, i.e. the location of inducing variables. In our first large-scale experiment, we study the problem of predicting airline delays (using $N = 700,000$), and show that our method is on par with the state-of-the-art approach for scalable GP regression (Hensman et al., 2013), which uses full knowledge of the likelihood model. In our second large-scale experiment, we consider the MNIST8M dataset, which is an augmented version of MNIST (containing

$N = 8,100,000$ observations). We show that by using this augmented dataset we can improve performance significantly. Finally, in an experiment concerning a non-linear seismic inversion problem, we show that our approach can be applied easily (without any changes to the inference algorithm) to non-standard machine learning tasks and that our inference method can match closely the solution found by more computationally demanding approaches such as MCMC.

Before describing the family of GP models that we focus on, we start by relating our work to the previous literature concerning the key inference challenges mentioned above, i.e. scalability and dealing with non-linear likelihoods.

2. Related Work

As pointed out by Rasmussen and Williams (2006), there has been a long-standing interest in Gaussian processes with early work dating back at least to the 1880s when Danish astronomer and mathematician T. N. Thiel, concerned with determining the distance between Copenhagen and Lund from astronomical observations, essentially proposed the first mathematical formulation of Brownian motion (see e.g. Lauritzen, 1981, for details). In geostatistics, GP regression is known as Kriging (see e.g. Cressie, 1993) where, naturally, it has focused on 2-dimensional and 3-dimensional problems.

While the work of O'Hagan and Kingman (1978) has been quite influential in applying GPs to general regression problems, the introduction of GP regression to main-stream machine machine learning by Williams and Rasmussen (1996) sparked significant interest in the machine learning community. Indeed, henceforth, the community has taken GP models well beyond the standard regression setting, addressing other problems such as non-stationary and heteroscedastic regression (Paciorek and Schervish, 2004; Kersting et al., 2007; Wilson et al., 2012); nonlinear dimensionality reduction (Lawrence, 2005); classification (Williams and Barber, 1998; Nickisch and Rasmussen, 2008); multi-task learning (Bonilla et al., 2008; Yu and Chu, 2008; Alvarez and Lawrence, 2009; Wilson et al., 2012); preference learning (Bonilla et al., 2010); and ordinal regression (Chu and Ghahramani, 2005). In fact, their book (Rasmussen and Williams, 2006) is the de facto reference in any work related to Gaussian process models in machine learning. As we shall see below, despite all these significant advances, most previous work in the GP community has focused on addressing the scalability and the non-linear likelihood challenges in isolation.

2.1. Scalable Models

The cubic scaling on the number of observations of Gaussian process models, until very recently, has hindered the use of these models in a wider variety of applications. Work on approaching this problem has ranged from selecting informative (inducing) datapoints from the training data so as to facilitate sparse approximations to the posterior (Lawrence et al., 2002) to considering these inducing points as continuous parameters and optimizing them within a coherent probabilistic framework (Snelson and Ghahramani, 2006).

Although none of these methods actually scale to very large datasets as their time complexity is $\mathcal{O}(M^2N)$, where N is the number of observations and M is the number of inducing points, unifying such approaches from a probabilistic perspective (Quiñonero-

Candela and Rasmussen, 2005) has been extremely valuable to the community, not only to understand what those methods are actually doing but also to develop new approaches to sparse GP models. In particular, the framework of Titsias (2009) which has been placed within a solid theoretical grounding by Matthews et al. (2016), has become the underpinning machinery of modern scalable approaches to GP regression and classification. This has been taken one step further by allowing optimization of variational parameters within a stochastic optimization framework (Hensman et al., 2013), hence enabling the applicability of these inference methods to very large datasets.

Besides variational approaches based on reverse-KL divergence minimization, other methods have adopted different inference engines, based e.g. on the minimization of the forward KL divergence, such as expectation propagation (Hernández-Lobato and Hernández-Lobato, 2016). It turns out that all these methods can be seen from a more general perspective as minimization of α-divergences, see Bui et al. (2017) and references therein.

In addition to inducing-variable approaches to scalability in GP-models, other approaches have exploited the relationship between infinite-feature linear-in-the-parameters models and GPs. In particular, early work investigated truncated linear-in-the-parameters models as approximations to GP regression (see e.g. Tipping, 2001). This idea has been developed in an alternative direction that exploits the relationship between a covariance function of a stationary process and its spectral density, hence providing random-feature approximations to the covariance function of a GP, similarly to the work of Rahimi and Recht (2008, 2009), who focused on non-probabilistic kernel machines. For example, Lázaro-Gredilla et al. (2010), Gal and Turner (2015) and Yang et al. (2015) have followed these types of approaches, with the latter mostly concerned about having fast kernel evaluations.

On another line of research, Wilson and Nickisch (2015) proposed a structured kernel interpolation method that uses local interpolation strategies that approximate the cross-covariances between the training data and the inducing inputs, while also allowing for the exploitation of Kronecker and Toeplitz algebra.

Unlike our work, none of these approaches deals with the harder task of developing scalable inference methods for multi-output problems and general likelihood models.

2.2. Multi-output and Multi-task Gaussian Processes

Developing multi-task or multi-output learning approaches using Gaussian processes has also proved an intensive area of research. Most GP-based models for these problems assume that correlations between the outputs are induced via linear combinations of a set of independent latent processes. Such linear combinations can use fixed coefficients (see e.g. Teh et al., 2005) or input-dependent coefficients (Wilson et al., 2012; Nguyen and Bonilla, 2013). Tasks dependencies can also be defined explicitly through the covariance function as done by Bonilla et al. (2008), who assume a covariance decomposition between tasks and inputs. More complex dependencies can be modeled by using the convolution formalism as done in earlier work by Boyle and Frean (2005) and generalized by Alvarez and Lawrence (2009); Álvarez et al. (2010), whose later work also provides efficient inference algorithms for such models (Álvarez and Lawrence, 2011). Finally, the work of Nguyen and Bonilla (2014b) also assumes a linear combination of independent latent processes but it exploits the developments of Hensman et al. (2013) to scale up to very large datasets.

Besides the work of Nguyen and Bonilla (2014b), these approaches do not scale to a very large number of observations and all of them are mainly concerned with regression problems.

2.3. General Nonlinear Likelihoods

The problem of inference in models with GP priors and nonlinear likelihoods has been tackled from a sampling perspective with algorithms such as elliptical slice sampling (ESS, Murray et al., 2010), which are more effective at drawing samples from strongly correlated Gaussians than generic MCMC algorithms. Nevertheless, as we shall see in Section 9.3.5, the sampling cost of ESS remains a major challenge for practical usages.

From a variational inference perspective, the work by Opper and Archambeau (2009) has been slightly under-recognized by the community even though it proposes an efficient full Gaussian posterior approximation for GP models with i.i.d. observations. Our work pushes this breakthrough further by allowing multiple latent functions, multiple outputs, and more importantly, scalability to large datasets.

Another approach to deterministic approximate inference is the integrated nested Laplace approximation (INLA, Rue et al., 2009). INLA uses numerical integration to approximate the marginal likelihood, which makes it unsuitable for GP models that contain a large number of hyperparameters.

2.4. More Recent Developments

As mentioned in §1, the scalability problem continues to attract the interest of researchers working with GP models, with recently developed distributed inference frameworks (Gal et al., 2014; Deisenroth and Ng, 2015), and the variational inference frameworks for scalable GP regression and classification by Hensman et al. (2013) and Hensman et al. (2015), respectively. As with the previous work described in §2.1, these approaches have been limited to classification or regression problems or specific to a particular class of likelihood models such as GP-modulated Cox processes (John and Hensman, 2018).

Contemporarily to the work of Nguyen and Bonilla (2014a), which underpins our approach, Ranganath et al. (2014) developed black-box variational inference (BBVI) for general latent variable models. Due to this generality, it under-utilizes the rich amount of information available in Gaussian process models. For example, BBVI approximates the KL-divergence term of the evidence lower bound but this is computed analytically in our method. Additionally, for practical reasons, BBVI imposes the variational distribution to fully factorize over the latent variables, while we make no such a restriction. A clear disadvantage of BBVI is that it does not provide a practical way of learning the covariance hyperparameters of GPs—in fact, these are set to fixed values. In principle, these values can be learned in BBVI using stochastic optimization, but experimentally, we have found this to be problematic, ineffectual, and time-consuming.

Very recently, Bonilla et al. (2016) have used the random-feature approach mentioned in Section 2.1 along with linearization techniques to provide scalable methods for inference in GP models with general likelihoods. Unlike our approach for estimating expectations of the conditional likelihood, such linearizations of the conditional likelihood are approximations

and generally do not converge to the exact expectations even in the limit of a large number of observations.

Following the recent advances in making automatic differentiation widely available and easy to use in practical systems (see e.g. Baydin et al., 2015), developments on stochastic variational inference for fairly complex probabilistic models (Rezende et al., 2014; Kingma and Welling, 2014) can be used when the likelihood can be implemented in such systems and can be expressed using the so-called *re-parametrization trick* (see e.g. Kingma and Welling, 2014, for details). Nevertheless, the advantage of our method over these generic approaches is twofold. Firstly, as with BBVI, in practice such a generality comes at the cost of making assumptions about the posterior (such as factorization), which is not suitable for GP models. Secondly, and more importantly, such methods are not truly black-box as they require explicit access to the implementation of the conditional likelihood. An interesting example where one cannot apply the re-parametrization trick is given by Challis and Barber (2013), who describe a large class of functions (that include the Laplace log likelihood) that are neither differentiable or continuous but their expectation over a Gaussian posterior is smooth. A more general setting where a truly black-box approach is required concerns inversion problems (Tarantola, 2004) where latent functions are passed through domain-specific forward models followed by a known noise model (Bonilla et al., 2016; Reid et al., 2013). These forward models may be non-differentiable, given as an executable, or too complex to re-implement quickly in an automatic differentiation framework. To illustrate this case, we present results on a seismic inversion application in §9.8.

Nevertheless, we acknowledge that some of these developments mentioned above have been extended to the GP literature, where the re-parametrization trick has been used (Krauth et al., 2017; Matthews et al., 2017; Hensman et al., 2017; Cutajar et al., 2017). These contemporary works show that it is worthwhile building more tailored methods that may require a lower number of samples to estimate the expectations in our framework.

Finally, a related area of research is that of modeling complex data with deep belief networks based on Gaussian process mappings (Damianou and Lawrence, 2013), which have been proposed primarily as hierarchical extensions of the Gaussian process latent variable model (Lawrence, 2005). Unlike our approach, these models target the unsupervised problem of discovering structure in high-dimensional data and have focused mainly on small-data applications. However, much like the recent work on "shallow"-GP architectures, inference in these models have also been made scalable for supervised and unsupervised learning, exploiting the reparameterization trick, e.g. using random-feature expansions (Cutajar et al., 2017) or inducing-variable approximations (Salimbeni and Deisenroth, 2017).

3. Latent Gaussian Process Models

Before starting our description of the types of models we are addressing in this paper, we refer the reader to Appendix L for a summary of the notation used henceforth.

We consider supervised learning problems where we are given a dataset $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$, where $\mathbf{x}_n$ is a D-dimensional input vector and $\mathbf{y}_n$ is a P-dimensional output. Our goal is to learn the mapping from inputs to outputs, which can be established via Q underlying latent functions $\{f_j\}_{j=1}^Q$. In many problems these latent functions have a physical interpretation,

while in others they are simply nuisance parameters and we just want to integrate them out in order to make probabilistic predictions.

A sensible modeling approach to the above problem is to assume that the Q latent functions $\{f_j\}$ are uncorrelated a priori and that they are drawn from Q zero-mean Gaussian processes (Rasmussen and Williams, 2006):

$$p(f_j|\boldsymbol{\theta}_j) \sim \mathcal{GP}\left(0, \kappa_j(\cdot,\cdot;\boldsymbol{\theta}_j)\right), \quad j = 1, \dots Q, \quad \text{then}$$
$$p(\mathbf{f}|\boldsymbol{\theta}) = \prod_{j=1}^{Q} p(\mathbf{f}_{\cdot j}|\boldsymbol{\theta}_j) = \prod_{j=1}^{Q} \mathcal{N}(\mathbf{f}_{\cdot j}; \mathbf{0}, \mathbf{K}^j_{\mathbf{xx}}), \tag{1}$$

where $\mathbf{f}$ is the set of all latent function values; $\mathbf{f}_{\cdot j} = \{f_j(\mathbf{x}_n)\}_{n=1}^N$ denotes the values of latent function j; $\mathbf{K}^j_{\mathbf{xx}}$ is the covariance matrix induced by the covariance function $\kappa_j(\cdot,\cdot;\boldsymbol{\theta}_j)$ evaluated at every pair of inputs; and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}$ are the parameters of the corresponding covariance functions. Along with the prior in Equation (1), we can also assume that our multi-dimensional observations $\{\mathbf{y}_n\}$ are i.i.d. given the corresponding set of latent functions $\{\mathbf{f}_n\}$:

$$p(\mathbf{y}|\mathbf{f}, \boldsymbol{\phi}) = \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi}), \tag{2}$$

where $\mathbf{y}$ is the set of all output observations; $\mathbf{y}_n$ is the nth output observation; $\mathbf{f}_{n\cdot} = \{f_j(\mathbf{x}_n)\}_{j=1}^Q$ is the set of latent function values which $\mathbf{y}_n$ depends upon; and $\boldsymbol{\phi}$ are the conditional likelihood parameters. In the sequel, we will refer to the covariance parameters ($\boldsymbol{\theta}$) as the model hyperparameters.

In other words, we are interested in models for which the following criteria are satisfied:

(i) *factorization of the prior over the latent functions*, as specified by Equation (1), and

(ii) *factorization of the conditional likelihood over the observations given the latent functions*, as specified by Equation (2).

We refer to the models satisfying the above assumptions (when using GP priors) as latent Gaussian process models (LGPMs). Interestingly, a large class of problems can be well modeled with the above assumptions. For example binary classification (Nickisch and Rasmussen, 2008; Williams and Barber, 1998), warped GPs (Snelson et al., 2003), log Gaussian Cox processes (Møller et al., 1998), multi-class classification (Williams and Barber, 1998), and multi-output regression (Wilson et al., 2012) all belong to this family of models.

More importantly, besides the i.i.d. assumption, there are not additional constraints on the conditional likelihood which can be any linear or nonlinear model. Furthermore, as we shall see in §4, our proposed inference algorithm only requires evaluations of this likelihood model in a black-box manner, i.e. without requiring detailed knowledge of its implementation or its gradients.

3.1. Inference Challenges

As mentioned in §1, for general LGPMs, the inference problem of estimating the posterior distribution over the latent functions given the observed data $p(\mathbf{f}|\mathcal{D})$ and, ultimately, for

a new observation $\mathbf{x}_\star$, estimating the predictive posterior distribution $p(\mathbf{f}_{\star\cdot}|\mathbf{x}_\star, \mathcal{D})$, poses two important challenges (even in the case of a single latent function $Q = 1$) from the computational and statistical perspectives. These challenges are (i) scalability to large datasets and (ii) dealing with nonlinear likelihoods.

We address the *scalability challenge* inherent to GP models (given their cubic time complexity on the number of observations) by augmenting our priors using an inducing-variable approach (see e.g. Quiñonero-Candela and Rasmussen, 2005) and embedding our model into a variational inference framework (Titsias, 2009). In short, inducing variables in sparse GP models act as latent summary statistics, avoiding the computation of large inverse covariances. Crucially, unlike other approaches (Quiñonero-Candela and Rasmussen, 2005), we keep an explicit representation of these variables (and integrate them out variationally), which facilitates the decomposition of the variational objective into a sum on the individual datapoints. This allows us to devise stochastic optimization strategies and parallel implementations in cloud computing services such as Amazon EC2. Such a strategy, also allows us to learn the location of the inducing variables (i.e. the inducing inputs) in conjunction with variational parameters and hyperparameters, which in general provides better performance, especially in high-dimensional problems.

To address the *nonlinear likelihood challenge*, which from a variational perspective boils down to estimating expectations of a nonlinear function over the approximate posterior, we follow a stochastic estimation approach, in which we develop low-variance Monte Carlo estimates of the expected log-likelihood term in the variational objective. Crucially, we will show that the expected log-likelihood term can be estimated efficiently by using only samples from univariate Gaussian distributions.

4. Scalable Inference

Here we describe our scalable inference method for the model class specified in §3. We build upon the inducing-variable formalism underlying most sparse GP approximations (Quiñonero-Candela and Rasmussen, 2005; Titsias, 2009) and obtain an algorithm with time complexity $\mathcal{O}(M^3)$, where M is the number of inducing variables per latent process. We show that, under this sparse approximation and the variational inference framework, the expected log likelihood term and its gradient can be estimated using only samples from univariate Gaussian distributions.

4.1. Augmented Prior

In order to make inference scalable we redefine our prior in terms of some auxiliary variables $\{\mathbf{u}_{\cdot j}\}_{j=1}^Q$, which we will refer to as the *inducing variables*. These inducing variables lie in the same space as $\{\mathbf{f}_{\cdot j}\}$ and are drawn from the same zero-mean GP priors. As before, we assume factorization of the prior across the Q latent functions. Hence the resulting augmented prior is given by:

$$p(\mathbf{u}) = \prod_{j=1}^{Q} \mathcal{N}(\mathbf{u}_{\cdot j}; \mathbf{0}, \mathbf{K}^j_{\mathbf{zz}}), \qquad p(\mathbf{f}|\mathbf{u}) = \prod_{j=1}^{Q} \mathcal{N}(\mathbf{f}_{\cdot j}; \tilde{\boldsymbol{\mu}}_j, \widetilde{\mathbf{K}}_j), \text{ where} \tag{3}$$
$$\tilde{\boldsymbol{\mu}}_j = \mathbf{K}^j_{\mathbf{xz}}(\mathbf{K}^j_{\mathbf{zz}})^{-1}\mathbf{u}_{\cdot j}, \text{ and}$$

$$\widetilde{\mathbf{K}}_j = \mathbf{K}^j_{\mathbf{xx}} - \mathbf{A}_j \mathbf{K}^j_{\mathbf{zx}} \text{ with } \mathbf{A}_j = \mathbf{K}^j_{\mathbf{xz}} (\mathbf{K}^j_{\mathbf{zz}})^{-1}, \tag{4}$$

where $\mathbf{u}_{\cdot j}$ are the inducing variables for latent process j and $\mathbf{u}$ is the set of all the inducing variables. The covariance matrices above are obtained by evaluating the covariance functions at the corresponding inputs, i.e., $\mathbf{K}^j_{\mathbf{zz}} \stackrel{\text{def}}{=} \kappa_j(\mathbf{Z}_j, \mathbf{Z}_j; \boldsymbol{\theta}_j)$, $\mathbf{K}^j_{\mathbf{xx}} \stackrel{\text{def}}{=} \kappa_j(\mathbf{X}, \mathbf{X}; \boldsymbol{\theta}_j)$, $\mathbf{K}^j_{\mathbf{xz}} \stackrel{\text{def}}{=} \kappa_j(\mathbf{X}, \mathbf{Z}_j; \boldsymbol{\theta}_j) = (\mathbf{K}^j_{\mathbf{zx}})^T$, where $\mathbf{Z}_j$ are all the inducing inputs for latent process j and $\mathbf{X}$ is the matrix of all input locations $\{\mathbf{x}_n\}$. We note that while each of the inducing variables in $\mathbf{u}_{\cdot j}$ lies in the same space as the elements in $\mathbf{f}_{\cdot j}$, each of the M inducing inputs in $\mathbf{Z}_j$ lies in the same space as each input data point $\mathbf{x}_n$. Therefore, while $\mathbf{u}_{\cdot j}$ is a M-dimensional vector, $\mathbf{Z}_j$ is a $M \times D$ matrix where each of the rows corresponds to a D-dimensional inducing input. We refer the reader to Appendix L for a summary of the notation and dimensionality of the above kernel matrices.

As thoroughly investigated by Quiñonero-Candela and Rasmussen (2005), most GP approximations can be formulated using the augmented prior above and additional assumptions on the training and test conditional distributions $p(\mathbf{f}|\mathbf{u})$ and $p(\mathbf{f}_{\star\cdot}|\mathbf{u})$, respectively. Such approaches have been traditionally referred to as *sparse* approximations and we will use this terminology as well. Analogously, we will refer to models with a larger number of inducing inputs as *denser* models.

It is important to emphasize that the joint prior $p(\mathbf{f}, \mathbf{u})$ defined in Equation (3) is an equivalent prior to that in the original model, as if we integrate out the inducing variables $\mathbf{u}$ from this joint prior we will obtain the prior $p(\mathbf{f})$ in Equation (1) exactly. Nevertheless, as we shall see later, following a variational inference approach and having an explicit representation of the (approximate) posterior over the inducing variables will be fundamental to scaling up inference in these types of models without making the assumptions on the training or test conditionals described in Quiñonero-Candela and Rasmussen (2005).

Along with the joint prior defined above, we maintain the factorization assumption of the conditional likelihood given in Equation (2).

4.2. Variational Inference and the Evidence Lower Bound

Given the prior in Equation (3) and the likelihood in Equation (2), posterior inference for general (non-Gaussian) likelihoods is analytically intractable. Therefore, we resort to approximate methods such as variational inference (Jordan et al., 1998). Variational inference methods entail positing a tractable family of distributions and finding the member of the family that is "closest" to the true posterior in terms of their the Kullback-Leibler divergence. In our case, we are seeking to approximate the joint posterior $p(\mathbf{f}, \mathbf{u}|\mathbf{y})$ with a variational distribution $q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda})$.

4.3. Approximate Posterior

Motivated by the fact that the true joint posterior is given by $p(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u}, \mathbf{y})p(\mathbf{u}|\mathbf{y})$, our approximate posterior has the form:

$$q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u}|\boldsymbol{\lambda}), \tag{5}$$

where $p(\mathbf{f}|\mathbf{u})$ is the conditional prior given in Equation (3) and $q(\mathbf{u}|\boldsymbol{\lambda})$ is our approximate (variational) posterior. This decomposition has proved effective in regression problems with a single latent process and a single output (see e.g. Titsias, 2009).

Hence, we can define our variational distribution using a mixture of Gaussians (MoG):

$$q(\mathbf{u}|\boldsymbol{\lambda}) = \sum_{k=1}^{K} \pi_k q_k(\mathbf{u}|\mathbf{m}_k, \mathbf{S}_k) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{Q} \mathcal{N}(\mathbf{u}_{\cdot j}; \mathbf{m}_{kj}, \mathbf{S}_{kj}), \tag{6}$$

where $\boldsymbol{\lambda} = \{\pi_k, \mathbf{m}_{kj}, \mathbf{S}_{kj}\}$ are the variational parameters: the mixture proportions $\{\pi_k\}$, the posterior means $\{\mathbf{m}_{kj}\}$ and posterior covariances $\{\mathbf{S}_{kj}\}$ of the inducing variables corresponding to mixture component k and latent function j. We also note that each of the mixture components $q_k(\mathbf{u}|\mathbf{m}_k, \mathbf{S}_k)$ is a Gaussian with mean $\mathbf{m}_k$ and block-diagonal covariance $\mathbf{S}_k$.

An early reference for using a mixture of Gaussians (MoG) within variational inference is given by Bishop et al. (1998) in the context of Bayesian networks. Similarly, Gershman et al. (2012) have used MoG for non-GP models and, unlike our approach, used a second-order Taylor series approximation of the variational lower bound.

4.4. Variational Lower Bound

It is easy to show that minimizing the Kullback-Leibler divergence between our approximate posterior and the true posterior, $\mathrm{KL}(q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda}) \| p(\mathbf{f}, \mathbf{u}|\mathbf{y}))$ is equivalent to maximizing the log-evidence lower bound ($\mathcal{L}_{\text{elbo}}$), which is composed of a KL-term ($\mathcal{L}_{\text{kl}}$) and an expected log-likelihood term ($\mathcal{L}_{\text{ell}}$). In other words:

$$\log p(\mathbf{y}) \geq \mathcal{L}_{\text{elbo}}(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \mathcal{L}_{\text{kl}}(\boldsymbol{\lambda}) + \mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}), \text{ where} \tag{7}$$

$$\mathcal{L}_{\text{kl}}(\boldsymbol{\lambda}) = -\mathrm{KL}(q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda}) \| p(\mathbf{f}, \mathbf{u})), \text{ and} \tag{8}$$

$$\mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda})}[\log p(\mathbf{y}|\mathbf{f})], \tag{9}$$

where $\mathbb{E}_{q(x)}[g(x)]$ denotes the expectation of function $g(x)$ over distribution $q(x)$. Here we note that $\mathcal{L}_{\text{kl}}$ is a negative KL divergence between the joint approximate posterior $q(\mathbf{f}, \mathbf{u}|\boldsymbol{\lambda})$ and the joint prior $p(\mathbf{f}, \mathbf{u})$. Therefore, maximization of the $\mathcal{L}_{\text{elbo}}$ in Equation (7) entails minimization of an expected loss (given by the negative expected log-likelihood $\mathcal{L}_{\text{ell}}$) regularized by $\mathcal{L}_{\text{kl}}$, which imposes the constraint of finding solutions to our approximate posterior that are close to the prior in the KL-sense.

An interesting observation of the decomposition of the $\mathcal{L}_{\text{elbo}}$ objective is that, unlike $\mathcal{L}_{\text{ell}}$, $\mathcal{L}_{\text{kl}}$ in Equation (8) does not depend on the conditional likelihood $p(\mathbf{y}|\mathbf{f})$, for which we do not assume any specific parametric form (i.e. black-box likelihood). We can thus address the technical difficulties regarding each component and their gradients separately using different approaches. In particular, for $\mathcal{L}_{\text{kl}}$ we will exploit the structure of the variational posterior in order to avoid computing KL-divergences over distributions involving all the data. Furthermore, we will obtain a lower bound for $\mathcal{L}_{\text{kl}}$ in the general case of $q(\mathbf{u})$ being a mixture-of-Gaussians (MoG) as given in Equation (6). For the expected log-likelihood term ($\mathcal{L}_{\text{ell}}$) we will use a Monte Carlo approach in order to estimate the required expectation and its gradients.

4.5. Computation of the KL-divergence term ($\mathcal{L}_{\text{kl}}$)

In order to have an explicit form for $\mathcal{L}_{\text{kl}}$ and its gradients, we start by expanding Equation (8):

$$\mathcal{L}_{\text{kl}}(\boldsymbol{\lambda}) = -\text{KL}(q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})\|p(\mathbf{f},\mathbf{u})) = -\mathbb{E}_{q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})}\left[\log \frac{q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})}{p(\mathbf{f},\mathbf{u})}\right], \tag{10}$$

$$= -\mathbb{E}_{p(\mathbf{f}|\mathbf{u})q(\mathbf{u}|\boldsymbol{\lambda})}\left[\log \frac{q(\mathbf{u}|\boldsymbol{\lambda})}{p(\mathbf{u})}\right], \tag{11}$$

$$= -\text{KL}(q(\mathbf{u}|\boldsymbol{\lambda})\|p(\mathbf{u})), \tag{12}$$

where we have applied the definition of the KL-divergence in Equation (10); used the variational joint posterior $q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})$ given in Equation (5) to go from Equation (10) to Equation (11); and integrated out $\mathbf{f}$ to obtain Equation (12). We note that the definition of the joint posterior $q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})$ in Equation (5) has been crucial to transform a KL-divergence between the joint approximate posterior and the joint prior into a KL-divergence between the variational posterior $q(\mathbf{u}|\boldsymbol{\lambda})$ and the prior $p(\mathbf{u})$ over the inducing variables. In doing that, we have avoided computing a KL-divergence between distributions over the N-dimensional variables $\mathbf{f}_{\cdot j}$. As we shall see later, this implies a reduction in time complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(M^3)$, where N is the number of datapoints and M is the number of inducing variables.

We now decompose the resulting KL-divergence term in Equation (12) as follows,

$$\begin{aligned}\mathcal{L}_{\text{kl}}(\boldsymbol{\lambda}) &= -\text{KL}(q(\mathbf{u}|\boldsymbol{\lambda})\|p(\mathbf{u})) = \mathcal{L}_{\text{ent}}(\boldsymbol{\lambda}) + \mathcal{L}_{\text{cross}}(\boldsymbol{\lambda}), \text{ where:}\\ \mathcal{L}_{\text{ent}}(\boldsymbol{\lambda}) &= -\mathbb{E}_{q(\mathbf{u}|\boldsymbol{\lambda})}[\log q(\mathbf{u}|\boldsymbol{\lambda})], \text{ and}\\ \mathcal{L}_{\text{cross}}(\boldsymbol{\lambda}) &= \mathbb{E}_{q(\mathbf{u}|\boldsymbol{\lambda})}[\log p(\mathbf{u})],\end{aligned} \tag{13}$$

where $\mathcal{L}_{\text{ent}}(\boldsymbol{\lambda})$ denotes the differential entropy of the approximating distribution $q(\mathbf{u}|\boldsymbol{\lambda})$ and $\mathcal{L}_{\text{cross}}(\boldsymbol{\lambda})$ denotes the negative cross-entropy between the approximating distribution $q(\mathbf{u}|\boldsymbol{\lambda})$ and the prior $p(\mathbf{u})$.

Computing the entropy of the variational distribution in Equation (6), which is a mixture-of-Gaussians (MoG), is analytically intractable. However, a lower bound of this entropy can be obtained using Jensen's inequality (see e.g. Huber et al., 2008) giving:

$$\mathcal{L}_{\text{ent}}(\boldsymbol{\lambda}) \geq -\sum_{k=1}^{K}\pi_k \log \sum_{\ell=1}^{K}\pi_\ell \mathcal{N}(\mathbf{m}_k;\mathbf{m}_\ell,\mathbf{S}_k+\mathbf{S}_\ell) \stackrel{\text{def}}{=} \widehat{\mathcal{L}}_{\text{ent}}. \tag{14}$$

The negative cross-entropy in Equation (13) between a Gaussian mixture $q(\mathbf{u}|\boldsymbol{\lambda})$ and a Gaussian $p(\mathbf{u})$, can be obtained analytically,

$$\mathcal{L}_{\text{cross}}(\boldsymbol{\lambda}) = -\frac{1}{2}\sum_{k=1}^{K}\pi_k\sum_{j=1}^{Q}[M\log 2\pi + \log\left|\mathbf{K}^j_{\mathbf{zz}}\right| + \mathbf{m}^T_{kj}(\mathbf{K}^j_{\mathbf{zz}})^{-1}\mathbf{m}_{kj} + \text{ tr }(\mathbf{K}^j_{\mathbf{zz}})^{-1}\mathbf{S}_{kj}]. \tag{15}$$

The derivation of the entropy term and the cross-entropy term in Equations (14) and (15) is given in Appendix A.

4.6. Estimation of the expected log likelihood term ($\mathcal{L}_{\text{ell}}$)

We now address the computation of the expected log likelihood term in Equation (9). The main difficulty of computing this term is that, unlike the $\mathcal{L}_{\text{kl}}$ where we have full knowledge of the prior and the approximate posterior, here we do not assume a specific form for the conditional likelihood $p(\mathbf{y}|\mathbf{f},\boldsymbol{\phi})$. Furthermore, we only require evaluations of $\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot},\boldsymbol{\phi})$ for each datapoint n, hence yielding a truly black-box likelihood method.

We will show one of our main results, that of *statistical efficiency* of our $\mathcal{L}_{\text{ell}}$ estimator. This means that, despite having a full Gaussian approximate posterior, estimation of the $\mathcal{L}_{\text{ell}}$ and its gradients only requires samples from univariate Gaussian distributions. We start by expanding Equation (9) using our definitions of the approximate posterior and the factorization of the conditional likelihood:

$$\mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}) = \mathbb{E}_{q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})}[\log p(\mathbf{y}|\mathbf{f},\boldsymbol{\phi})] = \mathbb{E}_{q(\mathbf{f}|\boldsymbol{\lambda})}[\log p(\mathbf{y}|\mathbf{f},\boldsymbol{\phi})], \tag{16}$$

where, given the definition of the approximate joint posterior $q(\mathbf{f},\mathbf{u}|\boldsymbol{\lambda})$ in Equation (5), the distribution $q(\mathbf{f}|\boldsymbol{\lambda})$ resulting from marginalizing $\mathbf{u}$ from this joint posterior can be obtained analytically,

$$q(\mathbf{f}|\boldsymbol{\lambda}) = \sum_{k=1}^{K}\pi_k q_k(\mathbf{f}|\boldsymbol{\lambda}_k) = \sum_{k=1}^{K}\pi_k\prod_{j=1}^{Q}\mathcal{N}(\mathbf{f}_j;\mathbf{b}_{kj},\boldsymbol{\Sigma}_{kj}), \text{ with} \tag{17}$$

$$\mathbf{b}_{kj} = \mathbf{A}_j\mathbf{m}_{kj}, \text{ and} \tag{18}$$

$$\boldsymbol{\Sigma}_{kj} = \widetilde{\mathbf{K}}_j + \mathbf{A}_j\mathbf{S}_{kj}\mathbf{A}_j^T, \tag{19}$$

where $\widetilde{\mathbf{K}}_j$ and $\mathbf{A}_j$ are given in Equation (4) and, as defined before, $\{\mathbf{m}_{kj}\}$ and $\{\mathbf{S}_{kj}\}$ are the posterior means and posterior covariances of the inducing variables corresponding to mixture component k and latent function j. We are now ready to state our result of a statistically efficient estimator for the $\mathcal{L}_{\text{ell}}$:

Theorem 1 *For the* GP *model with prior defined in Equations* (3) *to* (4)*, and conditional likelihood defined in Equation* (2)*, the expected log likelihood over the variational distribution in Equation* (5) *and its gradients can be estimated using samples from univariate Gaussian distributions.*

The proof is constructive and can be found in Appendix B. We note that a less general result, for the case of one latent function and a single variational Gaussian posterior, was obtained in Opper and Archambeau (2009) using a different derivation. Here we state our final result on how to compute these estimates:

$$\mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\pi_k\mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot},\boldsymbol{\phi})], \tag{20}$$

$$\nabla_{\boldsymbol{\lambda}_k}\mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}) = \pi_k\sum_{n=1}^{N}\mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}\left[\nabla_{\boldsymbol{\lambda}_k}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot},\boldsymbol{\phi})\right], \text{ for } \boldsymbol{\lambda}_k \in \{\mathbf{m}_k,\mathbf{S}_k\}, \tag{21}$$

$$\nabla_{\pi_k}\mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}) = \sum_{n=1}^{N}\mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot},\boldsymbol{\phi})], \tag{22}$$

where $q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)$ is a Q-dimensional Gaussian with:

$$q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k) = \mathcal{N}(\mathbf{f}_{n\cdot};\mathbf{b}_{k(n)},\boldsymbol{\Sigma}_{k(n)}),$$

where $\boldsymbol{\Sigma}_{k(n)}$ is a *diagonal* matrix. The jth element of the mean and the (j,j)th entry of the covariance of the above distribution are given by:

$$[\mathbf{b}_{k(n)}]_j = \mathbf{a}_{jn}^T\mathbf{m}_{kj}, \qquad [\boldsymbol{\Sigma}_{k(n)}]_{j,j} = [\widetilde{\mathbf{K}}_j]_{n,n} + \mathbf{a}_{jn}^T\mathbf{S}_{kj}\mathbf{a}_{jn}, \tag{23}$$

where $\mathbf{a}_{jn}^T \stackrel{\text{def}}{=} [\mathbf{A}_j]_{n,:}$ denotes the M-dimensional vector corresponding to the nth row of matrix $\mathbf{A}_j$; $\widetilde{\mathbf{K}}_j$ and $\mathbf{A}_j$ are given in Equation (4); and, as before, $\{\mathbf{m}_{kj},\mathbf{S}_{kj}\}$ are the variational parameters corresponding to the mean and covariance of the approximate posterior over the inducing variables for mixture component k and latent process j.

We emphasize that when $Q > 1$, $q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)$ is not a univariate marginal but a Q-dimensional marginal posterior with diagonal covariance. Therefore, only samples from univariate Gaussians are required to estimate the expressions in Equations (20) to (22).

4.6.1. Practical consequences and unbiased estimates

There are two immediate practical consequences of the result in Theorem 1. The first consequence is that we can use unbiased empirical estimates of the expected log likelihood term and its gradients. In our experiments we use Monte Carlo (MC) estimates, hence we can compute $\mathcal{L}_{\text{ell}}$ as:

$$\left\{\mathbf{f}_{n\cdot}^{(k,i)}\right\}_{i=1}^{S} \sim \mathcal{N}(\mathbf{f}_{n\cdot};\mathbf{b}_{k(n)},\boldsymbol{\Sigma}_{k(n)}),\ k=1,\ldots,K,$$
$$\widehat{\mathcal{L}}_{\text{ell}} = \frac{1}{S}\sum_{n=1}^{N}\sum_{k=1}^{K}\pi_k\sum_{i=1}^{S}\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}^{(k,i)},\boldsymbol{\phi}), \tag{24}$$

where $\mathbf{b}_{k(n)}$ and $\boldsymbol{\Sigma}_{k(n)}$ are the vector and matrix forms of the mean and covariance of the posterior over the latent functions as given in Equation (23). Analogous MC estimates of the gradients are given in Appendix C.2.

The second practical consequence is that in order to estimate the gradients of the $\mathcal{L}_{\text{ell}}$, using Equations (21) and (22), we only require evaluations of the conditional likelihood in a black-box manner, without resorting to numerical approximations or analytical derivations of its gradients.

5. Parameter Optimization

In order to learn the parameters of our model we seek to maximize our (estimated) log-evidence lower bound ($\widehat{\mathcal{L}}_{\text{elbo}}$) using gradient-based optimization. Let $\boldsymbol{\eta} = \{\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\phi}\}$ be all the model parameters for which point estimates are required. We have that:

$$\widehat{\mathcal{L}}_{\text{elbo}}(\boldsymbol{\eta}) \stackrel{\text{def}}{=} \widehat{\mathcal{L}}_{\text{ent}}(\boldsymbol{\eta}) + \mathcal{L}_{\text{cross}}(\boldsymbol{\eta}) + \widehat{\mathcal{L}}_{\text{ell}}(\boldsymbol{\eta}),$$

$$\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{L}}_{\text{elbo}} = \nabla_{\boldsymbol{\eta}}\widehat{\mathcal{L}}_{\text{ent}} + \nabla_{\boldsymbol{\eta}}\mathcal{L}_{\text{cross}} + \nabla_{\boldsymbol{\eta}}\widehat{\mathcal{L}}_{\text{ell}},$$

where we have made explicit the dependency of the log-evidence lower bound on any parameter of the model and $\widehat{\mathcal{L}}_{\text{ent}}$, $\mathcal{L}_{\text{cross}}$, and $\widehat{\mathcal{L}}_{\text{ell}}$ are given in Equations (14), (15), (24) respectively. The gradients of $\widehat{\mathcal{L}}_{\text{elbo}}$ wrt variational parameters $\boldsymbol{\lambda}$, covariance hyperparameters $\boldsymbol{\theta}$ and likelihood parameters $\boldsymbol{\phi}$ are given in Appendices C, D.1, and D.2, respectively. As shown in these appendices, not all constituents of $\widehat{\mathcal{L}}_{\text{elbo}}$ contribute to learning all parameters, for example $\nabla_{\boldsymbol{\theta}}\widehat{\mathcal{L}}_{\text{ent}} = \nabla_{\boldsymbol{\phi}}\widehat{\mathcal{L}}_{\text{ent}} = \nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{cross}} = 0$.

Using the above objective function ($\widehat{\mathcal{L}}_{\text{elbo}}$) and its gradients we can consider batch optimization with limited-memory BFGS for small datasets and medium-size datasets. However, under this batch setting, the computational cost can be too large to be afforded in practice even for medium-size datasets on single-core architectures.

5.1. Stochastic optimization

To deal with the above problem, we first note that the terms corresponding to the KL-divergence $\widehat{\mathcal{L}}_{\text{ent}}$ and $\mathcal{L}_{\text{cross}}$ in Equations (14) and (15) do not depend on the observed data, hence their computational complexity is independent of N. More importantly, we note that $\widehat{\mathcal{L}}_{\text{ell}}$ in Equation (24) decomposes as a sum of expectations over individual datapoints. This makes our inference framework amenable to parallel computation and stochastic optimization (Robbins and Monro, 1951). More precisely, we can rewrite $\widehat{\mathcal{L}}_{\text{elbo}}$ as:

$$\widehat{\mathcal{L}}_{\text{elbo}} = \sum_{n=1}^{N}\left[\frac{1}{N}\left(\widehat{\mathcal{L}}_{\text{ent}} + \mathcal{L}_{\text{cross}}\right) + \widehat{\mathcal{L}}_{\text{ell}}^{(n)}\right],$$

where $\widehat{\mathcal{L}}_{\text{ell}}^{(n)} = \frac{1}{S}\sum_{k=1}^{K}\pi_k\sum_{i=1}^{S}\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}^{(k,i)},\boldsymbol{\phi})$, which enables us to apply stochastic optimization techniques such as stochastic gradients descend (SGD, Robbins and Monro, 1951) or ADADELTA (Zeiler, 2012). The complexity of the resulting algorithm is independent of N and dominated by algebraic operations that are $\mathcal{O}(M^3)$ in time, where M is the number of inducing points per latent process. This makes our automated variational inference framework practical for very large datasets.

5.2. Reducing the variance of the gradients with control variates

Theorem 1 is fundamental to having a statistical efficient algorithm that only requires sampling from univariate Gaussian distributions (instead of sampling from very high-dimensional Gaussians) for the estimation of the expected log likelihood term and its gradients.

However, the variance of the gradient estimates may be too large for the algorithm to work in practice, and variance reduction techniques become necessary. Here we use the well-known technique of control variates (see e.g. Ross, 2006, §8.2), where a new gradient estimate is constructed so as to have the same expectation but lower variance than the original estimate. Our control variate is the so-called score function $h(\mathbf{f}_{n\cdot}) = \nabla_{\lambda_k}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)$ and full details are given in Appendix F.

5.3. Optimization of the inducing inputs

So far we have discussed optimization of variational parameters ($\boldsymbol{\lambda}$), i.e. the parameters of the approximate posterior ($\{\pi_k, \mathbf{m}_k, \mathbf{S}_k\}$); covariance hyperparameters ($\boldsymbol{\theta}$); and likelihood parameters ($\boldsymbol{\phi}$). However, as discussed by Titsias (2009), the inducing inputs $\{\mathbf{Z}_j\}$ can be seen as additional variational parameters and, therefore, their optimization should be somehow robust to overfitting. As described in the Experiments (Section 9.5), learning of the inducing inputs can improve performance, requiring a lower number of inducing variables than when these are fixed. This, of course, comes at an additional computational cost which can be significant when considering high-dimensional input spaces. As with the variational parameters, we study learning of the inducing inputs via gradient-based optimization, for which we use the gradients provided in Appendix E.

Early references in the machine learning community where a single variational objective is used for parameter inference (in addition to posterior estimation over latent variables) can be found in Hinton and van Camp (1993); MacKay (1995); Lappalainen and Miskin (2000). These methods are now known under the umbrella term of variational Bayes[2] and consider a prior and an approximate posterior for these parameters within the variational framework. As mentioned above, rather than a full variational Bayes approach, we provide point estimates of $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ and $\{\mathbf{Z}_j\}$, and our experiments in §9 confirm the efficacy of our approach. More specifically, we show in §9.5 that point-estimation of the inducing inputs $\{\mathbf{Z}_j\}$ using the variational objective can be significantly better than using heuristics such as k-means clustering.

6. Dense Posterior and Practical Distributions

In this section we consider the case when the inducing inputs are placed at the training points, i.e. $\mathbf{Z}_j = \mathbf{X}$ and consequently $M = N$. As mentioned in Section 4.1, we refer to this setting as dense to distinguish it from the case when $M < N$, for which the resulting models are usually called sparse approximations. It is important to realize that not all real datasets are very large and that in many cases the resulting time and memory complexity $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$ can be afforded. Besides the dense posterior case, we also study some particular variational distributions that make our framework more practical, especially in large-scale applications.

6.1. Dense Approximate Posterior

When our posterior is dense the only approximation made is the assumed variational distribution in Equation (6). We will show that, in this case, we recover the objective function in Nguyen and Bonilla (2014a) and that hyper-parameter learning is easier as the terms in the resulting objective function that depend on the hyperparameters do not involve MC estimates. Therefore, their analytical gradients can be used. Furthermore, in the following

2. We note that these methods were then referred to as "Ensemble Learning", since a full distribution (i.e. an ensemble of parameters) was used instead of a single point estimate. Fortunately, nowadays, variational Bayes is a preferred term since Ensemble Learning is more commonly associated with methods for combining multiple models.

section, we will provide an efficient exact parametrization of the posterior covariance that reduces the $\mathcal{O}(N^2)$ memory complexity to a linear complexity $\mathcal{O}(N)$.

We start by looking at the components of $\mathcal{L}_{\text{elbo}}$ when we make $\mathbf{Z}_j = \mathbf{X}$, which we can simply obtain by making $\mathbf{K}^j_{\mathbf{zz}} = \mathbf{K}^j_{\mathbf{xx}}$ and $M = N$, and realizing that the posterior parameters $\{\mathbf{m}_{kj}, \mathbf{S}_{kj}\}$ are now N-dimensional objects. Therefore, we leave the entropy term $\widehat{\mathcal{L}}_{\text{ent}}$ in Equation (14) unchanged and we replace all the appearances of $\mathbf{K}^j_{\mathbf{zz}}$ with $\mathbf{K}^j_{\mathbf{xx}}$ and all the appearances of M with N for the $\mathcal{L}_{\text{cross}}$. We refer the reader to Appendix G for details of the equations but it is easy to see that the resulting $\widehat{\mathcal{L}}_{\text{ent}}$ and $\mathcal{L}_{\text{cross}}$ are identical to those obtained by Nguyen and Bonilla (2014a, Equations 5 and 6).

For the expected log likelihood term, the generic expression in Equation (16) still applies but we need to figure out the resulting expressions for the approximate posterior parameters in Equations (18) and (19). It is easy to show that the resulting posterior means and covariances are in fact $\mathbf{b}_{kj} = \mathbf{m}_{kj}$ and $\boldsymbol{\Sigma}_{kj} = \mathbf{S}_{kj}$ (see Appendix G for details). This means that in the dense case we simply estimate the $\widehat{\mathcal{L}}_{\text{ell}}$ by using empirical expectations over the unconstrained variational posterior $q(\mathbf{f}|\boldsymbol{\lambda})$, with 'free' mean and covariance parameters. In contrast, in the sparse case, although these expectations are still computed over $q(\mathbf{f}|\boldsymbol{\lambda})$, the parameters of the variational posterior $q(\mathbf{f}|\boldsymbol{\lambda})$ are constrained by Equations (18) and (19) which are functions of the prior covariance and the parameters of the variational distribution over the inducing variables $q(\mathbf{u}|\boldsymbol{\lambda})$. As we shall see in the following section, this distinction between the dense case and sparse case has critical consequences on hyperparameter learning.

6.1.1. Exact Hyperparameter Optimization

The above insight reveals a remarkable property of the model in the dense case. Unlike the sparse case, the expected log likelihood term does not depend on the covariance hyperparameters, as the expectation of the conditional likelihood is taken over the variational distribution $q(\mathbf{f}|\boldsymbol{\lambda})$ with 'free' parameters. Therefore, only the cross-entropy term $\mathcal{L}_{\text{cross}}$ depends on the hyperparameters (as we also know that $\nabla_{\boldsymbol{\theta}}\widehat{\mathcal{L}}_{\text{ent}} = 0$). For this term, as seen in Appendix G.1 and corresponding gradients in Equation (37), we have derived the exact (analytical) expressions for the objective function and its gradients, avoiding empirical MC estimates altogether. This has a significant practical implication: despite using black-box inference, the hyperparameters are optimized wrt the true evidence lower bound (given fixed variational parameters). This is an additional and crucial advantage of our automated inference method over other generic inference techniques (see e.g. Ranganath et al., 2014), which do not exploit knowledge of the prior.

6.1.2. Exact Solution with Gaussian Likelihoods

Another interesting property of our approach arises from the fact that, as we are using MC estimates, $\widehat{\mathcal{L}}_{\text{ell}}$ is an unbiased estimator of $\mathcal{L}_{\text{ell}}$. This means that, as the number of samples S increases, $\widehat{\mathcal{L}}_{\text{elbo}}$ will converge to the true value $\mathcal{L}_{\text{elbo}}$. In the case of a Gaussian likelihood, the posterior over the latent functions is also Gaussian and a variational approach with a full Gaussian posterior will converge to the true parameters of the posterior, see e.g. Opper and Archambeau (2009) and Appendix G.2 for details.

6.2. Practical Variational Distributions

As we have seen in Section 5, learning of all parameters in the model can be done in a scalable way through stochastic optimization for general likelihood models, providing automated variational inference for models with Gaussian process priors. However, the general MoG approximate posterior in Equation (6) requires $\mathcal{O}(M^2)$ variational parameters for each covariance matrix of the corresponding latent process, yielding a total requirement of $\mathcal{O}(QKM^2)$ parameters. This may cause difficulties for learning when these parameters are optimized simultaneously. In this section we introduce two special members of the assumed variational posterior family that improve the practical tractability of our inference framework. These members are a full Gaussian posterior and a mixture of diagonal Gaussians posterior.

6.2.1. Full Gaussian Posterior

This instance considers the case of only one component in the mixture ($K = 1$) in Equation (6),which has a Gaussian distribution with a *full covariance* matrix for each latent process. Therefore, following the factorization assumption in the posterior across latent processes, the posterior distribution over the inducing variables $\mathbf{u}$, and consequently over the latent functions $\mathbf{f}$, is a Gaussian with block diagonal covariance, where each block is a full covariance corresponding to that of a single latent function. We thus refer to this approximate posterior as the full Gaussian posterior (FG).

6.2.2. Mixture of Diagonal Gaussians Posterior

Our second practical variational posterior considers a mixture distribution as in Equation (6), constraining each of the mixture components for each latent process to be a Gaussian distribution with *diagonal covariance* matrix. Therefore, following the factorization assumption in the posterior across latent processes, the posterior distribution over the inducing variables $\mathbf{u}$ is a mixture of diagonal Gaussians. However, we note that, as seen in Equations (18) and (19), the posterior over the latent functions $\mathbf{f}$ is not a mixture of diagonal Gaussians in the general sparse case. Obviously, in the dense case (where $\mathbf{Z}_j = \mathbf{X}$) the posterior covariance over $\mathbf{f}$ of each component does have a diagonal structure. Henceforth, we will refer to this approximation simply as MoG, to distinguish it from the FG case above, while avoiding the use of additional notation. One immediate benefit of using this approximate posterior is computational, as we avoid the inversion of a full covariance for each component in the mixture.

As we shall see in the following sections, there are additional benefits from the assumed practical distributions and they concern the efficient parametrization of the covariance for both distributions and the lower variance of the gradient estimates for the MoG posterior.

6.3. Efficient Re-parametrization

As mentioned above, one of the main motivations for having specific practical distributions is to reduce the computational overhead due to the large number of parameters to optimize. For the MoG approximation, it is obvious that only $\mathcal{O}(M)$ parameters for each latent process and mixture component are required to represent the posterior covariance,

hence one obtains an efficient parametrization by definition. However, for the full Gaussian (FG) approximation, naively, one would require $\mathcal{O}(M^2)$ parameters. The following theorem states that for settings that require a large number of inducing variables, the FG variational distribution can be represented using a significantly lower number of parameters.

Theorem 2 *The optimal full Gaussian variational posterior can be represented using a parametrization that is linear in the number of observations (N).*

Before proceeding with the analysis of this theorem, we remind the reader that the general form of our variational distribution in Equation (6) requires $\mathcal{O}(KQM^2)$ parameters for the covariance, for a model with K mixture components, Q latent processes and M inducing variables. Nevertheless, for simplicity and because usually $K \ll M$ and $Q \ll M$, we will omit K and Q in in the following discussion.

The proof the theorem can be found in Appendix H, where it is shown that in the FG case the optimal solution for the posterior covariance is given by:

$$\widehat{\mathbf{S}_j} = \mathbf{K}_{\mathbf{zz}}^j \left(\mathbf{K}_{\mathbf{zz}}^j + \mathbf{K}_{\mathbf{zx}}^j \mathbf{\Lambda}_j \mathbf{K}_{\mathbf{xz}}^j\right)^{-1} \mathbf{K}_{\mathbf{zz}}^j, \tag{25}$$

where $\mathbf{\Lambda}_j$ is a N-dimensional diagonal matrix. Since the optimal covariance can be expressed in terms of fixed kernel computations and a free set of parameters given by $\mathbf{\Lambda}_j$, only N parameters are necessary to represent the posterior covariance. As we shall see below, this parametrization becomes useful for denser models, i.e. for models that have a large number of inducing variables.

6.3.1. Sparse Posterior

In the sparse case the inducing inputs $\mathbf{Z}_j$ are at arbitrary locations and, more importantly, $M \ll N$, the number of inducing variables is considerably smaller than the number of training points. The result in Equation (25) implies that if we parameterize $\mathbf{S}_j$ in that way, we will need $\mathcal{O}(N)$ parameters instead of $\mathcal{O}(M^2)$. Of course this is useful when roughly $N < \frac{M^2}{2}$. A natural question arises when we define the number of inducing points as a fraction of the number of training points, i.e. $M = \epsilon N$ with $0 \leq \epsilon \leq 1$, when is such a parameterization useful? In this case, using the alternative parametrization will become beneficial when $\epsilon > \sqrt{\frac{2}{N}}$. To illustrate this, consider for example the MNIST dataset used in our medium-scale experiments in Section 9.4 where $N = 60,000$. This yields a beneficial regime around roughly $\epsilon > 0.006$, which is a realistic setting. In fact, in our experiments on this dataset we did consider sparsity factors of this magnitude. For example, our biggest experiment used $\epsilon = 0.04$. With a naive parametrization of the posterior covariance, we would need roughly 2×10^6 parameters. In contrast, by using the efficient parametrization we only need 50×10^3 parameters. As shown below, these gains are greater as the model becomes denser, yielding a dramatic reduction in the number of parameters when having a fully dense Gaussian posterior.

6.3.2. Dense Posterior

In the dense case we have that $\mathbf{Z}_j = \mathbf{X}, \forall j = 1, \ldots, Q$ and consequently $M = N$. Therefore we have that the optimal posterior covariance is given by:

$$\widehat{\mathbf{S}_j} = \left((\mathbf{K}_{\mathbf{xx}}^j)^{-1} + \mathbf{\Lambda}_j\right)^{-1}.$$

In principle, the parametrization of the posterior covariance would require $\mathcal{O}(N^2)$ parameters for each latent process. However, the above result shows that we can parametrize these covariances efficiently using only $\mathcal{O}(N)$ parameters.

We note that, for models with $Q = 1$, this efficient re-parametrization has been used by Sheth et al. (2015) in the sparse case and Opper and Archambeau (2009) in the dense case, while adopting an inference algorithm different to ours.

6.4. Automatic Variance Reduction with a Mixture-of-Diagonals Posterior

An additional benefit of having a mixture-of-diagonals (MoG) posterior in the dense case is that optimization of the variational parameters will typically converge faster when using a mixture of diagonal Gaussians. This is an immediate consequence of the following theorem.

Theorem 3 *When having a dense posterior, the estimator of the gradients wrt the variational parameters using the mixture of diagonal Gaussians has a lower variance than the full Gaussian posterior's.*

The proof is in Appendix I and is simply a manifestation of the Rao-Blackwellization technique (Casella and Robert, 1996). The theorem is only made possible due to the analytical tractability of the KL-divergence term ($\mathcal{L}_{\text{kl}}$) in the variational objective ($\mathcal{L}_{\text{elbo}}$). The practical consequence of this theorem is that optimization will typically converge faster when using a mixture-of-diagonals Gaussians than when using a full Gaussian posterior approximation.

7. Predictions

Given the general posterior over the inducing variables $q(\mathbf{u}|\boldsymbol{\lambda})$ in equation (6), the predictive distribution for a new test point $\mathbf{x}_\star$ is given by:

$$\begin{aligned} p(\mathbf{y}_\star|\mathbf{x}_\star) &= \sum_{k=1}^{K} \pi_k \int p(\mathbf{y}_\star|\mathbf{f}_{\star\cdot}) \int p(\mathbf{f}_{\star\cdot}|\mathbf{u}) q_k(\mathbf{u}|\boldsymbol{\lambda}_k) \mathrm{d}\mathbf{u}\mathrm{d}\mathbf{f}_{\star\cdot}. \\ &= \sum_{k=1}^{K} \pi_k \int p(\mathbf{y}_\star|\mathbf{f}_{\star\cdot}) q_k(\mathbf{f}_{\star\cdot}|\boldsymbol{\lambda}_k) \mathrm{d}\mathbf{f}_{\star\cdot}, \end{aligned} \tag{26}$$

where $q_k(\mathbf{f}_{\star\cdot}|\boldsymbol{\lambda}_k)$ is the predictive distribution over the Q latent functions corresponding to mixture component k given the learned variational parameters $\boldsymbol{\lambda} = \{\mathbf{m}_k, \mathbf{S}_k\}$:

$$\begin{aligned} q_k(\mathbf{f}_{\star\cdot}|\boldsymbol{\lambda}_k) &= \prod_{j=1}^{Q} \mathcal{N}(f_{\star j}; \mu^\star_{kj}, \sigma^{\star 2}_{kj}), \text{ with} \\ \mu^\star_{kj} &= \kappa_j(\mathbf{x}_\star, \mathbf{Z}_j)(\mathbf{K}^j_{\mathbf{zz}})^{-1}\mathbf{m}_{kj}, \text{ and} \\ \sigma^{\star 2}_{kj} &= \kappa_j(\mathbf{x}_\star, \mathbf{x}_\star) - \kappa_j(\mathbf{x}_\star, \mathbf{Z}_j)\left((\mathbf{K}^j_{\mathbf{zz}})^{-1} - (\mathbf{K}^j_{\mathbf{zz}})^{-1}\mathbf{S}_{kj}(\mathbf{K}^j_{\mathbf{zz}})^{-1}\right)\kappa_j(\mathbf{Z}_j, \mathbf{x}_\star). \end{aligned}$$

Thus, the probability of the test points taking values $\mathbf{y}_\star$ (e.g. in classification) in Equation (26) can be readily estimated via Monte Carlo sampling.

8. Complexity analysis

Throughout this paper we have only considered computational complexity with respect to the number of inducing points and training points for simplicity. Although this has also been common practice in previous work (see e.g. Dezfouli and Bonilla, 2015; Nguyen and Bonilla, 2014a; Hensman et al., 2013; Titsias, 2009), we believe it is necessary to provide an in-depth complexity analysis of the computational cost of our model. Here we analyze the computational cost of evaluating the $\mathcal{L}_{\text{elbo}}$ and its gradients once.

We begin by reminding the reader of the dimensionality notation used so far and by introducing additional notation specific to this section. We analyze the more general case of stochastic optimization using mini-batches. Let K be the number of mixture components in our variational posterior; Q the number of latent functions; D the dimensionality of the input data; S the number of samples used to estimate the required expectations via Monte Carlo; B the number of inputs used per mini-batch; M the number of inducing inputs; and N the number of observations. We note that in the case of batch optimization $B = N$, hence our analysis applies to both the stochastic and batch setting. We also let $T(e)$ represent the computational costs of expression e. Furthermore, we assume that the kernel function is simple enough such that evaluating its output between two points is $\mathcal{O}(D)$ and we denote with $T(\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi})) \in \mathcal{O}(L)$ the cost of a single evaluation of the conditional likelihood.

8.1. Overall Complexity

While the detailed derivations of the computational complexity are in Appendix J, here we state the main results. Assuming that $K \ll M$, the total computational cost is given by:

$$\begin{aligned} T(\mathcal{L}_{\text{elbo}}) &\in \mathcal{O}(Q(M^2D + BMD + K^2M^3 + KBM^2 + KBSL)), \\ &\text{and for diagonal posterior covariances we have:} \\ T(\mathcal{L}_{\text{elbo}}) &\in \mathcal{O}(Q(M^2D + BMD + M^3 + KBM + KBSL)). \end{aligned}$$

We note that it takes $\lceil N/B \rceil$ gradient updates to go over an entire pass of the data. If we assume that K is a small constant, the asymptotic complexity of a single pass over the data does not improve by increasing B beyond $B = M$. Hence if we assume that $B \in \mathcal{O}(M)$ the computational cost of a single gradient update is given by

$$T(\mathcal{L}_{\text{elbo}}) \in \mathcal{O}(QM(MD + M^2 + SL)),$$

for both diagonal and full covariances. If we assume that it takes a constant amount of time to compute the likelihood function between a sample and an output, we see that setting $S \in \mathcal{O}(MD + M^2)$ does not increase the computational complexity of our algorithm. As we shall see in section 9.4, even a complex likelihood function only requires $10,000$ samples to approximate it. Since we expect to have more than 100 inducing points in most large scale settings, it is reasonable to assume that the overhead from generated samples will not be significant in most cases.

9. Experiments

In this section we analyze the behavior of our model on a wide range of experiments involving small-scale to large-scale datasets. The main aim of the experiments is to evaluate the

performance of the model when considering different likelihoods and different dataset sizes. We analyze how our algorithm's performance is affected by the density and location of the inducing variables, and how the performance of batch and stochastic optimization compare.

We start by evaluating our algorithm using five small-scale datasets ($N < 1,300$) under different likelihood models and number of inducing variables (§9.3). Then, using medium-scale experiments ($N < 70,000$), we compare stochastic and batch optimization settings, and determine the effect of learning the inducing inputs on the performance (§9.4). Subsequently, we use SAVIGP on two large-scale datasets. The first one involves the prediction of airline delays, where we compare the convergence properties of our algorithm to models that leverage full knowledge of the likelihood (§9.7.1). The second large dataset considers an augmented version of the popular MNIST dataset for handwritten digit recognition, involving more than 8 million observations (§9.7.3). Finally, we showcase our algorithm on a non-standard inference problem concerning a seismic inversion task, where we show that our variational inference algorithm can yield solutions that closely match (non-scalable) sampling approaches (§9.8). Before proceeding with the experimental set-up, we give details of our implementation which uses GPUs.

9.1. Implementation

We have implemented our SAVIGP method in Python and all the code is publicly available at `https://github.com/Karl-Krauth/Sparse-GP`. Most current mainstream implementations of Gaussian process models do not support GPU computation, and instead opt to offload most of the work to the CPU, with the notable exception of Matthews et al. (2017). For example, neither of the popular packages GPML[3] or GPY[4] provide support for GPUs. This is despite the fact that matrix manipulation operations, which are easily parallelizable, are at the core of any Gaussian process model. In fact, the rate of progress between subsequent GPU models has been much larger than for CPUs, thus ensuring that any GPU implementation would run at an accelerated rate as faster hardware gets released.

With these advantages in mind, we provide an implementation of SAVIGP that uses Theano (Al-Rfou et al., 2016), a library that allows users to define symbolic mathematical expressions that get compiled to highly optimized GPU CUDA code. Any operation that involved the manipulation of large matrices or vectors was done in Theano. Most of our experiments were either run on g2.2 AWS instances, or on a desktop machine with an Intel core i5-4460 CPU, 8GB of RAM, and a GTX760 GPU.

Despite using a low-end outdated GPU, we found a time speed-up of 5x on average when we offloaded work to the GPU. For example, in the case of the MNIST-B dataset (used in section 9.4), we averaged the time it took to compute ten gradient evaluations of the $\mathcal{L}_{\text{elbo}}$ with respect to the posterior parameters over the entire training set, where we expressed the posterior as a full Gaussian and used a sparsity factor of 0.04. While it took 42.35 seconds, on average, per gradient computation when making use of the CPU only, it took a mere 8.52 seconds when work was offloaded to the GPU. We expect the difference to be even greater given a high-end current-generation GPU.

3. Available at `http://www.gaussianprocess.org/gpml/code/matlab/doc/`.
4. Available at `https://github.com/SheffieldML/GPy`.

9.2. Details of the Experiments

Here we give details of our experimental evaluation concerning performance measures, inducing input learning, optimization methods, how to read the figures reported and the different model configurations explored.

9.2.1. Performance Measures

We evaluated the performance of the algorithm using non-probabilistic and probabilistic measures according to the type of learning problem we are addressing. The standardized squared error (SSE) and the negative log predictive density (NLPD) were used in the case of continuous-output problems. The error rate (ER) and the negative log probability (NLP) were used in the case of discrete-output problems. In the experiments using the airline dataset, we used the root mean squared error (RMSE) instead of the SSE to be able to compare our method with previous work that used RMSE for performance evaluation.

9.2.2. Experimental Settings

In small-scale experiments, inducing inputs were placed on a subset of the training data in a nested fashion, so that experiments on less sparse models contained the inducing points of the sparser models. In medium-scale and large-scale experiments the location of the inducing points was initialized using the k-means clustering method. In all experiments the squared exponential covariance function was used.

9.2.3. Optimization Methods

Learning the model involves optimizing variational parameters, hyperparameters, likelihood parameters, and inducing inputs. These were optimized iteratively in a global loop. In every iteration, each set of parameters was optimized separately while keeping the other sets of parameters fixed, and this process was continued until the change in the objective function between two successive iterations was less than 10^{-6}. For optimization in the batch settings, each set of parameters was optimized using L-BFGS, with the maximum number of global iterations limited to 200. In the case of stochastic optimization, we used the ADADELTA method (Zeiler, 2012) with parameters $\epsilon = 10^{-6}$ and a decay rate of 0.95. The choice of this optimization algorithm was motivated by (and to be consistent with) the work of Hensman et al. (2013), who found this specific algorithm successful in the context of Gaussian process regression. We compare with the method of Hensman et al. (2013) in this context in §9.7.

9.2.4. Reading the Graphs

Results are presented using boxplots and bar/line charts. In the case of boxplots, the lower, middle, and upper hinges correspond to the 25th, 50th, and 75th percentile of the data. The upper whisker extends to the highest value that is within 1.5×IQR of the top of the box, and the lower whisker extends to the lowest value within 1.5×IQR of the bottom of the box (IQR is the distance between the first and third quantile). In the case of bar charts, the error bars represent 95% confidence intervals computed over multiple partitions of the dataset.

Dataset	N_{train}	N_{test}	D	Likelihood $p(y\|f)$	Model
MINING	811	0	1	$\lambda^y \exp(-\lambda)/y!$	Log Gaussian Cox process
BOSTON	300	206	13	$\mathcal{N}(y; f, \sigma^2)$	Standard regression
CREEP	800	1266	30	$\nabla_y t(y)\mathcal{N}(t(y); f, \sigma^2)$	Warped Gaussian processes
ABALONE	1000	3177	8	$\nabla_y t(y)\mathcal{N}(t(y); f, \sigma^2)$	Warped Gaussian processes
CANCER	300	383	9	$1/(1+\exp(-f))$	Binary classification
USPS	1233	1232	256	$\exp(f_c)/\sum_{i=1}\exp(f_i)$	Multi-class classification

Table 1: Details of the datasets used in the small-scale experiments and their corresponding likelihood models. N_{train}, N_{test}, D are the number of training points, test points and input dimensions respectively; 'likelihood' is the conditional likelihood used on each problem; and 'model' is the name of the model associated with that likelihood. For MINING $N_{test} = 0$ as only posterior inference is done (predictions on a test set are not made for this problem).

9.2.5. Model Configurations

We refer to the ratio of inducing points to training points as the sparsity factor (SF $= M/N$). For each sparsity factor, three different variations of SAVIGP corresponding to (i) a full Gaussian posterior, (ii) a diagonal Gaussian posterior, and (iii) a mixture of two diagonal Gaussian posteriors were tested, which are denoted respectively by FG, MoG1, and MoG2 in the graphs.

9.3. Small-scale Experiments

We tested SAVIGP on six small-scale datasets with different likelihood models. The datasets are summarized in Table 9.3, and are the same as those used by Nguyen and Bonilla (2014a). For each dataset, the model was tested five times across different subsets of the data; except for the mining dataset where only training data were used for evaluation of the posterior distribution.

9.3.1. Standard Regression

The model was evaluated on the BOSTON dataset (Bache and Lichman, 2013), which involves a standard regression problem with a univariate Gaussian likelihood model, i.e., $p(y_n|f_n) = \mathcal{N}(y_n|f_n, \sigma^2)$. Figure 1 shows the performance of SAVIGP for different sparsity factors, as well as the performance of exact Gaussian process inference (GP). As we can see, SSE increases slightly on sparser models. However, the SSE of all the models (FG, MoG1, MoG2) across all sparsity factors are comparable to the performance of exact inference (GP). In terms of NLPD, as expected, the dense (SF $= 1$) FG model performs exactly like the exact inference method (GP). In the sparse models, NLPD shows less variation in lower sparsity factors (especially for MoG1 and MoG2), which can be attributed to the tendency of such models to make less confident predictions under high sparsity settings.

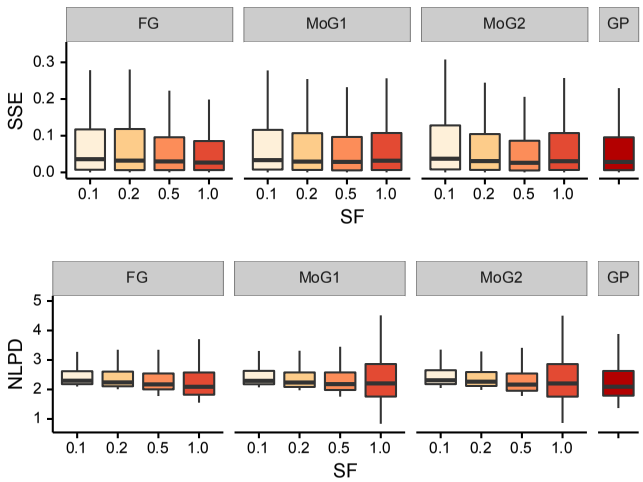


Figure 1: The distributions of SSE and NLPD for a regression problem with a univariate Gaussian likelihood model on the BOSTON housing dataset. Three approximate posteriors in SAVIGP are used: FG (full Gaussian), MoG1 (diagonal Gaussian), and MoG2 (mixture of two diagonal Gaussians), along with various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model, with SF $= 1$ corresponding to the dense model. GP corresponds to the performance of exact inference using standard Gaussian process regression.

9.3.2. Warped Gaussian Process

In warped Gaussian processes, the likelihood function is $p(y_n|f_n) = \nabla_y t(y_n)\mathcal{N}(t(y_n); f_n, \sigma^2)$, for some transformation t. We used the same neural-net style transformation as Snelson et al. (2003), and evaluated the performance of our model on two datasets: CREEP (Cole et al., 2000), and ABALONE (Bache and Lichman, 2013). The results are compared with the performance of exact inference for warped Gaussian processes (WGP) described by Snelson et al. (2003), and also with the performance of exact Gaussian process inference with a univariate Gaussian likelihood model (GP). As shown in Figure 2, in the case of the ABALONE dataset, the performance is similar across all the models (FG, MoG1 and MoG2) and sparsity factors, and is comparable to the performance of the exact inference method for warped Gaussian processes (WGP). The results on CREEP are given in Appendix K.1.

9.3.3. Binary Classification

For binary classification we used the logistic likelihood $p(y_n = 1|f_n) = 1/(1 + e^{-f_n})$ on the breast CANCER dataset (Bache and Lichman, 2013) and compared our model against the

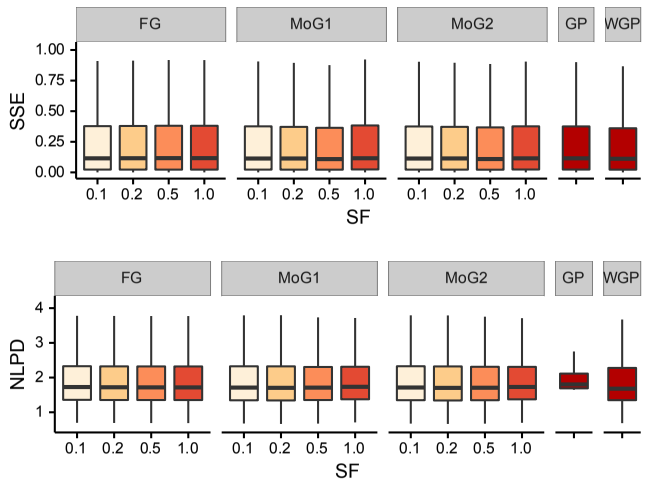


Figure 2: The distributions of SSE and NLPD for a warped Gaussian process likelihood model on the ABALONE dataset. Three approximate posteriors in SAVIGP are used: FG (full Gaussian), MoG1 (diagonal Gaussian), and MoG2 (mixture of two diagonal Gaussians), along with various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model, with SF $= 1$ corresponding to the non-sparse model. WGP corresponds to the performance of the exact inference method for warped Gaussian process models (Snelson et al., 2003), and GP is the performance of exact inference on a univariate Gaussian likelihood model.

expectation propagation (EP) and variational bounds (VBO) methods described by Nickisch and Rasmussen (2008). Results depicted in Figure 3 indicate that the error rate remains comparable across all models (FG, MoG1, MoG2) and sparsity factors, and is almost identical to the error rates obtained by inference using EP and VBO. Interestingly, the NLP shows more variation and generally degrades as the number of inducing points is increased, especially for MoG1 and MoG2 models, which can be attributed to the fact that these denser models are overly confident in their predictions.

9.3.4. Multi-class Classification

For multi-class classification, we used the softmax likelihood $p(y_n = c) = e^{-f_c}/\sum_i e^{-f_i}$, and trained the model to classify the digits 4, 7, and 9 from the USPS dataset (Rasmussen and Williams, 2006). We compared our model against a variational inference method (VQ) which represents the ELBO using a quadratic lower bound on the likelihood terms (Khan et al., 2012). As we see in Figure 4, the error rates are slightly lower in denser FG models.

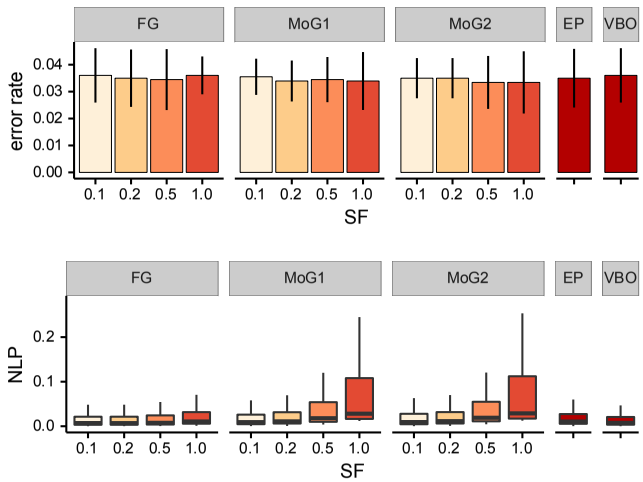


Figure 3: Error rates and NLP for binary classification using a logistic likelihood model on the Wisconsin breast CANCER dataset. Three approximate posteriors are used: FG (full Gaussian), MoG1 (diagonal Gaussian), and MoG2 (mixture of two diagonal Gaussians), along with various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model, with SF $= 1$ corresponding to the original model without sparsity. The performance of inference using expectation propagation and variational bounds are denoted by EP and VBO respectively.

We also note that all versions of SAVIGP achieve comparable error rates to VQ's. Similarly to the binary classification case, NLP shows higher variation with higher sparsity factor, especially in MoG1 and MoG2 models.

9.3.5. Log Gaussian Cox Process

The log Gaussian Cox process (LGCP) is an inhomogeneous Poisson process in which the log-intensity function is a shifted draw from a Gaussian process. Following Murray et al. (2010), we used the likelihood $p(y_n|f_n) = \frac{\lambda_n^{y_n}\exp(-\lambda_n)}{y_n!}$, where $\lambda_n = \exp f_n + m$ is the mean of a Poisson distribution and m is the offset of the log mean. We applied SAVIGP with the LGCP likelihood on a coal-mining disaster dataset (Jarrett, 1979), which can be seen in Figure 5 (top).

As baseline comparisons, we use hybrid Monte Carlo (HMC) and elliptical slice sampling (ESS) described by Duane et al. (1987) and Murray et al. (2010) respectively. We collected every 100th sample for a total of $10k$ samples after a burn-in period of $5k$ samples and used

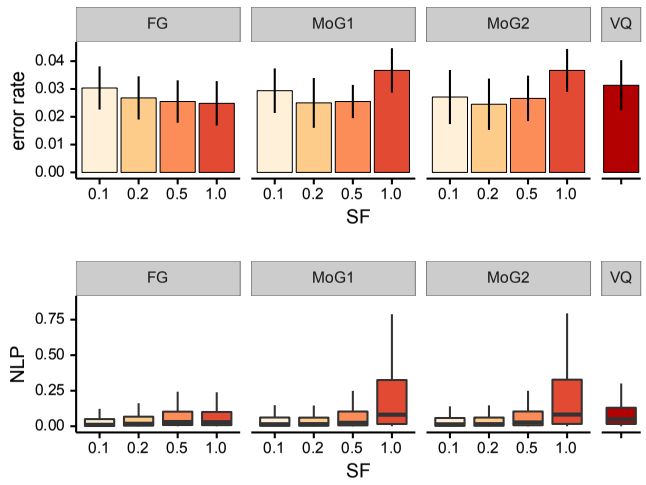


Figure 4: Classification error rates and NLP for the multi-class classification problem using a softmax likelihood model on the USPS dataset. Three approximate posteriors in SAVIGP are used: FG (full Gaussian), MoG1 (diagonal Gaussian), and MoG2 (mixture of two diagonal Gaussians), along with various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model, with SF $= 1$ corresponding to the original model without sparsity. VQ corresponds to a variational inference method, which represents the ELBO as a quadratic lower bound to the likelihood terms.

the Gelman-Rubin potential scale reduction factors (Gelman and Rubin, 1992) to check for convergence.

The bottom plot of Figure 5 shows the mean and variance of the predictions made by SAVIGP, HMC and ESS. We see that the FG model provides similar results across all sparsity factors, which is also comparable to the results provided by HMC and ESS. MoG1 and MoG2 models provide the same mean as the FG models, but tend to underestimate the posterior variance. This under-estimation of posterior variance is well known for variational methods, especially under factorized posteriors. This results are more significant when comparing the running times across models. When using a slower Matlab implementation of our model, for a fair comparison across all methods, SAVIGP was at least two orders of magnitude faster than HMC and one order of magnitude faster than ESS.

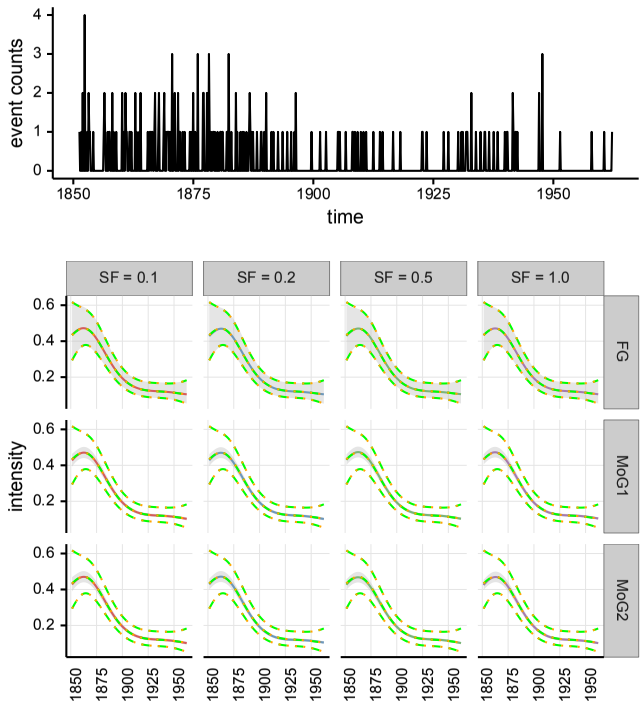


Figure 5: Top: the coal-mining disasters data. Bottom: the posteriors for a Log Gaussian Cox process on the data when using a FG (full Gaussian), MoG1 (diagonal Gaussian), and MoG2 (mixture of two diagonal Gaussians), along with various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model, with SF $= 1$ corresponding to non-sparse model. The solid line is the posterior mean and the shaded area represents 95% confidence intervals. HMC and ESS correspond to hybrid Monte Carlo and elliptical slice sampling inference methods and are represented by orange dots and dashed green lines respectively.

9.4. Medium-scale Experiments

In this section we investigate the performance of SAVIGP on four medium-scale problems on the datasets summarized in Table 9.4. Our goal here is to evaluate the model on medium-scale datasets, to study the effect of optimizing the inducing inputs, and to compare the performance of batch and stochastic optimization of the parameters. The first problem that

Dataset	N_{train}	N_{test}	D	Q	P	Model
MNIST-B	60,000	10,000	784	1	1	Binary classification
MNIST	60,000	10,000	784	10	10	Multi-class classification
SARCOS-2	44,484	4,449	21	3	2	GPRN (Wilson et al., 2012)
SARCOS	44,484	4,449	21	8	7	GPRN (Wilson et al., 2012)

Table 2: Datasets used on the medium-scale experiments. GPRN stands for Gaussian process regression networks; N_{train}, N_{test}, D are the number of training points, test points and input dimensions respectively; Q is the number of latent processes; P is the dimensionality of the output data; and 'model' is the model associated with the conditional likelihood used.

we consider is multi-class classification of handwriting digits on the MNIST dataset using a softmax likelihood model. The second problem uses the same dataset, but the task involves binary classification of odd and even digits using the logistic likelihood model. Therefore we refer to this dataset as the MNIST binary dataset (MNIST-B). The third problem uses the SARCOS dataset (Vijayakumar and Schaal, 2000) and concerns an inverse-dynamics problem for a seven degrees-of-freedom SARCOS anthropomorphic robot arm. The task is to map from a 21-dimensional input space (7 joint positions, 7 joint velocities, 7 joint accelerations) to the corresponding 7 joint torques. For this multi-output regression problem we use the Gaussian process regression network (GPRN) likelihood model of Wilson et al. (2012), which allows for nonlinear models where the correlation between the outputs can be spatially adaptive. This is achieved by taking a linear combination of latent Gaussian processes, where the weights are also drawn from Gaussian processes. Finally, SARCOS-2 is the same as SARCOS, but the model is learned using only data from joints 4 and 7. For both SARCOS and SARCOS-2, the mean of the SSE and NLPD across joints 4 and 7 are used for performance evaluation. We only consider joints 4 and 7 for SARCOS, despite the fact that predictions are made across all 7 joints to provide a direct comparison with SARCOS-2 and with previous literature (Nguyen and Bonilla, 2014b). We also made use of automatic relevance determination (ARD) for both the SARCOS and SARCOS-2 datasets.

9.4.1. Batch Optimization

We first evaluate the performance of batch optimization of model parameters on *multi-class classification* using MNIST and refer the reader to Appendix K.2 for the results on binary classification using MNIST-B. We optimized the kernel hyperparameters and the variational parameters, but fixed the inducing point locations using k-means clustering. Unlike most previous approaches (Ranzato et al., 2006; Jarrett et al., 2009), we did not tune model parameters using the validation set. Instead, we considered the validation set to be part of the training set and used our variational framework to learn all model parameters. As such, the current setting likely provides a lower performance on test accuracy compared to the approaches that use a validation dataset, however our goal is simply to show that we were able to achieve competitive performance in a sparse setting when no knowledge of the likelihood model is used.

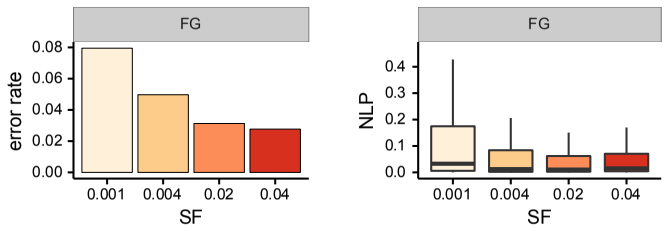


Figure 6: Error rates and NLP for multi-class classification on the MNIST dataset. We used a full Gaussian (FG) posterior approximation across various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model.

Figure 6 shows the result on the MNIST dataset. we see that the performance improves with denser models. Overall, SAVIGP achieves an error rate of 2.77% at SF = 0.04. This is a significant improvement over the results reported by Gal et al. (2014), in which a separate model was trained for each digit, which achieved an error rate of 5.95%. As a reference, previous literature reports about 12% error rate by linear classifiers and less than 1% error rate by state-of-the-art deep convolutional nets. Our results show that our method reduces the gap between GPs and deep nets while solving the harder problem of full posterior estimation. In Appendix K.2 we show that our model can achieve slightly better performance than that reported by Hensman et al. (2015) on MNIST-B.

Now we turn our attention to evaluating batch optimization when using *Gaussian process regression networks*. Figure 7 shows that SARCOS-2 gets a significant benefit from a higher number of inducing points, which is consistent with previous work that found that the performance on this dataset improves as more data is being used to train the model (Vijayakumar and Schaal, 2000). The performance is significantly better than the results reported by Nguyen and Bonilla (2014b), who achieved a mean standardized squared error (MSSE) of 0.2631 and 0.0127 across joints 4 and 7, against our values of 0.0033 and 0.0065 for SF = 0.04. However, we note that their setting was much sparser than ours on joint 4. The results on SARCOS (predicting on all joints) are given in Appendix K.3.

9.5. Inducing-input Learning

We now compare the effect of adaptively learning the inducing inputs, versus initializing them using k-means and leaving them fixed. We look at the performance of our model under two settings: (i) a low number of inducing variables (SF$= 0.001, 0.004$) where the inducing inputs are learned, and (ii) a large number of inducing variables (SF $= 0.02, 0.04$) without learning of their locations.

Figure 8 shows the performance of the model under the two settings on MNIST-B. We see that learning the location of the inducing variables yields a large gain in performance. In fact, the sparser models with inducing point learning performed similarly to the denser models, despite the fact that the two models differed by an order of magnitude when it came

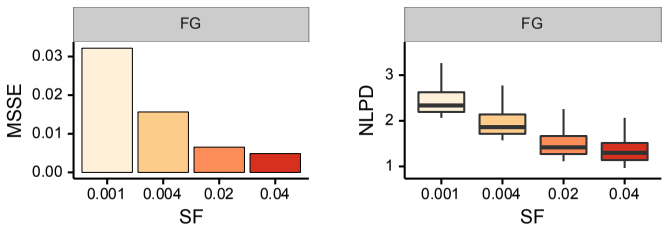


Figure 7: Mean SSE and NLPD for multi-output regression on the SARCOS-2 dataset. We used a full Gaussian (FG) posterior approximation across various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model.

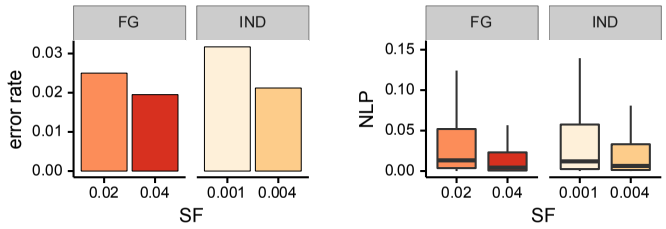


Figure 8: Comparison of error rate and NLP obtained by SAVIGP without learning (FG) and with learning (IND) of inducing inputs for binary classification on the MNIST-B dataset.

to the number of inducing variables. Additional results on MNIST, SARCOS and SARCOS-2 are shown in Appendix K.4. Our analyses indicate that there is a trade-off between the reduction in computational complexity gained by reducing the number of inducing variables and the increased computational cost of calculating inducing-input gradients. As such, the advantage of learning the inducing inputs is dataset dependent and it is affected mainly by the input dimensionality (D).

9.6. Batch Optimization vs Stochastic Optimization

In this section we compare the performance of our model after it has been trained in a batch setting versus a stochastic setting. We used ADADELTA as our stochastic optimization algorithm as it requires less hand-tuning than other algorithms such as vanilla stochastic gradient descend (SGD).

Table 9.6 shows only a slight deterioration in predictive performance on MNIST-B when using stochastic optimization instead of batch optimization. In fact, our exploratory experiments showed that the error metrics could have significantly been reduced by meticulously hand-tuning momentum stochastic gradient descent (SGD). However, our goal was simply to show that our model does not suffer from a large loss in performance when going from a batch to a stochastic setting in medium-scale experiments, without requiring extensive hand tuning of the optimization algorithm. As we shall see in the next section, when batch optimization is not feasible, stochastic optimization in our model performs well when compared to state-of-the-art approaches for inference in GP models on very large datasets.

Method	Batch SF = 0.001	Batch SF = 0.004	Stochastic SF = 0.001	Stochastic SF = 0.004
ER	3.17%	2.12%	3.11%	2.68%
NLP	0.097	0.068	0.099	0.083

Table 3: Error rate (ER) and negative log probability (NLP) obtained with SAVIGP optimized using batch optimization (with L-BFGS) and stochastic optimization (with ADADELTA) on the MNIST-B dataset. The inducing inputs are optimized in both cases.

9.7. Large-scale Experiments

In this section we evaluate SAVIGP on two large-scale problems involving prediction of airline delays (regression, $N = 700,000$) and handwritten digit recognition (classification, $N = 8,100,000$).

9.7.1. Airline Delays

Here we consider the problem of predicting airline delays using a univariate Gaussian likelihood model (Hensman et al., 2013). We note that this dataset has outliers that can significantly affect the performance of regression algorithms when using metrics such as the RMSE. This has also been pointed out by Das et al. (2015). Additionally, in order to match the application of the algorithms to a realistic setting, evaluation of learning algorithms on this dataset should always consider making predictions in the future. Therefore, unlike Hensman et al. (2013), we did not randomly select the training and test sets, instead we selected the first 700,000 data points starting at a given offset as the training set and the next 100,000 data points as the test set. We generated five training/test sets by setting the initial offset to 0 and increasing it by 200,000 each time.

We used the squared exponential covariance function with automatic relevance determination (ARD), and optimized all the parameters using ADADELTA (Zeiler, 2012). We compare the performance of our model with the stochastic variational inference on GPs method (SVIGP) described by Hensman et al. (2013), which assumes full knowledge of the likelihood function. Our method (SAVIGP) and SVIGP were optimized using ADADELTA with identical settings. We also report the results of Bayesian linear regression with a zero-mean unit-variance prior over the weights (LINEAR) and Gaussian process regression using subsets of the training data (GP1000 and GP2000 using 1,000 and 2,000 datapoints respectively).

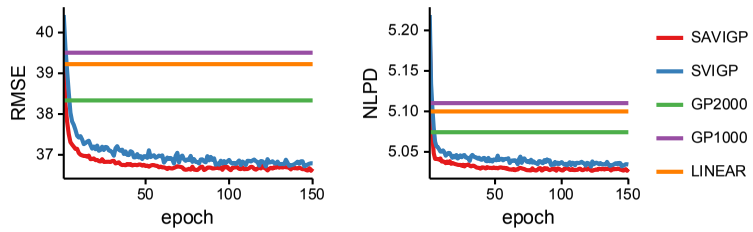


Figure 9: The RMSE (left) and NLPD (right) of SVIGP (Hensman et al., 2013), and SAVIGP (our method) averaged across all 5 airline-delay experiment runs. The x-axis represents the number of passes over the data (epochs). The figure also shows the performance of GP1000, GP2000 and Bayesian linear regression (LINEAR) after training is completed.

For each run we fit GP1000 and GP2000 ten times using a randomly selected subset of the training data. Kernel hyper-parameters of GP1000 and GP2000 were optimized using L-BFGS.

Figure 9 shows the performance of SAVIGP and the four baselines. We see that SAVIGP converges at a very similar rate to SVIGP, despite making no assumption about the likelihood model. Furthermore, SAVIGP performs better than all three simple baselines after less than 5 epochs. We note that our results are not directly comparable with those reported by Hensman et al. (2013), since we deal with the harder problem of predicting future events due to the way our dataset is selected.

9.7.2. Training Loss vs Test Performance

An additional interesting question about the behavior of our model relates to how well the training objective function correlates to the test error metrics. Figure 10 shows how the performance metrics (RMSE and NLPD) on the test data vary along with the negative evidence lower bound (NELBO). As we can see, changes in the NELBO closely mirror changes in RMSE and NLPD in our model, which is an indication that the variational objective generalizes well to test performance when using these metrics.

9.7.3. Large-scale Mnist

In this section we evaluate the performance of SAVIGP on the MNIST8M (Loosli et al., 2007) dataset, which artificially extends the MNIST dataset to 8.1 million training points by pseudo-randomly transforming existing MNIST images.

We train SAVIGP on the MNIST8M dataset by optimizing only variational parameters stochastically, with a batch size of 1000 and 2000 inducing points. We also use a squared exponential covariance function without automatic relevance determination.

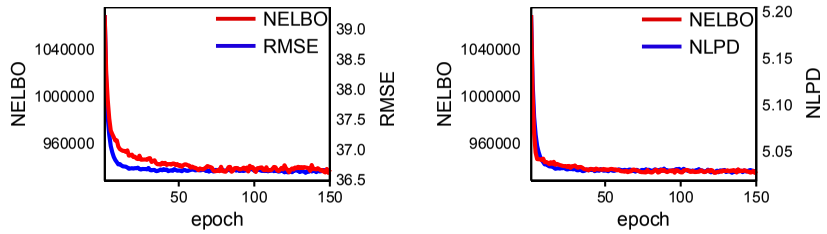


Figure 10: The RMSE (left) and NLPD (right) of SAVIGP on the test data alongside the negative evidence lower bound (NELBO) on the training data, averaged over five runs of the airline-delay experiment.

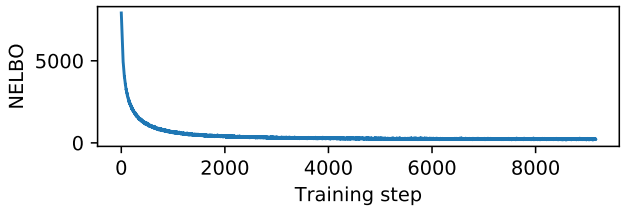


Figure 11: The negative evidence lower bound (NELBO) on the current batch of the MNIST8M training set over time on SAVIGP. The x-axis represents the number of training steps taken.

After optimizing the model for 19 hours, we observe an error rate of 1.54% on the test set and lower, middle and upper NLP quartiles of 8.61e−4, 3.81e−3, and 2.16e−2 respectively. We see that this outperforms significantly our previous result in §9.4 on standard MNIST, where we reported an error rate of 2.77%. As a point of comparison with hard-coded approaches, Henao and Winther (2012) reported 0.86% error rate on MNIST when considering an augmented active set (which is analogous to the concept of inducing inputs/variables) and a 9th-degree polynomial covariance function.

Finally, Figure 11 shows the training loss (the NELBO) of SAVIGP on MNIST8M as a function of the number of training steps, where we note that the loss decreases rapidly in the first 2,000 iterations and stabilizes at around 4,000 iterations.

9.8. Seismic Inversion

In this experiment we aim to evaluate our method qualitatively on a non-standard inference problem motivating the need for black-box likelihoods. This application considers a one-dimensional seismic inversion and was investigated by Bonilla et al. (2016). Given noisy surface observations of sound-reflection times, the goal is to infer the geometry (layer depths) of subsurface geological layers and seismic propagation velocities within each layer. For this, we consider a real dataset from a seismic survey carried out in the Otway basin region in the state of Victoria, Australia.

9.8.1. Setting

There are $N = 113$ site locations with $P = 4$ interface reflections (layers) per site. The inputs are given by the surface locations of the seismic sensors ($\mathbf{X}$), the outputs are the observed times at the different locations for each of the layers ($\mathbf{Y}$) and we aim to infer $Q = 2 * P$ latent functions, i.e. P functions corresponding to layer depths and P functions corresponding to seismic propagation velocities. For clarity, in this section we refer to the latent functions corresponding to depth and velocity as f^{d} and f^{v}, respectively.

9.8.2. Likelihood

The likelihood of the observed times y_{np} for location $\mathbf{x}_n$ and interface p is given by:

$$y_{np} = \begin{cases} 2\left(\frac{f^{\mathrm{d}}_{np}}{f^{\mathrm{v}}_{np}}\right) + \epsilon_{np}, & \text{for } p = 1, \\ 2\left(\frac{f^{\mathrm{d}}_{np} - f^{\mathrm{d}}_{np-1}}{f^{\mathrm{v}}_{np}}\right) + y_{np-1} + \epsilon_{np}, & \text{for } 1 < p \leq P, \end{cases}$$

where $\epsilon_{np} \sim \mathcal{N}(0, \sigma_p^2)$ and σ_p^2 is the output-dependent noise variance. As in Bonilla et al. (2016), we set the corresponding standard deviations to 0.025s, 0.05s, 0.075s and 0.1s.

9.8.3. Prior Setting

We used the same prior as in Bonilla et al. (2016), with prior mean depths of 200m, 500m, 1600m and 2200m and prior mean velocities of 1950m/s, 2300m/s, 2750m/s and 3650m/s. The corresponding standard deviations for the depths were set to 15% of the layer mean, and for the velocities they were set to 10% of the layer mean. A squared exponential covariance function with unit length-scale was used.

9.8.4. Posterior Estimation

We ran our algorithm for the dense case ($\mathbf{Z} = \mathbf{X}$) using a full Gaussian posterior and batch optimization, initializing the posterior means to the prior means and the posterior covariance to a diagonal matrix with entries corresponding to 0.1% of the prior variances. The results are given in Figure 12, where we see that SAVIGP's posterior closely match the "true" posterior obtained by the MCMC algorithm developed in Bonilla et al. (2016), although the variances are overestimated, which can be see as a consequence of our variational approach using a full Gaussian approximate posterior.

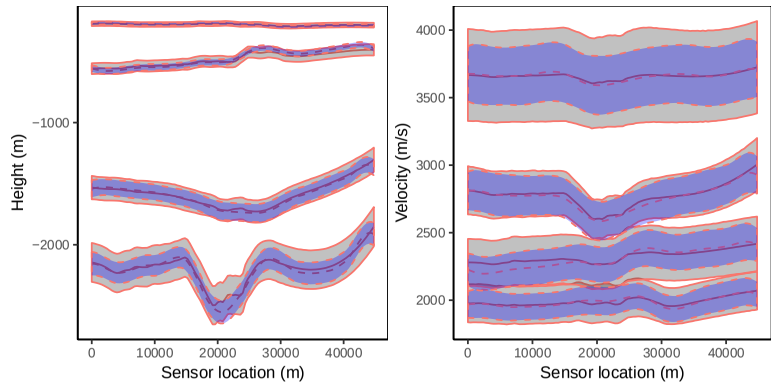


Figure 12: Results for the seismic inversion experiment using our algorithm (SAVIGP) and the MCMC algorithm developed by Bonilla et al. (2016). The mean and standard deviations envelopes are shown for SAVIGP and MCMC in solid and dashed lines, respectively. Left: inferred layer boundaries. Right: inferred seismic velocities.

9.9. The Role of the Number of Mixture Components (K), the Number of Latent Functions (Q), the Number of Samples (S) and the Mini-Batch Size (B)

As described in §4, our proposed variational inference method can be applied when using general mixtures of Gaussian distributions (MoGs), i.e. $\forall K \geq 1$ and without imposing constraints on the corresponding covariances of each component. The main result regarding statistical efficiency of our algorithm (as stated in Theorem 1) still holds in this general case. However, additional results such as those in Theorems 2 and 3 are specific to using $K = 1$ and a full Gaussian posterior or $K > 1$ and a mixture of diagonals posterior. Furthermore, in our experiments we have only reported results for MoG models with $K = 1, 2$ and diagonal covariances. In these cases we found that they can provide good point-predictive performance but underestimate the predictive variances. Additional experiments with more components not reported here did not yield significant improvements and in some cases the best result obtained after optimization of the ELBO converged to a single Gaussian posterior. There are obvious computational benefits when considering these approximate posteriors and further investigation for more complex settings such as those in Damianou and Lawrence (2013) can provide support for more flexible distributions.

With regards to the number of latent functions, Q, in some cases this is obtained straightforwardly from the problem definition, for example, in the case of multi-class classification where Q is the number of classes. In other problems we know the number of latent functions underlying a particular phenomenon, as was the case for the inversion problem in §9.8. However, in general, we do not know Q and it is necessary to set up this parameter

by empirical techniques such as cross-validation. Furthermore, caution must taken when interpreting the results as the model may not be identifiable.

Finally, concerning the number of samples (S) and the mini-batch size (B), our experience indicates that the main guiding principle is dictated by computational constraints. In the former case, S helps reduce the variance in the ELBO and gradient estimates, so the larger the better. Moreover, because of computer-architecture specifics there may be zero overhead when going from 1 sample to, say, 250 samples. Similarly, in some of our experiments the larger the mini-batch size the more stable the behavior of the ELBO, although for large problems such as MNIST8M in §9.7.3, we found that $B = 1,000$ provided sufficient stability in our experiments. Unfortunately, stability of stochastic optimization when using MC estimates and mini-batches, does depend on the nature of the conditional likelihood used, as described in §K.3 where we required $10,000$ samples to reach stability on the SARCOS dataset. We attribute this to the high-variance of the conditional likelihood (which requires the multiplication of two latent processes) along with the type of gradient estimator we have used (the score-function estimator). Recently, Krauth et al. (2017) have shown that this can be reduced significantly when using the re-parametrization trick.

10. Conclusions and Discussion

We have developed *scalable automated variational inference for Gaussian process* models (SAVIGP), an inference method for models with Gaussian process (GP) priors, multiple outputs, and nonlinear likelihoods. The method is generally applicable to black-box likelihoods, i.e. it does not need to know the details of the conditional likelihood (or its gradients), only requiring its evaluation as a black-box function.

One of the key properties of this method is that, despite using a flexible variational posterior such as a mixture-of-Gaussians distribution, it is *statistically efficient* in that it requires samples from univariate Gaussian distributions to estimate the evidence lower bound (ELBO) and its gradients.

In order to provide scalability to very large datasets, we have used an augmented prior via the so-called inducing variables, which are prevalent in most sparse GP approximations. This has allowed us to decompose the ELBO as a sum of terms over the datapoints, hence giving way for the application of stochastic optimization and parallel computation.

Our small-scale experiments have shown that SAVIGP can perform as accurately as solutions that have been hard-coded specifically for the conditional likelihood of the problem at hand, even under high sparsity levels.

Our medium-scale experiments also have shown the effectiveness of SAVIGP, when considering problems such as multi-class classification on the MNIST dataset and highly nonlinear likelihoods such as that used in GPRN. On these experiments, we also have analyzed the effect of learning the inducing inputs, i.e. the locations corresponding to the inducing variables, and concluded that doing this can yield significant improvements in performance, reducing the numbers of inducing variables required to achieve similar accuracy by an order of magnitude. Nevertheless, there is generally a trade-off between reducing the time cost gained by having a lower number of inducing inputs and the overhead of carrying out optimization over these locations.

Our first large-scale experiment, on the problem of predicting airline delays, showed that SAVIGP is on par with the state-of-the-art approach for Gaussian process regression on big data (Hensman et al., 2013). In fact, our results show that SAVIGP is slightly better but we attribute those differences to implementation specifics. The important message is that SAVIGP can attain state-of-the-art performance even without exploiting specific knowledge of the likelihood model. Our second large-scale experiment shows that having an inference algorithm for GP models with non-Gaussian likelihood for large datasets is worthwhile, as the performance obtained on MNIST8M (using 8.1M observations) was significantly better than on the standard MNIST (using 60K observations).

Our final experiment considered a non-standard inference problem concerning a seismic inversion task. In this problem SAVIGP yielded a solution for the posterior over latent functions that closely matched the solution obtained by a (non-scalable) sampling algorithm.

Overall, we believe SAVIGP has the potential to be a powerful tool for practitioners and researchers when devising models for new or existing problems with Gaussian process priors for which variational inference is not yet available. As mentioned in Section 2, we are very much aware of recent developments in the areas of probabilistic programming, stochastic variational inference and Bayesian deep learning. Advances such as those in black-box variational inference (BBVI, Ranganath et al., 2014) and variational auto-encoders (Rezende et al., 2014; Kingma and Welling, 2014) are incredibly exciting for the machine learning community. While the former, BBVI, is somewhat too general to be useful in practice for GP models, the latter (variational auto-encoders) requires specific knowledge of the likelihood and is not a truly black-box method.

Finally, we are working on extending our models to more complex settings such as structured prediction problems, i.e. where the conditional likelihood is a non-iid model such as a chain or a tree (see e.g. Galliani et al., 2017, for a recent reference). Such settings provide incredible challenges from the computational and modeling perspectives. For example, how to deal with the exponential increase in the number of parameters of the model, and how to reduce the number of calls to an expensive conditional likelihood model. We believe that the benefits of being Bayesian well outweigh the effort in exploring such challenges.

Acknowledgments

We acknowledge the contribution by Trung V. Nguyen to the original conference paper (Nguyen and Bonilla, 2014a). EVB started this work at the University of New South Wales (UNSW SYDNEY) and was partially supported by UNSW's Faculty of Engineering Research Grant Program project # PS37866; UNSW's Academic Start-Up Funding Scheme project # PS41327; and an AWS in Education Research Grant award. AD started this work at UNSW SYDNEY, and was supported by a Research Fellowship from UNSW SYDNEY.

Appendix A. Derivation of the KL-divergence Term

Here we derive the expressions for the terms composing the KL-divergence ($\mathcal{L}_{\text{kl}} = \mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{cross}}$) part of the log-evidence lower bound ($\mathcal{L}_{\text{elbo}}$).

The entropy term is given by:

$$\begin{aligned}
\mathcal{L}_{\text{ent}}(\boldsymbol{\lambda}) &= -\mathbb{E}_{q(\mathbf{u}|\boldsymbol{\lambda})}[\log q(\mathbf{u}|\boldsymbol{\lambda})] \\
&= -\int \sum_{k=1}^{K} \pi_k q_k(\mathbf{u}|\mathbf{m}_k, \mathbf{S}_k) \log \sum_{\ell=1}^{K} \pi_\ell q_\ell(\mathbf{u}|\mathbf{m}_\ell, \mathbf{S}_\ell) \mathrm{d}\mathbf{u} \\
&= -\sum_{k=1}^{K} \pi_k \int \mathcal{N}(\mathbf{u}; \mathbf{m}_k, \mathbf{S}_k) \log \sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\mathbf{u}; \mathbf{m}_\ell, \mathbf{S}_\ell) \mathrm{d}\mathbf{u} \qquad (27) \\
&\geq -\sum_{k=1}^{K} \pi_k \log \int \mathcal{N}(\mathbf{u}; \mathbf{m}_k, \mathbf{S}_k) \sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\mathbf{u}|\mathbf{m}_\ell, \mathbf{S}_\ell) \mathrm{d}\mathbf{u} \qquad (28) \\
&= -\sum_{k=1}^{K} \pi_k \log \sum_{\ell=1}^{K} \pi_\ell \int \mathcal{N}(\mathbf{u}; \mathbf{m}_k, \mathbf{S}_k) \mathcal{N}(\mathbf{u}|\mathbf{m}_\ell, \mathbf{S}_\ell) \mathrm{d}\mathbf{u} \\
&= -\sum_{k=1}^{K} \pi_k \log \sum_{\ell=1}^{K} \pi_\ell \mathcal{N}(\mathbf{m}_k; \mathbf{m}_\ell, \mathbf{S}_k + \mathbf{S}_\ell) \stackrel{\text{def}}{=} \widehat{\mathcal{L}}_{\text{ent}},
\end{aligned}$$

where we have used Jensen's inequality to bring the logarithm out of the integral from Equations (27) to (28).

The negative cross-entropy term can be computed as:

$$\begin{aligned}
\mathcal{L}_{\text{cross}}(\boldsymbol{\lambda}) &= \mathbb{E}_{q(\mathbf{u}|\boldsymbol{\lambda})}[\log p(\mathbf{u})] = \sum_{k=1}^{K} \int \pi_k q_k(\mathbf{u}|\boldsymbol{\lambda}_k) \log p(\mathbf{u}) \mathrm{d}\mathbf{u} \\
&= \sum_{k=1}^{K} \sum_{j=1}^{Q} \pi_k \int \mathcal{N}(\mathbf{u}_j; \mathbf{m}_{kj}, \mathbf{S}_{kj}) \log \mathcal{N}(\mathbf{u}_j; \mathbf{0}, \mathbf{K}_{\mathbf{zz}}^{j}) \mathrm{d}\mathbf{u}_j \\
&= \sum_{k=1}^{K} \sum_{j=1}^{Q} \pi_k \left[\log \mathcal{N}(\mathbf{m}_{kj}; \mathbf{0}, \mathbf{K}_{\mathbf{zz}}^{j}) - \frac{1}{2} \operatorname{tr} (\mathbf{K}_{\mathbf{zz}}^{j})^{-1} \mathbf{S}_{kj} \right] \\
&= -\frac{1}{2} \sum_{k=1}^{K} \pi_k \sum_{j=1}^{Q} \left[M \log 2\pi + \log \left| \mathbf{K}_{\mathbf{zz}}^{j} \right| + \mathbf{m}_{kj}^{T} (\mathbf{K}_{\mathbf{zz}}^{j})^{-1} \mathbf{m}_{kj} + \operatorname{tr} (\mathbf{K}_{\mathbf{zz}}^{j})^{-1} \mathbf{S}_{kj} \right].
\end{aligned}$$

Appendix B. Proof of Theorem 1

In this section we prove Theorem 1 concerning the statistical efficiency of the estimator of the expected log likelihood and its gradients, i.e. that both can be estimated using expectations over Univariate Gaussian distributions.

B.1. Estimation of $\mathcal{L}_{\text{ell}}$

Taking the original expression in Equation (16) we have that:

$$\begin{aligned}
\mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}) &= \mathbb{E}_{q(\mathbf{f}|\boldsymbol{\lambda})}[\log p(\mathbf{y}|\mathbf{f}, \boldsymbol{\phi})], \\
&= \sum_{n=1}^{N} \int_{\mathbf{f}} q(\mathbf{f}|\boldsymbol{\lambda}) \log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi}) \mathrm{d}\mathbf{f}, \qquad (29)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{n=1}^{N} \int_{\mathbf{f}_{n\cdot}} \int_{\mathbf{f}_{\neg n\cdot}} q(\mathbf{f}_{\neg n\cdot}|\mathbf{f}_{n\cdot}) q(\mathbf{f}_{n\cdot}) \log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi}) \mathrm{d}\mathbf{f}_{\neg n\cdot} \mathrm{d}\mathbf{f}_{n\cdot}, \\
&= \sum_{n=1}^{N} \mathbb{E}_{q_{(n)}(\mathbf{f}_{n\cdot})}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi})], \qquad (30) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi})], \qquad (31)
\end{aligned}$$

where we have applied the linear property of the expectation operation in Equation (29); used $\mathbf{f}_{\neg n\cdot}$ to denote all the latent functions except those corresponding to the nth observation, and integrated these out to obtain Equation (30); and used the form of the marginal posterior (MoG in Equation (17)) to get Equation (31).

As described in Section 4.6, $q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)$ is a Q-dimensional Gaussian with diagonal covariance, hence computation of Equation (31) only requires expectations over univariate Gaussian distributions. ■

B.2. Estimation of the Gradients of $\mathcal{L}_{\text{ell}}$

Denoting the kth term for the nth observation of the expected log likelihood with $\mathcal{L}_{\text{ell}}^{(k,n)}$ we have that:

$$\begin{aligned}
\mathcal{L}_{\text{ell}}^{(k,n)} &= \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi})] \\
&= \int_{\mathbf{f}_{n\cdot}} q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k) \log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi}) \mathrm{d}\mathbf{f}_{n\cdot} \\
\nabla_{\boldsymbol{\lambda}_k} \mathcal{L}_{\text{ell}}^{(k,n)} &= \int_{\mathbf{f}_{n\cdot}} q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k) \nabla_{\boldsymbol{\lambda}_k} \log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k) \log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi}) \mathrm{d}\mathbf{f}_{n\cdot} \qquad (32) \\
&= \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)} \left[\nabla_{\boldsymbol{\lambda}_k} \log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k) \log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi}) \right], \qquad (33)
\end{aligned}$$

for $\boldsymbol{\lambda}_k \in \{\mathbf{m}_k, \mathbf{S}_k\}$, and for the mixture proportions the gradients can be estimated straightforwardly using Equation (22). We have used in Equation (32) the fact that $\nabla_{\mathbf{x}} f(\mathbf{x}) = f(\mathbf{x}) \nabla_{\mathbf{x}} \log f(\mathbf{x})$ for any nonnegative function $f(\mathbf{x})$. We see that our resulting gradient estimate has an analogous form to that obtained in Equation (31), hence its computation only requires expectations over univariate Gaussian distributions. ■

Appendix C. Gradients of the Evidence Lower Bound wrt Variational Parameters

Here we specify the gradients of the log-evidence lower bound ($\mathcal{L}_{\text{elbo}}$) wrt variational parameters. For the covariance, we consider $\mathbf{S}_{kj}$ of general structure but also give the updates when $\mathbf{S}_{kj}$ is a diagonal matrix, denoted with $\tilde{\mathbf{S}}_{kj}$.

C.1. KL-divergence Term

Let $\mathbf{K}_{zz}$ be the block-diagonal covariance with Q blocks $\mathbf{K}_{\mathbf{zz}}^{j}$, $j = 1, \ldots Q$. Additionally, let us assume the following definitions:

$$\mathbf{C}_{kl} \stackrel{\text{def}}{=} \mathbf{S}_k + \mathbf{S}_\ell, \qquad (34)$$

$$\mathcal{N}_{k\ell} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{m}_k; \mathbf{m}_\ell, \mathbf{C}_{kl}),$$
$$z_k \stackrel{\text{def}}{=} \sum_{\ell=1}^{K} \pi_\ell \mathcal{N}_{k\ell}.$$

The gradients of $\mathcal{L}_{\text{kl}}$ wrt the posterior mean and posterior covariance for component k are:

$$\nabla_{\mathbf{m}_k} \mathcal{L}_{\text{cross}} = -\pi_k \mathbf{K}_{zz}^{-1} \mathbf{m}_k,$$
$$\nabla_{\mathbf{S}_k} \mathcal{L}_{\text{cross}} = -\frac{1}{2}\pi_k \mathbf{K}_{zz}^{-1}, \text{ and for diagonal covariance we have:}$$
$$\nabla_{\tilde{\mathbf{S}}_k} \mathcal{L}_{\text{cross}} = -\frac{1}{2}\pi_k \operatorname{diag}(\mathbf{K}_{zz}^{-1}),$$
$$\nabla_{\pi_k} \mathcal{L}_{\text{cross}} = -\frac{1}{2}\sum_{j=1}^{Q}[M \log 2\pi + \log \left|\mathbf{K}_{\mathbf{zz}}^{j}\right| + \mathbf{m}_{kj}^T (\mathbf{K}_{\mathbf{zz}}^{j})^{-1}\mathbf{m}_{kj} + \text{ tr } (\mathbf{K}_{\mathbf{zz}}^{j})^{-1}\mathbf{S}_{kj}],$$

where we note that we compute $\mathbf{K}_{zz}^{-1}$ by inverting the corresponding blocks $\mathbf{K}_{\mathbf{zz}}^{j}$ independently. The gradients of the entropy term wrt the variational parameters are:

$$\nabla_{\mathbf{m}_k} \widehat{\mathcal{L}}_{\text{ent}} = \pi_k \sum_{\ell=1}^{K} \pi_\ell \left(\frac{\mathcal{N}_{k\ell}}{z_k} + \frac{\mathcal{N}_{k\ell}}{z_\ell}\right) \mathbf{C}_{kl}^{-1}(\mathbf{m}_k - \mathbf{m}_\ell),$$
$$\nabla_{\mathbf{S}_k} \widehat{\mathcal{L}}_{\text{ent}} = \frac{1}{2}\pi_k \sum_{\ell=1}^{K} \pi_\ell \left(\frac{\mathcal{N}_{k\ell}}{z_k} + \frac{\mathcal{N}_{k\ell}}{z_\ell}\right) \left[\mathbf{C}_{kl}^{-1} - \mathbf{C}_{kl}^{-1}(\mathbf{m}_k - \mathbf{m}_\ell)(\mathbf{m}_k - \mathbf{m}_\ell)^T \mathbf{C}_{kl}^{-1}\right],$$
$$\text{and for diagonal covariance we have:}$$
$$\nabla_{\tilde{\mathbf{S}}_k} \widehat{\mathcal{L}}_{\text{ent}} = \frac{1}{2}\pi_k \sum_{\ell=1}^{K} \pi_\ell \left(\frac{\mathcal{N}_{k\ell}}{z_k} + \frac{\mathcal{N}_{k\ell}}{z_\ell}\right) \left[\tilde{\mathbf{C}}_{kl}^{-1} - \tilde{\mathbf{C}}_{kl}^{-1} \operatorname{diag}\left((\mathbf{m}_k - \mathbf{m}_\ell) \odot (\mathbf{m}_k - \mathbf{m}_\ell)\right) \tilde{\mathbf{C}}_{kl}^{-1}\right],$$
$$\nabla_{\pi_k} \widehat{\mathcal{L}}_{\text{ent}} = -\log z_k - \sum_{\ell=1}^{K} \pi_\ell \frac{\mathcal{N}_{k\ell}}{z_\ell},$$

where $\tilde{\mathbf{C}}_{kl}$ is the diagonal matrix defined analogously to $\mathbf{C}_{kl}$ in Equation (34) and $\odot$ is the Hadamard product.

C.2. Expected Log Likelihood Term

Monte Carlo estimates of the gradients of the expected log likelihood term are:

$$\nabla \mathbf{m}_{kj} \widehat{\mathcal{L}}_{\text{ell}} = \frac{\pi_k}{S}(\mathbf{K}_{\mathbf{zz}}^{j})^{-1} \sum_{n=1}^{N} \mathbf{k}_{\mathbf{z}n}^{j} [\mathbf{\Sigma}_{k(n)}]_{j,j}^{-1} \sum_{i=1}^{S} \left(f_{nj}^{(k,i)} - [\mathbf{b}_{k(n)}]_j\right) \log p(\mathbf{y}_n | \mathbf{f}_{n\cdot}^{(k,i)}),$$
$$\nabla \mathbf{S}_{kj} \widehat{\mathcal{L}}_{\text{ell}} = \frac{\pi_k}{2S} \sum_{n=1}^{N} \left(\mathbf{a}_{jn}\mathbf{a}_{jn}^T\right) \sum_{i=1}^{S} \left[[\mathbf{\Sigma}_{k(n)}]_{j,j}^{-2} \left(f_{nj}^{(k,i)} - [\mathbf{b}_{k(n)}]_j\right)^2 - [\mathbf{\Sigma}_{k(n)}]_{j,j}^{-1}\right] \log p(\mathbf{y}_n | \mathbf{f}_{n\cdot}^{(k,i)}), \tag{35}$$
$$\nabla \pi_k \widehat{\mathcal{L}}_{\text{ell}} = \frac{1}{S} \sum_{n=1}^{N} \sum_{i=1}^{S} \log p(\mathbf{y}_n | \mathbf{f}_{n\cdot}^{(k,i)}),$$

and for diagonal covariance $\tilde{\mathbf{S}}_{kj}$ we replace $\left(\mathbf{a}_{jn}\mathbf{a}_{jn}^T\right)$ with $\operatorname{diag}\left(\mathbf{a}_{jn} \odot \mathbf{a}_{jn}\right)$ in Equation (35), where $\odot$ is the Hadamard product and $\operatorname{diag}(\mathbf{v})$ takes the input vector $\mathbf{v}$ and outputs a matrix with $\mathbf{v}$ on its diagonal. We have also defined above $\mathbf{k}_{\mathbf{z}n}^{j} \stackrel{\text{def}}{=} \kappa_j(\mathbf{Z}_j, \mathbf{x}_n)$, i.e. the vector obtained from evaluating the covariance function j between the inducing points $\mathbf{Z}_j$ and datapoint $\mathbf{x}_n$.

Appendix D. Gradients of the Evidence Lower Bound wrt Covariance Hyperparameters and Likelihood Parameters

Here we give the gradients of the variational objective ($\mathcal{L}_{\text{elbo}}$) wrt the covariance hyperparameters and, when required, the conditional likelihood parameters. We note here that our method does not require gradients of the conditional likelihood $p(\mathbf{y}|\mathbf{f}, \boldsymbol{\phi})$ wrt the latent functions $\mathbf{f}$. However, if the conditional likelihood is parametrized by $\boldsymbol{\phi}$ and point-estimates of these parameters are needed, these can also be learned in our variational framework.

D.1. Covariance Hyperparameters

The gradients of the terms in the KL-divergence part of $\mathcal{L}_{\text{elbo}}$ wrt a covariance hyperparameter θ_j are:

$$\nabla_{\theta_j} \widehat{\mathcal{L}}_{\text{ent}} = 0,$$
$$\nabla_{\theta_j} \mathcal{L}_{\text{cross}} = -\frac{1}{2}\sum_{k=1}^{K} \pi_k \text{ tr } \left[(\mathbf{K}_{\mathbf{zz}}^{j})^{-1} \nabla_{\theta_j} \mathbf{K}_{\mathbf{zz}}^{j} - (\mathbf{K}_{\mathbf{zz}}^{j})^{-1} \nabla_{\theta_j} \mathbf{K}_{\mathbf{zz}}^{j} (\mathbf{K}_{\mathbf{zz}}^{j})^{-1} \left(\mathbf{m}_{kj}\mathbf{m}_{kj}^T + \mathbf{S}_j\right)\right].$$

For the $\widehat{\mathcal{L}}_{\text{ell}}$ we have that:

$$\nabla_{\theta_j} \widehat{\mathcal{L}}_{\text{ell}} = \sum_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)} \left[\nabla_{\theta_j} \log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k) \log p(\mathbf{y}_n|\mathbf{f}_{n\cdot})\right],$$

and computing the corresponding gradient we obtain:

$$\nabla_{\theta_j} \widehat{\mathcal{L}}_{\text{ell}} = -\frac{1}{2}\sum_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)} \Big[\Big([\mathbf{\Sigma}_{k(n)}]_{j,j}^{-1} \nabla_{\theta_j} [\mathbf{\Sigma}_{k(n)}]_{j,j} - 2(f_{nj} - [\mathbf{b}_{k(n)}]_j)[\mathbf{\Sigma}_{k(n)}]_{j,j}^{-1} \nabla_{\theta_j} [\mathbf{b}_{k(n)}]_j - (f_{nj} - [\mathbf{b}_{k(n)}]_j)^2 [\mathbf{\Sigma}_{k(n)}]_{j,j}^{-2} \nabla_{\theta_j} [\mathbf{\Sigma}_{k(n)}]_{j,j}\Big) \log p(\mathbf{y}_n|\mathbf{f}_{n\cdot})\Big], \tag{36}$$

for which we need:

$$\nabla_{\theta_j} [\mathbf{b}_{k(n)}]_j = \left(\nabla_{\theta_j} \mathbf{a}_{jn}^T\right) \mathbf{m}_{kj},$$
$$\nabla_{\theta_j} [\mathbf{\Sigma}_{k(n)}]_{j,j} = \nabla_{\theta_j} [\widetilde{\mathbf{K}}_j]_{n,n} + 2\left(\nabla_{\theta_j} \mathbf{a}_{jn}^T\right) \mathbf{S}_{kj} \mathbf{a}_{jn}$$
$$= \nabla_{\theta_j} \kappa(\mathbf{x}_n, \mathbf{x}_n) - \left(\nabla_{\theta_j} \mathbf{a}_{jn}^T\right) \mathbf{k}_{\mathbf{z}n}^{j} - \mathbf{a}_{jn}^T \nabla_{\theta_j} \mathbf{k}_{\mathbf{z}n}^{j} + 2\left(\nabla_{\theta_j} \mathbf{a}_{jn}^T\right) \mathbf{S}_{kj} \mathbf{a}_{jn},$$

where

$$\nabla_{\theta_j}\mathbf{a}_{jn}^T = \Big(\nabla_{\theta_j}(\mathbf{k}_{\mathbf{z}n}^j)^T - \mathbf{a}_{jn}^T\nabla_{\theta_j}\mathbf{K}_{\mathbf{zz}}^j\Big)(\mathbf{K}_{\mathbf{zz}}^j)^{-1},$$

and as in the main text we have defined $\mathbf{a}_{jn} \stackrel{\text{def}}{=} [\mathbf{A}_j]_{:,n}$, i.e. the vector corresponding to the nth column of $\mathbf{A}_j$. Furthermore, as in the previous section, $\mathbf{k}_{\mathbf{z}n}^j \stackrel{\text{def}}{=} \kappa_j(\mathbf{Z}_j, \mathbf{x}_n)$.

D.2. Likelihood Parameters

Since the terms in the KL-divergence do not depend on the likelihood parameters we have that $\nabla_{\boldsymbol{\phi}}\widehat{\mathcal{L}}_{\text{ent}} = \nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{cross}} = 0$. For the gradients of the expected log likelihood term we have that:

$$\nabla_{\boldsymbol{\phi}}\widehat{\mathcal{L}}_{\text{ell}} = \frac{1}{S}\sum_{n=1}^{N}\sum_{k=1}^{K}\pi_k\sum_{i=1}^{S}\nabla_{\boldsymbol{\phi}}\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}^{(k,i)}, \boldsymbol{\phi}),$$

where $\{\mathbf{f}_{n\cdot}^{(k,i)}\} \sim \mathcal{N}(\mathbf{f}_{n\cdot}; \mathbf{b}_{k(n)}, \boldsymbol{\Sigma}_{k(n)})$, for $k = 1, \ldots, K$ and $i = 1, \ldots, S$.

Appendix E. Gradients of the Log-Evidence Lower Bound wrt Inducing Inputs

These gradients can be obtained by using the same expression as the gradients wrt covariance hyperparameters in Appendix D.1, considering the inducing inputs as additional hyperparameters of the covariances. Therefore, we rewrite the gradients above keeping in mind that θ_j is an element of inducing input $\mathbf{Z}_j$, hence dropping those gradient terms that do not depend on $\mathbf{Z}_j$. As before we have that $\nabla_{\theta_j}\widehat{\mathcal{L}}_{\text{ent}} = 0$.

$$\nabla_{\theta_j}\mathcal{L}_{\text{cross}} = -\frac{1}{2}\sum_{k=1}^{K}\pi_k \text{ tr }\left\{\left[(\mathbf{K}_{\mathbf{zz}}^j)^{-1} - (\mathbf{K}_{\mathbf{zz}}^j)^{-1}\left(\mathbf{m}_{kj}\mathbf{m}_{kj}^T + \mathbf{S}_j\right)(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\right]\nabla_{\theta_j}\mathbf{K}_{\mathbf{zz}}^j\right\}. \quad (37)$$

Similarly, for the gradients of $\widehat{\mathcal{L}}_{\text{ell}}$ in Equation (36) we have that:

$$\nabla_{\theta_j}[\mathbf{b}_{k(n)}]_j = \nabla_{\theta_j}(\mathbf{k}_{\mathbf{z}n}^j)^T(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\mathbf{m}_{kj} - \mathbf{a}_{jn}^T\nabla_{\theta_j}\mathbf{K}_{\mathbf{zz}}^j(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\mathbf{m}_{kj},$$

and

$$\begin{aligned}\nabla_{\theta_j}[\boldsymbol{\Sigma}_{k(n)}]_{j,j} = \nabla_{\theta_j}(\mathbf{k}_{\mathbf{z}n}^j)^T\Big(&-(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\mathbf{k}_{\mathbf{z}n}^j - \mathbf{a}_{jn} + 2(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\mathbf{S}_{kj}\mathbf{a}_{jn}\Big)\\ &+ \mathbf{a}_{jn}^T\nabla_{\theta_j}\mathbf{K}_{\mathbf{zz}}^j(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\mathbf{k}_{\mathbf{z}n}^j - 2\mathbf{a}_{jn}^T\nabla_{\theta_j}\mathbf{K}_{\mathbf{zz}}^j(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\mathbf{S}_{kj}\mathbf{a}_{jn}.\end{aligned}$$

From the equations above, we see that in the computation of the gradients of $\widehat{\mathcal{L}}_{\text{ell}}$ there are two types of terms. The first term is of the form:

$$\nabla_{\theta}t_n^{(1)} \stackrel{\text{def}}{=} \mathbf{v}_n^T\nabla_{\theta}\mathbf{K}_{\mathbf{zz}}^j\mathbf{w}_n,$$

where we have dropped the index j on the LHS of the equation for simplicity in the notation and $\mathbf{v}_n, \mathbf{w}_n$ are M-dimensional vectors. Let $\mathbf{V}$ and $\mathbf{W}$ be the $M \times N$ matrices corresponding to the N vectors $\{\mathbf{v}_n\}$ and $\{\mathbf{w}_n\}$, respectively. Furthermore, let us assume $\tilde{\mathbf{Z}}_j^{(d)}$ is the $M \times M$

matrix of all pairwise differences on dimension d of all inducing points divided by the squared length-scale of the dimension ℓ_d^2,

$$[\tilde{\mathbf{Z}}_j^{(d)}]_{o,p} = \frac{[\mathbf{Z}_j]_{o,d} - [\mathbf{Z}_j]_{p,d}}{\ell_d^2}.$$

Hence, in the case of the squared exponential covariance function the above gradient can be calculated as follows (for all data points):

$$\nabla_{[\mathbf{Z}_j]_{:,d}}\mathbf{t}^{(1)} = -((\tilde{\mathbf{Z}}_j^{(d)} \odot \mathbf{K}_{\mathbf{zz}}^j)\mathbf{W}) \odot \mathbf{V} - ((\tilde{\mathbf{Z}}_j^{(d)} \odot \mathbf{K}_{\mathbf{zz}}^j)\mathbf{V}) \odot \mathbf{W},$$

where $\nabla_{[\mathbf{Z}_j]_{:,d}}\mathbf{t}^{(1)}$ is the $M \times N$ matrix of gradients corresponding to dimension d for all $m = 1, \ldots, M$ and $n = 1, \ldots N$.

Similarly, the second type of term is of the form:

$$\nabla_{\theta}t_n^{(2)} \stackrel{\text{def}}{=} \mathbf{v}_n^T\nabla_{\theta}\mathbf{k}_{\mathbf{z}n}^j,$$

where $\mathbf{v}_n$ and $\mathbf{V}$ are defined as before. The gradients in the above expression wrt to the inducing points (for all datapoints or a mini-batch) can be calculated as follows:

$$\nabla_{[\mathbf{Z}_j]_{:,d}}\mathbf{t}^{(2)} = -(\mathbf{K}_{\mathbf{xz}}^j \odot \mathbf{V}^T \odot \tilde{\mathbf{X}}^{(d)})^T,$$

where in the above equation $\tilde{\mathbf{X}}^{(d)}$ is the $N \times M$ matrix of all pairwise differences on dimension d between all datapoints and inducing points divided by the squared length-scale of the dimension ℓ_d^2:

$$[\tilde{\mathbf{X}}^{(d)}]_{o,p} = \frac{[\mathbf{X}]_{o,d} - [\mathbf{Z}_j]_{p,d}}{\ell_d^2}.$$

Appendix F. Control Variates

We use control variates (see e.g. Ross, 2006, §8.2) to reduce the variance of the gradient estimates. In particular, we are interested in estimating gradients of the form:

$$\begin{aligned}\nabla_{\lambda_k}\mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot})] &= \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[g(\mathbf{f}_{n\cdot})], \text{ with}\\ g(\mathbf{f}_{n\cdot}) &= \nabla_{\lambda_k}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}),\end{aligned}$$

where the expectations are computed using samples from $q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)$, which depends on the variational parameter λ_k. As suggested by Ranganath et al. (2014), a sensible control variate is the so-called score function

$$h(\mathbf{f}_{n\cdot}) = \nabla_{\lambda_k}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k),$$

whose expectation is zero. Hence, the function:

$$\tilde{g}(\mathbf{f}_{n\cdot}) = g(\mathbf{f}_{n\cdot}) - \hat{a}h(\mathbf{f}_{n\cdot}),$$

has the same expectation as $g(\mathbf{f}_{n\cdot})$ but lower variance when $\hat{a}$ is given by:

$$\hat{a} = \frac{\mathbb{C}\text{ov}[g(\mathbf{f}_{n\cdot}), h(\mathbf{f}_{n\cdot})]}{\mathbb{V}[h(\mathbf{f}_{n\cdot})]},$$

where $\mathbb{C}\text{ov}[g(\mathbf{f}_{n\cdot}), h(\mathbf{f}_{n\cdot})]$ is the covariance between $g(\mathbf{f}_{n\cdot})$ and $h(\mathbf{f}_{n\cdot})$; $\mathbb{V}[h(\mathbf{f}_{n\cdot})]$ is the variance of $h(\mathbf{f}_{n\cdot})$; and both are estimated using samples from $q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)$. Therefore, our corrected gradient is given by:

$$\begin{aligned}\tilde{\nabla}_{\lambda_k}\mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot})] &\stackrel{\text{def}}{=} \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}[\tilde{g}(\mathbf{f}_{n\cdot})] \\ &= \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)}\big[\nabla_{\lambda_k}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_k)(\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}) - \hat{a})\big].\end{aligned}$$

Appendix G. Derivations for the Dense Posterior Case

In this section we consider the case of having a dense variational posterior, i.e. not using the so-called sparse approximations. We derive the expressions for the components of the log-evidence lower bound and show that the posterior parameters of the variational distribution $q(\mathbf{f}|\boldsymbol{\lambda})$ are, in fact, of 'free'-form. Furthermore, we analyze the case of a Gaussian likelihood showing that, in the limit of a large number of samples, our estimates converge to the exact analytical solution.

G.1. Evidence Lower Bound

Here we show the derivations of all the terms in $\mathcal{L}_{\text{elbo}}$ when considering the dense case, i.e. $M = N$ and $\mathbf{Z}_j = \mathbf{X}$. As described in the main text, the resulting changes to these terms can be obtained by replacing M with N and $\mathbf{K}^j_{\mathbf{zz}}$ with $\mathbf{K}^j_{\mathbf{xx}}$. Hence, for the terms in the KL-divergence part of $\mathcal{L}_{\text{elbo}}$ we have that:

$$\widehat{\mathcal{L}}_{\text{ent}}(\boldsymbol{\lambda}) = -\sum_{k=1}^{K}\pi_k \log\sum_{\ell=1}^{K}\pi_\ell\mathcal{N}(\mathbf{m}_k;\mathbf{m}_\ell,\mathbf{S}_k+\mathbf{S}_\ell), \tag{38}$$

$$\mathcal{L}_{\text{cross}}(\boldsymbol{\lambda}) = -\frac{1}{2}\sum_{k=1}^{K}\pi_k\sum_{j=1}^{Q}[N\log 2\pi + \log\left|\mathbf{K}^j_{\mathbf{xx}}\right| + \mathbf{m}^T_{kj}(\mathbf{K}^j_{\mathbf{xx}})^{-1}\mathbf{m}_{kj} + \text{ tr }(\mathbf{K}^j_{\mathbf{xx}})^{-1}\mathbf{S}_{kj}], \tag{39}$$

where we recall that now $\mathbf{m}_{kj}$ and $\mathbf{S}_{kj}$ are N-dimensional objects. For the $\widehat{\mathcal{L}}_{\text{ell}}$ we still need to compute empirical expectations over the variational distribution $q(\mathbf{f}|\boldsymbol{\lambda})$, where the corresponding parameters are given by:

$$\begin{aligned}\mathbf{b}_{kj} &= \mathbf{A}_j\mathbf{m}_{kj} = \mathbf{K}^j_{\mathbf{xx}}(\mathbf{K}^j_{\mathbf{xx}})^{-1}\mathbf{m}_{kj} = \mathbf{m}_{kj}, \\ \boldsymbol{\Sigma}_{kj} &= \widetilde{\mathbf{K}}_j - \mathbf{A}_j\mathbf{K}^j_{\mathbf{xx}} + \mathbf{A}_j\mathbf{S}_{kj}\mathbf{A}_j \\ &= \mathbf{K}^j_{\mathbf{xx}} - \mathbf{K}^j_{\mathbf{xx}}(\mathbf{K}^j_{\mathbf{xx}})^{-1}\mathbf{K}^j_{\mathbf{xx}} + \mathbf{K}^j_{\mathbf{xx}}(\mathbf{K}^j_{\mathbf{xx}})^{-1}\mathbf{S}_{kj}\mathbf{K}^j_{\mathbf{xx}}(\mathbf{K}^j_{\mathbf{xx}})^{-1} \\ &= \mathbf{S}_{kj}.\end{aligned}$$

G.2. Gaussian Likelihoods

Consider the case of a single output and a single latent function ($Q = P = 1$) with a Gaussian conditional likelihood and a single full Gaussian variational posterior ($K = 1$).

$$\begin{aligned}p(\mathbf{y}|\mathbf{f}) &= \mathcal{N}(\mathbf{y};\mathbf{f},\sigma^2\mathbf{I}), \text{ and} \\ q(\mathbf{f}|\boldsymbol{\lambda}) &= \mathcal{N}(\mathbf{f};\mathbf{m},\mathbf{S}).\end{aligned}$$

The entropy and the cross entropy terms are given in Equations (38) and (39) with $Q = K = 1$. The expect log likelihood term can be determined analytically,

$$\begin{aligned}\mathcal{L}_{\text{ell}}(\boldsymbol{\lambda}) &= \sum_{n=1}^{N}\mathbb{E}_{q(\mathbf{f}|\boldsymbol{\lambda})}[\log p(y_n|f_n,\boldsymbol{\phi})], \\ &= \log\mathcal{N}(\mathbf{y};\mathbf{m},\sigma^2\mathbf{I}) - \frac{1}{2\sigma^2}\text{ tr }\mathbf{S}.\end{aligned}$$

Hence the gradients of $\mathcal{L}_{\text{elbo}}$ can also be determined analytically, yielding the optimal mean posterior as follows:

$$\begin{aligned}\nabla_{\mathbf{m}}\mathcal{L}_{\text{elbo}} &= \mathbf{K}^{-1}_{\mathbf{xx}}\mathbf{m} + \frac{1}{\sigma^2}(\mathbf{y}-\mathbf{m}) = 0 \\ \left(\mathbf{K}^{-1}_{\mathbf{xx}} + \frac{1}{\sigma^2}\right)\mathbf{m} &= \frac{1}{\sigma^2}\mathbf{y} \\ \widehat{\mathbf{m}} &= \left(\sigma^2\mathbf{K}^{-1}_{\mathbf{xx}} + \mathbf{I}\right)^{-1}\mathbf{y} \\ &= \mathbf{K}_{\mathbf{xx}}\left(\mathbf{K}_{\mathbf{xx}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}.\end{aligned}$$

Similarly for the posterior covariances we have that:

$$\nabla_{\mathbf{S}}\mathcal{L}_{\text{elbo}} = \frac{1}{2}\mathbf{S}^{-1} - \frac{1}{2}\mathbf{K}^{-1}_{\mathbf{xx}} - \frac{1}{2\sigma^2}\mathbf{I} = 0$$

$$\widehat{\mathbf{S}} = \left(\mathbf{K}^{-1}_{\mathbf{xx}} + \frac{1}{\sigma^2}\mathbf{I}\right)^{-1} \tag{40}$$

$$\begin{aligned}&= \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xx}}\frac{1}{\sigma^2}\left(\mathbf{I} + \mathbf{K}_{\mathbf{xx}}\frac{1}{\sigma^2}\right)^{-1}\mathbf{K}_{\mathbf{xx}} \qquad (41) \\ &= \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xx}}\left(\mathbf{K}_{\mathbf{xx}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{K}_{\mathbf{xx}},\end{aligned}$$

where we have used Woodbury's formula to go from Equation (40) to Equation (41). The Equations above for the optimal posterior mean and posterior covariance, $\widehat{\mathbf{m}}$ and $\widehat{\mathbf{S}}$, are the exact analytical expressions for regressions with Gaussian process priors and isotropic noise likelihoods, see Rasmussen and Williams (2006, §2.2).

Appendix H. Efficient Parametrization

In this appendix we prove Theorem 2, showing that it is possible to obtain an efficient parameterization of the posterior covariances when using a full Gaussian approximation. In this case we have that:

$$\begin{aligned}\nabla_{\mathbf{S}_j}\mathcal{L}_{\text{elbo}} &= \nabla_{\mathbf{S}_j}\mathcal{L}_{\text{ent}} + \nabla_{\mathbf{S}_j}\mathcal{L}_{\text{cross}} + \nabla_{\mathbf{S}_j}\mathcal{L}_{\text{ell}} \\ &= -\frac{1}{2}(\mathbf{K}^j_{\mathbf{zz}})^{-1} + \frac{1}{2}\mathbf{S}^{-1}_j + \sum_{n=1}^{N}\nabla_{\mathbf{S}_j}\mathbb{E}_{q_{(n)}(\mathbf{f}_{n\cdot})}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot})]\end{aligned}$$

Setting the gradients to zero we have that

$$\frac{1}{2}\mathbf{S}_j^{-1} = \frac{1}{2}(\mathbf{K}_{\mathbf{zz}}^j)^{-1} + \frac{1}{2}(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\sum_{n=1}^{N}\mathbf{k}_{\mathbf{z}n}^j\lambda_{jn}(\mathbf{k}_{\mathbf{z}n}^j)^T(\mathbf{K}_{\mathbf{zz}}^j)^{-1},$$

where

$$\lambda_{jn} = -2\frac{d\bar{\ell}_n}{d[\mathbf{\Sigma}_j]_{n,n}}$$
$$\bar{\ell}_n = \mathbb{E}_{q_{(n)}(\mathbf{f}_{n\cdot})}[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot})].$$

Therefore, the optimal solution for the posterior covariance is given by:

$$\begin{aligned}\widehat{\mathbf{S}_j} &= \left((\mathbf{K}_{\mathbf{zz}}^j)^{-1} + (\mathbf{K}_{\mathbf{zz}}^j)^{-1}\mathbf{K}_{\mathbf{zx}}^j\mathbf{\Lambda}_j\mathbf{K}_{\mathbf{xz}}^j(\mathbf{K}_{\mathbf{zz}}^j)^{-1}\right)^{-1},\\ &= \mathbf{K}_{\mathbf{zz}}^j\left(\mathbf{K}_{\mathbf{zz}}^j + \mathbf{K}_{\mathbf{zx}}^j\mathbf{\Lambda}_j\mathbf{K}_{\mathbf{xz}}^j\right)^{-1}\mathbf{K}_{\mathbf{zz}}^j,\end{aligned}$$

where $\mathbf{\Lambda}_j$ is a $N \times N$ diagonal matrix with $\{\lambda_{jn}\}_{n=1}^N$ on the diagonal. ■

Appendix I. Lower Variance of Mixture-of-Diagonals Posterior

In this section we prove Theorem 3. First we review Rao-Blackwellization (Casella and Robert, 1996), which is also known as partial averaging or conditional Monte Carlo. Suppose we want to estimate $V = \mathbb{E}_{p(\mathbf{x},\mathbf{y})}[h(\mathbf{X},\mathbf{Y})]$ where $(\mathbf{X},\mathbf{Y})$ is a random variable with probability density $p(\mathbf{x},\mathbf{y})$ and $h(\mathbf{X},\mathbf{Y})$ is a random variable that is a function of $\mathbf{X}$ and $\mathbf{Y}$. It is easy to see that

$$\begin{aligned}\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[h(\mathbf{X},\mathbf{Y})] &= \int p(\mathbf{x},\mathbf{y})h(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\\ &= \int p(\mathbf{y})p(\mathbf{x}|\mathbf{y})h(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\\ &= \mathbb{E}_{p(\mathbf{y})}\left[\hat{h}(\mathbf{Y})\right], \text{ with } \hat{h}(\mathbf{Y}) = \mathbb{E}_{p(\mathbf{x}|\mathbf{y})}[h(\mathbf{X},\mathbf{Y})|\mathbf{Y}],\end{aligned}$$

and, from the conditional variance formula,

$$\mathbb{V}[\hat{h}(\mathbf{Y})] < \mathbb{V}[h(\mathbf{X},\mathbf{Y})].$$

Therefore when $\hat{h}(\mathbf{Y})$ is easy to compute, it can be used to estimate V with a lower variance than the original estimator. When $p(\mathbf{x},\mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$, then $\hat{h}(\mathbf{Y})$ is simplified to

$$\hat{h}(\mathbf{Y}=\mathbf{y}) = \int p(\mathbf{x})h(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x} = \mathbb{E}_{p(\mathbf{x})}[h(\mathbf{X},\mathbf{Y})|\mathbf{Y}].$$

We apply Rao-Blackwellization to our problem with $\mathbf{f}_{n\cdot}$ playing the role of the conditioning variable $\mathbf{Y}$ and $\mathbf{f}_{\neg n\cdot}$ playing the role of $\mathbf{X}$, where $\mathbf{f}_{\neg n\cdot}$ denotes $\mathbf{f}$ excluding $\mathbf{f}_{n\cdot}$. We note that this Rao-Blackwellization analysis is only applicable to the dense case with a mixture of diagonals posterior, as it satisfies the independence condition, i.e. $p(\mathbf{x},\mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$.

First, we express the gradient of $\boldsymbol{\lambda}_{k(n)}$ as an expectation by interchanging the integral and gradient operators giving

$$\nabla_{\boldsymbol{\lambda}_{k(n)}}\mathbb{E}_{q_k(\mathbf{f}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}|\mathbf{f},\boldsymbol{\phi})] = \mathbb{E}_{q_k(\mathbf{f}|\boldsymbol{\lambda}_k)}\left[\nabla_{\boldsymbol{\lambda}_{k(n)}}\log q_k(\mathbf{f}|\boldsymbol{\lambda}_k)\log p(\mathbf{y}|\mathbf{f},\boldsymbol{\phi})\right].$$

The Rao-Blackwellized estimator is thus

$$\begin{aligned}\hat{h}(\mathbf{f}_{n\cdot}) &= \int q(\mathbf{f}_{\neg n\cdot})\nabla_{\boldsymbol{\lambda}_{k(n)}}\log q_k(\mathbf{f}|\boldsymbol{\lambda}_k)\log p(\mathbf{y}|\mathbf{f})\mathrm{d}\mathbf{f}_{\neg n\cdot}\\ &= \int q(\mathbf{f}_{\neg n\cdot})\nabla_{\boldsymbol{\lambda}_{k(n)}}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_{k(n)})\log p(\mathbf{y}|\mathbf{f})\mathrm{d}\mathbf{f}_{\neg n\cdot}\\ &= \nabla_{\boldsymbol{\lambda}_{k(n)}}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_{k(n)})\int q(\mathbf{f}_{\neg n\cdot})\left[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}) + \log p(\mathbf{y}_{\neg n}|\mathbf{f}_{\neg n\cdot})\right]\mathrm{d}\mathbf{f}_{\neg n\cdot} \quad (42)\\ &= \nabla_{\boldsymbol{\lambda}_{k(n)}}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_{k(n)})\left[\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}) + C\right],\end{aligned}$$

where we have used the factorization of the conditional likelihood in Equation (42), and where C is a constant w.r.t $\mathbf{f}_{n\cdot}$. This gives the Rao-Blackwellized gradient,

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_{k(n)}}\mathbb{E}_{q_k(\mathbf{f}|\boldsymbol{\lambda}_k)}[\log p(\mathbf{y}|\mathbf{f})] &= \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_{k(n)})}\left[\hat{h}(\mathbf{f}_{n\cdot})\right]\\ &= \mathbb{E}_{q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_{k(n)})}\left[\nabla_{\boldsymbol{\lambda}_{k(n)}}\log q_{k(n)}(\mathbf{f}_{n\cdot}|\boldsymbol{\lambda}_{k(n)})\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot})\right],\end{aligned}$$

where we have used the fact that $\mathbb{E}_q[\nabla\log q] = 0$ for any q. We see then that the expression above is exactly the gradient obtained in Equation (33). ■

Appendix J. Details of Complexity Analysis

Here we give the details of the computational complexity of a single evaluation of the $\mathcal{L}_{\text{elbo}}$ and its gradients. Let $\mathbf{A}$ be some matrix of size $p \times q$, $\mathbf{B}$ be some matrix of size $q \times r$, $\mathbf{C}$ be some matrix of size $q \times p$, and $\mathbf{D}$ be a square matrix of size $s \times s$. Then we assume that $T(\mathbf{AB}) \in \mathcal{O}(pqr)$, $T(\mathbf{D}^{-1}) \in \mathcal{O}(s^3)$, $T(|\mathbf{D}|) \in \mathcal{O}(s^3)$, $T(\text{diag}(\mathbf{AC})) \in \mathcal{O}(pq)$, and $T(\text{ tr }(\mathbf{AC})) \in \mathcal{O}(pq)$.

J.1. Interim Values

We first provide an analysis of the cost of various interim expressions that get re-used throughout the computation of the $\mathcal{L}_{\text{elbo}}$. The interim expressions include: various values computed from the kernel functions ($\mathbf{K}_{\mathbf{zz}}^j$, $(\mathbf{K}_{\mathbf{zz}}^j)^{-1}$, $\left|\mathbf{K}_{\mathbf{zz}}^j\right|$, $\mathbf{k}_{\mathbf{z}n}^j$), values used to determine the shape of the Gaussian distributions from which we sample ($[\mathbf{b}_{k(n)}]_j$, $[\mathbf{\Sigma}_{k(n)}]_{j,j}$), and various intermediate values ($[\widetilde{\mathbf{K}}_j]_{n,n}$, $\mathbf{a}_{jn}$). We note that expressions with a subscript or superscript of j, n, or k are dependent on Q, B, and K respectively. For example, in the case of $\mathbf{k}_{\mathbf{z}n}^j$, we actually need to consider the cost of $Q \times B$ vectors of size M.

As stated in the main text, we assume that the kernel function is simple enough such that evaluating its output between two points is in $\mathcal{O}(D)$. Hence the cost of evaluating interim values is given by:

$$T(\mathbf{K}_{\mathbf{zz}}^j) \in \mathcal{O}(QM^2D),$$

$$
\begin{aligned}
T((\mathbf{K}_{\mathbf{zz}}^{j})^{-1}) &\in \mathcal{O}(QM^3),\\
T(|\mathbf{K}_{\mathbf{zz}}^{j}|) &\in \mathcal{O}(QM^3),\\
T(\mathbf{k}_{\mathbf{z}n}^{j}) &\in \mathcal{O}(QBMD),\\
T([\mathbf{b}_{k(n)}]_j) &\in \mathcal{O}(KQBM),\\
T([\mathbf{\Sigma}_{k(n)}]_{j,j}) &\in \mathcal{O}(KQBM^2),\\
T([\widetilde{\mathbf{K}}_j]_{n,n}) &\in \mathcal{O}(QBM),\\
T(\mathbf{a}_{jn}) &\in \mathcal{O}(QBM^2),\\
&\text{and for diagonal covariance we have:}\\
T([\mathbf{\Sigma}_{k(n)}]_{j,j}) &\in \mathcal{O}(KQBM).
\end{aligned}
$$

The computational cost of evaluating all interim values is thus in $\mathcal{O}(QM(MD+M^2+BD+KBM))$ for the case of full covariance and in $\mathcal{O}(QM(MD+M^2, BD, KB))$ for diagonal covariances.

J.2. Analysis of the $\mathcal{L}_{\mathbf{kl}}$

The computational complexity of terms in the KL-divergence and its gradients is given by:

$$
\begin{aligned}
T(\mathcal{L}_{\text{cross}}) &\in \mathcal{O}(KQM^2),\\
T(\mathcal{L}_{\text{ent}}) &\in \mathcal{O}(K^2QM^3),\\
T(\nabla_{\mathbf{m}}\mathcal{L}_{\text{cross}}) &\in \mathcal{O}(KQM^2),\\
T(\nabla_{\mathbf{S}}\mathcal{L}_{\text{cross}}) &\in \mathcal{O}(KQM^2),\\
T(\nabla_{\pi_k}\mathcal{L}_{\text{cross}}) &\in \mathcal{O}(KQM^2),\\
T(\nabla_{\mathbf{m}}\mathcal{L}_{\text{ent}}) &\in \mathcal{O}(K^2QM^3),\\
T(\nabla_{\mathbf{S}}\mathcal{L}_{\text{ent}}) &\in \mathcal{O}(K^2QM^3),\\
T(\nabla_{\pi_k}\mathcal{L}_{\text{ent}}) &\in \mathcal{O}(K^2QM^3),\\
&\text{and for diagonal covariance we have:}\\
T(\nabla_{\tilde{\mathbf{S}}}\mathcal{L}_{\text{cross}}) &\in \mathcal{O}(KQM),\\
T(\nabla_{\mathbf{m}}\mathcal{L}_{\text{ent}}) &\in \mathcal{O}(K^2QM),\\
T(\nabla_{\tilde{\mathbf{S}}}\mathcal{L}_{\text{ent}}) &\in \mathcal{O}(K^2QM),\\
T(\nabla_{\pi_k}\mathcal{L}_{\text{ent}}) &\in \mathcal{O}(K^2QM).
\end{aligned}
$$

Hence the computational cost of evaluating the KL-term and its gradients is in $\mathcal{O}(K^2QM^3)$ for full covariances, and $\mathcal{O}(KQM(K+M))$ for diagonal covariances.

J.3. Analysis of the $\mathcal{L}_{\mathbf{ell}}$

Let $T(\log p(\mathbf{y}_n|\mathbf{f}_{n\cdot}, \boldsymbol{\phi})) \in \mathcal{O}(L)$, then the cost of evaluating the expected log likelihood and its gradients is given by:

$$
\begin{aligned}
T(\mathcal{L}_{\text{ell}}) &\in \mathcal{O}(KBS(L+Q)),\\
T(\nabla_{\mathbf{m}}\mathcal{L}_{\text{ell}}) &\in \mathcal{O}(KQ(M^2+BM+BSL)),\\
T(\nabla_{\mathbf{S}}\mathcal{L}_{\text{ell}}) &\in \mathcal{O}(KQB(M^2+SL)),\\
T(\nabla_{\pi_k}\mathcal{L}_{\text{ell}}) &\in \mathcal{O}(KBS(L+Q)),\\
&\text{and for diagonal covariance we have:}\\
T(\nabla_{\tilde{\mathbf{S}}}\mathcal{L}_{\text{ell}}) &\in \mathcal{O}(KQB(M+SL)).
\end{aligned}
$$

Hence the computational cost of evaluating the ell term and its gradients is in $\mathcal{O}(KQB(M^2+SL))$ for full covariances, and $\mathcal{O}(KQB(M+SL))$ for diagonal covariances.

Appendix K. Additional Results

Here we present additional results concerning the small-scale and medium-scale experiments.

K.1. Small-scale Experiments—Warped GPs

In the case of the CREEP dataset (Figure 13), SSE and NLPD are better in denser models. As expected, the performance of the dense (SF = 1) FG model is identical to the performance of WGP. We also note that NLPD in GP has less variation compared to the other models, which can be attributed to its assumption of Gaussian noise.

K.2. Medium-scale Experiments—Binary Classification

We see in Figure 14 that the performance improves significantly with denser models, i.e. when we use a larger number of inducing variables. Overall, our model achieves an accuracy of 98.1% and a mean NLP of 0.062 on MNIST-B with a sparsity factor of 0.04. This is slightly better than the performance obtained by Hensman et al. (2015), who report an accuracy of 97.8% and NLP of 0.069, although they used a smaller number of inducing variables. As we have seen in §9.5, our model also achieves similar performance when using the same sparsity factor as that used by Hensman et al. (2015) when the inducing inputs (i.e. the locations of the inducing variables) are learned.

K.3. Medium-scale Experiments—Gaussian Process Regression Networks

Figure 15 shows the performance on the SARCOS dataset. Compared to SARCOS-2, learning all joints yields a slight increase in MSSE for joints 4 and 7, but also resulted in a significant decrease in NLPD. This shows that the transition from learning only 2 joints to all 7 joints (as in SARCOS-2), while unhelpful for point estimates, led to a significantly better posterior density estimate. We also note that we had to make use of 10,000 samples for this dataset to get stable results over different sparsity factors. This is a much larger number than the 100 samples used in MNIST. This is likely because the GPRN likelihood model involves multiplication between latent processes, which increases the variance of sampling results by a squared factor, and therefore, it is reasonable to require more samples in this complex likelihood model.

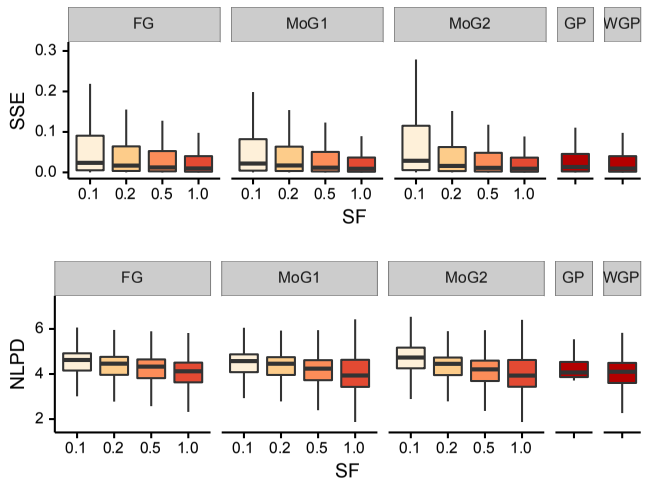


Figure 13: The distributions of SSE and NLPD for a warped Gaussian process likelihood model on the CREEP dataset. Three approximate posteriors in SAVIGP are used: FG (full Gaussian), MoG1 (diagonal Gaussian), and MoG2 (mixture of two diagonal Gaussians), along with various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model, with SF $= 1$ corresponding to the original model without sparsity. WGP corresponds to the performance of the exact inference method for warped Gaussian process models (Snelson et al., 2003), and GP is the performance of the exact inference on a univariate Gaussian likelihood model.

K.4. Medium-scale Experiments—Inducing-input Learning

Figure 16 shows the performance of the model under the two settings (with and without learning inducing inputs) on MNIST. As with MNIST-B, we see that learning the location of the inducing variables yields a large gain in performance. In fact, the sparser models with inducing point learning performed similarly to the denser models, despite the fact that the two models differed by an order of magnitude when it came to the number of inducing variables. Figures 17, 18 show the performance on SARCOS and SARCOS-2. We observe an improvement in performance from learning inducing inputs. However, the improvement is less significant than for MNIST, which shows that different datasets have significantly different sensitivity to the location of the inducing variables.

Appendix L. Notation

A summary of the notation used in this paper is shown in Table L.

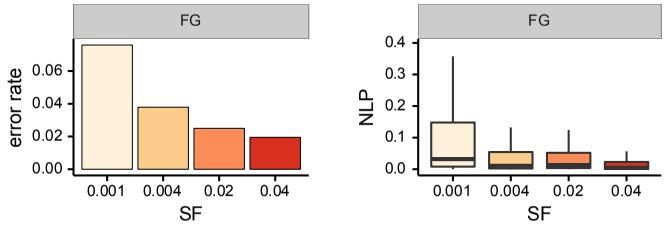


Figure 14: Error rates and NLP for binary classification on the MNIST-B dataset. We used a full Gaussian (FG) posterior approximation across various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model.

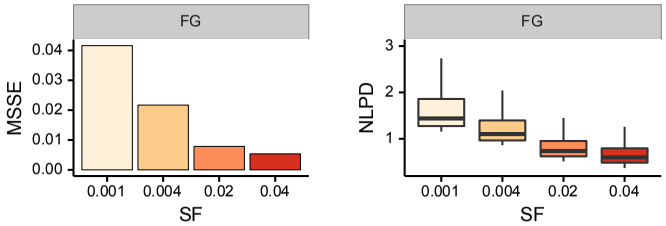


Figure 15: Mean SSE and NLPD for multi-output regression on the SARCOS dataset. We used a full Gaussian (FG) posterior approximation across various sparsity factors (SF $= M/N$). The smaller the SF the sparser the model.

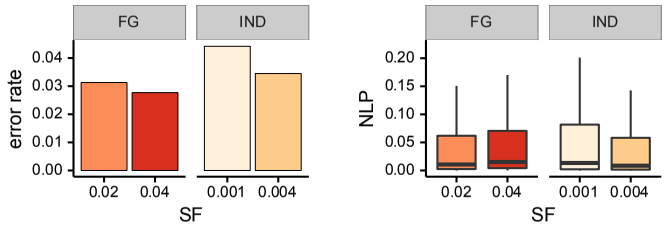


Figure 16: Comparison of error rate and NLP obtained by SAVIGP without learning (FG) and with learning (IND) of inducing inputs for multi-class classification on the MNIST dataset.

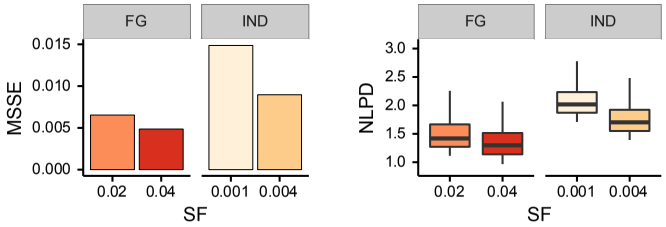


Figure 17: Comparison of mean SSE and NLPD obtained by SAVIGP without learning (FG) and with learning (IND) of inducing points for multi-output regression on the SARCOS-2 dataset.

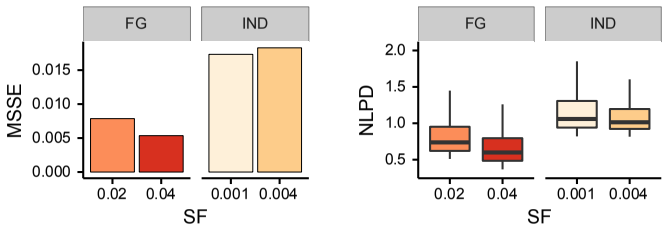


Figure 18: Comparison of mean SSE and NLPD obtained by SAVIGP without learning (FG) and with learning (IND) of inducing inputs for multi-output regression on the SARCOS dataset.

Symbol	Description
$\mathbf{v}$	Bold lower case denotes a vector
$\mathbf{C}$	Bold upper case denotes a matrix
$\lvert\mathbf{C}\rvert$	Determinant of matrix $\mathbf{C}$
tr $\mathbf{C}$	Trace of matrix $\mathbf{C}$
D	Input dimensionality
N	Number of training datapoints
M	Number of inducing variables per latent process
P	Number of outputs
Q	Number of latent functions
K	Number of mixture components in variational posterior
S	Number of samples in Monte Carlo estimates
$\mathbf{x}_n$	nth input datapoint
$\mathbf{X}$	$N \times D$ matrix of input data
$\mathbf{y}_n$	P-dimensional vector of outputs (label) for nth observation
$\mathbf{f}_{n\cdot}$	Q-dimensional vector of latent function values for nth observation
$\mathbf{f}_{\cdot j}$	N-dimensional vector of latent function values for jth process
$\mathbf{u}_{\cdot j}$	Inducing variables of latent process j
$\mathbf{u}$	Vector of all inducing variables
$\mathbf{Z}_j$	$M \times D$ Matrix of inducing inputs for latent process j
$\kappa_j(\mathbf{x}, \mathbf{x}')$	Covariance function of latent process j evaluated at $\mathbf{x}, \mathbf{x}'$
$\mathbf{K}^j_{\mathbf{xx}}$	$N \times N$ covariance matrix obtained by evaluating $\kappa_j(\mathbf{X}, \mathbf{X})$
$\mathbf{K}^j_{\mathbf{zz}}$	$M \times M$ covariance matrix obtained by evaluating $\kappa_j(\mathbf{Z}_j, \mathbf{Z}_j)$
$\mathbf{K}_{zz}$	Block-diagonal covariance with blocks $\mathbf{K}^j_{\mathbf{zz}}$
$\mathbf{K}^j_{\mathbf{xz}}$	$N \times M$ covariance between $\mathbf{X}$ and $\mathbf{Z}_j$
$\mathbf{k}^j_{\mathbf{z}n}$	$M \times 1$ covariance vector between $\mathbf{Z}_j$ and $\mathbf{x}_n$
$q(\mathbf{u}\lvert\boldsymbol{\lambda})$	Variational distribution over inducing variables
$q(\mathbf{f}\lvert\boldsymbol{\lambda})$	Variational distribution over latent functions
$\mathbf{m}$	Posterior mean over inducing variables
$\mathbf{S}$	Posterior covariance over inducing variables
$\mathbf{b}$	Posterior mean over latent functions
$\boldsymbol{\Sigma}$	Posterior covariance over latent functions

Table 4: Summary of notation used in the paper.

References

Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, and Dzmitry Bahdanau. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

Mauricio Alvarez and Neil D Lawrence. Sparse convolved Gaussian processes for multi-output regression. In *Neural Information Processing Systems*, pages 57–64. 2009.

Mauricio A Álvarez and Neil D Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12(5):1459–1500, 2011.

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *Artificial Intelligence and Statistics*, 2010.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.

Christopher M. Bishop, Neil D. Lawrence, Tommi Jaakkola, and Michael I. Jordan. Approximating posterior distributions in belief networks using mixtures. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Neural Information Processing Systems*, pages 416–422. MIT Press, 1998.

Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *Neural Information Processing Systems*. 2008.

Edwin V. Bonilla, Shengbo Guo, and Scott Sanner. Gaussian process preference elicitation. In *Neural Information Processing Systems*. 2010.

Edwin V Bonilla, Daniel Steinberg, and Alistair Reid. Extended and unscented kitchen sinks. In *International Conference on Machine Learning*, 2016.

Phillip Boyle and Marcus Frean. Dependent Gaussian processes. In *Neural Information Processing Systems*. 2005.

Thang D Bui, Josiah Yan, and Richard E Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research*, 18(1):3649–3720, 2017.

George Casella and Christian P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 1996.

Edward Challis and David Barber. Gaussian Kullback-Leibler approximate inference. *Journal of Machine Learning Research*, 14:2239–2286, 2013.

Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6(Jul):1019–1041, 2005.

D. Cole, C. Martin-Moran, A.G. Sheard, H.K.D.H. Bhadeshia, and D.J.C. MacKay. Modelling creep rupture strength of ferritic steel welds. *Science and Technology of Welding & Joining*, 5(2):81–89, 2000.

Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 1993.

Kurt Cutajar, Edwin V. Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep Gaussian processes. In *International Conference on Machine Learning*, volume 70, pages 884–893. PMLR, 06–11 Aug 2017.

Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*, 2013.

Sourish Das, Sasanka Roy, and Rajiv Sambasivan. Fast Gaussian process regression for big data. *arXiv preprint arXiv:1509.05142*, 2015.

Marc Peter Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *International Conference on Machine Learning*, 2015.

Amir Dezfouli and Edwin V. Bonilla. Scalable inference for Gaussian process models with black-box likelihoods. In *Neural Information Processing Systems*. 2015.

Pedro Domingos, Stanley Kok, Hoifung Poon, Matthew Richardson, and Parag Singla. Unifying logical and statistical AI. In *AAAI Conference on Artificial Intelligence*, 2006.

Kun Dong, David Eriksson, Hannes Nickisch, David Bindel, and Andrew G Wilson. Scalable log determinants for Gaussian process kernel learning. In *Neural Information Processing Systems*, pages 6327–6337, 2017.

Simon Duane, Anthony D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

Yarin Gal and Richard Turner. Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *International Conference on Machine Learning*, pages 655–664, 2015.

Yarin Gal, Mark van der Wilk, and Carl Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Neural Information Processing Systems*. 2014.

Pietro Galliani, Amir Dezfouli, Edwin V. Bonilla, and Novi Quadrianto. Gray-box inference for structured Gaussian process models. In Aarti Singh and Jerry Zhu, editors, *Artificial Intelligence and Statistics*, pages 353–361, Fort Lauderdale, FL, USA, Apr 2017. PMLR.

Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

Samuel J. Gershman, Matthew D. Hoffman, and David M. Blei. Nonparametric variational inference. In *International Conference on Machine Learning*, 2012.

Noah D. Goodman, Vikash K. Mansinghka, Daniel M. Roy, Keith Bonawitz, and Joshua B. Tenenbaum. Church: A language for generative models. In *Uncertainty in Artificial Intelligence*, 2008.

Ricardo Henao and Ole Winther. Predictive active set selection methods for Gaussian processes. *Neurocomputing*, 80:10–18, 2012.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, 2013.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, 2015.

James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18:151:1–151:52, 2017.

Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable Gaussian process classification via expectation propagation. In *Artificial Intelligence and Statistics*, pages 168–176, 2016.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Sixth Annual Conference on Computational Learning Theory*, pages 5–13. ACM, 1993.

Trong Nghia Hoang, Quang Minh Hoang, and Kian Hsiang Low. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *International Conference on Machine Learning*, 2015.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

Marco F. Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D. Hanebeck. On entropy approximation for Gaussian mixture random vectors. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008.

Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann Lecun. What is the best multi-stage architecture for object recognition? In *International Conference on Computer Vision*, 2009.

R.G. Jarrett. A note on the intervals between coal-mining disasters. *Biometrika*, 66(1): 191–193, 1979.

ST John and James Hensman. Large-scale Cox process inference using variational Fourier features. In *International Conference on Machine Learning*, 2018.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. *An introduction to variational methods for graphical models*. Springer, 1998.

Kristian Kersting, Christian Plagemann, Patrick Pfaff, and Wolfram Burgard. Most likely heteroscedastic Gaussian process regression. In *International Conference on Machine Learning*, 2007.

Mohammad E. Khan, Shakir Mohamed, Benjamin M. Marlin, and Kevin P. Murphy. A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *Artificial Intelligence and Statistics*, pages 610–618, 2012.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Karl Krauth, Edwin V. Bonilla, Kurt Cutajar, and Maurizio Filippone. AutoGP: Exploring the capabilities and limitations of gaussian process models. In *Uncertainty in Artificial Intelligence*, 2017.

Harri Lappalainen and James W. Miskin. *Ensemble Learning*, pages 75–92. Springer London, London, 2000.

Steffen L Lauritzen. Time series analysis in 1880: A discussion of contributions made by tn thiele. *International Statistical Review*, pages 319–331, 1981.

Neil Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

Neil D Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *Neural Information Processing Systems*, 2002.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum Gaussian process regression. *The Journal of Machine Learning Research*, 99:1865–1881, 2010.

Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. In *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA., 2007.

David J. C. MacKay. Developments in probabilistic modelling with neural networks — ensemble learning. In Bert Kappen and Stan Gielen, editors, *Third Annual Symposium on Neural Networks*, pages 191–198, London, 1995. Springer London.

David J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

Alexander G de G Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. *Journal of Machine Learning Research*, 51:231–239, 2016.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo Le'on-Villagr'a, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18 (40):1–6, apr 2017.

Vladimir Maz'ya and Gunther Schmidt. On approximate approximations using Gaussian kernels. *IMA Journal of Numerical Analysis*, 16(1):13–29, 1996.

Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence*, 2001.

Jesper Møller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. Log Gaussian Cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

Iain Murray, Ryan Prescott Adams, and David J.C. MacKay. Elliptical slice sampling. In *Artificial Intelligence and Statistics*, 2010.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto, 1993.

Trung V. Nguyen and Edwin Bonilla. Efficient variational inference for Gaussian process regression networks. In *Artificial Intelligence and Statistics*, pages 472–480, 2013.

Trung V. Nguyen and Edwin V. Bonilla. Automated variational inference for Gaussian process models. In *Neural Information Processing Systems*. 2014a.

Trung V. Nguyen and Edwin V. Bonilla. Collaborative multi-output Gaussian processes. In *Uncertainty in Artificial Intelligence*, 2014b.

Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(10), 2008.

Anthony O'Hagan and JFC Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42, 1978.

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural Computation*, 21(3):786–792, 2009.

C Paciorek and M Schervish. Nonstationary covariance functions for Gaussian process regression. *Neural Information Processing Systems*, 2004.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Neural Information Processing Systems*. 2008.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Neural Information Processing Systems*. 2009.

Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann Lecun. Efficient learning of sparse representations with an energy-based model. In *Neural Information Processing Systems*, 2006.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

Alistair Reid, Simon O'Callaghan, Edwin V. Bonilla, Lachlan McCalman, Tim Rawling, and Fabio Ramos. Bayesian joint inversions for the exploration of Earth resources. In *International Joint Conference on Artificial Intelligence*, 2013.

Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Sheldon M Ross. *Simulation*. Burlington, MA: Elsevier, 2006.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural gradients in practice: Non-conjugate variational inference in Gaussian process models. In *Artificial Intelligence and Statistics*, 2018.

Rishit Sheth, Yuyang Wang, and Roni Khardon. Sparse variational inference for generalized GP models. In *International Conference on Machine Learning*, 2015.

Ed Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Neural Information Processing Systems*, 2006.

Edward Snelson, Carl Edward Rasmussen, and Zoubin Ghahramani. Warped Gaussian processes. In *Neural Information Processing Systems*, 2003.

Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2004.

Yee Whye Teh, Matthias Seeger, and Michael I. Jordan. Semiparametric latent factor models. In *Artificial Intelligence and Statistics*, 2005.

Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, 2009.

Sethu Vijayakumar and Stefan Schaal. Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space. In *International Conference on Machine Learning*, 2000.

Christopher K.I. Williams and David Barber. Bayesian classification with Gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1342–1351, 1998.

Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for regression. In *Neural Information Processing Systems*, pages 514–520, 1996.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

Andrew G. Wilson, David A. Knowles, and Zoubin Ghahramani. Gaussian process regression networks. In *International Conference on Machine Learning*, 2012.

Frank Wood, Jan-Willem van de Meent, and Vikash Mansinghka. A new approach to probabilistic programming inference. In *Artificial Intelligence and Statistics*, pages 1024–1032, 2014.

Zichao Yang, Andrew Gordon Wilson, Alexander J. Smola, and Le Song. À la carte — learning fast kernels. In *Artificial Intelligence and Statistics*, 2015.

Kai Yu and Wei Chu. Gaussian process models for link analysis and transfer learning. In *Neural Information Processing Systems*, 2008.

Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Journal of Machine Learning Research 20 (2019) 1-34 Submitted 3/17; Revised 1/19; Published 8/19

Binarsity: a penalization for one-hot encoded features in linear supervised learning

Mokhtar Z. Alaya MOKHTARZAHDI.ALAYA@GMAIL.COM
Laboratoire de Probabilités Statistique et Modélisation, CNRS UMR 8001
Sorbonne University
Paris, France

Simon Bussy SIMON.BUSSY@GMAIL.COM
Laboratoire de Probabilités Statistique et Modélisation, CNRS UMR 8001
Sorbonne University
Paris, France

Stéphane Gaïffas STEPHANE.GAIFFAS@LPSM.PARIS
Laboratoire de Probabilités Statistique et Modélisation, CNRS UMR 8001
Université Paris Diderot
Paris, France

Agathe Guilloux AGATHE.GUILLOUX@MATH.CNRS.FR
LaMME, UEVE and UMR 8071
Université Paris Saclay
Evry, France

Editor: John Shawe-Taylor

Abstract

This paper deals with the problem of large-scale linear supervised learning in settings where a large number of continuous features are available. We propose to combine the well-known trick of one-hot encoding of continuous features with a new penalization called *binarsity*. In each group of binary features coming from the one-hot encoding of a single raw continuous feature, this penalization uses total-variation regularization together with an extra linear constraint. This induces two interesting properties on the model weights of the one-hot encoded features: they are piecewise constant, and are eventually block sparse. Non-asymptotic oracle inequalities for generalized linear models are proposed. Moreover, under a sparse additive model assumption, we prove that our procedure matches the state-of-the-art in this setting. Numerical experiments illustrate the good performances of our approach on several datasets. It is also noteworthy that our method has a numerical complexity comparable to standard ℓ_1 penalization.

Keywords: Supervised learning, Features binarization, Sparse additive modeling, Total-variation, Oracle inequalities, Proximal methods

1. Introduction

In many applications, datasets used for linear supervised learning contain a large number of continuous features, with a large number of samples. An example is web-marketing, where features are obtained from bag-of-words scaled using tf-idf (Russell, 2013), recorded during the visit of users on websites. A well-known trick (Wu and Coggeshall, 2012; Liu et al., 2002) in this setting is to

©2019 Mokhtar Z. Alaya, Simon Bussy, Stéphane Gaïffas and Agathe Guilloux.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-170.html.

replace each raw continuous feature by a set of binary features that one-hot encodes the interval containing it, among a list of intervals partitioning the raw feature range. This improves the linear decision function with respect to the raw continuous features space, and can therefore improve prediction. However, this trick is prone to over-fitting, since it increases significantly the number of features.

A new penalization. To overcome this problem, we introduce a new penalization called *binarsity*, that penalizes the model weights learned from such grouped one-hot encodings (one group for each raw continuous feature). Since the binary features within these groups are naturally ordered, the binarsity penalization combines a group total-variation penalization, with an extra linear constraint in each group to avoid collinearity between the one-hot encodings. This penalization forces the weights of the model to be as constant (with respect to the order induced by the original feature) as possible within a group, by selecting a minimal number of relevant cut-points. Moreover, if the model weights are all equal within a group, then the full block of weights is zero, because of the extra linear constraint. This allows to perform raw feature selection.

High-dimensional linear supervised learning. To address the high-dimensionality of features, sparse linear inference is now an ubiquitous technique for dimension reduction and variable selection, see for instance Bühlmann and van De Geer (2011) and Hastie et al. (2001) among many others. The principle is to induce sparsity (large number of zeros) in the model weights, assuming that only a few features are actually helpful for the label prediction. The most popular way to induce sparsity in model weights is to add a ℓ_1-penalization (Lasso) term to the goodness-of-fit (Tibshirani, 1996a). This typically leads to sparse parametrization of models, with a level of sparsity that depends on the strength of the penalization. Statistical properties of ℓ_1-penalization have been extensively investigated, see for instance Knight and Fu (2000); Zhao and Yu (2006); Bunea et al. (2007); Bickel et al. (2009) for linear and generalized linear models and Donoho and Huo (2001); Donoho and Elad (2002); Candès et al. (2008); Candès and Wakin (2008) for compressed sensing, among others.

However, the Lasso ignores ordering of features. In Tibshirani et al. (2005), a structured sparse penalization is proposed, known as fused Lasso, which provides superior performance in recovering the true model in such applications where features are ordered in some meaningful way. It introduces a mixed penalization using a linear combination of the ℓ_1-norm and the total-variation penalization, thus enforcing sparsity in both the weights and their successive differences. Fused Lasso has achieved great success in some applications such as comparative genomic hybridization (Rapaport et al., 2008), image denoising (Friedman et al., 2007), and prostate cancer analysis (Tibshirani et al., 2005).

Features discretization and cuts. For supervised learning, it is often useful to encode the input features in a new space to let the model focus on the relevant areas (Wu and Coggeshall, 2012). One of the basic encoding technique is *feature discretization* or *feature quantization* (Liu et al., 2002) that partitions the range of a continuous feature into intervals and relates these intervals with meaningful labels. Recent overviews of discretization techniques can be found in Liu et al. (2002) or Garcia et al. (2013).

Obtaining the optimal discretization is a NP-hard problem (Chlebus and Nguyen, 1998), and an approximation can be easily obtained using a greedy approach, as proposed in decision trees: CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993), among others, that sequentially select pairs

of features and cuts that minimize some purity measure (intra-variance, Gini index, information gain are the main examples). These approaches build decision functions that are therefore very simple, by looking only at a single feature at a time, and a single cut at a time. Ensemble methods (boosting (Lugosi and Vayatis, 2004), random forests (Breiman, 2001)) improve this by combining such decisions trees, at the expense of models that are harder to interpret.

Main contribution. This paper considers the setting of linear supervised learning. The main contribution of this paper is the idea to use a total-variation penalization, with an extra linear constraint, on the weights of a generalized linear model trained on a binarization of the raw continuous features, leading to a procedure that selects multiple cut-points per feature, looking at all features simultaneously. Our approach therefore increases the capacity of the considered generalized linear model: several weights are used for the binarized features instead of a single one for the raw feature. This leads to a more flexible decision function compared to the linear one: when looking at the decision function as a function of a single raw feature, it is now piecewise constant instead of linear, as illustrated in Figure 2 below.

Organization of the paper. The proposed methodology is described in Section 2. Section 3 establishes an oracle inequality for generalized linear models and provides a convergence rate for our procedure in the particular case of a sparse additive model. Section 4 highlights the results of the method on various datasets and compares its performances to well known classification algorithms. Finally, we discuss the obtained results in Section 5.

Notations. Throughout the paper, for every $q > 0$, we denote by $\|v\|_q$ the usual ℓ_q-quasi norm of a vector $v \in \mathbb{R}^m$, namely $\|v\|_q = (\sum_{k=1}^m |v_k|^q)^{1/q}$, and $\|v\|_\infty = \max_{k=1,\dots,m} |v_k|$. We also denote $\|v\|_0 = |\{k : v_k \neq 0\}|$, where $|A|$ stands for the cardinality of a finite set A. For $u, v \in \mathbb{R}^m$, we denote by $u \odot v$ the Hadamard product $u \odot v = (u_1 v_1, \dots, u_m v_m)^\top$. For any $u \in \mathbb{R}^m$ and any $L \subset \{1, \dots, m\}$, we denote u_L as the vector in $\mathbb{R}^m$ satisfying $(u_L)_k = u_k$ for $k \in L$ and $(u_L)_k = 0$ for $k \in L^\complement = \{1, \dots, m\} \backslash L$. We write, for short, $\mathbf{1}$ (resp. $\mathbf{0}$) for the vector of $\mathbb{R}^m$ having all coordinates equal to one (resp. zero). Finally, we denote by $\mathrm{sign}(x)$ the set of subdifferentials of the function $x \mapsto |x|$, namely $\mathrm{sign}(x) = \{1\}$ if $x > 0$, $\mathrm{sign}(x) = \{-1\}$ if $x < 0$ and $\mathrm{sign}(0) = [-1, 1]$.

2. The proposed method

Consider a supervised training dataset $(x_i, y_i)_{i=1,\dots,n}$ containing features $x_i = [x_{i,1} \cdots x_{i,p}]^\top \in \mathbb{R}^p$ and labels $y_i \in \mathcal{Y} \subset \mathbb{R}$, that are independent and identically distributed samples of (X, Y) with unknown distribution $\mathbb{P}$. Let us denote $\boldsymbol{X} = [x_{i,j}]_{1 \le i \le n; 1 \le j \le p}$ the $n \times p$ features matrix vertically stacking the n samples of p raw features. Let $\boldsymbol{X}_{\bullet,j}$ be the j-th feature column of $\boldsymbol{X}$.

Binarization. The binarized matrix $\boldsymbol{X}^B$ is a matrix with an extended number $d > p$ of columns, where the j-th column $\boldsymbol{X}_{\bullet,j}$ is replaced by $d_j \ge 2$ columns $\boldsymbol{X}^B_{\bullet,j,1}, \dots, \boldsymbol{X}^B_{\bullet,j,d_j}$ containing only zeros and ones. Its i-th row is written

$$x_i^B = [x^B_{i,1,1} \cdots x^B_{i,1,d_1} x^B_{i,2,1} \cdots x^B_{i,2,d_2} \cdots x^B_{i,p,1} \cdots x^B_{i,p,d_p}]^\top \in \mathbb{R}^d,$$

where $d = \sum_{j=1}^p d_j$. In order to simplify the presentation of our results, we assume in the paper that all raw features $\boldsymbol{X}_{\bullet,j}$ are continuous, so that they are transformed using the following one-hot

encoding. For each raw feature j, we consider a partition of intervals $I_{j,1}, \dots, I_{j,d_j}$ of $\mathrm{range}(\boldsymbol{X}_{\bullet,j})$, namely satisfying $\cup_{k=1}^{d_j} I_{j,k} = \mathrm{range}(\boldsymbol{X}_{\bullet,j})$ and $I_{j,k} \cap I_{j,k'} = \varnothing$ for $k \neq k'$ and define

$$x^B_{i,j,k} = \begin{cases} 1 & \text{if } x_{i,j} \in I_{j,k}, \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \dots, n$, $j = 1, \dots, p$ and $k = 1, \dots, d_j$. An example is interquantiles intervals, namely $I_{j,1} = \big[q_j(0), q_j(\frac{1}{d_j})\big]$ and $I_{j,k} = \big(q_j(\frac{k-1}{d_j}), q_j(\frac{k}{d_j})\big]$ for $k = 2, \dots, d_j$, where $q_j(\alpha)$ denotes a quantile of order $\alpha \in [0, 1]$ for $\boldsymbol{X}_{\bullet,j}$. In practice, if there are ties in the estimated quantiles for a given feature, we simply choose the set of ordered unique values to construct the intervals. This principle of binarization is a well-known trick (Garcia et al., 2013), that allows to improve over the linear decision function with respect to the raw feature space: it uses a larger number of model weights, for each interval of values for the feature considered in the binarization. If training data contains also unordered qualitative features, one-hot encoding with ℓ_1-penalization can be used for instance.

Goodness-of-fit. Given a loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$, we consider the goodness-of-fit term

$$R_n(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, m_\theta(x_i)), \tag{1}$$

where $m_\theta(x_i) = \theta^\top x_i^B$ and $\theta \in \mathbb{R}^d$ where we recall that $d = \sum_{j=1}^p d_j$. We then have $\theta = [\theta_{1,\bullet}^\top \cdots \theta_{p,\bullet}^\top]^\top$, with $\theta_{j,\bullet}$ corresponding to the group of coefficients weighting the binarized raw j-th feature. We focus on generalized linear models (Green and Silverman, 1994), where the conditional distribution $Y|X = x$ is assumed to be from a one-parameter exponential family distribution with a density of the form

$$y|x \mapsto f^0(y|x) = \exp\Big(\frac{y m^0(x) - b(m^0(x))}{\phi} + c(y, \phi)\Big), \tag{2}$$

with respect to a reference measure which is either the Lebesgue measure (e.g. in the Gaussian case) or the counting measure (e.g. in the logistic or Poisson cases), leading to a loss function of the form

$$\ell(y_1, y_2) = -y_1 y_2 + b(y_2).$$

The density described in (2) encompasses several distributions, see Table 1. The functions $b(\cdot)$ and $c(\cdot)$ are known, while the natural parameter function $m^0(\cdot)$ is unknown. The dispersion parameter ϕ is assumed to be known in what follows. It is also assumed that $b(\cdot)$ is three times continuously differentiable. It is standard to notice that

$$\mathbb{E}[Y|X = x] = \int y f^0(y|x) dy = b'(m^0(x)),$$

where b' stands for the derivative of b. This formula explains how b' links the conditional expectation to the unknown m^0. The results given in Section 3 rely on the following Assumption.

Assumption 1 *Assume that b is three times continuously differentiable, that there is $C_b > 0$ such that $|b'''(z)| \le C_b |b''(z)|$ for any $z \in \mathbb{R}$ and that there exist constants $C_n > 0$ and $0 < L_n \le U_n$ such that $C_n = \max_{i=1,\dots,n} |m^0(x_i)| < \infty$ and $L_n \le \max_{i=1,\dots,n} b''\big(m^0(x_i)\big) \le U_n$.*

This assumption is satisfied for most standard generalized linear models. In Table 1, we list some standard examples that fit in this framework, see also van de Geer (2008) and Rigollet (2012).

Model	ϕ	$b(z)$	$b'(z)$	$b''(z)$	$b'''(z)$	C_b	L_n	U_n
Normal	σ^2	$\frac{z^2}{2}$	z	1	0	0	1	1
Logistic	1	$\log(1+e^z)$	$\frac{e^z}{1+e^z}$	$\frac{e^z}{(1+e^z)^2}$	$\frac{1-e^z}{1+e^z}b''(z)$	2	$\frac{e^{C_n}}{(1+e^{C_n})^2}$	$\frac{1}{4}$
Poisson	1	e^z	e^z	e^z	$b''(z)$	1	e^{-C_n}	e^{C_n}

Tab. 1: Examples of standard distributions that fit in the considered setting of generalized linear models, with the corresponding constants in Assumption 1.

Binarsity. Several problems occur when using the binarization trick described above:

(P1) The one-hot-encodings satisfy $\sum_{k=1}^{d_j} \boldsymbol{X}^B_{i,j,k} = 1$ for $j = 1, \ldots, p$, meaning that the columns of each block sum to $\mathbf{1}$, making $\boldsymbol{X}^B$ not of full rank by construction.

(P2) Choosing the number of intervals d_j for binarization of each raw feature j is not an easy task, as too many might lead to overfitting: the number of model-weights increases with each d_j, leading to a over-parametrized model.

(P3) Some of the raw features $\boldsymbol{X}_{\bullet,j}$ might not be relevant for the prediction task, so we want to select raw features from their one-hot encodings, namely induce block-sparsity in θ.

A usual way to deal with (P1) is to impose a linear constraint (Agresti, 2015) in each block. In order to do so, let us introduce first $n_{j,k} = |\{i : x_{i,j} \in I_{j,k}\}|$ and the vector $n_j = [n_{j,1} \cdots n_{j,d_j}] \in \mathbb{N}^{d_j}$. In our penalization term, we impose the linear constraint

$$n_j^\top \theta_{j,\bullet} = \sum_{k=1}^{d_j} n_{j,k}\theta_{j,k} = 0 \tag{3}$$

for all $j = 1, \ldots, p$. Note that if the $I_{j,k}$ are taken as interquantiles intervals, then for each j, we have that $n_{j,k}$ for $k = 1, \ldots, d_j$ are equal and the constraint (3) becomes the standard constraint $\sum_{k=1}^{d_j} \theta_{j,k} = 0$.

The trick to tackle (P2) is to remark that within each block, binary features are ordered. We use a within block total-variation penalization

$$\sum_{j=1}^{p} \|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}}$$

where

$$\|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}} = \sum_{k=2}^{d_j} \hat{w}_{j,k}|\theta_{j,k} - \theta_{j,k-1}|, \tag{4}$$

with weights $\hat{w}_{j,k} > 0$ to be defined later, to keep the number of different values taken by $\theta_{j,\bullet}$ to a minimal level.

Finally, dealing with (P3) is actually a by-product of dealing with (P1) and (P2). Indeed, if the raw feature j is not-relevant, then $\theta_{j,\bullet}$ should have all entries constant because of the penalization (4), and in this case all entries are zero, because of (3). We therefore introduce the following penalization, called *binarsity*

$$\mathrm{bina}(\theta) = \sum_{j=1}^{p} \Big(\sum_{k=2}^{d_j} \hat{w}_{j,k}|\theta_{j,k} - \theta_{j,k-1}| + \delta_j(\theta_{j,\bullet}) \Big) \tag{5}$$

where the weights $\hat{w}_{j,k} > 0$ are defined in Section 3 below, and where

$$\delta_j(u) = \begin{cases} 0 & \text{if} \quad n_j^\top u = 0, \\ \infty & \text{otherwise.} \end{cases} \tag{6}$$

We consider the goodness-of-fit (1) penalized by (5), namely

$$\hat{\theta} \in \mathrm{argmin}_{\theta \in \mathbb{R}^d} \big\{ R_n(\theta) + \mathrm{bina}(\theta) \big\}. \tag{7}$$

An important fact is that this optimization problem is numerically cheap, as explained in the next paragraph. Figure 1 illustrates the effect of the binarsity penalization with a varying strength on an example.

In Figure 2, we illustrate on a toy example, when $p = 2$, the decision boundaries obtained for logistic regression (LR) on raw features, LR on binarized features and LR on binarized features with the binarsity penalization.

Proximal operator of binarsity. The proximal operator and proximal algorithms are important tools for non-smooth convex optimization, with important applications in the field of supervised learning with structured sparsity (Bach et al., 2012). The proximal operator of a proper lower semi-continuous (Bauschke and Combettes, 2011) convex function $g : \mathbb{R}^d \to \mathbb{R}$ is defined by

$$\mathrm{prox}_g(v) \in \mathrm{argmin}_{u \in \mathbb{R}^d} \Big\{ \frac{1}{2}\|v - u\|_2^2 + g(u) \Big\}.$$

Proximal operators can be interpreted as generalized projections. Namely, if g is the indicator of a convex set $C \subset \mathbb{R}^d$ given by

$$g(u) = \delta_C(u) = \begin{cases} 0 & \text{if } u \in C, \\ \infty & \text{otherwise,} \end{cases}$$

then prox_g is the projection operator onto C. It turns out that the proximal operator of binarsity can be computed very efficiently, using an algorithm (Condat, 2013) that we modify in order to include weights $\hat{w}_{j,k}$. It applies in each group the proximal operator of the total-variation since binarsity penalization is block separable, followed by a simple projection onto $\mathrm{span}(n_j)^\perp$ the orthogonal of $\mathrm{span}(n_j)$, see Algorithm 1 below. We refer to Algorithm 2 in Section 6.2 for the weighted total-variation proximal operator.

Proposition 1 *Algorithm 1 computes the proximal operator of* $\mathrm{bina}(\theta)$ *given by* (5).

A proof of Proposition 1 is given in Section 6.1. Algorithm 1 leads to a very fast numerical routine, see Section 4. The next section provides a theoretical analysis of our algorithm with an oracle inequality for the prediction error, together with a convergence rate in the particular case of a sparse additive model.

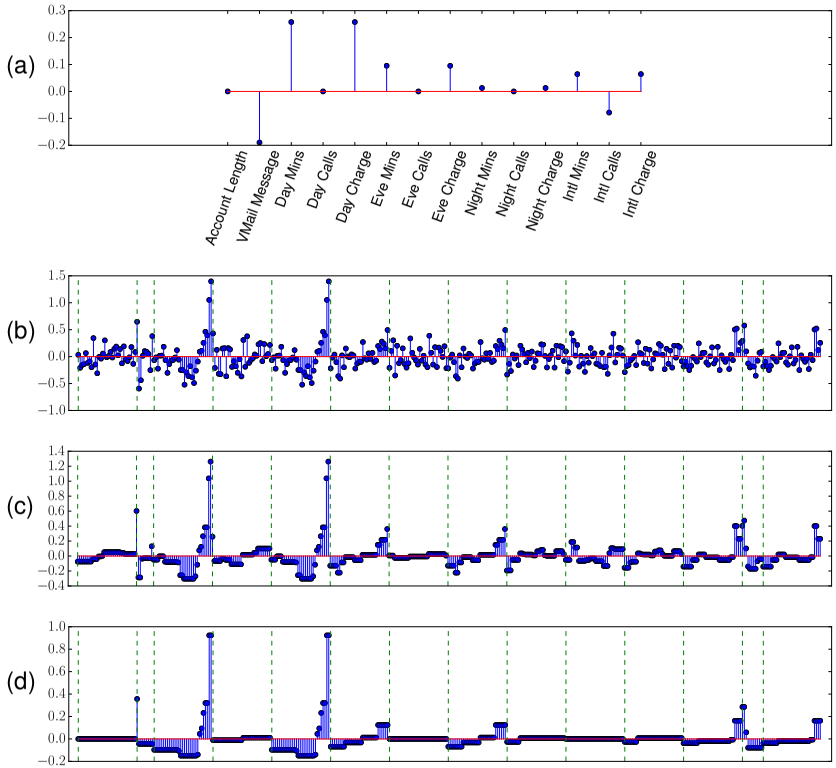


Fig. 1: Illustration of the binarsity penalization on the "Churn" dataset (see Section 4 for details) using logistic regression. Figure (a) shows the model weights learned by the Lasso method on the continuous raw features. Figure (b) shows the unpenalized weights on the binarized features, where the dotted green lines mark the limits between blocks corresponding to each raw features. Figures (c) and (d) show the weights with medium and strong binarsity penalization respectively. We observe in (c) that some significant cut-points start to be detected, while in (d) some raw features are completely removed from the model, the same features as those removed in (a).

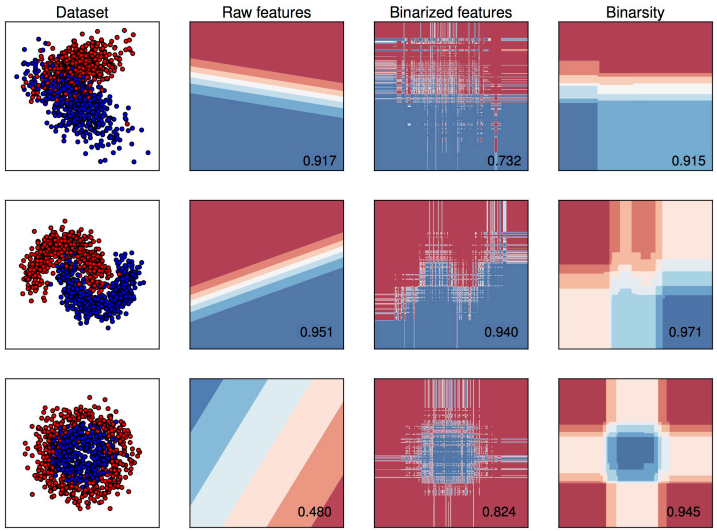


Fig. 2: Illustration of binarsity on 3 simulated toy datasets for binary classification with two classes (blue and red points). We set $n = 1000$, $p = 2$ and $d_1 = d_2 = 100$. In each row, we display the simulated dataset, followed by the decision boundaries for a logistic regression classifier trained on initial raw features, then on binarized features without regularization, and finally on binarized features with binarsity. The corresponding testing AUC score is given on the lower right corner of each figure. Our approach allows to keep an almost linear decision boundary in the first row, while a good decision boundaries are learned on the two other examples, which correspond to non-linearly separable datasets, without apparent overfitting.

Algorithm 1: Proximal operator of $\mathrm{bina}(\theta)$, see (5)

Input: vector $\theta \in \mathbb{R}^d$ and weights $\hat{w}_{j,k}$ and $n_{j,k}$ for $j = 1, \ldots, p$ and $k = 1, \ldots, d_j$
Output: vector $\eta = \mathrm{prox}_{\mathrm{bina}}(\theta)$
for $j = 1$ ***to*** p **do**
  $\beta_{j,\bullet} \leftarrow \mathrm{prox}_{\|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}}}(\theta_{j,\bullet})$ (TV-weighted prox in block j, see (4))
  $\eta_{j,\bullet} \leftarrow \beta_{j,\bullet} - \frac{n_j^\top \beta_{j,\bullet}}{\|n_j\|_2^2} n_j$ (projection onto $\mathrm{span}(n_j)^\perp$)
Return: η

3. Theoretical guarantees

We now investigate the statistical properties of (8) where the weights in the binarsity penalization have the form

$$\hat{w}_{j,k} = O\Big(\sqrt{\frac{\log d}{n}\,\hat{\pi}_{j,k}}\Big), \quad \text{with} \quad \hat{\pi}_{j,k} = \frac{\big|\{i = 1, \ldots, n : x_{i,j} \in \cup_{k'=k}^{d_j} I_{j,k'}\}\big|}{n}$$

for all $k \in \{2, \ldots, d_j\}$, see Theorem 2 for a precise definition of $\hat{w}_{j,k}$. Note that $\hat{\pi}_{j,k}$ corresponds to the proportion of ones in the sub-matrix obtained by deleting the first k columns in the j-th binarized block matrix $\boldsymbol{X}^B_{\bullet,j}$. In particular, we have $\hat{\pi}_{j,k} > 0$ for all j, k. We consider the risk measure defined by

$$R(m_\theta) = \frac{1}{n}\sum_{i=1}^{n}\big\{ -b'(m^0(x_i))m_\theta(x_i) + b(m_\theta(x_i))\big\},$$

which is standard with generalized linear models (van de Geer, 2008).

3.1. A general oracle inequality

We aim at evaluating how "close" to the minimal possible expected risk our estimated function $m_{\hat{\theta}}$ with $\hat{\theta}$ given by (8) is. To measure this closeness, we establish a non-asymptotic oracle inequality with a fast rate of convergence considering the excess risk of $m_{\hat{\theta}}$, namely $R(m_{\hat{\theta}}) - R(m^0)$. To derive this inequality, we consider for technical reasons the following problem instead of (7):

$$\hat{\theta} \in \operatorname{argmin}_{\theta \in B_d(\rho)} \big\{R_n(\theta) + \operatorname{bina}(\theta)\big\}, \tag{8}$$

where

$$B_d(\rho) = \Big\{\theta \in \mathbb{R}^d : \sum_{j=1}^{p} \|\theta_{j,\bullet}\|_\infty \le \rho\Big\}.$$

This constraint is standard in literature for the proof of oracle inequalities for sparse generalized linear models, see for instance van de Geer (2008), and is discussed in details below.

We also impose a restricted eigenvalue assumption on $\boldsymbol{X}^B$. For all $\theta \in \mathbb{R}^d$, let $J(\theta) = [J_1(\theta), \ldots, J_p(\theta)]$ be the concatenation of the support sets relative to the total-variation penalization, that is

$$J_j(\theta) = \{k = 2, \ldots, d_j \ : \ \theta_{j,k} \neq \theta_{j,k-1}\}.$$

Similarly, we denote $J^{\complement}(\theta) = \big[J^{\complement}_1(\theta), \ldots, J^{\complement}_p(\theta)\big]$ the complementary of $J(\theta)$. The restricted eigenvalue condition is defined as follow.

Assumption 2 *Let $K = [K_1, \ldots, K_p]$ be a concatenation of index sets such that*

$$\sum_{j=1}^{p} |K_j| \le J^\star, \tag{9}$$

where $J^\star$ is a positive integer. Define

$$\kappa(K) \in \inf_{u \in \mathscr{C}_{\mathrm{TV},\hat{w}}(K)\setminus\{\mathbf{0}\}} \left\{\frac{\|\boldsymbol{X}^B u\|_2}{\sqrt{n}\|u_K\|_2}\right\}$$

with

$$\mathscr{C}_{\mathrm{TV},\hat{w}}(K) = \Big\{u \in \mathbb{R}^d : \sum_{j=1}^{p} \|(u_{j,\bullet})_{K_j^{\complement}}\|_{\mathrm{TV},\hat{w}_{j,\bullet}} \le 2\sum_{j=1}^{p} \|(u_{j,\bullet})_{K_j}\|_{\mathrm{TV},\hat{w}_{j,\bullet}}\Big\}. \tag{10}$$

We assume that the following condition holds

$$\kappa(K) > 0 \tag{11}$$

for any K satisfying (9).

The set $\mathscr{C}_{\mathrm{TV},\hat{w}}(K)$ is a cone composed by all vectors with a support "close" to K. Theorem 2 gives a risk bound for the estimator $m_{\hat{\theta}}$.

Theorem 2 *Let Assumptions 1 and 2 be satisfied. Fix $A > 0$ and choose*

$$\hat{w}_{j,k} = \sqrt{\frac{2U_n\phi(A + \log d)}{n}\,\hat{\pi}_{j,k}}. \tag{12}$$

Then, with probability at least $1 - 2e^{-A}$, any $\hat{\theta}$ given by (8) *satisfies*

$$R(m_{\hat{\theta}}) - R(m^0) \le \inf_\theta \Big\{3(R(m_\theta) - R(m^0)) + \frac{2560(C_b(C_n + \rho) + 2)}{L_n\kappa^2(J(\theta))}\,|J(\theta)| \max_{j=1,\ldots,p} \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|^2_\infty\Big\},$$

where the infimum is over the set of vectors $\theta \in B_d(\rho)$ such that $n_j^\top \theta_{j,\bullet} = 0$ for all $j = 1, \ldots, p$ and such that $|J(\theta)| \le J^*$.

The proof of Theorem 2 is given in Section 6.3 below. Note that the "variance" term or "complexity" term in the oracle inequality satisfies

$$|J(\theta)| \max_{j=1,\ldots,p} \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|^2_\infty \le 2U_n\phi\frac{|J(\theta)|(A + \log d)}{n}. \tag{13}$$

The value $|J(\theta)|$ characterizes the sparsity of the vector θ, given by

$$|J(\theta)| = \sum_{j=1}^{p} |J_j(\theta)| = \sum_{j=1}^{p} |\{k = 1, \ldots, d_j : \theta_{j,k} \neq \theta_{j,k-1}\}|.$$

It counts the number of non-equal consecutive values of θ. If θ is block-sparse, namely whenever $|\mathcal{J}(\theta)| \ll p$ where $\mathcal{J}(\theta) = \{j = 1, \ldots, p : \theta_{j,\bullet} \neq 0_{d_j}\}$ (meaning that few raw features are useful for prediction), then $|J(\theta)| \le |\mathcal{J}(\theta)| \max_{j \in \mathcal{J}(\theta)} |J_j(\theta)|$, which means that $|J(\theta)|$ is controlled by the block sparsity $|\mathcal{J}(\theta)|$.

The oracle inequality from Theorem 2 is stated uniformly for vectors $\theta \in B_d(\rho)$ satisfying $n_j^\top \theta_{j,\bullet} = 0$ for all $j = 1, \ldots, p$ and $|J(\theta)| \le J^*$. Writing this oracle inequality under the assumption $|J(\theta)| \le J^*$ meets the standard way of stating sparse oracle inequalities, see e.g. Bühlmann and van De Geer (2011). Note that J^* is introduced in Assumption 2 and corresponds to a maximal sparsity for which the matrix $\boldsymbol{X}^B$ satisfies the restricted eigenvalue assumption. Also, the oracle inequality stated in Theorem 2 stands for vectors such that $n_j^\top \theta_{j,\bullet} = 0$, which is natural since the binarsity penalization imposes these extra linear constraints.

The assumption that $\theta \in B_d(\rho)$ is a technical one, that allows to establish a connection, via the notion of self-concordance, see Bach (2010), between the empirical squared ℓ_2-norm and the empirical Kullback divergence (see Lemma 9 in Section 6.3). It corresponds to a technical constraint which is commonly used in literature for the proof of oracle inequalities for sparse generalized linear models, see for instance van de Geer (2008), a recent contribution for the particular case of Poisson regression being Ivanoff et al. (2016). Also, note that

$$\max_{i=1,\ldots,n} |\langle x_i^B, \theta\rangle| \le \sum_{j=1}^{p} \|\theta_{j,\bullet}\|_\infty \le |\mathcal{J}(\theta)| \times \|\theta\|_\infty, \tag{14}$$

where $\|\theta\|_\infty = \max_{j=1,\dots,p} \|\theta_{j,\bullet}\|_\infty$. The first inequality in (14) comes from the fact that the entries of $\boldsymbol{X}^B$ are in $\{0,1\}$, and it entails that $\max_{i=1,\dots,n} |\langle x_i^B, \theta\rangle| \leq \rho$ whenever $\theta \in B_d(\rho)$. The second inequality in (14) entails that ρ can be upper bounded by $|\mathcal{J}(\theta)| \times \|\theta\|_\infty$, and therefore the constraint $\theta \in B_d(\rho)$ becomes only a box constraint on θ, which depends on the dimensionality of the features through $|\mathcal{J}(\theta)|$ only. The fact that the procedure depends on ρ, and that the oracle inequality stated in Theorem 2 depends linearly on ρ is commonly found in literature about sparse generalized linear models, see van de Geer (2008); Bach (2010); Ivanoff et al. (2016). However, the constraint $B_d(\rho)$ is a technicality which is not used in the numerical experiments provided in Section 4 below.

In the next Section, we exhibit a consequence of Theorem 2, whenever one considers the Gaussian case (least-squares loss) and where m^0 has a sparse additive structure defined below. This structure allows to control the bias term from Theorem 2 and to exhibit a convergence rate.

3.2. Sparse linear additive regression

Theorem 2 allows to study a particular case, namely an additive model, see e.g. Hastie and Tibshirani (1990); Horowitz et al. (2006) and in particular a sparse additive linear model, which is of particular interest in high-dimensional statistics, see Meier et al. (2009); Ravikumar et al. (2009); Bühlmann and van De Geer (2011). We prove in Theorem 3 below that our procedure matches the convergence rates previously known from literature. In this setting, we work under the following assumptions.

Assumption 3 *We assume to simplify that $x_i \in [0,1]^d$ for all $i = 1,\dots,n$. We consider the Gaussian setting with the least-squares loss, namely $\ell(y,y') = \frac{1}{2}(y-y')^2$, $b(y) = \frac{1}{2}y^2$ and $\phi = \sigma^2$ (noise variance) in Equation (2), with $L_n = U_n = 1$, $C_b = 0$ in Assumption 1. Moreover, we assume that m^0 has the following sparse additive structure*

$$m^0(x) = \sum_{j \in \mathcal{J}_*} m_j^0(x_j)$$

for $x = [x_1 \cdots x_p] \in \mathbb{R}^p$, where $m_j^0 : \mathbb{R} \to \mathbb{R}$ are L-Lipschitz functions, namely satisfying $|m_j^0(z) - m_j^0(z')| \leq L|z-z'|$ for any $z, z' \in \mathbb{R}$, and where $\mathcal{J}_* \subset \{1,\dots,p\}$ is a set of active features (sparsity means that $|\mathcal{J}_*| \ll p$). Also, we assume the following identifiability condition

$$\sum_{i=1}^n m_j^0(x_{i,j}) = 0$$

for all $j = 1,\dots,p$.

Assumption 3 contains identifiability and smoothness requirements that are standard when studying additive models, see e.g. Meier et al. (2009). We restrict the functions m_j^0 to be Lipschitz and not smoother, since our procedure produces a piecewise constant decision function with respect to each j, that can approximate optimally only Lipschitz functions. For more regular functions, our procedure would lead to suboptimal rates, see also the discussion below the statement of Theorem 3.

Theorem 3 *Consider procedure (7) with $d_j = D$, where D is the integer part of $n^{1/3}$, and $I_{j,1} = [0, \frac{1}{D}]$, $I_{j,k} = (\frac{k-1}{D}, \frac{k}{D}]$ for all $k = 2,\dots,D$ and $j = 1,\dots,p$, and keep the weights $\hat{w}_{j,k}$ the same as in Theorem 2. Introduce also $\theta^*_{j,k} = \sum_{i=1}^n m_j^0(x_{i,j}) \mathbf{1}_{I_k}(x_{i,j}) / \sum_{i=1}^n \mathbf{1}_{I_k}(x_{i,j})$ for $j \in \mathcal{J}_*$ and $\theta^*_{j,\bullet} = \mathbf{0}_D$ for $j \notin \mathcal{J}_*$. Then, under Assumption 2 with $J^* = J(\theta^*)$ and Assumption 3, we have*

$$\|m_{\hat{\theta}} - m^0\|_n^2 \leq \Big(3L^2|\mathcal{J}_*| + \frac{5120 M_n \sigma^2 (A + \log(pn^{1/3}M_n)}{\kappa^2(J(\theta^*))}\Big) \frac{|\mathcal{J}_*|}{n^{2/3}},$$

where $M_n = \max_{j=1,\dots,p} \max_{i=1,\dots,n} |m_j^0(x_{i,j})|$.

The proof of Theorem 3 is given in Section 6.8 below. It is an easy consequence of Theorem 2 under the sparse additive model assumption. It uses Assumption 2 with $J^* = J(\theta^*)$, since $\theta^*_{j,\bullet}$ is the minimizer of the bias for each $j \in \mathcal{J}_*$, see the proof of Theorem 3 for details.

The rate of convergence is, up to constants and logarithmic terms, of order $|\mathcal{J}_*|^2 n^{-2/3}$. Recalling that we work under a Lipschitz assumption, namely Hölder smoothness of order 1, the scaling of this rate w.r.t. to n is $n^{-2r/(2r+1)}$ with $r = 1$, which matches the one-dimensional minimax rate. This rate matches the one obtained in Bühlmann and van De Geer (2011), see Chapter 8 p. 272, where the rate $|\mathcal{J}_*|^2 n^{-2r/(2r+1)} = |\mathcal{J}_*|^2 n^{-4/5}$ is derived under a C^2 smoothness assumption, namely $r = 2$. Hence, Theorem 3 shows that, in the particular case of a sparse additive model, our procedure matches in terms of convergence rate the state of the art. Further improvements could consider more general smoothness (beyond Lipschitz) and adaptation with respect to the regularity, at the cost of a more complicated procedure which is beyond the scope of this paper.

4. Numerical experiments

In this section, we first illustrate the fact that the binarsity penalization is roughly only two times slower than basic ℓ_1-penalization, see the timings in Figure 3. We then compare binarsity to a large number of baselines, see Table 2, using 9 classical binary classification datasets obtained from the UCI Machine Learning Repository (Lichman, 2013), see Table 3.

For each method, we randomly split all datasets into a training and a test set (30% for testing), and all hyper-parameters are tuned on the training set using V-fold cross-validation with $V = 10$. For support vector machine with radial basis kernel (SVM), random forests (RF) and gradient boosting (GB), we use the reference implementations from the `scikit-learn` library (Pedregosa et al., 2011), and we use the `LogisticGAM` procedure from the `pygam` library[1] for the GAM baseline. The binarsity penalization is proposed in the `tick` library (Bacry et al., 2018), we provide sample code for its use in Figure 4. Logistic regression with no penalization or ridge penalization gave similar or lower scores for all considered datasets, and are therefore not reported in our experiments.

The binarsity penalization does not require a careful tuning of d_j (number of bins for the one-hot encoding of raw feature j). Indeed, past a large enough value, increasing d_j even further barely changes the results since the cut-points selected by the penalization do not change anymore. This is illustrated in Figure 5, where we observe that past 50 bins, increasing d_j even further does not affect the performance, and only leads to an increase of the training time. In all our experiments, we therefore fix $d_j = 50$ for $j = 1,\dots,p$.

The results of all our experiments are reported in Figures 6 and 7. In Figure 6 we compare the performance of binarsity with the baselines on all 9 datasets, using ROC curves and the Area Under the Curve (AUC), while we report computing (training) timings in Figure 7. We observe that binarsity consistently outperforms Lasso, as well as Group L1: this highlights the importance of the TV

1. `https://github.com/dswah/pyGAM`

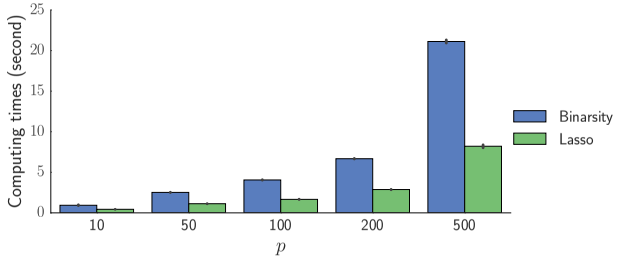


Fig. 3: Average computing time in second (with the black lines representing $\pm$ the standard deviation) obtained on 100 simulated datasets for training a logistic model with binarsity VS Lasso penalization, both trained on $\boldsymbol{X}^B$ with $d_j = 10$ for all $j \in 1, \ldots, p$. Features are Gaussian with a Toeplitz covariance matrix with correlation 0.5 and $n = 10000$. Note that the computing time ratio between the two methods stays roughly constant and equal to 2.

Name	Description	Reference
Lasso	Logistic regression (LR) with ℓ_1 penalization	Tibshirani (1996b)
Group L1	LR with group ℓ_1 penalization	Meier et al. (2008)
Group TV	LR with group total-variation penalization	
SVM	Support vector machine with radial basis kernel	Schölkopf and Smola (2002)
GAM	Generalized additive model	Hastie and Tibshirani (1990)
RF	Random forest classifier	Breiman (2001)
GB	Gradient boosting	Friedman (2002)

Tab. 2: Baselines considered in our experiments. Note that Group L1 and Group TV are considered on binarized features.

Dataset	#Samples	#Features	Reference
Ionosphere	351	34	Sigillito et al. (1989)
Churn	3333	21	Lichman (2013)
Default of credit card	30000	24	Yeh and Lien (2009)
Adult	32561	14	Kohavi (1996)
Bank marketing	45211	17	Moro et al. (2014)
Covertype	550088	10	Blackard and Dean (1999)
SUSY	5000000	18	Baldi et al. (2014)
HEPMASS	10500000	28	Baldi et al. (2016)
HIGGS	11000000	24	Baldi et al. (2014)

Tab. 3: Basic informations about the 9 considered datasets.


```python
# input: features X, labels y
from tick.inference import LogisticRegression
from tick.preprocessing import FeaturesBinarizer
from sklearn.model_selection import train_test_split

# binarize data
binarizer = FeaturesBinarizer(n_cuts=50)
X = binarizer.fit_transform(X)

# shuffle and split training and test sets
X, X_test, y, y_test = train_test_split(X, y, stratify=y)

# fit the model
learner = LogisticRegression(penalty='binarsity', C=1,
                             blocks_start=binarizer.blocks_start,
                             blocks_length=binarizer.blocks_length)
learner.fit(X, y)

# predict on test set
y_pred = learner.predict_proba(X_test)[:, 1]
```

Fig. 4: Sample python code for the use of binarsity with logistic regression in the `tick` library, with the use of the `FeaturesBinarizer` transformer for features binarization.

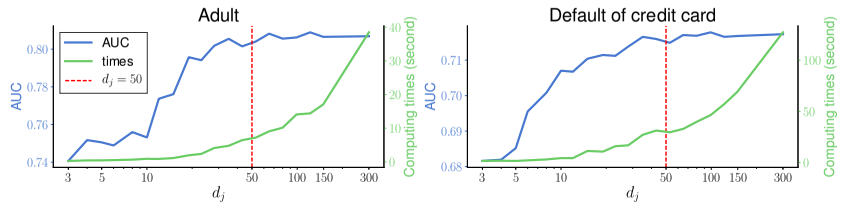


Fig. 5: Impact of the number of bins used in each block (d_j) on the classification performance (measured by AUC) and on the training time using the "Adult" and "Default of credit card" datasets. All d_j are equal for $j = 1, \ldots, p$, and we consider in all cases the best hyper-parameters selected after cross validation. We observe that past $d_j = 50$ bins, performance is roughly constant, while training time strongly increases.

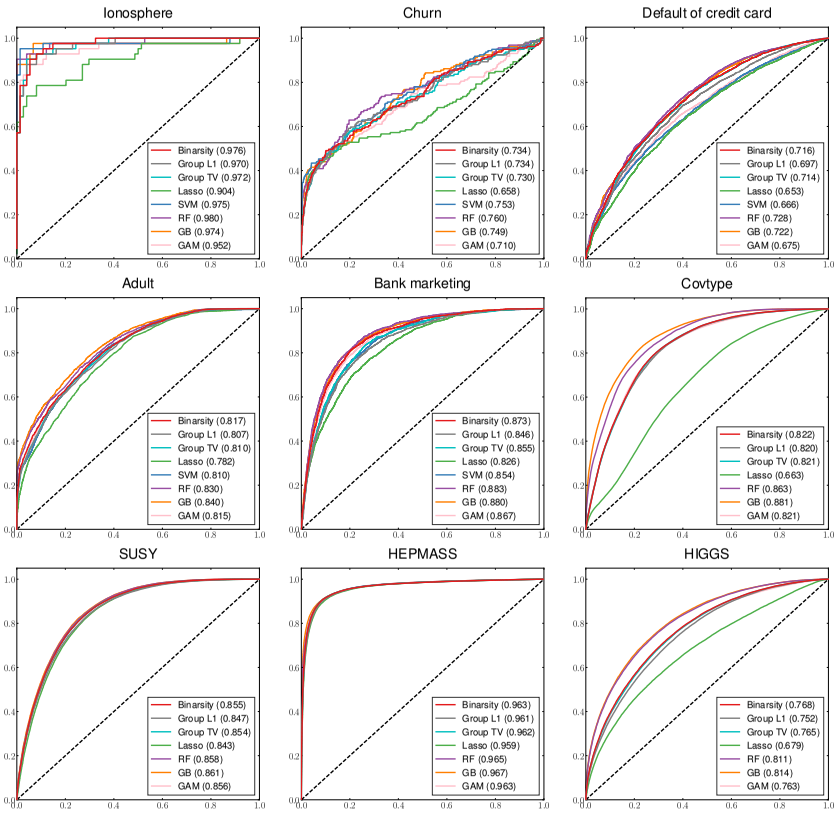


Fig. 6: Performance comparison using ROC curves and AUC scores (given between parenthesis) computed on test sets. The 4 last datasets contain too many examples for SVM (RBF kernel). Binarsity consistently does a better job than Lasso, Group L1, Group TV and GAM. Its performance is comparable to SVM, RF and GB but with computational timings that are orders of magnitude faster, see Figure 7.

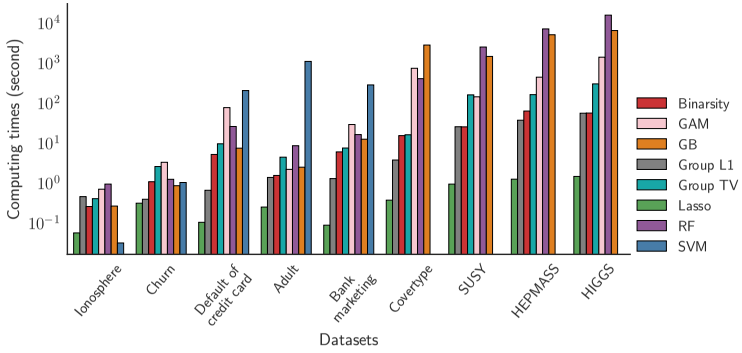


Fig. 7: Computing time comparisons (in seconds) between the methods on the considered datasets. Note that the time values are log-scaled. These timings concern the learning task for each model with the best hyper parameters selected, after the cross validation procedure. The 4 last datasets contain too many examples for the SVM with RBF kernel to be trained in a reasonable time. Roughly, binarsity is between 2 and 5 times slower than ℓ_1 penalization on the considered datasets, but is more than 100 times faster than random forests or gradient boosting algorithms on large datasets, such as HIGGS.

norm within each group. The AUC of Group TV is always slightly below the one of binarsity, and more importantly it involves a much larger training time: convergence is slower for Group TV, since it does not use the linear constraint of binarsity, leading to a ill-conditioned problem (sum of binary features equals 1 in each block). Finally, binarsity outperforms also GAM and its performance is comparable in all considered examples to RF and GB, with computational timings that are orders of magnitude faster, see Figure 7. All these experiments illustrate that binarsity achieves an extremely competitive compromise between computational time and performance, compared to all considered baselines.

5. Conclusion

In this paper, we introduced the binarsity penalization for one-hot encodings of continuous features. We illustrated the good statistical properties of binarsity for generalized linear models by proving non-asymptotic oracle inequalities. We conducted extensive comparisons of binarsity with state-of-the-art algorithms for binary classification on several standard datasets. Experimental results illustrate that binarsity significantly outperforms Lasso, Group L1 and Group TV penalizations and also generalized additive models, while being competitive with random forests and boosting. Moreover, it can be trained orders of magnitude faster than boosting and other ensemble methods. Even more importantly, it provides interpretability. Indeed, in addition to the raw feature selection ability of binarsity, the method pinpoints significant cut-points for all continuous feature. This leads to a much more precise and deeper understanding of the model than the one provided by Lasso on raw features. These results illustrate the fact that binarsity achieves an extremely competitive compromise between computational time and performance, compared to all considered baselines.

6. Proofs

In this Section we gather the proofs of all the theoretical results proposed in the paper. Throughout this Section, we denote by $\partial(\phi)$ the subdifferential mapping of a convex function ϕ.

6.1. Proof of Proposition 1

Recall that the indicator function δ_j is given by (6). For any fixed $j = 1, \ldots, p$, we prove that $\mathrm{prox}_{\|\cdot\|_{\mathrm{TV},\hat{w}_{j,\bullet}}+\delta_j}$ is the composition of $\mathrm{prox}_{\|\cdot\|_{\mathrm{TV},\hat{w}_{j,\bullet}}}$ and prox_{δ_j}, namely

$$\mathrm{prox}_{\|\cdot\|_{\mathrm{TV},\hat{w}_{j,\bullet}}+\delta_j}(\theta_{j,\bullet}) = \mathrm{prox}_{\delta_j}\left(\mathrm{prox}_{\|\cdot\|_{\mathrm{TV},\hat{w}_{j,\bullet}}}(\theta_{j,\bullet})\right)$$

for all $\theta_{j,\bullet} \in \mathbb{R}^{d_j}$. Using Theorem 1 in Yu (2013), it is sufficient to show that for all $\theta_{j,\bullet} \in \mathbb{R}^{d_j}$, we have

$$\partial\big(\|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}}\big) \subseteq \partial\big(\|\,\mathrm{prox}_{\delta_j}(\theta_{j,\bullet})\|_{\mathrm{TV},\hat{w}_{j,\bullet}}\big). \tag{15}$$

We have $\mathrm{prox}_{\delta_j}(\theta_{j,\bullet}) = \Pi_{\mathrm{span}\{n_j\}^\perp}(\theta_{j,\bullet})$, where $\Pi_{\mathrm{span}\{n_j\}^\perp}(\cdot)$ stands for the projection onto the orthogonal of $\mathrm{span}\{n_j\}$. This projection simply writes

$$\Pi_{\mathrm{span}\{n_j\}^\perp}(\theta_{j,\bullet}) = \theta_{j,\bullet} - \frac{n_j^\top \theta_{j,\bullet}}{\|n_j\|_2^2} n_j$$

Now, let us define the $d_j \times d_j$ matrix D_j by

$$D_j = \begin{bmatrix} 1 & 0 & & 0 \\ -1 & 1 & & \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{d_j} \times \mathbb{R}^{d_j}. \tag{16}$$

We then remark that for all $\theta_{j,\bullet} \in \mathbb{R}^{d_j}$,

$$\|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}} = \sum_{k=2}^{d_j} \hat{w}_{j,k}|\theta_{j,k} - \theta_{j,k-1}| = \|\hat{w}_{j,\bullet} \odot D_j\theta_{j,\bullet}\|_1. \tag{17}$$

Using subdifferential calculus (see details in the proof of Proposition 5 below), one has

$$\partial\big(\|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}}\big) = \partial\big(\|\hat{w}_{j,\bullet} \odot D_j\theta_{j,\bullet}\|_1\big) = {D_j}^\top \hat{w}_{j,\bullet} \odot \mathrm{sign}(D_j\theta_{j,\bullet}).$$

Then, the linear constraint $n_j^\top \theta_{j,\bullet} = 0$ entails

$${D_j}^\top \hat{w}_{j,\bullet} \odot \mathrm{sign}(D_j\theta_{j,\bullet}) = {D_j}^\top \hat{w}_{j,\bullet} \odot \mathrm{sign}\left(D_j\big(\theta_{j,\bullet} - \frac{n_j^\top \theta_{j,\bullet}}{\|n_j\|_2^2} n_j\big)\right),$$

which leads to (15) and concludes the proof of the Proposition. □

6.2. Proximal operator of the weighted TV penalization

We recall in Algorithm 2 an algorithm provided in Alaya et al. (2015) for the computation of the proximal operator of the weighted total-variation penalization

$$\beta = \mathrm{prox}_{\|\cdot\|_{\mathrm{TV},\hat{w}}}(\theta) \in \mathrm{argmin}_{\theta\in\mathbb{R}^m}\left\{\frac{1}{2}\|\beta - \theta\|_2^2 + \|\theta\|_{\mathrm{TV},\hat{w}}\right\}. \tag{18}$$

A quick explanation of this algorithm is as follows. The algorithm runs forwardly through the input vector $(\theta_1, \ldots, \theta_m)$. Using Karush-Kuhn-Tucker (KKT) optimality conditions (Boyd and Vandenberghe, 2004), we have that at a location k, the weight β_k stays constant whenever $|u_k| < \hat{w}_{k+1}$, where u_k is a solution to a dual problem associated to the primal problem (18). If not possible, it goes back to the last location where a jump can be introduced in β, validates the current segment until this location, starts a new segment, and continues.

6.3. Proof of Theorem 2

The proof relies on several technical properties that are described below. From now on, we consider $\boldsymbol{y} = [y_1 \cdots y_n]^\top$, $\boldsymbol{X} = [x_1 \cdots x_n]^\top$, $m^0(\boldsymbol{X}) = [m^0(x_1) \cdots m^0(x_n)]^\top$, and recalling that $m_\theta(x_i) = \theta^\top x_i^B)$ we introduce $m_\theta(\boldsymbol{X}) = [m_\theta(x_1) \cdots m_\theta(x_n)]^\top$ and $b'(m_\theta(\boldsymbol{X})) = [b'(m_\theta(x_1)) \cdots b'(m_\theta(x_n))]^\top$.

Let us now define the Kullback-Leibler divergence between the true probability density funtion f^0 defined in (2) and a candidate f_θ within the generalized linear model $f_\theta(y|x) = \exp\big(ym_\theta(x) - b(m_\theta(x))\big)$ as follows

$$\begin{aligned}\mathrm{KL}_n(f^0, f_\theta) &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\mathbb{P}_{\boldsymbol{y}|X}}\Big[\log\frac{f^0(y_i|x_i)}{f_\theta(y_i|x_i)}\Big] \\ &:= \mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),\end{aligned}$$

where $\mathbb{P}_{\boldsymbol{y}|X}$ is the joint distribution of $\boldsymbol{y}$ given $\boldsymbol{X}$. We then have the following Lemma.

Lemma 4 *The excess risk satisfies*

$$R(m_\theta) - R(m^0) = \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),$$

where we recall that ϕ is the dispertion parameter of the generalized linear model, see (2).

Proof. If follows from the following simple computation

$$\begin{aligned}&\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})) \\ &\quad= \phi^{-1}\frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\mathbb{P}_{\boldsymbol{y}|X}}\Big[\big(-y_i m_\theta(x_i) + b(m_\theta(x_i))\big) - \big(-y_i m^0(x_i) + b(m^0(x_i))\big)\Big] \\ &\quad= \phi^{-1}\big(R(m_\theta) - R(m^0)\big)\end{aligned}$$

which proves the Lemma. □

Algorithm 2: Proximal operator of weighted TV penalization

Input: vector $\theta = \big(\theta_1, \ldots, \theta_m\big)^\top \in \mathbb{R}^m$ and weights $\hat{w} = (\hat{w}_1, \ldots, \hat{w}_m) \in \mathbb{R}^m_+$.
Output: vector $\beta = \mathrm{prox}_{\|\cdot\|_{\mathrm{TV},\hat{w}}}(\theta)$

1. **Set** $k = k_0 = k_- = k_+ \leftarrow 1$
   $\beta_{\min} \leftarrow \theta_1 - \hat{w}_2$; $\beta_{\max} \leftarrow \theta_1 + \hat{w}_2$
   $u_{\min} \leftarrow \hat{w}_2$; $u_{\max} \leftarrow -\hat{w}_2$
2. **if** $k = m$ **then**
   - $\beta_m \leftarrow \beta_{\min} + u_{\min}$
3. **if** $\theta_{k+1} + u_{\min} < \beta_{\min} - \hat{w}_{k+2}$ **then** `/* negative jump */`
   - $\beta_{k_0} = \cdots = \beta_{k_-} \leftarrow \beta_{\min}$
   - $k = k_0 = k_- = k_+ \leftarrow k_- + 1$
   - $\beta_{\min} \leftarrow \theta_k - \hat{w}_{k+1} + \hat{w}_k$; $\beta_{\max} \leftarrow \theta_k + \hat{w}_{k+1} + \hat{w}_k$
   - $u_{\min} \leftarrow \hat{w}_{k+1}$; $u_{\max} \leftarrow -\hat{w}_{k+1}$
4. **else if** $\theta_{k+1} + u_{\max} > \beta_{\max} + \hat{w}_{k+2}$ **then** `/* positive jump */`
   - $\beta_{k_0} = \ldots = \beta_{k_+} \leftarrow \beta_{\max}$
   - $k = k_0 = k_- = k_+ \leftarrow k_+ + 1$
   - $\beta_{\min} \leftarrow \theta_k - \hat{w}_{k+1} - \hat{w}_k$; $\beta_{\max} \leftarrow \theta_k + \hat{w}_{k+1} - \hat{w}_k$
   - $u_{\min} \leftarrow \hat{w}_{k+1}$; $u_{\max} \leftarrow -\hat{w}_{k+1}$
5. **else** `/* no jump */`
   - **set** $k \leftarrow k + 1$
     $u_{\min} \leftarrow \theta_k + \hat{w}_{k+1} - \beta_{\min}$
     $u_{\max} \leftarrow \theta_k - \hat{w}_{k+1} - \beta_{\max}$ **if** $u_{\min} \geq \hat{w}_{k+1}$ **then**
     - $\beta_{\min} \leftarrow \beta_{\min} + \frac{u_{\min} - \hat{w}_{k+1}}{k - k_0 + 1}$
     - $u_{\min} \leftarrow \hat{w}_{k+1}$
     - $k_- \leftarrow k$
   - **if** $u_{\max} \leq -\hat{w}_{k+1}$ **then**
     - $\beta_{\max} \leftarrow \beta_{\max} + \frac{u_{\max} + \hat{w}_{k+1}}{k - k_0 + 1}$
     - $u_{\max} \leftarrow -\hat{w}_{k+1}$
     - $k_+ \leftarrow k$
6. **if** $k < m$ **then**
   - go to **3.**
7. **if** $u_{\min} < 0$ **then**
   - $\beta_{k_0} = \cdots = \beta_{k_-} \leftarrow \beta_{\min}$
   - $k = k_0 = k_- \leftarrow k_- + 1$
   - $\beta_{\min} \leftarrow \theta_k - \hat{w}_{k+1} + \hat{w}_k$
   - $u_{\min} \leftarrow \hat{w}_{k+1}$; $u_{\max} \leftarrow \theta_k + \hat{w}_k - v_{\max}$
   - go to **2.**
8. **else if** $u_{\max} > 0$ **then**
   - $\beta_{k_0} = \cdots = \beta_{k_+} \leftarrow \beta_{\max}$
   - $k = k_0 = k_+ \leftarrow k_+ + 1$
   - $\beta_{\max} \leftarrow \theta_k + \hat{w}_{k+1} - \hat{w}_k$
   - $u_{\max} \leftarrow -\hat{w}_{k+1}$; $u_{\min} \leftarrow \theta_k - \hat{w}_k - u_{\min}$
   - go to **2.**
9. **else**
   - $\beta_{k_0} = \cdots = \beta_m \leftarrow \beta_{\min} + \frac{u_{\min}}{k - k_0 + 1}$

6.4. Optimality conditions

As explained in the following Proposition, a solution to problem (8) can be characterized using the Karush-Kuhn-Tucker (KKT) optimality conditions (Boyd and Vandenberghe, 2004).

Proposition 5 *A vector* $\hat{\theta} = [\hat{\theta}_{1,\bullet}^\top \cdots \hat{\theta}_{p,\bullet}^\top]^\top \in \mathbb{R}^d$ *is an optimum of the objective function* (8) *if and only if there are subgradients* $\hat{h} = [\hat{h}_{j,\bullet}]_{j=1,\ldots,p} \in \partial\|\hat{\theta}\|_{\mathrm{TV},\hat{w}}$ *and* $\hat{g} = [\hat{g}_{j,\bullet}]_{j=1,\ldots,p} \in \partial[\delta_j(\hat{\theta}_{j,\bullet})]_{j=1,\ldots,p}$ *such that*

$$\nabla R_n(\hat{\theta}_{j,\bullet}) + \hat{h}_{j,\bullet} + \hat{g}_{j,\bullet} = \mathbf{0},$$

where

$$\begin{cases} \hat{h}_{j,\bullet} = D_j^\top\big(\hat{w}_{j,\bullet} \odot \mathrm{sign}(D_j\hat{\theta}_{j,\bullet})\big) & \text{if } j \in J(\hat{\theta}), \\ \hat{h}_{j,\bullet} \in D_j^\top\big(\hat{w}_{j,\bullet} \odot [-1,+1]^{d_j}\big) & \text{if } j \in J^{\complement}(\hat{\theta}), \end{cases} \tag{19}$$

and where we recall that $J(\hat{\theta})$ *is the support set of* $\hat{\theta}$*. The subgradient* $\hat{g}_{j,\bullet}$ *belongs to*

$$\partial\big(\delta_j(\hat{\theta}_{j,\bullet})\big) = \big\{\mu_{j,\bullet} \in \mathbb{R}^{d_j} : \mu_{j,\bullet}^\top\theta_{j,\bullet} \leq \mu_{j,\bullet}^\top\hat{\theta}_{j,\bullet} \text{ for all } \theta_{j,\bullet} \text{ such that } n_j^\top\theta_{j,\bullet} = 0\big\}.$$

For the generalized linear model, we have

$$\frac{1}{n}(\boldsymbol{X}_{\bullet,j}^B)^\top\big(b'(m_{\hat{\theta}}(\boldsymbol{X})) - \boldsymbol{y}\big) + \hat{h}_{j,\bullet} + \hat{g}_{j,\bullet} + \hat{f}_{j,\bullet} = \mathbf{0}, \tag{20}$$

where $\hat{f} = [\hat{f}_{j,\bullet}]_{j=1,\ldots,p}$ *belongs to the normal cone of the ball* $B_d(\rho)$*.*

Proof. The function $\theta \mapsto R_n(\theta)$ is differentiable, so the subdifferential of $R_n(\cdot) + \mathrm{bina}(\cdot)$ at a point $\theta = (\theta_{j,\bullet})_{j=1,\ldots,p} \in \mathbb{R}^d$ is given by

$$\partial(R_n(\theta) + \mathrm{bina}(\theta)) = \nabla R_n(\theta) + \partial(\mathrm{bina}(\theta)),$$

where $\nabla R_n(\theta) = \Big[\frac{\partial R_n(\theta)}{\partial\theta_{1,\bullet}} \cdots \frac{\partial R_n(\theta)}{\partial\theta_{p,\bullet}}\Big]^\top$ and

$$\partial\,\mathrm{bina}(\theta) = \Big[\partial\|\theta_{1,\bullet}\|_{\mathrm{TV},\hat{w}_{1,\bullet}} + \partial\delta_j(\theta_{1,\bullet}) \;\cdots\; \partial\|\theta_{p,\bullet}\|_{\mathrm{TV},\hat{w}_{p,\bullet}} + \partial\delta_j(\theta_{p,\bullet})\Big]^\top.$$

We have $\|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}} = \|\hat{w}_{j,\bullet} \odot D_j\theta_{j,\bullet}\|_1$ for all $j = 1, \ldots, p$. Then, by applying some properties of the subdifferential calculus, we get

$$\partial\|\theta_{j,\bullet}\|_{\mathrm{TV},\hat{w}_{j,\bullet}} = \begin{cases} D_j^\top\,\mathrm{sign}(\hat{w}_{j,\bullet} \odot D_j\theta_{j,\bullet}) & \text{if } D_j\theta \neq \mathbf{0}, \\ D_j^\top\big(\hat{w}_{j,\bullet} \odot v_j\big) & \text{otherwise}, \end{cases} \tag{21}$$

where $v_j \in [-1,+1]^{d_j}$ for all $j = 1, \ldots, p$. For generalized linear models, we rewrite

$$\hat{\theta} \in \mathrm{argmin}_{\theta\in\mathbb{R}^d}\big\{R_n(\theta) + \mathrm{bina}(\theta) + \delta_{B_d(\rho)}(\theta)\big\}, \tag{22}$$

where $\delta_{B_d(\rho)}$ is the indicator function of $B_d(\rho)$. Now, $\hat{\theta} = [\hat{\theta}_{1,\bullet}^\top \cdots \hat{\theta}_{p,\bullet}^\top]^\top$ is an optimum of (22) if and only if $\mathbf{0} \in \nabla R_n(m_{\hat{\theta}}) + \partial\|\hat{\theta}\|_{\mathrm{TV},\hat{w}} + \partial\delta_{B_d(\rho)}(\hat{\theta})$. Recall that the subdifferential of $\delta_{B_d(\rho)}(\cdot)$ is the normal cone of $B_d(\rho)$, namely

$$\partial\delta_{B_d(\rho)}(\hat{\theta}) = \big\{\eta \in \mathbb{R}^d : \eta^\top\theta \leq \eta^\top\hat{\theta} \text{ for all } \theta \in B_d(\rho)\big\}. \tag{23}$$

One has

$$\frac{\partial R_n(\theta)}{\partial\theta_{j,\bullet}} = \frac{1}{n}(\boldsymbol{X}_{\bullet,j}^B)^\top(b'(m_{\hat{\theta}}(\boldsymbol{X})) - \boldsymbol{y}), \tag{24}$$

so that together with (24) and (23) we obtain (20), which concludes the proof of Proposition 5. □

6.5. Compatibility conditions

Let us define the block diagonal matrix $\mathbf{D} = \mathrm{diag}(D_1, \ldots, D_p)$ with D_j defined in (16). We denote its inverse T_j which is defined by the $d_j \times d_j$ lower triangular matrix with entries $(T_j)_{r,s} = 0$ if $r < s$ and $(T_j)_{r,s} = 1$ otherwise. We set $\mathbf{T} = \mathrm{diag}(T_1, \ldots, T_p)$, so that one has $\mathbf{D}^{-1} = \mathbf{T}$.

In order to prove Theorem 2, we need the following results which give a compatibility property (van de Geer, 2008; van de Geer and Lederer, 2013; Dalalyan et al., 2017) for the matrix $\mathbf{T}$, see Lemma 6 below and for the matrix $\boldsymbol{X}^B\mathbf{T}$, see Lemma 7 below. For any concatenation of subsets $K = [K_1, \ldots, K_p]$, we set

$$K_j = \{\tau_j^1, \ldots, \tau_j^{b_j}\} \subset \{1, \ldots, d_j\} \tag{25}$$

for all $j = 1, \ldots, p$ with the convention that $\tau_j^0 = 0$ and $\tau_j^{b_j+1} = d_j + 1$.

Lemma 6 *Let $\gamma \in \mathbb{R}_+^d$ be given and $K = [K_1, \ldots, K_p]$ with K_j given by* (25) *for all $j = 1, \ldots, p$. Then, for every $u \in \mathbb{R}^d \backslash \{\mathbf{0}\}$, we have*

$$\frac{\|\mathbf{T}u\|_2}{|\|u_K \odot \gamma_K\|_1 - \|u_{K^\complement} \odot \gamma_{K^\complement}\|_1|} \geq \kappa_{\mathbf{T},\gamma}(K),$$

where

$$\kappa_{\mathbf{T},\gamma}(K) = \left\{ 32 \sum_{j=1}^{p} \sum_{k=1}^{d_j} |\gamma_{j,k+1} - \gamma_{j,k}|^2 + 2|K_j| \|\gamma_{j,\bullet}\|_\infty^2 \Delta_{\min,K_j}^{-1} \right\}^{-1/2},$$

and $\Delta_{\min,K_j} = \min_{r=1,\ldots b^j} |\tau_j^{r_j} - \tau_j^{r_j-1}|$.

Proof. Using Proposition 3 in Dalalyan et al. (2017), we have

$$\begin{aligned}
&\|u_K \odot \gamma_K\|_1 - \|u_{K^\complement} \odot \gamma_{K^\complement}\|_1 \\
&\quad = \sum_{j=1}^{p} \|u_{K_j} \odot \gamma_{K_j}\|_1 - \|u_{K_j^\complement} \odot \gamma_{K_j^\complement}\|_1 \\
&\quad \leq \sum_{j=1}^{p} 4\|T_j u_{j,\bullet}\|_2 \left\{ 2 \sum_{k=1}^{d_j} |\gamma_{j,k+1} - \gamma_{j,k}|^2 + 2(b_j+1)\|\gamma_{j,\bullet}\|_\infty^2 \Delta_{\min,K_j}^{-1} \right\}^{1/2}.
\end{aligned}$$

Using Hölder's inequality for the right hand side of the last inequality gives

$$\begin{aligned}
&\|u_K \odot \gamma_K\|_1 - \|u_{K^\complement} \odot \gamma_{K^\complement}\|_1 \\
&\quad \leq \|\mathbf{T}u\|_2 \left\{ 32 \sum_{j=1}^{p} \sum_{k=1}^{d_j} |\gamma_{j,k+1} - \gamma_{j,k}|^2 + 2|K_j| \|\gamma_{j,\bullet}\|_\infty^2 \Delta_{\min,K_j}^{-1} \right\}^{1/2},
\end{aligned}$$

which completes the proof of the Lemma. □

Combining Assumption 2 and Lemma 6 allows to establish a compatibility condition satisfied by $\boldsymbol{X}^B\mathbf{T}$.

Lemma 7 *Let $\gamma \in \mathbb{R}_+^d$ be given and $K = [K_1, \ldots, K_p]$ with K_j given by* (25) *for $j = 1, \ldots, p$. Then, if Assumption 2 holds, one has*

$$\inf_{u \in \mathscr{C}_{1,\hat{w}}(K) \backslash \{\mathbf{0}\}} \left\{ \frac{\|\boldsymbol{X}^B\mathbf{T}u\|_2}{\sqrt{n}\ |\ \|u_K \odot \gamma_K\|_1 - \|u_{K^\complement} \odot \gamma_{K^\complement}\|_1|} \right\} \geq \kappa_{\mathbf{T},\gamma}(K)\kappa(K), \tag{26}$$

where

$$\mathscr{C}_{1,\hat{w}}(K) = \left\{ u \in \mathbb{R}^d : \sum_{j=1}^{p} \|(u_{j,\bullet})_{K_j^\complement}\|_{1,\hat{w}_{j,\bullet}} \leq 2 \sum_{j=1}^{p} \|(u_{j,\bullet})_{K_j}\|_{1,\hat{w}_{j,\bullet}} \right\}. \tag{27}$$

Proof. Lemma 6 gives

$$\frac{\|\boldsymbol{X}^B\mathbf{T}u\|_2}{\sqrt{n}|\|u_K \odot \gamma_K\|_1 - \|u_{K^\complement} \odot \gamma_{K^\complement}\|_1|} \geq \kappa_{\mathbf{T},\gamma}(K) \frac{\|\boldsymbol{X}^B\mathbf{T}u\|_2}{\sqrt{n}\|\mathbf{T}u\|_2}.$$

Now, we note that if $u \in \mathscr{C}_{1,\hat{w}}(K)$, then $\mathbf{T}u \in \mathscr{C}_{\mathrm{TV},\hat{w}}(K)$. Hence, Assumption 2 entails

$$\frac{\|\boldsymbol{X}^B\mathbf{T}u\|_2}{\sqrt{n}|\|u_K \odot \gamma_K\|_1 - \|u_{K^\complement} \odot \gamma_{K^\complement}\|_1|} \geq \kappa_{\mathbf{T},\gamma}(K)\kappa(K),$$

which concludes the proof of the Lemma. □

6.6. Connection between the empirical Kullback-Leibler divergence and the empirical squared norm

The next Lemma is from Bach (2010) (see Lemma 1 herein).

Lemma 8 *Let $\varphi : \mathbb{R} \to \mathbb{R}$ be a three times differentiable convex function such that for all $t \in \mathbb{R}$, $|\varphi'''(t)| \leq M|\varphi''(t)|$ for some $M \geq 0$. Then, for all $t \geq 0$, one has*

$$\frac{\varphi''(0)}{M^2}\psi(-Mt) \leq \varphi(t) - \varphi(0) - \varphi'(0)t \leq \frac{\varphi''(0)}{M^2}\psi(Mt),$$

with $\psi(u) = e^u - u - 1$.

This Lemma entails the following in our setting.

Lemma 9 *Under Assumption 1, one has*

$$\begin{aligned}
\frac{L_n\psi(-2(C_n+\rho))}{4\phi(C_n+\rho)^2}\frac{1}{n}\|m^0(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\|_2^2 &\leq \mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})), \\
\frac{U_n\psi(2(C_n+\rho))}{4\phi(C_n+\rho)^2}\frac{1}{n}\|m^0(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\|_2^2 &\geq \mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),
\end{aligned}$$

for all $\theta \in B_d(\rho)$.

Proof. Let us consider the function $G_n : \mathbb{R} \to \mathbb{R}$ defined by $G_n(t) = R_n(m^0 + tm_\eta)$, with m_η to be defined later, which writes

$$G_n(t) = \frac{1}{n}\sum_{i=1}^{n} b(m^0(x_i) + tm_\eta(x_i)) - \frac{1}{n}\sum_{i=1}^{n} y_i(m^0(x_i) + tm_\eta(x_i)).$$

We have

$$G_n'(t) = \frac{1}{n}\sum_{i=1}^n m_\eta(x_i) b'(m^0(x_i) + t m_\eta(x_i)) - \frac{1}{n}\sum_{i=1}^n y_i m_\eta(x_i),$$
$$G_n''(t) = \frac{1}{n}\sum_{i=1}^n m_\eta^2(x_i) b''(m^0(x_i) + t m_\eta(x_i)),$$
$$\text{and} \quad G_n'''(t) = \frac{1}{n}\sum_{i=1}^n m_\eta^3(x_i) b'''(m^0(x_i) + t m_\eta(x_i)).$$

Using Assumption 1, we have $|G_n'''(t)| \leq C_b \|m_\eta\|_\infty |G_n''(t)|$ where $\|m_\eta\|_\infty := \max_{i=1,\dots,n} |m_\eta(x_i)|$. Lemma 8 with $M = C_b\|m_\eta\|_\infty$ gives

$$G_n''(0)\frac{\psi(-C_b\|m_\eta\|_\infty t)}{C_b^2\|m_\eta\|_\infty^2} \leq G_n(t) - G_n(0) - tG_n'(0) \leq G_n''(0)\frac{\psi(C_b\|m_\eta\|_\infty t)}{C_b^2\|m_\eta\|_\infty^2}$$

for all $t \geq 0$ and $t = 1$ leads to

$$G_n''(0)\frac{\psi(-C_b\|m_\eta\|_\infty)}{C_b^2\|m_\eta\|_\infty^2} \leq R_n(m^0 + m_\eta) - R_n(m^0) - G_n'(0) \leq G_n''(0)\frac{\psi(C_b\|m_\eta\|_\infty)}{C_b^2\|m_\eta\|_\infty^2}.$$

An easy computation gives

$$-G_n'(0) = \frac{1}{n}\sum_{i=1}^n m_\eta(x_i)\big(y_i - b'(m^0(x_i))\big) \;\text{ and }\; G_n''(0) = \frac{1}{n}\sum_{i=1}^n m_\eta^2(x_i) b''(m_\eta(x_i)),$$

and since obviously $\mathbb{E}_{\mathbb{P}_{\mathbf{y}|X}}[G_n'(0)] = 0$, we obtain

$$G_n''(0)\frac{\psi(-C_b\|m_\eta\|_\infty)}{C_b^2\|m_\eta\|_\infty^2} \leq R(m^0 + m_\eta) - R(m^0) \leq G_n''(0)\frac{\psi(C_b\|m_\eta\|_\infty)}{C_b^2\|m_\eta\|_\infty^2}.$$

Now, choosing $m_\eta = m_\theta - m^0$ and combining Assumption 1 with Equation (14) gives

$$C_b\|m_\eta\|_\infty \leq C_b \max_{i=1,\dots,n}(|\langle x_i^B, \theta\rangle| + |m^0(x_i)|) \leq C_b(\rho + C_n).$$

Hence, since $x \mapsto \psi(x)/x^2$ is an increasing function on $\mathbb{R}^+$, we end up with

$$G_n''(0)\frac{\psi(-C_b(C_n+\rho))}{C_b^2(C_n+\rho)^2} \leq R(m_\theta) - R(m_0) = \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),$$
$$G_n''(0)\frac{\psi(C_b(C_n+\rho))}{C_b^2(C_n+\rho)^2} \geq R(m_\theta) - R(m_0) = \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),$$

and since $G_n''(0) = n^{-1}\sum_{i=1}^n (m_\theta(x_i) - m^0(x_i))^2 b''(m^0(x_i))$, we obtain

$$\frac{L_n\psi(-C_b(C_n+\rho))}{C_n^2\phi(C_n+\rho)^2}\frac{1}{n}\|m^0(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\|_2^2 \leq \mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),$$
$$\frac{U_n\psi(C_b(C_n+\rho))}{C_b^2\phi(C_n+\rho)^2}\frac{1}{n}\|m^0(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\|_2^2 \geq \mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),$$

which concludes the proof of the Lemma. □

6.7. Proof of Theorem 2

Let us recall that

$$R_n(m_\theta) = \frac{1}{n}\sum_{i=1}^n b(m_\theta(x_i)) - \frac{1}{n}\sum_{i=1}^n y_i m_\theta(x_i)$$

for all $\theta \in \mathbb{R}^d$ and that

$$\hat{\theta} \in \mathrm{argmin}_{\theta \in B_d(\rho)}\big\{R_n(\theta) + \mathrm{bina}(\theta)\big\}. \tag{28}$$

Proposition 5 above entails that there is $\hat{h} = [\hat{h}_{j,\bullet}]_{j=1,\dots,p} \in \partial\|\hat{\theta}\|_{\mathrm{TV},\hat{w}}$, $\hat{g} = [\hat{g}_{j,\bullet}]_{j=1,\cdots,p} \in [\partial\delta_j(\hat{\theta}_{j,\bullet})]_{j=1,\dots,p}$ and $\hat{f} = [\hat{f}_{j,\bullet}]_{j=1,\dots,p} \in \partial\delta_{B_d(\rho)}(\hat{\theta})$ such that

$$\Big\langle \frac{1}{n}(\boldsymbol{X}^B)^\top(b'(m_{\hat{\theta}}(\boldsymbol{X})) - \boldsymbol{y}) + \hat{h} + \hat{g} + \hat{f}, \hat{\theta} - \theta\Big\rangle = 0$$

for all $\theta \in \mathbb{R}^d$. This can be rewritten as

$$\frac{1}{n}\langle b'(m_{\hat{\theta}}(\boldsymbol{X})) - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle$$
$$- \frac{1}{n}\langle \boldsymbol{y} - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle + \langle \hat{h} + \hat{g} + \hat{f}, \hat{\theta} - \theta\rangle = 0.$$

For any $\theta \in B_d(\rho)$ such that $n_j^\top \theta_{j,\bullet} = 0$ for all j and $h \in \partial\|\theta\|_{\mathrm{TV},\hat{w}}$, the monotony of the subdifferential mapping implies $\langle \hat{h}, \theta - \hat{\theta}\rangle \leq \langle h, \theta - \hat{\theta}\rangle$, $\langle \hat{g}, \theta - \hat{\theta}\rangle \leq 0$, and $\langle \hat{f}, \theta - \hat{\theta}\rangle \leq 0$, so that

$$\begin{aligned}\frac{1}{n}&\langle b'(m_{\hat{\theta}}(\boldsymbol{X})) - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle \\ &\leq \frac{1}{n}\langle \boldsymbol{y} - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle - \langle h, \hat{\theta} - \theta\rangle.\end{aligned} \tag{29}$$

Now, consider the function $H_n : \mathbb{R} \to \mathbb{R}$ defined by

$$H_n(t) = \frac{1}{n}\sum_{i=1}^n b(m_{\hat{\theta}+t\eta}(x_i)) - \frac{1}{n}\sum_{i=1}^n b'(m^0(x_i)) m_{\hat{\theta}+t\eta}(x_i),$$

where η will be defined later. We use again the same arguments as in the proof of Lemma 9. We differentiate H_n three times with respect t, so that

$$H_n'(t) = \frac{1}{n}\sum_{i=1}^n m_\eta(x_i) b'(m_{\hat{\theta}+t\eta}(x_i)) - \frac{1}{n}\sum_{i=1}^n b'(m^0(x_i)) m_\eta(x_i),$$
$$H_n''(t) = \frac{1}{n}\sum_{i=1}^n m_\eta^2(x_i) b''(m_{\hat{\theta}+t\eta}(x_i)),$$
$$\text{and} \quad H_n'''(t) = \frac{1}{n}\sum_{i=1}^n m_\eta^3(x_i) b'''(m_{\hat{\theta}+t\eta}(x_i)),$$

and in the way as in the proof of Lemma 9, we have $|H_n'''(t)| \leq C_b(C_n+\rho)|H_n''(t)|$, and Lemma 8 entails

$$H_n''(0)\frac{\psi(-C_b t(C_n+\rho))}{C_b^2(C_n+\rho)^2} \leq H_n(t) - H_n(0) - tH_n'(0) \leq H_n''(0)\frac{\psi(C_b t(C_n+\rho))}{C_b^2(C_n+\rho)^2},$$

for all $t \geq 0$. Taking $t = 1$ and $\eta = \theta - \hat{\theta}$ implies

$$H_n(1) = \frac{1}{n}\sum_{i=1}^{n} b(m_\theta(x_i)) - \frac{1}{n}\sum_{i=1}^{n} b'(m^0(x_i))m_\theta(x_i) = R(m_\theta),$$
$$\text{and} \quad H_n(0) = \frac{1}{n}\sum_{i=1}^{n} b(m_{\hat{\theta}}(x_i)) - \frac{1}{n}\sum_{i=1}^{n} b'(m^0(x_i))m_{\hat{\theta}}(x_i) = R(m_{\hat{\theta}}).$$

Moreover, we have

$$\begin{aligned} H_n'(0) &= \frac{1}{n}\sum_{i=1}^{n} \langle x_i^B, \theta - \hat{\theta}\rangle b'(m_{\hat{\theta}}(x_i)) - \frac{1}{n}\sum_{i=1}^{n} b'(m^0(x_i))\langle x_i^B, \hat{\theta} - \theta\rangle \\ &= \frac{1}{n}\langle b'(m_{\hat{\theta}}(\boldsymbol{X})) - b'(m^0(\boldsymbol{X})), \boldsymbol{X}^B(\theta - \hat{\theta})\rangle, \\ \text{and} \quad H_n''(0) &= \frac{1}{n}\sum_{i=1}^{n} \langle x_i^B, \hat{\theta} - \theta\rangle^2 b''(m_{\hat{\theta}}(x_i)). \end{aligned}$$

Then, we deduce that

$$\begin{aligned} H_n''(0)\frac{\psi(-C_b(C_n+\rho))}{C_b^2(C_n+\rho)^2} &\leq R(m_\theta) - R(m_{\hat{\theta}}) - \frac{1}{n}\langle b'(m_{\hat{\theta}}(\boldsymbol{X})) - b'(m^0(\boldsymbol{X})), \boldsymbol{X}^B(\theta - \hat{\theta})\rangle \\ &= \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})) - \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) \\ &\quad + \frac{1}{n}\langle b'(m_{\hat{\theta}}(\boldsymbol{X})) - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle. \end{aligned}$$

Then, with Equation (29), one has

$$\begin{aligned} &\phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) + H_n''(0)\frac{\psi(-C_b(C_n+\rho))}{C_b^2(C_n+\rho)^2} \\ &\quad \leq \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})) + \frac{1}{n}\langle \boldsymbol{y} - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle - \langle h, \hat{\theta} - \theta\rangle. \end{aligned} \tag{30}$$

As $H_n''(0) \geq 0$, it implies that

$$\begin{aligned} \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) &\leq \phi\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})) \\ &\quad + \frac{1}{n}\langle \boldsymbol{y} - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle - \langle h, \hat{\theta} - \theta\rangle. \end{aligned} \tag{31}$$

If $\frac{1}{n}\langle \boldsymbol{y} - b'(m^0(\boldsymbol{X})), \boldsymbol{X}^B(\hat{\theta} - \theta)\rangle - \langle h, \hat{\theta} - \theta\rangle < 0$, it follows that

$$\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) \leq \mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})),$$

then Theorem 2 holds. From now on, let us assume that

$$\frac{1}{n}\langle \boldsymbol{y} - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle - \langle h, \hat{\theta} - \theta\rangle \geq 0. \tag{32}$$

We first derive a bound on $\frac{1}{n}\langle \boldsymbol{y} - b'(m^0(\boldsymbol{X})), m_{\hat{\theta}}(\boldsymbol{X}) - m_\theta(\boldsymbol{X})\rangle$. Recall that $\mathbf{D}^{-1} = \mathbf{T}$ (see beginning of Section 6.5). We focus on finding out a bound for $\frac{1}{n}\langle (\boldsymbol{X}^B\mathbf{T})^\top(\boldsymbol{y} - b'(m^0(\boldsymbol{X}))), \mathbf{D}(\hat{\theta} - \theta)\rangle$. On the one hand, one has

$$\begin{aligned} &\frac{1}{n}\langle (\boldsymbol{X}^B)^\top(\boldsymbol{y} - b'(m^0(\boldsymbol{X}))), \hat{\theta} - \theta\rangle \\ &\quad = \frac{1}{n}\langle (\boldsymbol{X}^B\mathbf{T})^\top(\boldsymbol{y} - b'(m^0(\boldsymbol{X}))), \mathbf{D}(\hat{\theta} - \theta)\rangle \\ &\quad \leq \frac{1}{n}\sum_{j=1}^{p}\sum_{k=1}^{d_j} |((\boldsymbol{X}_{\bullet,j}^B T_j)_{\bullet,k})^\top(\boldsymbol{y} - b'(m^0(\boldsymbol{X})))|\, |(D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet}))_k| \end{aligned}$$

where $(\boldsymbol{X}_{\bullet,j}^B T_j)_{\bullet,k} = [(\boldsymbol{X}_{\bullet,j}^B T_j)_{1,k} \cdots (\boldsymbol{X}_{\bullet,j}^B T_j)_{n,k}]^\top \in \mathbb{R}^n$ is the k-th column of the matrix $\boldsymbol{X}_{\bullet,j}^B T_j$. Let us consider the event

$$\mathcal{E}_n = \bigcap_{j=1}^{p}\bigcap_{k=2}^{d_j} \mathcal{E}_{n,j,k}, \text{ where } \mathcal{E}_{n,j,k} = \Big\{\frac{1}{n}|(\boldsymbol{X}_{\bullet,j}^B T_j)_{\bullet,k}^\top(\boldsymbol{y} - b'(m^0(\boldsymbol{X})))| \leq \hat{w}_{j,k}\Big\},$$

so that, on $\mathcal{E}_n$, we have

$$\begin{aligned} \frac{1}{n}\langle (\boldsymbol{X}^B)^\top(\boldsymbol{y} - b'(m^0(\boldsymbol{X})), \hat{\theta} - \theta\rangle &\leq \sum_{j=1}^{p}\sum_{k=1}^{d_j} \hat{w}_{j,k}|(D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet}))_k| \\ &\leq \sum_{j=1}^{p} \|\hat{w}_{j,\bullet} \odot D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})\|_1. \end{aligned} \tag{33}$$

On the other hand, from the definition of the subgradient $[h_{j,\bullet}]_{j=1,\ldots,p} \in \partial\|\theta\|_{\mathrm{TV},\hat{w}}$ (see Equation (19)), one can choose h such that

$$h_{j,k} = (D_j^\top(\hat{w}_{j,\bullet} \odot \mathrm{sign}(D_j\theta_{j,\bullet})))_k$$

for all $k \in J_j(\theta)$ and

$$h_{j,k} = (D_j^\top(\hat{w}_{j,\bullet} \odot \mathrm{sign}(D_j\hat{\theta}_{j,\bullet}))_k = (D_j^\top(\hat{w}_{j,\bullet} \odot \mathrm{sign}(D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})))_k$$

for all $k \in J_j^{\complement}(\theta)$. Using a triangle inequality and the fact that $\mathrm{sign}(x)^\top x = \|x\|_1$, we obtain

$$\begin{aligned} -\langle h, \hat{\theta} - \theta\rangle &\leq \sum_{j=1}^{p} \|(\hat{w}_{j,\bullet})_{J_j(\theta)} \odot D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j(\theta)}\|_1 \\ &\quad - \sum_{j=1}^{p} \|(\hat{w}_{j,\bullet})_{J_j^{\complement}(\theta)} \odot D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j^{\complement}(\theta)}\|_1 \\ &\leq \sum_{j=1}^{p} \|(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j(\theta)}\|_{\mathrm{TV},\hat{w}_{j,\bullet}} - \sum_{j=1}^{p} \|(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j^{\complement}(\theta)}\|_{\mathrm{TV},\hat{w}_{j,\bullet}}. \end{aligned} \tag{34}$$

Combining inequalities (33) and (34), we get

$$\sum_{j=1}^{p} \|(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j^{\complement}(\theta)}\|_{\mathrm{TV},\hat{w}_{j,\bullet}} \leq 2 \sum_{j=1}^{p} \|(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j(\theta)}\|_{\mathrm{TV},\hat{w}_{j,\bullet}}$$

on $\mathcal{E}_n$. Hence

$$\sum_{j=1}^{p} \|(\hat{w}_{j,\bullet})_{J_j^{\complement}(\theta)} \odot D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j^{\complement}(\theta)}\|_1 \leq 2 \sum_{j=1}^{p} \|(\hat{w}_{j,\bullet})_{J_j(\theta)} \odot D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j(\theta)}\|_1.$$

This means that

$$\hat{\theta} - \theta \in \mathscr{C}_{\mathrm{TV},\hat{w}}(J(\theta)) \text{ and } \mathbf{D}(\hat{\theta} - \theta) \in \mathscr{C}_{1,\hat{w}}(J(\theta)), \tag{35}$$

see (10) and (27). Now, going back to (31) and taking into account (35), the compatibility of $\boldsymbol{X}^B\mathbf{T}$ given in Equation (26) provides the following on the event $\mathcal{E}_n$:

$$\begin{aligned}\phi \mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) &\leq \phi \mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\theta}(\boldsymbol{X})) \\ &\quad + 2 \sum_{j=1}^{p} \|(\hat{w}_{j,\bullet})_{J_j(\theta)} \odot D_j(\hat{\theta}_{j,\bullet} - \theta_{j,\bullet})_{J_j(\theta)}\|_1.\end{aligned}$$

Then

$$\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) \leq \mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\theta}(\boldsymbol{X})) + \frac{\|m_{\hat{\theta}}(\boldsymbol{X}) - m_{\theta}(\boldsymbol{X})\|_2}{\sqrt{n}\, \phi\, \kappa_{\mathbf{T},\hat{\gamma}}(J(\theta))\kappa(J(\theta))}, \tag{36}$$

where $\hat{\gamma} = (\hat{\gamma}_{1,\bullet}^{\top}, \ldots, \hat{\gamma}_{p,\bullet}^{\top})^{\top}$ is such that

$$\hat{\gamma}_{j,k} = \begin{cases} 2\hat{w}_{j,k} & \text{if } k \in J_j(\theta), \\ 0 & \text{if } k \in J_j^{\complement}(\theta), \end{cases}$$

for all $j = 1, \ldots, p$ and

$$\kappa_{\mathbf{T},\hat{\gamma}}(J(\theta)) = \left\{ 32 \sum_{j=1}^{p} \sum_{k=1}^{d_j} |\hat{\gamma}_{j,k+1} - \hat{\gamma}_{j,k}|^2 + 2|J_j(\theta)| \|\hat{\gamma}_{j,\bullet}\|_{\infty}^2 \Delta_{\min,J_j(\theta)}^{-1} \right\}^{-1/2}.$$

Now, we find an upper bound for

$$\frac{1}{\kappa_{\mathbf{T},\hat{\gamma}}^2(J(\theta))} = 32 \sum_{j=1}^{p} \sum_{k=1}^{d_j} |\hat{\gamma}_{j,k+1} - \hat{\gamma}_{j,k}|^2 + 2|J_j(\theta)| \|\hat{\gamma}_{j,\bullet}\|_{\infty}^2 \Delta_{\min,J_j(\theta)}^{-1}.$$

Note that $\|\hat{\gamma}_{j,\bullet}\|_{\infty} \leq 2\|\hat{w}_{j,\bullet}\|_{\infty}$. Let us write $J_j(\theta) = \{k_j^1, \ldots, k_j^{|J_j(\theta)|}\}$ and set $B_r = [\![k_j^{r-1}, k_j^r[\![= \{k_j^{r-1}, k_j^{r-1}+1, \ldots, k_j^r - 1\}$ for $r = 1, \ldots, |J_j(\theta)| + 1$ with the convention that $k_j^0 = 0$ and

$k_j^{|J_j(\theta)|+1} = d_j + 1$. Then

$$\begin{aligned}\sum_{k=1}^{d_j} |\hat{\gamma}_{j,k+1} - \hat{\gamma}_{j,k}|^2 &= \sum_{r=1}^{|J_j(\theta)|+1} \sum_{k \in B_r} |\hat{\gamma}_{j,k+1} - \hat{\gamma}_{j,k}|^2 \\ &= \sum_{r=1}^{|J_j(\theta)|+1} |\hat{\gamma}_{j,k_j^{r-1}+1} - \hat{\gamma}_{j,k_j^{r-1}}|^2 + |\hat{\gamma}_{j,k_j^r} - \hat{\gamma}_{j,k_j^r-1}|^2 \\ &= \sum_{r=1}^{|J_j(\theta)|+1} \hat{\gamma}_{j,k_j^{r-1}}^2 + \hat{\gamma}_{j,k_j^r}^2 \\ &= \sum_{r=1}^{|J_j(\theta)|} 2\, \hat{\gamma}_{j,k_j^r}^2 \\ &\leq 8\, |J_j(\theta)|\, \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|_{\infty}^2.\end{aligned}$$

Therefore

$$\begin{aligned}\frac{1}{\kappa_{\mathbf{T},\hat{\gamma}}^2(J(\theta))} &\leq 512 \sum_{j=1}^{p} \Big(|J_j(\theta)|\, \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|_{\infty}^2 + |J_j(\theta)|\, \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|_{\infty}^2 \Delta_{\min,J_j(\theta)}^{-1} \Big) \\ &\leq 512 \sum_{j=1}^{p} \Big(1 + \frac{1}{\Delta_{\min,J_j(\theta)}} \Big) |J_j(\theta)| \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|_{\infty}^2 \\ &\leq 512 |J(\theta)| \max_{j=1,\ldots,p} \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|_{\infty}^2.\end{aligned} \tag{37}$$

Now, we use the connection between the empirical norm and Kullback-Leibler divergence. Indeed, using Lemma 9, we get

$$\begin{aligned}&\frac{\|m_{\hat{\theta}}(\boldsymbol{X}) - m_{\theta}(\boldsymbol{X})\|_2}{\sqrt{n}\phi\kappa_{\mathbf{T},\hat{\gamma}}(J(\theta))\kappa(J(\theta))} \\ &\quad \leq \frac{1}{\sqrt{\phi}\kappa_{\mathbf{T},\hat{\gamma}}(J(\theta))\kappa(J(\theta))} \Big(\frac{1}{\sqrt{n}} \|m_{\hat{\theta}}(\boldsymbol{X}) - m^0(\boldsymbol{X})\|_2 + \frac{1}{\sqrt{n}} \|m^0(\boldsymbol{X}) - m_{\theta}(\boldsymbol{X})\|_2 \Big) \\ &\quad \leq \frac{2}{\sqrt{\phi}\kappa_{\mathbf{T},\hat{\gamma}}(J(\theta))\kappa(J(\theta))\sqrt{C_n(\rho, L_n)}} \Big(\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X}))^{1/2} \\ &\qquad\qquad + \mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\theta}(\boldsymbol{X}))^{1/2} \Big),\end{aligned}$$

where we defined $C_n(\rho, L_n) = \frac{L_n \psi(-C_b(C_n+\rho))}{C_b^2 \phi(C_n+\rho)^2}$, so that combined with Equation (36), we obtain

$$\begin{aligned}\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) &\leq \mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\theta}(\boldsymbol{X})) \\ &\quad + \frac{2}{\sqrt{\phi}\kappa_{\mathbf{T},\hat{\gamma}}(J(\theta))\kappa(J(\theta))\sqrt{C_n(\rho, L_n)}} \big(\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X}))^{1/2} \\ &\qquad + \mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\theta}(\boldsymbol{X}))^{1/2} \big).\end{aligned}$$

This inequality entails the following upper bound

$$\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) \leq 3\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})) + \frac{5}{\phi\kappa^2_{\mathbf{T},\hat{\gamma}}(J(\theta))\kappa^2(J(\theta))C_n(\rho, L_n)},$$

since whenever we have $x \leq c + b\sqrt{x}$ for some $x, b, c > 0$, then $x \leq 2c + b^2$. Introducing $g(x) = x^2/\psi(-x) = x^2/(e^{-x} + 1 - x)$, we note that

$$\frac{1}{C_n(\rho, L_n)} = \frac{\phi}{L_n} g(C_b(C_n + \rho)) \leq \frac{\phi}{L_n}(C_b(C_n + \rho) + 2),$$

since $g(x) \leq x + 2$ for any $x > 0$. Finally, by using also (37), we end up with

$$\mathrm{KL}_n(m^0(\boldsymbol{X}), m_{\hat{\theta}}(\boldsymbol{X})) \leq 3\mathrm{KL}_n(m^0(\boldsymbol{X}), m_\theta(\boldsymbol{X})) + \frac{2560(C_b(C_n + \rho) + 2)}{L_n\kappa^2(J(\theta))} \, |J(\theta)| \, \|(\hat{w}_{j,\bullet})_{J_j(\theta)}\|^2_\infty,$$

which is the statement provided in Theorem 2. The only thing remaining is to control the probability of the event $\mathcal{E}_n^{\complement}$. This is given by the following:

$$\begin{aligned}\mathbb{P}[\mathcal{E}_n^{\complement}] &\leq \sum_{j=1}^p \sum_{k=2}^{d_j} \mathbb{P}\Big[\frac{1}{n}|(\boldsymbol{X}^B_{\bullet,j}T_j)^\top_{\bullet,k}(\boldsymbol{y} - b'(m^0(\boldsymbol{X})))| \geq \hat{w}_{j,k}\Big] \\ &\leq \sum_{j=1}^p \sum_{k=2}^{d_j} \mathbb{P}\Big[\sum_{i=1}^n |(\boldsymbol{X}^B_{\bullet,j}T_j)_{i,k}(y_i - b'(m^0(x_i)))| \geq n\hat{w}_{j,k}\Big].\end{aligned}$$

Let $\xi_{i,j,k} = (\boldsymbol{X}^B_{\bullet,j}T_j)_{i,k}$ and $Z_i = y_i - b'(m^0(x_i))$. Note that conditionally on x_i, the random variables (Z_i) are independent. It can be easily shown (see Theorem 5.10 in Lehmann and Casella (1998)) that the moment generating function of Z (copy of Z_i) is given by

$$\mathbb{E}[\exp(tZ)] = \exp(\phi^{-1}\{b(m^0(x) + t) - tb'(m^0(x) - b(m^0(x)))\}). \tag{38}$$

Applying Lemma 6.1 in Rigollet (2012), using (38) and Assumption 1, we can derive the following Chernoff-type bounds

$$\mathbb{P}\Big[\sum_{i=1}^n |\xi_{i,j,k}Z_i| \geq n\hat{w}_{j,k}\Big] \leq 2\exp\Big(-\frac{n^2\hat{w}^2_{j,k}}{2U_n\phi\|\xi_{\bullet,j,k}\|^2_2}\Big), \tag{39}$$

where $\xi_{\bullet,j,k} = [\xi_{1,j,k} \cdots \xi_{n,j,k}]^\top \in \mathbb{R}^n$. We have

$$\boldsymbol{X}^B_{\bullet,j}T_j = \begin{bmatrix} 1 & \sum_{k=2}^{d_j} x^B_{1,j,k} & \sum_{k=3}^{d_j} x^B_{1,j,k} & \cdots & \sum_{k=d_j-1}^{d_j} x^B_{1,j,k} & x^B_{1,j,d_j} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \sum_{k=2}^{d_j} x^B_{n,j,k} & \sum_{k=3}^{d_j} x^B_{n,j,k} & \cdots & \sum_{k=d_j-1}^{d_j} x^B_{n,j,k} & x^B_{n,j,d_j} \end{bmatrix},$$

therefore

$$\|\xi_{\bullet,j,k}\|^2_2 = \sum_{i=1}^n (\boldsymbol{X}^B_{\bullet,j}T_j)^2_{\bullet,k} = \#\Big(\Big\{i : x_{i,j} \in \bigcup_{r=k}^{d_j} I_{j,r}\Big\}\Big) = n\hat{\pi}_{j,k}. \tag{40}$$

So, using the weights $\hat{w}_{j,k}$ given by (12) together with (39) and (40), we obtain that the probability of $\mathcal{E}_n^{\complement}$ is smaller than $2e^{-A}$. This concludes the proof of the first part of Theorem 2. □

6.8. Proof of Theorem 3

First, let us note that in the least squares setting, we have $R(m_\theta) - R(m^0) = \|m_\theta - m^0\|^2_n$ for any $\theta \in \mathbb{R}^d$ where $\|g\|^2_n = \frac{1}{n}\sum_{i=1}^n g(x_i)^2$, and that $b(y) = \frac{1}{2}y^2$, $\phi = \sigma^2$ (noise variance) in Equation (2), and $L_n = U_n = 1$, $C_b = 0$. Theorem 2 provides

$$\|m_{\hat{\theta}} - m^0\|^2_n \leq 3\|m_\theta - m_{\theta^0}\|^2_n + \frac{5120\sigma^2}{\kappa^2(J(\theta))}\frac{|J(\theta)|(A + \log d)}{n}$$

for any $\theta \in \mathbb{R}^d$ such that $n_j^\top\theta_{j\bullet} = 0$ and $J(\theta) \leq J_*$. Since $d_j = D$ for all $j = 1, \ldots, p$, we have $d = Dp$ and

$$|J(\theta)| = \sum_{j=1}^p |\{k = 2, \ldots, D : \theta_{j,k} \neq \theta_{j,k-1}\}| \leq (D-1)|\mathcal{J}(\theta)|\|\theta\|_\infty \leq Dp\|\theta\|_\infty \tag{41}$$

for any $\theta \in \mathbb{R}^d$, where we recall that $\mathcal{J}(\theta) = \{j = 1, \ldots, p : \theta_{j,\bullet} \neq \mathbf{0}_D\}$. Also, recall that $I_{j,1} = I_1 = [0, \frac{1}{D}]$ and $I_{j,k} = I_k = (\frac{k-1}{D}, \frac{k}{D}]$ for $k = 2, \ldots, D$ and $j = 1, \ldots, p$. Also, we consider $\theta = \theta^*$, where $\theta^*_{j,\bullet}$ is defined, for any $j \in \mathcal{J}_*$, as the minimizer of

$$\sum_{i=1}^n \Big(\sum_{k=1}^D (\theta_{j,k} - m^0_j(x_{i,j}))\mathbf{1}_{I_k}(x_{i,j})\Big)^2$$

over the set of vectors $\theta_{j,\bullet} \in \mathbb{R}^D$ satisfying $n_j^\top\theta_{j,\bullet} = 0$, and we put $\theta^*_{j,\bullet} = \mathbf{0}_D$ for $j \notin \mathcal{J}_*$. It is easy to see that the solution is given by

$$\theta^*_{j,k} = \frac{\sum_{i=1}^n m^0_j(x_{i,j})\mathbf{1}_{I_k}(x_{i,j})}{n_{j,k}},$$

where we recall that $n_{j,k} = \sum_{i=1}^n \mathbf{1}_{I_k}(x_{i,j})$. Note in particular that the identifiability assumption $\sum_{i=1}^n m^0_j(x_{i,j}) = 0$ entails that $n_j^\top\theta^*_{j,\bullet} = 0$. In order to control the bias term, an easy computation gives that, whenever $x_{i,j} \in I_k$

$$|\theta^*_{j,k} - m^0_j(x_{i,j})| \leq \frac{\sum_{i'=1}^n |m^0_j(x_{i',j}) - m^0_j(x_{i,j})|\mathbf{1}_{I_k}(x_{i',j})}{n_{j,k}} \leq L|I_k| = \frac{L}{D},$$

where we used the fact that m_j^0 is L-Lipschitz, so that

$$\begin{aligned}
\|m_{\theta^*} - m^0\|_n^2 &= \frac{1}{n}\sum_{i=1}^n (m_{\theta^*}(x_{i,j}) - m^0(x_{i,j}))^2 \\
&= \frac{1}{n}\sum_{i=1}^n \Big(\sum_{j\in\mathcal{J}_*}\sum_{k=1}^D (\theta^*_{j,k} - m^0(x_{i,j}))\mathbf{1}_{I_k}\Big)^2 \\
&\le \frac{|\mathcal{J}_*|}{n}\sum_{i=1}^n \Big(\sum_{j\in\mathcal{J}_*}\sum_{k=1}^D (\theta^*_{j,k} - m^0(x_{i,j}))\mathbf{1}_{I_k}\Big)^2 \\
&\le \frac{|\mathcal{J}_*|}{n}\sum_{i=1}^n \sum_{j\in\mathcal{J}_*}\sum_{k=1}^D (\theta^*_{j,k} - m^0(x_{i,j}))^2\mathbf{1}_{I_k}(x_{i,j}) \\
&\le |\mathcal{J}_*|\sum_{j\in\mathcal{J}_*}\sum_{k=1}^D L^2|I_k|^2\mathbf{1}_{I_k}(x_{i,j}) \le \frac{L^2|\mathcal{J}_*|^2}{D^2}.
\end{aligned}$$

Note that $|\theta^*_{j,k}| \le \|m_j^0\|_{n,\infty}$ where $\|m_j^0\|_{n,\infty} = \max_{i=1,\dots,n}|m_j^0(x_{i,j})|$. This entails that $\|\theta^*\|_\infty \le \max_{j=1,\dots,p}\|m_j^0\|_{n,\infty} = M_n$. So, using also (41), we end up with

$$\|m_{\hat\theta} - m^0\|_n^2 \le \frac{3L^2|\mathcal{J}_*|^2}{D^2} + \frac{5120\sigma^2}{\kappa^2(J(\theta^*))}\frac{D\mathcal{J}_*M_n(A+\log(DpM_n))}{n},$$

which concludes the proof Theorem 3 using $D = n^{1/3}$. □

References

A. Agresti. *Foundations of Linear and Generalized Linear Models*. John Wiley & Sons, 2015.

M. Z. Alaya, S. Gaïffas, and A. Guilloux. Learning the intensity of time events with change-points. *Information Theory, IEEE Transactions on*, 61(9):5148–5171, 2015.

F. Bach. Self-concordant analysis for logistic regression. *Electron. J. Statist.*, 4:384–414, 2010.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

E. Bacry, M Bompaire, P. Deegan, S. Gaïffas, and S. V. Poulsen. tick: a python library for statistical learning, with an emphasis on hawkes processes and time-dependent models. *Journal of Machine Learning Research*, 18(214):1–5, 2018. URL `http://jmlr.org/papers/v18/17-381.html`.

P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5, 2014.

P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson. Parameterized neural networks for high-energy physics. *The European Physical Journal C*, 76(5):1–7, Apr 2016.

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009. ISSN 0090-5364.

J. A. Blackard and D. J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004. ISBN 0-521-83378-7.

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

P. Bühlmann and S. van De Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011.

F. Bunea, A. Tsybakov, and M. Wegkamp. Sparsity oracle inequalities for the Lasso. *Electron. J. Statist.*, 1:169–194, 2007.

E. J. Candès and M. B. Wakin. An Introduction To Compressive Sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

B. Chlebus and S. H. Nguyen. On finding optimal discretizations for two attributes. In Lech Polkowski and Andrzej Skowron, editors, *Rough Sets and Current Trends in Computing*, volume 1424 of *Lecture Notes in Computer Science*, pages 537–544. Springer Berlin Heidelberg, 1998.

L. Condat. A Direct Algorithm for 1D Total Variation Denoising. *IEEE Signal Processing Letters*, 20(11):1054–1057, 2013.

A. S. Dalalyan, M. Hebiri, and J. Lederer. On the prediction performance of the Lasso. *Bernoulli*, 23(1):552–581, 2017.

D. L. Donoho and M. Elad. Optimally sparse representation in general (non-orthogonal) dictionaries via ℓ_1 minimization. In *PROC. NATL ACAD. SCI. USA 100 2197202*, 2002.

D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *Information Theory, IEEE Transactions on*, 47(7):2845–2862, 2001.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007.

J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4): 367–378, 2002.

S. Garcia, J. Luengo, J. A. Saez, V. Lopez, and F. Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750, 2013.

P. J. Green and B. W. Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach.* Chapman and Hall, London, 1994.

T. Hastie and R. Tibshirani. *Generalized additive models.* Wiley Online Library, 1990.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer-Verlag, New York, 2001.

J. Horowitz, J. Klemelä, and E. Mammen. Optimal estimation in additive regression models. *Bernoulli*, 12(2):271–298, 2006.

S. Ivanoff, F. Picard, and V. Rivoirard. Adaptive lasso and group-lasso for functional poisson regression. *The Journal of Machine Learning Research*, 17(1):1903–1948, 2016.

K. Knight and W. Fu. Asymptotics for Lasso-type estimators. *Ann. Statist.*, 28(5):1356–1378, 2000.

R. Kohavi. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *KDD*, volume 96, pages 202–207, 1996.

E. L. Lehmann and G. Casella. *Theory of point estimation*. Springer texts in statistics. Springer, New York, 1998.

M. Lichman. UCI Machine Learning Repository, 2013.

H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: an enabling technique. *Data Min. Knowl. Discov.*, 6(4):393–423, 2002.

G. Lugosi and N. Vayatis. On the Bayes-risk consistency of regularized boosting methods. *Annals of Statistics*, pages 30–55, 2004.

L. Meier, S. van De Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

L. Meier, S. Van de Geer, and P. Bühlmann. High-dimensional additive modeling. *The Annals of Statistics*, 37(6B):3779–3821, 2009.

S. Moro, P. Cortez, and P. Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. R. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition, 1993.

F. Rapaport, E. Barillot, and J. P. Vert. Classification of arraycgh data using fused SVM. *Bioinformatics*, 24(13):i375–i382, 2008.

P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

P. Rigollet. Kullback Leibler aggregation and misspecified generalized linear models. *Ann. Statist.*, 40(2):639–665, 2012.

M. A. Russell. *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. O'Reilly Media, 2013.

B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1): 267–288, 1996a.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996b.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused Lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(1):91–108, 2005.

S. van de Geer. High-dimensional generalized linear models and the Lasso. *Ann. Statist.*, 36(2): 614–645, 2008.

S. van de Geer and J. Lederer. *The Lasso, correlated design, and improved oracle inequalities*, volume Volume 9 of *Collections*, pages 303–316. Institute of Mathematical Statistics, 2013.

J. Wu and S. Coggeshall. *Foundations of Predictive Analytics (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 1st edition, 2012.

I. C. Yeh and C. H. Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

Y. L. Yu. On decomposing the proximal map. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 91–99. 2013.

P. Zhao and B. Yu. On model selection consistency of Lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006.

Journal of Machine Learning Research 20 (2019) 1-38 Submitted 9/17; Revised 8/19; Published 8/19

Layer-Wise Learning Strategy for Nonparametric Tensor Product Smoothing Spline Regression and Graphical Models

Kean Ming Tan KEANMING@UMICH.EDU
Department of Statistics
University of Michigan
Ann Arbor MI, 48109

Junwei Lu JUNWEILU@HSPH.HARVARD.EDU
Department of Biostatistics
Harvard T.H. Chan School of Public Health
Boston MA, 02115

Tong Zhang TONGZHANG@TONGZHANG-ML.ORG
Department of Computer Science and Engineering
Department of Mathematics
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong.

Han Liu HANLIU@NORTHWESTERN.EDU
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston IL, 60208

Editor: Sara van de Geer

Abstract

Nonparametric estimation of multivariate functions is an important problem in statistical machine learning with many applications, ranging from nonparametric regression to nonparametric graphical models. Several authors have proposed to estimate multivariate functions under the smoothing spline analysis of variance (SSANOVA) framework, which assumes that the multivariate function can be decomposed into the summation of main effects, two-way interaction effects, and higher order interaction effects. However, existing methods are not scalable to the dimension of the random variables and the order of interactions. We propose a LAyer-wiSE leaRning strategy (LASER) to estimate multivariate functions under the SSANOVA framework. The main idea is to approximate the multivariate function sequentially starting from a model with only the main effects. Conditioned on the support of the estimated main effects, we estimate the two-way interaction effects only when the corresponding main effects are estimated to be non-zero. This process is continued until no more higher order interaction effects are identified. The proposed strategy provides a data-driven approach for estimating multivariate functions under the SSANOVA framework. Our proposal yields a sequence of estimators. To study the theoretical properties of the sequence of estimators, we establish the notion of post-selection persistency. Extensive numerical studies are performed to evaluate the performance of LASER.

Keywords: Persistency, nonparametric regression, nonparametric graphical models, sequential algorithm, model selection

©2019 Kean Ming Tan, Junwei Lu, Tong Zhang, and Han Liu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-525.html.

1. Introduction

Much progress has been made in nonparametric estimation of univariate functions. However, nonparametric estimation of multivariate functions remains a challenging problem due to the curse of dimensionality. A number of algorithms were proposed to estimate low-dimensional multivariate functions, but there are few practical algorithms for estimating multivariate functions with higher dimension.

To address this issue, many authors proposed to restrict the multivariate function to some specific model classes. One popular model class is the additive model in which the high-dimensional multivariate function $f(\cdot)$ is decomposed into the sum of d one-dimensional functions (Stone, 1985; Hastie and Tibshirani, 1990). To increase the flexibility of the additive model to accommodate situation in which interactions among the variables may be present, Lin (2000) proposed to estimate the multivariate function under the smoothing spline analysis of variance (SSANOVA) framework. More specifically, Lin (2000) proposed to decompose the d-dimensional function $f(\cdot)$ as the summation of some constant μ, one-dimensional functions (main effects), two-dimensional functions (two-way interaction effects), and so on:

$$f(\boldsymbol{x}) = \mu + \sum_{j=1}^{d} f_j(x_j) + \sum_{j<k} f_{jk}(x_j, x_k) + \cdots . \tag{1}$$

We refer the reader to Lin (2000) and Gu (2013) for a detailed discussion of such models.

Most existing methods under the SSANOVA framework truncate (1) to the rth order interaction effects. Even so, existing methods are computationally intensive and are not scalable to the dimension of the random variables d and the order of the interaction term r. For instance, to fit a nonparametric regression model under the SSANOVA framework with the rth order interaction, it involves simultaneously fitting $O(d^r)$ terms and is often infeasible when both d and r are large (Lin and Zhang, 2006). Thus, one way to reduce the computation complexity is to consider only the two-way interaction terms and remove all of the higher order interaction terms from the model (Lin, 2000; Zhang et al., 2004; Lin and Zhang, 2006; Jeon and Lin, 2006; Yau et al., 2012).

The strong heredity assumption is often used for modeling regression with two-way interaction effects (Bien et al., 2013; Hao and Zhang, 2014; Haris et al., 2016; Hao et al., 2018; Radchenko and James, 2010). The strong heredity assumption mandates that both of the corresponding main effects must be present when an interaction term is included in the model. Under the strong heredity assumption, Hao and Zhang (2014) and Hao et al. (2018) proposed approaches that perform variable selection on the main effects, and allow interactions into the model once the main effects have been identified, in the context of linear regression. They proved theoretically that the main effects and interactions can be selected consistently as long as the variables are Gaussian with mean zero. On the other hand, Bien et al. (2013) and Haris et al. (2016) proposed penalty functions that are specifically designed for two-way interaction models with sparsity and strong heredity assumption in the context of linear regression. Similar work has also been done in the context of nonparametric regression (Radchenko and James, 2010). However, Bien et al. (2013), Haris et al. (2016), and Radchenko and James (2010) are computationally intractable when d is large since it

involves modeling $\binom{d}{2}$ terms. We refer the reader to Haris et al. (2016) for a comprehensive review of the literature.

In this paper, we propose a framework to estimate multivariate functions that take the form (1) without restricting it to only modeling the two-way interaction terms. We impose a hierarchical structure on the higher order interaction terms under the SSANOVA framework. Let $J, J' \subseteq \{1, \ldots, d\}$. Given a d-dimensional function f, we denote f_J as the $|J|$th order interaction term in f with variables $\{x_j\}_{j \in J}$. We impose the hierarchical structural assumption that

$$f_J = 0 \implies f_{J'} = 0, \qquad \text{if } J \subset J'. \tag{2}$$

In other words, we assume that a higher order interaction is not active when some of the lower order terms containing some variables that belong to the higher order term are not active. For instance, if the main effect of the jth variable $f_j = 0$, then any higher order term that involves the jth variable is not active. The hierarchical structural assumption is plausible in many data applications. For instance, in the context of nonparametric regression, the hierarchical structural assumption implies that the main effects are more important in modeling the response than the higher order interactions. Therefore, if the leading order interaction function does not have any explanatory power, the higher order interaction terms should be inactive. The hierarchical assumption in (2) can be thought of as an extension of the strong heredity assumption to modeling higher order interaction terms.

Under the hierarchical structural assumption, we propose a layer-wise learning algorithm to estimate multivariate functions under the SSANOVA framework. Our algorithm sequentially estimates the multivariate function starting from the main effects to the higher order interaction terms. In each step of our algorithm, we utilize the support of the estimated function from the previous step and estimate the function based on the hierarchical structural assumption in (2). This process is continued until no more higher order interaction effects are active. Instead of fitting the SSANOVA model with $d + \binom{d}{2} + \cdots + \binom{d}{r}$ terms, our proposal fits the SSANOVA model with at most $d + \binom{s_1}{2} + \cdots + \binom{s_{r-1}}{r}$ terms, where s_k is the cardinality of the support of the estimated kth order interaction effects. Thus, our algorithm is scalable to modeling higher order interaction terms with large dimension d. Our proposed framework can be interpreted as an extension of Hao and Zhang (2014) and Hao et al. (2018) to modeling multivariate functions, as well as modeling higher order interaction effects. Compared to Radchenko and James (2010) that involves modeling all two-way interaction terms, our approach is a multi-stage procedure and therefore is computationally efficient and scalable for problems with large dimension d.

To quantify the theoretical properties of our estimator, we propose the notion of post-selection persistency. We show that conditioned on the support of the estimator obtained from the $(r-1)$th step of the algorithm, the excessive risk between the estimator obtained from the rth step and the best rth order SSANOVA model converges to zero. Our results hold without assuming that the true multivariate function takes the form of the SSANOVA model in (1). In addition, we impose minimal distributional assumptions on the data.

We apply the proposed method to fitting nonparametric regression and graphical models. For nonparametric graphical models, our proposal is the first feasible approach to learn the graph without restricting the model to contain only pairwise interaction terms. In Section 2, we describe the problem setup and define the tensor product space for the functional component in the SSANOVA decomposition in (1). We then propose the layer-wise learning algorithm for estimating multivariate function under a generic loss function. We apply the proposed algorithm to fitting nonparametric regression and graphical models in Sections 3 and 4, respectively. Numerical studies are performed in Section 5. We close with a discussion in Section 6. The proofs of the theoretical results are given in the Appendix.

2. Layer-Wise Learning Strategy

We describe the problem setup and define some notation. We then propose the layer-wise learning strategy for estimating nonparametric functions with high order interactions.

2.1 Problem Setup and Notation

We start with a brief overview of the tensor product space of Sobolev spaces and refer the reader to Lin (2000) for a detailed review. For any non-negative integer m, the mth order Sobolev space with a univariate variable $x_j \in [0,1]$ is defined as

$$H_j^m = \left\{ g \;\middle|\; g^{(\nu)} \text{ is absolutely continuous for } 0 \le \nu \le m-1;\ g^{(m)} \in L^2([0,1]);\ \int_0^1 g(u)du = 0 \right\},$$

where $g^{(\nu)}$ is the νth order derivative of g. The Sobolev norm for function $g \in H_j^m$ is defined as

$$\|g\|_{H_j^m}^2 = \sum_{\nu=0}^{m} \int_0^1 \left[g^{(\nu)}(u) \right]^2 du.$$

For notational convenience, let $[d] = \{1, \ldots, d\}$. Let $J \subseteq [d]$ be an index set with cardinality $|J| = r$, and let $\mathcal{H}_J = \otimes_{j \in J} H_j^m$ be the completed tensor product space of H_j^m for all $j \in J$. We assume that $g_J \in \mathcal{H}_J$. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_r)^T$ be an r-dimensional vector with integer entries and let $\|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^r \alpha_i \le m$. Let $J = \{j_1, \ldots, j_r\}$. The Sobolev norm for the multivariate function $g_J \in \mathcal{H}_J$ is defined as

$$\|g_J\|_{\mathcal{H}_J}^2 = \sum_{\boldsymbol{\alpha}: \|\boldsymbol{\alpha}\|_1 \le m} \|D^{\boldsymbol{\alpha}} g_J\|_2 \qquad \text{where} \qquad D^{\boldsymbol{\alpha}} = \frac{\partial^{\|\boldsymbol{\alpha}\|_1}}{\partial x_{j_1}^{\alpha_1} \cdots \partial x_{j_r}^{\alpha_r}}. \tag{3}$$

We define the smoothing spline ANOVA function class as

$$\{1\} \oplus \sum_{j=1}^{d} H_j^m \oplus \sum_{J \subseteq [d], |J|=2} \mathcal{H}_J \oplus \sum_{J \subseteq [d], |J|=3} \mathcal{H}_J \oplus \cdots.$$

Each functional component in the SSANOVA decomposition (1) lies in a subspace in the orthogonal decomposition of the tensor product space. We define the rth order smoothing spline ANOVA function class as

$$\mathcal{H}^{(r)} = \{1\} \oplus \sum_{j=1}^{d} H_j^m \oplus \sum_{J \subseteq [d], |J|=2} \mathcal{H}_J \oplus \cdots \oplus \sum_{J \subseteq [d], |J|=r} \mathcal{H}_J. \tag{4}$$

The additive model introduced in Stone (1985) is a special case of (4) with $r = 1$.

2.2 Layer-Wise Learning Algorithm

We propose the layer-wise learning algorithm to learn a d-dimensional multivariate function under the SSANOVA framework. Recall from (2) that we assume a hierarchical structure on the model, i.e., higher order interaction terms are not active when the lower order terms are not active. To this end, we define some additional notation that will be used throughout the paper. Given a set $\mathcal{S} \subseteq \{J \mid J \subseteq [d]\}$, let $|\mathcal{S}|_{\max} = \max\{|I| \mid I \in \mathcal{S}\}$ be the largest cardinality among the sets in $\mathcal{S}$. In addition, we define

$$\sigma(\mathcal{S}) = \left\{ I \;\middle|\; |I| = |\mathcal{S}|_{\max} + 1, I \subseteq \bigcup_{|J|=|\mathcal{S}|_{\max},\, J \in \mathcal{S}} J \right\}.$$

For example, if $\mathcal{S} = \{\{1\}, \{2\}, \{3\}\}$, then $\sigma(\mathcal{S}) = \{\{1,2\}, \{1,3\}, \{2,3\}\}$. For notational convenience, we also define $\delta(\mathcal{S}) = \mathcal{S} \cup \sigma(\mathcal{S})$ throughout the paper. Thus, in this example, $\delta(\mathcal{S}) = \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}\}$.

The main crux of our proposal is to estimate the multivariate function sequentially starting from the main effects. Using the support of the estimated function from the previous step, we estimate the multivariate function by considering higher order terms only when the lower order terms are estimated to be non-zeros. Let D be the data and let $\mathcal{L}_n(D, f)$ be a generic loss function for estimating f. At the first step of the algorithm, we estimate the function f subject to $f \in \mathcal{H}^{(1)}$. In other words, f is assumed to be additive and we propose to estimate f by

$$\widehat{f}^{(1)} = \operatorname*{argmin}_{f \in \mathcal{H}^{(1)}} \mathcal{L}_n(D, f), \qquad \text{subject to} \qquad \sum_{j=1}^{d} \|P_j(f)\|_2 \le \tau, \tag{5}$$

where $P_j(f)$ is the orthogonal projection of f onto H_j^m. The penalty term $\sum_{j=1}^{d} \|P_j(f)\|_2 \le \tau$ can be interpreted as an ℓ_1 constraint across components to encourage sparsity, and an ℓ_2 constraint within components to encourage smoothness. The tuning parameter τ controls the number of main effects that are estimated to be non-zero.

Let $\mathcal{S}^{(1)}$ be the support of $\widehat{f}^{(1)}$, that is, $\mathcal{S}^{(1)} = \{j \mid P_j(\widehat{f}^{(1)}) \neq 0\}$. Given the support $\mathcal{S}^{(1)}$, we fit the following model at the second step of our algorithm

$$\begin{aligned} \widehat{f}^{(2)} = &\operatorname*{argmin}_{f \in \mathcal{H}^{(2)}, \mathcal{S}(f) = \delta(\mathcal{S}^{(1)})} \mathcal{L}_n(D, f), \\ &\text{subject to} \sum_{j \in \mathcal{S}^{(1)}} \|P_j(f)\|_2^2 + \sum_{\{j_1, j_2\} \in \sigma(\mathcal{S}^{(1)})} \|P_{j_1 j_2}(f)\|_2 \le \tau, \end{aligned} \tag{6}$$

where $P_{j_1 j_2}(f)$ is the orthogonal projection of f onto $H_{j_1}^m \otimes H_{j_2}^m$. For notational convenience, let $\mathcal{S}(f)$ be the support of f. At the second step of our algorithm, we update both the main effects and the second order interactions, with the support of the function constrained on $\mathcal{S}(f) = \delta(\mathcal{S}^{(1)})$. Since we have selected the support for the main effects, we use a ridge penalty to encourage smoothness for the main effects.

More generally, let $\mathcal{S}^{(r-1)}$ be the support identified at the $(r-1)$th step of our proposed algorithm. Let $J \subseteq [d]$ be an index set and let $P_J(f)$ be the orthogonal projection of f onto

Algorithm 1 Layer-Wise Learning Method (LASER).

Input: : Data D.
Initialize: $\mathcal{S}^{(0)} = \emptyset$, $\sigma(\mathcal{S}^{(0)}) = \{1, \ldots, d\}$, and $r = 1$.
repeat

1. Update the function with rth order interaction effects:

$$\begin{aligned} \widehat{f}^{(r)} = &\operatorname*{argmin}_{f \in \mathcal{H}^{(r)}, \mathcal{S}(f) = \delta(\mathcal{S}^{(r-1)})} \mathcal{L}_n(D, f), \\ &\text{subject to} \sum_{J \in \mathcal{S}^{(r-1)}} \|P_J(f)\|_2^2 + \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} \|P_J(f)\|_2 \le \tau. \end{aligned}$$

2. Update the support $\mathcal{S}^{(r)} = \mathcal{S}(\widehat{f}^{(r)})$.
3. $r \leftarrow r + 1$.

until $\mathcal{S}^{(r-1)} = \mathcal{S}^{(r)}$.
Output: A sequence of estimators $\{\widehat{f}^{(\ell)}\}_{\ell=1}^{r}$.

the $|J|$th order interaction effect space $\otimes_{j \in J} H_j^m$. At the rth step of the algorithm, we fit the model

$$\begin{aligned} \widehat{f}^{(r)} = &\operatorname*{argmin}_{f \in \mathcal{H}^{(r)}, \mathcal{S}(f) = \delta(\mathcal{S}^{(r-1)})} \mathcal{L}_n(D, f), \\ &\text{subject to} \sum_{J \in \mathcal{S}^{(r-1)}} \|P_J(f)\|_2^2 + \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} \|P_J(f)\|_2 \le \tau. \end{aligned} \tag{7}$$

We continue this process until no more higher order interaction effects are estimated to be non-zero. We summarize the proposed method in Algorithm 1. Step 1 in Algorithm 1 depends on a specific loss function $\mathcal{L}_n(D, f)$. We will present the details for Step 1 in the context of nonparametric regression and nonparametric graphical models in Sections 3 and 4, respectively.

2.3 Post-Selection Persistency

We first provide a brief review of the definition of persistency introduced by Greenshtein and Ritov (2004). We define the risk of some function f as $R(f) = \mathbb{E}[\mathcal{L}_n(D, f)]$. An estimator $\widehat{f}$ is said to be persistent relative to a class of function $\mathcal{F}$ if

$$R(\widehat{f}) - \inf_{f \in \mathcal{F}} R(f) = o_P(1). \tag{8}$$

In other words, the risk of the estimator $\widehat{f}$ is consistent to that of the oracle function under the model class $\mathcal{F}$. In the statistical literature, many authors have shown that the estimators for various statistical models are persistent (see, for instance, Greenshtein and Ritov, 2004 for the lasso regression, and Ravikumar et al., 2009 for the sparse additive model). However, most of the existing results on persistency are derived for a single estimator, and not much work has been done to characterize a sequence of estimators.

We establish the notion of post-selection persistency to characterize the theoretical properties of a sequence of estimators. Recall from Algorithm 1 that our proposed method

yields a sequence of estimators $\{\widehat{f}^{(\ell)}\}_{\ell=1}^{r}$. Also recall that we denote $\mathcal{S}^{(r-1)}$ to be the support of $\widehat{f}^{(r-1)}$. Let $\mathcal{F}^{(r)}$ be some function class with support constrained on $\delta(\mathcal{S}^{(r-1)}) = \mathcal{S}^{(r-1)} \cup \sigma(\mathcal{S}^{(r-1)})$. Conditioned on the support $\mathcal{S}^{(r-1)}$, we say that $\widehat{f}^{(r)}$ is *post-selection persistent* if

$$R(\widehat{f}^{(r)}) - \inf_{f \in \mathcal{F}^{(r)}} R(f) = o_P(1). \tag{9}$$

We will show that our proposed estimators are post-selection persistent in the context of nonparametric regression and nonparametric graphical models in Sections 3 and 4, respectively.

3. Nonparametric Regression

We apply the layer-wise learning strategy to the setting of nonparametric regression. We consider a nonparametric regression problem of a univariate response $Y \in \mathbb{R}$ on a d-dimensional covariates $\boldsymbol{X} \in [0,1]^d$:

$$Y = f(\boldsymbol{x}) + \epsilon,$$

where ϵ is the random noise variable. It is generally agreed upon in the literature that estimating a general multivariate function without restricting the function into a smaller function class $\mathcal{F}$ is infeasible.

Hastie and Tibshirani (1990) and Stone (1985) introduced a class of additive models of the form $f(\boldsymbol{x}) = \sum_{j=1}^{d} f_j(x_j)$, which decomposed the multivariate function $f(\cdot)$ into the summation of d univariate functions. One caveat of the additive model is the assumption that there are no interaction terms among the covariates. To address this issue, Lin (2000) proposed to estimate $f(\cdot)$ by assuming that it takes the form in (1) in the nonparametric regression setting. However, their proposal is infeasible for high-dimensional problem in which the number of covariates d and the order of interaction terms are large. Thus, they truncated (1) to only modeling two-way interaction terms. More specifically, they considered the following decomposition for $f(\cdot)$:

$$f(\boldsymbol{x}) = \mu + \sum_{j=1}^{d} f_j(x_j) + \sum_{j<k} f_{jk}(x_j, x_k).$$

Several authors have extended the aforementioned models to perform variable selection and estimation simultaneously (among others, Lin and Zhang, 2006; Ravikumar et al., 2009). However, Ravikumar et al. (2009) models only the main effects and Lin and Zhang (2006) is computationally infeasible for large d problems even when they model only the second order interaction terms. The nonparametric regression literature is vast and we refer the reader to several recent proposals for more references (see, Tibshirani, 2014; Fan et al., 2015; Lou et al., 2016).

We now apply the proposed layer-wise learning strategy for fitting a nonparametric regression. Our proposal is scalable to the dimension of the covariates and does not need to truncate model (1) to only modeling the two-way interaction terms. We show that the resulting sequence of estimators from our algorithm is post-selection persistent under the squared error risk in the high-dimensional setting in which $d > n$.

3.1 Method and Optimization Problem

Let $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$ be n independent pairs of observations. We assume that the covariates $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are standardized such that $\mathbf{x}_i \in [0,1]^d$ and that $y_i = f(\mathbf{x}_i) + \epsilon_i$ with $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i^2] < \infty$. The function $f(\cdot)$ is an arbitrary function and is not assumed to take the form of (1). To approximate the function $f(\cdot)$, we fit the model in (7) with the squared error loss function $\mathcal{L}_n(D, f) = \sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2/n$. This yields the optimization problem

$$\begin{aligned} &\underset{f \in \mathcal{H}^{(r)}, \mathcal{S}(f) = \delta(\mathcal{S}^{(r-1)})}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2, \\ &\text{subject to} \ \sum_{J \in \mathcal{S}^{(r-1)}} \|P_J(f)\|_2^2 + \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} \|P_J(f)\|_2 \le \tau, \end{aligned} \tag{10}$$

where $\tau > 0$ is a positive tuning parameter.

It is useful to write the function $f(\cdot)$ in terms of its basis function. Let $\{\phi_{j\ell}, \ell = 1, 2, \ldots\}$ denote a uniformly bounded basis with respect to H_j^m. Given $J = \{j_1, \ldots, j_r\}$, for any $f_J \in \mathcal{H}_J$, the basis expansion of f_J is

$$f_J = \sum_{1 \le k_1, \ldots, k_r < \infty} \theta_{j_1 \cdots j_r}^{k_1 \cdots k_r} \phi_{j_1 k_1}(x_{j_1}) \cdots \phi_{j_r k_r}(x_{j_r}). \tag{11}$$

In practice, we approximate (11) by its kth order basis expansion

$$\widetilde{f}_J = \sum_{1 \le k_1, \ldots, k_r \le k} \theta_{j_1 \cdots j_r}^{k_1 \cdots k_r} \phi_{j_1 k_1}(x_{j_1}) \cdots \phi_{j_r k_r}(x_{j_r}) = \boldsymbol{\phi}_J^T(\boldsymbol{x})\boldsymbol{\theta}_J, \tag{12}$$

where $\boldsymbol{\theta}_J = \text{vec}(\{\theta_{j_1 \cdots j_r}^{k_1 \cdots k_r}\})$ and $\boldsymbol{\phi}_J(\boldsymbol{x}) = \text{vec}(\{\phi_{j_1 k_1}(x_{j_1}) \cdots \phi_{j_r k_r}(x_{j_r})\})$.

Let $\boldsymbol{\Phi}_J$ denote the $n \times k^{|J|}$ matrix with rows $\boldsymbol{\phi}_J(\mathbf{x}_1), \ldots, \boldsymbol{\phi}_J(\mathbf{x}_n)$ and let $\mathbf{y} = (y_1, \ldots, y_n)^T$. We approximate (10) in terms of the kth order basis expansion:

$$\begin{aligned} &\underset{\boldsymbol{\theta}_J, \mathcal{S}(f) = \delta(\mathcal{S}^{(r-1)})}{\text{minimize}} \ \frac{1}{n}\Big\|\mathbf{y} - \sum_{J \in \mathcal{S}(f)} \boldsymbol{\Phi}_J \boldsymbol{\theta}_J\Big\|_2^2, \\ &\text{subject to} \ \frac{1}{n}\sum_{J \in \mathcal{S}^{(r-1)}} \|\boldsymbol{\Phi}_J \boldsymbol{\theta}_J\|_2^2 + \frac{1}{\sqrt{n}} \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} \|\boldsymbol{\Phi}_J \boldsymbol{\theta}_J\|_2 \le \tau. \end{aligned} \tag{13}$$

Instead of solving optimization problem in (13) directly, we consider solving the following problem

$$\underset{\boldsymbol{\theta}_J, \mathcal{S}(f) = \delta(\mathcal{S}^{(r-1)})}{\text{minimize}} \ \frac{1}{n}\Big\|\mathbf{y} - \sum_{J \in \mathcal{S}(f)} \boldsymbol{\Phi}_J \boldsymbol{\theta}_J\Big\|_2^2 + \lambda\Big(\frac{1}{n}\sum_{J \in \mathcal{S}^{(r-1)}} \|\boldsymbol{\Phi}_J \boldsymbol{\theta}_J\|_2^2 + \frac{1}{\sqrt{n}} \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} \|\boldsymbol{\Phi}_J \boldsymbol{\theta}_J\|_2\Big) \tag{14}$$

Problems (13) and (14) are equivalent in the sense that for a given tuning parameter $\lambda > 0$, there exists a $\tau > 0$ such that the two problems share the same solution, and vice versa.

Algorithm 2 Block Coordinate Descent Algorithm for Solving (14).

Initialize $\widehat{\boldsymbol{\theta}}_J^{(0)}$.
repeat
  for $J \in \delta(\mathcal{S}^{(r-1)})$ **do**
    Update the coefficients:

$$\widehat{\boldsymbol{\theta}}_J^{(t)} = \left(\boldsymbol{\Phi}_J^T \boldsymbol{\Phi}_J\right)^{-1} \left(\boldsymbol{\Phi}_J^T \mathbf{y} - \boldsymbol{\Phi}_J^T \sum_{J' \in \{\delta(\mathcal{S}^{(r-1)}) \setminus J\}} \boldsymbol{\Phi}_{J'}^T \widehat{\boldsymbol{\theta}}_{J'}^{(t-1)}\right).$$

    Penalize the coefficients:

$$\widehat{\boldsymbol{\theta}}_J^{(t)} = \begin{cases} \widehat{\boldsymbol{\theta}}_J^{(t)}/(1+\lambda) & \text{if } J \in \mathcal{S}^{(r-1)}, \\ \left(1 - \frac{\sqrt{n}\lambda}{2\|\boldsymbol{\Phi}_J \widehat{\boldsymbol{\theta}}_J^{(t)}\|_2}\right)_+ \widehat{\boldsymbol{\theta}}_J^{(t)} & \text{if } J \in \sigma(\mathcal{S}^{(r-1)}), \end{cases}$$

    where $(a)_+ = \max(0, a)$.
  end for
  Update $t = t + 1$.
until converge such that $\sum_J \left\|\widehat{\boldsymbol{\theta}}_J^{(t)} - \widehat{\boldsymbol{\theta}}_J^{(t-1)}\right\|_2 \le \epsilon$.

It can be verified that when $r = 1$, (14) is equivalent to the sparse additive model in Ravikumar et al. (2009).

Since (14) is quadratic in terms of $\boldsymbol{\theta}_J$ and both the penalty terms are convex, standard convexity theory implies the existence of a global minimizer. We propose a block coordinate descent algorithm to solve (14), which details are given in Algorithm 2. The convergence of block coordinate descent algorithm is studied in Tseng (2001). The derivation of Algorithm 2 is straightforward and hence omitted. In this section, our estimation procedure and algorithm are designed based on basis representation of the functions. We note that in principle, other nonparametric methods such as that of Wang et al. (2016) and Benkeser and van der Laan (2016) can be used to estimate the individual functions in our framework.

3.2 Post-Selection Persistency for Nonparametric Regression

In this section, we show that the sequence of estimators obtained from our proposal is post-selection persistent. The population version of the optimization problem in (10) is

$$\begin{aligned} &\underset{f \in \mathcal{H}^{(r)}, \mathcal{S}(f) = \delta(\mathcal{S}^{(r-1)})}{\text{minimize}} \ \mathbb{E}\left[(Y - f(\boldsymbol{X}))^2\right], \\ &\text{subject to} \ \sum_{J \in \mathcal{S}^{(r-1)}} \mathbb{E}\left[(P_J(f))^2\right] + \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} \sqrt{\mathbb{E}\left[(P_J(f))^2\right]} \le \tau, \quad \mathbb{E}\left[P_J(f)\right] = 0, \end{aligned} \tag{15}$$

where the expectation is taken with respect to $\boldsymbol{X}$ and the noise ϵ. To simplify our theoretical analysis, let $f_J(\boldsymbol{X}_J) = \beta_J g_J(\boldsymbol{X}_J)$ and consider the following equivalent population problem

$$\begin{aligned} &\underset{g \in \mathcal{H}^{(r)}, \beta_J, \mathcal{S}(g) = \delta(\mathcal{S}^{(r-1)})}{\text{minimize}} \ \mathbb{E}\left[\left(Y - \sum_{J \in \delta(\mathcal{S}^{(r-1)})} \beta_J g_J(\boldsymbol{X}_J)\right)^2\right], \\ &\text{subject to} \ \sum_{J \in \mathcal{S}^{(r-1)}} \beta_J^2 + \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} |\beta_J| \le \tau, \quad \mathbb{E}[P_J(g)] = 0, \quad \mathbb{E}\left[(P_J(g))^2\right] = 1. \end{aligned} \tag{16}$$

Problems (16) and (15) are equivalent in the sense that their solutions are equivalent. A similar formulation was also considered in Ravikumar et al. (2009) for sparse additive models.

Let $(\boldsymbol{X}, Y)$ denote a new pair of independent data and define the predictive risk as

$$R(f) = \mathbb{E}\left[(Y - f(\boldsymbol{X}))^2\right].$$

In this section, we assume that our estimator $\widehat{f}^{(r)}$ is chosen to minimize the empirical version of (16). Let

$$\mathcal{F}^{(r)} = \left\{ f : f(\boldsymbol{x}) = \sum_{J \in \delta(\mathcal{S}^{(r-1)})} \beta_J g_J(\boldsymbol{x}_J), \mathbb{E}[g_J] = 0, \|g_J\|_{\mathcal{H}_J} \le 1, \sum_{J \in \mathcal{S}^{(r-1)}} \beta_J^2 + \sum_{J \in \sigma(\mathcal{S}^{(r-1)})} |\beta_J| \le \tau \right\}.$$

The following theorem establishes that the sequence of estimators is post-selection persistent.

Theorem 1. *Let $s_0 = 1$ and let s_{r-1} be the cardinality of the support $\mathcal{S}^{(r-1)}$. Conditioned on $\mathcal{S}^{(r-1)}$, under the square error risk $R(f) = \mathbb{E}[(Y - f(\boldsymbol{X}))^2]$ and for any $1 \le r < 2m$, we have*

$$R(\widehat{f}^{(r)}) - \inf_{f \in \mathcal{F}^{(r)}} R(f) = O_P\left(\tau^2 \cdot \sqrt{\frac{r s_{r-1}^2 \log d}{n}}\right).$$

Thus, if $\tau = o([n/(r s_{r-1}^2 \log d)]^{1/4})$, the estimator $\widehat{f}^{(r)}$ is post-selection persistent.

In other words, Theorem 1 states that conditioned on the selected support on the $(r-1)$th step of our proposed method, $\mathcal{S}^{(r-1)}$, the estimator $\widehat{f}^{(r)}$ converges to the best rth order approximation of the form (1) with support constrained on $\delta(\mathcal{S}^{(r-1)})$. Given the support $\mathcal{S}^{(r-1)}$, the term s_{r-1} is a fixed constant that is much smaller than n. The proof of Theorem 1 involves obtaining the bracketing number of some function classes and applying empirical process tools to obtain an upper bound of the supremum between the empirical and the expected value of the function. The condition $r < 2m$ in Theorem 1 is needed to guarantee that the integral of the log bracketing number is well defined. The details are given in Appendix A.

Theorem 1 holds without assuming that the true regression function $f(\cdot)$ takes the form of SSANOVA framework in (1). In addition, we do not impose any distributional assumptions on $(Y, \boldsymbol{X})$. We recover the persistency result in Ravikumar et al. (2009) for the sparse additive model as a special case when $r = 1$ and $m = 2$.

3.3 From Post-Selection Persistency to Persistency

Post-selection persistency results can serve as motivations for intermediate steps of LASER. However, the results do not concern properties of the final estimator. In this section, we establish sufficient conditions such that a post-selection persistency result can be strengthen into a persistency result for the final estimator. For simplicity, we consider nonparametric regression models with two-way interaction terms, i.e., $f \in \mathcal{H}^{(2)}$.

In general, without any conditions on the bivariate functions f_{jk}, it is extremely challenging to show that the main effects f_j can be identified without modeling the two-way interaction effects. This problem is related to proving model selection consistency under model misspecification for sparse additive model, and such theoretical results have not been well established in the literature. In fact, the same problem is not well understood in the context of linear regression with two-way interaction effects until recently (Hao et al., 2018).

In the following, we impose sufficient conditions on the bivariate functions such that the active main effects can be identified at the first stage of LASER in the context of nonparametric regression. Therefore, conditioned on the correctly identified main effects, the estimator obtained from the second stage is persistent. With some abuse of notation, let $\bar{\mathcal{S}}^{(1)}$ and $\bar{\mathcal{S}}^{(2)}$ be two sets containing indices for the underlying active main and bivariate effects, respectively. Assume that $\bar{\mathcal{S}}^{(2)}$ satisfies the hierarchical structural assumption in (2).

Recall from Section 3.1 that we approximate the main effects by its kth order basis expansion, i.e., $\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}=(\boldsymbol{\theta}_1^T,\ldots,\boldsymbol{\theta}_d^T)^T} \| \sum_{j=1}^d f_j - \sum_{j=1}^d \boldsymbol{\theta}_j^T \boldsymbol{\phi}_j \|_2^2$. In the following proposition, our conditions are written in terms of an approximation of both the main and the partial derivatives of the bivariate function using first order basis expansion. To this end, we define some additional notation. Given a bivariate function $g(x_1, x_2)$, let

$$g^{(2)}(x_1,x_2)\big|_{a_1,a_2} = g(a_1,a_2) + \sum_{j\in\{1,2\}} \left(\frac{\partial g(a_1,a_2)}{\partial x_j}(x_j - a_j) + \frac{1}{2}\frac{\partial^2 g(a_1,a_2)}{\partial x_j^2}(x_j-a_j)^2 \right),$$

and

$$\boldsymbol{\beta}^* = \operatorname*{argmin}_{\boldsymbol{\beta}=(\boldsymbol{\beta}_1^T,\ldots,\boldsymbol{\beta}_d^T)^T} \left\| \sum_{j=1}^d f_j + \sum_{s<t} f_{st}^{(2)}\big|_{1/2,1/2} - \sum_{j=1}^d \boldsymbol{\beta}_j^T \boldsymbol{\phi}_j \right\|_2^2.$$

The following proposition establishes that the true underlying support, $\bar{\mathcal{S}}^{(1)}$, is a subset of the estimated support from first stage of LASER, $\mathcal{S}^{(1)}$.

Proposition 1. *Assume that $f \in \mathcal{H}^{(2)}$. Suppose that there exist a positive constant $C_{\min}$ such that the minimum eigenvalue $\lambda_{\min}(\boldsymbol{\Phi}_{\bar{\mathcal{S}}^{(1)}}^T \boldsymbol{\Phi}_{\bar{\mathcal{S}}^{(1)}}) \geq C_{\min} > 0$. Let $\rho_n^* = \min_{j\in\bar{\mathcal{S}}^{(1)}} \|\boldsymbol{\beta}_j^*\|_\infty$ and $q_n^* = \| \sum_{(s,t)\in\bar{\mathcal{S}}^{(2)}} \partial^2_{x_s x_t} f_{st} \|_\infty$. Assume that $\sqrt{|\bar{\mathcal{S}}^{(1)}|k} \cdot q_n^*/\rho_n^* = o(1)$, $\sqrt{k} q_n^*/\lambda = o(1)$, and*

$$\frac{1}{\rho_n^*}\left[\sqrt{\log(|\bar{\mathcal{S}}^{(1)}|k)/n} + |\bar{\mathcal{S}}^{(1)}|^{3/2}/k + \lambda\sqrt{|\bar{\mathcal{S}}^{(1)}|k} \right] = o(1).$$

We have $\mathbb{P}(\mathcal{S}^{(1)} \supseteq \bar{\mathcal{S}}^{(1)}) \to 1$.

The proof of Proposition 1 is similar to that of Theorem 2 in Ravikumar et al. (2009): we provide a sketch proof in Appendix B. Intuitively, the conditions on q_n^* states that the bivariate functions should be sufficiently smooth relative to the signal ρ_n^*. As a reviewer pointed out, this implies that the bivariate functions should be close to linear, relative to the main effects. We have removed the incoherence condition in Ravikumar et al. (2009) and we now allow the active and non-active main effects to be correlated. Relaxing the imposed assumptions on the bivariate functions is out of the scope of this paper, and we leave it as an open problem for future research.

4. Nonparametric Graphical Models

Undirected graphical models, also known as Markov random field, have been used extensively to model the conditional dependence relationships among a set of random variables. In a graph, each node represents a random variable and an edge between two nodes indicates that the two random variables are conditionally dependent, given all of the other variables. Let $\boldsymbol{X}$ be a d-dimensional random variable with joint density function of the form

$$p(\boldsymbol{x}) = \frac{1}{Z(f)} \exp(-f(\boldsymbol{x})), \tag{17}$$

where $Z(f) = \int \exp(-f(\boldsymbol{x}))d\boldsymbol{x}$ is the partition function such that the density $p(\boldsymbol{x})$ integrates to one. By the Hammersley-Clifford theorem, a set of random variables $\boldsymbol{X}$ forms a Markov random field with respect to a graph G if $f(\boldsymbol{x})$ takes the form $f(\boldsymbol{x}) = \sum_{J\in J_G} f_J(\boldsymbol{x}_J)$, where J_G is a set of all cliques in G and f_J is the potential function. For a set $J \subseteq [d]$, if $f_J(\boldsymbol{x}_J) \neq 0$, then the set of random variables $\boldsymbol{X}_J$ forms a clique and are conditionally dependent given all of the other variables.

Currently, most of the research on graphical models are limited to the case when the maximal clique is of size two. This is referred to as the pairwise Markov random field with the following joint density

$$p(\boldsymbol{x}) = \frac{1}{Z(f)} \exp\left\{ -\sum_{j=1}^d f_j(x_j) - \sum_{j<k} f_{jk}(x_j,x_k) \right\}. \tag{18}$$

Under the pairwise Markov random field, the jth and kth random variables are conditionally independent if and only if $f_{jk}(x_j,x_k) = 0$. The pairwise Markov random field in (18) is fully nonparametric and consists of many recently studied pairwise graphical models as its special cases.

Example 1. Gaussian graphical models*: Let $\boldsymbol{X} \sim N_d(\boldsymbol{0}, \boldsymbol{\Sigma})$ and let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ be the inverse covariance matrix. The Gaussian graphical model has joint density*

$$p(\boldsymbol{x}) \propto \exp\left\{ -\frac{1}{2}\sum_{j=1}^d \Theta_{jj} x_j^2 - \sum_{j=1}^{d-1}\sum_{k>j} \Theta_{jk} x_j x_k \right\}.$$

Thus, the Gaussian graphical model is a special case of (18) with $f_{jk}(x_j,x_k) = \Theta_{jk}x_jx_k$ and $f_j(x_j) = \frac{1}{2}\Theta_{jj}x_j^2$. The Gaussian graphical model is well studied in the literature (see, for instance, Meinshausen and Bühlmann, 2006; Yuan and Lin, 2007; Friedman et al., 2008; Rothman et al., 2008; Peng et al., 2009; Ravikumar et al., 2011; Cai et al., 2011; Sun and Zhang, 2013; Tan et al., 2014, 2015; Liu and Luo, 2015; Drton and Maathuis, 2017).

Example 2. Exponential family graphical models*: The exponential family graphical model has joint density*

$$p(\boldsymbol{x}) \propto \exp\left\{\sum_{j=1}^{d}(t(x_j)+C(x_j)) + \sum_{j=1}^{d-1}\sum_{k>j}\Theta_{jk}t(x_j)t(x_k)\right\},$$

where $t(x_j)$ is a univariate sufficient statistics function, $C(x_j)$ is some function of x_j specified by the exponential family distribution, and Θ_{jk} is the canonical parameter. Thus, this is a special case of (18) with $f_{jk}(x_j,x_k) = -\Theta_{jk}t(x_j)t(x_k)$ and $f_j(x_j) = -t(x_j) - C(x_j)$. This model is recently studied by many authors (see, for instance, Yang et al., 2013, 2015; Tan et al., 2016; Yang et al., 2018; Chen et al., 2014)

Example 3. Nonparanormal graphical models*: Let $g = \{g_1,\ldots,g_d\}$ be a set of monotone univariate functions. A d-dimensional random vector $\boldsymbol{X}$ has a nonparanormal distribution $\boldsymbol{X} \sim \mathrm{NPN}_d(g,\boldsymbol{\Sigma})$ if $g(\boldsymbol{X}) \sim N_d(\boldsymbol{0},\boldsymbol{\Sigma})$. Let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$. Then, the nonparanormal graphical model has joint density*

$$p(\boldsymbol{x}) \propto \exp\left\{\sum_{j=1}^{d}\left(-\frac{1}{2}\Theta_{jj}g_j(x_j)^2 + \log\left|g_j'(x_j)\right|\right) - \sum_{j=1}^{d-1}\sum_{k>j}\Theta_{jk}g_j(x_j)g_k(x_k)\right\}.$$

This is a special case of (18) with $f_{jk}(x_j,x_k) = \Theta_{jk}g_j(x_j)g_k(x_k)$ and $f_j(x_j) = \Theta_{jj}g_j(x_j)^2/2 - \log|g_j'(x_j)|$. This model is studied in Liu et al. (2009) and Liu et al. (2012).

We consider modeling the Markov random field in (17) under the SSANOVA framework, that is, the function $f(\cdot)$ can be decomposed as in (1). This general model has been considered in the literature and an estimate of $f(\cdot)$ can be obtained by optimizing over the penalized maximum likelihood function (see, for instance, Leonard, 1978; Silverman, 1982; Gu and Wang, 2003; Jeon and Lin, 2006). However, due to the log-partition function $Z(f)$, the proposed algorithms are not scalable to large dimension and higher order interaction terms. To the best of our knowledge, most methods involve truncating the functional decomposition (1) to only modeling the two-way interaction terms, which corresponds to pairwise nonparametric graphical models in (18).

In this section, we propose a novel method to estimate nonparametric graphical models of the form (17) without restricting it to only modeling two-way interaction terms. More specifically, we are interested in estimating nonparametric graphical models with joint density function

$$p(\boldsymbol{x}) = \frac{1}{Z(f)}\exp(-f(\boldsymbol{x})) \qquad \text{and} \qquad f(\boldsymbol{x}) = \mu + \sum_{j=1}^{d} f_j(x_j) + \sum_{j<k} f_{jk}(x_j,x_k) + \cdots.$$

Rather than using the penalized maximum likelihood function to estimate $f(\cdot)$, we propose to estimate $f(\cdot)$ under the score matching loss function proposed in Hyvärinen (2005) and Hyvärinen (2007), which is independent of the log-partition function $Z(f)$ that is computationally intractable. Thus, our algorithm is scalable to the dimension of the random variables as well as the size of cliques among the random variables compared to existing proposals. We also show that our proposal is post-selection persistent under the score matching risk in the high-dimensional setting.

4.1 Score Matching Loss

The score matching loss function was introduced to estimate densities of the form (17), which involves a computationally intractable log-partition function $Z(f)$. In the following, we provide a brief discussion on the score matching loss and refer the reader to Hyvärinen (2005) and Hyvärinen (2007) for more details. Let $\boldsymbol{X}$ be a d-dimensional continuous random vector with distribution $\mathcal{P}$ and joint density function $p(\cdot)$. For a twice differentiable function $f:\mathbb{R}^d \to \mathbb{R}$, we define the Laplacian operator and the gradient of $f(\cdot)$ as

$$\Delta f(\boldsymbol{x}) = \sum_{j=1}^{d}\frac{\partial^2}{\partial x_j^2}f(\boldsymbol{x}) \in \mathbb{R} \qquad \text{and} \qquad \nabla_j f(\boldsymbol{x}) = \frac{\partial}{\partial x_j}f(\boldsymbol{x}), \tag{19}$$

respectively. For a distribution $\mathcal{Q}$ with density $q(\cdot)$, Hyvärinen (2005) defined the score matching loss of $\mathcal{Q}$ with respect to $\mathcal{P}$ as

$$\frac{1}{2}\int_{\mathbb{R}^d} p(\boldsymbol{x})\|\nabla\log p(\boldsymbol{x}) - \nabla\log q(\boldsymbol{x})\|_2^2 d\boldsymbol{x}. \tag{20}$$

Equation (20) is also referred to as the Fisher divergence. It can be seen that (20) is minimized as a function of $\mathcal{Q}$ when $\mathcal{Q} = \mathcal{P}$, which depends on the true distribution $\mathcal{P}$. Hyvärinen (2005) showed that under the condition that $\|p(\boldsymbol{x})\nabla\log q(\boldsymbol{x})\|_2 \to 0$ as $\|\boldsymbol{x}\|_2 \to \infty$, the score matching loss can be rewritten as

$$\int_{\mathbb{R}^d} p(\boldsymbol{x})\left[\Delta\log q(\boldsymbol{x}) + \frac{1}{2}\|\nabla\log q(\boldsymbol{x})\|_2^2\right]d\boldsymbol{x} + C, \tag{21}$$

where C is a constant that is independent of $\mathcal{Q}$. The term in the integrand (21) is referred to as the Hyvärinen scoring rule, and no longer depends on the true distribution $\mathcal{P}$ and the log-partition function. Thus, an estimator of $f(\cdot)$ can be obtained by minimizing the Hyvärinen scoring rule. The statistical properties of the estimator obtained by minimizing the Hyvärinen scoring rule have been studied in the classical setting in which $n > d$ (among others, Hyvärinen, 2005, 2007; Forbes and Lauritzen, 2015).

Recently, Lin et al. (2016) proposed to estimate parametric pairwise graphical models in the high-dimensional setting in which $d > n$ under the score matching loss function. In addition, in his dissertation, Janofsky (2015) proposed to estimate fully nonparametric pairwise graphical models as in (18) using the score matching loss function. However, their proposal is limited to pairwise interactions between two random variables and are not able to estimate clique of size greater than two in a graph. We now generalize the aforementioned proposals to accommodate general nonparametric graphical models of the form (17) using the score matching loss function.

We start with establishing a proper score matching loss function for estimating nonparametric graphical models in (17). In the context of nonparametric graphical model setting, we consider distribution $\mathcal{P}$ that is supported on $[0,1]^d$. The Hyvärinen scoring rule (21) no longer applies since it is derived for distribution that is supported on $\mathbb{R}^d$. We now make a modification to the Hyvärinen scoring rule for densities with support $[0,1]^d$. To this end, we define $r_j(x_j)$ to be a function of x_j and $\mathbf{r}(\boldsymbol{x}) = (r_1(x_1),\ldots,r_d(x_d))^T$. We define $\mathbf{r}'(\boldsymbol{x}) = (r_1'(x_1),\ldots,r_d'(x_d))^T$ to be the element-wise differentiation of the vector $\mathbf{r}(\boldsymbol{x})$, that

is, $r_j'(x_j) = \partial r_j(x_j)/\partial x_j$. We define the modified score matching loss of $\mathcal{Q}$ with respect to $\mathcal{P}$ as

$$\frac{1}{2}\int_{[0,1]^d} p(\boldsymbol{x})\,\|\mathbf{r}(\boldsymbol{x})\circ[\nabla\log p(\boldsymbol{x})-\nabla\log q(\boldsymbol{x})]\|_2^2\,d\boldsymbol{x}, \tag{22}$$

where $\circ$ is the Hadamard product between two vectors. The following lemma establishes a scoring rule similar to that of (21) for random variables $\boldsymbol{X}\in[0,1]^d$.

Lemma 1. *Assume that the density $p(\boldsymbol{x})$ for $\mathcal{P}$ satisfies the regularity conditions that*

$$\lim_{x_j\to 0} p(\boldsymbol{x})\cdot\nabla_j\log q(\boldsymbol{x})r_j^2(x_j)\to 0 \qquad\text{and}\qquad \lim_{x_j\to 1} p(\boldsymbol{x})\cdot\nabla_j\log q(\boldsymbol{x})r_j^2(x_j)\to 0.$$

for any $1\le j\le d$. Then, the modified score matching loss can be written as

$$\int_{[0,1]^d} p(\boldsymbol{x})S(\boldsymbol{x},q)d\boldsymbol{x}+C,\ \text{where}$$

$$S(\boldsymbol{x},q)=2\left(\mathbf{r}(\boldsymbol{x})\circ\mathbf{r}'(\boldsymbol{x})\right)^T\nabla\log q(\boldsymbol{x})+(\mathbf{r}(\boldsymbol{x})\circ\mathbf{r}(\boldsymbol{x}))^T\nabla^2\log q(\boldsymbol{x})+\frac{1}{2}\|\mathbf{r}(\boldsymbol{x})\circ\nabla\log q(\boldsymbol{x})\|_2^2, \tag{23}$$

C is some constant independent of $\mathcal{Q}$, and $\nabla^2\log q(\boldsymbol{x})$ is a vector of second order derivative of $\boldsymbol{x}$.

The assumption in Lemma 1 requires that $r_j(x_j)\to 0$ as $x_j\to 0$ and $x_j\to 1$. One possible choice of r_j is $r_j(x_j)=x_j(1-x_j)$, which was considered in Janofsky (2015) in the context of pairwise nonparametric graphical models. Thus, an estimate of $f(\cdot)$ for the nonparametric graphical model (17) can be obtained by minimizing the modified scoring rule

$$S(\boldsymbol{x},f)=-2\sum_{j=1}^d r_j(x_j)r_j'(x_j)f^{(j)}(\boldsymbol{x})-\sum_{j=1}^d r_j^2(x_j)f^{(jj)}(\boldsymbol{x})+\frac{1}{2}\sum_{j=1}^d r_j^2(x_j)\left(f^{(j)}(\boldsymbol{x})\right)^2, \tag{24}$$

where $f^{(j)}$ and $f^{(jj)}$ are the first and second order derivative of $f(\boldsymbol{x})$ with respect to x_j, respectively. From (24), we see that the score matching loss function depends only on $\nabla\log q(\boldsymbol{x})$ and $\nabla^2\log q(\boldsymbol{x})$, which are free of the log-partition function $Z(f)$.

4.2 Method and Optimization Problem

Let $\mathbf{x}_1,\ldots,\mathbf{x}_n$ be n independent and identically distributed observations drawn from $\mathcal{P}$ with support $[0,1]^d$. To estimate the conditional dependencies among the random variables, we fit the model in (7) with the score matching loss function $\mathcal{L}_n(D,f)=\frac{1}{n}\sum_{i=1}^n S(\mathbf{x}_i,f)$, where $S(\mathbf{x}_i,f)$ is as defined in (24).

In the context of graphical models, the main effect f_j is always non-zero unless the jth random variable is uniformly distributed. Thus, we start estimating nonparametric graphical models with the second step of our algorithm since we do not have to perform

variable selection for the main effects. This yields the following optimization problem at the rth step of our proposed algorithm

$$\begin{aligned}&\underset{f\in\mathcal{H}^{(r)},\mathcal{S}(f)=\delta(\mathcal{S}^{(r-1)})}{\text{minimize}}\ \frac{1}{n}\sum_{i=1}^n S(\mathbf{x}_i,f),\\ &\text{subject to}\ \sum_{J\in\mathcal{S}^{(r-1)}}\|P_J(f)\|_2^2+\sum_{J\in\sigma(\mathcal{S}^{(r-1)})}\|P_J(f)\|_2\le\tau.\end{aligned} \tag{25}$$

Similar to Section 3.1, we solve the penalized version of (25) in terms of its basis expansion. To this end, we define additional notation. Let $\boldsymbol{\phi}_J^{(j)}(\boldsymbol{x})$ and $\boldsymbol{\phi}_J^{(jj)}$ be the first and second order derivative of $\boldsymbol{\phi}_J(\boldsymbol{x})$ with respect to x_j, respectively. Similarly, let $\boldsymbol{\Phi}_J^{(j)}$ and $\boldsymbol{\Phi}_J^{(jj)}$ denote the $n\times k^{|J|}$ matrix with rows $\boldsymbol{\phi}_J^{(j)}(\mathbf{x}_1)^T,\ldots,\boldsymbol{\phi}_J^{(j)}(\mathbf{x}_n)^T$ and $\boldsymbol{\phi}_J^{(jj)}(\mathbf{x}_1)^T,\ldots,\boldsymbol{\phi}_J^{(jj)}(\mathbf{x}_n)^T$, respectively. Writing (25) in terms of its basis expansion yields the optimization problem

$$\underset{\boldsymbol{\theta}_J,\mathcal{S}(f)=\delta(\mathcal{S}^{(r-1)})}{\text{minimize}}\ \frac{1}{n}\sum_{i=1}^n S_\phi(\mathbf{x}_i,\boldsymbol{\theta})+\lambda\Big(\frac{1}{n}\sum_{J\in\mathcal{S}^{(r-1)}}\|\boldsymbol{\Phi}_J\boldsymbol{\theta}_J\|_2^2+\frac{1}{\sqrt{n}}\sum_{J\in\sigma(\mathcal{S}^{(r-1)})}\|\boldsymbol{\Phi}_J\boldsymbol{\theta}_J\|_2\Big), \tag{26}$$

where $S_\phi(\mathbf{x}_i,\boldsymbol{\theta})$ is a function of $\boldsymbol{\Phi}_J,\boldsymbol{\Phi}_J^{(j)}$, and $\boldsymbol{\Phi}_J^{(jj)}$. Problem (26) is convex and can be solved directly via the block coordinate descent algorithm (Tseng, 2001).

The block coordinate descent algorithm involves cycling through the updates for $\boldsymbol{\theta}_J$ for all J until convergence. Since the loss function $\frac{1}{n}\sum_{i=1}^n S_\phi(\mathbf{x}_i,\boldsymbol{\theta})$ is quadratic in $\boldsymbol{\theta}_J$, there is a closed form update for any $J\in\mathcal{S}^{(r-1)}$. However, for $J\in\sigma(\mathcal{S}^{(r-1)})$, there is no closed form update for $\boldsymbol{\theta}_J$ due to the composite function in the group lasso penalty. In the context of pairwise nonparametric graphical models, Janofsky (2015) proposed to use the alternating direction method of multiplies algorithm to obtain updates for $\boldsymbol{\theta}_J$ with $J\in\sigma(\mathcal{S}^{(1)})$. For higher order terms, a similar algorithm can be used. We omit the details and refer the reader to Janofsky (2015) for the derivation of the algorithm.

4.3 Post-Selection Persistency for the Nonparametric Graphical Model

We now establish that the sequence of estimators $\{\widehat{f}^{(\ell)}\}_{\ell=1}^r$ obtained from solving (25) is post-selection persistent under the score matching risk function. For density estimation, a natural risk function is the distance between two density functions. There are various measures to quantify the distance between two density functions p and q. One of the most popular distance measure is the Kullback-Leibler divergence. Since we derive the score matching loss function based on the Fisher divergence criterion, it is natural to define the risk function using the Fisher divergence:

$$R(p,q)=\frac{1}{2}\int_{[0,1]^d} p(\boldsymbol{x})\left\|\mathbf{r}(\boldsymbol{x})\circ\nabla\log\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right\|_2^2 d\boldsymbol{x}. \tag{27}$$

The population version of the optimization problem in (25) is

$$\begin{aligned}&\underset{f\in\mathcal{H}^{(r)},\mathcal{S}(f)=\delta(\mathcal{S}^{(r-1)})}{\text{minimize}}\ \mathbb{E}\left[S(\boldsymbol{X},f)\right],\\ &\text{subject to}\ \sum_{J\in\mathcal{S}^{(r-1)}}\mathbb{E}\left[(P_J(f))^2\right]+\sum_{J\in\sigma(\mathcal{S}^{(r-1)})}\sqrt{\mathbb{E}\left[(P_J(f))^2\right]}\le\tau,\quad \mathbb{E}\left[P_J(f)\right]=0,\end{aligned} \tag{28}$$

where the expectation is taken with respect to the random variables $\boldsymbol{X}$. Similar to Section 3.2, we let $f_J(\boldsymbol{X}_J) = \beta_J g_J(\boldsymbol{X}_J)$ and consider the following equivalent population problem

$$\begin{aligned} &\underset{g\in\mathcal{H}^{(r)},\beta_J,\mathcal{S}(g)=\delta(\mathcal{S}^{(r-1)})}{\text{minimize}} \quad \mathbb{E}\left[S(\boldsymbol{X},\boldsymbol{\beta},g)\right], \\ &\text{subject to} \quad \sum_{J\in\mathcal{S}^{(r-1)}} \beta_J^2 + \sum_{J\in\sigma(\mathcal{S}^{(r-1)})} |\beta_J| \le \tau, \quad \mathbb{E}[P_J(g)] = 0, \quad \mathbb{E}\left[(P_J(g))^2\right] = 1. \end{aligned} \tag{29}$$

For theoretical purposes, at the rth step of LASER, we assume that the estimator is chosen to minimize the empirical version of (29). Recall that

$$\mathcal{F}^{(r)} = \left\{ f : f(\boldsymbol{x}) = \sum_{J\in\delta(\mathcal{S}^{(r-1)})} \beta_J g_J(\boldsymbol{x}_J), \mathbb{E}[g_J] = 0, \|g_J\|_{\mathcal{H}_J} \le 1, \sum_{J\in\mathcal{S}^{(r-1)}} \beta_J^2 + \sum_{J\in\sigma(\mathcal{S}^{(r-1)})} |\beta_J| \le \tau \right\}.$$

We consider the following density function class

$$\mathcal{Q}^{(r)} = \{q \mid q \propto \exp(-f(\boldsymbol{x})), f \in \mathcal{F}^{(r)}\}.$$

We now state the main theorem on the post-selection persistency property of the estimator $\widehat{p}^{(r)}$ obtained from the rth step of LASER.

Theorem 2. *Let $\widehat{p}^{(r)} \propto \exp(-\widehat{f}^{(r)})$ and let s_{r-1} be the cardinality of the support $\mathcal{S}^{(r-1)}$. Given $\mathcal{S}^{(r-1)}$, for any $1 \le r < 2(m-2)$, $\widehat{p}^{(r)}$ is post-selection persistent under the Fisher divergence risk function*

$$R(p, \widehat{p}^{(r)}) - \inf_{q\in\mathcal{Q}^{(r)}} R(p,q) = O_P\left(\tau^2\sqrt{\frac{r^3 s_{r-1}^2 \log d}{n}}\right).$$

Thus, if $\tau = o([n/(r^3 s_{r-1}^2 \log d)]^{1/4})$, then the estimator $\widehat{p}^{(r)}$ is post-selection persistent given $\mathcal{S}^{(r-1)}$.

Theorem 2 states that conditioned on the support, $\mathcal{S}^{(r-1)}$, the estimator $\widehat{p}^{(r)}$ converges to the best rth order approximation of the form in $\mathcal{Q}^{(r)}$. The proof of Theorem 2 is given in Appendix D.

5. Numerical Studies

We perform numerical studies for both nonparametric regression and nonparametric graphical models.

5.1 Nonparametric Regression

We perform extensive numerical studies to evaluate the performance of our proposal for fitting multivariate nonparametric regression. In all of our numerical studies, we generate a training set and a test set. Each model is fit using the training set, and the trained model is used to predict the response on the test set. To compare the performance across different

methods, we calculate the sum of squares error between the predicted response and the true response from the test set. These results are reported in Tables 1—3. In addition, we report in Appendix F the true and false positive rates for the main effects and interaction effects, defined as the proportion of correctly estimated active variables and the proportion of inactive variables that are incorrectly estimated to be active, respectively.

Seven approaches are compared in our numerical studies: our proposal, `LASER`; the sparse additive model, `SpAM` (Ravikumar et al., 2009); the nonparametric additive regression model with two-way interactions, `VANISH` (Radchenko and James, 2010); the backtracking method for modeling high-dimensional linear regression with two-way interaction terms, `BT` (Shah, 2016); the convex modeling of interactions with strong heredity, `FAMILY` (Haris et al., 2016); the regularization approach for high-dimensional quadratic regression, RAMP (Hao et al., 2018); the oracle approach by assuming that the active variables were known a priori, `ORACLE`. In particular, the `ORACLE` is obtained by fitting nonparametric regression model (14) using only the active variables with the ridge penalty for smoothness.

Our proposal `LASER`, `SpAM`, `VANISH`, and `ORACLE` are nonparametric. `LASER` involves fitting the nonparametric regression model (14) sequentially. In each step of Algorithm 1, we select the tuning parameter using a five-fold cross-validation on the training data set. Algorithm 1 is stopped when there is no more higher order interaction terms to be estimated. We fit `LASER` using the kth basis expansion with $k = 3$. Note that the solution for `SpAM` can be obtained from the first layer of `LASER`. For `VANISH`, we simply use the default setting as in Radchenko and James (2010) and select the tuning parameter with cross-validation. For `ORACLE`, we select the tuning parameter that yields the smallest sum of squares error on the test set. In other words, `ORACLE` serves as a gold standard for nonparametric regression. The `FAMILY`, `RAMP`, and `BT` are sparse high-dimensional linear regression with two-way interaction terms. The tuning parameters for `RAMP` are selected using the extended BIC described in Hao et al. (2018). For `FAMILY` and `BT`, we consider a fine grid of tuning parameters and report the best results. In other words, we are giving unfair advantage to `FAMILY` and `BT`.

Most methods that estimate pairwise interaction terms are not computationally feasible for high-dimensional problem when d is large. Therefore, we consider both the low-dimensional and high-dimensional settings in Sections 5.1.1 and 5.1.2, respectively. We then perform numerical studies to assess how correlation among the covariates affects the performance of `LASER` in Section 5.1.3.

5.1.1 Low-Dimensional Setting with Two-Way Interactions

In our simulation studies, we generate $\epsilon \sim N(0,1)$ and each element of $\boldsymbol{X}$ from a uniform distribution on the interval $[0,1]$. We consider the following regression models with $d = 30$ covariates and $n = \{200, 400\}$:

A1 — A linear regression model with two-way interaction terms:

$$y = x_1 + x_2 + x_3 + 5x_1x_2 - 2x_1x_3 + 5x_2x_3 + \epsilon.$$

A2 — A non-linear regression model with two-way interaction terms (product of two individual functions):

$$f_1(z) = \sqrt{2}\,[\sin(6z) - 0.0066]\,, \qquad f_2(z) = \sqrt{11}[(2z-1)^2 - 1/3],$$
$$f_3(z) = \sqrt{12}(z - 1/2), \qquad f_4(z) = \sqrt{16.6}[\exp(-5z) - 0.2],$$
$$f_5(z) = \sqrt{50}[1/(1+z) - 0.69],$$
$$y = \sum_{j=1}^{4} f_j(x_j) + f_3(x_1)f_3(x_2) + f_5(x_2)f_4(x_4) + f_3(x_3)f_4(x_4) + \sqrt{0.5}\epsilon,$$

where the constants are designed such that each function has mean zero and variance approximately one.

A3 — A non-linear regression model with two-way interaction terms (bivariate functions that cannot be decomposed as product of two individual functions):

$$y = \sum_{j=1}^{3} f_j(x_j) + \sqrt{19}(\sqrt{x_1x_2} - 4/9) + \sqrt{50}[\exp(-5x_2x_3) - 0.438] + \sqrt{0.5}\epsilon,$$

where $f_j(x_j)$ is as defined in Scenario A2.

The results, averaged over 200 data sets, are reported in Table 1.

Table 1: The sum of squares error (standard error) out of 200 test samples for the three different scenarios in Section 5.1.1, averaged over 200 data sets. The results are for models trained with n training samples. Numbers are rounded to the nearest integer.

	Scenario A1		Scenario A2		Scenario A3	
	$n = 200$	$n = 400$	$n = 200$	$n = 400$	$n = 200$	$n = 400$
FAMILY	273 (2)	255 (2)	825 (10)	615 (7)	412 (3)	379 (3)
BT	216 (1)	203 (2)	766 (14)	751 (13)	411 (3)	407 (3)
RAMP	216 (2)	207 (2)	848 (21)	672 (22)	333 (9)	300 (9)
SpAM	323 (3)	296 (2)	825 (10)	743 (10)	91 (1)	84 (1)
VANISH	289 (3)	234 (2)	123 (3)	83 (1)	84 (1)	74 (1)
LASER	279 (4)	226 (2)	124 (4)	84 (1)	75 (1)	63 (1)
ORACLE	246 (2)	216 (2)	91 (1)	76 (1)	64 (1)	59 (1)

From Table 1, we see that BT and RAMP have the best performance in Scenario A1. This is not surprising since BT and RAMP are designed for modeling linear regression with two-way interaction terms. Both BT and RAMP outperform FAMILY that models the two-way interaction terms using a hierarchical penalty. For the nonparametric methods, SpAM has the highest sum of squares error since the true model contains two-way interaction terms that SpAM fails to model. VANISH and LASER have similar performance, and perform significantly better than SpAM. As we increase the sample size from $n = 200$ to $n = 400$, we see that the performance of LASER becomes more comparable to that of ORACLE, BT, and RAMP. For Scenarios A2—A3, FAMILY, RAMP, and BT have the highest sum of squares error since these methods are intended for modeling linear regression. Again, VANISH and LASER have similar

performance, and outperform SpAM. In summary, LASER is able to adaptively estimate the higher order terms accurately in both linear and non-linear regression settings.

5.1.2 High-Dimensional Setting with Three-Way Interactions

In this section, we consider the high-dimensional setting in which the number of variables d is potentially larger than the number of observations. FAMILY and VANISH are computationally infeasible since there are a total of $d + \binom{d}{2}$ parameters and functions to estimate. Moreover, we consider settings with a three-way interaction effect to illustrate the flexibility of our proposal compared to existing methods such as BT and RAMP that are limited to modeling two-way interaction terms. We generate ϵ and $\boldsymbol{X}$ as in Section 5.1.1. We consider three regression models with $d = \{200, 400\}$ covariates and $n = \{350, 700\}$:

B1 — A linear regression model with three-way interaction terms:

$$y = x_1 + x_2 + x_3 - 2x_1x_2 - 2x_1x_3 - 2x_2x_3 + 25x_1x_2x_3 + \sqrt{0.5}\epsilon.$$

B2 — A non-linear regression model with three-way interaction terms (product of individual functions):

$$y = \sum_{j=1}^{3} f_j(x_j) + f_3(x_1)f_3(x_2) + f_5(x_2)f_4(x_3) + f_1(x_1)f_4(x_3) + 25(x_1x_2x_3 - 1/8) + \sqrt{0.5}\epsilon,$$

where $f_j(\cdot)$ is as defined in Scenario A2.

B3 — A non-linear regression model with three-way interaction terms (bivariate functions that cannot be decomposed as product of two individual functions):

$$y = \sum_{j=1}^{3} f_j(x_j) + \sqrt{19}(\sqrt{x_1x_2} - 4/9) + \sqrt{50}[\exp(-5x_2x_3) - 0.438]$$
$$+ f_3(x_1)f_4(x_3) + 25(x_1x_2x_3 - 1/8) + \sqrt{0.5}\epsilon,$$

where $f_j(\cdot)$ is as defined in Scenario A2.

The results, averaged over 200 data sets, are reported in Table 2.

From Table 2, we see that LASER has the best performance across all scenarios since it is the only method that models the three-way interaction term. The sum of squares error is quite close to that of ORACLE even in the high-dimensional setting when $n = 350$ and $d = 400$. As we increase the sample size, we see that the performance of LASER becomes more similar to that of ORACLE.

5.1.3 Two-Way Interactions With Correlated Data

In this section, we assess the performance of LASER when the covariates $\boldsymbol{X}$ are correlated. We generate $\epsilon \sim N(0, 1)$ and $\boldsymbol{X} \sim N_d(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}_{jk} = \rho^{|j-k|}$ for $1 \le j, k \le d$. For Scenario C2, we normalize the covariates such that the observed values lie within the unit interval. We consider the following regression models with $d = 30$, $n = 200$, and $\rho = \{0, 0.2, 0.4, 0.6, 0.8\}$:

Table 2: The sum of squares error (standard error) out of 200 test samples for the three different scenarios in Section 5.1.2, averaged over 200 data sets. The results are for models trained with n training samples. Numbers are rounded to the nearest integer.

		Scenario B1		Scenario B2		Scenario B3	
		$n = 350$	$n = 700$	$n = 350$	$n = 700$	$n = 350$	$n = 700$
$d = 200$	BT	134 (1)	127 (1)	543 (6)	513 (5)	570 (5)	539 (4)
	RAMP	128 (1)	125 (1)	336 (6)	309 (3)	308 (4)	291 (3)
	SpAM	745 (11)	663 (9)	2203 (28)	2008 (21)	876 (13)	780 (10)
	LASER	95 (4)	71 (2)	149 (11)	102 (3)	127 (7)	82 (2)
	ORACLE	73 (1)	62 (1)	107 (3)	90 (2)	86 (1)	72 (1)
$d = 400$	BT	136 (1)	128 (1)	548 (6)	521 (6)	578 (5)	550 (5)
	RAMP	127 (1)	124 (1)	402 (9)	309 (4)	339 (7)	288 (3)
	SpAM	755 (11)	670 (9)	2224 (30)	2017 (24)	896 (13)	792 (11)
	LASER	97 (3)	70 (1)	166 (11)	95 (2)	122 (6)	80 (2)
	ORACLE	74 (1)	62 (1)	109 (3)	87 (2)	87 (1)	71 (1)

C1 — A linear regression model with two-way interaction terms:

$$y = x_1 + x_6 + x_{11} + 0.5x_1x_6 - 0.5x_1x_{11} + 0.5x_6x_{11} + \epsilon.$$

C2 — A non-linear regression model with two-way interaction terms:

$$y = \sum_{j=1}^{4} f_j(x_j) + f_3(x_1)f_3(x_6) + f_5(x_6)f_4(x_{16}) + f_3(x_{11})f_4(x_{16}) + \sqrt{0.5}\epsilon,$$

where the $f_j(\cdot)$ is as defined in Scenario A2.

The results, averaged over 200 data sets, are reported in Table 3. From Table 3, we see that LASER and ORACLE have similar sum of squares error. Moreover, LASER outperforms SpAM significantly. These results suggest that LASER is able to select the important main effects and interaction effects even when there are correlation among covariates.

5.2 Nonparametric Graphical Models

In this section, we perform some numerical studies to estimate nonparametric graphical models. We compare LASER to the graphical lasso (glasso) which estimate pairwise Gaussian graphical models (Friedman et al., 2008). We also consider the proposal of Liu et al. (2012) (kendall), a semiparametric approach for estimating nonparanormal graphical models. To evaluate the performance across different methods, we define the true positive rate as the proportion of correctly identified non-zeros, and the false positive rate as the proportion of zeros that are incorrectly identified to be non-zeros. In addition, we illustrate the main advantage of LASER in a stock price data by modeling three way cliques that quantify conditional dependencies among three variables, conditioned on the others.

Since existing approaches are limited to estimating pairwise graphical models, to compare across different methods, we first perform some numerical studies to estimate pairwise graphical models. In his dissertation, Janofsky (2015) has also conducted some simulation

Table 3: The sum of squares error (standard error) out of 200 test samples for the two scenarios in Section 5.1.3, averaged over 200 data sets. The results are for models trained with $n = 200$ training samples with $d = 30$ covariates. Numbers are rounded to the nearest integer.

		$\rho = 0$	$\rho = 0.2$	$\rho = 0.4$	$\rho = 0.6$	$\rho = 0.8$
Scenario C1	FAMILY	236 (2)	236 (2)	235 (2)	236 (2)	240 (2)
	BT	218 (2)	217 (2)	217 (2)	218 (2)	217 (1)
	RAMP	210 (1)	209 (2)	209 (1)	210 (1)	215 (2)
	SpAM	448 (5)	452 (5)	448 (5)	431 (5)	355 (4)
	VANISH	731 (17)	852 (66)	749 (28)	745 (21)	621 (25)
	LASER	345 (7)	353 (7)	348 (8)	340 (5)	339 (4)
	ORACLE	309 (5)	314 (6)	312 (7)	305 (6)	301 (5)
Scenario C2	FAMILY	309 (5)	315 (6)	306 (5)	318 (5)	328 (5)
	BT	254 (4)	261 (5)	250 (4)	247 (5)	225 (5)
	RAMP	250 (6)	261 (7)	256 (6)	254 (6)	269 (9)
	SpAM	213 (4)	224 (6)	216 (5)	214 (5)	191 (4)
	VANISH	379 (15)	393 (17)	403 (19)	413 (22)	452 (22)
	LASER	137 (3)	143 (5)	140 (4)	141 (4)	156 (4)
	ORACLE	117 (3)	122 (4)	119 (4)	122 (4)	120 (4)

studies to assess the performance of pairwise nonparametric graphical models using the score matching approach. More specifically, we consider two different simulation settings:

1. Gaussian graphical models: we simulate $\boldsymbol{X} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is generated such that $(\boldsymbol{\Sigma}^{-1})_{jk} = 0.4$ for $|k - j| = 1$, $(\boldsymbol{\Sigma}^{-1})_{jj} = 1$, and setting the other elements to zero.

2. Nonparametric graphical models: we simulate the data from the joint density

$$p(\boldsymbol{x}) \propto \exp\left(-\sum_{j=1}^{d} x_j - \sum_{j<k} \beta_{jk} x_j^2 x_k^2\right), \tag{30}$$

where $\beta_{jk} = 1$ for $|k - j| = 1$ and $\beta_{jk} = 0$ otherwise. Noting that the conditional distribution for each variable on the others is Gaussian, we employ a Gibbs sampler to simulate data from (30).

All of the aforementioned methods involve a sparsity tuning parameter. We applied a fine grid of tuning parameter values for all methods to obtain the curves in Figure 1. For Gaussian and nonparametric graphical models, we present results for $n = 100$ and $p = 25$, and $n = 300$ and $p = 25$, respectively. Results are averaged over 200 data sets.

For Gaussian graphical models, we see from Figure 1 that the graphical lasso and the proposal of Liu et al. (2012) outperform LASER when $n = 100$. This is not surprising since both of the methods are developed based on the Gaussian assumption and Gaussian copula assumption, whereas LASER is entirely nonparametric. We essentially loses some efficiency relative to the parametric and semiparametric approaches when the parametric and semiparametric assumptions are satisfied. For nonparametric graphical models, both Friedman et al. (2008) and Liu et al. (2012) are no longer able to estimate the graph accurately since

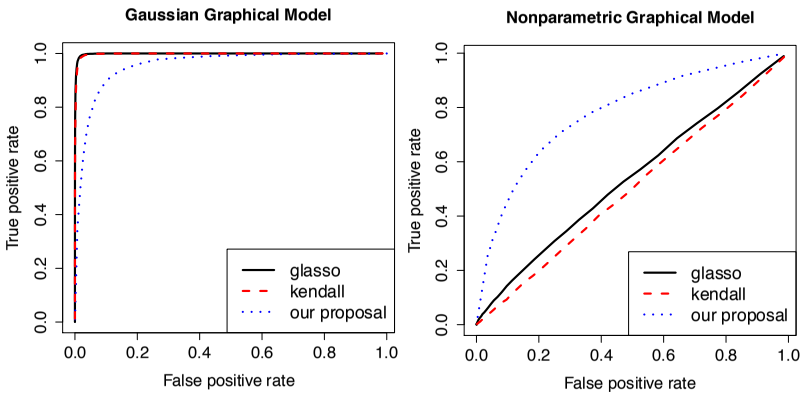


Figure 1: True and false positive rates, averaged over 200 data sets, for pairwise Gaussian and nonparametric graphical models. Left panel: Gaussian graphical models with $n = 100$ and $d = 25$. The curves are obtained by varying the tuning parameter. Right panel: nonparametric graphical models with $n = 300$ and $d = 25$.

the joint density in (30) is clearly not multivariate Gaussian. Their performances are similar to random guess even when we increase n by two-fold. LASER clearly outperforms the parametric approaches in this case. In conclusion, we sacrifice some performance in the parametric setting to gain flexibility in modeling nonparametric graphical models.

Next, we illustrate the main advantage of LASER by modeling three-way cliques that quantify conditional dependencies among three variables, conditioned on all of the other variables. To this end, we analyze the stock price data from Yahoo! Finance, which consists of daily closing prices for stocks in the S&P 500 index between January 1, 2003 and January 1, 2008. Stocks that are not consistently listed in the S&P 500 index during this time period are removed, leaving us with $n = 1258$ daily closing prices with 452 stocks. In this study, we categorize the stocks into six Global Industry Classification Standard sectors: Financials, Energy, Health Care, Information Technology, Materials, and Utilities.

The goal of our analysis is to understand the conditional dependence relationships among the six sectors. More specifically, we seek to learn the three-way conditional dependence relationships among the $d = 6$ sectors by modeling the three-way interaction terms in (17). Note that existing approaches for modeling graphical models in the literature are not able to model three-way cliques.

We first estimate a nonparametric graphical model with two-way interaction terms, corresponding to pairwise conditional dependencies. For the ease of interpretation, we pick the tuning parameter λ such that there are six edges in the estimated conditional independence graph. The results are summarized in Figure 2(a). We see that the three pairs of sectors "health care—materials", "health care—financials", and "financials—materials"

are conditionally dependent, given the other sectors. However, since we are estimating only the second order term at the second step of LASER, we cannot conclude that the three sectors financials, materials, and health care are jointly conditionally dependent.

To assess whether the three sectors are jointly conditionally dependent, we proceed to the next step of LASER for estimating three way interaction terms. We use the same tuning parameter to fit the model at the second step. The results are shown in Figure 2(b). Since the three way interaction term for health care, materials, and financials is estimated to be non-zero, we conclude that the three sectors are jointly conditionally dependent. Similar results hold for the sectors financials, materials, and information technology. Finally, LASER is terminated since there is no potential four way interaction terms to be estimated.

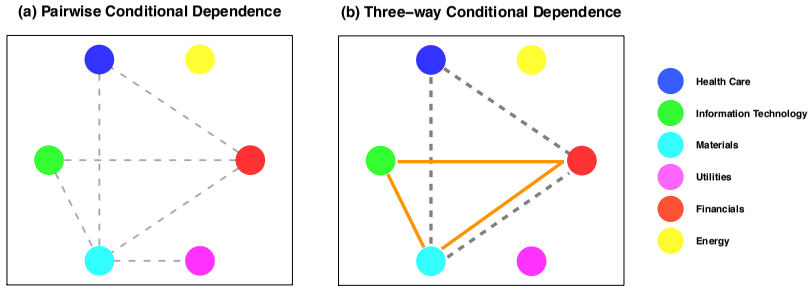


Figure 2: Estimated Conditional Dependence Graphs using the proposed method. Panel (a): Estimated pairwise conditional dependence relationships between two variables, conditioned on the others. Panel (b): Estimated three-way conditional dependence relationships among three variables, conditioned on the others.

6. Discussion

In this paper, we propose a layer-wise learning strategy (LASER) for fitting multivariate function under the SSANOVA framework. LASER provides a computationally feasible framework for estimating SSANOVA models with higher order interaction effects. In addition, we have shown that the estimators obtained from LASER is post-selection persistent. We illustrate LASER in the context of nonparametric regression and nonparametric graphical models problems. In the graphical modeling literature, most work have focused on estimating pairwise graphical models, which corresponds to estimating the conditional dependence relationships between pairs of variables. LASER provides an alternative way to estimate conditional dependence relationships among a set of more than two variables. More generally, LASER can be easily applied to other problems that involves estimating multivariate function such as the generalized nonparametric regression.

In the context of lasso regression with two-way interaction terms, Shah (2016) considered the scenario in which the main effects are useful for prediction only when certain two-

way interaction effects are present. In this case, two-stage methods that perform variable screening on the main effects at the first stage, and then include the interaction effects based on the identified main effects may fail to identify some important covariates. To address this issue, Shah (2016) proposed a backtracking algorithm. The main idea is to first select a few variables that are most correlated with the response, and then include its corresponding interaction effects into the model. This process is repeated until no more variables are included in the model.

As a reviewer pointed out, the backtracking algorithm of Shah (2016) can be modified to accommodate higher order terms in the context of nonparametric regression. The main idea is as follows: (i) select a few univariate functions that are most highly predictive of the response by fitting the additive model for the main effects; (ii) include its associated two-way interaction effects, and subsequently the higher order interaction effects into the model; (iii) repeat (i) and (ii) until no more variables are included in the model. It is out of the scope of this paper to study such an extension carefully and we leave it for future work.

Acknowledgments

We thank two reviewers and the associate editor for providing helpful comments that improved the quality of this manuscript. We thank Peter Radchenko for providing R code to fit the model proposed in Radchenko and James (2010). This research is partially supported by the NSF DMS-1811315.

Appendix A. Proof of Theorem 1

Let $\mathcal{S}^{(r-1)}$ be the support of $\widehat{f}^{(r-1)}$ and recall from (9) that conditional on the support $\mathcal{S}^{(r-1)}$, $\widehat{f}^{(r)}$ is post-selection persistent if

$$R(\widehat{f}^{(r)}) - \inf_{f\in\mathcal{F}^{(r)}} R(f) = o_P(1), \text{ where}$$

$$\mathcal{F}^{(r)} = \left\{ f : f(\boldsymbol{x}) = \sum_{J\in\delta(\mathcal{S}^{(r-1)})} \beta_J g_J(\boldsymbol{x}_J), \mathbb{E}[g_J] = 0, \|g_J\|_{\mathcal{H}_J} \le 1, \sum_{J\in\mathcal{S}^{(r-1)}} \beta_J^2 + \sum_{J\in\sigma(\mathcal{S}^{(r-1)})} |\beta_J| \le \tau \right\}.$$

For notational convenience, let $f^* = \arg\inf_{f\in\mathcal{F}^{(r)}} R(f)$. The goal is to show that $R(\widehat{f}^{(r)}) - R(f^*) = o_P(1)$.

Under the squared error loss, we define the risk $R(f)$ and empirical risk $\widehat{R}(f)$ as

$$R(f) = \mathbb{E}[(Y - f(\boldsymbol{X}))^2] \qquad \text{and} \qquad \widehat{R}(f) = \frac{1}{n}\sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2,$$

respectively. By the definition of f^*, we have $R(f^*) \le R(\widehat{f}^{(r)})$. Thus, by the triangle inequality, we have

$$\begin{aligned} 0 &\le R(\widehat{f}^{(r)}) - R(f^*) \le R(\widehat{f}^{(r)}) - \widehat{R}(\widehat{f}^{(r)}) + \widehat{R}(\widehat{f}^{(r)}) - R(f^*) \\ &\le |R(\widehat{f}^{(r)}) - \widehat{R}(\widehat{f}^{(r)})| + |\widehat{R}(f^*) - R(f^*)| \\ &\le 2 \sup_{f\in\mathcal{F}^{(r)}} |\widehat{R}(f) - R(f)|, \end{aligned} \tag{31}$$

where the third inequality holds by the definition that $\widehat{f}^{(r)}$ is the minimizer of $\widehat{R}(f)$, that is, $\widehat{R}(\widehat{f}^{(r)}) \le \widehat{R}(f^*)$. Thus, it suffices to obtain an upper bound on $\sup_{f\in\mathcal{F}^{(r)}} |\widehat{R}(f) - R(f)|$.

Let $\mathcal{J}_r = \delta(\mathcal{S}^{(r-1)}) \cup \{\emptyset\}$. With some abuse of notation, we write $g_J(\boldsymbol{x}_J) = y$ when J is an empty set. In addition, when $f(\boldsymbol{x}) = \sum_{J\in\delta(\mathcal{S}^{(r-1)})} \beta_J g_J(\boldsymbol{x}_J)$, we also write the risk function as $R(\beta, g)$. Then, for any $f \in \mathcal{F}^{(r)}$, the risk and empirical risk can be rewritten as

$$R(\beta, g) = \sum_{J,J'\in\mathcal{J}_r} \beta_J\beta_{J'}\mathbb{E}[g_J(\boldsymbol{X}_J)g_{J'}(\boldsymbol{X}_{J'})] \text{ and } \widehat{R}(\beta, g) = \frac{1}{n}\sum_{i=1}^n \sum_{J,J'\in\mathcal{J}_r} \beta_J\beta_{J'} g_J(\mathbf{x}_{iJ})g_{J'}(\mathbf{x}_{iJ'}),$$

respectively. Thus, for all (β, g), we have

$$|\widehat{R}(\beta, g) - R(\beta, g)| \le \left(\sum_{J\in J_r} |\beta_J|\right)^2 \max_{J\in\mathcal{J}_r} \sup_{g_J\in\mathcal{H}_J, g_{J'}\in\mathcal{H}_{J'}} (\mathbb{E}_n - \mathbb{E})[g_J g_{J'}],$$

where $\mathbb{E}_n[g_J g_{J'}] = n^{-1}\sum_{i=1}^n g_J(\mathbf{x}_{iJ})g_{J'}(\mathbf{x}_{iJ'})$. Let s_{r-1} be the cardinality of the set $\mathcal{S}^{(r-1)}$. We have

$$\begin{aligned} \left(\sum_{J\in J_r} |\beta_J|\right)^2 &\le 2\left(\sum_{J\in\mathcal{S}^{(r-1)}} |\beta_J|\right)^2 + 2\left(\sum_{J\in\sigma(\mathcal{S}^{(r-1)})} |\beta_J|\right)^2 \\ &\le 2s_{r-1}\left(\sum_{J\in\mathcal{S}^{(r-1)}} \beta_J^2\right)^2 + 2\tau^2 \le (2s_{r-1}+2)\tau^2, \end{aligned} \tag{32}$$

where we use the inequality $2ab \le a^2 + b^2$ for any $a > 0$ and $b > 0$, and the constrained in the function class.

To obtain an upper bound, we begin with some notation. For a function class $\mathcal{F}$ and for any measure Q, the L^∞ bracketing number $\mathcal{N}_{[]}(\mathcal{F}, L^\infty(Q), \epsilon)$ is defined as the smallest number of pairs $B = \{(l_1, u_1), \ldots, (l_k, u_k)\}$ such that $\|u_j - l_j\|_\infty \le \epsilon$ for $1 \le j \le k$, and such that for every $f \in \mathcal{F}$, there exists $(l, u) \in B$ such that $l \le f \le u$. Define the function class

$$\mathcal{W} = \{g_J g_{J'} \mid g_J \in \mathcal{H}_J, g_{J'} \in \mathcal{H}_{J'}, \|g_J\|_{\mathcal{H}_J} \le 1, \|g_{J'}\|_{\mathcal{H}_{J'}} \le 1, J, J' \in \mathcal{J}_r\}. \tag{33}$$

By Lemma 4, we have

$$\log\mathcal{N}_{[]}(\mathcal{W}, L^\infty(Q), \epsilon) \le C(r\log d + \epsilon^{-r/m}), \tag{34}$$

where $C > 0$ is some constant. By Corollary 19.35 of Van der Vaart (2000) and (34), we obtain

$$\begin{aligned} \mathbb{E}\left(\max_{J\in\mathcal{J}_r} \sup_{g_J\in\mathcal{H}_J, g_{J'}\in\mathcal{H}_{J'}} (\mathbb{E}_n - \mathbb{E})[g_J g_{J'}]\right) &\le \frac{C}{\sqrt{n}}\int_0^C \sqrt{\log\mathcal{N}_{[]}(\mathcal{W}, L^\infty(Q), \epsilon)}\, d\epsilon \\ &\le \frac{C}{\sqrt{n}}\int_0^C \sqrt{r\log d + \epsilon^{-r/m}}\, d\epsilon \\ &\le \frac{C}{\sqrt{n}}\int_0^C \sqrt{r\log d} + \epsilon^{-r/2m}\, d\epsilon \\ &\le C\sqrt{\frac{r\log d}{n}}, \end{aligned} \tag{35}$$

where the third inequality follows from the fact that $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$, the last inequality follows from the assumption that $r < 2m$, and C is a constant that may vary line to line.

By an application of Markov's inequality, we have

$$\max_{J \in \mathcal{J}_r} \sup_{g_J \in \mathcal{H}_J, g_{J'} \in \mathcal{H}_{J'}} (\mathbb{E}_n - \mathbb{E})[g_J g_{J'}] = O_P\left(\sqrt{\frac{r \log d}{n}}\right). \tag{36}$$

Combining the above with (32), we have that for all (β, g),

$$|\widehat{R}(\beta, g) - R(\beta, g)| = O_P\left(\tau^2 \sqrt{\frac{r s_{r-1}^2 \log d}{n}}\right),$$

as desired.

Appendix B. Proof of Proposition 1

The proof of Proposition 1 is similar to that of the proof of Theorem 2 in Ravikumar et al. (2009). The main difference is that we have additional bivariate interaction terms in the true underlying model. Recall that $\mathcal{S}^{(1)}$ is the support of the estimator obtained from solving (14) with $r = 1$. Recall from Section 3.1 that we approximate the main effects f_j by its kth order basis expansion, i.e., $\widetilde{f}_j = \sum_{l=1}^k \theta_j^l \phi_{jl}(x_j) = \boldsymbol{\theta}_j^T \boldsymbol{\phi}_j(x_j)$. Let $\boldsymbol{\theta}_j^*$ be the underlying coefficients corresponding to the kth order basis expansion of the main effects. To simplify the notation, we let $S = \bar{\mathcal{S}}^{(1)}$.

Let $\boldsymbol{\Phi}_S$ be the $n \times k|S|$ matrix with rows $\boldsymbol{\phi}_S(\mathbf{x}_1), \ldots, \boldsymbol{\phi}_S(\mathbf{x}_n)$, where $\boldsymbol{\phi}_S(\cdot)$ is obtained by concatenating $\boldsymbol{\phi}_j(\cdot)$ for all $j \in S$. Similarly, $\boldsymbol{\theta}_S^*$ is obtained by concatenating $\boldsymbol{\theta}_j^*$ for all $j \in S$. Recall that we denote the projection operator $P_{\bar{\mathcal{S}}^{(2)}} f = \sum_{(s,t) \in \bar{\mathcal{S}}^{(2)}} f_{st}$. We now consider a second order Taylor expansion of f_{st} at $(1/2, 1/2)$ for each $(s,t) \in \bar{\mathcal{S}}^{(2)}$:

$$\begin{aligned} f_{st}(x_s, x_t) =& f_{st}(0.5, 0.5) + \sum_{j \in \{s,t\}} \left(\partial_{x_j} f_{st}(0.5, 0.5)(x_j - 0.5) + \frac{1}{2}\partial_{x_j}^2 f_{st}(0.5, 0.5)(x_j - 0.5)^2\right) \\ &+ \partial_{x_s x_t}^2 f_{st}(0.5 + \eta(x_s - 0.5), 0.5 + \eta(x_t - 0.5))(x_s - 0.5)(x_t - 0.5), \end{aligned} \tag{37}$$

where $\eta \in [0, 1]$. Since $\boldsymbol{\Phi}_S$ is the design matrix generated from B-spline polynomials, we can represent the vector $\mathbf{u} := (P_{\bar{\mathcal{S}}^{(2)}} f(\mathbf{x}_1), \ldots, P_{\bar{\mathcal{S}}^{(2)}} f(\mathbf{x}_n))^T$ as $\mathbf{u} = \boldsymbol{\Phi}_S \boldsymbol{\gamma}_S^* + \Delta$, where the ith entry of $\boldsymbol{\Phi}_S \boldsymbol{\gamma}_S^*$ represents the leading terms in (37) and $\boldsymbol{\gamma}_S^*$ is the corresponding basis coefficients vector:

$$[\boldsymbol{\Phi}_S \boldsymbol{\gamma}_S^*]_i = \textstyle\sum_{(s,t) \in \bar{\mathcal{S}}^{(2)}} \left[f_{st}(0.5, 0.5) + \sum_{j \in \{s,t\}} (\partial_{x_j} f_{st}(0.5, 0.5)(x_{ij} - 0.5) + \frac{1}{2}\partial_{x_j}^2 f_{st}(0.5, 0.5)(x_{ij} - 0.5)^2)\right].$$

The remainder Δ corresponds to the last term in (37), i.e., for each $i = 1, \ldots, n$,

$$\Delta_i = \textstyle\sum_{(s,t) \in \bar{\mathcal{S}}^{(2)}} \partial_{x_s x_t}^2 f_{st}(0.5 + \eta(x_{is} - 0.5), 0.5 + \eta(x_{it} - 0.5))(x_{is} - 0.5)(x_{it} - 0.5).$$

Thus, $\|\Delta\|_\infty \le q_n^*$, where $q_n^* := \|\sum_{(s,t) \in \bar{\mathcal{S}}^{(2)}} \partial_{x_s x_t}^2 f_{st}\|_\infty$.

Denote $\boldsymbol{\Sigma}_{SS} = \frac{1}{n}\boldsymbol{\Phi}_S^T \boldsymbol{\Phi}_S$. Let $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$, and let $\mathbf{v} = \mathbf{y} - \boldsymbol{\Phi}_S(\boldsymbol{\theta}_S^* + \boldsymbol{\gamma}_S^*) - \Delta - \boldsymbol{\epsilon}$ denote the error due to finite truncation of the orthogonal basis. Compared to the proof of

Theorem 2 in Ravikumar et al. (2009), we have an additional term Δ. Therefore, following the same proof, we have

$$\|\widehat{\boldsymbol{\theta}}_S - \boldsymbol{\theta}_S^* - \boldsymbol{\gamma}_S^*\|_\infty \le \left\|\boldsymbol{\Sigma}_{SS}^{-1}\left(\frac{1}{n}\boldsymbol{\Phi}_S^T \boldsymbol{\epsilon}\right)\right\|_\infty + \left\|\boldsymbol{\Sigma}_{SS}^{-1}\left(\frac{1}{n}\boldsymbol{\Phi}_S^T(\mathbf{v} + \Delta)\right)\right\|_\infty + \lambda\|\boldsymbol{\Sigma}_{SS}^{-1}\widehat{\mathbf{g}}_S\|_\infty, \tag{38}$$

where $\boldsymbol{\theta}_S$ and $\widehat{\mathbf{g}}_S$ are subvectors of the estimator in (14) for $r = 1$ and the gradient of the penalty in (14) for $r = 1$, respectively. If we can show that

$$\|\widehat{\boldsymbol{\theta}}_S - \boldsymbol{\beta}_S^*\|_\infty < \frac{1}{2}\min_{j \in S}\|\boldsymbol{\theta}_j^* + \boldsymbol{\gamma}_j^*\|_\infty = \frac{1}{2}\min_{j \in S}\|\boldsymbol{\beta}_j^*\|_\infty = \rho_n^*,$$

then $\|\widehat{\boldsymbol{\theta}}_j\|_\infty > 0$ for each $j \in S$, i.e., $\mathcal{S}^{(1)} \supseteq S$. Therefore, it suffices to show that the right hand side of (38) is smaller than ρ_n^*. Compared to the right hand side of Equation (83) in Ravikumar et al. (2009), we only have an additional term $\|\boldsymbol{\Sigma}_{SS}^{-1}(\frac{1}{n}\boldsymbol{\Phi}_S^T \Delta)\|_\infty$ in (38).

Since $\|\Delta\|_\infty \le q_n^*$, we can bound $\|\boldsymbol{\Sigma}_{SS}^{-1}(\frac{1}{n}\boldsymbol{\Phi}_S^T \Delta)\|_\infty$ following the same procedure as $\|\boldsymbol{\Sigma}_{SS}^{-1}(\frac{1}{n}\boldsymbol{\Phi}_S^T \mathbf{v})\|_\infty$. According to Equations (99) and (111) in Ravikumar et al. (2009), if $\sqrt{|\bar{\mathcal{S}}^{(1)}|k}q_n^*/\rho_n^* = o(1)$ and $\sqrt{k}q_n^*/\lambda = o(1)$, terms related to Δ will be dominated by terms related to $\boldsymbol{\epsilon}$ and $\mathbf{v}$, and the remaining proof in Ravikumar et al. (2009) follows through.

Appendix C. Proof of Lemma 1

Recall from Section 4.3 that we define $\mathbf{r}'(\boldsymbol{x})$ to be the element-wise differentiation of the vector $\mathbf{r}(\boldsymbol{x})$. Also, recall from (22) that the modified Fisher divergence is defined as

$$\frac{1}{2}\int_{[0,1]^d} p(\boldsymbol{x})\,\|\mathbf{r}(\boldsymbol{x}) \circ [\nabla \log p(\boldsymbol{x}) - \nabla \log q(\boldsymbol{x})]\|_2^2\, d\boldsymbol{x} = T_1 + T_2 + T_3, \text{ where} \tag{39}$$

$$\begin{aligned} T_1 &= \frac{1}{2}\int_{[0,1]^d} p(\boldsymbol{x})\|\mathbf{r}(\boldsymbol{x}) \circ \nabla \log q(\boldsymbol{x})\|_2^2 d\boldsymbol{x}, \\ T_2 &= -\int_{[0,1]^d} p(\boldsymbol{x})\,(\mathbf{r}(\boldsymbol{x}) \circ \nabla \log p(\boldsymbol{x}))^T\,(\mathbf{r}(\boldsymbol{x}) \circ \nabla \log q(\boldsymbol{x}))\, d\boldsymbol{x}, \\ T_3 &= \frac{1}{2}\int_{[0,1]^d} p(\boldsymbol{x})\|\mathbf{r}(\boldsymbol{x}) \circ \nabla \log p(\boldsymbol{x})\|_2^2 d\boldsymbol{x}. \end{aligned}$$

By some algebraic manipulation, we have

$$\begin{aligned} T_2 &= -\int_{[0,1]^d} p(\boldsymbol{x})\,(\mathbf{r}(\boldsymbol{x}) \circ \nabla \log p(\boldsymbol{x}))^T\,(\mathbf{r}(\boldsymbol{x}) \circ \nabla \log q(\boldsymbol{x}))\, d\boldsymbol{x} \\ &= -\int_{[0,1]^d} (\nabla p(\boldsymbol{x}))^T\,(\mathbf{r}(\boldsymbol{x}) \circ \mathbf{r}(\boldsymbol{x})) \circ \nabla \log q(\boldsymbol{x}) d\boldsymbol{x} \\ &= \sum_{j=1}^d \left[-p(\boldsymbol{x})\nabla_j \log q(\boldsymbol{x}) r_j^2(x_j)\big|_0^1 + \int_{[0,1]^d} p(\boldsymbol{x})\frac{\partial}{\partial x_j}\left\{r_j^2(x_j)\nabla_j \log q(\boldsymbol{x})\right\} d\boldsymbol{x}\right] \\ &= \int_{[0,1]^d} p(\boldsymbol{x})\left[\left(2\mathbf{r}(\boldsymbol{x}) \circ \mathbf{r}'(\boldsymbol{x})\right)^T \nabla \log q(\boldsymbol{x}) + (\mathbf{r}(\boldsymbol{x}) \circ \mathbf{r}(\boldsymbol{x}))^T \nabla^2 \log q(\boldsymbol{x})\right] d\boldsymbol{x} \end{aligned}$$

where the second equality uses the fact that $\nabla \log p(\boldsymbol{x}) = \nabla p(\boldsymbol{x})/p(\boldsymbol{x})$, the third equality holds using integration by parts, and the last equality holds by the assumption that

$$\lim_{x_j \to 0} p(\boldsymbol{x}) \cdot \nabla_j \log q(\boldsymbol{x}) r_j^2(x_j) \to 0 \qquad \text{and} \qquad \lim_{x_j \to 1} p(\boldsymbol{x}) \cdot \nabla_j \log q(\boldsymbol{x}) r_j^2(x_j) \to 0.$$

Since T_3 is independent of $q(\boldsymbol{x})$, we obtain

$$F(p, q) = \int_{[0,1]^d} p(\boldsymbol{x}) S(\boldsymbol{x}, q) d\boldsymbol{x} + C$$

with

$$S(\boldsymbol{x}, q) = 2\left(\mathbf{r}(\boldsymbol{x}) \circ \mathbf{r}'(\boldsymbol{x})\right)^T \nabla \log q(\boldsymbol{x}) + (\mathbf{r}(\boldsymbol{x}) \circ \mathbf{r}(\boldsymbol{x}))^T \nabla^2 \log q(\boldsymbol{x}) + \frac{1}{2}\|\mathbf{r}(\boldsymbol{x}) \circ \nabla \log q(\boldsymbol{x})\|_2^2,$$

and C is a constant that does not depend on $q(\boldsymbol{x})$.

Appendix D. Proof of Theorem 2

Recall that we define the risk function as the Fisher's divergence between two densities

$$R(p, q) = \frac{1}{2} \int_{[0,1]^d} p(\boldsymbol{x}) \left\| \mathbf{r}(\boldsymbol{x}) \circ \nabla \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \right\|_2^2 d\boldsymbol{x}.$$

For notational convenient, let $q^* = \arg\inf_{q \in \mathcal{Q}^{(r)}} R(p, q)$ be the oracle estimator and let $\widehat{p}^{(r)} \propto \exp(-\widehat{f}^{(r)})$. The goal is to show that $R(p, \widehat{p}^{(r)}) - R(p, q^*) = o_P(1)$. By Lemma 1, the difference between the two risk functions can be rewritten as

$$R(p, \widehat{p}^{(r)}) - R(p, q^*) = \mathbb{E}[S(\boldsymbol{X}, \widehat{p}^{(r)})] - \mathbb{E}[S(\boldsymbol{X}, q^*)],$$

since the constant term in Lemma 1 is independent of q. By an argument similar to (31), we have

$$\begin{aligned}
0 &\le \mathbb{E}[S(\boldsymbol{X}, \widehat{p}^{(r)})] - \mathbb{E}[S(\boldsymbol{X}, q^*)] \\
&\le \left| \mathbb{E}[S(\boldsymbol{X}, \widehat{p}^{(r)})] - \frac{1}{n}\sum_{i=1}^n S(\mathbf{x}_i, \widehat{p}^{(r)}) \right| + \left| \frac{1}{n}\sum_{i=1}^n S(\mathbf{x}_i, q^*) - \mathbb{E}[S(\boldsymbol{X}, q^*)] \right| \\
&\le 2 \sup_{q \in \mathcal{Q}^{(r)}} \left| \frac{1}{n}\sum_{i=1}^n S(\mathbf{x}_i, q) - \mathbb{E}[S(\boldsymbol{X}, q)] \right|,
\end{aligned}$$

where the second inequality holds from the fact that $\widehat{p}^{(r)}$ is a minimizer of $n^{-1}\sum_{i=1}^n S(\mathbf{x}_i, q)$. Thus, it suffices to establish the rate of convergence for the above quantity.

Let $\mathcal{J}_r = \delta(\mathcal{S}^{(r-1)}) \cup \{\emptyset\}$. Recall that for any $f \in \mathcal{F}^{(r)}$, we have $f(\boldsymbol{x}) = \sum_{J \in \mathcal{J}_r} \beta_J g_J(\boldsymbol{x}_J)$. Then, $S(\boldsymbol{x}, q)$ can be rewritten as

$$\begin{aligned}
S(\boldsymbol{x}, q) &= 2(\mathbf{r}(\boldsymbol{x}) \circ \mathbf{r}(\boldsymbol{x}))^T \nabla \log q(\boldsymbol{x}) + (\mathbf{r}(\boldsymbol{x}) \circ \mathbf{r}(\boldsymbol{x}))^T \nabla^2 \log q(\boldsymbol{x}) + \frac{1}{2}\|\mathbf{r}(\boldsymbol{x}) \circ \nabla \log q(\boldsymbol{x})\|_2^2 \\
&= -2\sum_{j=1}^d \sum_{J \in \mathcal{J}_r} r_j r_j' \beta_J g_J^{(j)} - \sum_{j=1}^d \sum_{J \in \mathcal{J}_r} r_j^2 \beta_J g_J^{(jj)} + \frac{1}{2}\sum_{j=1}^d \sum_{J \in \mathcal{J}_r} \sum_{J' \in \mathcal{J}_r} r_j^2 \beta_J \beta_{J'} g_J^{(j)} g_{J'}^{(j)},
\end{aligned}$$

where we denote $g_J^{(j)}$ and $g_J^{(jj)}$ as the first and second order derivatives of g_J with respect to X_j, respectively. For notational convenience, we suppress the dependencies of $\boldsymbol{x}$ in $r_j(\cdot)$ and $g_J(\cdot)$, and use the fact that $g_J^{(j)} = 0$ if $j \notin J$.

Thus, for any $q \in \mathcal{Q}^{(r)}$, we have

$$\left| \frac{1}{n}\sum_{i=1}^n S(\mathbf{x}_i, q) - \mathbb{E}[S(\boldsymbol{X}, q)] \right| \le 2I_1 + I_2 + I_3,$$

where the three terms are

$$\begin{aligned}
I_1 &= \max_{J \in \mathcal{J}_r} \sup_{g_J \in \mathcal{H}_J} \left| \sum_{j=1}^d \sum_{J \in \mathcal{J}_r} \beta_J (\mathbb{E}_n - \mathbb{E})[r_j r_j' g_J^{(j)}] \right|, \\
I_2 &= \max_{J \in \mathcal{J}_r} \sup_{g_J \in \mathcal{H}_J} \left| \sum_{j=1}^d \sum_{J \in \mathcal{J}_r} \beta_J (\mathbb{E}_n - \mathbb{E})[r_j^2 g_J^{(jj)}] \right|, \\
I_3 &= \max_{J, J' \in \mathcal{J}_r} \sup_{g_J \in \mathcal{H}_J, g_{J'} \in \mathcal{H}_{J'}} \left| \sum_{j=1}^d \sum_{J \in \mathcal{J}_r} \sum_{J' \in \mathcal{J}_r} \beta_J \beta_{J'} (\mathbb{E}_n - \mathbb{E})[r_j^2 g_J^{(j)} g_{J'}^{(j)}] \right|.
\end{aligned}$$

We now obtain upper bounds for I_1, I_2, and I_3 separately. To this end, we define three function classes

$$\begin{aligned}
\mathcal{W}_1 &= \left\{ r_j r_j' g_J^{(j)} \mid \|g_J\|_{\mathcal{H}_J} \le 1, J \in \mathcal{J}_r, j \in [d] \right\}, \\
\mathcal{W}_2 &= \left\{ r_j^2 g_J^{(jj)} \mid \|g_J\|_{\mathcal{H}_J} \le 1, J \in \mathcal{J}_r, j \in [d] \right\}, \\
\mathcal{W}_3 &= \left\{ r_j^2 g_J^{(j)} g_{J'}^{(j)} \mid \|g_J\|_{\mathcal{H}_J} \le 1, \|g_{J'}\|_{\mathcal{H}_{J'}} \le 1, J, J' \in \mathcal{J}_r, j \in [d] \right\}.
\end{aligned}$$

Upper bound for I_1: We first note that

$$I_1 \le r \left(\sum_{J \in \mathcal{J}_r} |\beta_J| \right) \max_{j \in [d]} \max_{J \in \mathcal{J}_r} \sup_{g_J \in \mathcal{H}_J} \left| (\mathbb{E}_n - \mathbb{E})[r_j r_j' g_J^{(j)}] \right|,$$

since $\sum_{j=1}^d g_J^{(j)}$ can be expressed as the sum of a maximum of r terms for any $J \in \mathcal{J}_r$.

By Lemma 5, the bracketing number for the function class $\mathcal{W}_1$ is

$$\log \mathcal{N}_{[]}(\mathcal{W}_1, L^\infty(Q), \epsilon) \le C(r \log d + \epsilon^{-r/(m-1)}).$$

Following the arguments in (35), we have

$$\mathbb{E}\left(\max_{j \in [d]} \max_{J \in \mathcal{J}_r} \sup_{g_J \in \mathcal{H}_J} (\mathbb{E}_n - \mathbb{E})[r_j r_j' g_J^{(j)}] \right) \le C\sqrt{\frac{r \log d}{n}}.$$

Similar to the proof of (36), we obtain

$$\max_{j \in [d]} \max_{J \in \mathcal{J}_r} \sup_{g_J \in \mathcal{H}_J} (\mathbb{E}_n - \mathbb{E})[r_j r_j' g_J^{(j)}] = O_P\left(\sqrt{\frac{r \log d}{n}} \right).$$

Similar to (32), we have

$$\sum_{J\in\mathcal{J}_r} |\beta_J| \le \sqrt{2s_{r-1}+2}\tau,$$

which implies that $I_1 = O_P(\tau\sqrt{(r^3 s_{r-1}\log d)/n})$.

Upper bound for I_2: We have

$$I_2 \le r\left(\sum_{J\in\mathcal{J}_r}|\beta_J|\right)\max_{j\in[d]}\max_{J\in\mathcal{J}_r}\sup_{g_J\in\mathcal{H}_J}\left|(\mathbb{E}_n-\mathbb{E})[r_j^2 g_J^{(jj)}]\right|.$$

By Lemma 5, the bracketing number for the function class $\mathcal{W}_2$ is

$$\log\mathcal{N}_{[]}(\mathcal{W}_2, L^\infty(Q),\epsilon) \le C(r\log d + \epsilon^{-r/(m-2)}).$$

Following the arguments for the upper bound on I_1, we have

$$\max_{j\in[d]}\max_{J\in\mathcal{J}_r}\sup_{g_J\in\mathcal{H}_J}(\mathbb{E}_n-\mathbb{E})[r_j^2 g_J^{(jj)}] = O_P\left(\sqrt{\frac{r\log d}{n}}\right)$$

and that $I_2 = O_P(\tau\sqrt{(r^3 s_{r-1}\log d)/n})$.

Upper bound for I_3: To obtain an upper bound for I_3, we have

$$I_3 \le r\left(\sum_{J\in\mathcal{J}_r}|\beta_J|\right)^2\max_{j\in[d]}\max_{J\in\mathcal{J}_r}\sup_{g_J\in\mathcal{H}_J}\left|(\mathbb{E}_n-\mathbb{E})[r_j^2 g_J^{(jj)}]\right|.$$

Similar to (32), we have

$$\left(\sum_{J\in\mathcal{J}_r}|\beta_J|\right)^2 \le (2s_{r-1}+2)\tau^2.$$

By an application of Lemma 5, the bracketing number for the function class $\mathcal{W}_3$ is

$$\log\mathcal{N}_{[]}(\mathcal{W}_3, L^\infty(Q),\epsilon) \le C(r\log d + \epsilon^{-r/(m-1)}).$$

Thus, following the same arguments for the upper bound on I_1, we have

$$\max_{j\in[d]}\max_{J\in\mathcal{J}_r}\sup_{g_J\in\mathcal{H}_J}\left|(\mathbb{E}_n-\mathbb{E})[r_j^2 g_J^{(jj)}]\right| = O_P\left(\sqrt{\frac{r\log d}{n}}\right),$$

which implies that $I_3 = O_P(\tau^2\sqrt{(r^3 s_{r-1}^2\log d/n)})$.

Combining the upper bounds on I_1, I_2, and I_3, we have

$$R(p,\widehat{p}^{(r)}) - R(p,q^*) = O_P\left(\tau^2\sqrt{\frac{r^3 s_{r-1}^2\log d}{n}}\right),$$

as desired.

Appendix E. Auxiliary Results on Bracketing Number

In this section, we provide some technical results on bracketing number for some function classes. Lemma 2 provides the bracketing number of the Sobolev space. Lemma 3 provides an upper bound on the bracketing number for function classes generated from the product and addition of two function classes. With applications of Lemmas 2 and 3, we obtain upper bounds on bracketing number for function classes arise in the proof of Theorems 1 and 2.

Lemma 2 (Corollary 1 in Nickl and Pötscher, 2007)**.** *Let Ω be a bounded, convex subset of $\mathbb{R}^r$ with non-empty interior and $Q(\Omega)<\infty$ for some measure Q. Denote $\mathbb{B}_m(\Omega) = \{f\in\mathcal{H}^m(\Omega) \mid \|f\|_{\mathcal{H}^m}\le 1\}$. There exists a constant K such that*

$$\log\mathcal{N}_{[]}(\mathbb{B}_m(\Omega), L_\infty(Q),\epsilon) \le K\cdot\epsilon^{-r/m}.$$

Lemma 3 (Lemma 9.24 in Kosorok, 2007)**.** *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two function classes. Define $\|\mathcal{F}_\ell\|_\infty = \sup_{f\in\mathcal{F}_\ell}\|f\|_\infty$ for $\ell=1,2$ and $U = \|\mathcal{F}_1\|_\infty\vee\|\mathcal{F}_2\|_\infty$. For the function classes $\mathcal{F}_+ = \{f_1+f_2 \mid f_1\in\mathcal{F}_1,\ f_2\in\mathcal{F}_2\}$ and $\mathcal{F}_\times = \{f_1 f_2 \mid f_1\in\mathcal{F}_1,\ f_2\in\mathcal{F}_2\}$, we have for any $\epsilon\in(0,1)$,*

$$\mathcal{N}_{[]}(\mathcal{F}_+, L_\infty(Q),\epsilon) \le \mathcal{N}_{[]}(\mathcal{F}_1, L_\infty(Q),\epsilon)\cdot\mathcal{N}_{[]}(\mathcal{F}_2, L_\infty(Q),\epsilon),$$
$$\mathcal{N}_{[]}(\mathcal{F}_\times, L_\infty(Q),\epsilon) \le \mathcal{N}_{[]}(\mathcal{F}_1, L_\infty(Q),\epsilon/U)\cdot\mathcal{N}_{[]}(\mathcal{F}_2, L_\infty(Q),\epsilon/U).$$

Together with Lemmas 2 and 3, we can now obtain the bracketing number

$$\mathcal{W} = \left\{g_J g_{J'} \mid g_J\in\mathcal{H}_J, g_{J'}\in\mathcal{H}_{J'}, \|g_J\|_{\mathcal{H}_J}\le 1, \|g_{J'}\|_{\mathcal{H}_{J'}}\le 1, J,J'\in\mathcal{J}_r\right\}$$

as defined in (33). The result is summarized in the following lemma.

Lemma 4. *For any measure Q and $\epsilon>0$, there exists a constant $C>0$ such that*

$$\log\mathcal{N}_{[]}(\mathcal{W}, L^\infty(Q),\epsilon) \le C(r\log d + \epsilon^{-r/m}).$$

Proof. Let J be an index set with $|J|\le r$. For a given J, By Lemma 2, the covering number for the function class $\mathcal{W}^{(J)} = \{g_J \mid g_J\in\mathcal{H}_J, \|g_J\|_{\mathcal{H}_J}\le 1\}$ is

$$\log\mathcal{N}_{[]}(\mathcal{W}^{(J)}, L_\infty(Q),\epsilon) \le K\cdot\epsilon^{-r/m}.$$

By an application of Lemma 3, there exists a constant $C>0$ such that

$$\begin{aligned}\log\mathcal{N}_{[]}(\mathcal{W}, L_\infty(Q),\epsilon) &\le \log\left(\binom{d}{r}\cdot[\mathcal{N}_{[]}(\mathcal{W}^{(J)}, L_\infty(Q),\epsilon)]^2\right)\\ &\le r\log d + 2K\epsilon^{-r/m} \le C(r\log d + \epsilon^{-r/m}).\end{aligned}$$

The term $r\log d$ is an upper bound of the log cardinality of $\mathcal{J}_r$. □

Next, we obtain the bracketing number for the following function classes:

$$\mathcal{W}_1 = \left\{r_j r_j' g_J^{(j)} \mid \|g_J\|_{\mathcal{H}_J}\le 1, J\in\mathcal{J}_r, j\in[d]\right\},$$
$$\mathcal{W}_2 = \left\{r_j^2 g_J^{(jj)} \mid \|g_J\|_{\mathcal{H}_J}\le 1, J\in\mathcal{J}_r, j\in[d]\right\},$$
$$\mathcal{W}_3 = \left\{r_j^2 g_J^{(j)} g_{J'}^{(j)} \mid \|g_J\|_{\mathcal{H}_J}\le 1, \|g_{J'}\|_{\mathcal{H}_{J'}}\le 1, J,J'\in\mathcal{J}_r, j\in[d]\right\}.$$

Lemma 5. *For any measure Q and $\epsilon > 0$, there exists a constant $C > 0$ such that*

$$\log \mathcal{N}_{[]}(\mathcal{W}_1, L_\infty(Q), \epsilon) \le C \cdot (r \log d + \epsilon^{-r/(m-1)}),$$
$$\log \mathcal{N}_{[]}(\mathcal{W}_2, L_\infty(Q), \epsilon) \le C \cdot (r \log d + \epsilon^{-r/(m-2)}),$$
$$\log \mathcal{N}_{[]}(\mathcal{W}_3, L_\infty(Q), \epsilon) \le C \cdot (r \log d + \epsilon^{-r/(m-1)}).$$

Proof. For a given index set J with $|J| \le r$, we have $\|g_J\|_{\mathcal{H}_J} \le 1$ by the definition of the function classes $\mathcal{W}_1$, $\mathcal{W}_2$, and $\mathcal{W}_3$. This implies that $\|g_J^{(j)}\|_{\mathcal{H}_J} \le 1$, and that $g_J^{(j)} \in \mathbb{B}_{m-1}$. By Lemma 2, the log bracketing number for the function class $\{g_J^{(j)} \mid g_J^{(j)} \in \mathbb{B}_{m-1}\}$ is upper bounded by $K\epsilon^{-r/(m-1)}$. Since r_j and r_j' are both fixed and bounded functions on $[0,1]$, by Lemma 3, we have

$$\log \mathcal{N}_{[]}(\mathcal{W}_1, L_\infty(Q), \epsilon) \le C \cdot (r \log d + \epsilon^{-r/(m-1)}),$$

where the term $r \log d$ arises from the upper bound on the log cardinality of $\mathcal{J}_r$.

Similarly, since $\|g_J\|_{\mathcal{H}_J} \le 1$, we have $\|g_J^{(jj)}\|_{\mathcal{H}_J} \le 1$, and that $g_J^{(jj)} \in \mathbb{B}_{m-2}$. By Lemma 2, for a given index set J with $|J| \le r$, the log bracketing number for the function class $\{g_J^{(j)} \mid g_J^{(j)} \in \mathbb{B}_{m-2}\}$ is upper bounded by $K\epsilon^{-r/(m-2)}$. Following the same argument, we have

$$\log \mathcal{N}_{[]}(\mathcal{W}_2, L_\infty(Q), \epsilon) \le C \cdot (r \log d + \epsilon^{-r/(m-2)}).$$

For two given index set J and J' with $|J| \le r$ and $|J'| \le r$, the function class $\{g_J^{(j)} g_{J'}^{(j)} \mid g_J^{(j)}, g_{J'}^{(j)} \in \mathbb{B}_{m-1}\}$ is upper bounded by $2K\epsilon^{-r/(m-1)}$ by an application of Lemmas 2 and 3. Similarly, since r_j is fixed and bounded function on $[0,1]$, by Lemma 3, there exists a constant $C > 0$ such that

$$\log \mathcal{N}_{[]}(\mathcal{W}_3, L_\infty(Q), \epsilon) \le C \cdot (r \log d + \epsilon^{-r/(m-1)}).$$

□

Appendix F. Results on Model Selection for Numerical Studies in Section 5.1

As noted in Sections 3.2 and 3.3, it is vital for `LASER` to include all of the active main effects and the lower-order interaction effects to achieve persistency. In this section, we assess the model selection performance of `LASER` by reporting the true and false positive rates. We consider the simulation studies in Sections 5.1.1 and 5.1.2. The results, averaged over 100 replications, are summarized in Tables 4—6. Note that the results for `FAMILY` are omitted since `FAMILY` is outperformed by both `BT` and `RAMP` in terms of sum of squared error.

In the low-dimensional setting, we see from Table 4 that `LASER` has true positive rate of one for all scenarios. This illustrates that `LASER` is persistent for Scenarios A1—A3 since it is able to identify all of the active main effects before estimating the second-order interaction terms. We notice that `BT` tends to have high false positive rate on all scenarios, and that `RAMP` has a low true positive rate for Scenarios A2 and A3 when the active variables are non-linear. `LASER` has high TPRs for all scenarios for the interaction effects. Moreover, it

achieves a TPR of approximately one for Scenario A3 when other methods fail. `SPAM` has NAs on the interaction effects since it estimates only the main effects. Similar results are observed for the high-dimensional setting in Tables 5 and 6. The results for `VANISH` are omitted for the high-dimensional setting due to computational reasons.

Table 4: The true and false positive rates (TPR and FPR) for the main effects and the second-order interaction effects for three different scenarios in Section 5.1.1, averaged over 100 data sets. The results are for models trained with n training samples and $d = 30$ variables. Most of the standard errors are approximately zero and are omitted.

		$n = 200$				$n = 400$			
		Main Effects		Interaction Effects		Main Effects		Interaction Effects	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Scenario A1	`BT`	1	0.861	0.977	0.013	1	0.922	1	0.017
	`RAMP`	1	0.004	0.663	0.001	1	0.005	0.690	0.001
	`SPAM`	1	0.077	NA	NA	1	0.039	NA	NA
	`VANISH`	1	0.001	0.537	0	1	0.001	0.759	0
	`LASER`	1	0.077	0.940	0.008	1	0.039	0.990	0.006
Scenario A2	`BT`	0.903	0.546	0.497	0.015	0.920	0.620	0.480	0.019
	`RAMP`	0.8	0.043	0.227	0.002	0.830	0.012	0.493	0.005
	`SPAM`	1	0.039	NA	NA	1	0.035	NA	NA
	`VANISH`	1	0.001	1	0.001	1	0.001	1	0
	`LASER`	1	0.039	1	0.016	1	0.035	1	0.013
Scenario A3	`BT`	0.677	0.304	0.120	0.009	0.793	0.460	0.145	0.013
	`RAMP`	0.427	0.058	0.005	0.003	0.613	0.087	0.155	0.005
	`SPAM`	1	0.068	NA	NA	1	0.039	NA	NA
	`VANISH`	0.967	0.001	0.420	0.001	1	0.001	0.365	0.001
	`LASER`	1	0.068	0.985	0.008	1	0.039	1	0.007

References

David Benkeser and Mark van der Laan. The highly adaptive lasso estimator. In *Proceedings of the International Conference on Data Science and Advanced Analytics*, pages 689–696, 2016.

Jacob Bien, Jonathan Taylor, and Robert Tibshirani. A lasso for hierarchical interactions. *The Annals of Statistics*, 41(3):1113–1141, 2013.

Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494): 594–607, 2011.

Shizhe Chen, Daniela M Witten, and Ali Shojaie. Selection and estimation for mixed graphical models. *Biometrika*, 102(1):47–64, 2014.

Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

Table 5: The true and false positive rates (TPR and FPR) for the main effects and the second-order interaction effects for three different scenarios in Section 5.1.2, averaged over 100 data sets. The results are for models trained with n training samples and $d = 200$. Most of the standard errors are approximately zero and are omitted.

		$n = 350$				$n = 700$			
		Main Effects		Interaction Effects		Main Effects		Interaction Effects	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Scenario B1	BT	1	0.477	0.997	0.001	1	0.502	0.973	0.001
	RAMP	1	0.001	1	0	1	0.001	1	0
	SPAM	1	0.003	NA	NA	1	0.003	NA	NA
	LASER	1	0.003	1	0.001	1	0.003	1	0.001
Scenario B2	BT	0.997	0.457	0.906	0.001	1	0.453	0.963	0.001
	RAMP	1	0.018	0.667	0.001	1	0.001	0.670	0.001
	SPAM	0.997	0.002	NA	NA	1	0.001	NA	NA
	LASER	0.997	0.002	0.993	0.001	1	0.001	1	0.001
Scenario B3	BT	0.993	0.565	0.870	0.001	1	0.507	0.987	0.001
	RAMP	1	0.013	0.667	0.001	1	0.001	0.670	0.001
	SPAM	1	0.003	NA	NA	1	0.003	NA	NA
	LASER	1	0.003	1	0.001	1	0.003	1	0.001

Table 6: The true and false positive rates (TPR and FPR) for the main effects and the second-order interaction effects for three different scenarios in Section 5.1.2, averaged over 100 data sets. The results are for models trained with n training samples and $d = 400$. Most of the standard errors are approximately zero and are omitted.

		$n = 350$				$n = 700$			
		Main Effects		Interaction Effects		Main Effects		Interaction Effects	
		TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Scenario B1	BT	1	0.208	1	0	1	0.435	0.99	0.001
	RAMP	1	0	1	0	1	0.001	1	0
	SPAM	1	0.002	NA	NA	1	0.002	NA	NA
	LASER	1	0.002	1	0.001	1	0.002	1	0.001
Scenario B2	BT	1	0.143	0.880	0	1	0.335	0.953	0.001
	RAMP	1	0.007	0.673	0.001	1	0.001	0.670	0.001
	SPAM	1	0.003	NA	NA	1	0.001	NA	NA
	LASER	1	0.003	1	0.001	1	0.001	1	0.001
Scenario B3	BT	1	0.200	0.883	0.001	1	0.677	0.907	0.001
	RAMP	1	0.008	0.634	0.001	1	0.001	0.670	0.001
	SPAM	1	0.002	NA	NA	1	0.001	NA	NA
	LASER	1	0.002	1	0.001	1	0.001	1	0.001

Yingying Fan, Gareth M James, and Peter Radchenko. Functional additive regression. *The Annals of Statistics*, 43(5):2296–2325, 2015.

Peter GM Forbes and Steffen Lauritzen. Linear estimating equations for exponential families with application to Gaussian linear concentration models. *Linear Algebra and its Applications*, 473:261–283, 2015.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Eitan Greenshtein and Ya'Acov Ritov. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10(6):971–988, 2004.

Chong Gu. *Smoothing Spline ANOVA Models*, volume 297. Springer Science & Business Media, 2013.

Chong Gu and Jingyuan Wang. Penalized likelihood density estimation: Direct cross-validation and scalable approximation. *Statistica Sinica*, 13(3):811–826, 2003.

Ning Hao and Hao Helen Zhang. Interaction screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, 109(507):1285–1301, 2014.

Ning Hao, Yang Feng, and Hao Helen Zhang. Model selection for high-dimensional quadratic regression via regularization. *Journal of the American Statistical Association*, 113(522): 615–625, 2018.

Asad Haris, Daniela Witten, and Noah Simon. Convex modeling of interactions with strong heredity. *Journal of Computational and Graphical Statistics*, 25(4):981–1004, 2016.

Trevor J Hastie and Robert J Tibshirani. *Generalized Additive Models*, volume 43. CRC Press, 1990.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–708, 2005.

Aapo Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5):2499–2512, 2007.

Eric Janofsky. Exponential series approaches for nonparametric graphical models. *arXiv preprint arXiv:1506.03537*, 2015.

Yongho Jeon and Yi Lin. An effective method for high-dimensional log-density ANOVA estimation, with application to nonparametric graphical model building. *Statistica Sinica*, 16(2):353–374, 2006.

Michael R Kosorok. *Introduction to Empirical Processes and Semiparametric Inference*. Springer Science & Business Media, 2007.

Tom Leonard. Density estimation, stochastic processes and prior information. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(2):113–146, 1978.

Lina Lin, Mathias Drton, and Ali Shojaie. High-dimensional inference of graphical models using regularized score matching. *Electronic Journal of Statistics*, 10(1):806–854, 2016.

Yi Lin. Tensor product space ANOVA models. *The Annals of Statistics*, 28(3):734–755, 2000.

Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

Han Liu, John Lafferty, and Larry Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295–2328, 2009.

Han Liu, Fang Han, Ming Yuan, John Lafferty, and Larry Wasserman. High-dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

Weidong Liu and Xi Luo. Fast and adaptive sparse precision matrix estimation in high dimensions. *Journal of Multivariate Analysis*, 135:153–162, 2015.

Yin Lou, Jacob Bien, Rich Caruana, and Johannes Gehrke. Sparse partially linear additive models. *Journal of Computational and Graphical Statistics*, 25(4):1126–1140, 2016.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34:1436–1462, 2006.

Richard Nickl and Benedikt M Pötscher. Bracketing metric entropy rates and empirical central limit theorems for function classes of Besov-and Sobolev-type. *Journal of Theoretical Probability*, 20(2):177–199, 2007.

Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

Peter Radchenko and Gareth M James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105 (492):1541–1553, 2010.

Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Adam J Rothman, Peter J Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

Rajen D Shah. Modelling interactions in high-dimensional data with backtracking. *The Journal of Machine Learning Research*, 17(207):1–31, 2016.

Bernard W Silverman. On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics*, 10(3):795–810, 1982.

Charles J Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13(2):689–705, 1985.

Tingni Sun and Cun-Hui Zhang. Sparse matrix inversion with scaled lasso. *The Journal of Machine Learning Research*, 14(1):3385–3418, 2013.

Kean Ming Tan, Palma London, Karthik Mohan, Su-In Lee, Maryam Fazel, and Daniela Witten. Learning graphical models with hubs. *The Journal of Machine Learning Research*, 15(1):3297–3331, 2014.

Kean Ming Tan, Daniela Witten, and Ali Shojaie. The cluster graphical lasso for improved estimation of Gaussian graphical models. *Computational Statistics & Data Analysis*, 85: 23–36, 2015.

Kean Ming Tan, Yang Ning, Daniela M Witten, and Han Liu. Replicates in high dimensions, with applications to latent variable graphical models. *Biometrika*, 103(4):761–777, 2016.

Ryan J Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

Aad W Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge university press, 2000.

Yu-Xiang Wang, James Sharpnack, Alexander J Smola, and Ryan J Tibshirani. Trend filtering on graphs. *The Journal of Machine Learning Research*, 17(1):3651–3691, 2016.

Eunho Yang, Pradeep K Ravikumar, Genevera I Allen, and Zhandong Liu. On Poisson graphical models. In *Advances in Neural Information Processing Systems*, pages 1718–1726, 2013.

Eunho Yang, Pradeep Ravikumar, Genevera I Allen, and Zhandong Liu. Graphical models via univariate exponential family distributions. *The Journal of Machine Learning Research*, 16(1):3813–3847, 2015.

Zhuoran Yang, Yang Ning, and Han Liu. On semiparametric exponential family graphical models. *Journal of Machine Learning Research*, 19(57):1–59, 2018.

Paul Yau, Robert Kohn, and Sally Wood. Bayesian variable selection and model averaging in high-dimensional multinomial nonparametric regression. *Journal of Computational and Graphical Statistics*, 12(1), 2012.

Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Hao Helen Zhang, Grace Wahba, Yi Lin, Meta Voelker, Michael Ferris, Ronald Klein, and Barbara Klein. Variable selection and model building via likelihood basis pursuit. *Journal of the American Statistical Association*, 99(467):659–672, 2004.

Journal of Machine Learning Research 20 (2019) 1-49 Submitted 11/17; Revised 6/19; Published 9/19

Ivanov-Regularised Least-Squares Estimators over Large RKHSs and Their Interpolation Spaces

Stephen Page S.PAGE@LANCASTER.AC.UK
STOR-i
Lancaster University
Lancaster, LA1 4YF, United Kingdom

Steffen Grünewälder S.GRUNEWALDER@LANCASTER.AC.UK
Department of Mathematics and Statistics
Lancaster University
Lancaster, LA1 4YF, United Kingdom

Editor: Ingo Steinwart

Abstract

We study kernel least-squares estimation under a norm constraint. This form of regularisation is known as Ivanov regularisation and it provides better control of the norm of the estimator than the well-established Tikhonov regularisation. Ivanov regularisation can be studied under minimal assumptions. In particular, we assume only that the RKHS is separable with a bounded and measurable kernel. We provide rates of convergence for the expected squared L^2 error of our estimator under the weak assumption that the variance of the response variables is bounded and the unknown regression function lies in an interpolation space between L^2 and the RKHS. We then obtain faster rates of convergence when the regression function is bounded by clipping the estimator. In fact, we attain the optimal rate of convergence. Furthermore, we provide a high-probability bound under the stronger assumption that the response variables have subgaussian errors and that the regression function lies in an interpolation space between L^∞ and the RKHS. Finally, we derive adaptive results for the settings in which the regression function is bounded.

Keywords: Interpolation Space, Ivanov Regularisation, Regression, RKHS, Training and Validation

1. Introduction

One of the key problems to overcome in nonparametric regression is overfitting, due to estimators coming from large hypothesis classes. To avoid this phenomenon, it is common to ensure that both the empirical risk and some regularisation function are small when defining an estimator. There are three natural ways to achieve this goal. We can minimise the empirical risk subject to a constraint on the regularisation function, minimise the regularisation function subject to a constraint on the empirical risk or minimise a linear combination of the two. These techniques are known as Ivanov regularisation, Morozov regularisation and Tikhonov regularisation respectively (Oneto, Ridella, and Anguita, 2016). Ivanov and

©2019 Stephen Page and Steffen Grünewälder.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-672.html.

Morozov regularisation can be viewed as dual problems, while Tikhonov regularisation can be viewed as the Lagrangian relaxation of either.

Tikhonov regularisation has gained popularity as it provides a closed-form estimator in many situations. In particular, Tikhonov regularisation in which the estimator is selected from a reproducing-kernel Hilbert space (RKHS) has been extensively studied (Smale and Zhou, 2007; Caponnetto and de Vito, 2007; Steinwart and Christmann, 2008; Mendelson and Neeman, 2010; Steinwart, Hush, and Scovel, 2009). Although Tikhonov regularisation produces an estimator in closed form, it is Ivanov regularisation which provides the greatest control over the hypothesis class, and hence over the estimator it produces. For example, if the regularisation function is the norm of the RKHS, then the bound on this function forces the estimator to lie in a ball of predefined radius inside the RKHS. An RKHS norm measures the smoothness of a function, so the norm constraint bounds the smoothness of the estimator. By contrast, Tikhonov regularisation provides no direct control over the smoothness of the estimator.

The control we have over the Ivanov-regularised estimator is useful in many settings. The most obvious use of Ivanov regularisation is when the regression function lies in a ball of known radius inside the RKHS. In this case, Ivanov regularisation can be used to constrain the estimator to lie in the same ball. Suppose, for example, that we are interested in estimating the trajectory of a particle from noisy observations over time. Assume that the velocity or acceleration of the particle is constrained by certain physical conditions. Constraints of this nature can be imposed by bounding the norm of the trajectory in a Sobolev space. Certain Sobolev spaces are RKHSs, so it is possible to use Ivanov regularisation to enforce physical conditions on an estimator of the trajectory which match those of the trajectory itself. Ivanov regularisation can also be used within larger inference methods. It is compatible with validation, allowing us to control an estimator selected from an uncountable collection. This is because the Ivanov-regularised estimator is continuous in the size of the ball containing it (see Lemma 32), so the estimators parametrised by an interval of ball sizes can be controlled simultaneously using chaining.

In addition to the other useful properties of the Ivanov-regularised estimator, Ivanov regularisation can be performed almost as quickly as Tikhonov regularisation. The Ivanov-regularised estimator is a support vector machine (SVM) with regularisation parameter selected to match the norm constraint (see Lemma 3). This parameter can be selected to within a tolerance ε using interval bisection with order $\log(1/\varepsilon)$ iterations. In general, Ivanov regularisation requires the calculation of order $\log(1/\varepsilon)$ SVMs.

In this paper, we study the behaviour of the Ivanov-regularised least-squares estimator with regularisation function equal to the norm of the RKHS. We derive a number of novel results concerning the rate of convergence of the estimator in various settings and under various assumptions. Our analysis is performed by controlling empirical processes over balls in the RKHS. By contrast, the analysis of Tikhonov-regularised estimators usually relies on the spectral decomposition of the kernel operator T on $L^2(P)$. Here, P is the covariate distribution.

We first prove an expectation bound on the squared $L^2(P)$ error of our estimator of order $n^{-\beta/2}$, under the weak assumption that the response variables have bounded variance. Here, n is the number of data points, and β parametrises the interpolation space between $L^2(P)$ and H containing the regression function. As far as we are aware, the analysis of an estimator in this setting has not previously been considered. The definition of an interpolation space is given in Section 2. The expected squared $L^2(P)$ error can be viewed as the expected squared error of our estimator at a new independent covariate, with the same distribution P. If we also assume that the regression function is bounded, then it makes sense to clip our estimator so that it takes values in the same interval as the regression function. This further assumption allows us to achieve an expectation bound on the squared $L^2(P)$ error of the clipped estimator of order $n^{-\beta/(1+\beta)}$.

We then move away from the average behaviour of the error towards its behaviour in the worst case. We obtain high-probability bounds of the same order, under the stronger assumption that the response variables have subgaussian errors and the interpolation space is between L^∞ and H. The second assumption is quite natural as we already assume that the regression function is bounded, and H can be continuously embedded in L^∞ since it has a bounded kernel k. Note that this assumption means that the set of possible regression functions is independent of the covariate distribution.

When the regression function is bounded, we also analyse an adaptive version of our estimator, which does not require us to know which interpolation space contains the regression function. This adaptive estimator obtains bounds of the same order as the non-adaptive one. The adaptive estimator is created using training and validation. The bounds do not contain higher-order terms because we treat the validation step as a form of Ivanov regularisation, as opposed to, for example, using a union bound.

Our expectation bound of order $n^{-\beta/(1+\beta)}$, when the regression function is bounded, improves on the high-probability bound of Smale and Zhou (2007) of order $n^{-\beta/2}$. Their bound is attained under the stronger assumption that the regression function lies in the image of a power of the kernel operator, instead of an interpolation space (see Steinwart and Scovel, 2012). The authors also assume that the response variables are bounded. Furthermore, for a fixed $\beta \in (0,1)$, Steinwart et al. (2009) show that there is an instance of our problem with a bounded regression function such that the following holds. For all estimators $\hat{f}$ of g, for some $\varepsilon > 0$, we have

$$\|\hat{f} - g\|^2_{L^2(P)} \geq C_{\alpha,\varepsilon} n^{-\alpha}$$

with probability at least ε for all $n \geq 1$, for some constant $C_{\alpha,\varepsilon} > 0$, for all $\alpha > \beta/(1+\beta)$. Hence, for all estimators $\hat{f}$ of g, we have

$$\mathbb{E}\left(\|\hat{f} - g\|^2_{L^2(P)}\right) \geq C_{\alpha,\varepsilon}\varepsilon n^{-\alpha}$$

for all $n \geq 1$, for all $\alpha > \beta/(1+\beta)$. In this sense, our expectation bound in this setting is optimal because it attains the order $n^{-\beta/(1+\beta)}$, the smallest possible power of n. Our expectation bound on the adaptive version of our estimator is also optimal, because the bound is of the same order as in the easier non-adaptive setting.

The high-probability bound of Steinwart et al. (2009) is optimal in a similar sense, although the authors achieve faster rates by assuming a fixed rate of decay of the eigenvalues of the kernel operator T, as discussed in Section 3. Since there is an additional parameter for the decay of the eigenvalues, the collection of problem instances for a fixed set of parameters is smaller in their paper. This means that our optimal rates are the slowest of the optimal rates in Steinwart et al. (2009).

Note that all of the bounds in this paper are on the squared $L^2(P)$ error of our estimators. We do not consider errors based on other norms, such as the interpolation space norms, as it is unclear how the techniques used in this paper could be adapted to such errors. It seems likely that a completely different approach would be necessary to bound these other errors.

2. RKHSs and Their Interpolation Spaces

A Hilbert space H of real-valued functions on S is an RKHS if the evaluation functional $L_x : H \to \mathbb{R}$, $L_x h = h(x)$, is bounded for all $x \in S$. In this case, $L_x \in H^*$ the dual of H and the Riesz representation theorem tells us that there is some $k_x \in H$ such that $h(x) = \langle h, k_x \rangle_H$ for all $h \in H$. The kernel is then given by $k(x_1, x_2) = \langle k_{x_1}, k_{x_2} \rangle_H$ for $x_1, x_2 \in S$, and is symmetric and positive-definite.

Now suppose that $(S, \mathcal{S})$ is a measurable space on which P is a probability measure. We can define a range of interpolation spaces between $L^2(P)$ and H (Bergh and Löfström, 1976). Let $(Z, \|\cdot\|_Z)$ be a Banach space and $(V, \|\cdot\|_V)$ be a dense subspace of Z. The K-functional of (Z, V) is

$$K(z,t) = \inf_{v \in V}(\|z - v\|_Z + t\|v\|_V)$$

for $z \in Z$ and $t > 0$. For $\beta \in (0,1)$ and $1 \leq q < \infty$, we define

$$\|z\|_{\beta,q} = \left(\int_0^\infty (t^{-\beta}K(z,t))^q t^{-1} dt\right)^{1/q} \text{ and } \|z\|_{\beta,\infty} = \sup_{t>0}(t^{-\beta}K(z,t))$$

for $z \in Z$. The interpolation space $[Z, V]_{\beta,q}$ is defined to be the set of $z \in Z$ such that $\|z\|_{\beta,q} < \infty$. Smaller values of β give larger spaces. The space $[Z, V]_{\beta,q}$ is not much larger than V when β is close to 1, but we obtain spaces which get closer to Z as β decreases. The following result is essentially Theorem 3.1 of Smale and Zhou (2003). The authors only consider the case in which $\|v\|_Z \leq \|v\|_V$ for all $v \in V$, however the result holds by the same proof even without this condition.

Lemma 1 *Let $(Z, \|\cdot\|_Z)$ be a Banach space, $(V, \|\cdot\|_V)$ be a dense subspace of Z and $z \in [Z, V]_{\beta,\infty}$. We have*

$$\inf\{\|v - z\|_Z : v \in V, \|v\|_V \leq r\} \leq \frac{\|z\|^{1/(1-\beta)}_{\beta,\infty}}{r^{\beta/(1-\beta)}}.$$

When H is dense in $L^2(P)$, we can define the interpolation spaces $[L^2(P), H]_{\beta,q}$, where $L^2(P)$ is the space of measurable functions f on $(S, \mathcal{S})$ such that f^2 is integrable with

respect to P. We work with $q = \infty$, which gives the largest space of functions for a fixed $\beta \in (0,1)$. We can then use the approximation result in Lemma 1. When H is dense in L^∞, we also require $[L^\infty, H]_{\beta,q}$, where L^∞ is the space of bounded measurable functions on $(S, \mathcal{S})$.

3. Literature Review

Early research on RKHS regression does not make assumptions on the rate of decay of the eigenvalues of the kernel operator. For example, Smale and Zhou (2007) assume that the response variables are bounded and the regression function is of the form $g = T^{\beta/2} f$ for $\beta \in (0,1]$ and $f \in L^2(P)$. Here, $T : L^2(P) \to L^2(P)$ is the kernel operator and P is the covariate distribution. The authors achieve a squared $L^2(P)$ error of order $n^{-\beta/2}$ with high probability by using SVMs.

Initial research which does make assumptions on the rate of decay of the eigenvalues of the kernel operator, such as that of Caponnetto and de Vito (2007), assumes that the regression function is at least as smooth as an element of H. However, their paper still allows for regression functions of varying smoothness by letting $g \in T^{(\beta-1)/2}(H)$ for $\beta \in [1,2]$. By assuming that the ith eigenvalue of T is of order $i^{-1/p}$ for $p \in (0,1]$, the authors achieve a squared $L^2(P)$ error of order $n^{-\beta/(\beta+p)}$ with high probability by using SVMs. This squared $L^2(P)$ error is shown to be of optimal order for $\beta \in (1,2]$.

Later research focuses on the case in which the regression function is at most as smooth as an element of H. Often, this research demands that the response variables are bounded. For example, Mendelson and Neeman (2010) assume that $g \in T^{\beta/2}(L^2(P))$ for $\beta \in (0,1)$ to obtain a squared $L^2(P)$ error of order $n^{-\beta/(1+p)}$ with high probability by using Tikhonov-regularised least-squares estimators. The authors also show that if the eigenfunctions of the kernel operator T are uniformly bounded in L^∞, then the order can be improved to $n^{-\beta/(\beta+p)}$. Steinwart et al. (2009) relax the condition on the eigenfunctions to the condition

$$\|h\|_\infty \leq C_p \|h\|_H^p \|h\|_{L^2(P)}^{1-p}$$

for all $h \in H$ and some constant $C_p > 0$. The same rate is attained by using clipped Tikhonov-regularised least-squares estimators, including clipped SVMs, and is shown to be optimal. The authors assume that g is in an interpolation space between $L^2(P)$ and H, which is slightly more general than the assumption of Mendelson and Neeman (2010). A detailed discussion about the image of $L^2(P)$ under powers of T and interpolation spaces between $L^2(P)$ and H is given by Steinwart and Scovel (2012).

Lately, the assumption that the response variables must be bounded has been relaxed to allow for subexponential errors. However, the assumption that the regression function is bounded has been maintained. For example, Fischer and Steinwart (2017) assume that $g \in T^{\beta/2}(L^2(P))$ for $\beta \in (0,2]$ and that g is bounded. The authors also assume that $T^{\alpha/2}(L^2(P))$ is continuously embedded in L^∞, with respect to an appropriate norm on $T^{\alpha/2}(L^2(P))$, for some $\alpha < \beta$. This gives the same squared $L^2(P)$ error of order $n^{-\beta/(\beta+p)}$ with high probability by using SVMs.

4. Contribution

In this paper, we provide bounds on the squared $L^2(P)$ error of our Ivanov-regularised least-squares estimator when the regression function comes from an interpolation space between $L^2(P)$ and an RKHS H, which is separable with a bounded and measurable kernel k. We use the norm of the RKHS as our regularisation function. Under the weak assumption that the response variables have bounded variance, we prove a bound on the expected squared $L^2(P)$ error of order $n^{-\beta/2}$ (Theorem 5 on page 12). As far as we are aware, the analysis of an estimator in this setting has not previously been considered. If we assume that the regression function is bounded, then we can clip the estimator and achieve an expected squared $L^2(P)$ error of order $n^{-\beta/(1+\beta)}$ (Theorem 7 on page 13).

Under the stronger assumption that the response variables have subgaussian errors and the regression function comes from an interpolation space between L^∞ and H, we show that the squared $L^2(P)$ error is of order $n^{-\beta/(1+\beta)}$ with high probability (Theorem 11 on page 17). For the settings in which the regression function is bounded, we use training and validation on the data in order to select the size of the constraint on the norm of our estimator. This gives us an adaptive estimation result which does not require us to know which interpolation space contains the regression function. We obtain a squared $L^2(P)$ error of order $n^{-\beta/(1+\beta)}$ in expectation and with high probability, depending on the setting (Theorems 9 and 13 on pages 15 and 18). In order to perform training and validation, the response variables in the validation set must have subgaussian errors. The expectation results for bounded regression functions are of optimal order in the sense discussed at the end of Section 1.

Regression Function	$L^2(P)$ Interpolation	L^∞ Bound	L^∞ Interpolation
Response Variables	Bounded Variance	Bounded Variance	Subgaussian Errors
Bound Type	Expectation	Expectation	High Probability
Bound Order	$n^{-\beta/2}$	$n^{-\beta/(1+\beta)}$ $(*)$	$n^{-\beta/(1+\beta)}$

Table 1: Orders of bounds on squared $L^2(P)$ error

The validation results are summarised in Table 2. Again, the columns for which there is an L^∞ bound on the regression function also make the $L^2(P)$ interpolation assumption. The assumptions on the response variables relate to those in the validation set, which has $\tilde{n}$ data points. We assume that $\tilde{n}$ is equal to some multiple of n. Again, orders of bounds marked with $(*)$ are known to be optimal.

Regression Function	L^∞ Bound	L^∞ Interpolation
Response Variables	Subgaussian Errors	Subgaussian Errors
Bound Type	Expectation	High Probability
Bound Order	$n^{-\beta/(1+\beta)}$ $(*)$	$n^{-\beta/(1+\beta)}$

Table 2: Orders of validation bounds on squared $L^2(P)$ error

The validation results are summarised in Table 2. Again, the columns for which there is an L^∞ bound on the regression function also make the $L^2(P)$ interpolation assumption. The

assumptions on the response variables relate to those in the validation set, which has $\tilde{n}$ data points. We assume that $\tilde{n}$ is equal to some multiple of n. Again, orders of bounds marked with $(*)$ are known to be optimal.

5. Problem Definition

We now formally define our regression problem. For a topological space T, let $\mathcal{B}(T)$ be the Borel σ-algebra of T. Let $(S, \mathcal{S})$ be a measurable space. Assume that (X_i, Y_i) for $1 \leq i \leq n$ are $(S \times \mathbb{R}, \mathcal{S} \otimes \mathcal{B}(\mathbb{R}))$-valued random variables on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, which are i.i.d. with $X_i \sim P$ and $\mathbb{E}(Y_i^2) < \infty$, where $\mathbb{E}$ denotes integration with respect to $\mathbb{P}$. Since any version of $\mathbb{E}(Y_i|X_i)$ is $\sigma(X_i)$-measurable, where $\sigma(X_i)$ is the σ-algebra generated by X_i, we have that $\mathbb{E}(Y_i|X_i) = g(X_i)$ almost surely for some function g which is measurable on $(S, \mathcal{S})$ (Section A3.2 of Williams, 1991). From the definition of conditional expectation and the identical distribution of the (X_i, Y_i), it is clear that we can choose g to be the same for all $1 \leq i \leq n$. The conditional expectation used is that of Kolmogorov, defined using the Radon–Nikodym derivative. Its definition is unique almost surely. Since $\mathbb{E}(Y_i^2) < \infty$, it follows that $g \in L^2(P)$ by Jensen's inequality. To summarise, $\mathbb{E}(Y_i|X_i) = g(X_i)$ almost surely for $1 \leq i \leq n$ with $g \in L^2(P)$. We assume throughout that

$$(Y1) \qquad \mathrm{var}(Y_i|X_i) \leq \sigma^2 \text{ almost surely for } 1 \leq i \leq n.$$

Our results depend on how well g can be approximated by elements of an RKHS H with kernel k. We make the following assumptions.

(H) The RKHS H with kernel k has the following properties:

- The RKHS H is separable.
- The kernel k is bounded.
- The kernel k is a measurable function on $(S \times S, \mathcal{S} \otimes \mathcal{S})$.

We define

$$\|k\|_\infty = \sup_{x \in S} k(x,x)^{1/2} < \infty.$$

We can guarantee that H is separable by, for example, assuming that k is continuous and S is a separable topological space (Lemma 4.33 of Steinwart and Christmann, 2008). The fact that H has a kernel k which is measurable on $(S \times S, \mathcal{S} \otimes \mathcal{S})$ guarantees that all functions in H are measurable on $(S, \mathcal{S})$ (Lemma 4.24 of Steinwart and Christmann, 2008).

6. Ivanov Regularisation

We now consider Ivanov regularisation for least-squares estimators. Let P_n be the empirical distribution of the X_i for $1 \leq i \leq n$. The definition of Ivanov regularisation provides us with the following result.

Lemma 2 *Let $A \subseteq L^2(P)$. It may be that A is a function of $\omega \in \Omega$ and does not contain g. Let*

$$\hat{f} \in \operatorname*{arg\,min}_{f \in A} \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)^2.$$

Then, for all $f \in A$ and all $\omega \in \Omega$, we have

$$\|\hat{f} - f\|^2_{L^2(P_n)} \leq \frac{4}{n} \sum_{i=1}^n (Y_i - g(X_i))(\hat{f}(X_i) - f(X_i)) + 4\|f - g\|^2_{L^2(P_n)}.$$

The function g in Lemma 2 need not be the regression function, however this is the case of interest. In general, the first term of the right-hand side of the inequality must be controlled by bounding it with

$$\sup_{f_1, f_2 \in A} \frac{4}{n} \sum_{i=1}^n (Y_i - g(X_i))(f_1(X_i) - f_2(X_i)). \tag{1}$$

This is usually not measurable. However, if A is a fixed subset of a separable RKHS, then A is separable and the function which evaluates $f \in A$ at X_i is continuous for $1 \leq i \leq n$. This means that the supremum can be replaced with a countable supremum, so the quantity is a random variable on $(\Omega, \mathcal{F})$. Clearly, this term increases as A gets larger. However, if A gets larger, then we may select $f \in A$ closer to g. Hence, we can make the second term of the right-hand side of the inequality in Lemma 2 smaller. This demonstrates the trade-off in selecting the size of A for the Ivanov-regularised least-squares estimator constrained to lie in A.

The next step in analysing $\hat{f}$ is to move to a bound on

$$\|\hat{f} - f\|^2_{L^2(P)} \leq \|\hat{f} - f\|^2_{L^2(P_n)} + \sup_{f_1, f_2 \in A} \left| \|f_1 - f_2\|^2_{L^2(P_n)} - \|f_1 - f_2\|^2_{L^2(P)} \right|. \tag{2}$$

The second term on the right-hand side of this inequality is measurable when A is a fixed subset of a separable RKHS. It also increases with A. Finally, we obtain a bound on

$$\|\hat{f} - g\|^2_{L^2(P)} \leq 2\|\hat{f} - f\|^2_{L^2(P)} + 2\|f - g\|^2_{L^2(P)}.$$

This again demonstrates why $f \in A$ should be close to g.

7. Estimator Definition

In this section, we consider the Ivanov-regularised estimators of Section 6 in the context of RKHS regression. Recall the RKHS H from the definition of our regression problem in Section 5. Let B_H be the closed unit ball of H and $r > 0$. The Ivanov-regularised least-squares estimator constrained to lie in rB_H is

$$\hat{h}_r = \operatorname*{arg\,min}_{f \in rB_H} \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)^2.$$

We also define $\hat{h}_0 = 0$. The next result shows that $\hat{h}_r$ is an SVM with regularisation parameter $\mu(r)$.

Lemma 3 *Assume (H). Let K be the $n \times n$ symmetric matrix with $K_{i,j} = k(X_i, X_j)$. Then K is an $(\mathbb{R}^{n\times n}, \mathcal{B}(\mathbb{R}^{n\times n}))$-valued measurable matrix on $(\Omega, \mathcal{F})$ and there exist an orthogonal matrix A and a diagonal matrix D which are both $(\mathbb{R}^{n\times n}, \mathcal{B}(\mathbb{R}^{n\times n}))$-valued measurable matrices on $(\Omega, \mathcal{F})$ such that $K = ADA^\mathsf{T}$. Furthermore, the diagonal entries of D are non-negative and non-increasing. Let $m = \operatorname{rk} K$, which is a random variable on $(\Omega, \mathcal{F})$. For $r > 0$, if*

$$r^2 < \sum_{i=1}^m D_{i,i}^{-1} (A^\mathsf{T} Y)_i^2,$$

then define $\mu(r) > 0$ by

$$\sum_{i=1}^m \frac{D_{i,i}}{(D_{i,i} + n\mu(r))^2} (A^\mathsf{T} Y)_i^2 = r^2. \tag{3}$$

Otherwise, let $\mu(r) = 0$. We have that $\mu(r)$ is strictly decreasing when $\mu(r) > 0$, and $\mu(r)$ is measurable on $(\Omega \times (0, \infty), \mathcal{F} \otimes \mathcal{B}((0, \infty)))$, where r varies in $(0, \infty)$. Let $a \in \mathbb{R}^n$ be defined by

$$(A^\mathsf{T} a)_i = (D_{i,i} + n\mu(r))^{-1} (A^\mathsf{T} Y)_i$$

for $1 \le i \le m$ and $(A^\mathsf{T} a)_i = 0$ for $m + 1 \le i \le n$, noting that A^T has the inverse A since it is an orthogonal matrix. For $r \ge 0$, we can uniquely define $\hat{h}_r$ by demanding that $\hat{h}_r \in \operatorname{sp}\{k_{X_i} : 1 \le i \le n\}$. This gives

$$\hat{h}_r = \sum_{i=1}^n a_i k_{X_i}$$

for $r > 0$ and $\hat{h}_0 = 0$. We have that $\hat{h}_r$ is a $(H, \mathcal{B}(H))$-valued measurable function on $(\Omega \times [0, \infty), \mathcal{F} \otimes \mathcal{B}([0, \infty)))$, where r varies in $[0, \infty)$.

Let $r > 0$. There are multiple methods for calculating $\mu(r)$ to within a given tolerance $\varepsilon > 0$. We call this value $\nu(r)$.

7.1. Diagonalising K

Firstly, $\mu(r) = 0$ if and only if

$$r \ge \left(\sum_{i=1}^m D_{i,i}^{-1} (A^\mathsf{T} Y)_i^2 \right)^{1/2},$$

so in this case we set $\nu(r) = 0$. Otherwise, $\mu(r) > 0$ and

$$\begin{aligned} r^2 &= \sum_{i=1}^m \frac{D_{i,i}}{(D_{i,i} + n\mu(r))^2} (A^\mathsf{T} Y)_i^2 \\ &\le n^{-2} \left(\sum_{i=1}^m D_{i,i} (A^\mathsf{T} Y)_i^2 \right) \mu(r)^{-2}. \end{aligned}$$

Hence,

$$\mu(r) \le n^{-1} \left(\sum_{i=1}^m D_{i,i} (A^\mathsf{T} Y)_i^2 \right)^{1/2} r^{-1}. \tag{4}$$

The function

$$\sum_{i=1}^m \frac{D_{i,i}}{(D_{i,i} + n\mu)^2} (A^\mathsf{T} Y)_i^2$$

of $\mu \ge 0$ is continuous. Hence, we can calculate $\nu(r)$ using interval bisection on the interval with lower end point 0 and upper end point equal to the right-hand side of (4). We can then approximate a by replacing $\mu(r)$ with $\nu(r)$ in the calculation of a in Lemma 3.

7.2. Not Diagonalising K

We can calculate an alternative $\nu(r)$ without diagonalising K. Note that if $\mu(r) > 0$, then (3) can be written as

$$Y^\mathsf{T} (K + n\mu(r) I)^{-1} K (K + n\mu(r) I)^{-1} Y = r^2.$$

Since $\mu(r)$ is strictly decreasing for $\mu(r) > 0$, we have

$$r \ge \left(Y^\mathsf{T} (K + n\varepsilon I)^{-1} K (K + n\varepsilon I)^{-1} Y \right)^{1/2}$$

if and only if $\mu(r) \in [0, \varepsilon]$, so in this case we set $\nu(r) = \varepsilon$. Otherwise, $\mu(r) > \varepsilon$ and (4) can be written as

$$\mu(r) \le n^{-1} (Y^\mathsf{T} K Y)^{1/2} r^{-1}. \tag{5}$$

The function

$$Y^\mathsf{T} (K + n\mu I)^{-1} K (K + n\mu I)^{-1} Y$$

of $\mu > 0$ is continuous. Hence, we can calculate $\nu(r)$ using interval bisection on the interval with lower end point ε and upper end point equal to the right-hand side of (5). When $\mu(r) > 0$ or K is invertible, we can also calculate a in Lemma 3 using $a = (K + n\mu(r) I)^{-1} Y$. Since $\nu(r) > 0$, we can approximate a by $(K + n\nu(r) I)^{-1} Y$.

If we have that K is invertible, then we can calculate the $\nu(r)$ in Subsection 7.1 while still not diagonalising K. We have $\mu(r) = 0$ if and only if $r \ge (Y^\mathsf{T} K^{-1} Y)^{1/2}$, so in this case we set $\nu(r) = 0$. Otherwise, $\mu(r) > 0$ and (4) can be written as

$$\mu(r) \le n^{-1} (Y^\mathsf{T} K Y)^{1/2} r^{-1},$$

so we can again use interval bisection to calculate $\nu(r)$. We can still approximate a by $(K + n\nu(r) I)^{-1} Y$.

7.3. Approximating $\hat{h}_r$

Having discussed how to approximate $\mu(r)$ by $\nu(r)$ to within a given tolerance $\varepsilon > 0$, we now consider the estimator produced by this approximation. We find that this estimator is equal to $\hat{h}_s$ for some $s > 0$. We only have $\nu(r) = 0$ for the methods considered above when $\mu(r) = 0$, in which case we can let $s = r$ to obtain the approximate estimator $\hat{h}_s = \hat{h}_r$. Otherwise, $\nu(r) > 0$. Let

$$s = \left(\sum_{i=1}^{m} \frac{D_{i,i}}{(D_{i,i} + n\nu(r))^2} (A^\mathsf{T} Y)_i^2 \right)^{1/2}.$$

By (3), we have $\mu(s) = \nu(r)$ and the approximate estimator is equal to $\hat{h}_s$. Assume that r is bounded away from 0 as $n \to \infty$ and let $C > 0$ be some constant not depending on n. We can ensure that s is of the same order as r as $n \to \infty$ by demanding that s is within C of r. This is enough to ensure that the orders of convergence for $\hat{h}_r$ apply to $\hat{h}_s$. In order to attain this value of $\nu(r)$, interval bisection should terminate at $x \in \mathbb{R}$ such that

$$\left(\sum_{i=1}^{m} \frac{D_{i,i}}{(D_{i,i} + nx)^2} (A^\mathsf{T} Y)_i^2 \right)^{1/2}$$

is within C of r. Note that this guarantees $\|\hat{h}_s - \hat{h}_r\|_H \leq C^{1/2}(r+s)^{1/2}$ by Lemma 32.

8. Expectation Bounds

To capture how well g can be approximated by elements of H, we define

$$I_2(g, r) = \inf \left\{ \|h_r - g\|_{L^2(P)}^2 : h_r \in rB_H \right\}$$

for $r > 0$. We consider the distance of g from rB_H because we constrain our estimator $\hat{h}_r$ to lie in this set. The supremum in (1) with $A = rB_H$ can be controlled using the reproducing kernel property and the Cauchy–Schwarz inequality to obtain

$$8r \left(\frac{1}{n^2} \sum_{i,j=1}^{n} (Y_i - g(X_i))(Y_j - g(X_j)) k(X_i, X_j) \right)^{1/2}.$$

The expectation of this quantity can be bounded using Jensen's inequality. Something very similar to this argument gives the first term of the bound in Theorem 4 below. The expectation of the supremum in (2) with $A = rB_H$ can be controlled using symmetrisation (Lemma 2.3.1 of van der Vaart and Wellner, 1996) to obtain

$$2\,\mathbb{E} \left(\sup_{f \in 2rB_H} \left| \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i f(X_i)^2 \right| \right),$$

where the ε_i for $1 \leq i \leq n$ are i.i.d. Rademacher random variables on $(\Omega, \mathcal{F}, \mathbb{P})$, independent of the X_i. Since $\|f\|_\infty \leq 2\|k\|_\infty r$ for all $f \in 2rB_H$, we can remove the squares on the $f(X_i)$ by using the contraction principle for Rademacher processes (Theorem 3.2.1 of Giné and Nickl, 2016). This quantity can then be bounded in a similar way to the supremum in (1), giving the second term of the bound in Theorem 4 below.

Theorem 4 *Assume (Y1) and (H). Let $r > 0$. We have*

$$\mathbb{E}\left(\|\hat{h}_r - g\|_{L^2(P)}^2 \right) \leq \frac{8\|k\|_\infty \sigma r}{n^{1/2}} + \frac{64\|k\|_\infty^2 r^2}{n^{1/2}} + 10 I_2(g, r).$$

We can obtain rates of convergence for our estimator $\hat{h}_r$ if we make an assumption about how well g can be approximated by elements of H. Let us assume

$(g1)$ $\quad g \in [L^2(P), H]_{\beta,\infty}$ with norm at most B for $\beta \in (0,1)$ and $B > 0$.

The assumption $(g1)$, together with Lemma 1, give

$$I_2(g, r) \leq \frac{B^{2/(1-\beta)}}{r^{2\beta/(1-\beta)}} \tag{6}$$

for $r > 0$. We obtain an expectation bound on the squared $L^2(P)$ error of our estimator $\hat{h}_r$ of order $n^{-\beta/2}$.

Theorem 5 *Assume (Y1), (H) and (g1). Let $r > 0$. We have*

$$\mathbb{E}\left(\|\hat{h}_r - g\|_{L^2(P)}^2 \right) \leq \frac{8\|k\|_\infty \sigma r}{n^{1/2}} + \frac{64\|k\|_\infty^2 r^2}{n^{1/2}} + \frac{10 B^{2/(1-\beta)}}{r^{2\beta/(1-\beta)}}.$$

Let $D_1 > 0$. Setting

$$r = D_1 \|k\|_\infty^{-(1-\beta)} B n^{(1-\beta)/4}$$

gives

$$\mathbb{E}\left(\|\hat{h}_r - g\|_{L^2(P)}^2 \right) \leq D_2 \|k\|_\infty^{2\beta} B^2 n^{-\beta/2} + D_3 \|k\|_\infty^{\beta} B \sigma n^{-(1+\beta)/4}$$

for constants $D_2, D_3 > 0$ depending only on D_1 and β.

Since we must let $r \to \infty$ for the initial bound in Theorem 5 to tend to 0, the second term of the initial bound is asymptotically larger than the first. If we ignore the first term and minimise the second and third terms over $r > 0$, we get

$$r = \left(\frac{5\beta}{32(1-\beta)} \right)^{(1-\beta)/2} \|k\|_\infty^{-(1-\beta)} B n^{(1-\beta)/4}.$$

In particular, r is of the form in Theorem 5. This choice of r gives

$$D_2 = 64 \left(\frac{5\beta}{32(1-\beta)} \right)^{1-\beta} + 10 \left(\frac{32(1-\beta)}{5\beta} \right)^{\beta} \text{ and } D_3 = 8 \left(\frac{5\beta}{32(1-\beta)} \right)^{(1-\beta)/2}.$$

The fact that the second term of the initial bound is larger than the first produces some interesting observations. Firstly, the choice of r above does not depend on σ^2. Secondly,

we can decrease the bound if we can find a way to reduce the second term, without having to alter the other terms. The increased size of the second term is due to the fact that the bound on $f \in 2rB_H$ is given by $\|f\|_\infty \leq 2\|k\|_\infty r$ when applying the contraction principle for Rademacher processes. If we can use a bound which does not depend on r, then we can reduce the size of the second term.

We now also assume

$$(g2) \qquad \|g\|_\infty \leq C \text{ for } C > 0$$

and clip our estimator. Let $r > 0$. Since g is bounded in $[-C, C]$, we can make $\hat{h}_r$ closer to g by constraining it to lie in the same interval. Similarly to Chapter 7 of Steinwart and Christmann (2008) and Steinwart et al. (2009), we define the projection $V : \mathbb{R} \to [-C, C]$ by

$$V(t) = \begin{cases} -C & \text{if } \ t < -C \\ t & \text{if } \ |t| \leq C \\ C & \text{if } \ t > C \end{cases}$$

for $t \in \mathbb{R}$. We can apply the inequality

$$\|V\hat{h}_r - Vh_r\|^2_{L^2(P_n)} \leq \|\hat{h}_r - h_r\|^2_{L^2(P_n)}$$

for all $h_r \in rB_H$. We continue analysing $V\hat{h}_r$ by bounding

$$\sup_{f_1, f_2 \in rB_H} \left| \|Vf_1 - Vf_2\|^2_{L^2(P_n)} - \|Vf_1 - Vf_2\|^2_{L^2(P)} \right|.$$

The expectation of the supremum can be bounded in the same way as before, with some adjustments. After symmetrisation, we can remove the squares on the $Vf_1(X_i) - Vf_2(X_i)$ for $f_1, f_2 \in rB_H$ and $1 \leq i \leq n$ by using the contraction principle for Rademacher processes with $\|Vf_1 - Vf_2\|_\infty \leq 2C$. We can then use the triangle inequality to remove $Vf_2(X_i)$, before applying the contraction principle again to remove V. The expectation bound on the squared $L^2(P)$ error of our estimator $V\hat{h}_r$ follows in the same way as before.

Theorem 6 *Assume (Y1), (H) and (g2). Let $r > 0$. We have*

$$\mathbb{E}\left(\|V\hat{h}_r - g\|^2_{L^2(P)}\right) \leq \frac{8\|k\|_\infty(16C + \sigma)r}{n^{1/2}} + 10I_2(g, r).$$

We can obtain rates of convergence for our estimator $V\hat{h}_r$ by again assuming $(g1)$. We obtain an expectation bound on the squared $L^2(P)$ error of $V\hat{h}_r$ of order $n^{-\beta/(1+\beta)}$.

Theorem 7 *Assume (Y1), (H), (g1) and (g2). Let $r > 0$. We have*

$$\mathbb{E}\left(\|V\hat{h}_r - g\|^2_{L^2(P)}\right) \leq \frac{8\|k\|_\infty(16C + \sigma)r}{n^{1/2}} + \frac{10B^{2/(1-\beta)}}{r^{2\beta/(1-\beta)}}.$$

Let $D_1 > 0$. Setting

$$r = D_1\|k\|_\infty^{-(1-\beta)/(1+\beta)} B^{2/(1+\beta)}(16C + \sigma)^{-(1-\beta)/(1+\beta)} n^{(1-\beta)/(2(1+\beta))}$$

gives

$$\mathbb{E}\left(\|V\hat{h}_r - g\|^2_{L^2(P)}\right) \leq D_2\|k\|_\infty^{2\beta/(1+\beta)} B^{2/(1+\beta)}(16C + \sigma)^{2\beta/(1+\beta)} n^{-\beta/(1+\beta)}$$

for a constant $D_2 > 0$ depending only on D_1 and β.

If we minimise the initial bound in Theorem 7 over $r > 0$, we get

$$r = \left(\frac{5\beta}{2(1-\beta)}\right)^{(1-\beta)/(1+\beta)} \|k\|_\infty^{-(1-\beta)/(1+\beta)} B^{2/(1+\beta)}(16C + \sigma)^{-(1-\beta)/(1+\beta)} n^{(1-\beta)/(2(1+\beta))}.$$

In particular, r is of the form in Theorem 7. This choice of r gives

$$D_2 = 2 \cdot 5^{(1-\beta)/(1+\beta)} \cdot 4^{2\beta/(1+\beta)} \left(\left(\frac{2\beta}{1-\beta}\right)^{(1-\beta)/(1+\beta)} + \left(\frac{1-\beta}{2\beta}\right)^{2\beta/(1+\beta)} \right).$$

Although the second bound in Theorem 7 is of theoretical interest, it is in practice impossible to select r of the correct order in n for the bound to hold without knowing β. Since assuming that we know β is not realistic, we must use some other method for determining a good choice of r.

8.1. Validation

Suppose that we have an independent second data set $(\tilde{X}_i, \tilde{Y}_i)$ for $1 \leq i \leq \tilde{n}$ which are $(S \times \mathbb{R}, \mathcal{S} \otimes \mathcal{B}(\mathbb{R}))$-valued random variables on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let the $(\tilde{X}_i, \tilde{Y}_i)$ be i.i.d. with $\tilde{X}_i \sim P$ and $\mathbb{E}(\tilde{Y}_i|\tilde{X}_i) = g(\tilde{X}_i)$ almost surely. Let $\rho \geq 0$ and $R \subseteq [0, \rho]$ be non-empty and compact. Furthermore, let $F = \{V\hat{h}_r : r \in R\}$. We estimate a value of r which makes the squared $L^2(P)$ error of $V\hat{h}_r$ small by

$$\hat{r} = \arg\min_{r \in R} \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} (V\hat{h}_r(\tilde{X}_i) - \tilde{Y}_i)^2.$$

The minimum is attained because Lemma 32 shows that it is the minimum of a continuous function over a compact set. In the event of ties, we may take $\hat{r}$ to be the infimum of all points attaining the minimum. Lemma 33 shows that the estimator $\hat{r}$ is a random variable on $(\Omega, \mathcal{F})$. Hence, by Lemma 3, $\hat{h}_{\hat{r}}$ is a $(H, \mathcal{B}(H))$-valued random variable on $(\Omega, \mathcal{F})$.

The definition of $\hat{r}$ means that we can analyse $V\hat{h}_{\hat{r}}$ using Lemma 2. The expectation of the supremum in (1) with $A = F$ can be bounded using chaining (Theorem 2.3.7 of Giné and Nickl, 2016). The diameter of $(F, \|\cdot\|_\infty)$ is $2C$, which is an important bound for the use of chaining. Hence, this form of analysis can only be performed under the assumption $(g2)$. After symmetrisation, the expectation of the supremum in (2) with $A = F$ can be bounded in the same way. In order to perform chaining, we need to make an assumption on the behaviour of the errors of the response variables $\tilde{Y}_i$ for $1 \leq i \leq \tilde{n}$. Let U and V be random variables on $(\Omega, \mathcal{F}, \mathbb{P})$. We say U is σ^2-subgaussian if

$$\mathbb{E}(\exp(tU)) \leq \exp(\sigma^2 t^2/2)$$

for all $t \in \mathbb{R}$. We say U is σ^2-subgaussian given V if

$$\mathbb{E}(\exp(tU)|V) \leq \exp(\sigma^2 t^2/2)$$

almost surely for all $t \in \mathbb{R}$. We assume

$$(\tilde{Y}) \qquad \tilde{Y}_i - g(\tilde{X}_i) \text{ is } \tilde{\sigma}^2\text{-subgaussian given } \tilde{X}_i \text{ for } 1 \leq i \leq \tilde{n}.$$

This is stronger than the equivalent of the assumption $(Y1)$, that $\mathrm{var}(\tilde{Y}_i|\tilde{X}_i) \leq \tilde{\sigma}^2$ almost surely.

Theorem 8 *Assume (H) and $(\tilde{Y})$. Let $r_0 \in R$. We have*

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)}\right)$$

is at most

$$\frac{32C(4C+\tilde{\sigma})}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|^2_\infty \rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right) + 10\,\mathbb{E}\left(\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right).$$

In order for us to apply the validation result in Theorem 8 to the initial bound in Theorem 7, we need to make an assumption on R. We assume either

$$(R1) \qquad R = [0, \rho] \text{ for } \rho = an^{1/2} \text{ and } a > 0$$

or

$$(R2) \qquad R = \{bi : 0 \leq i \leq I-1\} \cup \{an^{1/2}\} \text{ and } \rho = an^{1/2} \text{ for } a, b > 0 \text{ and } I = \lceil an^{1/2}/b \rceil.$$

The assumption $(R1)$ is mainly of theoretical interest and would make it difficult to calculate $\hat{r}$ in practice. The estimator $\hat{r}$ can be computed under the assumption $(R2)$, since in this case R is finite. We obtain an expectation bound on the squared $L^2(P)$ error of $V\hat{h}_{\hat{r}}$ of order $n^{-\beta/(1+\beta)}$. This is the same order in n as the second bound in Theorem 7.

Theorem 9 *Assume (Y1), (H), (g1), (g2) and $(\tilde{Y})$. Also assume (R1) or (R2) and that $\tilde{n}$ increases at least linearly in n. We have*

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)}\right) \leq D_1 n^{-\beta/(1+\beta)}$$

for a constant $D_1 > 0$ not depending on n or $\tilde{n}$.

9. High-Probability Bounds

In this section, we look at how to extend our expectation bounds on our estimators to high-probability bounds. In order to do this, we must control the second term of the bound in Lemma 2 with $A = rB_H$ for $r > 0$, which is

$$\|h_r - g\|^2_{L^2(P_n)} \tag{7}$$

for $h_r \in rB_H$. There is no way to bound (7) in high-probability without strict assumptions on g. In fact, the most natural assumption is $(g2)$ that $\|g\|_\infty \leq C$ for $C > 0$, which we assume throughout this section. Bounding (7) also requires us to introduce a new measure of how well g can be approximated by elements of H. We define

$$I_\infty(g, r) = \inf\left\{\|h_r - g\|^2_\infty : h_r \in rB_H\right\}$$

for $r > 0$. Note that $I_\infty(g, r) \geq I_2(g, r)$. Using $I_\infty(g, r)$ instead of $I_2(g, r)$ means that we do not have to control (7) by relying on $\|h_r - g\|_\infty \leq \|k\|_\infty r + C$. Using Hoeffding's inequality, this would add a term of order $r^2 t^{1/2}/n^{1/2}$ for $t \geq 1$ to the bound in Theorem 10 below, which holds with probability $1 - 3e^{-t}$, substantially increasing its size.

It may be possible to avoid this problem by instead considering the Ivanov-regularised least-squares estimator

$$\hat{f}_r = \operatorname*{arg\,min}_{f \in V(rB_H)} \frac{1}{n}\sum_{i=1}^n (f(X_i) - Y_i)^2$$

for $r > 0$, where $V(rB_H) = \{Vh_r : h_r \in rB_H\}$. The second term of the bound in Lemma 2 with $A = V(rB_H)$ is

$$\|Vh_r - g\|^2_{L^2(P_n)} \tag{8}$$

for $h_r \in rB_H$. Since $\|Vh_r - g\|_\infty \leq 2C$, using Hoeffding's inequality to bound (8) would only add a term of order $C^2 t^{1/2}/n^{1/2}$ to the bound in Theorem 10 below, which would not alter its size. However, the calculation and analysis of the estimator $\hat{f}_r$ is outside the scope of this paper. This is because the calculation of $\hat{f}_r$ involves minimising a quadratic form subject to a series of linear constraints, and its analysis requires a bound on the supremum in (1) with $A = V(rB_H)$.

The rest of the analysis of $V\hat{h}_r$ is similar to that of the expectation bound. The supremum in (1) with $A = rB_H$ can again be bounded by

$$8r\left(\frac{1}{n^2}\sum_{i,j=1}^n (Y_i - g(X_i))(Y_j - g(X_j))k(X_i, X_j)\right)^{1/2}.$$

The quadratic form can be bounded using Lemma 35, under an assumption on the behaviour of the errors of the response variables Y_i for $1 \leq i \leq n$. The proof of Theorem 10 below uses a very similar argument to this one. The supremum in (2) with $A = rB_H$ can be bounded using Talagrand's inequality (Theorem A.9.1 of Steinwart and Christmann, 2008). In order to use Lemma 35, we must assume

$$(Y2) \qquad Y_i - g(X_i) \text{ is } \sigma^2\text{-subgaussian given } X_i \text{ for } 1 \leq i \leq n.$$

This assumption is stronger than $(Y1)$. In particular, Theorem 6 still holds under the assumptions $(Y2)$, (H) and $(g2)$.

Theorem 10 *Assume (Y2), (H) and (g2). Let $r > 0$ and $t \geq 1$. With probability at least $1 - 3e^{-t}$, we have*

$$\|V\hat{h}_r - g\|^2_{L^2(P)} \leq \left(D_1 + D_2 r^{1/2} + D_3 r\right) t^{1/2} n^{-1/2} + D_4 t n^{-1} + 10 I_\infty(g, r)$$

for some constants $D_1, D_2, D_3, D_4 > 0$ not depending on n or t.

We can obtain rates of convergence for our estimator $V\hat{h}_r$, but we must make a new assumption about how well g can be approximated by elements of H, instead of $(g1)$. We now assume

$(g3)$ $\quad g \in [L^\infty, H]_{\beta,\infty}$ with norm at most B for $\beta \in (0,1)$ and $B > 0$,

instead of $g \in [L^2(P), H]_{\beta,\infty}$ with norm at most B. This assumption is stronger than $(g1)$, as it implies that the norm of $g \in [L^2(P), H]_{\beta,\infty}$ is

$$\sup_{t>0}(t^{-\beta} \inf_{h\in H}(\|g-h\|_{L^2(P)} + t\|h\|_H)) \leq \sup_{t>0}(t^{-\beta} \inf_{h\in H}(\|g-h\|_{L^\infty} + t\|h\|_H)) \leq B.$$

In particular, Theorem 7 still holds under the assumptions $(Y1)$, (H), $(g2)$ and $(g3)$ or $(Y2)$, (H), $(g2)$ and $(g3)$. The assumption $(g3)$, together with Lemma 1, give

$$I_\infty(g,r) \leq \frac{B^{2/(1-\beta)}}{r^{2\beta/(1-\beta)}}. \tag{9}$$

We obtain a high-probability bound on the squared $L^2(P)$ error of $V\hat{h}_r$ of order $t^{\beta/(1+\beta)}n^{-\beta/(1+\beta)}$ with probability at least $1-e^{-t}$.

Theorem 11 *Assume (Y2), (H), (g2) and (g3). Let $r > 0$ and $t \geq 1$. With probability at least $1-3e^{-t}$, we have*

$$\|V\hat{h}_r - g\|^2_{L^2(P)} \leq \left(D_1 + D_2 r^{1/2} + D_3 r\right) t^{1/2} n^{-1/2} + D_4 t n^{-1} + D_5 r^{-2\beta/(1-\beta)}$$

for some constants $D_1, D_2, D_3, D_4, D_5 > 0$ not depending on n or t. Let $D_6 > 0$. Setting

$$r = D_6 \left(\frac{n}{t}\right)^{\frac{1-\beta}{2(1+\beta)}}$$

gives

$$\|V\hat{h}_r - g\|^2_{L^2(P)} \leq D_7 \left(\frac{t}{n}\right)^{\frac{\beta}{1+\beta}} + D_8 \left(\frac{t}{n}\right)^{\frac{1+3\beta}{4(1+\beta)}} + D_1 \left(\frac{t}{n}\right)^{1/2} + D_4 \left(\frac{t}{n}\right)$$

for constants $D_7, D_8 > 0$ not depending on n or t.

Since we must let $r \to \infty$ for the initial bound in Theorem 11 to tend to 0, the asymptotically largest terms in the bound are

$$D_3 r t^{1/2} n^{-1/2} + D_5 r^{-2\beta/(1-\beta)}.$$

If we minimise this over $r > 0$, we get r of the form in Theorem 11 with

$$D_6 = \left(\frac{2D_5\beta}{D_3(1-\beta)}\right)^{(1-\beta)/(1+\beta)}.$$

It is possible to avoid the dependence of r on t by setting

$$r = D_6 n^{(1-\beta)/(2(1+\beta))}$$

to obtain the bound

$$D_7 t^{1/2} n^{-\beta/(1+\beta)} + D_8 t^{1/2} n^{-(1+3\beta)/(4(1+\beta))} + D_1 t^{1/2} n^{-1/2} + D_4 t n^{-1}.$$

9.1. Validation

We now extend our expectation bound on $V\hat{h}_{\hat{r}}$ to a high-probability bound. The supremum in (1) with $A = F$ can be bounded using chaining (Exercise 1 of Section 2.3 of Giné and Nickl, 2016), while the supremum in (2) with $A = F$ can be bounded using Talagrand's inequality.

Theorem 12 *Assume (H) and $(\tilde{Y})$. Let $r_0 \in R$ and $t \geq 1$. With probability at least $1-3e^{-t}$, we have*

$$\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)}$$

is at most

$$\frac{20C(C+\tilde{\sigma})t^{1/2}}{\tilde{n}^{1/2}}\left(1 + 32\left(\left(2\log\left(2 + \frac{\|k\|^2_\infty \rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right)\right)$$
$$+ \frac{48C^2 t^{1/2}}{\tilde{n}^{1/2}} + \frac{16C^2 t}{3\tilde{n}} + 10\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}.$$

We can apply the validation result in Theorem 12 to the initial bound in Theorem 11 by assuming either $(R1)$ or $(R2)$. We obtain a high-probability bound on the squared $L^2(P)$ error of $V\hat{h}_{\hat{r}}$ of order $t^{1/2}n^{-\beta/(1+\beta)}$ with probability at least $1-e^{-t}$. This is the same order in n as the second bound in Theorem 11.

Theorem 13 *Assume (Y2), (H), (g2), (g3) and $(\tilde{Y})$. Let $t \geq 1$. Also assume (R1) or (R2) and that $\tilde{n}$ increases at least linearly in n. With probability at least $1-6e^{-t}$, we have*

$$\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)} \leq D_1 t^{1/2} n^{-\beta/(1+\beta)} + D_2 t n^{-1}$$

for constants $D_1, D_2 > 0$ not depending on n, $\tilde{n}$ or t.

10. Discussion

In this paper, we show how Ivanov regularisation can be used to produce smooth estimators which have a small squared $L^2(P)$ error. We first consider the case in which the regression function lies in an interpolation space between $L^2(P)$ and an RKHS H. We achieve bounds on the squared $L^2(P)$ error under the assumption that H is separable with a bounded and measurable kernel. Under the weak assumption that the response variables have bounded variance, we prove an expectation bound on the squared $L^2(P)$ error of our estimator of order $n^{-\beta/2}$. Here, β parametrises the interpolation space between $L^2(P)$ and H containing the regression function. As far as we are aware, the analysis of an estimator in this setting has not previously been considered.

If we assume that the regression function is bounded, then we can clip the estimator and show that the clipped estimator has an expected squared $L^2(P)$ error of order $n^{-\beta/(1+\beta)}$. Under the stronger assumption that the response variables have subgaussian errors and that the regression function comes from an interpolation space between L^∞ and H, we show that the squared $L^2(P)$ error is of order $n^{-\beta/(1+\beta)}$ with high probability. For the settings in which

the regression function is bounded, we can use training and validation on the data set to obtain bounds of the same order of $n^{-\beta/(1+\beta)}$. This allows us to select the size of the norm constraint for our Ivanov regularisation without knowing which interpolation space contains the regression function. The response variables in the validation set must have subgaussian errors.

The expectation bounds of order $n^{-\beta/(1+\beta)}$ for bounded regression functions is optimal in the sense discussed at the end of Section 1. We use Ivanov regularisation instead of Tikhonov regularisation to control empirical processes over balls in the RKHS. By contrast, the analysis of Tikhonov-regularised estimators usually uses the spectral decomposition of the kernel operator (Mendelson and Neeman, 2010; Steinwart et al., 2009). Analysing the Ivanov-regularised estimator using this decomposition would give a more complete picture of the differences between Ivanov and Tikhonov regularisation for RKHS regression.

It would be useful to extend the lower bound of order $n^{-\beta/(1+\beta)}$, discussed at the end of Section 1, to the case in which the regression function lies in an interpolation space between L^∞ and the RKHS. This would show that our high-probability bounds are also of optimal order. However, it is possible that estimation can be performed with a high-probability bound on the squared $L^2(P)$ error of smaller order.

Appendix A. Proof of Expectation Bound for Unbounded Regression Function

The proofs of all of the bounds in this paper follow the outline in Section 6. We first prove Lemma 2.

Proof of Lemma 2 Since $f \in A$, the definition of $\hat{f}$ gives

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(X_i) - Y_i)^2 \leq \frac{1}{n}\sum_{i=1}^{n}(f(X_i) - Y_i)^2.$$

Expanding

$$(\hat{f}(X_i) - Y_i)^2 = \Big((\hat{f}(X_i) - f(X_i)) + (f(X_i) - Y_i)\Big)^2,$$

substituting into the above and rearranging gives

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(X_i) - f(X_i))^2 \leq \frac{2}{n}\sum_{i=1}^{n}(Y_i - f(X_i))(\hat{f}(X_i) - f(X_i)).$$

Substituting

$$Y_i - f(X_i) = (Y_i - g(X_i)) + (g(X_i) - f(X_i))$$

into the above and applying the Cauchy–Schwarz inequality to the second term gives

$$\begin{aligned}\|\hat{f} - f\|^2_{L^2(P_n)} \leq{}& \frac{2}{n}\sum_{i=1}^{n}(Y_i - g(X_i))(\hat{f}(X_i) - f(X_i)) \\ &+ 2\|g - f\|_{L^2(P_n)}\|\hat{f} - f\|_{L^2(P_n)}.\end{aligned}$$

For constants $a, b \in \mathbb{R}$ and a variable $x \in \mathbb{R}$, we have

$$x^2 \leq a + 2bx \implies x^2 \leq 2a + 4b^2$$

by completing the square and rearranging. Applying this result to the above inequality proves the lemma. ■

The following lemma is useful for bounding the expectation of both of the suprema in Section 6.

Lemma 14 *Assume (H). Let the ε_i be random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ such that $\mathbb{E}(\varepsilon_i|X) = 0$ almost surely and $\mathrm{var}(\varepsilon_i|X) \leq \sigma^2$ almost surely for $1 \leq i \leq n$ and $\mathrm{cov}(\varepsilon_i, \varepsilon_j|X) = 0$ almost surely for $1 \leq i, j \leq n$ with $i \neq j$. Then*

$$\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(X_i)\right|\right) \leq \frac{\|k\|_\infty \sigma r}{n^{1/2}}.$$

Proof This proof method is due to Remark 6.1 of Sriperumbudur (2016). By the reproducing kernel property and the Cauchy–Schwarz inequality, we have

$$\begin{aligned}\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(X_i)\right| &= \sup_{f \in rB_H}\left|\left\langle \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i k_{X_i}, f\right\rangle_H\right| \\ &= r\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i k_{X_i}\right\|_H \\ &= r\left(\frac{1}{n^2}\sum_{i,j=1}^{n}\varepsilon_i\varepsilon_j k(X_i, X_j)\right)^{1/2}.\end{aligned}$$

By Jensen's inequality, we have

$$\begin{aligned}\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(X_i)\right|\middle| X\right) &\leq r\left(\frac{1}{n^2}\sum_{i,j=1}^{n}\mathrm{cov}(\varepsilon_i, \varepsilon_j|X)k(X_i, X_j)\right)^{1/2} \\ &\leq r\left(\frac{\sigma^2}{n^2}\sum_{i=1}^{n}k(X_i, X_i)\right)^{1/2}\end{aligned}$$

almost surely and again, by Jensen's inequality, we have

$$\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(X_i)\right|\right) \leq r\left(\frac{\sigma^2\|k\|^2_\infty}{n}\right)^{1/2}.$$

The result follows. ■

We bound the distance between $\hat{h}_r$ and h_r in the $L^2(P_n)$ norm for $r > 0$ and $h_r \in rB_H$.

Lemma 15 *Assume (Y1) and (H). Let $h_r \in rB_H$. We have*

$$\mathbb{E}\left(\|\hat{h}_r - h_r\|^2_{L^2(P_n)}\right) \leq \frac{4\|k\|_\infty \sigma r}{n^{1/2}} + 4\|h_r - g\|^2_{L^2(P)}.$$

Proof By Lemma 2 with $A = rB_H$, we have

$$\|\hat{h}_r - h_r\|^2_{L^2(P_n)} \leq \frac{4}{n}\sum_{i=1}^n (Y_i - g(X_i))(\hat{h}_r(X_i) - h_r(X_i)) + 4\|h_r - g\|^2_{L^2(P_n)}.$$

We now bound the expectation of the right-hand side. We have

$$\mathbb{E}\left(\|h_r - g\|^2_{L^2(P_n)}\right) = \|h_r - g\|^2_{L^2(P)}.$$

Furthermore,

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))h_r(X_i)\right) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^n \mathbb{E}(Y_i - g(X_i)|X_i)h_r(X_i)\right) = 0.$$

Finally, by Lemma 14 with $\varepsilon_i = Y_i - g(X_i)$, we have

$$\begin{aligned}\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))\hat{h}_r(X_i)\right) &\leq \mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))f(X_i)\right|\right)\\ &\leq \frac{\|k\|_\infty \sigma r}{n^{1/2}}.\end{aligned}$$

The result follows. ■

The following lemma is useful for moving the bound on the distance between $\hat{h}_r$ and h_r from the $L^2(P_n)$ norm to the $L^2(P)$ norm for $r > 0$ and $h_r \in rB_H$.

Lemma 16 *Assume (H). We have*

$$\mathbb{E}\left(\sup_{f \in rB_H}\left|\|f\|^2_{L^2(P_n)} - \|f\|^2_{L^2(P)}\right|\right) \leq \frac{8\|k\|^2_\infty r^2}{n^{1/2}}.$$

Proof Let the ε_i for $1 \leq i \leq n$ be i.i.d. Rademacher random variables on $(\Omega, \mathcal{F}, \mathbb{P})$, independent of the X_i. Lemma 2.3.1 of van der Vaart and Wellner (1996) shows

$$\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^n f(X_i)^2 - \int f^2 dP\right|\right) \leq 2\,\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(X_i)^2\right|\right)$$

by symmetrisation. Since $|f(X_i)| \leq \|k\|_\infty r$ for all $f \in rB_H$, we find

$$\frac{f(X_i)^2}{2\|k\|_\infty r}$$

is a contraction vanishing at 0 as a function of $f(X_i)$ for all $1 \leq i \leq n$. By Theorem 3.2.1 of Giné and Nickl (2016), we have

$$\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i \frac{f(X_i)^2}{2\|k\|_\infty r}\right| \middle| X\right) \leq 2\,\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(X_i)\right| \middle| X\right)$$

almost surely. By Lemma 14 with $\sigma^2 = 1$, we have

$$\mathbb{E}\left(\sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(X_i)\right|\right) \leq \frac{\|k\|_\infty r}{n^{1/2}}.$$

The result follows. ■

We move the bound on the distance between $\hat{h}_r$ and h_r from the $L^2(P_n)$ norm to the $L^2(P)$ norm for $r > 0$ and $h_r \in rB_H$.

Corollary 17 *Assume (Y1) and (H). Let $h_r \in rB_H$. We have*

$$\mathbb{E}\left(\|\hat{h}_r - h_r\|^2_{L^2(P)}\right) \leq \frac{4\|k\|_\infty \sigma r}{n^{1/2}} + \frac{32\|k\|^2_\infty r^2}{n^{1/2}} + 4\|h_r - g\|^2_{L^2(P)}.$$

Proof By Lemma 15, we have

$$\mathbb{E}\left(\|\hat{h}_r - h_r\|^2_{L^2(P_n)}\right) \leq \frac{4\|k\|_\infty \sigma r}{n^{1/2}} + 4\|h_r - g\|^2_{L^2(P)}.$$

Since $\hat{h}_r - h_r \in 2rB_H$, by Lemma 16 we have

$$\begin{aligned}\mathbb{E}\left(\|\hat{h}_r - h_r\|^2_{L^2(P)} - \|\hat{h}_r - h_r\|^2_{L^2(P_n)}\right) &\leq \mathbb{E}\left(\sup_{f \in 2rB_H}\left|\|f\|^2_{L^2(P_n)} - \|f\|^2_{L^2(P)}\right|\right)\\ &\leq \frac{32\|k\|^2_\infty r^2}{n^{1/2}}.\end{aligned}$$

The result follows. ■

We bound the distance between $\hat{h}_r$ and g in the $L^2(P)$ norm for $r > 0$ to prove Theorem 4.

Proof of Theorem 4 Fix $h_r \in rB_H$. We have

$$\begin{aligned}\|\hat{h}_r - g\|^2_{L^2(P)} &\leq \left(\|\hat{h}_r - h_r\|^2_{L^2(P)} + \|h_r - g\|^2_{L^2(P)}\right)^2\\ &\leq 2\|\hat{h}_r - h_r\|^2_{L^2(P)} + 2\|h_r - g\|^2_{L^2(P)}.\end{aligned}$$

By Corollary 17, we have

$$\mathbb{E}\left(\|\hat{h}_r - h_r\|^2_{L^2(P)}\right) \leq \frac{4\|k\|_\infty \sigma r}{n^{1/2}} + \frac{32\|k\|^2_\infty r^2}{n^{1/2}} + 4\|h_r - g\|^2_{L^2(P)}.$$

Hence,

$$\mathbb{E}\left(\|\hat{h}_r - g\|^2_{L^2(P)}\right) \leq \frac{8\|k\|_\infty \sigma r}{n^{1/2}} + \frac{64\|k\|^2_\infty r^2}{n^{1/2}} + 10\|h_r - g\|^2_{L^2(P)}.$$

Taking an infimum over $h_r \in rB_H$ proves the result. ■

We assume $(g1)$ to prove Theorem 5.

Proof of Theorem 5 The initial bound follows from Theorem 4 and (6). Based on this bound, setting

$$r = D_1 \|k\|_\infty^{-(1-\beta)} B n^{(1-\beta)/4}$$

gives

$$\mathbb{E}\left(\|\hat{h}_r - g\|_{L^2(P)}^2\right) \leq \left(64D_1^2 + 10D_1^{-2\beta/(1-\beta)}\right) \|k\|_\infty^{2\beta} B^2 n^{-\beta/2} + 8D_1 \|k\|_\infty^{\beta} B\sigma n^{-(1+\beta)/4}.$$

Hence, the next bound follows with

$$D_2 = 64D_1^2 + 10D_1^{-2\beta/(1-\beta)} \text{ and } D_3 = 8D_1.$$

■

Appendix B. Proof of Expectation Bound for Bounded Regression Function

We can obtain a bound on the distance between $V\hat{h}_r$ and Vh_r in the $L^2(P_n)$ norm for $r > 0$ and $h_r \in rB_H$ from Lemma 15. The following lemma is useful for moving the bound on the distance between $V\hat{h}_r$ and Vh_r from the $L^2(P_n)$ norm to the $L^2(P)$ norm.

Lemma 18 *Assume (H). We have*

$$\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \|Vf_1 - Vf_2\|_{L^2(P_n)}^2 - \|Vf_2 - Vf_2\|_{L^2(P)}^2 \right|\right) \leq \frac{64\|k\|_\infty Cr}{n^{1/2}}.$$

Proof Let the ε_i for $1 \leq i \leq n$ be i.i.d. Rademacher random variables on $(\Omega, \mathcal{F}, \mathbb{P})$, independent of the X_i. Lemma 2.3.1 of van der Vaart and Wellner (1996) shows

$$\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n (Vf_1(X_i) - Vf_2(X_i))^2 - \int (Vf_1 - Vf_2)^2 dP \right|\right)$$

is at most

$$2\,\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n \varepsilon_i (Vf_1(X_i) - Vf_2(X_i))^2 \right|\right)$$

by symmetrisation. Since $|Vf_1(X_i) - Vf_2(X_i)| \leq 2C$ for all $f_1, f_2 \in rB_H$, we find

$$\frac{(Vf_1(X_i) - Vf_2(X_i))^2}{4C}$$

is a contraction vanishing at 0 as a function of $Vf_1(X_i) - Vf_2(X_i)$ for all $1 \leq i \leq n$. By Theorem 3.2.1 of Giné and Nickl (2016), we have

$$\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n \varepsilon_i \frac{(Vf_1(X_i) - Vf_2(X_i))^2}{4C} \right| \middle| X\right)$$

is at most

$$2\,\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n \varepsilon_i (Vf_1(X_i) - Vf_2(X_i)) \right| \middle| X\right)$$

almost surely. Therefore,

$$\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n (Vf_1(X_i) - Vf_2(X_i))^2 - \int (Vf_1 - Vf_2)^2 dP \right|\right)$$

is at most

$$16C\,\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n \varepsilon_i (Vf_1(X_i) - Vf_2(X_i)) \right|\right) \leq 32C\,\mathbb{E}\left(\sup_{f \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n \varepsilon_i Vf(X_i) \right|\right)$$

by the triangle inequality. Again, by Theorem 3.2.1 of Giné and Nickl (2016), we have

$$\mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n (Vf_1(X_i) - Vf_2(X_i))^2 - \int (Vf_1 - Vf_2)^2 dP \right|\right)$$

is at most

$$64C\,\mathbb{E}\left(\sup_{f \in rB_H} \left| \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(X_i) \right|\right)$$

since V is a contraction vanishing at 0. The result follows from Lemma 14 with $\sigma^2 = 1$. ■

We move the bound on the distance between $V\hat{h}_r$ and Vh_r from the $L^2(P_n)$ norm to the $L^2(P)$ norm for $r > 0$ and $h_r \in rB_H$.

Corollary 19 *Assume (Y1) and (H). Let $h_r \in rB_H$. We have*

$$\mathbb{E}\left(\|V\hat{h}_r - Vh_r\|_{L^2(P)}^2\right) \leq \frac{4\|k\|_\infty (16C + \sigma) r}{n^{1/2}} + 4\|h_r - g\|_{L^2(P)}^2.$$

Proof By Lemma 15, we have

$$\mathbb{E}\left(\|\hat{h}_r - h_r\|_{L^2(P_n)}^2\right) \leq \frac{4\|k\|_\infty \sigma r}{n^{1/2}} + 4\|h_r - g\|_{L^2(P)}^2,$$

so

$$\mathbb{E}\left(\|V\hat{h}_r - Vh_r\|_{L^2(P_n)}^2\right) \leq \frac{4\|k\|_\infty \sigma r}{n^{1/2}} + 4\|h_r - g\|_{L^2(P)}^2.$$

Since $\hat{h}_r, h_r \in rB_H$, by Lemma 18 we have

$$\begin{aligned}
&\mathbb{E}\left(\|V\hat{h}_r - Vh_r\|_{L^2(P)}^2 - \|V\hat{h}_r - Vh_r\|_{L^2(P_n)}^2\right) \\
&\leq \mathbb{E}\left(\sup_{f_1, f_2 \in rB_H} \left| \|Vf_1 - Vf_2\|_{L^2(P_n)}^2 - \|Vf_2 - Vf_2\|_{L^2(P)}^2 \right|\right) \\
&\leq \frac{64\|k\|_\infty Cr}{n^{1/2}}.
\end{aligned}$$

The result follows. ■

We assume $(g2)$ to bound the distance between $V\hat{h}_r$ and g in the $L^2(P)$ norm for $r > 0$ and prove Theorem 6.

Proof of Theorem 6 Fix $h_r \in rB_H$. We have

$$\begin{aligned}\|V\hat{h}_r - g\|^2_{L^2(P)} &\leq \left(\|V\hat{h}_r - Vh_r\|^2_{L^2(P)} + \|Vh_r - g\|^2_{L^2(P)}\right)^2 \\ &\leq 2\|V\hat{h}_r - Vh_r\|^2_{L^2(P)} + 2\|Vh_r - g\|^2_{L^2(P)} \\ &\leq 2\|V\hat{h}_r - Vh_r\|^2_{L^2(P)} + 2\|h_r - g\|^2_{L^2(P)}.\end{aligned}$$

By Corollary 19, we have

$$\mathbb{E}\left(\|V\hat{h}_r - Vh_r\|^2_{L^2(P)}\right) \leq \frac{4\|k\|_\infty(16C+\sigma)r}{n^{1/2}} + 4\|h_r - g\|^2_{L^2(P)}.$$

Hence,

$$\mathbb{E}\left(\|V\hat{h}_r - g\|^2_{L^2(P)}\right) \leq \frac{8\|k\|_\infty(16C+\sigma)r}{n^{1/2}} + 10\|h_r - g\|^2_{L^2(P)}.$$

Taking an infimum over $h_r \in rB_H$ proves the result. ■

We assume $(g1)$ to prove Theorem 7.

Proof of Theorem 7 The initial bound follows from Theorem 6 and (6). Based on this bound, setting

$$r = D_1\|k\|_\infty^{-(1-\beta)/(1+\beta)}B^{2/(1+\beta)}(16C+\sigma)^{-(1-\beta)/(1+\beta)}n^{(1-\beta)/(2(1+\beta))}$$

gives

$$\mathbb{E}\left(\|V\hat{h}_r - g\|^2_{L^2(P)}\right)$$

is at most

$$\left(8D_1 + 10D_1^{-2\beta/(1-\beta)}\right)\|k\|_\infty^{2\beta/(1+\beta)}B^{2/(1+\beta)}(16C+\sigma)^{2\beta/(1+\beta)}n^{-\beta/(1+\beta)}.$$

Hence, the next bound follows with

$$D_2 = 8D_1 + 10D_1^{-2\beta/(1-\beta)}.$$

■

Appendix C. Proof of Expectation Bound for Validation

We need to introduce some definitions for stochastic processes. A stochastic process W on $(\Omega, \mathcal{F})$ indexed by a metric space (M, d) is d^2-subgaussian if it is centred and $W(s) - W(t)$ is $d(s,t)^2$-subgaussian for all $s, t \in M$. W is separable if there exists a countable set $M_0 \subseteq M$ such that the following holds for all $\omega \in \Omega_0$, where $\mathbb{P}(\Omega_0) = 1$. For all $s \in M$ and $\varepsilon > 0$, $W(s)$ is in the closure of $\{W(t) : t \in M_0, d(s,t) \leq \varepsilon\}$.

We also need to introduce the concept of covering numbers for the next result. The covering number $N(M, d, \varepsilon)$ is the minimum number of d-balls of size $\varepsilon > 0$ needed to cover M.

The following lemma is useful for bounding the expectation of both of the suprema in Section 6.

Lemma 20 *Assume (H). Let the ε_i be random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ such that $(\tilde{X}_i, \varepsilon_i)$ is i.i.d. for $1 \leq i \leq \tilde{n}$ and let ε_i be $\tilde{\sigma}^2$-subgaussian given $\tilde{X}_i$. Let $r_0 \in R$, $f_0 = V\hat{h}_{r_0}$ and*

$$W(f) = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \varepsilon_i(f(\tilde{X}_i) - f_0(\tilde{X}_i))$$

for $f \in F$. Then W is $\tilde{\sigma}^2\|\cdot\|^2_\infty/\tilde{n}$-subgaussian given $\tilde{X}$ and separable on $(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2})$. Furthermore,

$$\mathbb{E}\left(\sup_{f\in F}|W(f)|\right) \leq \frac{4C\tilde{\sigma}}{\tilde{n}^{1/2}}\left(\left(2\log\left(2 + \frac{\|k\|^2_\infty\rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right).$$

Proof Let $W_i(f) = \varepsilon_i(f(\tilde{X}_i) - f_0(\tilde{X}_i))$ for $1 \leq i \leq \tilde{n}$ and $f \in F$. Note that the W_i are independent and centred. Since $W_i(f_1) - W_i(f_2)$ is $\tilde{\sigma}^2\|f_1 - f_2\|^2_\infty$-subgaussian given $\tilde{X}_i$ for $1 \leq i \leq \tilde{n}$ and $f_1, f_2 \in F$, the process W is $\tilde{\sigma}^2\|\cdot\|^2_\infty/\tilde{n}$-subgaussian given $\tilde{X}$. By Lemma 32, we have that $(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2})$ is separable. Hence, W is separable on $(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2})$ since it is continuous. The diameter of $(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2})$ is

$$D = \sup_{f_1, f_2 \in F} \tilde{\sigma}\|f_1 - f_2\|_\infty/\tilde{n}^{1/2} \leq 2C\tilde{\sigma}/\tilde{n}^{1/2}.$$

We have

$$\begin{aligned}\int_0^\infty (\log(N(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2}, \varepsilon)))^{1/2}d\varepsilon &= \int_0^\infty (\log(N(F, \|\cdot\|_\infty, \tilde{n}^{1/2}\varepsilon/\tilde{\sigma})))^{1/2}d\varepsilon \\ &= \frac{\tilde{\sigma}}{\tilde{n}^{1/2}}\int_0^\infty (\log(N(F, \|\cdot\|_\infty, u)))^{1/2}du.\end{aligned}$$

This is finite by Lemma 37. Hence, by Theorem 2.3.7 of Giné and Nickl (2016) and Lemma 37, we have

$$\mathbb{E}\left(\sup_{f\in F}|W(f)|\middle|\tilde{X}, X, Y\right)$$

is at most

$$\begin{aligned}
&\mathbb{E}(|W(f_0)|\,|\tilde{X},X,Y) + 2^{5/2}\int_0^{C\tilde{\sigma}/\tilde{n}^{1/2}} (\log(2N(F,\tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2},\varepsilon)))^{1/2}d\varepsilon \\
&= 2^{5/2}\int_0^{C\tilde{\sigma}/\tilde{n}^{1/2}} (\log(2N(F,\|\cdot\|_\infty,\tilde{n}^{1/2}\varepsilon/\tilde{\sigma})))^{1/2}d\varepsilon \\
&= \frac{2^{5/2}\tilde{\sigma}}{\tilde{n}^{1/2}}\int_0^{C} (\log(2N(F,\|\cdot\|_\infty,u)))^{1/2}du \\
&\le \frac{2^{5/2}\tilde{\sigma}}{\tilde{n}^{1/2}}\left(\left(\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2}C + \left(\frac{\pi}{2}\right)^{1/2}C\right) \\
&= \frac{4C\tilde{\sigma}}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right)
\end{aligned}$$

almost surely, noting $W(f_0)=0$. The result follows. ■

We bound the distance between $V\hat{h}_{\hat{r}}$ and $V\hat{h}_{r_0}$ in the $L^2(\tilde{P}_{\tilde{n}})$ norm for $r_0 \in R$.

Lemma 21 *Assume (H) and ($\tilde{Y}$). Let $r_0 \in R$. We have*

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(\tilde{P}_{\tilde{n}})}\right)$$

is at most

$$\frac{16C\tilde{\sigma}}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right) + 4\,\mathbb{E}\left(\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right).$$

Proof By Lemma 2 with $A=F$ and n, X, Y and P_n replaced by $\tilde{n}$, $\tilde{X}$, $\tilde{Y}$ and $\tilde{P}_{\tilde{n}}$, we have

$$\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(\tilde{P}_{\tilde{n}})} \le \frac{4}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(\tilde{Y}_i - g(\tilde{X}_i))(V\hat{h}_{\hat{r}}(\tilde{X}_i) - V\hat{h}_{r_0}(\tilde{X}_i)) + 4\|V\hat{h}_{r_0} - g\|^2_{L^2(\tilde{P}_{\tilde{n}})}.$$

We now bound the expectation of the right-hand side. We have

$$\mathbb{E}\left(\|V\hat{h}_{r_0} - g\|^2_{L^2(\tilde{P}_{\tilde{n}})}\right) = \mathbb{E}\left(\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right).$$

Let $f_0 = V\hat{h}_{r_0}$. By Lemma 20 with $\varepsilon_i = Y_i - g(X_i)$, we have

$$\begin{aligned}
&\mathbb{E}\left(\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(\tilde{Y}_i - g(\tilde{X}_i))(V\hat{h}_{\hat{r}}(\tilde{X}_i) - V\hat{h}_{r_0}(\tilde{X}_i))\right) \\
&\le \mathbb{E}\left(\sup_{f\in F}\left|\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(\tilde{Y}_i - g(\tilde{X}_i))(f(\tilde{X}_i) - f_0(\tilde{X}_i))\right|\right) \\
&\le \frac{4C\tilde{\sigma}}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right).
\end{aligned}$$

The result follows. ■

The following lemma is useful for moving the bound on the distance between $V\hat{h}_{\hat{r}}$ and $V\hat{h}_{r_0}$ from the $L^2(\tilde{P}_{\tilde{n}})$ norm to the $L^2(P)$ norm for $r_0 \in R$.

Lemma 22 *Assume (H). Let $r_0 \in R$ and $f_0 = V\hat{h}_{r_0}$. We have*

$$\mathbb{E}\left(\sup_{f\in F}\left|\|f-f_0\|^2_{L^2(\tilde{P}_{\tilde{n}})} - \|f-f_0\|^2_{L^2(P)}\right|\right) \le \frac{64C^2}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right).$$

Proof Let the ε_i be i.i.d. Rademacher random variables on $(\Omega,\mathcal{F},\mathbb{P})$ for $1\le i\le\tilde{n}$, independent of $\tilde{X}$, X and Y. Lemma 2.3.1 of van der Vaart and Wellner (1996) shows

$$\mathbb{E}\left(\sup_{f\in F}\left|\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(f(\tilde{X}_i) - f_0(\tilde{X}_i))^2 - \int(f-f_0)^2dP\right|\,\middle|\,X,Y\right)$$

is at most

$$2\,\mathbb{E}\left(\sup_{f\in F}\left|\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\varepsilon_i(f(\tilde{X}_i) - f_0(\tilde{X}_i))^2\right|\,\middle|\,X,Y\right)$$

almost surely by symmetrisation. Since $|f(\tilde{X}_i) - f_0(\tilde{X}_i)| \le 2C$ for all $f\in F$, we find

$$\frac{(f(\tilde{X}_i) - f_0(\tilde{X}_i))^2}{4C}$$

is a contraction vanishing at 0 as a function of $f(\tilde{X}_i) - f_0(\tilde{X}_i)$ for all $1\le i\le\tilde{n}$. By Theorem 3.2.1 of Giné and Nickl (2016), we have

$$\mathbb{E}\left(\sup_{f\in F}\left|\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\varepsilon_i\frac{(f(\tilde{X}_i) - f_0(\tilde{X}_i))^2}{4C}\right|\,\middle|\,\tilde{X},X,Y\right)$$

is at most

$$2\,\mathbb{E}\left(\sup_{f\in F}\left|\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\varepsilon_i(f(\tilde{X}_i) - f_0(\tilde{X}_i))\right|\,\middle|\,\tilde{X},X,Y\right)$$

almost surely. Therefore,

$$\mathbb{E}\left(\sup_{f\in F}\left|\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(f(\tilde{X}_i) - f_0(\tilde{X}_i))^2 - \int(f-f_0)^2dP\right|\,\middle|\,X,Y\right)$$

is at most

$$16C\,\mathbb{E}\left(\sup_{f\in F}\left|\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\varepsilon_i(f(\tilde{X}_i) - f_0(\tilde{X}_i))\right|\,\middle|\,X,Y\right)$$

almost surely. The result follows from Lemma 20 with $\tilde{\sigma}^2 = 1$. ■

We move the bound on the distance between $V\hat{h}_{\hat{r}}$ and $V\hat{h}_{r_0}$ from the $L^2(\tilde{P}_{\tilde{n}})$ norm to the $L^2(P)$ norm for $r_0 \in R$.

Corollary 23 *Assume (H) and ($\tilde{Y}$). Let $r_0 \in R$. We have*

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(P)}\right)$$

is at most

$$\frac{16C(4C+\tilde{\sigma})}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|^2_\infty\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right)+4\,\mathbb{E}\left(\|V\hat{h}_{r_0}-g\|^2_{L^2(P)}\right).$$

Proof By Lemma 21, we have

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(\tilde{P}_{\tilde{n}})}\right)$$

is at most

$$\frac{16C\tilde{\sigma}}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|^2_\infty\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right)+4\,\mathbb{E}\left(\|V\hat{h}_{r_0}-g\|^2_{L^2(P)}\right).$$

Let $f_0 = V\hat{h}_{r_0}$. Since $\hat{h}_{\hat{r}} \in F$, by Lemma 22 we have

$$\begin{aligned}
&\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(P)} - \|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(\tilde{P}_{\tilde{n}})}\right)\\
\le\ &\mathbb{E}\left(\sup_{f\in F}\left|\|f-f_0\|^2_{L^2(\tilde{P}_{\tilde{n}})} - \|f-f_0\|^2_{L^2(P)}\right|\right)\\
\le\ &\frac{64C^2}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|^2_\infty\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right).
\end{aligned}$$

The result follows. ■

We bound the distance between $V\hat{h}_{\hat{r}}$ and g in the $L^2(P)$ norm to prove Theorem 8.

Proof of Theorem 8 We have

$$\begin{aligned}
\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)} &\le \left(\|V\hat{h}_{\hat{r}} - Vh_{r_0}\|^2_{L^2(P)} + \|Vh_{r_0} - g\|^2_{L^2(P)}\right)^2\\
&\le 2\|V\hat{h}_{\hat{r}} - Vh_{r_0}\|^2_{L^2(P)} + 2\|Vh_{r_0} - g\|^2_{L^2(P)}.
\end{aligned}$$

By Corollary 23, we have

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(P)}\right)$$

is at most

$$\frac{16C(4C+\tilde{\sigma})}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|^2_\infty\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right)+4\,\mathbb{E}\left(\|V\hat{h}_{r_0}-g\|^2_{L^2(P)}\right).$$

Hence,

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)}\right)$$

is at most

$$\frac{32C(4C+\tilde{\sigma})}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|^2_\infty\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right)+10\,\mathbb{E}\left(\|V\hat{h}_{r_0}-g\|^2_{L^2(P)}\right).$$

■

We assume the conditions of Theorem 7 to prove Theorem 9.

Proof of Theorem 9 If we assume ($R1$), then $r_0 = an^{(1-\beta)/(2(1+\beta))} \in R$ and

$$\mathbb{E}\left(\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right) \le \frac{8\|k\|_\infty(16C+\sigma)an^{(1-\beta)/(2(1+\beta))}}{n^{1/2}} + \frac{10B^{2/(1-\beta)}}{a^{2\beta/(1-\beta)}n^{\beta/(1+\beta)}}$$

by Theorem 7. If we assume ($R2$), then there is at least one $r_0 \in R$ such that

$$an^{(1-\beta)/(2(1+\beta))} \le r_0 < an^{(1-\beta)/(2(1+\beta))} + b$$

and

$$\begin{aligned}
\mathbb{E}\left(\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right) &\le \frac{8\|k\|_\infty(16C+\sigma)r_0}{n^{1/2}} + \frac{10B^{2/(1-\beta)}}{r_0^{2\beta/(1-\beta)}}\\
&\le \frac{8\|k\|_\infty(16C+\sigma)\left(an^{(1-\beta)/(2(1+\beta))}+b\right)}{n^{1/2}} + \frac{10B^{2/(1-\beta)}}{a^{2\beta/(1-\beta)}n^{\beta/(1+\beta)}}
\end{aligned}$$

by Theorem 7. In either case,

$$\mathbb{E}\left(\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right) \le D_2 n^{-\beta/(1+\beta)}$$

for some constant $D_2 > 0$ not depending on n or $\tilde{n}$. By Theorem 8, we have

$$\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)}\right) \le D_3\log(n)^{1/2}\tilde{n}^{-1/2} + 10D_2 n^{-\beta/(1+\beta)}$$

for some constant $D_3 > 0$ not depending on n or $\tilde{n}$. Since $\tilde{n}$ increases at least linearly in n, there exists some constant $D_4 > 0$ such that $\tilde{n} \ge D_4 n$. We then have

$$\begin{aligned}
\mathbb{E}\left(\|V\hat{h}_{\hat{r}} - g\|^2_{L^2(P)}\right) &\le D_4^{-1/2}D_3\log(n)^{1/2}n^{-1/2} + 10D_2 n^{-\beta/(1+\beta)}\\
&\le D_1 n^{-\beta/(1+\beta)}
\end{aligned}$$

for some constant $D_1 > 0$ not depending on n or $\tilde{n}$. ■

Appendix D. Proof of High-Probability Bound for Bounded Regression Function

We bound the distance between $\hat{h}_r$ and h_r in the $L^2(P_n)$ norm for $r > 0$ and $h_r \in rB_H$.

Lemma 24 *Assume (Y2) and (H). Let $r > 0$, $h_r \in rB_H$ and $t \geq 1$. With probability at least $1 - 2e^{-t}$, we have*

$$\|\hat{h}_r - h_r\|^2_{L^2(P_n)} \leq \frac{20\|k\|_\infty \sigma r t^{1/2}}{n^{1/2}} + 4\|h_r - g\|^2_\infty.$$

Proof By Lemma 2 with $A = rB_H$, we have

$$\|\hat{h}_r - h_r\|^2_{L^2(P_n)} \leq \frac{4}{n}\sum_{i=1}^n (Y_i - g(X_i))(\hat{h}_r(X_i) - h_r(X_i)) + 4\|h_r - g\|^2_{L^2(P_n)}.$$

We now bound the right-hand side. We have

$$\|h_r - g\|^2_{L^2(P_n)} \leq \|h_r - g\|^2_\infty.$$

Furthermore,

$$-\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))h_r(X_i)$$

is subgaussian given X with parameter

$$\frac{1}{n^2}\sum_{i=1}^n \sigma^2 h_r(X_i)^2 \leq \frac{\|k\|^2_\infty \sigma^2 r^2}{n}.$$

So we have

$$-\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))h_r(X_i) \leq \frac{\|k\|_\infty \sigma r (2t)^{1/2}}{n^{1/2}} \leq \frac{2\|k\|_\infty \sigma r t^{1/2}}{n^{1/2}}$$

with probability at least $1 - e^{-t}$ by Chernoff bounding. Finally, we have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))\hat{h}_r(X_i) &\leq \sup_{f \in rB_H}\left|\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))f(X_i)\right| \\
&= \sup_{f \in rB_H}\left|\left\langle \frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))k_{X_i}, f\right\rangle_H\right| \\
&= r\left\|\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))k_{X_i}\right\|_H \\
&= r\left(\frac{1}{n^2}\sum_{i,j=1}^n (Y_i - g(X_i))(Y_j - g(X_j))k(X_i, X_j)\right)^{1/2}
\end{aligned}$$

by the reproducing kernel property and the Cauchy–Schwarz inequality. Let K be the $n \times n$ matrix with $K_{i,j} = k(X_i, X_j)$ and let ε be the vector of the $Y_i - g(X_i)$. Then

$$\frac{1}{n^2}\sum_{i,j=1}^n (Y_i - g(X_i))(Y_j - g(X_j))k(X_i, X_j) = \varepsilon^\mathsf{T}(n^{-2}K)\varepsilon.$$

Furthermore, since k is a measurable function on $(S \times S, \mathcal{S} \otimes \mathcal{S})$, we have that $n^{-2}K$ is an $(\mathbb{R}^{n\times n}, \mathcal{B}(\mathbb{R}^{n\times n}))$-valued measurable matrix on $(\Omega, \mathcal{F})$ and non-negative-definite. Let a_i for $1 \leq i \leq n$ be the eigenvalues of $n^{-2}K$. Then

$$\max_i a_i \leq \operatorname{tr}(n^{-2}K) \leq n^{-1}\|k\|^2_\infty$$

and

$$\operatorname{tr}((n^{-2}K)^2) = \|a\|^2_2 \leq \|a\|^2_1 \leq n^{-2}\|k\|^4_\infty.$$

Therefore, by Lemma 35 with $M = n^{-2}K$, we have

$$\varepsilon^\mathsf{T}(n^{-2}K)\varepsilon \leq \|k\|^2_\infty \sigma^2 n^{-1}(1 + 2t + 2(t^2 + t)^{1/2})$$

and

$$\frac{1}{n}\sum_{i=1}^n (Y_i - g(X_i))\hat{h}_r(X_i) \leq \frac{3\|k\|_\infty \sigma r t^{1/2}}{n^{1/2}}$$

with probability at least $1 - e^{-t}$. The result follows. ■

The following lemma is useful for bounding the supremum in (2).

Lemma 25 *Let $D > 0$ and $A \subseteq L^\infty$ be separable with $\|f\|_\infty \leq D$ for all $f \in A$. Let*

$$Z = \sup_{f \in A}\left|\|f\|^2_{L^2(P_n)} - \|f\|^2_{L^2(P)}\right|.$$

Then, for $t > 0$, we have

$$Z \leq \mathbb{E}(Z) + \left(\frac{2D^4 t}{n} + \frac{4D^2\,\mathbb{E}(Z)t}{n}\right)^{1/2} + \frac{2D^2 t}{3n}$$

with probability at least $1 - e^{-t}$.

Proof We have

$$Z = \sup_{f \in A}\left|\sum_{i=1}^n n^{-1}\left(f(X_i)^2 - \|f\|^2_{L^2(P)}\right)\right|$$

and

$$\begin{aligned}
\mathbb{E}\left(n^{-1}\left(f(X_i)^2 - \|f\|^2_{L^2(P)}\right)\right) &= 0, \\
n^{-1}\left|f(X_i)^2 - \|f\|^2_{L^2(P)}\right| &\leq \frac{D^2}{n}, \\
\mathbb{E}\left(n^{-2}\left(f(X_i)^2 - \|f\|^2_{L^2(P)}\right)^2\right) &\leq \frac{D^4}{n^2}
\end{aligned}$$

for all $1 \leq i \leq n$ and $f \in A$. Furthermore, A is separable, so Z is a random variable on $(\Omega, \mathcal{F})$ and we can use Talagrand's inequality (Theorem A.9.1 of Steinwart and Christmann, 2008) to show

$$Z > \mathbb{E}(Z) + \left(2t\left(\frac{D^4}{n} + \frac{2D^2\,\mathbb{E}(Z)}{n}\right)\right)^{1/2} + \frac{2tD^2}{3n}$$

with probability at most e^{-t}. The result follows. ■

The following lemma is useful for moving the bound on the distance between $V\hat{h}_r$ and Vh_r from the $L^2(P_n)$ norm to the $L^2(P)$ norm for $r > 0$ and $h_r \in rB_H$.

Lemma 26 *Assume (H). Let $r > 0$ and $t \geq 1$. With probability at least $1 - e^{-t}$, we have*

$$\sup_{f_1, f_2 \in rB_H} \left| \|Vf_1 - Vf_2\|^2_{L^2(P_n)} - \|Vf_1 - Vf_2\|^2_{L^2(P)} \right|$$

is at most

$$\frac{8\left(C^2 + 4\|k\|_\infty^{1/2} C^{3/2} r^{1/2} + 8\|k\|_\infty Cr\right) t^{1/2}}{n^{1/2}} + \frac{8C^2 t}{3n}.$$

Proof Let $A = \{Vf_1 - Vf_2 : f_1, f_2 \in rB_H\}$ and

$$Z = \sup_{f_1, f_2 \in rB_H} \left| \|Vf_1 - Vf_2\|^2_{L^2(P_n)} - \|Vf_1 - Vf_2\|^2_{L^2(P)} \right|.$$

Then $A \subseteq L^\infty$ is separable because H is separable and has a bounded kernel k. Furthermore, $\|Vf_1 - Vf_2\|_\infty \leq 2C$ for all $f_1, f_2 \in rB_H$. By Lemma 25, we have

$$Z \leq \mathbb{E}(Z) + \left(\frac{32C^4 t}{n} + \frac{16C^2\, \mathbb{E}(Z) t}{n} \right)^{1/2} + \frac{8C^2 t}{3n}$$

with probability at least $1 - e^{-t}$. By Lemma 18, we have

$$\mathbb{E}(Z) \leq \frac{64\|k\|_\infty Cr}{n^{1/2}}.$$

The result follows. ■

We move the bound on the distance between $V\hat{h}_r$ and Vh_r from the $L^2(P_n)$ norm to the $L^2(P)$ norm for $r > 0$ and $h_r \in rB_H$.

Corollary 27 *Assume (Y2) and (H). Let $r > 0$, $h_r \in rB_H$ and $t \geq 1$. With probability at least $1 - 3e^{-t}$, we have*

$$\|V\hat{h}_r - Vh_r\|^2_{L^2(P)}$$

is at most

$$\frac{4\left(2C^2 + 8\|k\|_\infty^{1/2} C^{3/2} r^{1/2} + \|k\|_\infty (16C + 5\sigma) r\right) t^{1/2}}{n^{1/2}} + \frac{8C^2 t}{3n} + 4\|h_r - g\|^2_\infty.$$

Proof By Lemma 24, we have

$$\|\hat{h}_r - h_r\|^2_{L^2(P_n)} \leq \frac{20\|k\|_\infty \sigma r t^{1/2}}{n^{1/2}} + 4\|h_r - g\|^2_\infty.$$

with probability at least $1 - 2e^{-t}$, so

$$\|V\hat{h}_r - Vh_r\|^2_{L^2(P_n)} \leq \frac{20\|k\|_\infty \sigma r t^{1/2}}{n^{1/2}} + 4\|h_r - g\|^2_\infty.$$

Since $\hat{h}_r, h_r \in rB_H$, by Lemma 26 we have

$$\begin{aligned}
&\|V\hat{h}_r - Vh_r\|^2_{L^2(P)} - \|V\hat{h}_r - Vh_r\|^2_{L^2(P_n)} \\
\leq &\sup_{f_1, f_2 \in rB_H} \left| \|Vf_1 - Vf_2\|^2_{L^2(P_n)} - \|Vf_2 - Vf_2\|^2_{L^2(P)} \right| \\
\leq &\frac{8\left(C^2 + 4\|k\|_\infty^{1/2} C^{3/2} r^{1/2} + 8\|k\|_\infty Cr\right) t^{1/2}}{n^{1/2}} + \frac{8C^2 t}{3n}
\end{aligned}$$

with probability at least $1 - e^{-t}$. The result follows. ■

We assume (g2) to bound the distance between $V\hat{h}_r$ and g in the $L^2(P)$ norm for $r > 0$ and prove Theorem 10.

Proof of Theorem 10 Fix $h_r \in rB_H$. We have

$$\begin{aligned}
\|V\hat{h}_r - g\|^2_{L^2(P)} &\leq \left(\|V\hat{h}_r - Vh_r\|^2_{L^2(P)} + \|Vh_r - g\|^2_{L^2(P)} \right)^2 \\
&\leq 2\|V\hat{h}_r - Vh_r\|^2_{L^2(P)} + 2\|Vh_r - g\|^2_{L^2(P)} \\
&\leq 2\|V\hat{h}_r - Vh_r\|^2_{L^2(P)} + 2\|h_r - g\|^2_{L^2(P)}.
\end{aligned}$$

By Corollary 27, we have

$$\|V\hat{h}_r - Vh_r\|^2_{L^2(P)}$$

is at most

$$\frac{4\left(2C^2 + 8\|k\|_\infty^{1/2} C^{3/2} r^{1/2} + \|k\|_\infty (16C + 5\sigma) r\right) t^{1/2}}{n^{1/2}} + \frac{8C^2 t}{3n} + 4\|h_r - g\|^2_\infty.$$

with probability at least $1 - 3e^{-t}$. Hence,

$$\|V\hat{h}_r - g\|^2_{L^2(P)}$$

is at most

$$\frac{8\left(2C^2 + 8\|k\|_\infty^{1/2} C^{3/2} r^{1/2} + \|k\|_\infty (16C + 5\sigma) r\right) t^{1/2}}{n^{1/2}} + \frac{16C^2 t}{3n} + 10\|h_r - g\|^2_\infty.$$

Take a sequence of $h_{r,n} \in rB_H$ for $n \geq 1$ with

$$\|h_{r,n} - g\|^2_\infty \downarrow I_\infty(g, r)$$

as $n \to \infty$ and let $A_n \in \mathcal{F}$ be the set with $\mathbb{P}(A_n) \geq 1 - 3e^{-t}$ on which the above inequality holds with $h_r = h_{r,n}$. Since the A_n are non-increasing sets, we have that

$$\mathbb{P}\left(\bigcap_{n=1}^\infty A_n \right) = \lim_{n \to \infty} \mathbb{P}(A_n) \geq 1 - 3e^{-t}.$$

The result follows with

$$D_1 = 16C^2,\ D_2 = 64\|k\|_\infty^{1/2} C^{3/2},\ D_3 = 8\|k\|_\infty (16C + 5\sigma) \text{ and } D_4 = 16C^2/3.$$

■

We assume $(g3)$ to prove Theorem 11.

Proof of Theorem 11 The initial bound follows from Theorem 10 and (9) with $D_5 = 10B^{2/(1-\beta)}$. Based on this bound, setting

$$r = D_6 t^{-(1-\beta)/(2(1+\beta))} n^{(1-\beta)/(2(1+\beta))}$$

gives

$$\|V\hat{h}_r - g\|^2_{L^2(P)}$$

is at most

$$\left(D_3 D_6 + D_5 D_6^{-2\beta/(1-\beta)}\right) t^{\beta/(1+\beta)} n^{-\beta/(1+\beta)} + D_2 D_6^{1/2} t^{(1+3\beta)/(4(1+\beta))} n^{-(1+3\beta)/(4(1+\beta))}$$
$$+ D_1 t^{1/2} n^{-1/2} + D_4 t n^{-1}.$$

Hence, the next bound follows with

$$D_7 = D_3 D_6 + D_5 D_6^{-2\beta/(1-\beta)} \text{ and } D_8 = D_2 D_6^{1/2}.$$

■

Appendix E. Proof of High-Probability Bound for Validation

We need to introduce some new notation for the next result. Let U and V be random variables on $(\Omega, \mathcal{F})$. Then

$$\|U\|_{\psi_2} = \inf\{a \in (0,\infty) : \mathbb{E}\,\psi_2(|U|/a) \le 1\},$$
$$\|U|V\|_{\psi_2} = \inf\{a \in (0,\infty) : \mathbb{E}(\psi_2(|U|/a)|V) \le 1 \text{ almost surely}\},$$

where $\psi_2(x) = \exp(x^2) - 1$ for $x \in \mathbb{R}$. Note that these infima are attained by the monotone convergence theorem. Exercise 5 of Section 2.3 of Giné and Nickl (2016) shows that $\|U\|_{\psi_2}$ is a norm on the space of U such that $\|U\|_{\psi_2} < \infty$ and $\|U|V\|_{\psi_2}$ is a norm on the space of U such that $\|U|V\|_{\psi_2} < \infty$.

We bound the distance between $V\hat{h}_{\hat{r}}$ and $V\hat{h}_{r_0}$ in the $L^2(\tilde{P}_{\tilde{n}})$ norm for $r_0 \in R$.

Lemma 28 *Assume (H) and $(\tilde{Y})$. Let $r_0 \in R$ and $t \ge 1$. With probability at least $1 - 2e^{-t}$, we have*

$$\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(\tilde{P}_{\tilde{n}})}$$

is at most

$$\frac{292C\tilde{\sigma}t^{1/2}}{\tilde{n}^{1/2}}\left(\left(2\log\left(1 + \frac{\|k\|^2_\infty \rho^2}{8C^2}\right)\right)^{1/2} + \pi^{1/2}\right) + \frac{24C^2 t^{1/2}}{\tilde{n}^{1/2}} + 4\|V\hat{h}_{r_0} - g\|^2_{L^2(P)}.$$

Proof By Lemma 2 with $A = F$ and n, X, Y and P_n replaced by $\tilde{n}$, $\tilde{X}$, $\tilde{Y}$ and $\tilde{P}_{\tilde{n}}$, we have

$$\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|^2_{L^2(\tilde{P}_{\tilde{n}})} \le \frac{4}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(\tilde{Y}_i - g(\tilde{X}_i))(V\hat{h}_{\hat{r}}(\tilde{X}_i) - V\hat{h}_{r_0}(\tilde{X}_i)) + 4\|V\hat{h}_{r_0} - g\|^2_{L^2(\tilde{P}_{\tilde{n}})}.$$

We now bound the right-hand side. We have

$$\|V\hat{h}_{r_0} - g\|^2_{L^2(\tilde{P}_{\tilde{n}})} = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\left((V\hat{h}_{r_0}(\tilde{X}_i) - g(\tilde{X}_i))^2 - \|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right) + \|V\hat{h}_{r_0} - g\|^2_{L^2(P)}.$$

Since

$$\left|(V\hat{h}_{r_0}(\tilde{X}_i) - g(\tilde{X}_i))^2 - \|V\hat{h}_{r_0} - g\|^2_{L^2(P)}\right| \le 4C^2$$

for all $1 \le i \le \tilde{n}$, we find

$$\|V\hat{h}_{r_0} - g\|^2_{L^2(\tilde{P}_{\tilde{n}})} - \|V\hat{h}_{r_0} - g\|^2_{L^2(P)} > t$$

with probability at most

$$\exp\left(-\frac{\tilde{n}t^2}{32C^4}\right).$$

by Hoeffding's inequality. Therefore, we have

$$\|V\hat{h}_{r_0} - g\|^2_{L^2(\tilde{P}_{\tilde{n}})} - \|V\hat{h}_{r_0} - g\|^2_{L^2(P)} \le \frac{32^{1/2}C^2 t^{1/2}}{\tilde{n}^{1/2}} \le \frac{6C^2 t^{1/2}}{\tilde{n}^{1/2}}$$

with probability at least $1 - e^{-t}$. Now let $f_0 = V\hat{h}_{r_0}$ and

$$W(f) = \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(\tilde{Y}_i - g(\tilde{X}_i))(f(\tilde{X}_i) - f_0(\tilde{X}_i))$$

for $f \in F$. W is $\tilde{\sigma}^2\|\cdot\|^2_\infty/\tilde{n}$-subgaussian given $\tilde{X}$ and separable on $(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2})$ by Lemma 20. The diameter of $(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2})$ is

$$D = \sup_{f_1, f_2 \in F} \tilde{\sigma}\|f_1 - f_2\|_\infty/\tilde{n}^{1/2} \le 2C\tilde{\sigma}/\tilde{n}^{1/2}.$$

From Lemma 37, we have

$$\int_0^\infty (\log(N(F, \tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2}, \varepsilon)))^{1/2} d\varepsilon = \int_0^\infty (\log(N(F, \|\cdot\|_\infty, \tilde{n}^{1/2}\varepsilon/\tilde{\sigma})))^{1/2} d\varepsilon$$
$$= \frac{\tilde{\sigma}}{\tilde{n}^{1/2}}\int_0^\infty (\log(N(F, \|\cdot\|_\infty, u)))^{1/2} du$$

is finite. Hence, by Exercise 1 of Section 2.3 of Giné and Nickl (2016) and Lemma 37, we have

$$\left\|\sup_{f \in F}|W(f)|\,\middle|\,\tilde{X}, X, Y\right\|_{\psi_2}$$

is at most

$$\begin{aligned}
&\left\|W(f_0)\middle|\tilde{X},X,Y\right\|_{\psi_2} + 1536^{1/2}\int_0^{2C\tilde{\sigma}/\tilde{n}^{1/2}}(\log N(F,\tilde{\sigma}\|\cdot\|_\infty/\tilde{n}^{1/2},\varepsilon))^{1/2}d\varepsilon \\
&= 1536^{1/2}\int_0^{2C\tilde{\sigma}/\tilde{n}^{1/2}}(\log N(F,\|\cdot\|_\infty,\tilde{n}^{1/2}\varepsilon/\tilde{\sigma}))^{1/2}d\varepsilon \\
&= \frac{1536^{1/2}\tilde{\sigma}}{\tilde{n}^{1/2}}\int_0^{2C}(\log N(F,\|\cdot\|_\infty,u))^{1/2}du \\
&\leq \frac{1536^{1/2}\tilde{\sigma}}{\tilde{n}^{1/2}}\left(2\left(\log\left(1+\frac{\|k\|_\infty^2\rho^2}{8C^2}\right)\right)^{1/2}C+(2\pi)^{1/2}C\right) \\
&= \frac{3072^{1/2}C\tilde{\sigma}}{\tilde{n}^{1/2}}\left(\left(2\log\left(1+\frac{\|k\|_\infty^2\rho^2}{8C^2}\right)\right)^{1/2}+\pi^{1/2}\right),
\end{aligned}$$

noting $W(f_0)=0$. By Chernoff bounding, we have $\sup_{f\in F}|W(f)|$ is at most

$$\begin{aligned}
&\frac{3072^{1/2}C\tilde{\sigma}(t+\log(2))^{1/2}}{\tilde{n}^{1/2}}\left(\left(2\log\left(1+\frac{\|k\|_\infty^2\rho^2}{8C^2}\right)\right)^{1/2}+\pi^{1/2}\right) \\
&\leq \frac{73C\tilde{\sigma}t^{1/2}}{\tilde{n}^{1/2}}\left(\left(2\log\left(1+\frac{\|k\|_\infty^2\rho^2}{8C^2}\right)\right)^{1/2}+\pi^{1/2}\right)
\end{aligned}$$

with probability at least $1-e^{-t}$. In particular,

$$\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}(\tilde{Y}_i-g(\tilde{X}_i))(V\hat{h}_{\hat{r}}(\tilde{X}_i)-V\hat{h}_{r_0}(\tilde{X}_i))$$

is at most

$$\frac{73C\tilde{\sigma}t^{1/2}}{\tilde{n}^{1/2}}\left(\left(2\log\left(1+\frac{\|k\|_\infty^2\rho^2}{8C^2}\right)\right)^{1/2}+\pi^{1/2}\right)$$

with probability at least $1-e^{-t}$. The result follows. ■

The following lemma is useful for moving the bound on the distance between $V\hat{h}_{\hat{r}}$ and $V\hat{h}_{r_0}$ from the $L^2(\tilde{P}_{\tilde{n}})$ norm to the $L^2(P)$ norm for $r_0\in R$.

Lemma 29 *Assume (H). Let $r_0\in R$, $f_0=V\hat{h}_{r_0}$ and $t\geq 1$. With probability at least $1-e^{-t}$, we have*

$$\sup_{f\in F}\left|\|f-f_0\|_{L^2(\tilde{P}_{\tilde{n}})}^2-\|f-f_0\|_{L^2(P)}^2\right|$$

is at most

$$\frac{10C^2t^{1/2}}{\tilde{n}^{1/2}}\left(1+32\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right)\right)+\frac{8C^2t}{3\tilde{n}}.$$

Proof Let $A=\{f-f_0:f\in F\}$ and

$$Z=\sup_{f\in F}\left|\|f-f_0\|_{L^2(\tilde{P}_{\tilde{n}})}^2-\|f-f_0\|_{L^2(P)}^2\right|.$$

Then $A\subseteq L^\infty$ is separable by Lemma 32. Furthermore, $\|f-f_0\|_\infty\leq 2C$ for all $f\in F$. By Lemma 25 with n and P_n replaced by $\tilde{n}$ and $\tilde{P}_{\tilde{n}}$, we have

$$Z\leq\mathbb{E}(Z)+\left(\frac{32C^4t}{\tilde{n}}+\frac{16C^2\,\mathbb{E}(Z)t}{\tilde{n}}\right)^{1/2}+\frac{8C^2t}{3\tilde{n}}$$

with probability at least $1-e^{-t}$. By Lemma 22, we have

$$\mathbb{E}(Z)\leq\frac{64C^2}{\tilde{n}^{1/2}}\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right).$$

The result follows. ■

We move the bound on the distance between $V\hat{h}_{\hat{r}}$ and $V\hat{h}_{r_0}$ from the $L^2(\tilde{P}_{\tilde{n}})$ norm to the $L^2(P)$ norm for $r_0\in R$.

Corollary 30 *Assume (H) and ($\tilde{Y}$). Let $r_0\in R$ and $t\geq 1$. With probability at least $1-3e^{-t}$, we have*

$$\|V\hat{h}_{\hat{r}}-V\hat{h}_{r_0}\|_{L^2(P)}^2$$

is at most

$$\begin{aligned}
&\frac{10C(C+\tilde{\sigma})t^{1/2}}{\tilde{n}^{1/2}}\left(1+32\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right)\right) \\
&+\frac{24C^2t^{1/2}}{\tilde{n}^{1/2}}+\frac{8C^2t}{3\tilde{n}}+4\|V\hat{h}_{r_0}-g\|_{L^2(P)}^2.
\end{aligned}$$

Proof By Lemma 28, we have

$$\|V\hat{h}_{\hat{r}}-V\hat{h}_{r_0}\|_{L^2(\tilde{P}_{\tilde{n}})}^2$$

is at most

$$\frac{292C\tilde{\sigma}t^{1/2}}{\tilde{n}^{1/2}}\left(\left(2\log\left(1+\frac{\|k\|_\infty^2\rho^2}{8C^2}\right)\right)^{1/2}+\pi^{1/2}\right)+\frac{24C^2t^{1/2}}{\tilde{n}^{1/2}}+4\|V\hat{h}_{r_0}-g\|_{L^2(P)}^2$$

with probability at least $1-2e^{-t}$. Let $f_0=V\hat{h}_{r_0}$. Since $\hat{h}_{\hat{r}}\in F$, by Lemma 29 we have

$$\begin{aligned}
&\|V\hat{h}_{\hat{r}}-V\hat{h}_{r_0}\|_{L^2(P)}^2-\|V\hat{h}_{\hat{r}}-V\hat{h}_{r_0}\|_{L^2(\tilde{P}_{\tilde{n}})}^2 \\
&\leq\sup_{f\in F}\left|\|f-f_0\|_{L^2(\tilde{P}_{\tilde{n}})}^2-\|f-f_0\|_{L^2(P)}^2\right| \\
&\leq\frac{10C^2t^{1/2}}{\tilde{n}^{1/2}}\left(1+32\left(\left(2\log\left(2+\frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2}+\pi^{1/2}\right)\right)+\frac{8C^2t}{3\tilde{n}}
\end{aligned}$$

with probability at least $1 - e^{-t}$. The result follows. ■

We bound the distance between $V\hat{h}_{\hat{r}}$ and g in the $L^2(P)$ norm to prove Theorem 12.

Proof of Theorem 12 We have

$$\begin{aligned}\|V\hat{h}_{\hat{r}} - g\|_{L^2(P)}^2 &\leq \left(\|V\hat{h}_{\hat{r}} - Vh_{r_0}\|_{L^2(P)}^2 + \|Vh_{r_0} - g\|_{L^2(P)}^2\right)^2 \\ &\leq 2\|V\hat{h}_{\hat{r}} - Vh_{r_0}\|_{L^2(P)}^2 + 2\|Vh_{r_0} - g\|_{L^2(P)}^2.\end{aligned}$$

By Corollary 30, we have

$$\|V\hat{h}_{\hat{r}} - V\hat{h}_{r_0}\|_{L^2(P)}^2$$

is at most

$$\begin{aligned}&\frac{10C(C+\tilde{\sigma})t^{1/2}}{\tilde{n}^{1/2}}\left(1 + 32\left(\left(2\log\left(2 + \frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right)\right) \\ &+ \frac{24C^2t^{1/2}}{\tilde{n}^{1/2}} + \frac{8C^2t}{3\tilde{n}} + 4\|V\hat{h}_{r_0} - g\|_{L^2(P)}^2.\end{aligned}$$

with probability at least $1 - 3e^{-t}$. Hence,

$$\|V\hat{h}_{\hat{r}} - g\|_{L^2(P)}^2$$

is at most

$$\begin{aligned}&\frac{20C(C+\tilde{\sigma})t^{1/2}}{\tilde{n}^{1/2}}\left(1 + 32\left(\left(2\log\left(2 + \frac{\|k\|_\infty^2\rho^2}{C^2}\right)\right)^{1/2} + \pi^{1/2}\right)\right) \\ &+ \frac{48C^2t^{1/2}}{\tilde{n}^{1/2}} + \frac{16C^2t}{3\tilde{n}} + 10\|V\hat{h}_{r_0} - g\|_{L^2(P)}^2.\end{aligned}$$

The result follows. ■

We assume the conditions of Theorem 11 to prove Theorem 13.

Proof of Theorem 13 If we assume $(R1)$, then $r_0 = an^{(1-\beta)/(2(1+\beta))} \in R$ and

$$\|V\hat{h}_{r_0} - g\|_{L^2(P)}^2$$

is at most

$$\begin{aligned}&\left(D_3 + D_4a^{1/2}n^{(1-\beta)/(4(1+\beta))} + D_5an^{(1-\beta)/(2(1+\beta))}\right)t^{1/2}n^{-1/2} \\ &+ D_6tn^{-1} + D_7a^{-2\beta/(1-\beta)}n^{-\beta/(1+\beta)}\end{aligned}$$

with probability at least $1 - 3e^{-t}$ for some constants $D_3, D_4, D_5, D_6, D_7 > 0$ not depending on n, $\tilde{n}$ or t by Theorem 11. If we assume $(R2)$, then there is at least one $r_0 \in R$ such that

$$an^{(1-\beta)/(2(1+\beta))} \leq r_0 < an^{(1-\beta)/(2(1+\beta))} + b$$

and

$$\|V\hat{h}_{r_0} - g\|_{L^2(P)}^2$$

is at most

$$\begin{aligned}&\left(D_3 + D_4r_0^{1/2} + D_5r_0\right)t^{1/2}n^{-1/2} + D_6tn^{-1} + D_7r_0^{-2\beta/(1-\beta)} \\ \leq\ &\left(D_3 + D_4\left(a^{1/2}n^{(1-\beta)/(4(1+\beta))} + b^{1/2}\right) + D_5\left(an^{(1-\beta)/(2(1+\beta))} + b\right)\right)t^{1/2}n^{-1/2} \\ &+ D_6tn^{-1} + D_7a^{-2\beta/(1-\beta)}n^{-\beta/(1+\beta)}\end{aligned}$$

with probability at least $1 - 3e^{-t}$ by Theorem 11. In either case,

$$\|V\hat{h}_{r_0} - g\|_{L^2(P)}^2 \leq D_8t^{1/2}n^{-\beta/(1+\beta)} + D_9tn^{-1}$$

for some constants $D_8, D_9 > 0$ not depending on n, $\tilde{n}$ or t. By Theorem 12, we have

$$\|V\hat{h}_{\hat{r}} - g\|_{L^2(P)}^2 \leq D_{10}t^{1/2}\log(n)^{1/2}\tilde{n}^{-1/2} + D_{11}t\tilde{n}^{-1} + 10D_8t^{1/2}n^{-\beta/(1+\beta)} + 10D_9tn^{-1}$$

with probability at least $1 - 6e^{-t}$ for some constants $D_{10}, D_{11} > 0$ not depending on n, $\tilde{n}$ or t. Since $\tilde{n}$ increases at least linearly in n, there exists some constant $D_{12} > 0$ such that $\tilde{n} \geq D_{12}n$. We then have

$$\|V\hat{h}_{\hat{r}} - g\|_{L^2(P)}^2$$

is at most

$$\begin{aligned}&D_{12}^{-1/2}D_{10}t^{1/2}\log(n)^{1/2}n^{-1/2} + D_{12}^{-1}D_{11}tn^{-1} + 10D_8t^{1/2}n^{-\beta/(1+\beta)} + 10D_9tn^{-1} \\ \leq\ &D_1t^{1/2}n^{-\beta/(1+\beta)} + D_2tn^{-1}\end{aligned}$$

for some constants $D_1, D_2 > 0$ not depending on n, $\tilde{n}$ or t. ■

Appendix F. Estimator Calculation and Measurability

The following result is essentially Theorem 2.1 from Quintana and Rodríguez (2014). The authors show that that a strictly-positive-definite matrix which is a $(\mathbb{C}^{n\times n}, \mathcal{B}(\mathbb{C}^{n\times n}))$-valued measurable matrix on $(\Omega, \mathcal{F})$ can be diagonalised by an unitary matrix and a diagonal matrix which are both $(\mathbb{C}^{n\times n}, \mathcal{B}(\mathbb{C}^{n\times n}))$-valued measurable matrices on $(\Omega, \mathcal{F})$. The result holds for non-negative-definite matrices by adding the identity matrix before diagonalisation and subtracting it afterwards. Furthermore, the construction of the unitary matrix produces a matrix with real entries, which is to say an orthogonal matrix, when the strictly-positive-definite matrix has real entries.

Lemma 31 *Let M be a non-negative-definite matrix which is an $(\mathbb{R}^{n\times n}, \mathcal{B}(\mathbb{R}^{n\times n}))$-valued measurable matrix on $(\Omega, \mathcal{F})$. There exist an orthogonal matrix A and a diagonal matrix D which are both $(\mathbb{R}^{n\times n}, \mathcal{B}(\mathbb{R}^{n\times n}))$-valued measurable matrices on $(\Omega, \mathcal{F})$ such that $M = ADA^\mathsf{T}$.*

We prove Lemma 3.

Proof of Lemma 3 Let $H_n = \mathrm{sp}\{k_{X_i} : 1 \leq i \leq n\}$. The subspace H_n is closed in H, so there is an orthogonal projection $Q : H \to H_n$. Since $f - Qf \in H_n^\perp$ for all $f \in H$, we have

$$f(X_i) - (Qf)(X_i) = \langle f - Qf, k_{X_i}\rangle = 0$$

for all $1 \leq i \leq n$. Hence,

$$\inf_{f \in rB_H} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 = \inf_{f \in rB_H} \frac{1}{n} \sum_{i=1}^{n} ((Qf)(X_i) - Y_i)^2$$
$$= \inf_{f \in (rB_H) \cap H_n} \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2.$$

Let $f \in (rB_H) \cap H_n$ and write

$$f = \sum_{i=1}^{n} a_i k_{X_i}$$

for some $a \in \mathbb{R}^n$. Then

$$\frac{1}{n} \sum_{i=1}^{n} (f(X_i) - Y_i)^2 = n^{-1}(Ka - Y)^\mathsf{T}(Ka - Y)$$

and $\|f\|_H^2 = a^\mathsf{T} K a$, so we can write the norm constraint as $a^\mathsf{T} K a + s = r^2$, where $s \geq 0$ is a slack variable. The Lagrangian can be written as

$$L(a, s; \mu) = n^{-1}(Ka - Y)^\mathsf{T}(Ka - Y) + \mu(a^\mathsf{T} K a + s - r^2)$$
$$= a^\mathsf{T}(n^{-1}K^2 + \mu K)a - 2n^{-1}Y^\mathsf{T} K a + \mu s + n^{-1}Y^\mathsf{T} Y - \mu r^2,$$

where μ is the Lagrangian multiplier for the norm constraint. We seek to minimise the Lagrangian for a fixed value of μ. Note that we require $\mu \geq 0$ for the Lagrangian to have a finite minimum, due to the term in s. We have

$$\frac{\partial L}{\partial a} = 2(n^{-1}K^2 + \mu K)a - 2n^{-1}KY.$$

This being 0 is equivalent to $K((K + n\mu I)a - Y) = 0$.

Since the kernel k is a measurable function on $(S \times S, \mathcal{S} \otimes \mathcal{S})$ and the X_i are $(S, \mathcal{S})$-valued random variables on $(\Omega, \mathcal{F})$, we find that K is an $(\mathbb{R}^{n \times n}, \mathcal{B}(\mathbb{R}^{n \times n}))$-valued measurable matrix on $(\Omega, \mathcal{F})$. Furthermore, since the kernel k takes real values and is non-negative definite, K is non-negative definite with real entries. By Lemma 31, there exist an orthogonal matrix A and a diagonal matrix D which are both $(\mathbb{R}^{n \times n}, \mathcal{B}(\mathbb{R}^{n \times n}))$-valued measurable matrices on $(\Omega, \mathcal{F})$ such that $K = ADA^\mathsf{T}$. Note that the diagonal entries of D must be non-negative and we may assume that they are non-increasing. Inserting this diagonalisation into $K((K + n\mu I)a - Y) = 0$ gives

$$AD((D + n\mu I)A^\mathsf{T} a - A^\mathsf{T} Y) = 0.$$

Since A has the inverse A^T, this is equivalent to

$$D((D + n\mu I)A^\mathsf{T} a - A^\mathsf{T} Y) = 0.$$

This in turn is equivalent to

$$(A^\mathsf{T} a)_i = (D_{i,i} + n\mu)^{-1}(A^\mathsf{T} Y)_i$$

for $1 \leq i \leq m$. The same f is produced for all such a, because if w is the difference between two such a, then $(A^\mathsf{T} w)_i = 0$ for $1 \leq i \leq m$ and the squared H norm of

$$\sum_{i=1}^{n} w_i k_{X_i}$$

is $w^\mathsf{T} K w = w^\mathsf{T} A D A^\mathsf{T} w = 0$. Hence, we are free to set $(A^\mathsf{T} a)_i = 0$ for $m + 1 \leq i \leq n$. This uniquely defines $A^\mathsf{T} a$, which in turn uniquely defines a, since A^T has the inverse A. Note that this definition of a is measurable on $(\Omega \times [0, \infty), \mathcal{F} \otimes \mathcal{B}([0, \infty)))$, where μ varies in $[0, \infty)$.

We now search for a value of μ such that a and s satisfy the norm constraint. We call this value $\mu(r)$. There are two cases. If

$$r^2 < \sum_{i=1}^{m} D_{i,i}^{-1} (A^\mathsf{T} Y)_i^2,$$

then the a above and $s = 0$ minimise L for $\mu = \mu(r) > 0$ and satisfy the norm constraint, where $\mu(r)$ satisfies

$$\sum_{i=1}^{m} \frac{D_{i,i}}{(D_{i,i} + n\mu(r))^2} (A^\mathsf{T} Y)_i^2 = r^2.$$

Otherwise, the a above and

$$s = r^2 - \sum_{i=1}^{m} D_{i,i}^{-1} (A^\mathsf{T} Y)_i^2 \geq 0$$

minimise L for $\mu = \mu(r) = 0$ and satisfy the norm constraint. Hence, the Lagrangian sufficiency theorem shows

$$\hat{h}_r = \sum_{i=1}^{n} a_i k_{X_i}$$

for the a above with $\mu = \mu(r)$ for $r > 0$. We also have $\hat{h}_0 = 0$.

Since $\mu(r) > 0$ is strictly decreasing for

$$r^2 < \sum_{i=1}^{m} D_{i,i}^{-1} (A^\mathsf{T} Y)_i^2$$

and $\mu(r) = 0$ otherwise, we find

$$\{\mu(r) \leq \mu\} = \left\{ \sum_{i=1}^{m} \frac{D_{i,i}}{(D_{i,i} + n\mu)^2} (A^\mathsf{T} Y)_i^2 \leq r^2 \right\}$$

for $\mu \in [0, \infty)$. Therefore, $\mu(r)$ is measurable on $(\Omega \times [0, \infty), \mathcal{F} \otimes \mathcal{B}((0, \infty)))$, where r varies in $(0, \infty)$. Hence, the a above with $\mu = \mu(r)$ for $r > 0$ is measurable on $(\Omega \times [0, \infty), \mathcal{F} \otimes \mathcal{B}((0, \infty)))$, where r varies in $(0, \infty)$. By Lemma 4.25 of Steinwart and Christmann (2008),

the function $\Phi : S \to H$ by $\Phi(x) = k_x$ is a $(H, \mathcal{B}(H))$-valued measurable function on $(S, \mathcal{S})$. Hence, k_{X_i} for $1 \leq i \leq n$ are $(H, \mathcal{B}(H))$-valued random variables on $(\Omega, \mathcal{F})$. Together, these show that $\hat{h}_r$ is a $(H, \mathcal{B}(H))$-valued measurable function on $(\Omega \times [0, \infty), \mathcal{F} \otimes \mathcal{B}([0, \infty)))$, where r varies in $[0, \infty)$, recalling that $\hat{h}_0 = 0$. ■

We prove a continuity result about our estimator.

Lemma 32 *Let $r, s \in [0, \infty)$. We have $\|\hat{h}_r - \hat{h}_s\|_H^2 \leq |r^2 - s^2|$.*

Proof Recall the diagonalisation of $K = ADA^\mathsf{T}$ from Lemma 3. If $u, v \in \mathbb{R}^n$ and

$$h_1 = \sum_{i=1}^n u_i k_{X_i} \text{ and } h_2 = \sum_{i=1}^n v_i k_{X_i},$$

then $\langle h_1, h_2 \rangle_H = u^\mathsf{T} K v = (A^\mathsf{T} u)^\mathsf{T} D (A^\mathsf{T} v)$. Let $s > r$. If $r > 0$ then, by Lemma 3, we have

$$\begin{aligned}\langle \hat{h}_r, \hat{h}_s \rangle_H &= \sum_{i=1}^m \frac{D_{i,i}}{(D_{i,i} + n\mu(r))(D_{i,i} + n\mu(s))} (A^\mathsf{T} Y)_i^2 \\ &\geq \sum_{i=1}^m \frac{D_{i,i}}{(D_{i,i} + n\mu(r))^2} (A^\mathsf{T} Y)_i^2 \\ &= \|\hat{h}_r\|_H^2.\end{aligned}$$

Furthermore, again by Lemma 3, if $\mu(r) > 0$ then $\|\hat{h}_r\|_H^2 = r^2$ and

$$\begin{aligned}\|\hat{h}_r - \hat{h}_s\|_H^2 &= \|\hat{h}_r\|_H^2 + \|\hat{h}_s\|_H^2 - 2\langle \hat{h}_r, \hat{h}_s \rangle_H \\ &\leq \|\hat{h}_s\|_H^2 - \|\hat{h}_r\|_H^2 \\ &= \|\hat{h}_s\|_H^2 - r^2 \\ &\leq s^2 - r^2.\end{aligned}$$

Otherwise, $\mu(r) = 0$ and so $\mu(s) = 0$ by Lemma 3, which means $\hat{h}_r = \hat{h}_s$. If $r = 0$ then $\hat{h}_r = 0$ and $\|\hat{h}_r - \hat{h}_s\|_H^2 = \|\hat{h}_s\|_H^2 \leq s^2$. Hence, whenever $r < s$, we have $\|\hat{h}_r - \hat{h}_s\|_H^2 \leq s^2 - r^2$. The result follows. ■

We also have the estimator $\hat{r}$ when performing validation.

Lemma 33 *We have that $\hat{r}$ is a random variable on $(\Omega, \mathcal{F})$.*

Proof Let

$$W(s) = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} (V \hat{h}_s(\tilde{X}_i) - \tilde{Y}_i)^2$$

for $s \in R$. Note that $W(s)$ is a random variable on $(\Omega, \mathcal{F})$ and continuous in s by Lemma 32. Since $R \subseteq \mathbb{R}$, it is separable. Let R_0 be a countable dense subset of R. Then $\inf_{s \in R} W(s) = \inf_{s \in R_0} W(s)$ is a random variable on $(\Omega, \mathcal{F})$ as the right-hand side is the infimum of countably many random variables on $(\Omega, \mathcal{F})$. Let $r \in [0, \rho]$. By the definition of $\hat{r}$, we have

$$\{\hat{r} \leq r\} = \bigcup_{s \in R \cap [0,r]} \{W(s) \leq \inf_{t \in R} W(t)\}.$$

Since $R \cap [0, r] \subseteq \mathbb{R}$, it is separable. Let A_r be a countable dense subset of $R \cap [0, r]$. By the sequential compactness of $R \cap [0, r]$ and continuity of $W(s)$, we have

$$\{\hat{r} \leq r\} = \bigcap_{a=1}^{\infty} \bigcup_{s \in A_r} \{W(s) \leq \inf_{t \in R} W(t) + a^{-1}\}.$$

This set is an element of $\mathcal{F}$. ■

Appendix G. Subgaussian Random Variables

We need the definition of a sub-σ-algebra for the next result. The σ-algebra $\mathcal{G}$ is a sub-σ-algebra of the σ-algebra $\mathcal{F}$ if $\mathcal{G} \subseteq \mathcal{F}$. The following lemma relates a quadratic form of subgaussians to that of centred normal random variables.

Lemma 34 *Let ε_i for $1 \leq i \leq n$ be random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ which are independent conditional on some sub-σ-algebra $\mathcal{G} \subseteq \mathcal{F}$ and let*

$$\mathbb{E}(\exp(t\varepsilon_i)|\mathcal{G}) \leq \exp(\sigma^2 t^2/2)$$

almost surely for all $t \in \mathbb{R}$. Also, let δ_i for $1 \leq i \leq n$ be random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ which are independent of each other and $\mathcal{G}$ with $\delta_i \sim \mathrm{N}(0, \sigma^2)$. Let M be an $n \times n$ non-negative-definite matrix which is an $(\mathbb{R}^{n \times n}, \mathcal{B}(\mathbb{R}^{n \times n}))$-valued measurable matrix on $(\Omega, \mathcal{G})$. We have

$$\mathbb{E}(\exp(z\varepsilon^\mathsf{T} M \varepsilon)|\mathcal{G}) \leq \mathbb{E}(\exp(z\delta^\mathsf{T} M \delta)|\mathcal{G})$$

almost surely for all $z \geq 0$.

Proof This proof method uses techniques from the proof of Lemma 9 of Abbasi-Yadkori, Pál, and Szepesvári (2011). We have

$$\mathbb{E}(\exp(t_i \varepsilon_i/\sigma)|\mathcal{G}) \leq \exp(t_i^2/2)$$

almost surely for all $1 \leq i \leq n$ and $t_i \in \mathbb{R}$. Furthermore, the ε_i are independent conditional on $\mathcal{G}$, so

$$\mathbb{E}(\exp(t^\mathsf{T} \varepsilon/\sigma)|\mathcal{G}) \leq \exp(\|t\|_2^2/2)$$

almost surely for all $t \in \mathbb{R}^n$. By Lemma 31 with $\mathcal{F}$ replaced by $\mathcal{G}$, there exist an orthogonal matrix A and a diagonal matrix D which are both $(\mathbb{R}^{n \times n}, \mathcal{B}(\mathbb{R}^{n \times n}))$-valued measurable matrices on $(\Omega, \mathcal{G})$ such that $M = ADA^\mathsf{T}$. Hence, M has a square root $M^{1/2} = AD^{1/2}A^\mathsf{T}$ which is an $(\mathbb{R}^{n \times n}, \mathcal{B}(\mathbb{R}^{n \times n}))$-valued measurable matrix on $(\Omega, \mathcal{G})$, where $D^{1/2}$ is the diagonal matrix with entries equal to the square root of those of D. Note that these entries are non-negative because M is non-negative definite. We can then replace t with $sM^{1/2}u$ for $s \in \mathbb{R}$ and $u \in \mathbb{R}^n$ to get

$$\mathbb{E}(\exp(su^\mathsf{T} M^{1/2} \varepsilon/\sigma)|\mathcal{G}) \leq \exp(s^2 \|M^{1/2}u\|_2^2/2)$$

almost surely. Integrating over u with respect to the distribution of δ gives

$$\mathbb{E}(\exp(s\delta^\mathsf{T} M^{1/2}\varepsilon/\sigma)|\mathcal{G}) \leq \mathbb{E}(\exp(s^2\delta^\mathsf{T} M\delta/2)|\mathcal{G})$$

almost surely. The moment generating function of δ gives the left-hand side as being

$$\mathbb{E}(\exp(s^2\varepsilon^\mathsf{T} M\varepsilon/2)|\mathcal{G})$$

almost surely. The result follows. ■

Having established this relationship, we can now obtain a probability bound on a quadratic form of subgaussians by using Chernoff bounding. The following result is a conditional subgaussian version of the Hanson–Wright inequality.

Lemma 35 *Let ε_i for $1 \leq i \leq n$ be random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ which are independent conditional on some sub-σ-algebra $\mathcal{G} \subseteq \mathcal{F}$ and let*

$$\mathbb{E}(\exp(t\varepsilon_i)|\mathcal{G}) \leq \exp(\sigma^2 t^2/2)$$

almost surely for all $t \in \mathbb{R}$. Let M be an $n \times n$ non-negative-definite matrix which is an $(\mathbb{R}^{n\times n}, \mathcal{B}(\mathbb{R}^{n\times n}))$-valued measurable matrix on $(\Omega, \mathcal{G})$ and $z \geq 0$. We have

$$\varepsilon^\mathsf{T} M\varepsilon \leq \sigma^2 \operatorname{tr}(M) + 2\sigma^2\|M\|z + 2\sigma^2(\|M\|^2 z^2 + \operatorname{tr}(M^2)z)^{1/2}$$

with probability at least $1 - e^{-z}$ almost surely conditional on $\mathcal{G}$. Here, $\|M\|$ is the operator norm of M, which is a random variable on $(\Omega, \mathcal{G})$.

Proof This proof method follows that of Theorem 3.1.9 of Giné and Nickl (2016). By Lemma 31 with $\mathcal{F}$ replaced by $\mathcal{G}$, there exist an orthogonal matrix A and a diagonal matrix D which are both $(\mathbb{R}^{n\times n}, \mathcal{B}(\mathbb{R}^{n\times n}))$-valued measurable matrices on $(\Omega, \mathcal{G})$ such that $M = ADA^\mathsf{T}$. Let δ_i for $1 \leq i \leq n$ be random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ which are independent of each other and $\mathcal{G}$, with $\delta_i \sim \mathrm{N}(0, \sigma^2)$. By Lemma 34 and the fact that $A^\mathsf{T}\delta$ has the same distribution as δ, we have

$$\mathbb{E}(\exp(t\varepsilon^\mathsf{T} M\varepsilon)|\mathcal{G}) \leq \mathbb{E}(\exp(t\delta^\mathsf{T} M\delta)|\mathcal{G}) = \mathbb{E}(\exp(t\delta^\mathsf{T} D\delta)|\mathcal{G})$$

almost surely for all $t \geq 0$. Furthermore,

$$\mathbb{E}(\exp(t\delta_i^2/\sigma^2)) = \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{1/2}} \exp(tx^2 - x^2/2)dx = \frac{1}{(1-2t)^{1/2}}$$

for $0 \leq t < 1/2$ and $1 \leq i \leq n$, so

$$\mathbb{E}(\exp(t(\delta_i^2/\sigma^2 - 1))) = \exp(-(\log(1-2t) + 2t)/2).$$

We have

$$-2(\log(1-2t) + 2t) \leq \sum_{i=2}^{\infty} (2t)^i (2/i) \leq 4t^2/(1-2t)$$

for $0 \leq t < 1/2$. Therefore, since the δ_i are independent of $\mathcal{G}$, we have

$$\mathbb{E}(\exp(tD_{i,i}(\delta_i^2 - \sigma^2))|\mathcal{G}) \leq \exp\left(\frac{\sigma^4 D_{i,i}^2 t^2}{1 - 2\sigma^2 D_{i,i} t}\right)$$

almost surely for $0 \leq t < 1/(2\sigma^2 D_{i,i})$ and $1 \leq i \leq n$. Since the $D_{i,i}$ are random variables on $(\Omega, \mathcal{G})$ and the $D_{i,i}\delta_i$ for $1 \leq i \leq n$ are independent conditional on $\mathcal{G}$, we have

$$\mathbb{E}(\exp(t(\delta^\mathsf{T} D\delta - \sigma^2 \operatorname{tr}(D)))|\mathcal{G}) \leq \exp\left(\frac{\sigma^4 \operatorname{tr}(D^2)t^2}{1 - 2\sigma^2(\max_i D_{i,i})t}\right)$$

almost surely for $0 \leq t < 1/(2\sigma^2(\max_i D_{i,i}))$. Combining this with $\mathbb{E}(\exp(t\varepsilon^\mathsf{T} M\varepsilon)|\mathcal{G}) \leq \mathbb{E}(\exp(t\delta^\mathsf{T} D\delta)|\mathcal{G})$, we find

$$\mathbb{E}(\exp(t(\varepsilon^\mathsf{T} M\varepsilon - \sigma^2 \operatorname{tr}(M)))|\mathcal{G}) \leq \exp\left(\frac{\sigma^4 \operatorname{tr}(M^2)t^2}{1 - 2\sigma^2\|M\|t}\right)$$

almost surely for $0 \leq t < 1/(2\sigma^2\|M\|)$. By Chernoff bounding, we have

$$\varepsilon^\mathsf{T} M\varepsilon - \sigma^2 \operatorname{tr}(M) > s$$

for $s \geq 0$ with probability at most

$$\exp\left(\frac{\sigma^4 \operatorname{tr}(M^2)t^2}{1 - 2\sigma^2\|M\|t} - ts\right)$$

almost surely conditional on $\mathcal{G}$ for $0 \leq t < 1/(2\sigma^2\|M\|)$. Letting

$$t = \frac{s}{2\sigma^4 \operatorname{tr}(M^2) + 2\sigma^2\|M\|s}$$

gives the bound

$$\exp\left(-\frac{s^2}{4\sigma^4 \operatorname{tr}(M^2) + 4\sigma^2\|M\|s}\right).$$

Rearranging gives the result. ■

Appendix H. Covering Numbers

The following lemma gives a bound on the covering numbers of F.

Lemma 36 *Let $\varepsilon > 0$. We have*

$$N(F, \|\cdot\|_\infty, \varepsilon) \leq 1 + \frac{\|k\|_\infty^2 \rho^2}{2\varepsilon^2}.$$

Proof Let $a \geq 1$ and $r_i \in R$ and $f_i = V\hat{h}_{r_i} \in F$ for $1 \leq i \leq a$. Also, let $f = V\hat{h}_r \in F$ for $r \in R$. Since V is a contraction, we have $\|f - f_i\|_\infty \leq \varepsilon$ whenever $\|\hat{h}_r - \hat{h}_{r_i}\|_\infty \leq \varepsilon$. By Lemma 32, we have $\|\hat{h}_r - \hat{h}_{r_i}\|_\infty \leq \varepsilon$ whenever $|r^2 - r_i^2| \leq \varepsilon^2/\|k\|_\infty^2$. Hence, if we let $r_i^2 = \varepsilon^2(2i-1)/\|k\|_\infty^2$ and let ρ be such that

$$\rho^2 - \varepsilon^2(2a-1)/\|k\|_\infty^2 \leq \varepsilon^2/\|k\|_\infty^2,$$

then we find $N(F, \|\cdot\|_\infty, \varepsilon) \leq a$. Rearranging the above shows that we can choose

$$a = \left\lceil \frac{\|k\|_\infty^2 \rho^2}{2\varepsilon^2} \right\rceil$$

and the result follows. ■

We also calculate integrals of these covering numbers.

Lemma 37 *Let $a \geq 1$. We have*

$$\int_0^L (\log(aN(F, \|\cdot\|_\infty, \varepsilon)))^{1/2} d\varepsilon \leq \left(\log\left(\left(1 + \frac{\|k\|_\infty^2 \rho^2}{2L^2}\right) a\right)\right)^{1/2} L + \left(\frac{\pi}{2}\right)^{1/2} L$$

for $L \in (0, \infty)$. When $a = 1$, we have

$$\int_0^L (\log(N(F, \|\cdot\|_\infty, \varepsilon)))^{1/2} d\varepsilon \leq 2\left(\log\left(1 + \frac{\|k\|_\infty^2 \rho^2}{8C^2}\right)\right)^{1/2} C + (2\pi)^{1/2} C$$

for $L \in (0, \infty]$.

Proof Let $L \in (0, \infty)$. Then

$$\int_0^L (\log(aN(F, \|\cdot\|_\infty, \varepsilon)))^{1/2} d\varepsilon \leq \int_0^L \left(\log\left(a\left(1 + \frac{\|k\|_\infty^2 \rho^2}{2\varepsilon^2}\right)\right)\right)^{1/2} d\varepsilon$$

by Lemma 36. Changing variables to $u = \varepsilon/L$ gives

$$\begin{aligned} & L \int_0^1 \left(\log\left(a\left(1 + \frac{\|k\|_\infty^2 \rho^2}{2L^2 u^2}\right)\right)\right)^{1/2} du \\ \leq\ & L \int_0^1 \left(\log\left(a\left(1 + \frac{\|k\|_\infty^2 \rho^2}{2L^2}\right)\frac{1}{u^2}\right)\right)^{1/2} du \\ =\ & L \int_0^1 \left(\log\left(a\left(1 + \frac{\|k\|_\infty^2 \rho^2}{2L^2}\right)\right) + \log\left(\frac{1}{u^2}\right)\right)^{1/2} du. \end{aligned}$$

For $b, c \geq 0$ we have $(b + c)^{1/2} \leq b^{1/2} + c^{1/2}$, so the above is at most

$$\begin{aligned} & L \int_0^1 \left(\log\left(a\left(1 + \frac{\|k\|_\infty^2 \rho^2}{2L^2}\right)\right)\right)^{1/2} du + L \int_0^1 \left(\log\left(\frac{1}{u^2}\right)\right)^{1/2} du \\ =\ & L\left(\log\left(a\left(1 + \frac{\|k\|_\infty^2 \rho^2}{2L^2}\right)\right)\right)^{1/2} + L \int_0^1 \left(\log\left(\frac{1}{u^2}\right)\right)^{1/2} du. \end{aligned}$$

Changing variables to

$$s = \left(\log\left(\frac{1}{u^2}\right)\right)^{1/2}$$

shows

$$\begin{aligned} \int_0^1 \left(\log\left(\frac{1}{u^2}\right)\right)^{1/2} du &= \int_0^\infty s^2 \exp(-s^2/2) ds \\ &= \frac{1}{2}\int_{-\infty}^\infty s^2 \exp(-s^2/2) ds \\ &= \left(\frac{\pi}{2}\right)^{1/2}, \end{aligned}$$

since the last integral is a multiple of the variance of an $\mathrm{N}(0, 1)$ random variable. The first result follows. Note that $N(F, \|\cdot\|_\infty, \varepsilon) = 1$ whenever $\varepsilon \geq 2C$, as the ball of radius $2C$ about any point in F is the whole of F. Hence, when $a = 1$, we have

$$\begin{aligned} \int_0^L (\log(N(F, \|\cdot\|_\infty, \varepsilon)))^{1/2} d\varepsilon &\leq \int_0^\infty (\log(N(F, \|\cdot\|_\infty, \varepsilon)))^{1/2} d\varepsilon \\ &= \int_0^{2C} (\log(N(F, \|\cdot\|_\infty, \varepsilon)))^{1/2} d\varepsilon \\ &\leq 2\left(\log\left(1 + \frac{\|k\|_\infty^2 \rho^2}{8C^2}\right)\right)^{1/2} C + (2\pi)^{1/2} C \end{aligned}$$

for $L \in (0, \infty]$. ■

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2312–2320, 2011.

Jöran Bergh and Jörgen Löfström. *Interpolation Spaces. An Introduction.* Springer–Verlag, Berlin–New York, 1976.

Andrea Caponnetto and Ernesto de Vito. Optimal rates for the regularized least-squares algorithm. *Found. Comput. Math.*, 7(3):331–368, 2007.

Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithm. *arXiv preprint arXiv:1702.07254*, 2017.

Evarist Giné and Richard Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models.* Cambridge University Press, New York, 2016.

Shahar Mendelson and Joseph Neeman. Regularization in kernel learning. *Ann. Statist.*, 38 (1):526–565, 2010.

Luca Oneto, Sandro Ridella, and Davide Anguita. Tikhonov, Ivanov and Morozov regularization for support vector machine learning. *Mach. Learn.*, 103(1):103–136, 2016.

Yamilet Quintana and José M. Rodríguez. Measurable diagonalization of positive definite matrices. *J. Approx. Theory*, 185:91–97, 2014.

Steve Smale and Ding-Xuan Zhou. Estimating the approximation error in learning theory. *Anal. Appl. (Singap.)*, 1(1):17–41, 2003.

Steve Smale and Ding-Xuan Zhou. Learning theory estimates via integral operators and their approximations. *Constr. Approx.*, 26(2):153–172, 2007.

Bharath Sriperumbudur. On the optimal estimation of probability measures in weak and strong topologies. *Bernoulli*, 22(3):1839–1893, 2016.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer–Verlag, New York, 2008.

Ingo Steinwart and Clint Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constr. Approx.*, 35(3):363–417, 2012.

Ingo Steinwart, Don R. Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *Proceedings of the 22nd Conference on Learning Theory*, 2009.

Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer–Verlag, New York, 1996.

David Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, 1991.

Journal of Machine Learning Research 20 (2019) 1-39 Submitted 12/17; Revised 1/19; Published 7/19

Scaling Up Sparse Support Vector Machines by Simultaneous Feature and Sample Reduction

Bin Hong* HONGBINZJU@GMAIL.COM
State Key Lab of CAD&CG, College of Computer Science, Zhejiang University
Hangzhou, 310058, China

Weizhong Zhang* ZHANGWEIZHONGZJU@GMAIL.COM
Tencent AI Lab, Shenzhen, 518000, China
State Key Lab of CAD&CG, College of Computer Science, Zhejiang University
Hangzhou, 310058, China

Wei Liu WL2223@COLUMBIA.EDU
Tencent AI Lab, Shenzhen, 518000, China

Jieping Ye JPYE@UMICH.EDU
Department of Computational Medicine and Bioinformatics, University of Michigan
Ann Arbor, MI, 48104-2218, USA.

Deng Cai† DENGCAI@CAD.ZJU.EDU.CN
Xiaofei He XIAOFEIHE@CAD.ZJU.EDU.CN
State Key Lab of CAD&CG, College of Computer Science, Zhejiang University
Hangzhou, 310058, China

Jie Wang JIEWANGUSTC@GMAIL.COM
Department of Electronic Engineering and Information Science
University of Science and Technology of China, Hefei, 230026, China

Editor: Sanjiv Kumar

Abstract

Sparse support vector machine (SVM) is a popular classification technique that can simultaneously learn a small set of the most interpretable features and identify the support vectors. It has achieved great successes in many real-world applications. However, for large-scale problems involving a huge number of samples and ultra-high dimensional features, solving sparse SVMs remains challenging. By noting that sparse SVMs induce sparsities in both feature and sample spaces, we propose a novel approach, which is based on accurate estimations of the primal and dual optima of sparse SVMs, to simultaneously identify the inactive features and samples that are guaranteed to be irrelevant to the outputs. Thus, we can remove the identified inactive samples and features from the training phase, leading to substantial savings in the computational cost without sacrificing the accuracy. Moreover, we show that our method can be extended to multi-class sparse support vector machines. To the best of our knowledge, the proposed method is the *first static* feature and sample reduction method for sparse SVMs and multi-class sparse SVMs. Experiments on both synthetic and real data sets demonstrate that our approach significantly outperforms state-of-the-art methods and the speedup gained by our approach can be orders of magnitude.

Keywords: screening, SVM, dual problem, optimization, classification

∗. The first two authors contribute equally. †. Corresponding author.

©2019 Bin Hong, Weizhong Zhang, Wei Liu, Jieping Ye, Deng Cai, Xiaofei He and Jie Wang.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-723.html.

1. Introduction

Sparse support vector machine (SVM) (Bi et al., 2003; Wang et al., 2006) is a powerful technique that can simultaneously perform classification by margin maximization and variable selection via ℓ_1-norm penalty. The last few years have witnessed many successful applications of sparse SVMs, such as text mining (Joachims, 1998; Yoshikawa et al., 2014), bioinformatics (Narasimhan and Agarwal, 2013), and image processing (Mohr and Obermayer, 2004; Kotsia and Pitas, 2007). Many algorithms (Hastie et al., 2004; Fan et al., 2008; Catanzaro et al., 2008; Hsieh et al., 2008; Shalev-Shwartz et al., 2011) have been proposed to efficiently solve sparse SVM problems. However, the applications of sparse SVMs to large-scale learning problems, which involve a huge number of samples and extremely high-dimensional features, still remain challenging.

An emerging technique, called *screening* (El Ghaoui et al., 2012), has been shown to be promising in accelerating the training processes of large-scale sparse learning models. The essential idea of screening is to quickly identify the zero coefficients in the sparse solutions without solving any optimization problems such that the corresponding features or samples—that are called *inactive* features or samples—can be removed from the training phase. Then, we only need to perform optimization on the reduced data sets instead of the full data sets, leading to a substantial saving in the computational cost. Here, we need to emphasize that screening differs greatly from feature selection, although they look similar at the first glance. To be precise, screening is devoted to accelerating the training processes of many sparse models including Lasso, sparse SVMs, etc., while feature selection is the goal of these models. In the past few years, many screening methods are proposed for a large set of sparse learning techniques, such as Lasso (Tibshirani et al., 2012; Xiang and Ramadge, 2012; Wang et al., 2015), group Lasso (Ndiaye et al., 2016), ℓ_1-regularized logistic regression (Wang et al., 2014b), and SVMs (Ogawa et al., 2013). Most of the existing methods are in the same framework, i.e., estimating the optimum of the dual (resp. primal) problem and then developing the screening rules based on the estimations to infer which components of the primal (resp. dual) optimum are zero from the KKT conditions. The main differences among them are the techniques they use to develop their optima estimations and rules and the different sparse models they focus on. For example, SAFE (El Ghaoui et al., 2012) estimates the dual optimum by calculating the optimal value's lower bound of the dual problem. The Lasso screening method (Wang et al., 2015) estimates the optimum based on the non-expensiveness (Bauschke et al., 2011) of the projection operator by noting that its dual problem boils down to finding the projection of a given point on a convex set. Moreover, a screening method is called *static* if it triggers its screening rules before training, and *dynamic* if during the training process. Empirical studies indicate that screening methods can lead to orders of magnitude of speedup in computation time.

However, most existing screening methods study either feature screening or sample screening individually and their applications have very different scenarios. Specifically, to achieve better performance (say, in terms of speedup), we favor feature screening methods when the number of features p is much larger than the sample size n, while sample screening methods are preferable when $n \gg p$. Note that there is another class of sparse learning techniques, like sparse SVMs, which induce sparsities in both feature and sample spaces. All these screening methods tend to be helpless in accelerating the training process of these

models with large n and p. We cannot address this problem by simply combining the existing feature and sample screening methods either. The reason is that they could mistakenly discard relevant data as they are specifically designed for different sparse models. Recently, Shibagaki et al. (2016) considered this issue and proposed a method to simultaneously identify the inactive features and samples in a dynamic manner, and they trigger their testing rule when there is a sufficient decrease in the duality gap during the training process. Thus, the method in Shibagaki et al. 2016 can discard more and more inactive features and samples as the training proceeds and one has small-scale problems to solve in the late stage of the training process. Nevertheless, the overall speedup is limited as the problems' size can be large in the early stage of the training process. To be specific, the method in Shibagaki et al. 2016 depends heavily on the duality gap during the training process. The duality gap in the early stage can always be large, which makes the dual and primal estimations inaccurate and finally results in ineffective screening rules. Hence, it is essentially solving a large problem in the early stage. This will be verified in the experimental comparisons in Section 6 and similar results can also be found in the recent work (Massias et al., 2018), which merely focuses on Lasso.

In this paper, to address the limitations in the dynamic screening method, we propose a novel screening method that can **S**imultaneously identify **I**nactive **F**eatures and **S**amples (SIFS) for sparse SVMs in a static manner, and we only need to perform SIFS once *before* (instead of during) training. Thus, we only need to run the training algorithm on small-scale problems. The major technical challenge in developing SIFS is that we need to accurately estimate the primal and dual optima. The more accurate the estimations are, the more effective SIFS is in detecting inactive features and samples. Thus, our major technical contribution is a novel framework, which is based on the strong convexity of the primal and dual problems of sparse SVMs for deriving accurate estimations of the primal and dual optima (see Section 3). Another appealing feature of SIFS is the so-called *synergistic effect* (Shibagaki et al., 2016). Specifically, the proposed SIFS consists of two parts, i.e., **I**nactive **F**eature **S**creening (IFS) and **I**nactive **S**ample **S**creening (ISS). We show that discarding inactive features (resp. samples) identified by IFS (resp. ISS) leads to a more accurate estimation of the primal (resp. dual) optimum, which in turn dramatically enhances the capability of ISS (resp. IFS) in detecting inactive samples (resp. features). Thus, SIFS applies IFS and ISS in an alternating manner until no more inactive features and samples can be identified, leading to much better performance in scaling up large-scale problems than the application of ISS or IFS individually. Moreover, SIFS (see Section 4) is safe in the sense that the detected features and samples are guaranteed to be absent from the sparse representations. To the best of our knowledge, SIFS is the first static screening rule for sparse SVMs, which is able to simultaneously detect inactive features and samples. Experiments (see Section 6) on both synthetic and real data sets demonstrate that SIFS significantly outperforms the state-of-the-art (Shibagaki et al., 2016) in improving the training efficiency of sparse SVMs and the speedup can be orders of magnitude.

To demonstrate the flexibility of our proposed method SIFS, we extend its idea to multi-class sparse support vector machine (see Section 5), which also induces sparsities in both feature and sample spaces. Although multi-class sparse SVM has a very different structure and is more complex than sparse SVM, we will see that its dual problem has some similar properties (especially the strong convexity) with those of sparse SVM. Recall that

SIFS we developed for sparse SVMs is mainly based on the strong convexity of the primal and dual problems. Therefore, the idea of SIFS is also applicable for multi-class sparse SVMs. Experimental results show that the speedup gained by SIFS in multi-class sparse SVMs can also be orders of magnitude.

For the convenience of presentation, we postpone the detailed proofs of all the theorems in this paper to the appendix. At last, we should point out that this journal paper is an extension of our own previous work (Zhang et al., 2017) published at the International Conference on Machine Learning (ICML) 2017.

Notations: Let $\|\cdot\|_1$, $\|\cdot\|$, and $\|\cdot\|_\infty$ be ℓ_1, ℓ_2, and ℓ_∞ norms, respectively. We denote the inner product between vectors $\mathbf{x}$ and $\mathbf{y}$ by $\langle \mathbf{x}, \mathbf{y}\rangle$, and the i-th component of $\mathbf{x}$ by $[\mathbf{x}]_i$. Let $[p] = \{1, 2..., p\}$ for a positive integer p. Given a subset $\mathcal{J} := \{j_1, ..., j_k\}$ of $[p]$, let $|\mathcal{J}|$ be the cardinality of $\mathcal{J}$. For a vector $\mathbf{x}$, let $[\mathbf{x}]_{\mathcal{J}} = ([\mathbf{x}]_{j_1}, ..., [\mathbf{x}]_{j_k})^\top$. For a matrix $\mathbf{X}$, let $[\mathbf{X}]_{\mathcal{J}} = (\mathbf{x}_{j_1}, ..., \mathbf{x}_{j_k})$ and ${}_{\mathcal{J}}[\mathbf{X}] = ((\mathbf{x}^{j_1})^\top, ..., (\mathbf{x}^{j_k})^\top)^\top$, where $\mathbf{x}^i$ and $\mathbf{x}_j$ are the ith row and j^{th} column of $\mathbf{X}$, respectively. For a scalar t, we denote $\max\{0, t\}$ by $[t]_+$. Let $\mathbf{e}_k \in \mathbb{R}^K$ be the index vector, that is, $[\mathbf{e}_k]_i = 1$ if $i = k$, otherwise $[\mathbf{e}_k]_i = 0$. At last, we denote the set of nonnegative real numbers as $\mathbb{R}_+$.

2. Basics and Motivations

In this section, we briefly review some basics of sparse SVMs and then motivate SIFS via the KKT conditions. Specifically, we focus on an ℓ_1-regularized SVM with a smoothed hinge loss, which takes the form of

$$\min_{\mathbf{w}\in\mathbb{R}^p} P(\mathbf{w}; \alpha, \beta) = \frac{1}{n}\sum_{i=1}^n \ell(1 - \langle \bar{\mathbf{x}}_i, \mathbf{w}\rangle) + \frac{\alpha}{2}\|\mathbf{w}\|^2 + \beta||\mathbf{w}||_1, \tag{P*}$$

where $\mathbf{w} \in \mathbb{R}^p$ is the parameter vector to be estimated, $\{\mathbf{x}_i, y_i\}_{i=1}^n$ is the training set, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \{-1, +1\}$, $\bar{\mathbf{x}}_i = y_i\mathbf{x}_i$, α and β are two positive parameters, and $\ell(\cdot) : \mathbb{R} \to \mathbb{R}_+$ is the smoothed hinge loss, i.e.,

$$\ell(t) = \begin{cases} 0, & \text{if } t < 0, \\ \frac{t^2}{2\gamma}, & \text{if } 0 \le t \le \gamma, \\ t - \frac{\gamma}{2}, & \text{if } t > \gamma, \end{cases} \tag{1}$$

where $\gamma \in (0, 1)$.

Remark 1 *We use the smoothed hinge loss instead of the vanilla one in order to make the objective of the Lagrangian dual problem of (P*) strongly convex, which is needed in developing our accurate optima estimations. We should point out that the smoothed hinge loss is a pretty good approximation to the vanilla one with strong theoretical guarantees. See Section 5 in Shalev-Shwartz and Zhang 2016 for the details.*

We present the Lagrangian dual problem of problem (P*) and the KKT conditions in the following theorem, which play a fundamentally important role in developing our screening rules. We will not provide its proof in the appendix since it follows from Fenchel Duality (see Corollary 31.2.1 in Rockafellar, 1970) and a similar result can be found in Shibagaki et al. 2016.

Theorem 2 (Rockafellar, 1970) *Let $\bar{\mathbf{X}} = (\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, ..., \bar{\mathbf{x}}_n)$ and $\mathcal{S}_\beta(\cdot)$ be the soft-thresholding operator (Hastie et al., 2015), i.e., $[\mathcal{S}_\beta(\mathbf{u})]_i = \text{sign}([\mathbf{u}]_i)(|[\mathbf{u}]_i| - \beta)_+$. Then, for problem* (P*)*, the followings hold:*
(i) *the dual problem of* (P*) *is*

$$\min_{\theta \in [0,1]^n} D(\theta; \alpha, \beta) = \frac{1}{2\alpha} \left\| \mathcal{S}_\beta \left(\frac{1}{n} \bar{\mathbf{X}} \theta \right) \right\|^2 + \frac{\gamma}{2n} \|\theta\|^2 - \frac{1}{n} \langle \mathbf{1}, \theta \rangle, \tag{D*}$$

where $\mathbf{1} \in \mathbb{R}^n$ is a vector with all components equal to 1;
(ii) *denote the optima of* (P*) *and* (D*) *by $\mathbf{w}^*(\alpha, \beta)$ and $\theta^*(\alpha, \beta)$, respectively, then,*

$$\mathbf{w}^*(\alpha, \beta) = \frac{1}{\alpha} \mathcal{S}_\beta \left(\frac{1}{n} \bar{\mathbf{X}} \theta^*(\alpha, \beta) \right), \tag{KKT-1}$$

$$[\theta^*(\alpha, \beta)]_i = \begin{cases} 0, & \text{if } 1 - \langle \bar{\mathbf{x}}_i, \mathbf{w}^*(\alpha, \beta) \rangle < 0; \\ 1, & \text{if } 1 - \langle \bar{\mathbf{x}}_i, \mathbf{w}^*(\alpha, \beta) \rangle > \gamma; \\ \frac{1}{\gamma}(1 - \langle \bar{\mathbf{x}}_i, \mathbf{w}^*(\alpha, \beta) \rangle), & \text{otherwise.} \end{cases} \tag{KKT-2}$$

According to KKT-1 and KKT-2, we define 4 index sets:

$$\begin{aligned} \mathcal{F} &= \left\{ j \in [p] : \frac{1}{n} |[\bar{\mathbf{X}} \theta^*(\alpha, \beta)]_j| \leq \beta \right\}, \\ \mathcal{R} &= \{ i \in [n] : 1 - \langle \mathbf{w}^*(\alpha, \beta), \bar{\mathbf{x}}_i \rangle < 0 \}, \\ \mathcal{E} &= \{ i \in [n] : 1 - \langle \mathbf{w}^*(\alpha, \beta), \bar{\mathbf{x}}_i \rangle \in [0, \gamma] \}, \\ \mathcal{L} &= \{ i \in [n] : 1 - \langle \mathbf{w}^*(\alpha, \beta), \bar{\mathbf{x}}_i \rangle > \gamma \}, \end{aligned}$$

which imply that

$$\text{(i) } i \in \mathcal{F} \Rightarrow [\mathbf{w}^*(\alpha, \beta)]_i = 0, \text{(ii) } \begin{cases} i \in \mathcal{R} \Rightarrow & [\theta^*(\alpha, \beta)]_i = 0, \\ i \in \mathcal{L} \Rightarrow & [\theta^*(\alpha, \beta)]_i = 1. \end{cases} \tag{R}$$

Thus, we call the j-th feature *inactive* if $j \in \mathcal{F}$. The samples in $\mathcal{E}$ are the so-called support vectors and we call the samples in $\mathcal{R}$ and $\mathcal{L}$ *inactive* samples.

Suppose that we are given subsets of $\mathcal{F}$, $\mathcal{R}$, and $\mathcal{L}$. Then by the rules in (R), we can see that many coefficients of $\mathbf{w}^*(\alpha, \beta)$ and $\theta^*(\alpha, \beta)$ are known. Thus, we may have much fewer unknowns to solve and the problem size can be dramatically reduced. We formalize this idea in Lemma 3.

Lemma 3 *Given index sets $\hat{\mathcal{F}} \subseteq \mathcal{F}, \hat{\mathcal{R}} \subseteq \mathcal{R}$, and $\hat{\mathcal{L}} \subseteq \mathcal{L}$, the followings hold:*
(i) $[\mathbf{w}^*(\alpha, \beta)]_{\hat{\mathcal{F}}} = 0$, $[\theta^*(\alpha, \beta)]_{\hat{\mathcal{R}}} = 0$, $[\theta^*(\alpha, \beta)]_{\hat{\mathcal{L}}} = 1$;
(ii) *let $\hat{\mathcal{D}} = \hat{\mathcal{R}} \cup \hat{\mathcal{L}}$, $\hat{\mathbf{G}}_1 = {}_{\hat{\mathcal{F}}^c}[\bar{\mathbf{X}}]_{\hat{\mathcal{D}}^c}$, and $\hat{\mathbf{G}}_2 = {}_{\hat{\mathcal{F}}^c}[\bar{\mathbf{X}}]_{\hat{\mathcal{L}}}$, where $\hat{\mathcal{F}}^c = [p] \setminus \hat{\mathcal{F}}$, $\hat{\mathcal{D}}^c = [n] \setminus \hat{\mathcal{D}}$, and $\hat{\mathcal{L}}^c = [n] \setminus \hat{\mathcal{L}}$. Then, $[\theta^*(\alpha, \beta)]_{\hat{\mathcal{D}}^c}$ solves the following scaled dual problem:*

$$\min_{\hat{\theta} \in [0,1]^{|\hat{\mathcal{D}}^c|}} \left\{ \frac{1}{2\alpha} \left\| \mathcal{S}_\beta \left(\frac{1}{n} \hat{\mathbf{G}}_1 \hat{\theta} + \frac{1}{n} \hat{\mathbf{G}}_2 \mathbf{1} \right) \right\|^2 + \frac{\gamma}{2n} \|\hat{\theta}\|^2 - \frac{1}{n} \langle \mathbf{1}, \hat{\theta} \rangle \right\}; \tag{scaled-D*}$$

(iii) *suppose that $\theta^*(\alpha, \beta)$ is known, then,*

$$[\mathbf{w}^*(\alpha, \beta)]_{\hat{\mathcal{F}}^c} = \frac{1}{\alpha} \mathcal{S}_\beta \left(\frac{1}{n} {}_{\hat{\mathcal{F}}^c}[\bar{\mathbf{X}}] \theta^*(\alpha, \beta) \right).$$

Lemma 3 indicates that, if we can identify index sets $\hat{\mathcal{F}}$ and $\hat{\mathcal{D}}$ and the cardinalities of $\hat{\mathcal{F}}^c$ and $\hat{\mathcal{D}}^c$ are much smaller than the feature dimension p and the sample size n, we only need to solve problem (scaled-D*) that may be much *smaller* than problem (D*) to exactly recover the optima $\mathbf{w}^*(\alpha, \beta)$ and $\theta^*(\alpha, \beta)$ *without sacrificing any accuracy.*

However, we cannot directly apply Rules in (R) to identify subsets of $\mathcal{F}$, $\mathcal{R}$, and $\mathcal{L}$, as they require the knowledge of $\mathbf{w}^*(\alpha, \beta)$ and $\theta^*(\alpha, \beta)$ that are usually unavailable. Inspired by the idea in El Ghaoui et al. 2012, we can first estimate regions $\mathcal{W}$ and Θ that contain $\mathbf{w}^*(\alpha, \beta)$ and $\theta^*(\alpha, \beta)$, respectively. Then, by denoting

$$\hat{\mathcal{F}} := \left\{ j \in [p] : \max_{\theta \in \Theta} \left\{ \left| \frac{1}{n} [\bar{\mathbf{X}} \theta]_j \right| \right\} \leq \beta \right\}, \tag{2}$$

$$\hat{\mathcal{R}} := \left\{ i \in [n] : \max_{\mathbf{w} \in \mathcal{W}} \{ 1 - \langle \mathbf{w}, \bar{\mathbf{x}}_i \rangle \} < 0 \right\}, \tag{3}$$

$$\hat{\mathcal{L}} := \left\{ i \in [n] : \min_{\mathbf{w} \in \mathcal{W}} \{ 1 - \langle \mathbf{w}, \bar{\mathbf{x}}_i \rangle \} > \gamma \right\}, \tag{4}$$

since it is easy to know that $\hat{\mathcal{F}} \subset \mathcal{F}, \hat{\mathcal{R}} \subset \mathcal{R}$, and $\hat{\mathcal{L}} \subset \mathcal{L}$, the rules in (R) can be relaxed as:

$$\text{(i) } j \in \hat{\mathcal{F}} \Rightarrow [\mathbf{w}^*(\alpha, \beta)]_j = 0, \tag{R1}$$

$$\text{(ii) } \begin{cases} i \in \hat{\mathcal{R}} \Rightarrow & [\theta^*(\alpha, \beta)]_i = 0, \\ i \in \hat{\mathcal{L}} \Rightarrow & [\theta^*(\alpha, \beta)]_i = 1. \end{cases} \tag{R2}$$

In view of Rules R1 and R2, we sketch the development of SIFS as follows.

Step 1: Derive estimations $\mathcal{W}$ and Θ such that $\mathbf{w}^*(\alpha, \beta) \in \mathcal{W}$ and $\theta^*(\alpha, \beta) \in \Theta$, respectively.

Step 2: Develop SIFS by deriving the relaxed screening rules R1 and R2, i.e., by solving the optimization problems in Eqs. (2), (3) and (4).

3. Estimate the Primal and Dual Optima

In this section, we first show that the primal and dual optima admit closed-form solutions for specific values of α and β (Section 3.1). Then, in Sections 3.2 and 3.3, we present accurate estimations of the primal and dual optima, respectively. We would like to point out that in order to extend the optimum estimation results below to multi-class sparse SVM and avoid redundancy, we consider the optimum estimations for the primal and dual problems in more general forms.

3.1. Effective Intervals of Parameters α and β

Below we show two things. One is that if the value of β is sufficiently large, no matter what α is, the primal solution is 0. The other is for any β, if α is large enough, the primal and dual optima admit closed-form solutions.

Lemma 4 *Let $\beta_{\max} = \|\frac{1}{n} \bar{\mathbf{X}} \mathbf{1}\|_\infty$ and $\alpha_{\max}(\beta) = \frac{1}{1-\gamma} \max_{i \in [n]} \left\{ \langle \bar{\mathbf{x}}_i, \mathcal{S}_\beta(\frac{1}{n} \bar{\mathbf{X}} \mathbf{1}) \rangle \right\}$. Then,*
(i) *for $\alpha > 0$ and $\beta \geq \beta_{\max}$, we have*

$$\mathbf{w}^*(\alpha, \beta) = \mathbf{0}, \quad \theta^*(\alpha, \beta) = \mathbf{1};$$

(ii) *for all* $\alpha \in [\max\{\alpha_{\max}(\beta), 0\}, \infty) \cap (0, \infty)$, *we have*

$$\mathbf{w}^*(\alpha, \beta) = \frac{1}{\alpha}\mathcal{S}_\beta\left(\frac{1}{n}\bar{\mathbf{X}}\mathbf{1}\right), \quad \theta^*(\alpha, \beta) = \mathbf{1}. \tag{5}$$

By Lemma 4, we only need to consider the cases where $\beta \in (0, \beta_{\max}]$ and $\alpha \in (0, \alpha_{\max}(\beta)]$.

3.2. Primal Optimum Estimation

In Section 1, we mention that the proposed SIFS consists of IFS and ISS, which can identify the inactive features and samples, respectively. We also mention that an alternating application of IFS and ISS can improve the estimation accuracy of the primal and dual optimum estimations, which can in turn make ISS and IFS more effective in identifying inactive samples and features, respectively. We would like to show that discarding inactive features by IFS leads to a more accurate estimation of the primal optimum.

We consider the following general problem (g-P*):

$$\min_{\mathbf{w}\in\mathbb{R}^p} P(\mathbf{w}; \alpha, \beta) = L(\mathbf{w}) + \frac{\alpha}{2}\|\mathbf{w}\|^2 + \beta\|\mathbf{w}\|_1, \tag{g-P*}$$

where $L(\mathbf{w}) : \mathbb{R}^p \to \mathbb{R}_+$ is smooth and convex. Exploiting the strong convexity of the objective function, we obtain the optimum estimation in the following lemma.

Lemma 5 *Suppose that the optimum* $\mathbf{w}^*(\alpha_0, \beta_0)$ *of problem* (g-P*) *at* (α_0, β_0) *with* $\beta_0 \in (0, \beta_{\max}]$ *and* $\alpha_0 \in (0, \alpha_{\max}(\beta_0)]$ *is known. Consider problem* (g-P*) *with parameters* $\alpha > 0$ *and* β_0. *Let* $\hat{\mathcal{F}}$ *be the index set satisfying* $[\mathbf{w}^*(\alpha, \beta_0)]_{\hat{\mathcal{F}}} = \mathbf{0}$ *and define*

$$\mathbf{c} = \frac{\alpha_0 + \alpha}{2\alpha}[\mathbf{w}^*(\alpha_0, \beta_0)]_{\hat{\mathcal{F}}^c}, \tag{6}$$

$$r^2 = \frac{(\alpha_0 - \alpha)^2}{4\alpha^2}\|\mathbf{w}^*(\alpha_0, \beta_0)\|^2 - \frac{(\alpha_0 + \alpha)^2}{4\alpha^2}\|[\mathbf{w}^*(\alpha_0, \beta_0)]_{\hat{\mathcal{F}}}\|^2. \tag{7}$$

Then, the following holds:

$$[\mathbf{w}^*(\alpha, \beta_0)]_{\hat{\mathcal{F}}^c} \in \{\mathbf{w} : \|\mathbf{w} - \mathbf{c}\| \le r\}.$$

For problem (P*), let $\hat{\mathcal{F}}$ be the index set of inactive features identified by previous IFS steps. We have $[\mathbf{w}^*(\alpha, \beta_0)]_{\hat{\mathcal{F}}} = \mathbf{0}$. Hence, we only need to find an estimation for $[\mathbf{w}^*(\alpha, \beta_0)]_{\hat{\mathcal{F}}^c}$. Since problem (P*) is a special case of problem (g-P*), given the reference solution $\mathbf{w}^*(\alpha_0, \beta_0)$ and the set $\hat{\mathcal{F}}$, from Lemma 5 we have:

$$[\mathbf{w}^*(\alpha, \beta_0)]_{\hat{\mathcal{F}}^c} \in \mathcal{W} := \{\mathbf{w} : \|\mathbf{w} - \mathbf{c}\| \le r\}, \tag{8}$$

where $\mathbf{c}$ and $\mathbf{r}$ are defined in Eqs. (6) and (7), respectively.

It shows that $[\mathbf{w}^*(\alpha, \beta_0)]_{\hat{\mathcal{F}}^c}$ lies in a ball of radius r centered at $\mathbf{c}$. Note that, before we perform IFS, the set $\hat{\mathcal{F}}$ is empty and the second term on the right hand side of Eq. (7) is thus 0. If we apply IFS multiple times alternatively with ISS, the set $\hat{\mathcal{F}}$ will be monotonically increasing. Thus, Eq. (7) implies that the radius will be monotonically decreasing, leading to a more accurate primal optimum estimation.

3.3. Dual Optimum Estimation

We consider a general dual problem (g-D*) below:

$$\min_{\theta\in[0,1]^n} D(\theta; \alpha, \beta) = \frac{1}{2\alpha}f_\beta(\theta) + \frac{\gamma}{2n}\|\theta\|^2 - \frac{1}{n}\langle \mathbf{v}, \theta\rangle, \tag{g-D*}$$

where $\mathbf{v} \in \mathbb{R}^n$ and $f_\beta(\theta) : \mathbb{R}^n \to \mathbb{R}_+$ is smooth and convex. It is obvious that problem (g-D*) can be reduced to problem (D*) by letting $f_\beta(\theta) = \left\|\mathcal{S}_\beta\left(\frac{1}{n}\bar{\mathbf{X}}\theta\right)\right\|^2$ and $\mathbf{v} = \mathbf{1}$. The lemma below gives an optimum estimation of problem (g-D*) based on the strong convexity of its objective function.

Lemma 6 *Suppose that the optimum* $\theta^*(\alpha_0, \beta_0)$ *of problem* (g-D*) *with* $\beta_0 \in (0, \beta_{\max}]$ *and* $\alpha_0 \in (0, \alpha_{\max}(\beta_0)]$ *is known. Consider problem* (g-D*) *at* (α, β_0) *with* $\alpha > 0$ *and let* $\hat{\mathcal{R}}$ *and* $\hat{\mathcal{L}}$ *be two index sets satisfying* $[\theta^*(\alpha, \beta_0)]_{\hat{\mathcal{R}}} = \mathbf{0}$ *and* $[\theta^*(\alpha, \beta_0)]_{\hat{\mathcal{L}}} = \mathbf{1}$, *respectively. We denote* $\hat{\mathcal{D}} = \hat{\mathcal{R}} \cup \hat{\mathcal{L}}$ *and define*

$$\mathbf{c} = \frac{\alpha - \alpha_0}{2\gamma\alpha}[\mathbf{v}]_{\hat{\mathcal{D}}^c} + \frac{\alpha_0 + \alpha}{2\alpha}[\theta^*(\alpha_0, \beta_0)]_{\hat{\mathcal{D}}^c}, \tag{9}$$

$$\begin{aligned} r^2 =& (\frac{\alpha - \alpha_0}{2\alpha})^2||\theta^*(\alpha_0, \beta_0) - \frac{1}{\gamma}\mathbf{v}||^2 - ||\mathbf{1} - \frac{\alpha - \alpha_0}{2\gamma\alpha}[\mathbf{v}]_{\hat{\mathcal{L}}} - \frac{\alpha_0 + \alpha}{2\alpha}[\theta^*(\alpha_0, \beta_0)]_{\hat{\mathcal{L}}}||^2 \\ &- ||\frac{\alpha - \alpha_0}{2\gamma\alpha}[\mathbf{v}]_{\hat{\mathcal{R}}} + \frac{\alpha_0 + \alpha}{2\alpha}[\theta^*(\alpha_0, \beta_0)]_{\hat{\mathcal{R}}}||^2. \end{aligned} \tag{10}$$

Then, the following holds:

$$[\theta^*(\alpha, \beta_0)]_{\hat{\mathcal{D}}^c} \in \{\theta : \|\theta - \mathbf{c}\| \le r\}.$$

We now turn back to problem (D*). As it is a special case of problem (g-D*), given the reference solution $\theta^*(\alpha_0, \beta_0)$ at (α_0, β_0) and the index sets of inactive samples identified by the previous ISS steps $\hat{\mathcal{R}}$ and $\hat{\mathcal{L}}$, using Lemma 6, we can obtain:

$$[\theta^*(\alpha, \beta_0)]_{\hat{\mathcal{D}}^c} \in \Theta := \{\theta : \|\theta - \mathbf{c}\| \le r\}, \tag{11}$$

where $\mathbf{c}$ and $\mathbf{r}$ are defined by Eqs. (9) and (10) with $\mathbf{v} = \mathbf{1}$, respectively. Therefore, $[\theta^*(\alpha, \beta_0)]_{\hat{\mathcal{D}}^c}$ lies in the ball Θ. In view of Eq. (10), the index sets $\hat{\mathcal{L}}$ and $\hat{\mathcal{R}}$ monotonically increase and hence the last two terms on the right hand side of Eq. (10) monotonically increase when we perform ISS multiple times (alternating with IFS), which implies that the ISS steps can reduce the radius and thus improve the dual optimum estimation.

In addition, from both Lemmas 5 and 6, we can see that the radii of $\mathcal{W}$ and Θ can be potentially large when α is very small, which may affect our estimation accuracy. We can sidestep this issue by letting the ratio α/α_0 be a constant. That is why we space the values of α equally at the logarithmic scale on the parameter value path in the experiments.

Remark 7 *To estimate the optima* $\mathbf{w}^*(\alpha, \beta_0)$ *and* $\theta^*(\alpha, \beta_0)$ *of problems* (P*) *and* (D*) *using Lemmas 5 and 6, we have a free reference solution pair* $\mathbf{w}^*(\alpha_0, \beta_0)$ *and* $\theta^*(\alpha_0, \beta_0)$ *with* $\alpha_0 = \alpha_{\max}(\beta_0)$. *The reason is that* $\mathbf{w}^*(\alpha_0, \beta_0)$ *and* $\theta^*(\alpha_0, \beta_0)$ *admit closed-form solutions in this setting (see Lemma 4).*

4. The Proposed SIFS Screening Rule

We first present the IFS and ISS rules in Sections 4.1 and 4.2, respectively. Then, in Section 4.3, we develop the SIFS screening rule by an alternating application of IFS and ISS.

4.1. Inactive Feature Screening (IFS)

Suppose that $\mathbf{w}^*(\alpha_0, \beta_0)$ and $\theta^*(\alpha_0, \beta_0)$ are known, we derive IFS to identify inactive features for problem (P*) at (α, β_0) by solving the optimization problem in Eq. (2) (see Section A.5 in the appendix):

$$s^i(\alpha, \beta_0) = \max_{\theta \in \Theta} \left\{ \frac{1}{n} |\langle [\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c}, \theta \rangle + \langle [\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}}, \mathbf{1} \rangle| \right\}, i \in \hat{\mathcal{F}}^c, \tag{12}$$

where Θ is given by Eq. (11) and $\hat{\mathcal{F}}$ and $\hat{\mathcal{D}} = \hat{\mathcal{R}} \cup \hat{\mathcal{L}}$ are the index sets of inactive features and samples that have been identified in previous screening processes, respectively. The next result shows the closed-form solution of problem (12).

Lemma 8 *Consider problem* (12). *Let* $\mathbf{c}$ *and* r *be given by Eq. (9) and Eq. (10) with* $\mathbf{v} = \mathbf{1}$. *Then, for all* $i \in \hat{\mathcal{F}}^c$, *we have*

$$s^i(\alpha, \beta_0) = \frac{1}{n} (|\langle [\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c}, \mathbf{c} \rangle + \langle [\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}}, \mathbf{1} \rangle| + \|[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c}\| r).$$

We are now ready to present the IFS rule.

Theorem 9 [Feature screening rule IFS] *Consider problem* (P*). *We suppose that* $\mathbf{w}^*(\alpha_0, \beta_0)$ *and* $\theta^*(\alpha_0, \beta_0)$ *are known. Then:*

(1) *the feature screening rule* IFS *takes the form of*

$$s^i(\alpha, \beta_0) \leq \beta_0 \Rightarrow [\mathbf{w}^*(\alpha, \beta_0)]_i = 0, \forall i \in \hat{\mathcal{F}}^c; \tag{IFS}$$

(2) *we can update the index set* $\hat{\mathcal{F}}$ *by*

$$\hat{\mathcal{F}} \leftarrow \hat{\mathcal{F}} \cup \Delta\hat{\mathcal{F}} \text{ with } \Delta\hat{\mathcal{F}} = \{i : s^i \leq \beta_0, i \in \hat{\mathcal{F}}^c\}. \tag{13}$$

Recall that (see Lemma 6) previous sample screening results give us a tighter dual estimation, i.e., a smaller feasible region Θ for problem (12), which results in a smaller $s^i(\alpha, \beta_0)$. It finally brings about a more powerful feature screening rule IFS. This is the so called synergistic effect.

4.2. Inactive Sample Screening (ISS)

Similar to IFS, we derive ISS to identify inactive samples by solving the optimization problems in Eq. (3) and Eq. (4) (see Section A.7 in the appendix for details):

$$u_i(\alpha, \beta_0) = \max_{\mathbf{w} \in \mathcal{W}} \{1 - \langle [\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \mathbf{w} \rangle\}, i \in \hat{\mathcal{D}}^c, \tag{14}$$

$$l_i(\alpha, \beta_0) = \min_{\mathbf{w} \in \mathcal{W}} \{1 - \langle [\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \mathbf{w} \rangle\}, i \in \hat{\mathcal{D}}^c, \tag{15}$$

where $\mathcal{W}$ is given by Eq. (8) and $\hat{\mathcal{F}}$ and $\hat{\mathcal{D}} = \hat{\mathcal{R}} \cup \hat{\mathcal{L}}$ are the index sets of inactive features and samples that have been identified in previous screening processes. We show that problems (14) and (15) admit closed-form solutions.

Lemma 10 *Consider problems* (14) *and* (15). *Let* $\mathbf{c}$ *and* r *be given by Eq. (6) and Eq. (7). Then,*

$$u_i(\alpha, \beta_0) = 1 - \langle [\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \mathbf{c} \rangle + \|[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}\| r, i \in \hat{\mathcal{D}}^c,$$
$$l_i(\alpha, \beta_0) = 1 - \langle [\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \mathbf{c} \rangle - \|[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}\| r, i \in \hat{\mathcal{D}}^c.$$

We are now ready to present the ISS rule.

Theorem 11 [Sample screening rule ISS] *Consider problem (D*) and suppose that* $\mathbf{w}^*(\alpha_0, \beta_0)$ *and* $\theta^*(\alpha_0, \beta_0)$ *are known, then:*

(1) *the sample screening rule* ISS *takes the form of*

$$\begin{array}{l} u_i(\alpha, \beta_0) < 0 \Rightarrow [\theta^*(\alpha, \beta_0)]_i = 0, \\ l_i(\alpha, \beta_0) > \gamma \Rightarrow [\theta^*(\alpha, \beta_0)]_i = 1, \end{array} \forall i \in \hat{\mathcal{D}}^c; \tag{ISS}$$

(2) *we can update the index sets* $\hat{\mathcal{R}}$ *and* $\hat{\mathcal{L}}$ *by*

$$\hat{\mathcal{R}} \leftarrow \hat{\mathcal{R}} \cup \Delta\hat{\mathcal{R}} \text{ with } \Delta\hat{\mathcal{R}} = \{i : u_i(\alpha, \beta_0) < 0, i \in \hat{\mathcal{D}}^c\}, \tag{16}$$

$$\hat{\mathcal{L}} \leftarrow \hat{\mathcal{L}} \cup \Delta\hat{\mathcal{L}} \text{ with } \Delta\hat{\mathcal{L}} = \{i : l_i(\alpha, \beta_0) > \gamma, i \in \hat{\mathcal{D}}^c\}. \tag{17}$$

The synergistic effect also exists here. Recall that (see Lemma 5), previous feature screening results lead to a smaller feasible region $\mathcal{W}$ for problems (14) and (15), which results in smaller $u_i(\alpha, \beta_0)$ and larger $l_i(\alpha, \beta_0)$. It finally leads us to a more accurate sample screening rule ISS.

4.3. The Proposed SIFS Rule by An Alternating Application of IFS and ISS

In real applications, the optimal parameter values of α and β are usually unknown. To determine appropriate parameter values, common approaches, like cross validation and stability selection, need to solve the model over a grid of parameter values $\{(\alpha_{i,j}, \beta_j) : i \in [M], j \in [N]\}$ with $\beta_{\max} > \beta_1 > ... > \beta_N > 0$ and $\alpha_{\max}(\beta_j) > \alpha_{1,j} > ... > \alpha_{M,j} > 0$. This can be very time-consuming. Inspired by Strong Rule (Tibshirani et al., 2012) and SAFE (El Ghaoui et al., 2012), we develop a sequential version of SIFS in Algorithm 1. Specifically, given the primal and dual optima $\mathbf{w}^*(\alpha_{i-1,j}, \beta_j)$ and $\theta^*(\alpha_{i-1,j}, \beta_j)$ at $(\alpha_{i-1,j}, \beta_j)$, we apply SIFS to identify the inactive features and samples for problem (P*) at $(\alpha_{i,j}, \beta_j)$. Then, we perform training on the reduced data set and solve the primal and dual optima at $(\alpha_{i,j}, \beta_j)$. We repeat this process until we solve problem (P*) at all pairs of parameter values.

Note that we insert $\alpha_{0,j}$ into every sequence $\{\alpha_{i,j} : i \in [M]\}$ (see line 1 in Algorithm 1) to obtain a closed-form solution as the first reference solution. In this way, we can avoid solving problem at $(\alpha_{1,j}, \beta_j), j \in [N]$ directly (without screening), which is time consuming. At last, we would like to point out that the values $\{(\alpha_{i,j}, \beta_j) : i \in [M], j \in [N]\}$ in SIFS can be specified by users arbitrarily.

Algorithm 1 SIFS

1: **Input:** $\beta_{\max} > \beta_1 > ... > \beta_N > 0$ and $\alpha_{\max}(\beta_j) = \alpha_{0,j} > \alpha_{1,j} > ... > \alpha_{M,j} > 0$.
2: **for** $j = 1$ to N **do**
3: Compute the first reference solution $\mathbf{w}^*(\alpha_{0,j}, \beta_j)$ and $\theta^*(\alpha_{0,j}, \beta_j)$ using the close-form formulas in Eq. (5).
4: **for** $i = 1$ to M **do**
5: **Initialization:** $\hat{\mathcal{F}} = \hat{\mathcal{R}} = \hat{\mathcal{L}} = \emptyset$.
6: **repeat**
7: Run sample screening using rule ISS based on $\mathbf{w}^*(\alpha_{i-1,j}, \beta_j)$.
8: Update the inactive sample sets $\hat{\mathcal{R}}$ and $\hat{\mathcal{L}}$:
$\hat{\mathcal{R}} \leftarrow \hat{\mathcal{R}} \cup \Delta\hat{\mathcal{R}}$ and $\hat{\mathcal{L}} \leftarrow \hat{\mathcal{L}} \cup \Delta\hat{\mathcal{L}}$.
9: Run feature screening using rule IFS based on $\theta^*(\alpha_{i-1,j}, \beta_j)$.
10: Update the inactive feature set $\hat{\mathcal{F}}$:
$\hat{\mathcal{F}} \leftarrow \hat{\mathcal{F}} \cup \Delta\hat{\mathcal{F}}$.
11: **until** No new inactive features or samples are identified
12: Compute $\mathbf{w}^*(\alpha_{i,j}, \beta_j)$ and $\theta^*(\alpha_{i,j}, \beta_j)$ by solving the scaled problem.
13: **end for**
14: **end for**
15: **Output:** $\mathbf{w}^*(\alpha_{i,j}, \beta_j)$ and $\theta^*(\alpha_{i,j}, \beta_j), i \in [M], j \in [N]$.

SIFS applies ISS and IFS in an alternating manner to reinforce their capability in identifying inactive samples and features. In Algorithm 1, we apply ISS first. Of course, we can also apply IFS first. The theorem below demonstrates that the orders have no impact on the performance of SIFS.

Theorem 12 *Given the optimal solutions $\mathbf{w}^*(\alpha_{i-1,j}, \beta_j)$ and $\theta^*(\alpha_{i-1,j}, \beta_j)$ at $(\alpha_{i-1,j}, \beta_j)$ as the reference solution pair at $(\alpha_{i,j}, \beta_j)$ for SIFS, we assume SIFS with ISS first stops after applying IFS and ISS for s times and denote the identified inactive features and samples as $\hat{\mathcal{F}}_s^A, \hat{\mathcal{R}}_s^A$, and $\hat{\mathcal{L}}_s^A$. Similarly, when we apply IFS first, the results are denoted as $\hat{\mathcal{F}}_t^B, \hat{\mathcal{R}}_t^B$, and $\hat{\mathcal{L}}_t^B$. Then, the followings hold:*
(1) $\hat{\mathcal{F}}_s^A = \hat{\mathcal{F}}_t^B, \hat{\mathcal{R}}_s^A = \hat{\mathcal{R}}_t^B$, *and* $\hat{\mathcal{L}}_s^A = \hat{\mathcal{L}}_t^B$.
(2) *with different orders of applying ISS and IFS, the difference between the times of ISS and IFS we need to apply in SIFS can never be larger than 1, that is,* $|s - t| \leq 1$.

4.4. Discussion

After developing the proposed method SIFS above, we now turn to the related work discussion in order to highlight the differences between SIFS and the existing methods and also the novelty of our method, although we have mentioned some of them in the introduction section. We divide the previous work into two parts: screening for sparse SVM and for other sparse learning models.

To the best of our knowledge, the screening method in Shibagaki et al. 2016 is the only method besides our SIFS, which can simultaneously identify the inactive features and samples for sparse SVM. There are mainly three big differences between SIFS and this method. First, the techniques used in the optima estimations of SIFS and Shibagaki et al. 2016 are quite different. To be specific, the estimations in SIFS are developed by exploiting the reference solution pair and carefully studying the strong convexity of the objective functions and the optimum conditions of problems (P*) and (D*) at the current and reference parameter value pairs (see Lemmas 5 and 6 and also their proofs for the details). In contrast, Shibagaki et al. (2016) estimated the optima heavily based on the duality gap. As we mentioned in the introduction section, duality gap is usually large in the early stages and decreases gradually, which weakens its capability in the early stages and leads to a limited overall speedup. Second, algorithmically, Shibagaki et al. (2016) is dynamic while our method is static. In other words, Shibagaki et al. (2016) identifies the inactive features and samples during the training process, and in our method, we do this before the training process (see steps 6 to 11 in Algorithm 1), which means Shibagaki et al. (2016) needs to apply its screening rules for many times while we trigger SIFS only once. These two technical and algorithmic differences make our method outperform Shibagaki et al. (2016), which will be verified in the experimental section. At last, we theoretically prove that in the static scenario the orders of applying the feature and sample screening rules have no impact on the final performance, while Shibagaki et al. (2016) did not give the theoretical result accordingly in dynamic screening.

There are mainly three big differences between our SIFS and existing methods for other sparse learning models. First, these existing methods identify either features or samples individually and would be helpless in real applications involving a huge number of samples and extremely high-dimensional features, while SIFS identifies features and samples simultaneously. Second, technically, SIFS can incorporate the feature (resp. sample) screening results of the previous steps as the prior knowledge into the primal (reps. dual) optimum estimation to obtain a more accurate estimation. This is verified with a strong theoretical guarantee (see Lemmas 5 and 6). At last, we give a theoretical proof (see Sections 4.1 and 4.2) to show the existence of the synergistic effect between feature and sample screening for the first time in the static scenario.

Finally, we would like to point out that although the key ideas used in some of the screening methods including SIFS seem to be similar, they are developed specifically for the sparse models they focus on, which makes it nontrivial and even very difficult to reduce one method to another by simply resetting the loss and regularizer of the model. For example, SIFS cannot be reduced to Wang et al. (2014b) by setting $\alpha = 0$ and letting loss be the logistic loss, although both of their dual optimum estimations are based on the strong convexity of the objective. The reason is that they use the strong convexity in quite different ways due to their different dual problems including the feasible regions and the expressions of the dual objectives (see Theorem 2 in Wang et al., 2014b, Lemma 6 above, and their proofs for the details). Moreover, we cannot reduce SIFS to the methods (Ogawa et al., 2013; Wang et al., 2014a) for SVM by setting β to be 0, since they are based on the convexity of the objective while SIFS exploits the strong convexity.

5. SIFS for Multi-class Sparse SVMs

In this section, we consider extending SIFS to multi-class sparse SVMs. We will briefly review the basics of multi-class sparse SVMs and then derive a series of theorems as we did

for sparse SVMs above. Finally, we will present the detailed screening rule for multi-class sparse SVMs.

5.1. Basics of Multi-class Sparse SVMs

We focus on the ℓ_1-regularized multi-class SVM with a smoothed hinge loss, which takes the form of

$$\min_{\mathbf{w}\in\mathbb{R}^{Kp}} P(\mathbf{w};\alpha,\beta) = \frac{1}{n}\sum_{i=1}^{n}\ell_i(\mathbf{w}) + \frac{\alpha}{2}\|\mathbf{w}\|^2 + \beta\|\mathbf{w}\|_1, \qquad \text{(m-P*)}$$

where $\mathbf{w} = [\mathbf{w}_1;\mathbf{w}_2;...;\mathbf{w}_K] \in \mathbb{R}^{Kp}$ is the parameter vector to be estimated with $\mathbf{w}_k \in \mathbb{R}^p, k = 1,...,K$. The loss function $\ell_i(\mathbf{w}) = \sum_{k\neq y_i}\ell(\mathbf{w}_k^\top\mathbf{x}_i - \mathbf{w}_{y_i}^\top\mathbf{x}_i + 1)$, with $\{\mathbf{x}_i, y_i\}_{i=1}^n$ is the training data set of K classes, $\mathbf{x}_i \in \mathbb{R}^p, y_i \in \{1,...,K\}$, and $\ell(\cdot)$ is the smoothed hinge loss defined in Eq. (1).

The following theorem presents the Lagrangian dual problem of (m-P*) and the KKT conditions, which are essential for developing the screening rules.

Theorem 13 *For each sample $(\mathbf{x}_i, y_i)$, we define $\mathbf{u}_i = 1-\mathbf{e}_{y_i} \in \mathbb{R}^K$ and $\mathbf{X}_i = [\mathbf{X}_i^1, \mathbf{X}_i^2, ..., \mathbf{X}_i^K] \in \mathbb{R}^{Kp\times K}$, where $\mathbf{X}_i^k = vec(\mathbf{x}_i(\mathbf{e}_k - \mathbf{e}_{y_i})^\top) \in \mathbb{R}^{Kp}$. Let $\mathbf{u} = [\mathbf{u}_1;...;\mathbf{u}_n] \in \mathbb{R}^{Kn}$ and $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n] \in \mathbb{R}^{Kp\times Kn}$. Then, for the problem (m-P*), the followings hold:*
(i) *the dual problem of (m-P*) is*

$$\min_{\theta\in[0,1]^{Kn}} D(\theta;\alpha,\beta) = \frac{1}{2\alpha}\left\|\mathcal{S}_\beta\left(\frac{1}{n}\mathbf{X}\theta\right)\right\|^2 + \frac{\gamma}{2n}\|\theta\|^2 - \frac{1}{n}\langle\mathbf{u},\theta\rangle, \qquad \text{(m-D*)}$$

where $\theta = [\theta_1;...;\theta_n]$ with $\theta_i \in [0,1]^K, i = 1,...,n$;
(ii) *denote the optima of problems (m-P*) and (m-D*) by $\mathbf{w}^*(\alpha,\beta)$ and $\theta^*(\alpha,\beta)$, respectively, then,*

$$\mathbf{w}^*(\alpha,\beta) = \frac{1}{\alpha}\mathcal{S}_\beta\left(-\frac{1}{n}\mathbf{X}\theta^*(\alpha,\beta)\right), \qquad \text{(m-KKT-1)}$$

$$[\theta_i^*(\alpha,\beta)]_k = \begin{cases} 0, & \text{if } \langle\mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + [\mathbf{u}_i]_k \le 0; \\ 1, & \text{if } \langle\mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + [\mathbf{u}_i]_k \ge \gamma; \\ \frac{1}{\gamma}(\langle\mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + [\mathbf{u}_i]_k), & \text{otherwise;} \end{cases} \quad k = 1,2,...,K. \qquad \text{(m-KKT-2)}$$

As we did in Section 2, we can also define 3 index sets here:

$$\mathcal{F} = \left\{(k,j)\in[K]\times[p] : \frac{1}{n}|[\mathbf{X}\theta^*(\alpha,\beta)]_{k,j}| \le \beta\right\},$$
$$\mathcal{R} = \left\{(i,k)\in[n]\times[K] : \langle\mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + [\mathbf{u}_i]_k \le 0\right\},$$
$$\mathcal{L} = \left\{(i,k)\in[n]\times[K] : \langle\mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + [\mathbf{u}_i]_k \ge \gamma\right\},$$

which imply that

$$\text{(i) } (k,j)\in\mathcal{F} \Rightarrow [\mathbf{w}_k^*(\alpha,\beta)]_j = 0,$$
$$\text{(ii) } \begin{cases} (i,k)\in\mathcal{R} \Rightarrow & [\theta_i^*(\alpha,\beta)]_k = 0, \\ (i,k)\in\mathcal{L} \Rightarrow & [\theta_i^*(\alpha,\beta)]_k = 1. \end{cases} \qquad \text{(m-R)}$$

Suppose that we are given three subsets of $\mathcal{F}, \mathcal{R}$, and $\mathcal{L}$, then we can infer the values of many coefficients of $\mathbf{w}^*(\alpha,\beta)$ and $\theta^*(\alpha,\beta)$ via Rule m-R. The lemma below shows that the rest coefficients of $\mathbf{w}^*(\alpha,\beta)$ and $\theta^*(\alpha,\beta)$ can be recovered by solving a small sized problem.

Lemma 14 *Given index sets $\hat{\mathcal{F}} \subseteq \mathcal{F}, \hat{\mathcal{R}} \subseteq \mathcal{R}$, and $\hat{\mathcal{L}} \subset \mathcal{L}$, the followings hold:*
(i) $[\mathbf{w}^*(\alpha,\beta)]_{\hat{\mathcal{F}}} = 0$, $[\theta^*(\alpha,\beta)]_{\hat{\mathcal{R}}} = 0$, $[\theta^*(\alpha,\beta)]_{\hat{\mathcal{L}}} = 1$;
(ii) *let $\hat{\mathcal{D}} = \hat{\mathcal{R}}\cup\hat{\mathcal{L}}$, $\hat{\mathbf{G}}_1 = {}_{\hat{\mathcal{F}}^c}[\mathbf{X}]_{\hat{\mathcal{D}}^c}$, and $\hat{\mathbf{G}}_2 = {}_{\hat{\mathcal{F}}^c}[\mathbf{X}]_{\hat{\mathcal{L}}}$, where $\hat{\mathcal{F}}^c = [K]\times[p]\setminus\hat{\mathcal{F}}$, $\hat{\mathcal{D}}^c = [n]\times[K]\setminus\hat{\mathcal{D}}$, $\hat{\mathcal{L}}^c = [n]\times[K]\setminus\hat{\mathcal{L}}$ and $\hat{\mathbf{u}} = [\mathbf{u}]_{\hat{\mathcal{D}}^c}$, then, $[\theta^*(\alpha,\beta)]_{\hat{\mathcal{D}}^c}$ solves the following scaled dual problem:*

$$\min_{\hat{\theta}\in[0,1]^{|\hat{\mathcal{D}}^c|}} \left\{\frac{1}{2\alpha}\left\|\mathcal{S}_\beta\left(\frac{1}{n}\hat{\mathbf{G}}_1\hat{\theta} + \frac{1}{n}\hat{\mathbf{G}}_2\mathbf{1}\right)\right\|^2 + \frac{\gamma}{2n}\|\hat{\theta}\|^2 - \frac{1}{n}\langle\hat{\mathbf{u}},\hat{\theta}\rangle\right\}; \qquad \text{(m-scaled-D*)}$$

(iii) *suppose that $\theta^*(\alpha,\beta)$ is known, then,*

$$[\mathbf{w}^*(\alpha,\beta)]_{\hat{\mathcal{F}}^c} = \frac{1}{\alpha}\mathcal{S}_\beta\left(-\frac{1}{n}{}_{\hat{\mathcal{F}}^c}[\mathbf{X}]\theta^*(\alpha,\beta)\right).$$

Given two estimations $\mathcal{W}$ and Θ for $\mathbf{w}^*(\alpha,\beta)$ and $\theta^*(\alpha,\beta)$, we can define three subsets of $\mathcal{F}, \mathcal{R}$, and $\mathcal{L}$ below as we did in the binary case to relax Rule m-R into the applicable version:

$$\hat{\mathcal{F}} = \left\{(k,j)\in[K]\times[p] : \max_{\theta\in\Theta}\left\{\frac{1}{n}|[\mathbf{X}\theta]_{k,j}|\right\} \le \beta\right\},$$
$$\hat{\mathcal{R}} = \left\{(i,k)\in[n]\times[K] : \max_{\mathbf{w}\in\mathcal{W}}\left\{\langle\mathbf{X}_i^k, \mathbf{w}\rangle + [\mathbf{u}_i]_k\right\} \le 0\right\},$$
$$\hat{\mathcal{L}} = \left\{(i,k)\in[n]\times[K] : \min_{\mathbf{w}\in\mathcal{W}}\left\{\langle\mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + [\mathbf{u}_i]_k\right\} \ge \gamma\right\}.$$

5.2. Estimate the Primal and Dual Optima

We first derive the effective intervals of the parameters α and β.

Lemma 15 *Let $\beta_{\max} = \|\frac{1}{n}\mathbf{X}\mathbf{u}\|_\infty$ and $\alpha_{\max}(\beta) = \frac{1}{1-\gamma}\max_{(i,k)\in[n]\times[K]}\left\{\langle\mathbf{X}_i^k, \mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\mathbf{u})\rangle\right\}$. Then:*
(i) *for $\alpha > 0$ and $\beta \ge \beta_{\max}$, we have*

$$\mathbf{w}^*(\alpha,\beta) = \mathbf{0}, \;\; \theta^*(\alpha,\beta) = \mathbf{u};$$

(ii) *for all $\alpha \in [\max\{\alpha_{\max}(\beta), 0\}, \infty) \cap (0,\infty)$, we have*

$$\mathbf{w}^*(\alpha,\beta) = \frac{1}{\alpha}\mathcal{S}_\beta\left(-\frac{1}{n}\mathbf{X}\mathbf{u}\right), \;\; \theta^*(\alpha,\beta) = \mathbf{u}. \qquad (18)$$

Hence, we only need to consider the cases with $\beta \in (0, \beta_{\max}]$ and $\alpha \in (0, \alpha_{\max}(\beta)]$.

Since problem (m-P*) is a special case of problem (g-P*) with $L(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^n \ell_i(\mathbf{w})$, given the reference solution $\mathbf{w}^*(\alpha_0,\beta_0)$ and the index set $\hat{\mathcal{F}}$ of the inactive features identified by the previous IFS steps, we can apply Lemma 5 to obtain the estimation for $\mathbf{w}^*(\alpha,\beta_0)]_{\hat{\mathcal{F}}^c}$:

$$[\mathbf{w}^*(\alpha,\beta_0)]_{\hat{\mathcal{F}}^c} \in \mathcal{W} := \{\mathbf{w} : \|\mathbf{w} - \mathbf{c}\| \le r\}, \qquad (19)$$

where $\mathbf{c}$ and r are defined in Eqs. (6) and (7), respectively.

Moreover, noting that problem (m-D*) is also a special case of problem (g-D*), given the reference solution $\theta^*(\alpha_0, \beta_0)$ and the index sets of inactive samples identified by the previous ISS steps $\hat{\mathcal{R}}$ and $\hat{\mathcal{L}}$, we can obtain an estimation for $[\theta^*(\alpha, \beta_0)]_{\hat{\mathcal{D}}^c}$ from Lemma 6:

$$[\theta^*(\alpha, \beta_0)]_{\hat{\mathcal{D}}^c} \in \Theta := \{\theta : \|\theta - \mathbf{c}\| \leq r\}, \tag{20}$$

where $\mathbf{c}$ and r are defined by Eqs. (9) and (10) with $\mathbf{v} = \mathbf{u}$, respectively.

5.3. Screening Rule SIFS

Given the optima $\mathbf{w}^*(\alpha_0, \beta_0)$ and $\theta^*(\alpha_0, \beta_0)$, to derive the feature screening rule IFS, we need to solve the optimization problem below first:

$$s^{(k,j)}(\alpha, \beta_0) = \max_{\theta \in \Theta} \left\{ \frac{1}{n} |\langle [\mathbf{X}_{k,j}]_{\hat{\mathcal{D}}^c}, \theta \rangle + \langle [\mathbf{X}_{k,j}]_{\hat{\mathcal{L}}}, \mathbf{1} \rangle| \right\}, (k,j) \in \hat{\mathcal{F}}^c, \tag{21}$$

where $\mathbf{X}_{k,j}$ is the $((k-1)p + j)$-th row of $\mathbf{X}$, Θ is given by Eq. (20), and $\hat{\mathcal{F}}$ and $\hat{\mathcal{D}} = \hat{\mathcal{R}} \cup \hat{\mathcal{L}}$ are the index sets of the inactive features and samples identified in the previous screening process. We notice that this problem has exactly the same form as problem (12). Hence, from Lemma 8 we can obtain its closed-form solution directly. Now, from Rule m-R, we can obtain our IFS rule below.

- The feature screening rule IFS takes the form of

$$s^{(k,j)}(\alpha, \beta_0) \leq \beta_0 \Rightarrow [\mathbf{w}_k^*(\alpha, \beta_0)]_j = 0, \forall (k,j) \in \hat{\mathcal{F}}^c. \tag{IFS}$$

- The index set $\hat{\mathcal{F}}$ can be updated by

$$\hat{\mathcal{F}} \leftarrow \hat{\mathcal{F}} \cup \Delta\hat{\mathcal{F}} \text{ with } \Delta\hat{\mathcal{F}} = \{(k,j) : s^{(k,j)} \leq \beta_0, (k,j) \in \hat{\mathcal{F}}^c\}. \tag{22}$$

Similarly, we need to solve the following problems to derive our sample screening rule ISS:

$$u_{(i,k)}(\alpha, \beta_0) = \max_{\mathbf{w} \in \mathcal{W}} \{\langle [\mathbf{X}_i^k]_{\hat{\mathcal{F}}^c}, \mathbf{w} \rangle + [\mathbf{u}_i]_k\}, (i,k) \in \hat{\mathcal{D}}^c, \tag{23}$$

$$l_{(i,k)}(\alpha, \beta_0) = \min_{\mathbf{w} \in \mathcal{W}} \{\langle [\mathbf{X}_i^k]_{\hat{\mathcal{F}}^c}, \mathbf{w} \rangle + [\mathbf{u}_i]_k\}, (i,k) \in \hat{\mathcal{D}}^c, \tag{24}$$

where $\mathcal{W}$ is given by Eq. (19). We find that they can be solved by Lemma 10 directly. Therefore, we can obtain our sample screening rule ISS below.

- The sample screening rule ISS takes the form of

$$\begin{array}{l} u_{(i,k)}(\alpha, \beta_0) \leq 0 \Rightarrow [\theta_i^*(\alpha, \beta_0)]_k = 0, \\ l_{(i,k)}(\alpha, \beta_0) \geq \gamma \Rightarrow [\theta_i^*(\alpha, \beta_0)]_k = 1, \end{array} \forall (i,k) \in \hat{\mathcal{D}}^c. \tag{ISS}$$

- We can update the index sets $\hat{\mathcal{R}}$ and $\hat{\mathcal{L}}$ by

$$\hat{\mathcal{R}} \leftarrow \hat{\mathcal{R}} \cup \Delta\hat{\mathcal{R}} \text{ with } \Delta\hat{\mathcal{R}} = \{(i,k) : u_{(i,k)}(\alpha, \beta_0) \leq 0, (i,k) \in \hat{\mathcal{D}}^c\}, \tag{25}$$

$$\hat{\mathcal{L}} \leftarrow \hat{\mathcal{L}} \cup \Delta\hat{\mathcal{L}} \text{ with } \Delta\hat{\mathcal{L}} = \{(i,k) : l_{(i,k)}(\alpha, \beta_0) \geq \gamma, (i,k) \in \hat{\mathcal{D}}^c\}. \tag{26}$$

The same as we did sparse SVM, we can also develop SIFS to reinforce the capabilities of ISS and IFS by applying them alternatively. The framework of SIFS for solving the model over a grid of parameter values here is the same as that in the case of sparse SVM, i.e. Algorithm 1, except for the different rules IFS and ISS, and the updates of $\hat{\mathcal{R}}$, $\hat{\mathcal{L}}$, and $\hat{\mathcal{F}}$.

In this version of SIFS, the order of applying IFS and ISS also has no impact on the final performance. Since the form of the theorem and its proof are nearly the same as that of Theorem 12, to avoid redundancy, we omit them in this section.

6. Experiments

We evaluate SIFS on both synthetic and real data sets in terms of three measurements: *speedup*, *scaling ratio*, and *rejection ratio*. Speedup is given by the ratio of the running time of the solver without screening to that with screening.

For sparse SVMs, scaling ratio is defined as $1 - \frac{(n-\tilde{n})(p-\tilde{p})}{np}$, where $\tilde{n}$, $\tilde{p}$, n, and p are the numbers of inactive samples and features identified by SIFS, sample size, and feature dimension of the data set. From steps 6 to 11 in Algorithm 1, we can see that we trigger the rules ISS and IFS repetitively until no new features and samples are identified. We adopt the rejection ratios of the i-th triggering of ISS and IFS defined as $\frac{\tilde{n}_i}{n_0}$ and $\frac{\tilde{p}_i}{p_0}$ to evaluate their performances in each triggering, where $\tilde{n}_i$ and $\tilde{p}_i$ are the numbers of inactive samples and features identified in the i-th triggering of ISS and IFS. n_0 and p_0 are the numbers of inactive samples and features.

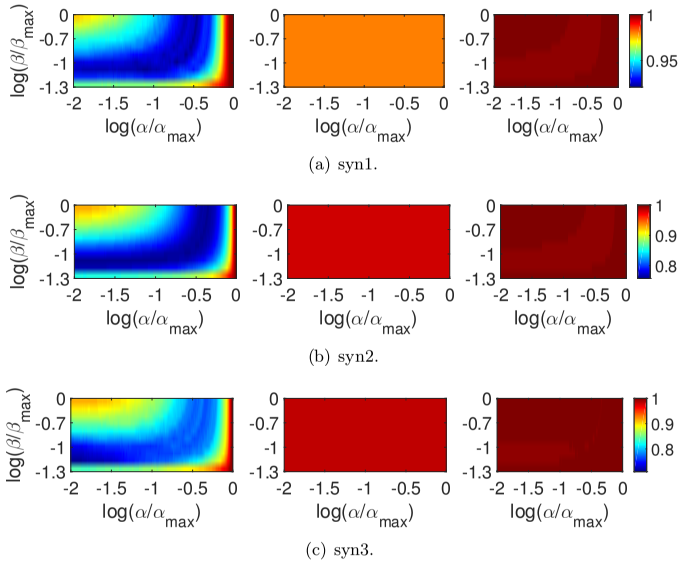
For multi-class sparse SVMs, we let scaling ratio be $1 - \frac{(Kn-\tilde{n})(Kp-\tilde{p})}{K^2np}$, where $\tilde{n} = |\hat{\mathcal{R}} \cup \hat{\mathcal{L}}|$, $\tilde{p} = |\hat{\mathcal{F}}|$, and n and p are the sample size and the feature dimension of the data set. The rejection ratios of the i-th triggering of ISS and IFS are $\frac{\tilde{n}_i}{n_0}$ and $\frac{\tilde{p}_i}{p_0}$, respectively, where $p_0 = |\mathcal{F}|$, $n_0 = |\mathcal{R} \cup \mathcal{L}|$, $\tilde{p}_i = |\Delta\hat{\mathcal{F}}|$ and $\tilde{n}_i = |\Delta\hat{\mathcal{R}} \cup \Delta\hat{\mathcal{L}}|$ with $\Delta\hat{\mathcal{F}}, \Delta\hat{\mathcal{R}}$, and $\Delta\hat{\mathcal{L}}$ are the increments of $\hat{\mathcal{F}}, \hat{\mathcal{R}}$ and $\hat{\mathcal{L}}$ in the i-th triggering of the rules IFS and ISS. Please see Eqs. (22), (25), and (26) for the detailed definitions of $\Delta\hat{\mathcal{F}}, \Delta\hat{\mathcal{R}}$, and $\Delta\hat{\mathcal{L}}$.

Recall that, we can integrate SIFS with any solvers for problem (P*). In this experiment, we use Accelerated Proximal Stochastic Dual Coordinate Ascent (AProx-SDCA) (Shalev-Shwartz and Zhang, 2016) as a baseline, as it is one of the state-of-the-arts. We choose the state-of-art screening method for sparse SVMs in Shibagaki et al. 2016 as another baseline only in the experiments of sparse SVMs, since it cannot be applied in multi-class sparse SVMs. As we mentioned in the introduction section that screening differs greatly from feature selection methods, it is not appropriate to make comparisons with feature selection methods.

For each data set, we solve problem (P*) at a grid of turning parameter values. Specifically, we first compute $\beta_{\max}$ by Lemma 4 and then select 10 values of β that are equally spaced at the logarithmic scale of $\beta/\beta_{\max}$ from 1 to 0.05. Then, for each value of β, we first compute $\alpha_{\max}(\beta)$ by Lemma 4 and then select 100 values of α that are equally spaced at the logarithmic scale of $\alpha/\alpha_{\max}(\beta)$ from 1 to 0.01. Thus, for each data set, we solve problem (P*) at 1,000 pairs of parameter values in total. We write the code in C++ along with Eigen library for numerical computations. We perform all the computations on a single core of Intel(R) Core(TM) i7-5930K 3.50GHz, 128GB MEM.

6.1. Simulation Studies with Spare SVMs

We evaluate SIFS on 3 synthetic data sets named syn1, syn2, and syn3 with sample and feature size $(n, p) \in \{(10000, 1000), (10000, 10000), (1000, 10000)\}$. We present each data point as $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2]$ with $\mathbf{x}_1 \in \mathbb{R}^{0.02p}$ and $\mathbf{x}_2 \in \mathbb{R}^{0.98p}$. We use Gaussian distributions $\mathcal{G}_1 = N(\mathbf{u}, 0.75\mathbf{I}), \mathcal{G}_2 = N(-\mathbf{u}, 0.75\mathbf{I})$, and $\mathcal{G}_3 = N(0, 1)$ to generate the data points, where $\mathbf{u} = 1.5\mathbf{1}$ and $\mathbf{I} \in \mathbb{R}^{0.02p \times 0.02p}$ is the identity matrix. To be precise, $\mathbf{x}_1$ for positive and negative points are sampled from $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. For each entry in $\mathbf{x}_2$, it has chance $\eta = 0.02$ to be sampled from $\mathcal{G}_3$ and chance $1 - \eta$ to be 0.



(a) syn1.

(b) syn2.

(c) syn3.

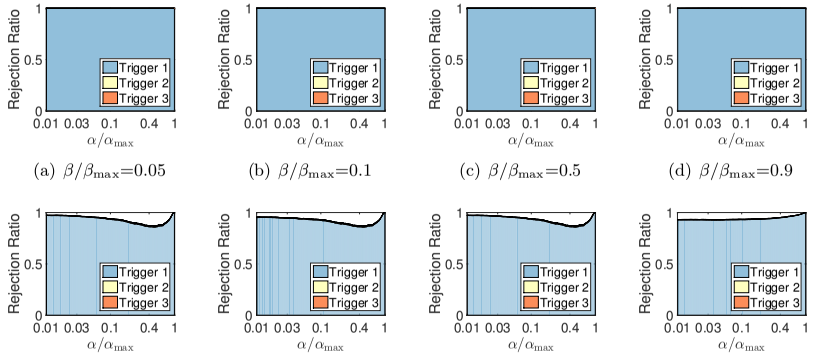
Figure 1: Scaling ratios of ISS, IFS, and SIFS (from left to right).

Fig. 1 shows the scaling ratios by ISS, IFS, and SIFS on the synthetic data sets at $1,000$ parameter values. We can see that IFS is more effective in scaling problem size than ISS, with scaling ratios roughly 98% against $70 - 90\%$. Moreover, SIFS, which is an alternating application of IFS and ISS, significantly outperforms ISS and IFS, with scaling ratios roughly 99.9%. This high scaling ratios imply that SIFS can lead to a significant speedup.

Due to the space limitation, we only report the rejection ratios of SIFS on syn2. Other results can be found in the appendix. Fig. 2 shows that SIFS can identify most of the inactive features and samples. However, few features and samples are identified in the second and later triggerings of ISS and IFS. The reason may be that the task here is so simple that one triggering is enough.

Table 1 reports the running times of solver without and with IFS, ISS and SIFS for solving problem (P*) at $1,000$ pairs of parameter values. We can see that SIFS leads to

significant speedups, that is, up to 76.8 times. Taking syn2 for example, without SIFS, the solver takes more than two hours to solve problem (P*) at $1,000$ pairs of parameter values. However, combined with SIFS, the solver only needs less than three minutes for solving the same set of problems. From the theoretical analysis in Shalev-Shwartz and Zhang 2016 for AProx-SDCA, we can see that its computational complexity rises proportionately to the sample size n and the feature dimension p. From this theoretical result, we can see that the results in Fig. 1 are roughly consistent with the speedups we achieved shown in Table 1.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

Figure 2: Rejection ratios of SIFS on syn 2 (first row: Feature Screening, second row: Sample Screening).

Data	Solver	ISS+Solver			IFS+Solver			SIFS+Solver		
		ISS	Solver	Speedup	IFS	Solver	Speedup	SIFS	Solver	Speedup
syn1	499.1	4.9	27.8	15.3	2.3	42.6	11.1	8.6	6.0	**34.2**
syn2	8749.9	24.9	1496.6	5.8	23.0	288.1	28.1	92.6	70.3	**53.7**
syn3	1279.7	2.0	257.1	4.9	2.2	33.4	36.0	7.2	9.5	**76.8**

Table 1: Running time (in seconds) for solving problem (P*) at $1,000$ pairs of parameter values on three synthetic data sets.

6.2. Experiments with Sparse SVMs on Real Datasets

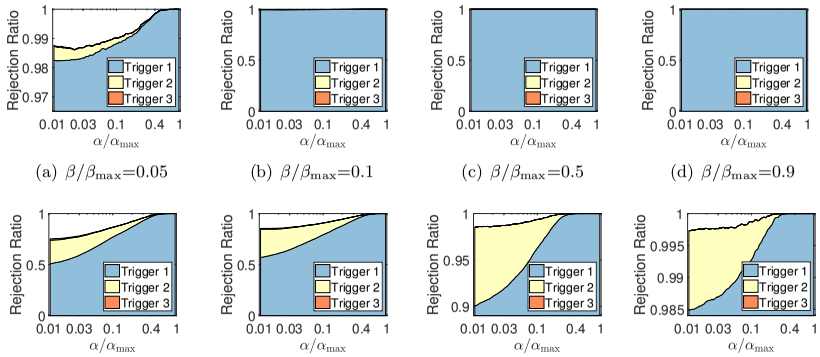
In this experiment, we evaluate the performance of SIFS on 5 large-scale real data sets: real-sim, rcv1-train, rcv1-test, url, and kddb, which are all collected from the project page of LibSVM (Chang and Lin, 2011). See Table 2 for a brief summary. We note that, the kddb data set has about 20 million samples with 30 million features.

Recall that, SIFS detects the inactive features and samples in a static manner, i.e., we perform SIFS only once before the training process and hence the size of the problem we need to perform optimization on is fixed. However, the method in Shibagaki et al. 2016 detects inactive features and samples in a dynamic manner (Bonnefoy et al., 2014), i.e.,

they perform their method along with the training process and thus the size of the problem would keep decreasing during the iterations. Thus, comparing SIFS with this baseline in terms of the rejection ratios is inapplicable. We compare the performance of SIFS with the method in Shibagaki et al. 2016 in terms of speedup, i.e., the speedup gained by these two methods in solving problem (P*) at $1,000$ pairs of parameter values. The code of the method in Shibagaki et al. 2016 is obtained from (`https://github.com/husk214/s3fs`).

Dataset	Feature size: p	Sample size:n	Classes
real-sim	20,958	72,309	2
rcv1-train	47,236	20,242	2
rcv1-test	47,236	677, 399	2
url	3,231,961	2,396,130	2
kddb	29,890,095	19,264,097	2

Table 2: Statistics of the binary real data sets.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

Figure 3: Rejection ratios of SIFS on real-sim when it performs **ISS first** (first row: Feature Screening, second row: Sample Screening).

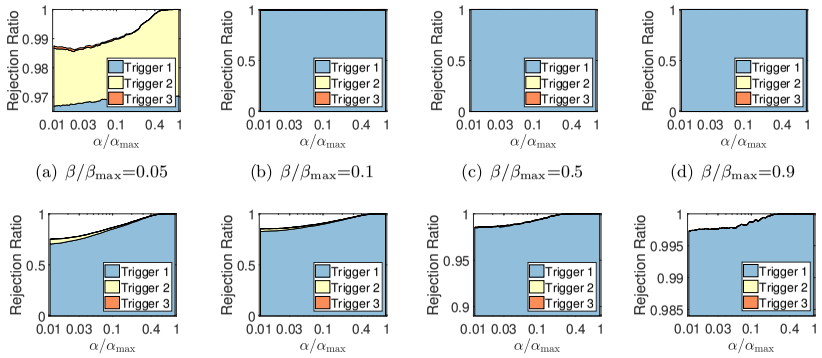
Fig. 3 shows the rejection ratios of SIFS on the real-sim data set (other results are in the appendix). In Fig. 3, we can see that some inactive features and samples are identified in the 2nd and 3rd triggerings of ISS and IFS, which verifies the necessity of the alternating application of ISS and IFS. SIFS is efficient since it always stops in 3 times of triggering. In addition, most of ($> 98\%$) the inactive features can be identified in the 1st triggering of IFS while identifying inactive samples needs to apply ISS two or more times. It may due to two reasons: 1) we run ISS first, which reinforces the capability of IFS due to the synergistic effect (see Sections 4.1 and 4.2), referring to the analysis below for further verification; 2) feature screening here may be easier than sample screening.

Table 3 reports the running times of solver without and with the method in Shibagaki et al. 2016 and SIFS for solving problem (P*) at $1,000$ pairs of parameter values on real data

sets. The speedup gained by SIFS is up to 300 times on real-sim, rcv1-train and rcv1-test. Moreover, SIFS significantly outperforms the method in Shibagaki et al. 2016 in terms of speedup—by about 30 to 40 times faster on the aforementioned three data sets. For data sets url and kddb, we do not report the results of the solver as the sizes of the data sets are huge and the computational cost is prohibitive. Instead, we can see that the solver with SIFS is about 25 times faster than the solver with the method in Shibagaki et al. 2016 on both data sets url and kddb. Let us take the data set kddb as an example. The solver with SIFS takes about 13 hours to solve problem (P*) for $1,000$ pairs of parameter values, while the solver with the method in Shibagaki et al. 2016 needs 11 days to finish the same task.

Data Set	Solver	Method in Shibagaki et al. 2016+Solver			SIFS+Solver		
		Screen	Solver	Speedup	Screen	Solver	**Speedup**
real-sim	3.93E+04	24.10	4.94E+03	7.91	60.01	140.25	**195.00**
rcv1-train	2.98E+04	10.00	3.73E+03	7.90	27.11	80.11	**277.10**
rcv1-test	1.10E+06	398.00	1.35E+05	8.10	1.17E+03	2.55E+03	**295.11**
url	>3.50E+06	3.18E+04	8.60E+05	>4.00	7.66E+03	2.91E+04	**>100**
kddb	>5.00E+06	4.31E+04	1.16E+06	>4.00	1.10E+04	3.6E+04	**>100**

Table 3: Running time (in seconds) for solving problem (P*) at $1,000$ pairs of parameter values on five real data sets.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

Figure 4: Rejection ratios of SIFS on real-sim when it performs **IFS first** (first row: Feature Screening, second row: Sample Screening).

Verification of the Synergy Effect In Fig. 3, SIFS performs ISS (sample screening) first, while in Fig. 4, it performs IFS (feature screening) first. All the rejection ratios of the 1st triggering of IFS in Fig. 3(a)-3(d) where SIFS performs ISS first are much higher than (at least equal to) those in Fig. 4(a)-4(d) where SIFS performs IFS first. In turn, all the rejection ratios of the 1st triggering of ISS in Fig. 4(e)-4(h) where SIFS performs IFS

first are also much higher than those in Fig. 3(e)-3(h) where SIFS performs ISS first. This demonstrates that the screening result of ISS can reinforce the capability of IFS and vice versa, which is the so called synergistic effect. At last, in Fig. 4 and Fig. 3, we can see that the overall rejection ratios at the end of SIFS are exactly the same. Hence, no matter which rule (ISS or IFS) we perform first in SIFS, SIFS has the same screening performances in the end, which verifies the conclusion we present in Theorem 12.

Performance in Solving Single Problems In the experiments above, we solve problem (P*) at a grid of turning parameter values. This setting is meaningful and it arises naturally in various cases, such as cross validation (Kohavi, 1995) and feature selection (Meinshausen and Bühlmann, 2010; Yang et al., 2015). Therefore, the results above demonstrate that our method would be helpful in such real applications. We notice that sometimes one may be interested in the model at a specific parameter value pair. To this end, we now evaluate the performance of our method in solving a single problem. Due to the space limitation, we solve sparse SVM at 16 specific parameter values pairs on the real-sim data set for examples, To be precise, $\beta \in \{0.05, 0.10, 0.5, 0.9\}$ and each β has 4 values of α satisfying $\alpha/\alpha_{\max}(\beta) \in \{0.05, 0.1, 0.5, 0.9\}$. For each (α_i, β_i), we construct a parameter value path $\{(\alpha_{j,i}, \beta_i), j = 0, ..., M\}$ with $\alpha_{j,i}$ equally spaced at the logarithmic scale of $\alpha_{j,i}/\alpha_{max}(\beta_i)$ between 1 and $\alpha_i/\alpha_{max}(\beta_i)$, which implies that $\alpha_{max}(\beta_i) = \alpha_{0,i} > \alpha_{1,i} > ... > \alpha_{M,i} = \alpha_i$. Then we use AProx-SDCA integrated with SIFS to solve problem (P*) at the parameter value pairs on the path one by one and report the total time cost. It can be expected that for α_i far from $\alpha_{max}(\beta_i)$, we need more points on the parameter value path in order to obtain higher rejection ratios and more significant speedups of SIFS and otherwise we need fewer ones. In this experiment, we fix $M = 50$ at each α_i for convenience. For comparison, we solve the problem at (α_i, β_i) using AProx-SDCA with random initializations, which would be faster than solving all the problems on the path. Table 4 reports the speedups achieved by SIFS at these parameter value pairs. It shows that SIFS can still obtain significant speedups. In addition, compared with the results in Tables 1 and 3, we can see that our method is much better at solving a problem sequence than solving a single problem.

$\beta/\beta_{\max}$ \ $\alpha/\alpha_{\max}(\beta)$	0.05	0.10	0.50	0.90
0.05	6.31	7.14	21.04	146.66
0.10	6.14	7.18	21.83	156.63
0.50	5.75	7.18	20.99	154.60
0.90	6.39	7.66	23.16	173.56

Table 4: The speedups achieved by SIFS in solving single problems

6.3. Experiments with Multi-class Sparse SVMs

We evaluate SIFS for multi-class Sparse SVMs on 3 synthetic data sets (syn-multi1, syn-multi2, and syn-multi3) and 2 real data sets (news20 and rcv1-multiclass). Their statistics are given in Table 5.

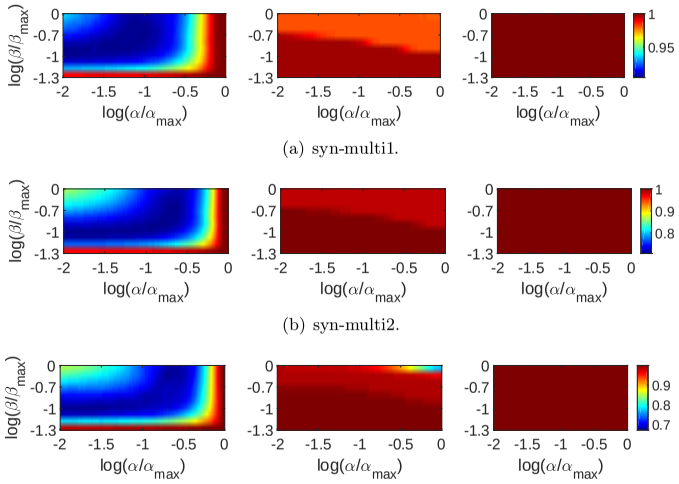
The synthetic data sets are generated in a similar way as we did in Section 6.1. Each of them has $K = 5$ classes and each class has n/K samples. The data points of them can be written as $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2]$, where $\mathbf{x}_1 = [\mathbf{x}_1^1; ...; \mathbf{x}_1^K] \in \mathbb{R}^{0.02p}$ with $\mathbf{x}_1^k \in \mathbb{R}^{0.02p/K}$ and $\mathbf{x}_2 \in \mathbb{R}^{0.98p}$. If $\mathbf{x}$ belongs to the k-th class, then $\mathbf{x}_1^k$ is sampled from a Gaussian distribution

$\mathcal{G} = N(\mathbf{u}, 0.75\mathbf{I})$ with $\mathbf{u} = 1.5\mathbf{1}$, $\mathbf{I} \in \mathbb{R}^{(0.02p/K)\times(0.02p/K)}$ and other components in $\mathbf{x}_1$ are from $N(0, 1)$. Each entry of $\mathbf{x}_2$ has chance $\eta = 0.2$ to be sampled from distribution $N(0, 1)$ and chance $1 - \eta$ to be 0.

Dataset	Feature size: p	Sample size: n	Classes
syn-multi1	1,000	10,000	5
syn-multi2	10,000	10,000	5
syn-multi3	10,000	1,000	5
news20	3,993	62,060	20
rcv1-multiclass	47,236	15,564	53

Table 5: Statistics of the multi-class data sets.

Fig. 5 shows the scaling ratios of ISS, IFS, and SIFS on the synthetic data sets at $1,000$ pairs of parameter values. Similar to the results in sparse SVMs, SIFS totally surpasses IFS and ISS, with scaling ratios larger than 98% at any parameter value pair against $70 - 90\%$. Thus, we can expect significant speedups by integrating SIFS with the solver.



(a) syn-multi1.

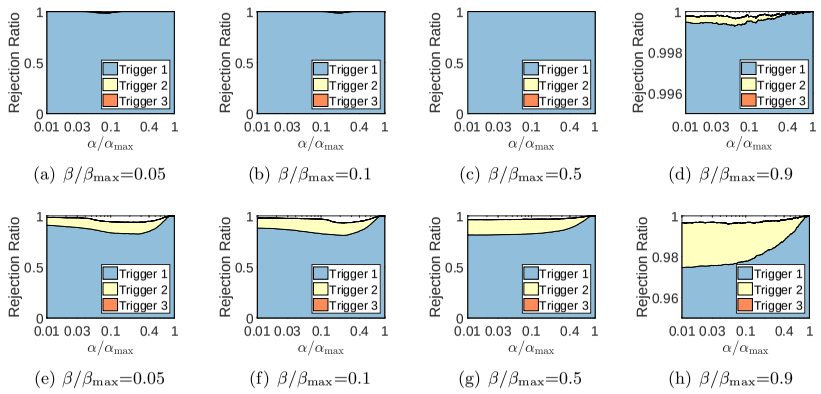
(b) syn-multi2.

(c) syn-multi3.

Figure 5: Scaling ratios of ISS, IFS, and SIFS (from left to right).

Fig. 6 and 7 present the rejection ratios of SIFS on the news20 data set (the results of other data sets can be found in the appendix). It indicates that, by alternatively applying IFS and ISS, we can finally identify most of the inactive samples ($> 95\%$) and features ($> 99\%$). The synergistic effect between IFS and ISS can also be found in these figures.

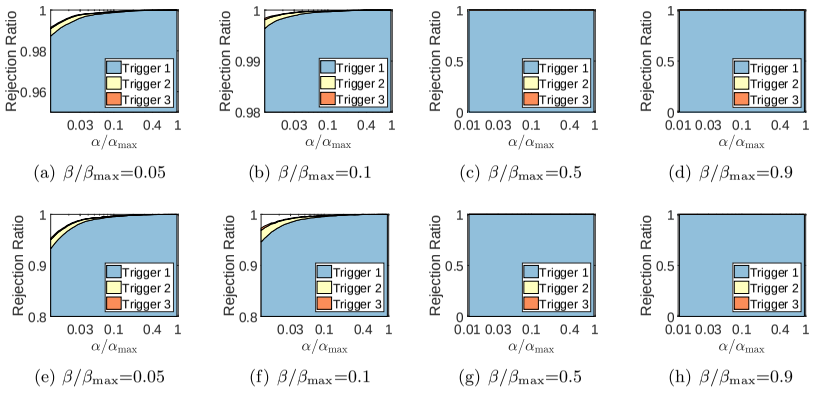
As a result of excellent performance of SIFS in identifying inactive features and samples, we observe significant speedups gained by SIFS in Table 6, which are up to 200 times on news20 and 300 times on syn-multi3. Specifically, on news20, the solver without any



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

Figure 6: Rejection ratios of SIFS on syn-multi2 (first row: Feature Screening, second row: Sample Screening).



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

Figure 7: Rejection ratios of SIFS on news20 (first row: Feature Screening, second row: Sample Screening).

screening method takes about 97 hours to solve the problem at $1,000$ pairs of parameter values; however, integrated with SIFS, the solver only needs less than 0.5 hour for the same task. Moreover, Table 6 also indicates that the speedup gained by SIFS is much more significant than that gained by IFS and ISS. This demonstrates again the great benefit of alternatively applying IFS and ISS in SIFS. At last, we notice that the speedup gained by

SIFS and IFS on syn-multi3 is far greater than those gained on syn-multi1 and syn-multi2. The reason is that the convergence rate of the solver in our problem may depend on the condition number of the data matrix. The dimension of $\mathbf{w}$ in syn-multi3 is much larger than its sample size ($50,000$ over $1,000$), which leads to a large condition number and a bad convergence rate. However, SIFS and IFS can remove the inactive features from the model, which can greatly improve the convergence rate of the solver by decreasing the condition number. This point can also be supported by the fact in the experiment on syn-multi3 that the solver with SIFS or IFS needs much fewer iterations to converge to the optima than that without SIFS or IFS.

Data	Solver	ISS+Solver			IFS+Solver			SIFS+Solver		
		ISS	Solver	Speedup	IFS	Solver	Speedup	SIFS	Solver	**Speedup**
syn-multi1	9009	72	1233	6.9	90	388	18.9	178	83	**34.5**
syn-multi2	105484	412	27661	3.8	920	2606	29.9	1936	893	**37.3**
syn-multi3	78373	44	18341	4.3	84	337	186.2	203	59	**299.9**
news20	350277	283	13950	24.6	1000	65175	5.3	1485	290	**197.3**
rcv1-multiclass	396205	793	44639	8.7	1620	33191	11.4	2850	1277	**96.0**

Table 6: Running time (in seconds) for solving problem (m-P*) at $1,000$ pairs of parameter values on synthetic and real data sets.

7. Conclusions

In this paper, we proposed a novel data reduction method SIFS to simultaneously identify inactive features and samples for sparse SVMs. Our major contribution is a novel framework for an accurate estimation of the primal and dual optima based on strong convexity. To the best of our knowledge, the proposed SIFS is the first static screening method that is able to simultaneously identify inactive features and samples for sparse SVMs. An appealing feature of SIFS is that all detected features and samples are guaranteed to be irrelevant to the outputs. Thus, the model learned on the reduced data is exactly identical to that learned on the full data. To show the flexibility of the proposed SIFS, we extended it to multi-class sparse SVMs. The experimental results demonstrate that, for both sparse SVMs and multi-class sparse SVMs, SIFS can dramatically reduce the problem size and the resulting speedup can be orders of magnitude. We plan to generalize SIFS to more complicated models, e.g., SVMs with a structured sparsity-inducing penalty.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) under Grant 2013CB336500, National Natural Science Foundation of China under Grant 61233011, and National Youth Top-notch Talent Support Program.

Appendix A. Detailed Proofs and More Experimental Results

In this appendix, we first present the detailed proofs of all the theorems in the main paper and then report the rest experimental results which are omitted in the experimental section.

A.1. Proof for Lemma 3

Proof of Lemma 3:

1) It is the immediate conclusion of the analysis above.

2) After feature screening, the primal problem (P*) is scaled into:

$$\min_{\tilde{\mathbf{w}}\in\mathbb{R}^{|\hat{\mathcal{F}}^c|}} \frac{\alpha}{2}||\tilde{\mathbf{w}}||^2 + \beta||\tilde{\mathbf{w}}||_1 + \frac{1}{n}\sum_{i=1}^{n} \ell(1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \tilde{\mathbf{w}}\rangle). \tag{scaled-P^*-1}$$

Thus, we can easily derive out the dual problem of (scaled-P^*-1):

$$\min_{\tilde{\theta}\in[0,\alpha]^n} \tilde{D}(\tilde{\theta};\alpha,\beta) = \frac{1}{2\alpha}||\mathcal{S}_\beta(\frac{1}{n}_{\hat{\mathcal{F}}^c}[\bar{\mathbf{X}}]\tilde{\theta})||^2 + \frac{\gamma}{2n}||\tilde{\theta}||^2 - \frac{1}{n}\langle\mathbf{1},\tilde{\theta}\rangle, \tag{scaled-D^*-1}$$

and the KKT conditions:

$$\tilde{\mathbf{w}}^*(\alpha,\beta) = \frac{1}{\alpha}\mathcal{S}_\beta(\frac{1}{n}_{\hat{\mathcal{F}}^c}[\bar{\mathbf{X}}]\tilde{\theta}^*(\alpha,\beta)), \tag{scaled-KKT-1}$$

$$[\tilde{\theta}^*(\alpha,\beta)]_i = \begin{cases} 0, & \text{if } 1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \tilde{\mathbf{w}}^*(\alpha,\beta)\rangle < 0, \\ \frac{1}{\gamma}(1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \tilde{\mathbf{w}}^*(\alpha,\beta)), & \text{if } 0 \le 1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \tilde{\mathbf{w}}^*(\alpha,\beta) \le \gamma, \\ 1, & \text{if } 1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \tilde{\mathbf{w}}^*(\alpha,\beta) > \gamma, \end{cases} \tag{scaled-KKT-2}$$

Then, it is obvious that $\tilde{\mathbf{w}}^*(\alpha,\beta) = [\mathbf{w}^*(\alpha,\beta)]_{\hat{\mathcal{F}}^c}$, since essentially, problem (scaled-P^*-1) can be derived by substituting 0 to the weights for the eliminated features in problem (P*) and optimizing over the rest weights.

Since the solutions $\mathbf{w}^*(\alpha,\beta)$ and $\theta^*(\alpha,\beta)$ satisfy the conditions (KKT-1) and (KKT-2) and $\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}, \tilde{\mathbf{w}}^*(\alpha,\beta)\rangle = \langle\bar{\mathbf{x}}_i, \mathbf{w}^*(\alpha,\beta)\rangle$ for all i , we know that $\tilde{\mathbf{w}}^*(\alpha,\beta)$ and $\theta^*(\alpha,\beta)$ satisfy the conditions (scaled-KKT-1) and (scaled-KKT-2). Thus they are the solutions of problems (scaled-P^*-1) and (scaled-D^*-1). Then, due to the uniqueness of the solution of problem (scaled-D^*-1), we have

$$\theta^*(\alpha,\beta) = \tilde{\theta}^*(\alpha,\beta). \tag{27}$$

From 1) we have $[\tilde{\theta}^*(\alpha,\beta)]_{\hat{\mathcal{R}}^c} = 0$ and $[\tilde{\theta}^*(\alpha,\beta)]_{\hat{\mathcal{L}}^c} = 1$. Therefore, from the dual problem (scaled-D*), we can see that $[\tilde{\theta}^*(C,\alpha)]_{\hat{\mathcal{D}}^c}$ can be recovered from the following problem:

$$\min_{\hat{\theta}\in[0,1]^{|\hat{\mathcal{D}}^c|}} \frac{1}{2\alpha}||\mathcal{S}_\beta(\frac{1}{n}\hat{\mathbf{G}}_1\hat{\theta} + \frac{1}{n}\hat{\mathbf{G}}_2\mathbf{1})||^2 + \frac{\gamma}{2n}||\hat{\theta}||^2 - \frac{1}{n}\langle\mathbf{1},\hat{\theta}\rangle.$$

Since $[\tilde{\theta}^*(\alpha,\beta)]_{\hat{\mathcal{D}}^c} = [\theta^*(\alpha,\beta)]_{\hat{\mathcal{D}}^c}$, the proof is therefore completed. ■

A.2. Proof for Lemma 4

Proof of Lemma 4:

(i) We prove this lemma by verifying that the solutions $\mathbf{w}^*(\alpha,\beta) = \mathbf{0}$ and $\theta^*(\alpha,\beta) = \mathbf{1}$ satisfy the conditions (KKT-1) and (KKT-2).

Firstly, since $\beta \ge \beta_{\max} = ||\frac{1}{n}\bar{\mathbf{X}}\mathbf{1}||_\infty$, we have $\mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\mathbf{1}) = 0$. Thus $\mathbf{w}^*(\alpha,\beta) = \mathbf{0}$ and $\theta^*(\alpha,\beta) = \mathbf{1}$ satisfy the condition (KKT-1).

Then, for all $i \in [n]$, we have

$$1 - \langle\bar{\mathbf{x}}_i, \mathbf{w}^*(\alpha,\beta)\rangle = 1 - 0 > \gamma.$$

Thus, $\mathbf{w}^*(\alpha,\beta) = \mathbf{0}$ and $\theta^*(\alpha,\beta) = \mathbf{1}$ satisfy the condition (KKT-2). Hence, they are the solutions of the primal problem (P*) and the dual problem (D*), respectively.

(ii) Similar to the proof of (i), we prove this by verifying that the solutions $\mathbf{w}^*(\alpha,\beta) = \frac{1}{\alpha}\mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\theta^*(\alpha,\beta))$ and $\theta^*(\alpha,\beta) = \mathbf{1}$ satisfy the conditions (KKT-1) and (KKT-2).

1. **Case 1:** $\alpha_{\max}(\beta) \le 0$. Then for all $\alpha > 0$, we have

$$\begin{aligned} &\min_{i\in[n]}\{1-\langle\bar{\mathbf{x}}_i, \mathbf{w}^*(\alpha,\beta)\rangle\} \\ =&\min_{i\in[n]}\{1-\frac{1}{\alpha}\langle\bar{\mathbf{x}}_i, \mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\theta^*(\alpha,\beta))\rangle\} = \min_{i\in[n]}\{1-\frac{1}{\alpha}\langle\bar{\mathbf{x}}_i, \mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\mathbf{1})\rangle\} \\ =&1-\frac{1}{\alpha}\max_{i\in[n]}\langle\bar{\mathbf{x}}_i, \mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\mathbf{1})\rangle = 1-(1-\gamma)\frac{1}{\alpha}\alpha_{\max}(\beta) \\ \ge&1 > \gamma. \end{aligned}$$

Then, $\mathcal{L} = [n]$, and $\mathbf{w}^*(\alpha,\beta) = \frac{1}{\alpha}\mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\theta^*(\alpha,\beta))$ and $\theta^*(\alpha,\beta) = \mathbf{1}$ satisfy the conditions (KKT-1) and (KKT-2). Hence, they are the optimal solution of the primal and dual problems (P*) and (D*).

2. **Case 2:** $\alpha_{\max}(\beta) > 0$. Then for any $\alpha \ge \alpha_{\max}(\beta)$, we have

$$\begin{aligned} &\min_{i\in[n]}\{1-\langle\bar{\mathbf{x}}_i, \mathbf{w}^*(\alpha,\beta)\rangle\} \\ =&\min_{i\in[n]}\{1-\frac{1}{\alpha}\langle\bar{\mathbf{x}}_i, \mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\theta^*(\alpha,\beta))\rangle\} = \min_{i\in[n]}\{1-\frac{1}{\alpha}\langle\bar{\mathbf{x}}_i, \mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\mathbf{1})\rangle\} \\ =&1-\frac{1}{\alpha}\max_{i\in[n]}\langle\bar{\mathbf{x}}_i, \mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\mathbf{1})\rangle = 1-(1-\gamma)\frac{1}{\alpha}\alpha_{\max}(\beta) \ge 1-(1-\gamma) = \gamma. \end{aligned}$$

Thus, $\mathcal{E} \cup \mathcal{L} = [n]$, and $\mathbf{w}^*(\alpha,\beta) = \frac{1}{\alpha}\mathcal{S}_\beta(\frac{1}{n}\bar{\mathbf{X}}\theta^*(\alpha,\beta))$ and $\theta^*(\alpha,\beta) = \mathbf{1}$ satisfy the conditions (KKT-1) and (KKT-2). Hence, they are the optimal solution of the primal and dual problems (P*) and (D*).

The proof is complete. ■

A.3. Proof for Lemma 5

Proof of Lemma 5:

Due to the α-strong convexity of the objective $P(\mathbf{w};\alpha,\beta)$, we have

$$P(\mathbf{w}^*(\alpha_0,\beta_0);\alpha,\beta_0) \geq P(\mathbf{w}^*(\alpha,\beta_0);\alpha,\beta_0) + \frac{\alpha}{2}||\mathbf{w}^*(\alpha_0,\beta_0)-\mathbf{w}^*(\alpha,\beta_0)||^2,$$
$$P(\mathbf{w}^*(\alpha,\beta_0);\alpha_0,\beta_0) \geq P(\mathbf{w}^*(\alpha_0,\beta_0);\alpha_0,\beta_0) + \frac{\alpha_0}{2}||\mathbf{w}^*(\alpha_0,\beta_0)-\mathbf{w}^*(\alpha,\beta_0)||^2,$$

which are equivalent to

$$\begin{aligned}
&\frac{\alpha}{2}||\mathbf{w}^*(\alpha_0,\beta_0)||^2 + \beta_0||\mathbf{w}^*(\alpha_0,\beta_0)||_1 + L(\mathbf{w}^*(\alpha_0,\beta_0))\\
&\geq \frac{\alpha}{2}||\mathbf{w}^*(\alpha,\beta_0)||^2 + \beta_0||\mathbf{w}^*(\alpha,\beta_0)||_1 + L(\mathbf{w}^*(\alpha,\beta_0)) + \frac{\alpha}{2}||\mathbf{w}^*(\alpha_0,\beta_0)-\mathbf{w}^*(\alpha,\beta_0)||^2,\\
&\frac{\alpha_0}{2}||\mathbf{w}^*(\alpha,\beta_0)||^2 + \beta_0||\mathbf{w}^*(\alpha,\beta_0)||_1 + L(\mathbf{w}^*(\alpha,\beta_0))\\
&\geq \frac{\alpha_0}{2}||\mathbf{w}^*(\alpha_0,\beta_0)||^2 + \beta_0||\mathbf{w}^*(\alpha_0,\beta_0)||_1 + L(\mathbf{w}^*(\alpha_0,\beta_0)) + \frac{\alpha_0}{2}||\mathbf{w}^*(\alpha_0,\beta_0)-\mathbf{w}^*(\alpha,\beta_0)||^2.
\end{aligned}$$

Adding the above two inequalities together, we obtain

$$\begin{aligned}
&\frac{\alpha-\alpha_0}{2}||\mathbf{w}^*(\alpha_0,\beta_0)||^2 \geq \frac{\alpha-\alpha_0}{2}||\mathbf{w}^*(\alpha,\beta_0)||^2 + \frac{\alpha_0+\alpha}{2}||\mathbf{w}^*(\alpha_0,\beta_0)-\mathbf{w}^*(\alpha,\beta_0)||^2\\
&\Rightarrow ||\mathbf{w}^*(\alpha,\beta_0) - \frac{\alpha_0+\alpha}{2\alpha}\mathbf{w}^*(\alpha_0,\beta_0)||^2 \leq \frac{(\alpha-\alpha_0)^2}{4\alpha^2}||\mathbf{w}^*(\alpha_0,\beta_0)||^2. \qquad (28)
\end{aligned}$$

Substituting the prior that $[\mathbf{w}^*(\alpha,\beta_0)]_{\hat{\mathcal{F}}}=0$ into Eq. (28), we obtain

$$\begin{aligned}
&||[\mathbf{w}^*(\alpha,\beta_0)]_{\hat{\mathcal{F}}^c} - \frac{\alpha_0+\alpha}{2\alpha}[\mathbf{w}^*(\alpha_0,\beta_0)]_{\hat{\mathcal{F}}^c}||^2\\
\leq &\frac{(\alpha-\alpha_0)^2}{4\alpha^2}||\mathbf{w}^*(\alpha_0,\beta_0)||^2 - \frac{(\alpha_0+\alpha)^2}{4\alpha^2}||[\mathbf{w}^*(\alpha_0,\beta_0)]_{\hat{\mathcal{F}}}||^2.
\end{aligned}$$

The proof is complete. ■

A.4. Proof for Lemma 6

Proof of Lemma 6: Firstly, we need to extend the definition of $D(\theta;\alpha,\beta)$ to $\mathbb{R}^n$:

$$\tilde{D}(\theta;\alpha,\beta) = \begin{cases} D(\theta;\alpha,\beta), & \text{if } \theta\in[0,1]^n,\\ +\infty, & \text{otherwise.}\end{cases} \qquad (29)$$

Due to the strong convexity of objective $\tilde{D}(\theta;\alpha,\beta)$, we have

$$\tilde{D}(\theta^*(\alpha_0,\beta_0),\alpha,\beta_0) \geq \tilde{D}(\theta^*(\alpha,\beta_0),\alpha,\beta_0) + \frac{\gamma}{2n}||\theta^*(\alpha_0,\beta_0)-\theta^*(\alpha,\beta_0)||^2,$$
$$\tilde{D}(\theta^*(\alpha,\beta_0),\alpha_0,\beta_0) \geq \tilde{D}(\theta^*(\alpha_0,\beta_0),\alpha_0,\beta_0) + \frac{\gamma}{2n}||\theta^*(\alpha_0,\beta_0)-\theta^*(\alpha,\beta_0)||^2.$$

Since $\theta^*(\alpha_0,\beta_0),\theta^*(\alpha,\beta_0)\in[0,1]^n$, the above inequalities are equivalent to

$$\begin{aligned}
&\frac{1}{2\alpha}f_{\beta_0}(\theta^*(\alpha_0,\beta_0)) + \frac{\gamma}{2n}||\theta^*(\alpha_0,\beta_0)||^2 - \frac{1}{n}\langle\mathbf{1},\theta^*(\alpha_0,\beta_0)\rangle\\
&\geq \frac{1}{2\alpha}f_{\beta_0}(\theta^*(\alpha,\beta_0)) + \frac{\gamma}{2n}||\theta^*(\alpha,\beta_0)||^2 - \frac{1}{n}\langle\mathbf{1},\theta^*(\alpha,\beta_0)\rangle + \frac{\gamma}{2n}||\theta^*(\alpha_0,\beta_0)-\theta^*(\alpha,\beta_0)||^2,\\
&\frac{1}{2\alpha_0}f_{\beta_0}(\theta^*(\alpha,\beta_0)) + \frac{\gamma}{2n}||\theta^*(\alpha,\beta_0)||^2 - \frac{1}{n}\langle\mathbf{1},\theta^*(\alpha,\beta_0)\rangle\\
&\geq \frac{1}{2\alpha_0}f_{\beta_0}(\theta^*(\alpha_0,\beta_0)) + \frac{\gamma}{2n}||\theta^*(\alpha_0,\beta_0)||^2 - \frac{1}{n}\langle\mathbf{1},\theta^*(\alpha_0,\beta_0)\rangle + \frac{\gamma}{2n}||\theta^*(\alpha_0,\beta_0)-\theta^*(\alpha,\beta_0)||^2.
\end{aligned}$$

Adding the above two inequalities, we obtain

$$\begin{aligned}
&\frac{\gamma(\alpha-\alpha_0)}{2n}||\theta^*(\alpha_0,\beta_0)||^2 - \frac{\alpha-\alpha_0}{n}\langle\mathbf{1},\theta^*(\alpha_0,\beta_0)\rangle\\
&\geq \frac{\gamma(\alpha-\alpha_0)}{2n}||\theta^*(\alpha,\beta_0)||^2 - \frac{\alpha-\alpha_0}{n}\langle\mathbf{1},\theta^*(\alpha,\beta_0)\rangle + \frac{\gamma(\alpha_0+\alpha)}{2n}||\theta^*(\alpha_0,\beta_0)-\theta^*(\alpha,\beta_0)||^2.
\end{aligned}$$

That is equivalent to

$$\begin{aligned}
&||\theta^*(\alpha,\beta_0)||^2 - \langle\frac{\alpha-\alpha_0}{\gamma\alpha}\mathbf{1} + \frac{\alpha_0+\alpha}{\alpha}\theta^*(\alpha_0,\beta_0),\theta^*(\alpha,\beta_0)\rangle\\
&\leq -\frac{\alpha_0}{\alpha}||\theta^*(\alpha_0,\beta_0)||^2 - \frac{\alpha-\alpha_0}{\gamma\alpha}\langle\mathbf{1},\theta^*(\alpha_0,\beta_0)\rangle. \qquad (30)
\end{aligned}$$

That is

$$||\theta^*(\alpha,\beta_0) - (\frac{\alpha-\alpha_0}{2\gamma\alpha}\mathbf{1} + \frac{\alpha_0+\alpha}{2\alpha}\theta^*(\alpha_0,\beta_0))||^2 \leq (\frac{\alpha-\alpha_0}{2\alpha})^2||\theta^*(\alpha_0,\beta_0) - \frac{1}{\gamma}\mathbf{1}||^2. \qquad (31)$$

Substituting the priors that $[\theta^*(\alpha,\beta_0)]_{\hat{\mathcal{R}}}=0$ and $[\theta^*(\alpha,\beta_0)]_{\hat{\mathcal{L}}}=1$ into Eq. (31), we have

$$\begin{aligned}
&||[\theta^*(\alpha,\beta_0)]_{\hat{\mathcal{D}}^c} - (\frac{\alpha-\alpha_0}{2\gamma\alpha}\mathbf{1} + \frac{\alpha_0+\alpha}{2\alpha}[\theta^*(\alpha_0,\beta_0)]_{\hat{\mathcal{D}}^c})||^2\\
&\leq (\frac{\alpha-\alpha_0}{2\alpha})^2||\theta^*(\alpha_0,\beta_0) - \frac{1}{\gamma}\mathbf{1}||^2 - ||\frac{(2\gamma-1)\alpha+\alpha_0}{2\gamma\alpha}\mathbf{1} - \frac{\alpha_0+\alpha}{2\alpha}[\theta^*(\alpha_0,\beta_0)]_{\hat{\mathcal{L}}}||^2\\
&- ||\frac{\alpha-\alpha_0}{2\gamma\alpha}\mathbf{1} + \frac{\alpha_0+\alpha}{2\alpha}[\theta^*(\alpha_0,\beta_0)]_{\hat{\mathcal{R}}}||^2.
\end{aligned}$$

The proof is complete. ■

A.5. Proof for Lemma 8

Before the proof of Lemma 8, we should prove that the optimization problem in (2) is equivalent to

$$s^i(\alpha,\beta_0) = \max_{\theta\in\Theta}\left\{\frac{1}{n}|\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},\theta\rangle + \langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}},\mathbf{1}\rangle|\right\}, i\in\hat{\mathcal{F}}^c. \qquad (32)$$

To avoid notational confusion, we denote the feasible region Θ and θ in Eq. (2) as $\tilde{\Theta}$ and $\tilde{\theta}$, respectively. Then,

$$\begin{aligned}
&\max_{\tilde{\theta}\in\tilde{\Theta}}\left\{\left|\frac{1}{n}[\bar{\mathbf{X}}\tilde{\theta}]_i\right|\right\} = \max_{\tilde{\theta}\in\tilde{\Theta}}\left\{\frac{1}{n}\left|\bar{\mathbf{x}}^i\tilde{\theta}\right|\right\}\\
=&\max_{\tilde{\theta}\in\tilde{\Theta}}\left\{\frac{1}{n}\left|[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c}[\tilde{\theta}]_{\hat{\mathcal{D}}^c}+[\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}}[\tilde{\theta}]_{\hat{\mathcal{L}}}+[\bar{\mathbf{x}}^i]_{\hat{\mathcal{R}}}[\tilde{\theta}]_{\hat{\mathcal{R}}}\right|\right\}\\
=&\max_{\tilde{\theta}\in\tilde{\Theta}}\left\{\frac{1}{n}|\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},[\tilde{\theta}]_{\hat{\mathcal{D}}^c}\rangle+\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}},\mathbf{1}\rangle|\right\}\\
=&\max_{\theta\in\Theta}\left\{\frac{1}{n}|\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},\theta\rangle+\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}},\mathbf{1}\rangle|\right\} = s^i(\alpha,\beta_0).
\end{aligned}$$

The last two equations hold since $[\theta]_{\hat{\mathcal{L}}}=\mathbf{1}$, $[\theta]_{\hat{\mathcal{R}}}=0$, and $[\theta_{\hat{\mathcal{D}}^c}]\in\Theta$.

Proof of Lemma 8:

$$\begin{aligned}
s^i(\alpha,\beta_0) =& \max_{\theta\in B(\mathbf{c},r)}\{\frac{1}{n}|\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},\theta\rangle+\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}},\mathbf{1}\rangle|\}\\
=& \max_{\eta\in B(\mathbf{0},r)}\{\frac{1}{n}|\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},\mathbf{c}\rangle+\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}},\mathbf{1}\rangle+\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},\eta\rangle|\}\\
=&\frac{1}{n}\big(|\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},\mathbf{c}\rangle+\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{L}}},\mathbf{1}\rangle|+\|[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c}\|r\big).
\end{aligned}$$

The last equality holds since $-\|[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c}\|r\le\langle[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c},\eta\rangle\le\|[\bar{\mathbf{x}}^i]_{\hat{\mathcal{D}}^c}\|r$.
The proof is complete. ■

A.6. Proof for Theorem 9

Proof of Theorem 9:
(1) It can be obtained from the rule R1.
(2) It is from the definition of $\hat{\mathcal{F}}$. ■

A.7. Proof for Lemma 10

Firstly, we need to point out that the optimization problems in Eqs. (3) and (4) are equivalent to the problems:

$$\begin{aligned}
u_i(\alpha,\beta_0) &= \max_{\mathbf{w}\in\mathcal{W}}\{1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{w}\rangle\}, i\in\hat{\mathcal{D}}^c,\\
l_i(\alpha,\beta_0) &= \min_{\mathbf{w}\in\mathcal{W}}\{1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{w}\rangle\}, i\in\hat{\mathcal{D}}^c.
\end{aligned}$$

They follow from the fact that $[\mathbf{w}]_{\hat{\mathcal{F}}^c}\in\mathcal{W}$ and

$$\begin{aligned}
&\{1-\langle\mathbf{w},\bar{\mathbf{x}}_i\rangle\}\\
=&\{1-\langle[\mathbf{w}]_{\hat{\mathcal{F}}^c},[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}\rangle-\langle[\mathbf{w}]_{\hat{\mathcal{F}}},[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}}\rangle\}\\
=&\{1-\langle[\mathbf{w}]_{\hat{\mathcal{F}}^c},[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}\rangle\}\ (\text{since } [\mathbf{w}]_{\hat{\mathcal{F}}}=0).
\end{aligned}$$

Proof of Lemma 10:

$$\begin{aligned}
u_i(\alpha,\beta_0) =& \max_{\mathbf{w}\in B(\mathbf{c},r)}\{1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{w}\rangle\}\\
=& \max_{\eta\in B(\mathbf{0},r)}\{1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{c}\rangle-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\eta\rangle\}\\
=&1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{c}\rangle+\max_{\eta\in B(\mathbf{0},r)}\{-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\eta\rangle\}\\
=&1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{c}\rangle+\|[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}\|r.
\end{aligned}$$

$$\begin{aligned}
l_i(\alpha,\beta_0) =& \min_{\mathbf{w}\in B(\mathbf{c},r)}\{1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{w}\rangle\}\\
=& \min_{\eta\in B(\mathbf{0},r)}\{1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{c}\rangle-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\eta\rangle\}\\
=&1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{c}\rangle+\min_{\eta\in B(\mathbf{0},r)}\{-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\eta\rangle\}\\
=&1-\langle[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c},\mathbf{c}\rangle-\|[\bar{\mathbf{x}}_i]_{\hat{\mathcal{F}}^c}\|r.
\end{aligned}$$

The proof is complete. ■

A.8. Proof for Theorem 11

Proof of Theorem 11:
(1) It can be obtained from the rule R2.
(2) It is from the definitions of $\hat{\mathcal{R}}$ and $\hat{\mathcal{L}}$. ■

A.9. Proof for Theorem 12

Proof of Theorem 12:
(1) Given the reference solutions pair $\mathbf{w}^*(\alpha_{i-1,j},\beta_j)$ and $\theta^*(\alpha_{i-1,j},\beta_j)$, if we do ISS first in SIFS and apply ISS and IFS for infinite times. If after s times of triggering, no new inactive features or samples are identified, then we can denote the sequence of $\hat{\mathcal{F}},\hat{\mathcal{R}}$, and $\hat{\mathcal{L}}$ as:

$$\hat{\mathcal{F}}_0^A=\hat{\mathcal{R}}_0^A=\hat{\mathcal{L}}_0^A=\emptyset\xrightarrow{ISS}\hat{\mathcal{F}}_1^A,\hat{\mathcal{R}}_1^A,\hat{\mathcal{L}}_1^A\xrightarrow{IFS}\hat{\mathcal{F}}_2^A,\hat{\mathcal{R}}_2^A,\hat{\mathcal{L}}_2^A\xrightarrow{ISS}...\hat{\mathcal{F}}_s^A,\hat{\mathcal{R}}_s^A,\hat{\mathcal{L}}_s^A\xrightarrow{IFS/ISS}... \quad (33)$$

$$\text{with } \hat{\mathcal{F}}_s^A=\hat{\mathcal{F}}_{s+1}^A=\hat{\mathcal{F}}_{s+2}^A=...,\hat{\mathcal{R}}_s^A=\hat{\mathcal{R}}_{s+1}^A=\hat{\mathcal{R}}_{s+2}^A=...\text{ and }\hat{\mathcal{L}}_s^A=\hat{\mathcal{L}}_{s+1}^A=\hat{\mathcal{L}}_{s+2}^A=... \quad (34)$$

In the same way, if we do IFS first in SIFS and no new inactive features or samples are identified after t times of triggering of ISS and IFS, then the sequence can be denoted as:

$$\hat{\mathcal{F}}_0^B=\hat{\mathcal{R}}_0^B=\hat{\mathcal{L}}_0^B=\emptyset\xrightarrow{IFS}\hat{\mathcal{F}}_1^B,\hat{\mathcal{R}}_1^B,\hat{\mathcal{L}}_1^B\xrightarrow{ISS}\hat{\mathcal{F}}_2^B,\hat{\mathcal{R}}_2^B,\hat{\mathcal{L}}_2^B\xrightarrow{IFS}...\hat{\mathcal{F}}_t^B,\hat{\mathcal{R}}_t^B,\hat{\mathcal{L}}_t^B\xrightarrow{IFS/ISS}... \quad (35)$$

$$\text{with } \hat{\mathcal{F}}_t^B=\hat{\mathcal{F}}_{t+1}^B=\hat{\mathcal{F}}_{t+2}^B=...,\hat{\mathcal{R}}_t^B=\hat{\mathcal{R}}_{t+1}^B=\hat{\mathcal{R}}_{t+2}^B=...\text{and }\hat{\mathcal{L}}_t^B=\hat{\mathcal{L}}_{t+1}^B=\hat{\mathcal{L}}_{t+2}^B=... \quad (36)$$

We first prove that $\hat{\mathcal{F}}_k^B\subseteq\hat{\mathcal{F}}_{k+1}^A,\hat{\mathcal{R}}_k^B\subseteq\hat{\mathcal{R}}_{k+1}^A$ and $\hat{\mathcal{L}}_k^B\subseteq\hat{\mathcal{L}}_{k+1}^A$ hold for all $k\ge0$ by induction.

1) When $k=0$, the equalities $\hat{\mathcal{F}}_0^B \subseteq \hat{\mathcal{F}}_1^A, \hat{\mathcal{R}}_0^B \subseteq \hat{\mathcal{R}}_1^A$ and $\hat{\mathcal{L}}_0^B \subseteq \hat{\mathcal{L}}_1^A$ hold since $\hat{\mathcal{F}}_0^B = \hat{\mathcal{R}}_0^B = \hat{\mathcal{L}}_0^B = \emptyset$.

2) If $\hat{\mathcal{F}}_k^B \subseteq \hat{\mathcal{F}}_{k+1}^A, \hat{\mathcal{R}}_k^B \subseteq \hat{\mathcal{R}}_{k+1}^A$ and $\hat{\mathcal{L}}_k^B \subseteq \hat{\mathcal{L}}_{k+1}^A$ hold, by the synergistic effect of ISS and IFS, we have that $\hat{\mathcal{F}}_{k+1}^B \subseteq \hat{\mathcal{F}}_{k+2}^A, \hat{\mathcal{R}}_{k+1}^B \subseteq \hat{\mathcal{R}}_{k+2}^A$ and $\hat{\mathcal{L}}_{k+1}^B \subseteq \hat{\mathcal{L}}_{k+2}^A$ hold.

Thus, $\hat{\mathcal{F}}_k^B \subseteq \hat{\mathcal{F}}_{k+1}^A, \hat{\mathcal{R}}_k^B \subseteq \hat{\mathcal{R}}_{k+1}^A$ and $\hat{\mathcal{L}}_k^B \subseteq \hat{\mathcal{L}}_{k+1}^A$ hold for all $k \geq 0$.

Similar to the analysis in (1), we can also prove that $\hat{\mathcal{F}}_k^A \subseteq \hat{\mathcal{F}}_{k+1}^B, \hat{\mathcal{R}}_k^A \subseteq \hat{\mathcal{R}}_{k+1}^B$ and $\hat{\mathcal{L}}_k^A \subseteq \hat{\mathcal{L}}_{k+1}^B$ hold for all $k \geq 0$.

Combining (1) and (2), we can acquire

$$\hat{\mathcal{F}}_0^B \subseteq \hat{\mathcal{F}}_1^A \subseteq \hat{\mathcal{F}}_2^B \subseteq \hat{\mathcal{F}}_3^A .... \tag{37}$$
$$\hat{\mathcal{F}}_0^A \subseteq \hat{\mathcal{F}}_1^B \subseteq \hat{\mathcal{F}}_2^A \subseteq \hat{\mathcal{F}}_3^B .... \tag{38}$$
$$\hat{\mathcal{R}}_0^B \subseteq \hat{\mathcal{R}}_1^A \subseteq \hat{\mathcal{R}}_2^B \subseteq \hat{\mathcal{R}}_3^A .... \tag{39}$$
$$\hat{\mathcal{R}}_0^A \subseteq \hat{\mathcal{R}}_1^B \subseteq \hat{\mathcal{R}}_2^A \subseteq \hat{\mathcal{R}}_3^A .... \tag{40}$$
$$\hat{\mathcal{L}}_0^B \subseteq \hat{\mathcal{L}}_1^A \subseteq \hat{\mathcal{L}}_2^B \subseteq \hat{\mathcal{L}}_3^A .... \tag{41}$$
$$\hat{\mathcal{L}}_0^A \subseteq \hat{\mathcal{L}}_1^B \subseteq \hat{\mathcal{L}}_2^A \subseteq \hat{\mathcal{L}}_3^B .... \tag{42}$$

By the first equality of Eqs. (34), (37), and (38), we can obtain $\hat{\mathcal{F}}_p^A = \hat{\mathcal{F}}_t^B$. Similarly, we can obtain $\hat{\mathcal{R}}_p^A = \hat{\mathcal{R}}_t^B$ and $\hat{\mathcal{L}}_p^A = \hat{\mathcal{L}}_t^B$.

(2) If p is odd, then by Eqs. (37), (39, and (41), we have $\hat{\mathcal{F}}_s^A \subseteq \hat{\mathcal{F}}_{p+1}^B, \hat{\mathcal{R}}_s^A \subseteq \hat{\mathcal{R}}_{p+1}^B$, and $\hat{\mathcal{L}}_s^A \subseteq \hat{\mathcal{L}}_{p+1}^B$. Thus $q \leq p+1$.

Else if p is even, then by Eqs. (38), (40), and (42), we have $\hat{\mathcal{F}}_s^A \subseteq \hat{\mathcal{F}}_{p+1}^B, \hat{\mathcal{R}}_s^A \subseteq \hat{\mathcal{R}}_{p+1}^B$, and $\hat{\mathcal{L}}_s^A \subseteq \hat{\mathcal{L}}_{p+1}^B$. Thus $q \leq p+1$.

Doing the same analysis for q, we can obtain $p \leq q+1$.

Hence, $|p-q| \leq 1$.

The proof is complete. ■

A.10. Proof for Theorem 13

Proof of Theorem 13:

We notice $(\mathbf{X}_i^k)^\top \mathbf{w} = \mathbf{w}_k^\top \mathbf{x}_i - \mathbf{w}_{y_i}^\top \mathbf{x}_i$. By denoting $\phi_i(\mathbf{a}) = \sum_{k=1}^K \ell([\mathbf{u}_i]_k + [\mathbf{a}]_k)$, it is easy to verify that $\ell_i(\mathbf{w}) = \phi_i(\mathbf{X}_i^\top \mathbf{w})$. Thus, the problem (m-P*) can be rewritten as:

$$\min_{\mathbf{w}\in\mathbb{R}^{Kp}} P(\mathbf{w};\alpha,\beta) = \frac{1}{n}\sum_{i=1}^n \phi_i(\mathbf{X}_i^\top \mathbf{w}) + \frac{\alpha}{2}\|\mathbf{w}\|^2 + \beta||\mathbf{w}||_1.$$

Let $\mathbf{z}_i = \mathbf{X}_i^\top \mathbf{w}$. The primal problem (P*) is then equivalent to

$$\min_{\mathbf{w}\in\mathbb{R}^p, \mathbf{z}\in\mathbb{R}^n} \frac{\alpha}{2}||\mathbf{w}||^2 + \beta||\mathbf{w}||_1 + \frac{1}{n}\sum_{i=1}^n \phi_i(\mathbf{z}_i),$$
$$\text{s.t. } \mathbf{z}_i = \mathbf{X}_i^\top \mathbf{w}, i = 1,2,..,n.$$

The Lagrangian then becomes

$$\begin{aligned} L(\mathbf{w},\mathbf{z},\theta) =& \frac{\alpha}{2}||\mathbf{w}||^2 + \beta||\mathbf{w}||_1 + \frac{1}{n}\sum_{i=1}^n \phi_i(\mathbf{z}_i) + \frac{1}{n}\sum_{i=1}^n \langle \mathbf{X}_i^\top \mathbf{w} - \mathbf{z}_i, \theta_i\rangle \\ =& \underbrace{\frac{\alpha}{2}||\mathbf{w}||^2 + \beta||\mathbf{w}||_1 + \frac{1}{n}\langle \mathbf{X}\theta, \mathbf{w}\rangle}_{:=f_1(\mathbf{w})} + \underbrace{\frac{1}{n}\sum_{i=1}^n (\phi_i(\mathbf{z}_i) - \langle \mathbf{z}_i, \theta_i\rangle)}_{:=f_2(\mathbf{z})}. \end{aligned} \tag{43}$$

We first consider the subproblem $\min_{\mathbf{w}} L(\mathbf{w},\mathbf{z},\theta)$:

$$\begin{aligned} &0 \in \partial_{\mathbf{w}} L(\mathbf{w},\mathbf{z},\theta) = \partial_{\mathbf{w}} f_1(\mathbf{w}) = \alpha\mathbf{w} + \frac{1}{n}\mathbf{X}\theta + \beta\partial||\mathbf{w}||_1 \Leftrightarrow \\ &\frac{1}{n}\mathbf{X}\theta \in -\alpha\mathbf{w} - \beta\partial||\mathbf{w}||_1 \Rightarrow \mathbf{w} = -\frac{1}{\alpha}\mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\theta). \end{aligned} \tag{44}$$

By substituting Eq. (44) into $f_1(\mathbf{w})$, we obtain

$$f_1(\mathbf{w}) = \frac{\alpha}{2}||\mathbf{w}||^2 + \beta||\mathbf{w}||_1 - \langle \alpha\mathbf{w} + \beta\partial||\mathbf{w}||_1, \mathbf{w}\rangle = -\frac{\alpha}{2}||\mathbf{w}||^2 = -\frac{1}{2\alpha}||\mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\theta)||^2. \tag{45}$$

Then, we consider the problem $\min_{\mathbf{z}} L(\mathbf{w},\mathbf{z},\theta)$:

$$\begin{aligned} 0 =& \nabla_{[\mathbf{z}_i]_k} L(\mathbf{w},\mathbf{z},\theta) = \nabla_{[\mathbf{z}_i]_k} f_2(\mathbf{z}) = \begin{cases} -\frac{1}{n}[\theta_i]_k, & \text{if } [\mathbf{z}_i]_k + [\mathbf{u}_i]_k < 0, \\ \frac{1}{\gamma n}([\mathbf{z}_i]_k + [\mathbf{u}_i]_k) - \frac{1}{n}[\theta_i]_k, & \text{if } 0 \leq [\mathbf{z}_i]_k + [\mathbf{u}_i]_k \leq \gamma, \\ \frac{1}{n} - \frac{1}{n}[\theta_i]_k, & \text{if } [\mathbf{z}_i]_k + [\mathbf{u}_i]_k > \gamma. \end{cases} \\ \Rightarrow& [\theta_i]_k = \begin{cases} 0, & \text{if } [\mathbf{z}_i]_k + [\mathbf{u}_i]_k < 0, \\ \frac{1}{\gamma}([\mathbf{z}_i]_k + [\mathbf{u}_i]_k), & \text{if } 0 \leq [\mathbf{z}_i]_k + [\mathbf{u}_i]_k \leq \gamma, \\ 1, & \text{if } [\mathbf{z}_i]_k + [\mathbf{u}_i]_k > \gamma. \end{cases} \end{aligned} \tag{46}$$

Thus, we have

$$f_2(\mathbf{z}) = \begin{cases} -\frac{\gamma}{2n}||\theta||^2 + \frac{1}{n}\langle \mathbf{u}, \theta\rangle, & \text{if } [\theta_i]_k \in [0,1], \forall i \in [n], k \in [K], \\ -\infty, & \text{otherwise}. \end{cases} \tag{47}$$

Combining Eqs.(43), (45), and (47), we obtain the dual problem:

$$\min_{\theta\in[0,1]^{Kn}} \frac{1}{2\alpha}||\mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\theta)||^2 + \frac{\gamma}{2n}||\theta||^2 - \frac{1}{n}\langle \mathbf{u}, \theta\rangle.$$

■

A.11. Proof for Lemma 14

Lemma 14 can be proved quite similarly with Lemma 3. Therefore, we omit this proof here.

A.12. Proof for Lemma 15

Proof of Lemma 15:

(i) We prove this lemma by verifying that the solutions $\mathbf{w}^*(\alpha,\beta)=\mathbf{0}$ and $\theta^*(\alpha,\beta)=\mathbf{u}$ satisfy the conditions (m-KKT-1) and (m-KKT-2).

Firstly, since $\beta \geq \beta_{\max} = ||\frac{1}{n}\mathbf{X}\mathbf{u}||_\infty$, we have $\mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\mathbf{u})=0$. Thus, $\mathbf{w}^*(\alpha,\beta)=\mathbf{0}$ and $\theta^*(\alpha,\beta)=\mathbf{u}$ satisfy the condition (m-KKT-1). Then, for all $i\in[n]$, we have

$$\langle \mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + [\mathbf{u}_i]_k = [\mathbf{u}_i]_k = \begin{cases} 1, & \text{if } k \neq y_i; \\ 0, & \text{if } k = y_i; \end{cases} \quad \forall i=1,...,n, k=1,...,K.$$

Thus, $\mathbf{w}^*(\alpha,\beta)=\mathbf{0}$ and $\theta^*(\alpha,\beta)=\mathbf{u}$ satisfy the condition (m-KKT-2). Hence, they are the solutions of the primal problem (m-P*) and the dual problem (m-D*), respectively.

(ii) Similar to the proof of (i), we prove this by verifying that the solutions $\mathbf{w}^*(\alpha,\beta)=\frac{1}{\alpha}\mathcal{S}_\beta(-\frac{1}{n}\mathbf{X}\mathbf{u})$ and $\theta^*(\alpha,\beta)=\mathbf{u}$ satisfy the conditions (m-KKT-1) and (m-KKT-2).

1. **Case 1:** $\alpha_{\max}(\beta)\leq 0$. Then for all $\alpha>0$, we have

$$\begin{aligned} &\text{if } k=y_i, \quad \langle \mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + \mathbf{u}_i^k = \mathbf{u}_i^k = 0, \\ &\text{if } k\neq y_i, \quad \langle \mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + \mathbf{u}_i^k = 1 - \langle \mathbf{X}_i^k, \frac{1}{\alpha}\mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\mathbf{u})\rangle \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = 1 - \frac{1}{\alpha}\langle \mathbf{X}_i^k, \mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\mathbf{u})\rangle \geq 1 > \gamma. \end{aligned}$$

   Thus, $\mathbf{w}^*(\alpha,\beta)=\frac{1}{\alpha}\mathcal{S}_\beta(-\frac{1}{n}\mathbf{X}\mathbf{u})$ and $\theta^*(\alpha,\beta)=\mathbf{u}$ satisfy the conditions (m-KKT-1) and (m-KKT-2). Hence, they are the optimal solution of problems (m-P*) and (m-D*).

2. **Case 2:** $\alpha_{\max}(\beta)>0$. Then for any $\alpha\geq\alpha_{\max}(\beta)$, we have

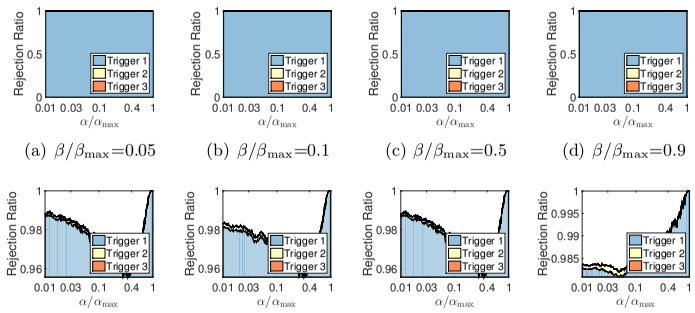
$$\begin{aligned} &\text{if } k=y_i, \quad \langle \mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + \mathbf{u}_i^k = \mathbf{u}_i^k = 0, \\ &\text{if } k\neq y_i, \quad \langle \mathbf{X}_i^k, \mathbf{w}^*(\alpha,\beta)\rangle + \mathbf{u}_i^k = 1 - \langle \mathbf{X}_i^k, \frac{1}{\alpha}\mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\mathbf{u})\rangle \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = 1 - \frac{1}{\alpha}\langle \mathbf{X}_i^k, \mathcal{S}_\beta(\frac{1}{n}\mathbf{X}\mathbf{u})\rangle \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \geq 1 - \frac{\alpha_{\max}(\beta)}{\alpha}(1-\gamma) \geq 1-(1-\gamma) = \gamma. \end{aligned}$$

   Thus, $\mathbf{w}^*(\alpha,\beta)=\frac{1}{\alpha}\mathcal{S}_\beta(-\frac{1}{n}\mathbf{X}\mathbf{u})$ and $\theta^*(\alpha,\beta)=\mathbf{u}$ satisfy the conditions (m-KKT-1) and (m-KKT-2). Hence, they are the optimal solutions of the primal and dual problems (m-P*) and (m-D*).

The proof is complete. ■

A.13. More Experimental Results

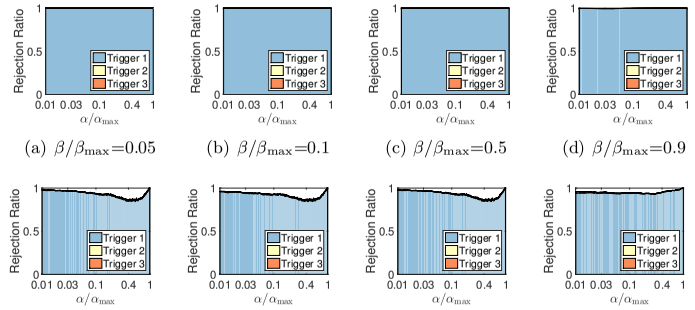
Below, we report the rejection ratios of SIFS on syn1 (Fig. 8), syn3 (Fig. 9), rcv1-train (Fig. 10), rcv1-test(Fig. 11), url (Fig. 12), kddb (Fig. 13), syn-multi1 (Fig. 14), syn-multi3 (Fig. 15), and rcv1-multiclass (Fig. 16), which are omitted in the main paper. In all the figures, the first row presents the rejection ratios of feature screening and the second row presents those of sample screening.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

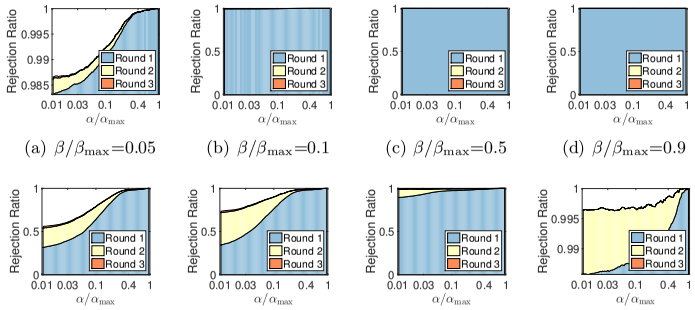
Figure 8: Rejection ratios of SIFS on syn1.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

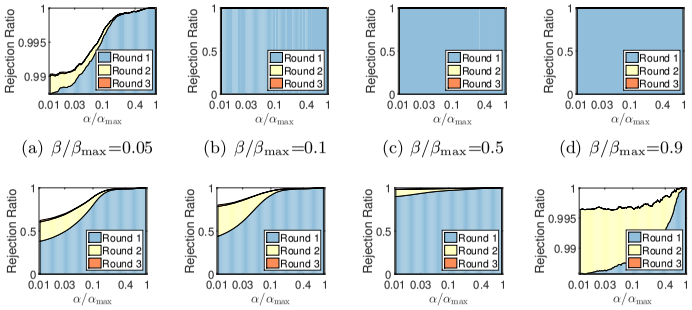
Figure 9: Rejection ratios of SIFS on syn3.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

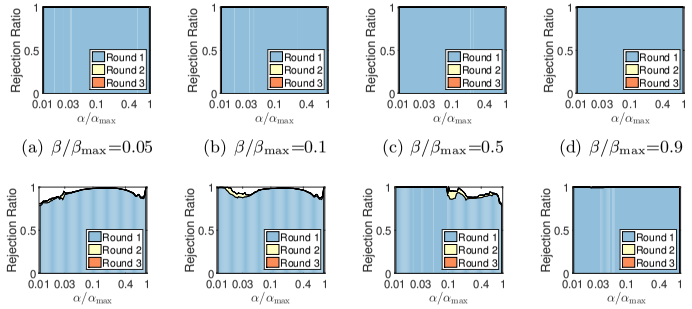
Figure 10: Rejection ratios of SIFS on rcv1-train.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

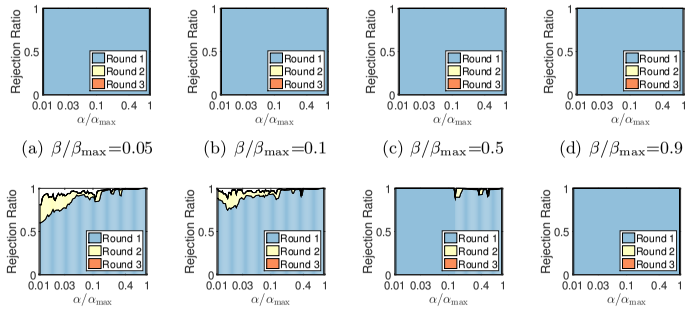
Figure 11: Rejection ratios of SIFS on rcv1-test.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

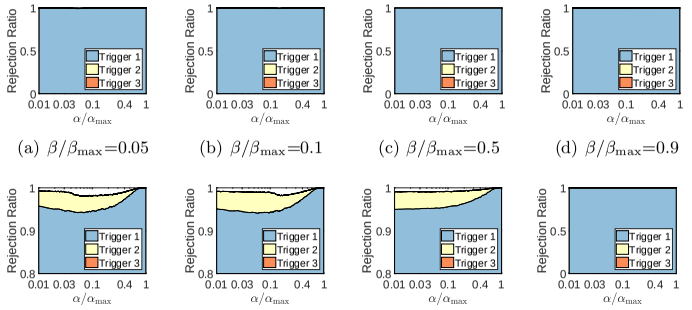
Figure 12: Rejection ratios of SIFS on url.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

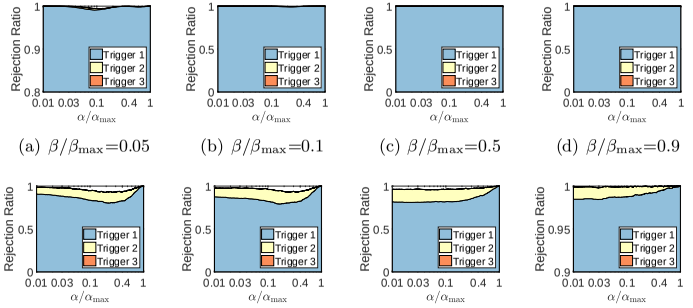
Figure 13: Rejection ratios of SIFS on kddb.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

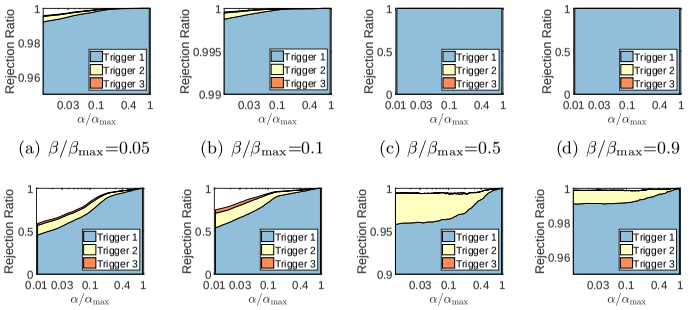
Figure 14: Rejection ratios of SIFS on syn-multi1.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

Figure 15: Rejection ratios of SIFS on syn-multi3.



(a) $\beta/\beta_{\max}$=0.05 (b) $\beta/\beta_{\max}$=0.1 (c) $\beta/\beta_{\max}$=0.5 (d) $\beta/\beta_{\max}$=0.9

(e) $\beta/\beta_{\max}$=0.05 (f) $\beta/\beta_{\max}$=0.1 (g) $\beta/\beta_{\max}$=0.5 (h) $\beta/\beta_{\max}$=0.9

Figure 16: Rejection ratios of SIFS on rcv1-multiclass.

References

Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

Jinbo Bi, Kristin Bennett, Mark Embrechts, Curt Breneman, and Minghu Song. Dimensionality reduction via sparse support vector machines. *The Journal of Machine Learning Research*, 3:1229–1243, 2003.

Antoine Bonnefoy, Valentin Emiya, Liva Ralaivola, and Rémi Gribonval. A dynamic screening principle for the lasso. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pages 6–10. IEEE, 2014.

Bryan Catanzaro, Narayanan Sundaram, and Kurt Keutzer. Fast support vector machine training and classification on graphics processors. In *Proceedings of the 25th international conference on Machine learning*, pages 104–111. ACM, 2008.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani. Safe feature elimination in sparse supervised learning. *Pacific Journal of Optimization*, 8:667–698, 2012.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9: 1871–1874, 2008.

Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *The Journal of Machine Learning Research*, 5: 1391–1415, 2004.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415. ACM, 2008.

Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, pages 1137–1145, 1995.

Irene Kotsia and Ioannis Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *Image Processing, IEEE Transactions on*, 16(1):172–187, 2007.

Mathurin Massias, Joseph Salmon, and Alexandre Gramfort. Celer: a fast solver for the lasso with dual extrapolation. In *International Conference on Machine Learning*, pages 3321–3330, 2018.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Johannes Mohr and Klaus Obermayer. A topographic support vector machine: Classification using local label configurations. In *Advances in Neural Information Processing Systems*, pages 929–936, 2004.

Harikrishna Narasimhan and Shivani Agarwal. Svm pauc tight: a new support vector method for optimizing partial auc based on a tight convex upper bound. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 167–175. ACM, 2013.

Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap safe screening rules for sparse-group lasso. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 388–396. Curran Associates, Inc., 2016.

Kohei Ogawa, Yoshiki Suzuki, and Ichiro Takeuchi. Safe screening of non-support vectors in pathwise svm computation. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1382–1390, 2013.

Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 1970.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1-2):105–145, 2016.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

Atsushi Shibagaki, Masayuki Karasuyama, Kohei Hatano, and Ichiro Takeuchi. Simultaneous safe screening of features and samples in doubly sparse modeling. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74 (2):245–266, 2012.

Jie Wang, Peter Wonka, and Jieping Ye. Scaling svm and least absolute deviations via exact data reduction. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 523–531, 2014a.

Jie Wang, Jiayu Zhou, Jun Liu, Peter Wonka, and Jieping Ye. A safe screening rule for sparse logistic regression. In *Advances in Neural Information Processing Systems*, pages 1053–1061, 2014b.

Jie Wang, Peter Wonka, and Jieping Ye. Lasso screening rules via dual polytope projection. *Journal of Machine Learning Research*, 16:1063–1101, 2015.

Li Wang, Ji Zhu, and Hui Zou. The doubly regularized support vector machine. *Statistica Sinica*, pages 589–615, 2006.

Zhen James Xiang and Peter J Ramadge. Fast lasso screening tests based on correlations. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2137–2140. IEEE, 2012.

Tao Yang, Jie Wang, Qian Sun, Derrek P Hibar, Neda Jahanshad, Li Liu, Yalin Wang, Liang Zhan, Paul M Thompson, and Jieping Ye. Detecting genetic risk factors for alzheimer's disease in whole genome sequence data via lasso screening. In *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, pages 985–989. IEEE, 2015.

Yuya Yoshikawa, Tomoharu Iwata, and Hiroshi Sawada. Latent support measure machines for bag-of-words data classification. In *Advances in Neural Information Processing Systems*, pages 1961–1969, 2014.

Weizhong Zhang, Bin Hong, Wei Liu, Jieping Ye, Deng Cai, Xiaofei He, and Jie Wang. Scaling up sparse support vector machine by simultaneous feature and sample reduction. In *Proceedings of The 34th International Conference on Machine Learning*, pages 4016–4025, 2017.

Journal of Machine Learning Research 20 (2019) 1-55 Submitted 2/18; Revised 7/19; Published 7/19

Approximate Profile Maximum Likelihood

Dmitri S. Pavlichin DMITRIP@STANFORD.EDU
Department of Applied Physics
Stanford University
Stanford, CA 94305, USA

Jiantao Jiao JIANTAO@EECS.BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720, USA

Tsachy Weissman TSACHY@STANFORD.EDU
Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA

Editor: Sara van de Geer

Abstract

We propose an efficient algorithm for approximate computation of the profile maximum likelihood (PML), a variant of maximum likelihood maximizing the probability of observing a sufficient statistic rather than the empirical sample. The PML has appealing theoretical properties, but is difficult to compute exactly. Inspired by observations gleaned from exactly solvable cases, we look for an approximate PML solution, which, intuitively, clumps comparably frequent symbols into one symbol. This amounts to lower-bounding a certain matrix permanent by summing over a subgroup of the symmetric group rather than the whole group during the computation. We extensively experiment with the approximate solution, and the empirical performance of our approach is competitive and sometimes significantly better than state-of-the-art performances for various estimation problems.

Keywords: Profile maximum likelihood, dynamic programming, sufficient statistic, partition of multi-partite numbers, integer partition

1. Introduction

The maximum likelihood principle, proposed by Ronald Fisher, has proved to be a powerful and versatile method used throughout nearly all scientific fields. However, it is still the source of considerable controversy in the statistics and machine learning community. Indeed, quoting Efron (Efron, 1982):

> *"The controversy centers on the relationship between decision theory and maximum likelihood. Beginning with the Neyman-Pearson lemma, decision theory has reshaped the theory and practice of hypothesis testing..."*

Indeed, Wald's statistical decision theory (Wald, 1950) provided a comprehensive framework for evaluating and proposing statistical procedures. The maximum likelihood approach, which can be proved to be asymptotically efficient under mild conditions in the

©2019 Dmitri S. Pavlichin, Jiantao Jiao, and Tsachy Weissman.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-075.html.

Hájek-Le Cam theory (Van der Vaart, 2000, Chap. 9), lacks a finite-sample justification. Indeed, as Le Cam (Le Cam, 1979) argued, even in the asymptotic regime the maximum likelihood approach may have weird behavior and is not always consistent.

The recent work (Acharya et al., 2017) provided an elegant justification of the maximum likelihood approach which is intimately connected to decision theory. It was shown in (Acharya et al., 2017) that plugging-in the *profile maximum likelihood* (PML) distribution estimators into a variety of functionals achieves the optimal sample complexity in estimating those functionals, which the generic *sequence maximum likelihood* (SML) fails to achieve. We also refer the readers to (Anevski et al., 2017) for statistical properties of PML.

To compute the profile maximum likelihood estimator we must solve the following optimization problem: given n samples with empirical distribution $\hat{p} = (\hat{p}_1, \hat{p}_2, \ldots)$, maximize the probability to observe $\hat{p}$ up to *relabeling* $\sigma\hat{p} \triangleq (\hat{p}_{\sigma(1)}, \hat{p}_{\sigma(2)}, \ldots)$ of the empirical distribution, where σ is some permutation. This amounts to computing the PML distribution p^*:

$$p^* = \operatorname*{argmax}_p \sum_\sigma \mathbb{P}_p(\sigma\hat{p}) = \operatorname*{argmax}_p \sum_\sigma e^{-nD(\sigma\hat{p}\|p)} \tag{1}$$

where the max is over all discrete distributions p with a known support set size (we treat the unknown case later; it is similar), the sum is over all permutations σ of the support set of distribution p, and where $D(\cdot\|\cdot)$ is the Kullback-Leibler divergence.

The profile maximum likelihood estimator is computationally challenging to find – as can shown to be equivalent to optimizing a certain matrix permanent – and the best known algorithm has running time exponential in the support size of the unknown discrete distribution. Our main contributions are the following:

1. We present an efficiently computable approximation for the PML distribution. The approximation idea is motivated by observations for the exact PML for small alphabets, where the PML distribution tends to put symbols whose empirical counts are close (within $O(\sqrt{n})$) into the same level set[1]. The idea is to lower bound the sum in (1) by summing over only the permutations that contribute "a lot" – namely the subgroup of permutations that only mix symbols within the level sets of p. This leads to the objective function in the argmax below (a lower bound to the objective function in (1)) to define the approximate PML (APML) distribution $\bar{p}^*$:

$$\bar{p}^* = \operatorname*{argmax}_p \left(e^{-nD(\hat{p}\|p)} \prod_{\alpha \in \mathcal{A}(p)} |\alpha|! \right) \tag{2}$$

   where $\mathcal{A}(p)$ is the partition of the support set of p into the level sets α of p. The second factor of the objective function rewards clumping many symbols together into a few large level sets, while the first factor encourages similarity to the empirical distribution and dominates as the sample size $n \to \infty$.

   We present a dynamic programming approach to compute the approximate PML distribution $\bar{p}^*$. Compared with existing approximations to the PML (Orlitsky et al., 2004b; Vontobel, 2012), our algorithm has no tuning parameters, is deterministic, runs

1. A level set of distribution p is $\{x : p_x = u\}$ for some value u.

in at most linear time in the sample size (usually computing the empirical histogram is the slowest part), and achieves state-of-the-art empirical performance in various estimation tasks that we detail below.

For the case of unknown support set size, we state the appropriate generalization of the PML distribution and its approximation. We modify our dynamic programming algorithm slightly to handle this case, preserving a linear worst-case runtime. It may happen that the PML and our approximate PML distributions have a discrete part with finite support and a "continuous part" in the terminology of (Orlitsky et al., 2004b); we are able to detect this case and find the discrete part of the approximate PML distribution and the probability mass of the continuous part.

2. Given the result in (Acharya et al., 2017) that plugging in the PML distribution into various functionals achieves the optimal sample complexity, we extensively experiment with the plug-in estimator using our APML distribution. We estimate entropy, Rényi entropy, support set size, L_1 distance to uniformity, and the sorted probability vector of a discrete distribution and compare with the state-of-the-art approaches with available code for estimating those functionals. We find that the performance of plugging in our APML distribution into those functionals is consistently competitive, and sometimes much better than state-of-the-art packages.

3. We extend our approximation scheme to the multi-dimensional PML problem, which is intimately connected to estimating divergence functions of discrete distributions. Utilizing results on partitions of bipartite numbers, we show that solving the two-dimensional PML problem leads to estimators that achieve the optimal sample complexity for estimating a variety of functionals such as the KL divergence, the L_1 distance, the squared Hellinger distance, and the χ^2 divergence. The multi-dimensional APML distribution turns out to be harder to compute than the one-dimensional APML, so we settle for a greedy heuristic to approximate the multi-dimensional APML distribution.

4. We provide extensive experimental results on the plug-in approach for estimating divergence functions using our approximation to the two-dimensional PML. This achieves competitive and sometimes much better results than state-of-the-art approaches in KL divergence and L_1 distance estimation.

5. We generalize the PML idea to general group actions, leading to the generalized PML approach for mutual information estimation. We analyze several candidates for the PML solution and show that one candidate makes the key arguments in (Acharya et al., 2017) fail, while the other succeeds.

We provide code at `https://github.com/dmitrip/PML` and (Pavlichin et al., 2017) for computing the approximate PML distributions.

This work is organized as follows. Section 2 reviews some results on profile maximum likelihood and functional estimation. Section 3 specializes the discussion of Section 2 to the setting of estimating functionals of discrete distributions, and defines the PML and multi-dimensional PML distributions. Section 4 presents our APML approach and observations about the exact PML that inspired it. Section 5 shows the results of numerical experiments

using the APML to estimate symmetric functionals of a distribution: the sorted probability vector, entropy, Rényi entropy, L_1 distance to uniformity, and support set size. Section 6 shows the results of numerical experiments using the APML to estimate symmetric functionals of multiple distributions: the KL divergence and L_1 distance. Appendices contain proofs, examples, most of the notation and algorithms related to the multi-dimensional APML (Appendix H), and application of the PML to estimation of mutual information (Appendix I).

2. The principle of profile maximum likelihood and partial sufficiency

We rephrase the key result in (Acharya et al., 2017) regarding the maximum likelihood principle in Theorem 3; similar ideas appeared in (Acharya et al., 2011). We define a statistical model

$$\mathcal{E} = \{\mathcal{X}, P_\theta, \theta \in \Theta\}, \tag{3}$$

where the observation $X \sim P_\theta$ for some $\theta \in \Theta$, $X \in \mathcal{X}$. Let $F(\theta) : \Theta \mapsto \mathcal{Y}$ be a measurable function. Let $d(F, \hat{F}) : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_{\geq 0}$ be the loss function, which is also assumed to be a metric.

Given observation x, upon asserting that a statistic T is *partially sufficient* (Basu, 2011) for estimating $F(\theta)$, the general profile maximum likelihood approach aims at maximizing the probability that $t = T(x)$ appears rather than the probability of x, which is the aim of traditional maximum likelihood. Then, we simply plug-in the PML estimator into the function $F(\cdot)$ to obtain an estimate of the functional $F(\theta)$. The profile maximum likelihood estimator of θ is defined as follows.

Definition 1 (Profile maximum likelihood) *The profile maximum likelihood estimator of θ is defined as*

$$\hat{\theta}_{\mathrm{T}}(t) \triangleq \operatorname*{argmax}_{\theta} P_\theta(T(X) = t). \tag{4}$$

Remark 2 *The notion of* profile likelihood *has appeared in the statistics literature before (Murphy and Van der Vaart, 2000), bearing a different meaning from the one we adopt. The principle of* profile maximum likelihood *we introduced in Definition 1 is known in the statistics community as* restricted maximum likelihood*, which was introduced in (Patterson and Thompson, 1971) and surveyed in (Harville, 1977).*

The following theorem provides a general performance guarantee for the PML algorithm introduced in Definition 1 in terms of estimating any functional $F(\theta)$.

Theorem 3 *(Acharya et al., 2017) We fix a statistical model $\mathcal{E} = \{\mathcal{X}, P_\theta : \theta \in \Theta\}$. Let $T = T(X)$ be a statistic such that $T : \mathcal{X} \mapsto \mathcal{T}$ and $|\mathcal{T}| < \infty$. Let $F(\theta) : \Theta \mapsto \mathcal{Y}$ be a measurable function. Suppose there exists an estimator $\hat{F} : \mathcal{T} \mapsto \mathcal{Y}$ such that*

$$\sup_{\theta \in \Theta} P_\theta \left(d(F(\theta), \hat{F}(T)) > \epsilon \right) < \delta. \tag{5}$$

Then,

$$\sup_{\theta \in \Theta} P_\theta \left(d(F(\theta), F(\hat{\theta}_{\mathrm{T}})) > 2\epsilon \right) \leq \delta \cdot |\mathcal{T}|. \tag{6}$$

The usefulness of Theorem 3 would rely on two factors: the existence of a good estimator $\hat{F}(T)$ in terms of *worst case risk*, and the small cardinality of the set $\mathcal{T}$ that the statistic T lies in. As we discussed above, it would be sensible to employ the PML approach if we can assert that the statistic T is "sufficient" for estimating $F(\theta)$. However, what is the right definition of sufficiency in this context?

The definition of sufficiency in this context is one of the most basic problems in decision theory, which turns out to be a highly non-trivial question. It is usually termed "partial sufficiency", and we refer the readers to (Basu, 2011) for a survey. To illustrate the subtleties of this definition, we review some historical developments below. This question was considered by Kolmogorov (Kolmogoroff, 1942), who proposed the following definition:

Definition 4 (Kolmogorov's definition of partial sufficiency) *(Kolmogoroff, 1942) A statistic $T = T(X)$ is called partially sufficient for $F(\theta)$, if the posterior distribution of $F(\theta)$ given $X = x$, depends only on $T = t$ and on the prior distribution of θ.*

It was later shown by Hájek (Hájek, 1967) that the Kolmogorov definition is void in the sense that if $F(\theta)$ is not a constant, and T is partially sufficient for $F(\theta)$ in the sense of Kolmogorov, then T is sufficient for θ in the usual sense as well.

In this context, Hájek proposed another definition of partial sufficiency. [2]

Definition 5 *(Hájek, 1967, Def. 2.2) A statistic $T = T(X)$ is called partially sufficient for $F(\theta)$ if there exists some other functional $R(\theta)$ such that $F(\theta) = F_1(R(\theta))$, and the following is satisfied:*

1. *The distribution of T will depend on $R(\theta)$ only, that is*

$$P_\theta(dT) = P_{R(\theta)}(dT); \tag{7}$$

2. *There exists a distribution $Q_R \in \mathcal{P}_R$ such that T is sufficient for the family $\{Q_R\}$, where $\mathcal{P}_R$ is the convex hull of the distributions $\{P_\theta \colon R(\theta) = R\}$.*

It was shown in (Hájek, 1967) that under the partial sufficiency in Def. 5, an analogue of the Rao-Blackwell theorem in terms of the worst case risk can be proved. [3]

It begs the question: in general how could one prove a certain statistic T is partially sufficient for $F(\theta)$? The following equivalence relation induced by group transformations is common in practice. Let $G = \{g\}$ be a group of one-to-one transformations of the $\mathcal{X}$ space on itself. We shall say that an event is G-invariant, if $gA = A$ for all $g \in G$. The set of G-invariant events is a sub-σ-algebra $\mathcal{B}$, and a measurable function f is $\mathcal{B}$-measurable if and only if $f(gx) = f(x)$ for all $g \in G, x \in \mathcal{X}$.

Theorem 6 *(Hájek, 1967) Suppose parameter θ can be represented as $\theta = (\tau, g)$, where $g \in G$, and G is a group of one-to-one transformations of the $\mathcal{X}$ space on itself. Suppose there exists a family of probability distributions $\{P_\tau\}$ indexed by τ such that*

$$P_\theta(A) = P_\tau(gX \in A), \tag{8}$$

2. For other definitions of sufficiency, see (Le, 1964; Birnbaum, 1962; Blackwell and Girshick, 1979).
3. Note that it is weaker than the usual Rao-Blackwell theorem, which holds pointwise for the risk function.

where $X \sim P_\tau$. Then, under mild conditions [4], the sub-σ-algebra of G-invariant events is partially sufficient for τ under statistical model $\{\mathcal{X}, P_\theta\}$ in the sense of Def. 5.

Theorem 6 can be interpreted in the following way. We introduce the equivalence relation $x_1 \sim x_2$ if and only if there exists some $g \in G$ such that $x_1 = gx_2$. Then, Theorem 6 states that it is partially sufficient for τ to look at the equivalence classes $[a] = \{x \in \mathcal{X} | x \sim a\}$.

3. Discrete distributions up to relabeling

In this section, we specialize Theorem 6 to several different settings, centered around the problem of estimating functionals of discrete distributions.

3.1. Permutation group and single sorted probability vector

The standard PML problem can be viewed as a special case of Theorem 6 with the group being the permutation group. Concretely, let G be the permutation group $\mathcal{S}_\mathcal{X}$ on $\mathcal{X}$, let p be a distribution supported on set $\mathcal{X}$ – that is, $p_x > 0\ \forall x \in \mathcal{X}$ – and let τ be the sorted non-increasing probability vector $(p_{(1)}, p_{(2)}, \ldots, p_{(|\mathcal{X}|)})$ of p. It is clear that the label-invariant properties of a distribution, such as the entropy, Rényi entropy, and support set size, depend on p only through τ.

Suppose we observe n i.i.d. samples $x_1^n = (x_1, \ldots, x_n)$ with distribution p. Denote by $\hat{p} = \hat{p}(x_1^n)$ the empirical distribution:

$$\hat{p} = (\hat{p}_x)_{x \in \mathcal{X}} = \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x_i = x) \right)_{x \in \mathcal{X}} \tag{9}$$

It is well known that the empirical distribution is the minimum complete sufficient statistic for p (Lehmann and Casella, 1998). The probability of observing a specific empirical distribution is given by

$$\mathbb{P}_p(\hat{p}) = \binom{n}{n\hat{p}} \prod_{x \in \mathcal{X}} p_x^{n\hat{p}_x} \tag{10}$$

where the prefactor in (10) is a multinomial coefficient[5].

Applying Theorem 6 with $\theta = p = (\tau, g)$, we obtain that the *fingerprint* statistic (Valiant and Valiant, 2011b, 2013)[6] is partially sufficient for τ. Concretely, the fingerprint $\mathcal{F} = (\mathcal{F}_i)_{i \geq 0}$ is defined so that $\mathcal{F}_i$ is the number of symbols observed exactly i times in x_1^n:

$$\mathcal{F} = \mathcal{F}(\hat{p}(x_1^n)) = (\mathcal{F}_i)_{i \geq 0} \triangleq (|\{x \in \mathcal{X} : n\hat{p}_x = i\}|)_{i \geq 0} \tag{11}$$

4. Precisely, Condition 3.1 in (Hájek, 1967).
5. $\binom{n}{n\hat{p}} \triangleq n! \prod_{x \in \mathcal{X}} \frac{1}{(n\hat{p}_x)!}$.
6. The fingerprint is also called the profile (Orlitsky et al., 2004b), histogram order statistics (Paninski, 2003), and histogram of histograms (Batu et al., 2000).

Below, we compute the probability of a specific fingerprint for the case that p is supported on finite alphabet $\mathcal{X} = \mathrm{Supp}(p)$ with

$$K \triangleq |\mathcal{X}| \tag{12}$$

and the empirical distribution $\hat{p}$ is supported on empirical alphabet $\hat{\mathcal{X}} \triangleq \mathrm{Supp}(\hat{p}) = \{x \in \mathcal{X} : \hat{p}_x > 0\} \subset \mathcal{X}$ with

$$\hat{K} \triangleq |\hat{\mathcal{X}}|. \tag{13}$$

Then $\mathcal{F}_0 = |\mathcal{X} \setminus \hat{\mathcal{X}}| = K - \hat{K}$ counts the number of "unseen" symbols, and is thus unknown if the support set size K is unknown.

In Appendix A we derive the following expression for the probability of a specific fingerprint for the case of finite $\mathcal{X}$:

$$\mathbb{P}_p(\mathcal{F}) = \left(\prod_{i\geq 0} \frac{1}{\mathcal{F}_i!}\right) \binom{n}{n\hat{p}} \operatorname{perm}\Bigg(\underbrace{\left(p_x^{n\hat{p}_{x'}}\right)_{x,x'\in\mathcal{X}}}_{Q}\Bigg) \tag{14}$$

where $\operatorname{perm}(A)$ denotes the matrix permanent of the $K \times K$ matrix A:

$$\operatorname{perm}(A) = \sum_{\sigma\in\mathcal{S}_\mathcal{X}} \prod_{x\in\mathcal{X}} A_{x,\sigma(x)} \tag{15}$$

where the sum is over the symmetric group $\mathcal{S}_\mathcal{X}$ on $\mathcal{X}$. To simplify notation, we let Q denote the $K \times K$ matrix in (14). Note that to evaluate expression (14) we need to know the support set $\mathcal{X}$, both to evaluate $\operatorname{perm}(Q)$ and $\mathcal{F}_0 = |\mathcal{X} \setminus \hat{\mathcal{X}}|$. Appendix B shows examples of the computation of $\mathbb{P}_p(\mathcal{F})$ over small alphabets.

For completeness, Appendix A also contains expression (87) for $\mathbb{P}_p(\mathcal{F})$ for the case of infinite countable $\mathcal{X}$, though this expression is not used elsewhere in this work for the reason stated in the last paragraph of Section 3.1.

For a given collection of distributions $\mathcal{P}$, the *profile maximum likelihood* distribution is defined as

$$p^* \triangleq \operatorname*{argmax}_{p\in\mathcal{P}} \mathbb{P}_p(\mathcal{F}) = \operatorname*{argmax}_{p\in\mathcal{P}} \frac{\operatorname{perm}(Q)}{(K-\hat{K})!}, \tag{16}$$

where in the second equality we discarded all p-independent factors of $\mathbb{P}_p(\mathcal{F})$ (14). $\mathcal{F}_0 = K - \hat{K} = |\mathcal{X} \setminus \hat{\mathcal{X}}|$ depends on p through its support set size. Note that $\mathbb{P}_p(\mathcal{F})$ is invariant under relabeling of the components of p, so we can choose p^* to be non-increasing in the same ordering as we choose for the support set $\mathcal{X}$. Note that the set $\mathcal{P}$ is not necessarily the same as the set of all discrete distributions.

If the collection of distributions $\mathcal{P}$ includes distributions with different support set sizes (for example, all finite support set sizes at least as large as $\hat{K}$), then we can estimate the support set size by breaking up the optimization in (16) into two steps:

$$K^* \triangleq \operatorname*{argmax}_K \left(\frac{1}{(K-\hat{K})!} \max_{p\in\mathcal{P}_K} \operatorname{perm}(Q)\right) \tag{17}$$

whenever the max over K exists, where $\mathcal{P}_K \triangleq \{p \in \mathcal{P} : |\mathrm{Supp}(p)| = K\}$. The maximizer K^* usually exists because increasing K makes the first factor in (17) smaller, but makes the $K \times K$ matrix Q bigger, boosting the number $K!$ of permutations to sum over in computing the permanent.

It may happen that K^* does not exist[7], in which case we are still able to define a PML distribution in terms of a discrete part and continuous part in the terminology of (Orlitsky et al., 2004b). (Orlitsky et al., 2004b) showed that for any fingerprint $\mathcal{F}$, there exists a distribution p^* maximizing $\mathbb{P}_p(\mathcal{F})$, but this distribution may assign some symbols to the continuous part, and others to the discrete part with finite support. The intuition is that if there are sufficiently many symbols that appear exactly once in the sample, then the PML distribution p^* assigns discrete probability 0 to infinitely many symbols to maximize the probability that each of them is seen only once. Then we can define K_d^* to be the optimal support set size of the discrete part of p^*. (Orlitsky et al., 2004b) further derived conditions lower- and upper-bounding K_d^* in terms of $\max_x \hat{p}_x$ and $\min_x \hat{p}_x$ and thus showing K_d^* exists, and upper-bounding the size of the continuous part in terms of $\mathcal{F}_1$ (the numbers of symbols occurring once). (Acharya et al., 2009) computed p^* for all sample sizes up to $n = 7$.

Note that either K^* exists or the PML distribution p^* has a continuous part, but it is never the case that p^* has infinite countable support. For this reason we never use expression (87) for the probability of a fingerprint when $\mathcal{X}$ is countably infinite.

3.2. Permutation group and multiple probability vectors

Suppose we have two distributions p, q on the same alphabet $\mathcal{X}$, such that $p_x + q_x > 0\ \forall x \in \mathcal{X}$. This condition ensures that there are no symbols x such that $p_x = q_x = 0$, which simplifies the expressions below. Draw n samples i.i.d. from p with empirical distribution $\hat{p}$ and draw m samples i.i.d. from q with empirical distribution $\hat{q}$. It is clear that the label-invariant properties of two distributions, such as the Kullback-Leibler divergence, L_1 distance, and the general family of divergences $\sum_{x\in\mathcal{X}} f(p_x, q_x)$, are invariant to the permutation group $\mathcal{S}_\mathcal{X}$ acting on pairs of distributions as $\sigma(p,q) \triangleq (\sigma p, \sigma q)$ for all $\sigma \in \mathcal{S}_\mathcal{X}$. Denote the 2-D fingerprint (Raghunathan et al., 2017) by $\mathcal{F}(\hat{p}, \hat{q})$, analogous to (11):

$$\mathcal{F} = \mathcal{F}(\hat{p}, \hat{q}) = (\mathcal{F}_{i,j})_{i,j\geq 0} \triangleq (|\{x \in \mathcal{X} : n\hat{p}_x = i, m\hat{q}_x = j\}|)_{i,j\geq 0} \tag{18}$$

The probability to draw the 2-D fingerprint under p, q is:

$$\mathbb{P}_{p,q}(\mathcal{F}) = \left(\prod_{i,j\geq 0} \frac{1}{\mathcal{F}_{i,j}!}\right) \binom{n}{n\hat{p}} \binom{m}{m\hat{q}} \operatorname{perm}\left(\left(p_x^{n\hat{p}_{x'}} q_x^{m\hat{q}_{x'}}\right)_{x,x'\in\mathcal{X}}\right) \tag{19}$$

Following the general profile maximum likelihood methodology, for a collection $\mathcal{P}$ of pairs of distributions on alphabet $\mathcal{X}$, the PML distributions p^*, q^* are:

$$(p^*, q^*) \triangleq \operatorname*{argmax}_{(p,q)\in\mathcal{P}} \mathbb{P}_{p,q}(\mathcal{F}) \tag{20}$$

Note that finding the PML pair (p^*, q^*) is not equivalent to solving two independent optimization problems since the matrix permanent in (19) does not factor into two terms. As

7. For example, if $n \geq 2$ and each symbol occurs exactly once in the sample, so $\hat{p}_x = \frac{1}{n}$ for all $x \in \mathcal{X}(\hat{p})$.

in the 1D PML case, if the support set size $K = |\mathcal{X}|$ is unknown, then we can attempt to estimate it analogously to (17). See Appendix H for a precise statement.

Appendix H generalizes the argument above in a straightforward manner to D-dimensional fingerprints (Raghunathan et al., 2017). This allows estimation of functionals of more than two distributions.

4. An approximation to the profile maximum likelihood distribution

4.1. Our approximate PML algorithm: an overview

It seems computationally challenging to compute the profile maximum likelihood distribution in all the cases discussed in Section 3, though the computational complexity of this problem is not known. Indeed, it involves computing the matrix permanent in (16), and computing permanents is $\#P$-hard (Valiant, 1979). Since the computation of permanents is exponentially slow in the support set size $|\mathcal{X}|$ with the best-known algorithms, one may lower the target and hope to find an approximate efficient algorithm to all the profile maximum likelihood problems in Section 3. Some algorithms on approximate PML have been developed: (Orlitsky et al., 2004a) proposes an expectation maximization algorithm and (Vontobel, 2012) proposes a Bethe approximation to the PML distribution. None of these works, including ours, provides guarantees on the nearness of the approximate and exact PML distributions.

Here we develop a PML distribution approximation inspired by some empirical observations in the small examples where this distribution is computable exactly in reasonable time. We provide a fast algorithm for computing this approximation. The algorithm has no tunable parameters and runs in time at most linear in the sample size in the standard PML problem (Section 3.1), its run time usually dominated by the computation of the empirical distribution $\hat{p}$. It is also efficient in solving divergence estimation problems as described in Section 3.2.

Observations from solving the PML exactly in small cases (see Section 4.2) yield the intuition that the PML distribution p^* assigns equal mass to ("clumps together") symbols whose empirical counts are close, differing on the order of $\sqrt{n}$, where n is the sample size. For such a "clumped" distribution p^*, we further guess that the value of the matrix permanent in $\mathbb{P}_{p^*}(\mathcal{F}) \sim \mathrm{perm}(Q)/\mathcal{F}_0!$ (14) is dominated by a subset of terms in the summation over the symmetric group $\mathcal{S}_{\mathcal{X}}$ – namely the subgroup of permutations that only mix symbols within the same "clump" – a level set of p^* – but not between different clumps. Armed with this intuition, we replace maximization of the matrix permanent, which seems hard, with maximization of a lower bound involving a sum over only this subgroup of permutations, which is much easier. Section 4.3 gives a precise statement of these remarks and further motivates our approach. The remainder of this section is devoted to an overview of our approximation algorithm.

Let $\mathcal{A}(p) = \{\alpha\}$ be the partition of $\mathcal{X}$ into level sets of distribution p:

$$\mathcal{A}(p) \triangleq \{\{x \in \mathcal{X} : p_x = u\} : u \in \{p_x : x \in \mathcal{X}\}\} \tag{21}$$

That is, $x, x' \in \alpha \in \mathcal{A}(p)$ if and only if $p_x = p_{x'}$.

Then our lower bound $\bar{V}$ to the log permanent is defined by summing over a subgroup of the symmetric group, and satisfies (see derivation in Section 4.3):

$$\log(\mathrm{perm}(Q)) \geq \bar{V}(p) = \sum_{\alpha \in \mathcal{A}(p)} \log(|\alpha|!) - n(D(\hat{p}||p) + H(\hat{p})) \tag{22}$$

where $D(\hat{p}||p) = \sum_{x \in \mathcal{X}} \hat{p}_x \log(\hat{p}_x/p_x)$ is the Kullback-Leibler divergence and $H(\hat{p}) = -\sum_{x \in \mathcal{X}} p_x \log(p_x)$ is the entropy of the empirical distribution. The approximate PML (APML) distribution is then the maximizer of the lower bound $\bar{V}$:

$$\bar{p}^* \triangleq \operatorname*{argmax}_{p \in \mathcal{P}} \bar{V}(p) \tag{23}$$

The right hand side of (22) yields some intuition for the approximate PML distribution. Suppose $\mathcal{P}$ contains only distributions with support set size $K = |\mathcal{X}|$, so $\mathcal{F}_0 = K - |\hat{\mathcal{X}}|$ is fixed in (16), and the first term on the right hand side of (22) encourages $\bar{p}^*$ to clump many symbols together into a few large clumps (since $\log(|\alpha|!) \sim |\alpha| \log(|\alpha|)$ is superlinear in $|\alpha|$), while the second term encourages $\bar{p}^*$ to be similar to the empirical distribution $\hat{p}$. As the sample size $n \to \infty$, the second term dominates, and we have $\bar{p}^* \to \hat{p}$, consistent with our intuition for the large sample size limit and consistent with the result of (Orlitsky et al., 2004b) (Theorem 16) for the exact PML distribution.

It turns out that computation of the approximate PML distribution $\bar{p}^*$ is equivalent to optimization over all partitions of $\mathcal{X}$. This is because, as we show, any distribution $\bar{p}^*$ maximizing $\bar{V}$ satisfies an averaging property with respect to the empirical distribution $\hat{p}$ for all $x \in \mathcal{X}$: Let $\alpha(x, \bar{p}^*) = \{x' \in \mathcal{X} : \bar{p}^*_{x'} = \bar{p}^*_x\} \in \mathcal{A}(\bar{p}^*)$ be the level set of $\bar{p}^*$ containing symbol x, then

$$\bar{p}^*_x = \frac{1}{|\alpha(x, \bar{p}^*)|} \sum_{x' \in \alpha(x, \bar{p}^*)} \hat{p}_{x'} \tag{24}$$

Therefore $\bar{p}^*$ is determined by its partition of the alphabet $\mathcal{X}$ into level sets, so we can replace maximization of $\bar{V}$ over p with maximization over all partitions $\mathcal{A}$. Let $\bar{\mathcal{A}}^*$ denote the optimal partition:

$$\bar{\mathcal{A}}^* \triangleq \operatorname*{argmax}_{\mathcal{A}\text{: partition of } \mathcal{X}} \bar{V}(\mathcal{A}) \tag{25}$$

where $\bar{V}(\mathcal{A}) \triangleq \bar{V}(p_{\mathcal{A}}(\hat{p}))$ and $p_{\mathcal{A}}(\hat{p})$ is the distribution obtained by averaging the empirical distribution over the partition elements – that is, satisfying property (24) for all $x \in \mathcal{X}$. The approximate PML distribution is then $\bar{p}^* = p_{\bar{\mathcal{A}}^*}(\hat{p})$.

We derive constraints on the optimal partition $\bar{\mathcal{A}}^*$ that enable an efficient[8] dynamic programming algorithm to compute $\bar{\mathcal{A}}^*$ and the approximate PML distribution $\bar{p}^*$. This algorithm is presented in Section 4.5.

For the case of $D \geq 2$ probability vectors and D-dimensional fingerprints – for example, when estimating the KL divergence, corresponding to $D = 2$ as in Section 3.2 – we are unable to give an efficient algorithm for maximizing our lower bound $\bar{V}$ suitably generalized to the D-dimensional case. The difficulty is that there is no natural ordering on $\mathbb{N}^D$ for

8. Running in time $O(|\operatorname{Supp}(\mathcal{F})|^2)$. Any empirical distribution $\hat{p}$ satisfies $|\operatorname{Supp}(\mathcal{F}(\hat{p}))| \leq (\sqrt{8n+1}+1)/2$ (see discussion at end of Section 4.5), where n is the sample size, so the running time is $O(n)$. The run time of our algorithm is usually dominated by computation of the empirical distribution $\hat{p}$.

$D \geq 2$, so the dynamic programming approach for $D = 1$ does not work here. In this case we settle for another layer of approximation, using a greedy heuristic to iteratively merge clumps of symbols: we enlarge the first term in (22) until the second term becomes too large. This algorithm is presented in Appendix H.

Figures 1 and 2 show the approximate PML distributions for the cases $D = 1$ and $D = 2$, respectively. For $D = 1$, the approximate PML distribution is as defined in (23). For $D \geq 2$, we approximately maximize our lower bound $\bar{V}$ to the log matrix permanent as described in Appendix H. Figure 14 in Appendix H shows the case $D = 3$.

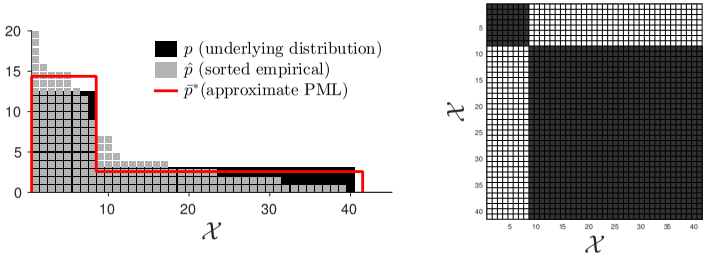


Figure 1: (left) An empirical histogram $\hat{p}$ sorted in non-increasing order (gray squares) drawn from underlying distribution p (black area) and the approximate PML (APML) distribution $\bar{p}^*$ (23) (red line), scaled by the sample size $n = 200$. The alphabet is $\mathcal{X} = \{1, \ldots, 40\}$, assumed unknown in computing $\bar{p}^*$. (right) Computation of the log permanent lower bound $\bar{V}(\bar{p}^*)$ (31) involves summing over all permutations that mix symbols only within the level sets of $\bar{p}^*$, corresponding to all permutation matrices with nonzero entries within the black blocks.

4.2. Intuitions from solving the PML exactly

For small alphabets, we can numerically and sometimes analytically optimize the permanent in (16) and (20). We observe that the PML distribution tends to assign equal mass to symbols whose empirical counts are close, differing on the order of $\sqrt{n}$, where n is the sample size. This observation motivates our approximate PML scheme in Section 4.3.

4.2.1. Exact solution for size 2 alphabet

Suppose the alphabet size is $|\mathcal{X}| = 2$. Given sample size n and empirical distribution $\hat{p} = (\hat{p}_1, \hat{p}_2) = (\hat{p}_1, 1 - \hat{p}_1)$, without loss of generality one may assume $\hat{p}_1 \geq \frac{1}{n}$ (otherwise the PML is given by $\mathsf{Bern}(1)$) and $\hat{p}_1 \leq \frac{1}{2}$. The exact PML distribution $\hat{p}^*$ was computed in (Orlitsky et al., 2004b):

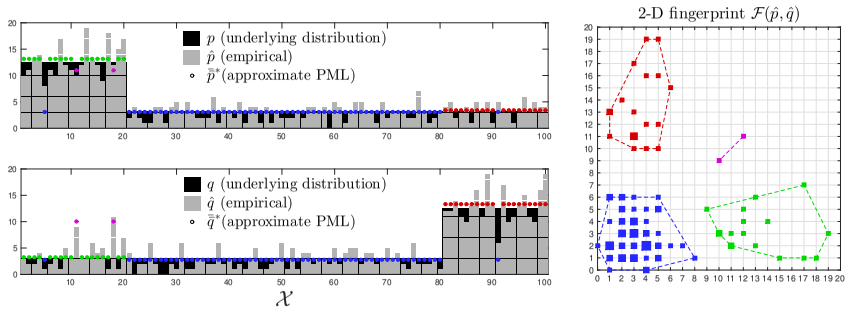


Figure 2: (Left) empirical distributions $\hat{p}$ and $\hat{q}$ (gray squares) drawn from underlying distributions p and q (black area) with $n = m = 400$ samples. The approximate PML distributions $(\bar{\hat{p}}^*, \bar{\hat{q}}^*)$ are computed by clumping entries of a 2-D fingerprint according to the greedy heuristic described in Appendix H. The approximate PML distributions are ordered in the same way as the empirical distributions (with a distinct color for each level set corresponding to the subplot on the right). All distributions are plotted scaled by the sample size. (Right) the 2-D fingerprint $\mathcal{F}_{i,j}$ at coordinates (i, j). Marker areas are roughly proportional to $\mathcal{F}_{i,j}$. All points within the colored convex hulls correspond to a level set of the approximate PML distributions. We can see three large level sets corresponding to the three distinct values of (p_x, q_x) for $x \in \mathcal{X}$.

Theorem 7 *((Orlitsky et al., 2004b)) For all $\frac{1}{n} \leq \hat{p}_1 \leq \frac{1}{2}$,*

$$\hat{p}^* = \begin{cases} \left(\frac{1}{2}, \frac{1}{2}\right) & : |\hat{p}_1 - \hat{p}_2| \leq \frac{1}{\sqrt{n}} \\ \left(\frac{1}{1+p}, \frac{p}{1+p}\right) & : |\hat{p}_1 - \hat{p}_2| > \frac{1}{\sqrt{n}} \end{cases}, \tag{26}$$

where p is the unique root in $(0, 1)$ of the polynomial

$$\hat{p}_1 p^{n(1-2\hat{p}_1)+1} - \hat{p}_2 p^{n(1-2\hat{p}_1)} + \hat{p}_2 p - \hat{p}_1. \tag{27}$$

Theorem 7 states that $|\hat{p}_1 - \hat{p}_2| \leq \frac{1}{\sqrt{n}} \Leftrightarrow p^*$ is uniform, which confirms our intuition that the relative ranking of two bins is "resolvable" if their empirical counts are more than about $\sqrt{n}$ apart, since the empirical counts are marginally binomially-distributed with standard deviation proportional to $\sqrt{n}$. Figure 3 summarizes these observations for the size-2 alphabet.

4.2.2. Size 3 alphabet

Next let's consider the case of $|\mathcal{X}| = 3$ bins. Given empirical distribution $\hat{p} = (\hat{p}_1, \hat{p}_2, \hat{p}_3)$, we use an expectation-maximization (EM) algorithm to numerically search for the PML distribution. The EM algorithm starts from an initial point $p \leftarrow \hat{p}$ and converges to a local maximum of $\mathrm{perm}(Q)$ (14). We sort the components of this local maximum in non-

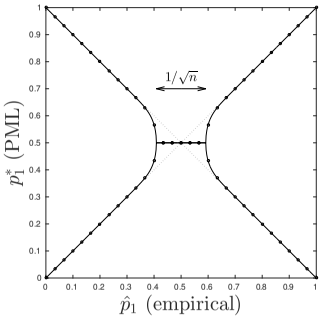


Figure 3: "Phase diagram" for the PML distribution $p^* = (p_1^*, 1 - p_1^*)$ on binary alphabet $\mathcal{X} = \{1, 2\}$ and empirical distribution $(\hat{p}_1, 1 - \hat{p}_1)$ on $n = 30$ samples (plotting the first component). Circles correspond to the possible histograms ($\hat{p}_1 \in \{0/30, 1/30, \ldots, 30/30\}$). Solid black lines correspond to the solution for arbitrary $\hat{p}_1 \in [0, 1]$, obtained numerically. The PML distribution p^* is uniform when $|\hat{p}_1 - \hat{p}_2| \leq 1/\sqrt{n}$. Outside this middle region there are two PML branches, since any permutation of the PML distribution is another PML distribution. Diagonal dashed lines correspond to the lines $(\hat{p}_1, \hat{p}_1)$ and $(\hat{p}_1, 1 - \hat{p}_1)$.

increasing order and denote the result by p^*_{EM}. Details of the EM algorithm can be found in Appendix D. We summarize our findings in Figure 4.

4.2.3. Exact solution for size 2 alphabet with D distributions

Suppose we have D distributions $((p_1^{(d)}, 1 - p_1^{(d)}))_{d=1}^D$ on the same alphabet $\mathcal{X} = \{1, 2\}$ and draw $(n_d)_{d=1}^D$ samples from each, obtaining empirical distributions $((\hat{p}_1^{(d)}, 1 - \hat{p}_1^{(d)}))_{d=1}^D$. Then the D-dimensional PML distribution $(p^{*(d)})_{d=1}^D$ (defined for $D = 2$ in Section 3.2 and for general D in Appendix H) is stated in Theorem 8.

Theorem 8 *The D-tuple of PML distributions $(p^{*(d)})_{d=1}^D$ on a binary alphabet satisfies:*

$$\sum_{d=1}^{D} 4 n_d \left(\hat{p}_1^{(d)} - \frac{1}{2}\right)^2 > 1 \quad \Rightarrow \quad (p^{*(d)})_{d=1}^D \neq \left(\left(\frac{1}{2}, \frac{1}{2}\right)\right)_{d=1}^D \tag{28}$$

See Appendix E for a proof. We conjecture that the converse of statement (28) holds.

Theorem 8 extends the results of (Orlitsky et al., 2004b) and (Fernandes and Kashyap, 2013) on the uniformity of the D-dimensional distribution beyond $D = 1$. The left side of (28) describes the interior of an ellipsoid in the space of distributions centered on $(\frac{1}{2})_{d=1}^D$, shown in Figure 5 for the case $D = 2$. Note that (28) is not consistent with D copies of condition (26) for $D \geq 2$; this reflects the fact that the probability of a D-dimensional fingerprint does not decompose into a product of D terms (see (19) for the case $D = 2$ and Appendix H for the general case).

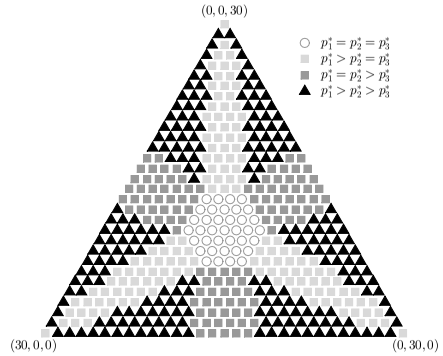


Figure 4: "Phase diagram" for the PML distribution approximation p^*_{EM} for alphabet size $|\mathcal{X}| = 3$ bins and $n = 30$ samples (analogous to Figure 3), computed by an expectation-maximization (EM) algorithm as discussed in Section 4.2.2 and Appendix D. We plot a marker for each possible empirical distribution – a type – on the simplex. Each type has three coordinates, which we then project onto the 2-dimensional image shown (so, e.g., the uniform type $n\hat{p} = (10, 10, 10)$ is in the center). The shape of the marker at position $n\hat{p}$ corresponds to the level set decomposition of the distribution p^*_{EM}, with components sorted in non-increasing order.

4.2.4. Empirical observations of the exact solutions

Figures 3, 4, and 5 suggest that the PML distribution tends to "cluster" together similar entries of the empirical histogram $\hat{p}$, rather than smooth them out. If $n|\hat{p}_x - \hat{p}_{x'}|$ is smaller than approximately $\sqrt{n}\hat{p}_x$, then the relative ranking of the true distribution's x and x' components is not well resolved statistically, and the permanent of Q (14) tends to be maximized with $p^*_x = p^*_{x'}$. On the other hand if $n|\hat{p}_x - \hat{p}_{x'}|$ is larger than approximately $\sqrt{n}\hat{p}_x$, then it tends to happen that $p^*_x \neq p^*_{x'}$. The similar phenomenon of a phase transition between a uniform and nonuniform PML distribution (for a different approximation of the PML (Vontobel, 2012)) has been reported by (Fernandes and Kashyap, 2013; Chan et al., 2015).

These observations lead to the intuition that permutations $\sigma \in \mathcal{S}_{\mathcal{X}}$ that exchange sufficiently different components of $\hat{p}$ do not contribute much to the permanent $\mathrm{perm}(Q)$. This intuition motivates our idea of simplifying the computation of the permanent by summing over only those permutations that contribute "a lot" to the sum. Section 4.3 develops this idea.

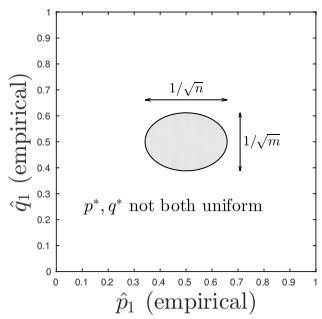


Figure 5: "Phase diagram" for the 2-D PML pair of distributions p^*, q^* (20) on binary alphabet $\mathcal{X} = \{1, 2\}$ and empirical distributions $(\hat{p}_1, 1-\hat{p}_1)$, $(\hat{q}_1, 1-\hat{q}_1)$ on $n = 10, m = 20$ samples. p^*, q^* are not both uniform whenever the empirical distribution components $\hat{p}_1, \hat{q}_1$ both lie outside the shaded ellipse $4n\left(\hat{p}_1 - \frac{1}{2}\right)^2 + 4m\left(\hat{q}_1 - \frac{1}{2}\right)^2 \leq 1$.

4.3. Approximate PML: the case of a single distribution

In the sequel, denote by $\mathcal{A}(p) = \{\alpha\}$ the partition of $\mathcal{X}$ into the level sets of p:

$$\mathcal{A}(p) \triangleq \{\{x \in \mathcal{X} : p_x = u\} : u \in \{p_x : x \in \mathcal{X}\}\} \tag{29}$$

In this Section we assume that the underlying distribution p is supported on finite set $\mathcal{X}$ of known cardinality. The case of unknown or infinite support set size is treated in Section 4.6. If the support set size is known, then so is the number of "unseen" symbols $\mathcal{F}_0$, so from (16) we conclude that the PML distribution p^* is a maximizer of the function:

$$V(p) \triangleq \log(\mathrm{perm}(Q)) \tag{30}$$

We lower bound the log permanent $V(p)$ by summing over a subset of the terms in the summation – a subgroup of the symmetric group $\mathcal{S}_{\mathcal{X}}$. Denote the lower bound by $\bar{V}(p)$:

$$\bar{V}(p) \triangleq \log\left(\sum_{\sigma \in \mathcal{S}_{\mathcal{X},p}} \prod_{x \in \mathcal{X}} p_x^{n\hat{p}_{\sigma(x)}}\right) \leq \log\left(\sum_{\sigma \in \mathcal{S}_{\mathcal{X}}} \prod_{x \in \mathcal{X}} p_x^{n\hat{p}_{\sigma(x)}}\right) = V(p) \tag{31}$$

where $\mathcal{S}_{\mathcal{X},p}$ is a subgroup of $\mathcal{S}_{\mathcal{X}}$ consisting of all permutations that exchange only those alphabet symbols $x, x' \in \mathcal{X}$ that are within the same level set of p. Equivalently, $\mathcal{S}_{\mathcal{X},p}$ is the stabilizer of distribution p under the action of relabeling its components:

$$\mathcal{S}_{\mathcal{X},p} \triangleq \{\sigma \in \mathcal{S}_{\mathcal{X}} : \sigma p = p\} \cong \bigtimes_{\alpha \in \mathcal{A}(p)} \mathcal{S}_\alpha \tag{32}$$

where $(\sigma p)_x \triangleq p_{\sigma(x)}$, $\cong$ denotes group isomorphism, $\times$ denotes the direct product of groups, and $\mathcal{S}_\alpha$ is the symmetric group acting on level set $\alpha \subset \mathcal{X}$. Then $|\mathcal{S}_{\mathcal{X},p}| = \prod_{\alpha \in \mathcal{A}(p)} |\alpha|!$.

Our approximate profile maximum likelihood distribution $\bar{p}^*$ maximizes the lower bound $\bar{V}(p)$. In other words, for the case of every distribution in collection $\mathcal{P}$ having the same support set size,

$$\bar{p}^* \triangleq \operatorname*{argmax}_{p \in \mathcal{P}} \bar{V}(p) \tag{33}$$

The case of unknown support set size is treated in Section 4.6.

Some intuition for the computational difficulty of the proposed approximation: the product structure (32) of $\mathcal{S}_{\mathcal{X},p}$ allows us to lower bound the log matrix permanent $V(p) = \log(\mathrm{perm}(Q)) \geq \log(\mathrm{perm}(\bar{Q})) = \bar{V}(p)$, where $\bar{Q}$ is the block-diagonal matrix $\bar{Q}_{x,x'} = Q_{x,x'}\mathbb{1}(\hat{p}_x = \hat{p}_{x'})$ whose permanent is the product of the permanents in each block. Since p is constant within each block, the permanent of each block is easy to evaluate, so evaluation of the lower bound $\bar{V}(p)$ is dramatically simpler than evaluation of $V(p)$, taking time $O(|\mathcal{X}|)$. It moreover turns out to be possible to optimize $\bar{V}(p)$ computationally efficiently to find the approximate PML distribution $\bar{p}^*$; this is done below.

The restriction to summing over $\mathcal{S}_{\mathcal{X},p}$ rather than $\mathcal{S}_{\mathcal{X}}$ seems large, but the hope is that $\bar{p}^*$ clusters together only comparably frequent symbols (that is, $\bar{p}^*_x = \bar{p}^*_{x'}$ implies $n|\hat{p}_x - \hat{p}_{x'}|$ is less than about $\sqrt{n\hat{p}_x}$), so we hope that $\bar{V}(\bar{p}^*) \approx V(\bar{p}^*)$ and that $\bar{p}^*$ is close to p^*, the exact profile maximum likelihood (16). Figure 1 shows that the approximate PML distribution $\bar{p}^*$ clustering together similar enough bins of $\hat{p}$. We thus expect our approximate PML scheme to perform best when data is drawn from a distribution with a few well-separated level sets (like a uniform distribution or mixture of uniforms) and less well for more smoothly-varying distributions (like a Zipf distribution); this intuition is borne by the numerical experiments of Section 5.

We can rewrite our log permanent lower bound $\bar{V}(p)$ (31) as (see proof in Appendix F):

$$\bar{V}(p) = \log\left(\sum_{\sigma \in \mathcal{S}_{\mathcal{X},p}} \prod_{x \in \mathcal{X}} p_x^{n\hat{p}_{\sigma(x)}}\right) \tag{34}$$

$$= -n\big(D(\hat{p}||p) + H(\hat{p})\big) + \sum_{\alpha \in \mathcal{A}(p)} \log(|\alpha|!) \tag{35}$$

where $\mathcal{A}(p)$ (29) is the partition of $\mathcal{X}$ into level sets of p, $D(\hat{p}||p)$ is the Kullback-Leibler divergence and $H(\hat{p})$ is the entropy of the empirical distribution.

Expression (35) yields some intuition about the approximate PML distribution $\bar{p}^*$ – a maximizer of $\bar{V}(p)$. A distribution p that "clumps" many symbols together (that is, has a few large level sets α) boosts the second term (the summation over $\alpha \in \mathcal{A}(p)$) in (35), but is very different from $\hat{p}$, thus lowering the first term. On the other hand, by setting $p = \hat{p}$, we maximize the first term, but reduce the second term. As the sample size $n \to \infty$, the contribution of the second term in (35) vanishes relative to the first term, and we have $\bar{p}^* \to \hat{p}$, the ML distribution, in this limit.

Section 4.4 establishes some properties of the approximate PML solution $\bar{p}^*$ (33) that enable us to compute it efficiently via a dynamic programming algorithm in Section 4.5.

4.4. Properties of the approximate PML distribution

Let $\alpha \subset \mathcal{X}$ and denote by $\hat{p}_\alpha$ the average value of the empirical distribution $\hat{p}$ over α:

$$\hat{p}_\alpha \triangleq \frac{1}{|\alpha|} \sum_{x \in \alpha} \hat{p}_x \tag{36}$$

We can show that the PML distribution approximation $\bar{p}^*$ (33) satisfies for all $x \in \mathcal{X}$

$$\bar{p}^*_x = \hat{p}_{\alpha(x, \bar{p}^*)} \tag{37}$$

where $\alpha(x, \bar{p}^*) = \{x' :\in \mathcal{X} : \bar{p}^*_{x'} = \bar{p}^*_x\}$ is the level set of $\bar{p}^*$ containing x. That is, $\bar{p}^*_x$ (37) is equal to the average value of the empirical histogram $\hat{p}$ over the symbols clumped with x into the same level set of $\bar{p}^*$. This follows from the fact that $D(\hat{p}\|p)$ is a Bregman divergence (Jiao et al., 2017, Lemma 4).

The *averaging* property (37) determines our approximate PML distribution $\bar{p}^*$ in terms of its partition of $\mathcal{X}$ into level sets. Therefore, instead of maximizing $\bar{V}(p)$, we maximize $\bar{V}(\mathcal{A})$, defined:

$$\bar{V}(\mathcal{A}) \triangleq \sum_{\alpha \in \mathcal{A}} \bar{V}(\alpha) \tag{38}$$

where for $\alpha \subset \mathcal{X}$

$$\bar{V}(\alpha) \triangleq \log(|\alpha|!) + n|\alpha|\hat{p}_\alpha \log(\hat{p}_\alpha) \tag{39}$$

We can check that $\bar{V}(\mathcal{A}) = \bar{V}(\hat{p}_\mathcal{A})$, where $\hat{p}_\mathcal{A}$ is the distribution obtained by averaging the empirical distribution $\hat{p}$ within each partition element $\alpha \in \mathcal{A}$. Let $\bar{\mathcal{A}}^*$ denote the optimal partition:

$$\bar{\mathcal{A}}^* \triangleq \operatorname*{argmax}_{\mathcal{A}\text{: partition of } \mathcal{X}} \bar{V}(\mathcal{A}) \tag{40}$$

Now optimizing the lower bound to the permanent $\bar{V}(p)$ (31) is equivalent to optimizing $\bar{V}(\mathcal{A})$ (38) over partitions of $\mathcal{X}$, since $\bar{V}(\bar{p}^*) = \bar{V}(\bar{\mathcal{A}}^*)$ and $\bar{p}^* = \hat{p}_{\bar{\mathcal{A}}^*}$.

We make two observations for the solution of the approximate PML problem (40) that allow us to restrict the set of partitions over which we optimize. Let $\mathcal{A}$ be a partition of $\mathcal{X}$. We say $\mathcal{A}$ has the *iso-clumping* or *convexity* properties defined below:

1. *Iso-clumping* property. Symbols with the same empirical probabilities are clumped together: For all $x, x' \in \mathcal{X}$:

$$\hat{p}_x = \hat{p}_{x'} \Rightarrow \alpha(x) = \alpha(x') \tag{41}$$

   Where $\alpha(x) \in \mathcal{A}$ is the partition element containing x.

2. *Convexity* property. For all $x, x', x'' \in \mathcal{X}$:

$$\Big(\hat{p}_x < \hat{p}_{x'} < \hat{p}_{x''} \text{ and } \alpha(x) = \alpha(x'')\Big) \Rightarrow \alpha(x) = \alpha(x') = \alpha(x'') \tag{42}$$

Theorem 9 *Let $\bar{\mathcal{A}}^*$ (40) be the partition of $\mathcal{X}$ into level sets of the approximate PML distribution $\bar{p}^*$ (33) and let $\hat{p}$ be the empirical distribution. Then $\bar{\mathcal{A}}^*$ has the iso-clumping and convexity properties (41) and (42).*

See Appendix G for a proof.

4.5. A dynamic programming computation of the approximate PML distribution

Theorem 9 lets us efficiently maximize $\bar{V}(\mathcal{A})$ (38) by restricting to only those partitions of $\mathcal{X}$ that have the iso-clumping and convexity properties. Once we find $\bar{\mathcal{A}}^*$ (40), the averaging property (37) lets us compute the approximate PML distribution $\bar{p}^*$.

Let $\mathcal{F}_+ \triangleq (\mathcal{F}_i)_{i \geq 1}$. Let

$$0 = m_0 < m_1 < m_2 < \ldots < m_{F_+} \leq n \tag{43}$$

where

$$\mathrm{Supp}(\mathcal{F}_+) = \{m \geq 1 : \mathcal{F}_m > 0\} = \{m_i\}_{i=1}^{F_+} \tag{44}$$

Then any distribution whose level set partition satisfies the iso-clumping (41) and convexity (42) properties has all level sets of the form $\mathcal{X}_{i:j}$ with $0 \leq i \leq j \leq F_+$:

$$\mathcal{X}_{i:j} \triangleq \{x \in \mathcal{X} : m_i \leq n\hat{p}_x \leq m_j\}. \tag{45}$$

Thus we optimize $\bar{V}(\mathcal{A})$ (38) over partitions of $\mathcal{X}$ into level sets of the form (45). Note that if $\mathcal{F}_0 > 0$, then the set $\mathcal{X}_{0:0} = \{x \in \mathcal{X} : \hat{p}_x = 0\}$ of unseen symbols can not appear in the optimal partition $\bar{\mathcal{A}}^*$ because if it does, then its probability mass under $p_{\bar{\mathcal{A}}^*}$ is 0, violating our assumption that p has support $\mathcal{X}$. If $\mathcal{F}_0 = 0$, then $\mathcal{X}_{0:0} = \emptyset$.

This optimization can be done by a dynamic programming algorithm. For integers $i \leq j$, let $[i, j] \triangleq \{k \in \mathbb{N} : i \leq k \leq j\}$. Let $\bar{V}_i$ denote for $0 \leq i \leq F_+$:

$$\bar{V}_i \triangleq \max_{j \in [i \vee 1, F_+]} (\bar{V}(\mathcal{X}_{i:j}) + \bar{V}_{j+1}) \tag{46}$$

$$\overset{\text{(a)}}{=} \max_{\mathcal{A}\text{: partition of } \mathcal{X}_{i:F_+}} \bar{V}(\mathcal{A}) \tag{47}$$

with boundary condition $\bar{V}_{F_+ + 1} \triangleq 0$, where $i \vee j$ denotes the greater of i and j, and where $\bar{V}(\mathcal{X}_{i:j})$ is as in (39). (a) follows by induction on i downwards from F_+. We let $j \in [i \vee 1, F_+]$ in (46) (rather than $j \in [i, F_+]$) in order to optimize over only those partitions of $\mathcal{X}$ that do not contain the set $\mathcal{X}_{0:0}$ of unseen symbols; this restriction forces the approximate PML distribution to have support set size $|\mathcal{X}|$, as we assumed in the beginning of Section 4.3.

Then the PML distribution approximation $\bar{p}^* = \operatorname{argmax}_p \bar{V}(p)$ satisfies (setting $i = 0$ and using $\mathcal{X}_{0:F_+} = \mathcal{X}$ in (47)):

$$\bar{V}(\bar{p}^*) = \max_{\mathcal{A}\text{: partition of } \mathcal{X}} \bar{V}(\mathcal{A}) = \bar{V}_0 \tag{48}$$

We can compute the term $\bar{V}(\mathcal{X}_{i:j})$ in (46), corresponding to clustering all symbols of $\mathcal{X}_{i:j}$ into a level set of p, using (39):

$$\bar{V}(\mathcal{X}_{i:j}) = \log(|\mathcal{X}_{i:j}|!) + n|\mathcal{X}_{i:j}|\hat{p}_{\mathcal{X}_{i:j}} \log\left(\hat{p}_{\mathcal{X}_{i:j}}\right) \tag{49}$$

$$= \log\left(\Big(\sum_{k=i}^{j} \mathcal{F}_{m_k}\Big)!\right) + \left(\sum_{k=i}^{j} m_k \mathcal{F}_{m_k}\right) \log\left(\frac{\sum_{k=i}^{j} m_k \mathcal{F}_{m_k}}{n \sum_{k=i}^{j} \mathcal{F}_{m_k}}\right) \tag{50}$$

where we used $|\mathcal{X}_{i:j}| = \sum_{k=i}^{j} \mathcal{F}_{m_k}$ and $\hat{p}_{\mathcal{X}_{i:j}} = \frac{1}{n|\mathcal{X}_{i:j}|} \sum_{k=i}^{j} m_k \mathcal{F}_{m_k}$.

Relations (46) and (50) allow us to compute the level set decomposition $\bar{p}^*$ by filling out a $(F_+ + 1) \times F_+$ array $(\bar{V}(\mathcal{X}_{i:j}))_{i\in[0,F_+],j\in[1,F_+]}$ and keeping track of the maximizing index each time we compute $\bar{V}_i$. Once we have the optimal level set decomposition $\bar{\mathcal{A}}^* = \{\mathcal{X}_{i:j}\}$ of $\bar{p}^*$, we set for all $x \in \mathcal{X}_{i:j}$, for all i, j

$$\bar{p}^*_x = \hat{p}_{\mathcal{X}_{i:j}} \tag{51}$$

For example, in Figure 1, we have $(m_i)_{i=0}^{F_+=12} = (0, 1, 2, 3, 4, 5, 7, 9, 12, 13, 15, 16, 20)$ and the approximate PML distribution $\bar{p}^*$ (red line) has level sets $\{\mathcal{X}_{0:6}, \mathcal{X}_{7:12}\}$.

The running time of this dynamic programming algorithm is $O(|\operatorname{Supp}(\mathcal{F})|^2)$, where $|\operatorname{Supp}(\mathcal{F})| \le F_+ + 1$. In terms of the sample size n, any empirical distribution $\hat{p}$ satisfies $|\operatorname{Supp}(\mathcal{F}(\hat{p}))| \le \frac{1}{2}\left(1 + \sqrt{8n+1}\right)$, with equality achieved by the empirical distribution $\hat{p}_i = \frac{2i}{|\mathcal{X}|(|\mathcal{X}|+1)}$ for $i \in [0, |\mathcal{X}|]$ with $|\mathcal{X}| = |\operatorname{Supp}(\mathcal{F})|$ and $n = \frac{1}{2}|\mathcal{X}|(|\mathcal{X}|+1)$. Thus the worst-case run time for the dynamic programming algorithm is $O(\sqrt{n}^2) = O(n)$. We "usually" have $|\operatorname{Supp}(\mathcal{F})|$ much smaller than $\sqrt{n}$, so this is a pessimistic estimate for a typical case. The run time of our approximate PML scheme is usually dominated by computation of $\hat{p}$. An implementation optimization is to pre-compute the sums $\sum_{k=i}^{j} \mathcal{F}_{m_k}$ and $\sum_{k=i}^{j} m_k \mathcal{F}_{m_k}$ for all $i \le j$ in computing $\bar{V}(\mathcal{X}_{i:j})$.

4.6. Unknown or infinite support set size

If the support set size $K = |\mathcal{X}|$ is unknown, then we can attempt to infer it. Our estimator $\bar{K}^*$ for the support set size is the same as (17), but replaces the permanent with our lower bound (31), $e^{\bar{V}(p)} \le \operatorname{perm}(Q)$:

$$\bar{K}^* \triangleq \operatorname*{argmax}_K \left(\frac{1}{(K-\hat{K})!} \max_{p\in\mathcal{P}_K} e^{\bar{V}(p)} \right) \tag{52}$$

whenever the max over K exists, where $\mathcal{P}_K \triangleq \{p \in \mathcal{P} : |\operatorname{Supp}(p)| = K\}$, where $\hat{K} = |\hat{\mathcal{X}}|$ is the support set size of the empirical distribution $\hat{p}$. Then the PML distribution is $\bar{p}^*$ with support set $\bar{K}^*$. The max over p is obtained via our approximate PML algorithm in Section 4.5. The maximizer $\bar{K}^*$ usually exists because a larger K boosts the value of the log permanent lower bound $\bar{V}(p)$ (since there are more permutations to sum over), but reduces the first factor in (52). It may happen that $\bar{K}^*$ does not exist, in which case we say our approximate PML distribution has a continuous part in the terminology of (Orlitsky et al., 2004b). In general the function in the argmax in (52) is multimodal in K.

It turns out we can efficiently compute $\bar{K}^*$ if it exists and compute the corresponding approximate PML distribution; if $\bar{K}^*$ does not exist, then we can efficiently detect this case and compute the corresponding approximate PML distribution, which in this case has a continuous part.

We state and derive our observations in Appendix J. The idea is to optimize over all possible ways to clump the unseen symbols $\mathcal{X}_{0:0}$ with the other symbols. Since the optimal clumping satisfies the iso-clumping and convexity properties, we show that there are only $|\operatorname{Supp}(\mathcal{F}_+)| \le \sqrt{2n} + 1$ possibilities to check and that the value of each possibility can be checked quickly, reusing the work done in the dynamic programming approach described in the previous Section, so this can be done efficiently.

4.7. Approximate PML: the case of multiple distributions (see Appendix H)

We generalize the PML distribution and approximate PML distribution to the case of $(n_d)_{d=1}^D$ samples drawn from distributions $(p^{(d)})_{d=1}^D$ on the same alphabet (see Section 3.2 for the case $D = 2$, used in estimating functionals of pairs of distributions). This requires some more notation that we delegate to Appendix H.

For $D \ge 2$, we are unable to give an efficient algorithm to compute the D-dimensional approximate PML. The difficulty is due to the lack of a natural ordering on $\mathbb{N}^D$ for $D \ge 2$, so the dynamic programming approach we use for $D = 1$ does not work for $D \ge 2$. We settle for approximately computing our the approximate PML distribution via a greedy heuristic: we iteratively enlarge level sets of a candidate solution by merging pairs of distinct level sets until the objective function $\bar{V}$ (suitably generalized to D distributions) stops increasing. See Appendix H for details.

4.8. Comparison to the (exact) PML distribution for size 2 alphabet

Recall the discussion of Section 4.2.1 on the PML distribution for alphabet size $|\mathcal{X}| = 2$. A computation establishes the condition for uniformity of the APML distribution:

$$\bar{p}^* = \begin{cases} \left(\frac{1}{2}, \frac{1}{2}\right) & : H(\hat{p}) \ge \log(2)\left(1 - \frac{1}{n}\right) \\ (\hat{p}_1, 1 - \hat{p}_1) & : \text{otherwise} \end{cases} \tag{53}$$

where $H(\hat{p})$ is the entropy of the empirical distribution and n is the sample size.

Let's consider the large n limit. Let $\hat{p}_1^{\text{unif}}$ be the critical value at which the APML distribution switches between uniform and nonuniform (the least value of $\hat{p}_1$ satisfying the uniformity condition in (53)). Then a computation shows that

$$\hat{p}_1^{\text{unif}} = \frac{1}{2}\left(1 - \sqrt{\frac{\log(4)}{n}} + O\left(\frac{1}{n}\right)\right) \tag{54}$$

Comparing this to condition (26) for the (exact) PML, we see that in the APML case, $\hat{p}_1^{\text{unif}}$ carries an extra factor of $\sqrt{\log(4)} \approx 1.18$ in the $1/\sqrt{n}$ term relative to the uniform/nonuniform critical value of $\hat{p}_1$ for the PML.

Figure 6 compares the APML and PML distributions for a size-2 alphabet. Both the APML and PML are uniform when $|\hat{p}_1 - \hat{p}_2|$ is small enough, though at different points, as remarked above. Whenever the APML solution is not uniform, it exactly matches (up to relabeling of components) the empirical distribution, unlike the PML.

5. Estimating symmetric functionals of a single discrete distribution

In the standard PML setting discussed in Section 3.1, we argued the sufficiency of the fingerprint $\mathcal{F}$ (11) in estimating the sorted probability vector of a discrete distribution, which implies the sufficiency of the fingerprint for estimating any functional F of the sorted probability vector of the form

$$F(p) = G\Big(\sum_{x\in\mathcal{X}} f(p_x)\Big), \tag{55}$$

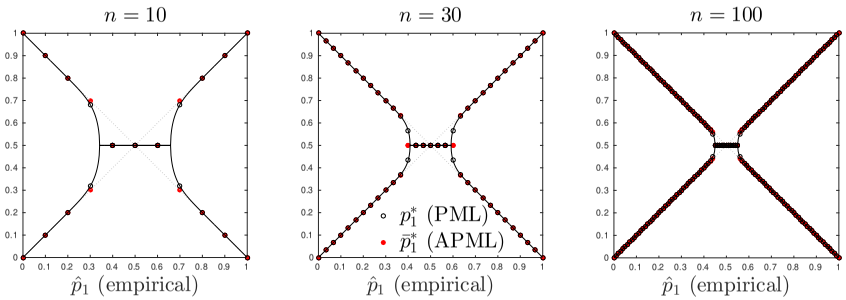


Figure 6: "Phase diagram" for the exact PML (hollow circles) and APML distributions (red circles) on binary alphabet with sample size n indicated above each subplot (compare to Figure 3). Solid black lines correspond to the solution for arbitrary $\hat{p}_1 \in [0,1]$, obtained numerically. The PML distribution p^* is uniform when $|\hat{p}_1 - \hat{p}_2| \le 1/\sqrt{n}$, while the APML distribution is uniform when $|\hat{p}_1 - \hat{p}_2| \le \sqrt{\log(4)}/\sqrt{n} + O(1/n)$. Outside this middle region there are two PML (APML) branches, since any permutation of the PML (APML) distribution is another PML (APML) distribution. The APML distribution exactly matches the empirical distribution outside the uniform region. Diagonal dashed lines correspond to the lines $(\hat{p}_1, \hat{p}_1)$ and $(\hat{p}_1, 1-\hat{p}_1)$.

where we constrain $f(0) = 0$ to accommodate the unknown alphabet setting. Note that the fingerprint is also partially sufficient for functionals of type $\sum_{x,y\in\mathcal{X}} f(p_x, p_y)$, etc.

In order to apply Theorem 3, one also needs to show the cardinality of the fingerprint $\mathcal{F}$ is small. The fingerprint $\mathcal{F} = (\mathcal{F}_i)_{i\ge 0}$ satisfies

$$\sum_{i\ge 0} i\mathcal{F}_i = n, \tag{56}$$

and each fingerprint corresponds to an unordered integer partition of the integer n. We now recall the Hardy-Ramanujan result on integer partitions.

Theorem 10 *(Hardy and Ramanujan, 1918)(Bárány and Vershik, 1992, Theorem 2) The cardinality of the set of fingerprints (11) on n samples is given by*

$$e^{\pi\sqrt{\frac{2n}{3}}(1-o(1))} \le |\{(\mathcal{F}_i)_{i\ge 1}\}| \le e^{\pi\sqrt{\frac{2n}{3}}}. \tag{57}$$

The key observation is that the cardinality of the fingerprint is sub-exponential. Then, combining with Theorem 3 and the fact that there exist estimators for various symmetric functionals with near exponential measure concentration, (Acharya et al., 2017) showed that plugging in the PML achieves the optimal sample complexity in estimating the Shannon entropy, the support size, the support coverage, and the L_1 distance to uniformity.

In this section, we extensively test the performance of symmetric functional estimation via plugging in our approximate PML (APML) distribution into the functionals. Recall

that the PML distribution p^* is defined as in (16) for empirical distribution $\hat{p}$ on n samples with fingerprint $\mathcal{F}(\hat{p})$:

$$p^* = \operatorname*{argmax}_{p\in\mathcal{P}} \mathbb{P}_p(\mathcal{F}) = \operatorname*{argmax}_{p\in\mathcal{P}} \frac{\operatorname{perm}\left(\left(p_x^{n\hat{p}_{x'}}\right)_{x,x'\in\mathcal{X}}\right)}{(K-\hat{K})!}. \tag{58}$$

where the optimization is over distributions of possibly different support set size, $\mathcal{X}$ denotes the support set of distribution p, $K = |\mathcal{X}|$, and $\hat{K}$ denotes the support set size of the empirical distribution $\hat{p}$. p^* is difficult to compute exactly, so we compute the APML distribution $\bar{p}^*$ (33), (52):

$$\bar{p}^* = \operatorname*{argmax}_{p\in\mathcal{P}} \frac{e^{\bar{V}(p)}}{(K-\hat{K})!} \tag{59}$$

which maximizes a lower bound to the probability $\mathbb{P}_p(\mathcal{F})$ since $\bar{V}(p)$ (31) lower bounds the log permanent in (58). Then our estimator for function F of the form (55) is the plugin $F(\bar{p}^*)$.

Our approximate PML estimator performs comparably well to the competition (Valiant and Valiant, 2011a) (Valiant and Valiant, 2013) (Jiao et al., 2015), (Wu and Yang, 2016), (Wu and Yang, 2015) across different functions F and distributions, and performs significantly better when the true distribution is uniform. This good performance for the uniform case makes some intuitive sense: the approximate PML distribution $\bar{p}^*$ maximizes a lower bound (22) to a matrix permanent obtained by summing over only those permutations that mix symbols within level sets of $\bar{p}^*$. If there is only one level set – that is, if $\bar{p}^*$ is uniform – then we have exactly optimized the matrix permanent (or at least found a local maximum), rather than a lower bound to the permanent.

5.1. Sorted L_1 loss

Given n samples from a distribution p, the usual L_1 loss in measuring the accuracy of estimating p is given by

$$\sum_{x\in\mathcal{X}} |p_x - q_x|. \tag{60}$$

The sorted L_1 loss measures the *shape* difference between the reconstruction distribution q and the true distribution p. In other words, we first sort the distributions p, q into non-increasing probability vectors $(p_{(1)}, p_{(2)}, \ldots, p_{(K)}), (q_{(1)}, q_{(2)}, \ldots, q_{(K)})$, and then measure the loss by

$$\sum_{1\le i\le K} |p_{(i)} - q_{(i)}|. \tag{61}$$

The sorted L_1 loss measures the error in reconstructing the sorted probability vector, for which the fingerprints are partially sufficient as shown by Theorem 6. The sorted L_1 loss can also be viewed as a Wasserstein distance (Vallender, 1974), i.e.,

$$\sum_{1\le i\le K} |p_{(i)} - q_{(i)}| = K \sup_{f\in\mathsf{Lip}_1} \int f(x) \left(\sum_x \frac{1}{K}\delta_{p_x} - \sum_x \frac{1}{K}\delta_{q_x}\right), \tag{62}$$

where the supremum is over all Lipschitz functions with Lipschitz constant one.

We estimate the sorted L_1 loss using the plug-in $q = \bar{p}^*$, the APML distribution (59), computed over collection of distributions $\mathcal{P} = \Delta$, where

$$\Delta \triangleq \{p : |\operatorname{Supp}(p)| < \infty\} \tag{63}$$

denotes the set of all discrete distributions with finite support set size, and collection of distributions $\mathcal{P} = \Delta_K$, where

$$\Delta_K \triangleq \{p : |\operatorname{Supp}(p)| = K\} \tag{64}$$

denotes the set of all discrete distributions with support set size K.

Figure 7 shows the sorted probability vector inferred by our approximate PML distribution in Δ (63) – that is, the case of unknown support set size – along with the sorted ML (empirical) distribution and the distribution of (Valiant and Valiant, 2013). We see that both the approximate PML distribution and the distribution of (Valiant and Valiant, 2013) perform much better than the sorted ML distribution, and that the PML distribution "prefers" to have fewer and larger level sets than the distribution of (Valiant and Valiant, 2013).

Figure 8 shows the performance of our approximate PML distribution in Δ_K (64) – that is, the case of known support set size – as a plugin estimator for the sorted L_1 loss, along with the estimator of (Valiant and Valiant, 2011a) and the empirical distribution plugin estimator (MLE) (see caption for performance test parameters). We see that the approximate PML distribution performs well for the uniform and mixture-of-two-uniforms distributions, but more poorly for the Zipf distributions, consistent with the remarks in the previous Section. Good performance on near-uniform distributions is observed for other performance tests as well (see below).

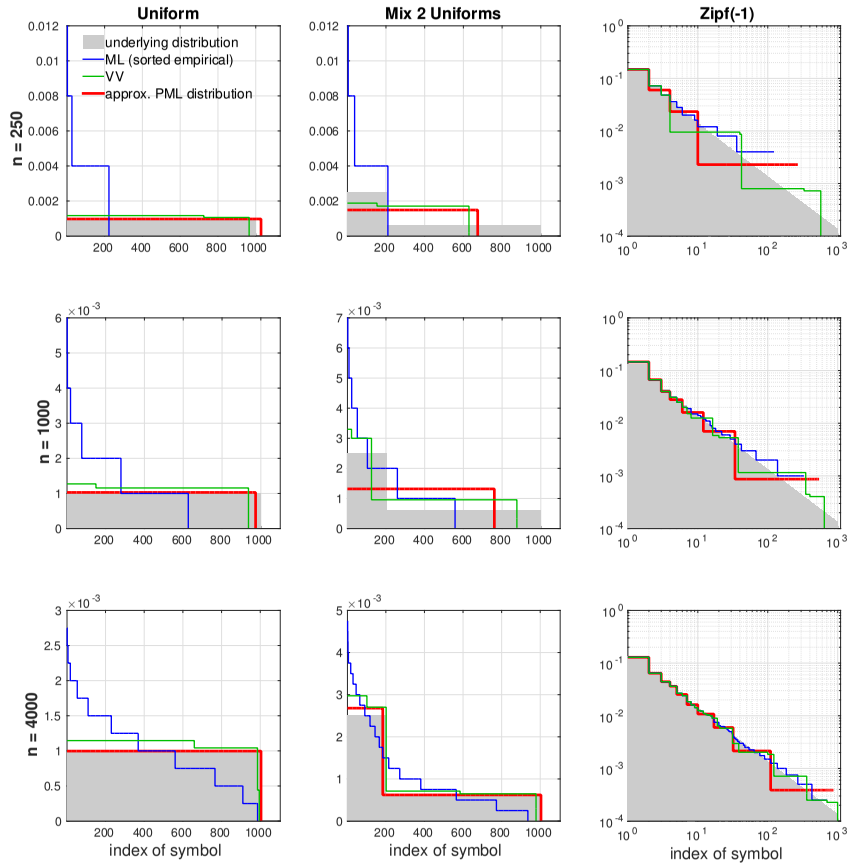


Figure 7: Inferring the sorted probability vector given samples from an underlying distribution (plotted in gray). The distribution is held constant in each column and the sample size n is held constant in each row of the figure. In all cases $K = |\mathcal{X}| = 1000$. "Uniform" is uniform on $\mathcal{X}$, "Mix 2 Uniforms" is a mixture of two uniform distributions, with half the probability mass on the first $K/5$ symbols, and the other half on the remaining symbols, and Zipf(α) $\sim 1/i^\alpha$ with $i \in \{1, \ldots, K\}$. ML denotes the sorted empirical distribution. VV is (Valiant and Valiant, 2013).

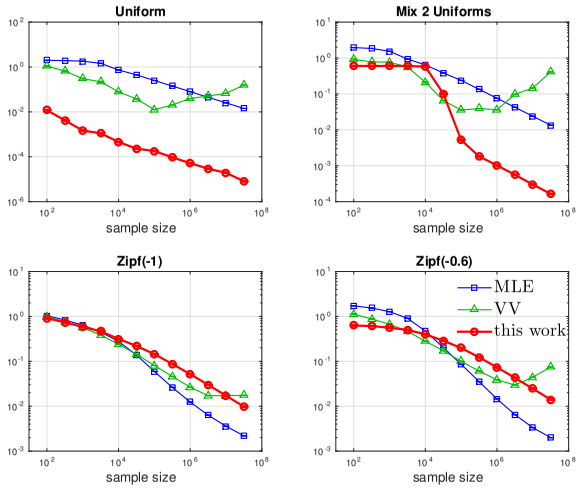


Figure 8: Sorted L_1 loss in estimating an unknown distribution p with known support set size $K = |\mathcal{X}|$. In all cases $K = 10^4$. "Uniform" is uniform on $\mathcal{X}$, "Mix 2 Uniforms" is a mixture of two uniform distributions, with half the probability mass on the first $K/5$ symbols, and the other half on the remaining symbols, and Zipf(α) $\sim 1/i^\alpha$ with $i \in \{1, \ldots, K\}$. MLE denotes the ML plugin ("naive") approach of using the sorted empirical distribution in (61). VV is (Valiant and Valiant, 2013). Each data point represents 100 random trials, with 2 standard error bars smaller than the plot marker for most points.

5.2. Entropy and Rényi entropy estimation

For entropy $H(p) = -\sum_{x\in\mathcal{X}} p_x \log(p_x)$ and Rényi entropy $H_\alpha(p) = \frac{1}{1-\alpha}\log\left(\sum_{x\in\mathcal{X}} p_x^\alpha\right)$ of distribution p, the corresponding approximate PML estimator is defined as the plugin estimator $H(\bar{p}^*)$, $H_\alpha(\bar{p}^*)$, where $\bar{p}^*$ is as in (59), optimized over the collection of distributions $\mathcal{P} = \Delta$ (63) for both the entropy and Rényi entropy. Additionally, to enable direct comparison with the entropy estimator of (Wu and Yang, 2016), which requires a support set size as input, we also find the approximate PML distribution optimized over $\mathcal{P} = \Delta_K$ (64).

Figure 9 shows the performance of our approximate PML scheme for estimating the entropy and Rényi entropy with $\alpha \in \{2, 1.5, 0.8\}$. Overall, our approximation looks competitive with the other approaches, and is particularly strong for the uniform distribution. Adding in knowledge of the true support set size in estimating the entropy did not make much of a difference, except for improvement in the case of the uniform distribution.

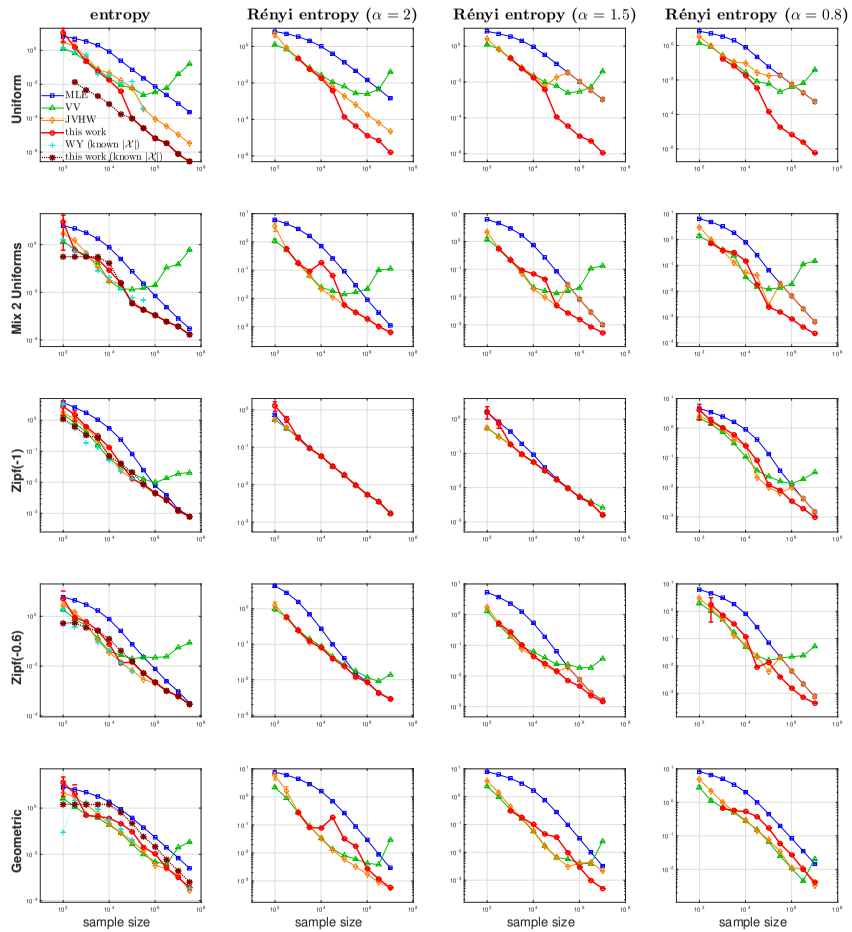


Figure 9: Root mean squared error of several estimators of entropy (first column) and Rényi entropy (last three columns for $\alpha \in \{2, 1.5, 0.8\}$). In all cases except "Geometric" we set the alphabet size $|\mathcal{X}| = 10^4$. "Uniform" is uniform on $\mathcal{X}$, Zipf(α) $\sim 1/i^\alpha$ with $i \in \{1, \ldots, K\}$, "Geometric" is the geometric distribution with infinite support and mean K. MLE denotes the ML plugin ("naive") approach of computing $H(\hat{p})$. VV is (Valiant and Valiant, 2013). JVHW is (Jiao et al., 2015). WY is (Wu and Yang, 2016) – this estimator requires the support set size $|\mathcal{X}|$ as an input. Our estimator optionally accepts the support set size $|\mathcal{X}|$ (shown in dark red). Each data point represents 100 random trials, with 2 standard error bars smaller than the plot marker for most points. Log base 2.

5.3. L_1 distance to uniformity

For estimating the L_1 distance to uniformity, which is $\sum_{x\in\mathcal{X}} |p_x - \frac{1}{K}|$, the corresponding PML estimator is (Acharya et al., 2017) the plugin of p^* (58) optimized over the collection of distributions $\mathcal{P} = \Delta_K$ (64) (the case of known support set size). The APML estimator is the plugin of $\bar{p}^*$ (59) optimized over $\mathcal{P} = \Delta_K$.

Figure 10 shows the performance of our approximate PML scheme for estimating the L_1 distance to a uniform distribution with known support set size. Our approach looks competitive, and is by far the best if the true distribution is uniform.

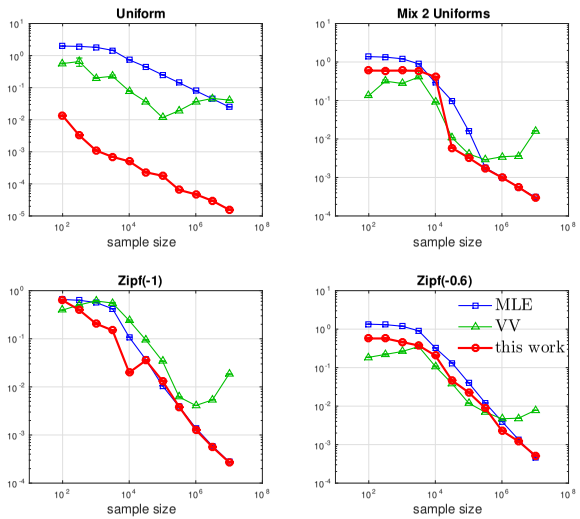


Figure 10: Root mean squared error for estimation of L_1 distance to uniformity with known support set size $K = |\mathcal{X}|$. In all cases $K = 10^4$. "Uniform" is uniform on $\mathcal{X}$, "Mix 2 Uniforms" is a mixture of two uniform distributions, with half the probability mass on the first $K/5$ symbols, and the other half on the remaining symbols, and Zipf(α) $\sim 1/i^\alpha$ with $i \in \{1,\ldots,K\}$. MLE denotes the ML plugin ("naive") approach of computing $\sum_{x\in\mathcal{X}} |\hat{p}_x - \frac{1}{K}|$. VV is (Valiant and Valiant, 2013). Each data point represents 100 random trials, with 2 standard error bars smaller than the plot marker for most points.

5.4. Support set size estimation

For support size estimation, the corresponding PML estimator is (Acharya et al., 2017) the plugin of p^* (58) optimized over the collection of distributions $\mathcal{P} = \Delta_{\geq \frac{1}{K}}$, where

$$\Delta_{\geq \frac{1}{K}} \triangleq \{p : p_x \geq \frac{1}{K}\ \forall x \in \mathcal{X}\} \tag{65}$$

denotes the set of all discrete distributions whose minimum nonzero probability for each symbol is at least $\frac{1}{K}$. The approximate PML estimator is the plugin of $\bar{p}^*$ (59) optimized over $\mathcal{P} = \Delta_{\geq \frac{1}{K}}$.

Figure 11 shows the performance of our approximate PML scheme for estimating the support set size. Here we plot the mean and standard error of the support set size inferred by the different estimation schemes rather than a root mean squared error. Overall, our approach is comparable to the others, performing worst on the Zipf distributions, and best on the uniform distribution.

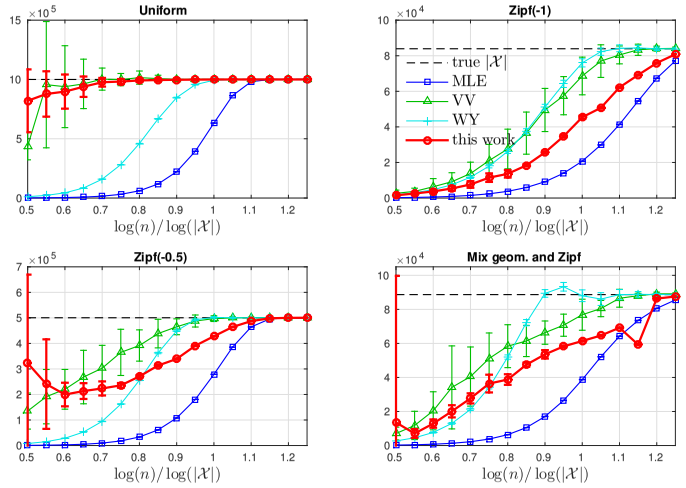


Figure 11: Support set size estimation for the same cases as in (Wu and Yang, 2015), plotting the mean and 2 standard deviations of the distribution of inferred support set sizes in 100 trials vs. sample size n. In all cases the support set set size $K = |\mathcal{X}|$ is chosen such that $\min_{i\in\{1,\ldots,K\}} p_i \approx 10^{-6}$. "Uniform" is uniform on $\{1,\ldots,K\}$, Zipf(α) $\sim 1/i^\alpha$ with $i \in \{1,\ldots,K\}$, and "Mix geom. and Zipf" is a mixture of $p_i \sim 1/i$ for $i \in \{1,\ldots,K/2\}$ and $p_i \sim (1-2/K)^{i-K/2-1}$ for $i \in \{K/2+1,\ldots,K\}$ normalized so that half the mass is on the first $K/2$ symbols. MLE denotes the ML plugin ("naive") approach of using the empirical support set size. VV is (Valiant and Valiant, 2013). WY is (Wu and Yang, 2015).

6. Estimating the divergence function between two distributions

In the 2-D PML setting discussed in Section 3.2, we argued the sufficiency of the fingerprint (18) in estimating functions of two discrete distributions of the form

$$F(p,q) = \sum_{x\in\mathcal{X}} f(p_x, q_x). \tag{66}$$

In order to apply Theorem 3, one needs to show the cardinality of the sufficient statistic is not too big. Below we present an argument that applies to the D-dimensional PML. Suppose we have D discrete distributions on the same alphabet $\mathcal{X}$, denoted as $p^{(1)}, p^{(2)}, \ldots, p^{(D)}$. Suppose we obtain n_d samples from each distribution $p^{(d)}$ with empirical distribution $\hat{p}^{(d)}$. Let the joint fingerprint be defined for every $(i_1, i_2, \ldots, i_D) \geq 0$,

$$\mathcal{F}_{i_1,i_2,\ldots,i_D} = |\{x \in \mathcal{X} : (n_d \hat{p}_x^{(d)})_{d=1}^D = (i_d)_{d=1}^D\}|. \tag{67}$$

We now argue that the joint fingerprint $(\mathcal{F}_{i_1,i_2,\ldots,i_D})_{(i_1,i_2,\ldots,i_D)\geq 0}$ corresponds to the partition of multipartite numbers. Indeed, the joint fingerprint satisfies the following equation:

$$\sum_{(i_1,i_2,\ldots,i_D)\geq 0} \mathcal{F}_{i_1,i_2,\ldots,i_D} \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_D \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_D \end{bmatrix}. \tag{68}$$

Theorem 11 quantifies the size of the sufficient statistic.

Theorem 11 *(Auluck, 1953; Bárány and Vershik, 1992; Acharya et al., 2011) Suppose $D = 2$. If $n_1 = n_2 = n$, then the cardinality of the set of 2-D fingerprints on n samples is given by*

$$|\{(\mathcal{F}_{ij})_{(i,j)\neq(0,0)}\}| = e^{3(\zeta(3))^{1/3} n^{2/3}(1+o(1))}. \tag{69}$$

For general $D > 2$, if $n_d \geq 2^{D+1}, 1 \leq d \leq D$, we have

$$|\{(\mathcal{F}_{i_1,i_2,\ldots,i_D})_{(i_1,i_2,\ldots,i_D)\neq\mathbf{0}}\}| \leq \exp\left(2\left(1+\frac{1}{D}\right)\sum_{d=1}^{D} n_d^{\frac{D}{D+1}}\right). \tag{70}$$

where $\mathbf{0} = (0,\ldots,0)$. Moreover, if $n_d = n$ for all $d \in \{1,\ldots,D\}$, we have

$$|\{(\mathcal{F}_{i_1,i_2,\ldots,i_D})_{(i_1,i_2,\ldots,i_D)\neq\mathbf{0}}\}| = \exp\left((D+1)(\zeta(D+1))^{1/(D+1)} n^{\frac{D}{D+1}}(1-o(1))\right). \tag{71}$$

Here $\zeta(D+1) = \sum_{k=1}^{\infty} k^{-(D+1)}$ is the Riemann zeta function.

Fortunately, the cardinality of the D-dimensional fingerprint is still sub-exponential in the sample size for all D. Then, combining with Theorem 3 and the fact that there exist estimators for the aforementioned divergence functions with near exponential measure concentration (which can be shown in a straightforward manner as in (Acharya et al., 2017; Han et al., 2016a,b; Bu et al., 2016; Valiant and Valiant, 2011a; Jiao et al., 2018)), we know that plugging in the 2-D PML achieves the optimal sample complexity in estimating the Kullback-Leibler divergence, L_1 distance, the squared Hellinger distance, and the χ^2 divergence. The proof of these results is to be reported elsewhere.

In this section, we extensively test the performance of divergence estimation via plugging in our approximate 2-D PML. We mainly compete with the released code of KL divergence estimator in (Han et al., 2016a,b). The concrete divergence functional estimation algorithm is as follows. Suppose we observe n samples from distribution p with empirical distribution $\hat{p}$ and m samples from distribution q with empirical distribution $\hat{q}$. Let the 2-D PML estimator be as (20):

$$(p^*, q^*) = \operatorname*{argmax}_{(p,q)\in\mathcal{P}} \mathbb{P}_{p,q}(\mathcal{F}), \tag{72}$$

where $\mathcal{F}$ is the 2-D fingerprint (18) and $\mathcal{P}$ is a collection of pairs of distributions on the same alphabet $\mathcal{X}$. The APML distributions maximize a lower bound to $\mathbb{P}_{p,q}(\mathcal{F})$:

$$(\bar{p}^*, \bar{q}^*) = \operatorname*{argmax}_{(p,q)\in\mathcal{P}} \frac{e^{\bar{V}(p,q)}}{(K-\hat{K})!} \tag{73}$$

where $\bar{V}$ is defined for the D-dimensional case in Appendix H, $\mathcal{X} = \mathrm{Supp}(p) \cup \mathrm{Supp}(q)$, $K = |\mathcal{X}|$, and $\hat{K} = |\mathrm{Supp}(\hat{p}) \cup \mathrm{Supp}(q)|$.

As discussed in Section 4.7, there is no natural ordering on the bins of the 2-D fingerprint, so we do not give a dynamic programming algorithm like the one in section 4.5 to find the APML distributions. Instead we use a greedy heuristic presented in Appendix H to approximately maximize $\bar{V}(p,q)$ and approximate the already-approximate PML distributions:

$$(\bar{\bar{p}}^*, \bar{\bar{q}}^*) \approx (\bar{p}^*, \bar{q}^*) \tag{74}$$

Then our estimator for functional F of the form (66) is $F(\bar{\bar{p}}^*, \bar{\bar{q}}^*)$.

Our approximate PML estimator performs overall best relative to the competition, which for the KL divergence consists of only (Han et al., 2016a,b) and the MLE plugin estimator, and for the L_1 distance consists of only the MLE plugin estimator. (Valiant and Valiant, 2013), (Valiant, 2008) generalize their approach to the 2-D fingerprint setting, but do not release code to repeat their experiments.

6.1. KL divergence estimation

For the KL divergence $D(p||q) = \sum_{x\in\mathcal{X}} p_x \log\left(\frac{p_x}{q_x}\right)$, the corresponding approximate PML estimator is the plugin $D(\bar{\bar{p}}^*||\bar{\bar{q}}^*)$, where $\bar{\bar{p}}^*, \bar{\bar{q}}^*$ are as in (74), optimized over collection of distributions $\mathcal{P} = \Delta_\rho$, where

$$\Delta_\rho = \{(p,q) : \sup_x (p_x/q_x) \leq \rho\} \tag{75}$$

denotes the set of all pairs of discrete distributions with finite support and maximum ratio ρ.

Figure 12 shows the performance of our approximate PML scheme for estimating the KL divergence. The approximate PML estimator looks mostly better than (Han et al., 2016a,b)

and the MLE. All three estimators show non-monotonicity of their root mean squared performance in the sample size. Qualitatively, the approximate PML estimator's performance looks best-behaved among the three in terms of being roughly monotone decaying in the sample size.

6.2. L_1 distance

For the L_1 distance $\|p-q\|_1 = \sum_{x\in\mathcal{X}} |p_x - q_x|$, the corresponding approximate PML estimator is the plugin $\|\bar{\bar{p}}^* - \bar{\bar{q}}^*\|_1$, where $\bar{\bar{p}}^*, \bar{\bar{q}}^*$ are as in (74), optimized over collection of distributions $\mathcal{P} = \Delta$, where Δ denotes the set of all pairs of discrete distributions with finite support set size (possibly different support set sizes within the pair).

Figure 13 shows the performance of our approximate PML scheme for estimating the L_1 distance between two distributions. The approximate PML estimator looks overall stronger than the MLE plugin estimator for all distributions and sample sizes, and seems to perform best when the two unknown distributions p and q are the same (corresponding to the diagonal of the matrix of plots in the figure). Although (Valiant and Valiant, 2013) did not release their code for estimating the L_1 distance, their Figure 2, leftmost pane corresponds exactly in its parameters to the top-leftmost pane of our Figure 13; from roughly checking a few sample sizes visually, the approximate PML seems to perform significantly better.

Acknowledgments

We are grateful to Huacheng Yu for providing the proof of Theorem 12. We would like to thank Jayadev Acharya, Nick Loehr, Greta Panova, Richard Stanley, and Arthur Yang for helpful discussions on the size of the sufficient statistic.

Appendix A. Proof of expression (14) for the probability of a fingerprint

Let $\mathcal{S}_\mathcal{X}$ denote the symmetric group on the discrete, possibly infinite set $\mathcal{X}$ and let $\mathcal{S}_\mathcal{X}\hat{p}$ denote the orbit of empirical distribution $\hat{p}$ under the action of $\mathcal{S}_\mathcal{X}$ by relabeling the components of $\hat{p}$. That is, if $\sigma \in \mathcal{S}_\mathcal{X}$, then $(\sigma\hat{p})_x \triangleq \hat{p}_{\sigma(x)}$, and $\mathcal{S}_\mathcal{X}\hat{p} = \{\sigma\hat{p}\}_{\sigma\in\mathcal{S}_\mathcal{X}}$. Each distribution in the orbit $\mathcal{S}_\mathcal{X}\hat{p}$ has the same fingerprint and distinct orbits have distinct fingerprints, so the set of distinct fingerprints $\mathcal{F}$ is in bijection with the set of distinct orbits under the action of $\mathcal{S}_\mathcal{X}$, so the probability of a specific fingerprint $\mathcal{F}$ is given by

$$\mathbb{P}_p(\mathcal{F}) = \mathbb{P}_p(\mathcal{S}_\mathcal{X}\hat{p}) = \sum_{\hat{p}'\in\mathcal{S}_\mathcal{X}\hat{p}} \mathbb{P}_p(\hat{p}') = \binom{n}{n\hat{p}} \sum_{\hat{p}'\in\mathcal{S}_\mathcal{X}\hat{p}} \prod_{x\in\mathcal{X}} p_x^{n\hat{p}'_x} \tag{76}$$

where the last equality follows from (10).

When the alphabet $\mathcal{X}$ is finite, then we can replace the summation over the orbit $\mathcal{S}_\mathcal{X}\hat{p}$ in (76) with a summation over the finite symmetric group $\mathcal{S}_\mathcal{X}$, taking care to avoid double-counting, since $|\mathcal{S}_\mathcal{X}\hat{p}| \le |\mathcal{S}_\mathcal{X}|$ with strict inequality if there exist distinct $x, x' \in \mathcal{X}$ such that $\hat{p}_x = \hat{p}_{x'}$ (see Appendix B for an example of this). The fingerprint notation lets us

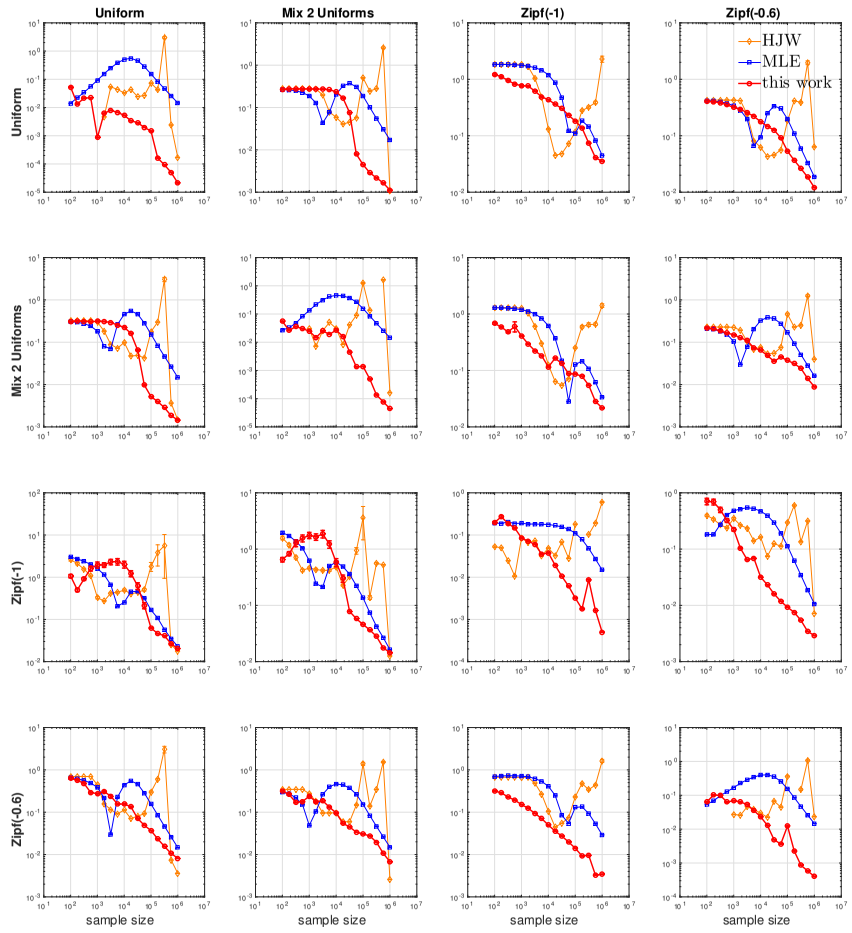


Figure 12: Root mean squared error for estimation of KL divergence between distributions indicated in row and column names (that is, $D(p_{\text{row}}||p_{\text{column}})$). In all cases $K = |\mathcal{X}| = 10^4$. "Uniform" is uniform on $\mathcal{X}$, "Mix 2 Uniforms" is a mixture of two uniform distributions, with half the probability mass on the first $K/5$ symbols, and the other half on the remaining symbols, and Zipf(α) $\sim 1/i^\alpha$ with $i \in \{1,\ldots,K\}$. MLE denotes the ML plugin ("naive") approach of computing $D(\hat{p}||\hat{q})$. HJW is (Han et al., 2016a,b). Each data point represents 100 random trials, with 2 standard error bars smaller than the plot marker for most points. Log base 2.

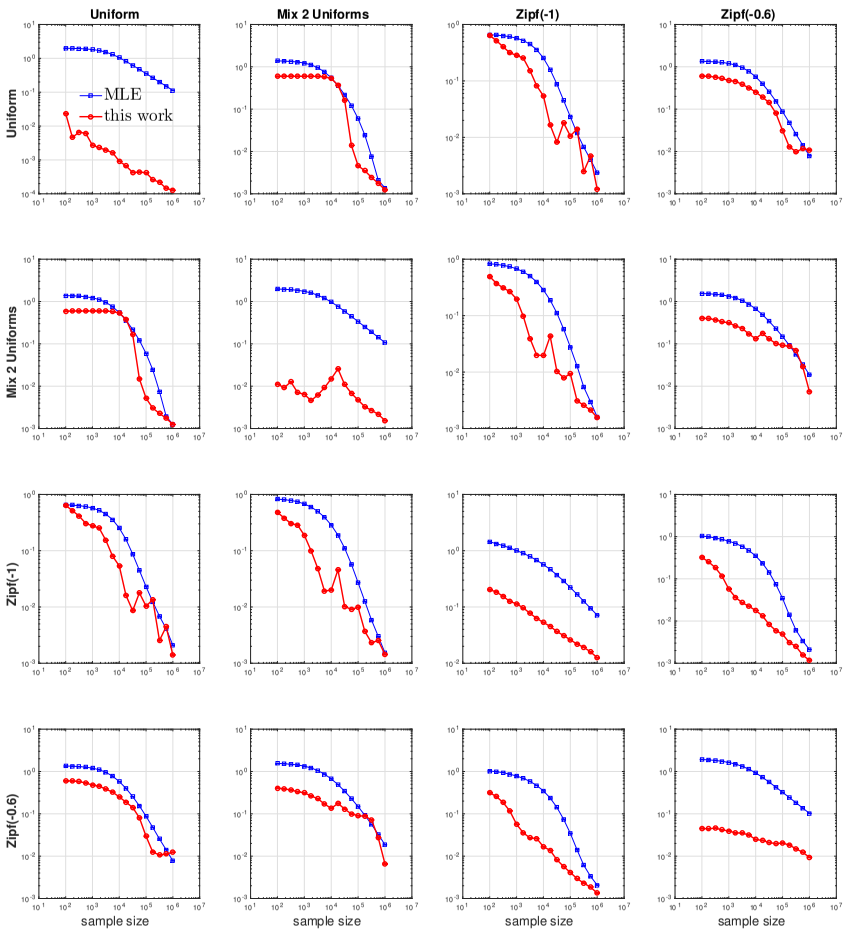


Figure 13: Root mean squared error for estimation of L_1 distance between distributions indicated in row and column names. Same parameters as in Figure 12. Since the L_1 distance is symmetric in its arguments, the performance matrix is symmetric.

conveniently express $|\mathcal{S}_{\mathcal{X}}\hat{p}|$:

$$|\mathcal{S}_{\mathcal{X}}\hat{p}| = \binom{|\mathcal{X}|}{\mathcal{F}} = |\mathcal{X}|! \prod_{i\geq 0} \frac{1}{\mathcal{F}_i!}. \tag{77}$$

The probability of a specific fingerprint $\mathcal{F}$ under distribution p with finite support $\mathcal{X}$ is given by

$$\begin{aligned}
\mathbb{P}_p(\mathcal{F}) = \mathbb{P}_p(\mathcal{S}_{\mathcal{X}}\hat{p}) &= \sum_{\hat{p}' \in \mathcal{S}_{\mathcal{X}}\hat{p}} \mathbb{P}_p(\hat{p}') && (78)\\
&\overset{(a)}{=} \frac{|\mathcal{S}_{\mathcal{X}}\hat{p}|}{|\mathcal{S}_{\mathcal{X}}|} \sum_{\sigma \in \mathcal{S}_{\mathcal{X}}} \mathbb{P}_p(\sigma\hat{p}) && (79)\\
&\overset{(b)}{=} \frac{1}{|\mathcal{X}|!} \binom{|\mathcal{X}|}{\mathcal{F}} \sum_{\sigma \in \mathcal{S}_{\mathcal{X}}} \mathbb{P}_p(\sigma\hat{p}) && (80)\\
&\overset{(c)}{=} \left(\prod_{i\geq 0} \frac{1}{\mathcal{F}_i!}\right) \binom{n}{n\hat{p}} \sum_{\sigma \in \mathcal{S}_{\mathcal{X}}} \prod_{x \in \mathcal{X}} p_x^{n\hat{p}_{\sigma(x)}} && (81)\\
&= \left(\prod_{i\geq 0} \frac{1}{\mathcal{F}_i!}\right) \binom{n}{n\hat{p}} \operatorname{perm}\Big(\underbrace{\big(p_x^{n\hat{p}_{x'}}\big)_{x,x' \in \mathcal{X}}}_{Q}\Big) && (82)\\
&\overset{(d)}{=} \frac{1}{|\mathcal{X} \setminus \operatorname{Supp}(\hat{p})|!} \left(\prod_{i\geq 1} \frac{1}{\mathcal{F}_i!}\right) \binom{n}{n\hat{p}} \operatorname{perm}(Q) && (83)
\end{aligned}$$

where in (a) the prefactor corrects for multiple-counting points in the orbit $\mathcal{S}_{\mathcal{X}}\hat{p}$ when we sum over $\sigma \in \mathcal{S}_{\mathcal{X}}$, (b) follows from (77), (c) follows from (10), and (d) follows since $\mathcal{F}_0 = |\mathcal{X} \setminus \operatorname{Supp}(\hat{p})|$.

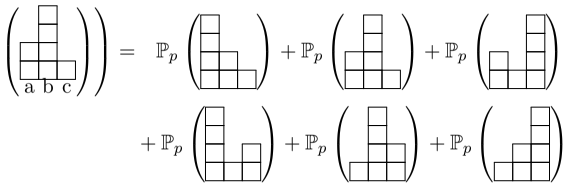
When the alphabet $\mathcal{X}$ is countably infinite, then we group terms in the sum (76) by the support set $\operatorname{Supp}(\hat{p}')$. Let $\hat{K} = |\operatorname{Supp}(\hat{p})|$ (13) denote the empirical support set size and let $\Pi_{\hat{K}}(\mathcal{X})$ denote the set of all subsets of $\mathcal{X}$ of size $\hat{K}$. Then

$$\begin{aligned}
\mathbb{P}_p(\mathcal{F}) &= \sum_{\hat{p}' \in \mathcal{S}_{\mathcal{X}}\hat{p}} \mathbb{P}_p(\hat{p}') && (84)\\
&= \sum_{\mathcal{X}' \in \Pi_{\hat{K}}(\mathcal{X})} \sum_{\hat{p}' \in S_{\mathcal{X}}\hat{p}:\ \operatorname{Supp}(\hat{p}') = \mathcal{X}'} \mathbb{P}_p(\hat{p}') && (85)\\
&\overset{(a)}{=} \sum_{\mathcal{X}' \in \Pi_{\hat{K}}(\mathcal{X})} \frac{|\mathcal{S}_{\mathcal{X}'}\hat{p}|}{|\mathcal{S}_{\mathcal{X}'}|} \sum_{\sigma \in \mathcal{S}_{\mathcal{X}'}} \mathbb{P}_p(\sigma\hat{p}) && (86)\\
&\overset{(b)}{=} \left(\prod_{i\geq 1} \frac{1}{\mathcal{F}_i!}\right) \binom{n}{n\hat{p}} \sum_{\mathcal{X}' \in \Pi_{\hat{K}}(\mathcal{X})} \operatorname{perm}\Big(\big(p_x^{n\hat{p}_{x'}}\big)_{x,x' \in \mathcal{X}'}\Big) && (87)
\end{aligned}$$

where in (a) the prefactor corrects for multiple-counting points in the orbit $\mathcal{S}_{\mathcal{X}'}\hat{p}$ when we sum over $\sigma \in \mathcal{S}_{\mathcal{X}'}$ and (b) follows from (10) and the fact that $\hat{p}'$ is supported on $\mathcal{X}'$ (so there is no $\mathcal{F}_0$ term, unlike the case when $\mathcal{X}$ is finite).

Appendix B. Illustration of computing the probability of a fingerprint

To illustrate the computation of the probability of the fingerprint, suppose that we have an alphabet of 3 symbols $\mathcal{X} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$ and consider the sequence $x_1^7 = (\mathrm{a,b,b,a,b,b,c})$. Then the empirical distribution is $\hat{p}(x_1^7) = (\hat{p}_\mathrm{a}, \hat{p}_\mathrm{b}, \hat{p}_\mathrm{c}) = \frac{1}{7}(2, 4, 1)$ and the fingerprint $\mathcal{F}(\hat{p})$ is $\mathcal{F}_i = 1$ if $i \in \{1, 2, 3\}$ and $\mathcal{F}_i = 0$ otherwise. The probability under $p = (p_\mathrm{a}, p_\mathrm{b}, p_\mathrm{c})$ of $\mathcal{F}(\hat{p})$ is indicated pictorially below:

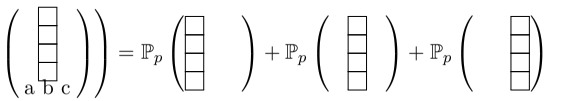


$$\mathbb{P}(\mathcal{F}(\hat{p})) = \mathbb{P}_p\left(\mathcal{S}_\mathcal{X}\left(\cdots\right)\right) = \mathbb{P}_p\left(\cdots\right) + \mathbb{P}_p\left(\cdots\right) + \mathbb{P}_p\left(\cdots\right) + \mathbb{P}_p\left(\cdots\right) + \mathbb{P}_p\left(\cdots\right) + \mathbb{P}_p\left(\cdots\right) \tag{88}$$

$$= \frac{1}{1!1!1!}\,\frac{7!}{2!\;4!\;1!}\;\mathrm{perm}\begin{pmatrix} p_\mathrm{a}^2 & p_\mathrm{a}^4 & p_\mathrm{a}^1 \\ p_\mathrm{b}^2 & p_\mathrm{b}^4 & p_\mathrm{b}^1 \\ p_\mathrm{c}^2 & p_\mathrm{c}^4 & p_\mathrm{c}^1 \end{pmatrix} \tag{89}$$

where each term on the right hand side of (88) is computed via (10) and in (89) the two prefactors are the multinomial coefficients in (83).

Suppose now that we happen to observe only the symbol 'b' in $x_1^4 = (\mathrm{b,b,b,b})$, but know that $\mathcal{X}$ consists of 3 symbols as in the previous example. Then $\hat{p}(x_1^4) = \frac{1}{4}(0, 1, 0)$ and the fingerprint is $\mathcal{F}_0 = 2$, $\mathcal{F}_1 = 1$, and $\mathcal{F}_i = 0$ for $i \geq 2$. Our pictorial computation of the probability under $p = (p_\mathrm{a}, p_\mathrm{b}, p_\mathrm{c})$ of $\mathcal{F}(\hat{p})$ is shown in (90):



$$\mathbb{P}_p(\mathcal{F}(\hat{p})) = \mathbb{P}_p\left(\mathcal{S}_\mathcal{X}\left(\cdots\right)\right) = \mathbb{P}_p\left(\cdots\right) + \mathbb{P}_p\left(\cdots\right) + \mathbb{P}_p\left(\cdots\right) \tag{90}$$

$$= \frac{1}{2!1!}\,\frac{4!}{0!4!0!}\;\mathrm{perm}\begin{pmatrix} 1 & p_\mathrm{a}^4 & 1 \\ 1 & p_\mathrm{b}^4 & 1 \\ 1 & p_\mathrm{c}^4 & 1 \end{pmatrix} = p_\mathrm{a}^4 + p_\mathrm{b}^4 + p_\mathrm{c}^4 \tag{91}$$

Note that there are now fewer than $|\mathcal{S}_\mathcal{X}| = |\mathcal{X}|!$ distinct permutations to sum over, reflected in the prefactor $1/\mathcal{F}_0!$ with $\mathcal{F}_0 = 2$ since there are two "unseen" symbols.

Appendix C. Proof of Theorem 12

Let Z be the number in Theorem 12. We first prove the lower bound.

Fix a (constant) parameter $\alpha \in (0, 1)$. Consider all *0-1* assignments to an $\alpha n \times \alpha n$ array such that the total number of 1's is exactly n. The number of such assignments is

$$\binom{\alpha^2 n^2}{n} \geq \left(\alpha^2 n\right)^n.$$

On the other hand, each equivalent class generates at most

$$((\alpha n)!)^2 \leq (\alpha n)^{2\alpha n}$$

such assignments. Therefore, the number of equivalent classes Z is at least

$$Z \geq \left(\alpha^2 n\right)^n / (\alpha n)^{2\alpha n} = n^{(1-2\alpha)n} \cdot \alpha^{2n(1-\alpha)}.$$

The above inequality holds for any $\alpha > 0$. In particular, for any constant α, we have $Z \geq n^{(1-2\alpha-o(1))n}$, which implies the claimed bound.

Now we prove the upper bound. Consider the following procedure for generating an assignment to a $n \times n$ array.

1. Choose an integer $1 \leq m \leq n$ arbitrarily;
2. Choose m entries from the n^2 entries arbitrarily;
3. For each of the m chosen entries from last step, fill in a positive integer such that the m entries sum up to n.

It is easy to verify that for every equivalent class, some assignment in the class can be generated by the above procedure. On the other hand, the number of different assignments that can be generated is at most

$$\sum_{m=1}^{n} \binom{n^2}{m} \cdot \binom{n}{m} \leq 2^n \cdot n \cdot \max_{1 \leq m \leq n} \binom{n^2}{m} \leq 2^n \cdot n \cdot (e \cdot n)^n = n^{(1+o(1))n}.$$

This proves the upper bound.

Appendix D. The EM algorithm used to numerically solve the PML for small alphabets in Section 4

D.1. 1-D case

We treat p as the unknown parameter to be inferred and assume that p is sorted. Recall that for a fixed alphabet size, the PML distribution is:

$$p^* = \operatorname*{argmax}_p \sum_{\sigma \in \mathcal{S}_\mathcal{X}} \prod_{x \in \mathcal{X}} p_x^{n\hat{p}_{\sigma(x)}} \tag{92}$$

We treat permutation σ as the hidden data drawn uniformly on $\mathcal{S}_\mathcal{X}$, and the observed data $\hat{p}$ drawn from the permuted distribution σp. Then if $p_{(t)}$ is our estimate for p after t iterations of the EM algorithm, the next value is:

$$p_{(t+1)} \sim \sum_{\sigma \in \mathcal{S}_\mathcal{X}} (\sigma\hat{p}) \prod_{x \in \mathcal{X}} p_{(t),x}^{n\hat{p}_{\sigma(x)}} \tag{93}$$

where $\sigma\hat{p}$ denotes the permuted empirical histogram and $\sim$ denotes equality up to normalization. For the starting point we choose $p_{(0)} = \hat{p}$.

D.2. 2-D case

Analogously to the 1D case above, let $p_{(t)}, q_{(t)}$ be our estimates for p, q after t iterations of the EM algorithm. Then the next values are:

$$p_{(t+1)} \sim \sum_{\sigma \in \mathcal{S}_{\mathcal{X}}} (\sigma \hat{p}) w_\sigma \tag{94}$$

$$q_{(t+1)} \sim \sum_{\sigma \in \mathcal{S}_{\mathcal{X}}} (\sigma \hat{q}) w_\sigma \tag{95}$$

where

$$w_\sigma \triangleq \prod_{x \in \mathcal{X}} p_{(t),x}^{n\hat{p}_{\sigma(x)}} q_{(t),x}^{m\hat{q}_{\sigma(x)}} \tag{96}$$

and $\sim$ denotes equality up to normalization. For the starting point we choose $p_{(0)} = \hat{p}$, $q_{(0)} = \hat{q}$.

Appendix E. Proof of Theorem 8 (D-dimensional PML distribution on a binary alphabet)

The D-dimensional PML distributions $\mathbf{p}^* \triangleq (p^{(d)*})_{d=1}^D$ are (see Appendix H for a derivation):

$$\mathbf{p}^* \triangleq \operatorname*{argmax}_{\mathbf{p} \in \mathcal{P}} \operatorname{perm}\left(\left(\prod_{d=1}^D (p_x^{(d)})^{n_d \hat{p}_{x'}^{(d)}} \right)_{x,x' \in \{1,2\}} \right) \tag{97}$$

$$= \operatorname*{argmax}_{\mathbf{p} \in \mathcal{P}} \left(\prod_{d=1}^D (p_1^{(d)})^{n_d \hat{p}_1^{(d)}} (1-p_1^{(d)})^{n_d(1-\hat{p}_1^{(d)})} + \prod_{d=1}^D (p_1^{(d)})^{n_d(1-\hat{p}_1^{(d)})} (1-p_1^{(d)})^{n_d \hat{p}_1^{(d)}} \right) \tag{98}$$

where $\mathcal{P}$ is a collection of D-tuples of distributions on the binary alphabet $\mathcal{X} = \{1,2\}$. To lighten notation, let

$$r_d \triangleq p_1^{(d)} \tag{99}$$

and

$$\Delta_d \triangleq n_d \left(\hat{p}_1^{(d)} - \frac{1}{2} \right) \tag{100}$$

Then $\mathbf{p}^* = ((r_d^*, 1-r_d^*))_{d=1}^D$, where

$$(r_d^*)_{d=1}^D = \operatorname*{argmax}_{(r_d)_{d=1}^D \in [0,1]^D} \underbrace{\left(\prod_{d=1}^D r_d^{n_d/2+\Delta_d} (1-r_d)^{n_d/2-\Delta_d} + \prod_{d=1}^D r_d^{n_d/2-\Delta_d} (1-r_d)^{n_d/2+\Delta_d} \right)}_{-U(r_1,\ldots,r_d)} \tag{101}$$

$$= \operatorname*{argmin}_{(r_d)_{d=1}^D \in [0,1]^D} U(r_1, \ldots, r_d) \tag{102}$$

If $r_d^* = 1/2$ for all d, then $U(r_1, \ldots, r_d)$ must have positive definite Hessian H_U at $(r_d)_{d=1}^D = (1/2)_{d=1}^D$. Let's compute the Hessian at $r_d = 1/2$ for all d, letting $[D] \triangleq \{1, \ldots, D\}$:

$$H_U^* \triangleq H_U|_{(r_d)_{d=1}^D = (1/2)_{d=1}^D} = \left(\frac{\partial^2 U}{\partial r_d \partial r_{d'}} \right)_{d,d' \in [D]} \Bigg|_{(r_d)_{d=1}^D = (1/2)_{d=1}^D} \tag{103}$$

$$= c \left((\Delta_d \Delta_{d'})_{d,d' \in [D]} - \operatorname{diag}\left(\left(\frac{n_d}{4} \right)_{d \in [D]} \right) \right) \tag{104}$$

where $c = 2^{5-\sum_{d=1}^D n_d}$ and in the second term in (104) $\operatorname{diag}(\cdot)$ denotes the diagonal matrix with entry n_d in the d-th row and d-th column.

The matrix H_U^* is positive definite when $\Delta_d = 0$ for all d with eigenvalues $(-cn_d/4)_{d=1}^D$. Since the eigenvalues of H_U^* vary continuously in $(\Delta_d)_{d=1}^D$, we conclude that H_U^* is positive definite when $(\Delta_d)_{d=1}^D$ is in the neighborhood of $(0)_{d=1}^D$ determined by:

$$\det(H) = c' \left(-1 + \sum_{d=1}^D \frac{4\Delta_d^2}{n_d} \right) > 0 \tag{105}$$

$$\Updownarrow$$

$$\sum_{d=1}^D 4n_d \left(\hat{p}_1^{(d)} - \frac{1}{2} \right)^2 > 1 \tag{106}$$

where $c' = c\, 2^{2D} \prod_{d=1}^D n_d$. Thus H_U^* is not positive definite at $(r_d)_{d=1}^D = (1/2)_{d=1}^D$ when the empirical distribution components $(\hat{p}_1^{(d)})_{d=1}^D$ lie outside the ellipse determined by (106), so the D PML distributions $(p^{(d)*})_{d=1}^D$ are not uniform in this case.

Appendix F. Proof of permanent lower bound identity (35)

Given distribution p supported on set $\mathcal{X}$, denote by $\mathcal{X}_u(p)$ a level set of p:

$$\mathcal{X}_u(p) \triangleq \{x \in \mathcal{X} : p_x = u\} \tag{107}$$

Denote by $\mathcal{U}(p)$ the set of unique entries of p:

$$\mathcal{U}(p) \triangleq \{p_x : x \in \mathcal{X}\} \tag{108}$$

and denote by $\mathcal{A}(p)$ the partition of $\mathcal{X}$ into level sets of p (equivalent to the definition (29)):

$$\mathcal{A}(p) \triangleq \{\mathcal{X}_u(p) : u \in \mathcal{U}(p)\} \tag{109}$$

It is convenient to express the group $\mathcal{S}_{\mathcal{X},p}$ (32) as isomorphic to:

$$\mathcal{S}_{\mathcal{X},p} \cong \bigtimes_{\alpha \in \mathcal{A}(p)} \mathcal{S}_\alpha = \bigtimes_{u \in \mathcal{U}(p)} \mathcal{S}_{\mathcal{X}_u} \tag{110}$$

We write

$$
\begin{aligned}
V(p) = \log(\mathrm{perm}(Q)) \geq \bar{V}(p) & \qquad (111)\\
= \log\left(\sum_{\sigma\in\mathcal{S}_{\mathcal{X},p}}\prod_{x\in\mathcal{X}} p_x^{n\hat{p}_{\sigma(x)}}\right) & \qquad (112)\\
\overset{(a)}{=} \log\left(\sum_{\sigma\in\mathcal{S}_{\mathcal{X},p}}\prod_{x\in\mathcal{X}} p_{\sigma(x)}^{n\hat{p}_x}\right) & \qquad (113)\\
\overset{(b)}{=} \log\left(\sum_{\sigma\in\times_{u\in\mathcal{U}(p)}\mathcal{S}_{\mathcal{X}_u}}\prod_{x\in\mathcal{X}} p_{\sigma(x)}^{n\hat{p}_x}\right) & \qquad (114)\\
\overset{(c)}{=} \log\left(\prod_{u\in\mathcal{U}(p)}\sum_{\sigma\in\mathcal{S}_{\mathcal{X}_u}}\prod_{x\in\mathcal{X}_u} u^{n\hat{p}_x}\right) & \qquad (115)\\
\overset{(d)}{=} \log\left(\prod_{u\in\mathcal{U}(p)}|\mathcal{S}_{\mathcal{X}_u}|\; u^{n\sum_{x\in\mathcal{X}_u}\hat{p}_x}\right) & \qquad (116)\\
= \sum_{u\in\mathcal{U}(p)}\left(\log(|\mathcal{S}_{\mathcal{X}_u}|) + n\Big(\sum_{x\in\mathcal{X}_u}\hat{p}_x\Big)\log(u)\right) & \qquad (117)\\
= \sum_{u\in\mathcal{U}(p)}\log(|\mathcal{S}_{\mathcal{X}_u}|!) + n\sum_{u\in\mathcal{U}(p)}\sum_{x\in\mathcal{X}_u}\hat{p}_x\log(u) & \qquad (118)\\
\overset{(e)}{=} \sum_{u\in\mathcal{U}(p)}\log(|\mathcal{X}_u|!) + n\sum_{x\in\mathcal{X}}\hat{p}_x\log(p_x) & \qquad (119)\\
= \sum_{u\in\mathcal{U}(p)}\log(|\mathcal{X}_u|!) + n\sum_{x\in\mathcal{X}}\hat{p}_x\left(\log\left(\frac{p_x}{\hat{p}_x}\right) + \log(\hat{p}_x)\right) & \qquad (120)\\
\overset{(f)}{=} \sum_{\alpha\in\mathcal{A}(p)}\log(|\alpha|!) - n(D(\hat{p}\|p) + H(\hat{p})) & \qquad (121)
\end{aligned}
$$

where in (a) we changed order of summation over the group $\mathcal{S}_{\mathcal{X}}$, in (b) we summed over the isomorphic group (110), in (c) we used the product structure (110) of $\mathcal{S}_{\mathcal{X},p}$ to rewrite the sum as a product of sums and used the fact that $p_x = u$ for all $x \in \mathcal{X}_u$ (107). In (d) we used the fact that the summands are independent of σ and moved the sum into the exponent. In (e) we used $|\mathcal{S}_\alpha| = |\alpha|!$ and used the fact that $u = p_x$ for all $x \in \mathcal{X}_u$. In (f) we used $\alpha \in \mathcal{A}(p) \Leftrightarrow \exists u \in \mathcal{U}(p) : \mathcal{X}_u = \alpha$.

Appendix G. Proof of Theorem 9 (properties of the approximate PML distribution's level set partition $\bar{\mathcal{A}}^*$)

G.1. Proof of iso-clumping property (41)

Suppose $\bar{\mathcal{A}}^*$ does not satisfy property (41). Then there exist $x, y \in \mathcal{X}$ such that $\hat{p}_x = \hat{p}_y$, but $\alpha(x) \neq \alpha(y)$. We show that the partition obtained by either "reassigning" y to the same level set as x or reassigning x to the same level set as y achieves a higher value of the log permanent lower bound $\bar{V}(\cdot)$ (38), contradicting the optimality of $\bar{\mathcal{A}}^*$ (40).

Let $m \triangleq n\hat{p}_x = n\hat{p}_y$, let $N_{\alpha(x)} \triangleq \sum_{x\in\alpha(x)} n\hat{p}_x$. Let V_δ denote for $\delta \in [-1, 1]$:

$$
\begin{aligned}
V_\delta \triangleq \log((|\alpha(x)| + \delta)!) + \big(N_{\alpha(x)} + m\delta\big)\log\left(\frac{N_{\alpha(x)} + m\delta}{n(|\alpha(x)| + \delta)}\right)\\
+ \log((|\alpha(y)| - \delta)!) + \big(N_{\alpha(y)} - m\delta\big)\log\left(\frac{N_{\alpha(y)} - m\delta}{n(|\alpha(y)| - \delta)}\right) \qquad (122)
\end{aligned}
$$

Then $V_0 = \bar{V}(\alpha(x)) + \bar{V}(\alpha(y))$ (39) corresponds to two terms in $\bar{V}(\bar{\mathcal{A}}^*)$ (38), and V_1 (resp. V_{-1}) corresponds to two terms in the value of the partition of $\mathcal{X}$ obtained by reassigning symbol y to the same level set as x (resp. reassigning symbol x to the same level set as y). We can check that V_δ is strictly convex in $\delta \in [-1, 1]$. A strictly convex function attains its maximum on the boundaries of a convex set, so either $V_{-1} > V_0$ or $V_1 > V_0$, contradicting the optimality of partition $\bar{\mathcal{A}}^*$, thus establishing the iso-clumping property.

G.2. Proof of convexity property (42)

Suppose $\bar{\mathcal{A}}^*$ does not satisfy property (42). Then there exist $x_1, x_2, x_3 \in \mathcal{X}$ such that $\hat{p}_{x_1} < \hat{p}_{x_2} < \hat{p}_{x_3}$, but $\alpha(x_1) = \alpha(x_3) \neq \alpha(x_2)$. For all $x \in \mathcal{X}$, let $N_{\alpha(x)} \triangleq \sum_{x\in\alpha(x)} n\hat{p}_x$. If $\frac{N_{\alpha(x_1)}}{|\alpha(x_1)|} \geq \frac{N_{\alpha(x_2)}}{|\alpha(x_2)|}$, then let $x = x_2$ and $y = x_1$; otherwise let $x = x_3$, $y = x_2$. Let $\Delta \triangleq n(\hat{p}_x - \hat{p}_y)$. Then $\Delta > 0$ and

$$
\frac{N_{\alpha(y)}}{|\alpha(y)|} \geq \frac{N_{\alpha(x)}}{|\alpha(x)|}. \qquad (123)
$$

Let $\mathcal{A}'$ denote the partition of $\mathcal{X}$ obtained by swapping the set memberships of x and y in $\bar{\mathcal{A}}^*$. We show that $\bar{V}(\mathcal{A}') > \bar{V}(\bar{\mathcal{A}}^*)$, contradicting the optimality of $\bar{\mathcal{A}}^*$ ($\mathcal{A}'$ might not satisfy the iso-clumping property (41)).

Since $\mathcal{A}'$ is obtained from $\bar{\mathcal{A}}^*$ by swapping the level set memberships of x and y, the elements of $\mathcal{A}'$ and $\bar{\mathcal{A}}^*$ have the same sizes, so $\sum_{\alpha\in\mathcal{A}'}\log(|\alpha|!) = \sum_{\alpha\in\bar{\mathcal{A}}^*}\log(|\alpha|!)$, so using (38) and (39) we write:

$$
\begin{aligned}
\bar{V}(\mathcal{A}') - \bar{V}(\bar{\mathcal{A}}^*) =& (N_{\alpha(x)} - \Delta)\log\left(\frac{N_{\alpha(x)} - \Delta}{n|\alpha(x)|}\right) + (N_{\alpha(y)} + \Delta)\log\left(\frac{N_{\alpha(y)} + \Delta}{n|\alpha(x)|}\right)\\
&- N_{\alpha(x)}\log\left(\frac{N_{\alpha_x}}{n|\alpha(x)|}\right) - N_{\alpha(y)}\log\left(\frac{N_{\alpha(y)}}{n|\alpha(y)|}\right) \qquad (124)\\
=& \Delta\log\left(\frac{N_{\alpha(y)}}{|\alpha(y)|}\frac{|\alpha(x)|}{N_{\alpha(x)}}\right) + f(N_{\alpha(x)}, \Delta) + f(N_{\alpha(y)}, -\Delta) \qquad (125)\\
\overset{(a)}{>}& 0 \qquad (126)
\end{aligned}
$$

where $f(N, \Delta) \triangleq \Delta + (N - \Delta)\log\left(\frac{N-\Delta}{N}\right)$ with $f(N, N) \triangleq \lim_{\Delta\to N} f(N, \Delta) = N$. (a) follows because we can show that $f(N, \Delta) \geq 0$ for $\Delta \leq N$ with strict inequality iff $\Delta \neq 0$, so the last two terms in (125) are positive since $\Delta > 0$. The first term in (125) is non-negative since $\Delta > 0$ and (123). Thus $\bar{V}(\mathcal{A}') > \bar{V}(\bar{\mathcal{A}}^*)$, contradicting the optimality of partition $\bar{\mathcal{A}}^*$, thus establishing the convexity property.

Appendix H. Our approximate PML algorithm: the case of multiple distributions

We generalize the discussion of Section 3.2 to D-dimensional fingerprints. In this case we are not able to optimize a D-dimensional analogue of our objective function $\bar{V}$ (31) and will instead settle for an approximation of our already-approximate PML distribution. We use a greedy heuristic to approximate the optimal partition $\bar{\mathcal{A}}^*$ (40) by iteratively, greedily merging partition elements (enlarging the second term in a D-dimensional analogue of (35) until the first term becomes too large) (see Section H.4). Even this doubly-approximate algorithm turns out to be too slow to run on examples of practical interest, so we introduce a further heuristic to speed things up (only trying to merge partition elements that are close in Euclidean distance), which comes at a cost of adding some tunable knobs to the algorithm.

H.1. Some notation

Suppose we have D distributions $(p^{(d)})_{d=1}^D$ on the same alphabet $\mathcal{X}$ and for each d draw n_d samples i.i.d. from $p^{(d)}$ with empirical distribution $\hat{p}^{(d)}$. Let bold symbols denote D-component collections:

$$\mathbf{p} \triangleq (p^{(d)})_{d=1}^D \quad \mathbf{n} \triangleq (n_d)_{d=1}^D \quad \hat{\mathbf{p}} \triangleq (\hat{p}^{(d)})_{d=1}^D \quad \mathbf{n}\hat{\mathbf{p}} \triangleq (n_d p^{(d)})_{d=1}^D \tag{127}$$

We assume that the D distributions $\mathbf{p}$ satisfy:

$$\sum_{d=1}^D p_x^{(d)} > 0 \; \forall x \in \mathcal{X} \tag{128}$$

This condition ensures that there are no symbols x such that $p_x^{(d)} = 0$ for all d, which simplifies the expressions below. Denote by $\hat{\mathcal{X}}$ the empirical support set:

$$\hat{\mathcal{X}} \triangleq \bigcup_{d=1}^D \operatorname{Supp}(\hat{p}^{(d)}) = \{x \in \mathcal{X} : \sum_{d=1}^D \hat{p}_x^{(d)} > 0\} \tag{129}$$

Finally let $K \triangleq |\mathcal{X}|$ and $\hat{K} \triangleq |\hat{\mathcal{X}}|$.

The probability to draw empirical distributions $\hat{\mathbf{p}}$ under distributions $\mathbf{p}$ is (using the independence of the D empirical distributions and (10)):

$$\mathbb{P}_{\mathbf{p}}(\hat{\mathbf{p}}) = \left(\prod_{d=1}^D \binom{n_d}{n_d \hat{p}^{(d)}}\right) \prod_{x \in \mathcal{X}} \prod_{d=1}^D (p_x^{(d)})^{n_d \hat{p}_x^{(d)}} \tag{130}$$

where the first product is over multinomial coefficients in the notation of (10).

Hereon denote by $\mathcal{F} = \mathcal{F}(\hat{p}^{(1)}, \ldots, \hat{p}^{(D)})$ the D-dimensional fingerprint (Raghunathan et al., 2017) indexed by vector $\mathbf{i} = (i_d)_{d=1}^D \in \mathbb{N}^D$ (which we write as satisfying $\mathbf{i} \geq \mathbf{0} \Leftrightarrow i_d \geq 0 \; \forall d$, where $\mathbf{0} \triangleq (0)_{d=1}^D$):

$$\mathcal{F} = \mathcal{F}(\hat{\mathbf{p}}) = (\mathcal{F}_{\mathbf{i}})_{\mathbf{i} \geq \mathbf{0}} \triangleq \left(|\{x \in \mathcal{X} : (n_d \hat{p}_x^{(d)})_{d=1}^D = \mathbf{i}\}|\right)_{\mathbf{i} \geq \mathbf{0}} \tag{131}$$

Note that $\mathcal{F}_{\mathbf{0}} = |\mathcal{X} \setminus \hat{\mathcal{X}}| = K - \hat{K}$ counts the number of symbols "unseen" by any of the D samples, and is thus unknown if the support set size K is unknown. Then the probability to draw the D-dimensional fingerprint $\mathcal{F}$ under $\mathbf{p}$ is (obtained analogously to the $D = 1$ case in Appendix A):

$$\mathbb{P}_{\mathbf{p}}(\mathcal{F}) = \left(\prod_{\mathbf{i} \geq \mathbf{0}} \frac{1}{\mathcal{F}_{\mathbf{i}}!}\right) \left(\prod_{d=1}^D \binom{n_d}{n_d \hat{p}^{(d)}}\right) \sum_{\sigma \in \mathcal{S}_{\mathcal{X}}} \prod_{x \in \mathcal{X}} \prod_{d=1}^D (p_x^{(d)})^{n_d \hat{p}_{\sigma(x)}^{(d)}} \tag{132}$$

$$= \frac{1}{(K - \hat{K})!} \left(\prod_{\mathbf{i} \geq \mathbf{0}, \mathbf{i} \neq \mathbf{0}} \frac{1}{\mathcal{F}_{\mathbf{i}}!}\right) \left(\prod_{d=1}^D \binom{n_d}{n_d \hat{p}^{(d)}}\right) \operatorname{perm}\Bigg(\underbrace{\left(\prod_{d=1}^D (p_x^{(d)})^{n_d \hat{p}_{x'}^{(d)}}\right)_{x, x' \in \mathcal{X}}}_{Q}\Bigg) \tag{133}$$

where in (133) the first product is over all D-dimensional indices $\mathbf{i}$ such that $\min_d i_d \geq 0$ and $\max_d i_d > 0$.

H.2. The PML distributions in D dimensions

For a given collection $\mathcal{P}$ of D-tuples of distributions, the D-dimensional PML distributions $\mathbf{p}^*$ are:

$$\mathbf{p}^* \triangleq (p^{(d)*})_{d=1}^D \triangleq \operatorname*{argmax}_{\mathbf{p} \in \mathcal{P}} \mathbb{P}_{\mathbf{p}}(\mathcal{F}(\hat{\mathbf{p}})) \tag{134}$$

$$= \operatorname*{argmax}_{\mathbf{p} \in \mathcal{P}} \frac{\operatorname{perm}(Q)}{(K - \hat{K})!} \tag{135}$$

where in (135) we discarded all $\mathbf{p}$-independent factors of $\mathbb{P}_{\mathbf{p}}(\mathcal{F})$ (133). $\mathcal{F}_{\mathbf{0}} = K - \hat{K} = |\mathcal{X} \setminus \hat{\mathcal{X}}|$ depends on p through its support set size. Note that $\mathbb{P}_{\mathbf{p}}(\mathcal{F})$ is invariant under relabeling of the components of $\mathbf{p}$, so we can choose $\mathbf{p}^*$ to be non-increasing in the same ordering as we choose for the support set $\mathcal{X}$. Note that the set $\mathcal{P}$ is not necessarily the same as the set of all discrete distributions.

Note that finding the D-dimensional PML distributions (135) is not equivalent to finding D 1-dimensional PML distributions since $\mathbb{P}_{\mathbf{p}}(\mathcal{F})$ (133) does not factor into D pieces.

If the collection of D-tuples of distributions $\mathcal{P}$ includes distributions with different support set sizes, then we can estimate the support set size by breaking up the optimization in (135) into two steps:

$$K^* \triangleq \operatorname*{argmax}_K \left(\frac{1}{(K - \hat{K})!} \max_{\mathbf{p} \in \mathcal{P}_K} \operatorname{perm}(Q)\right) \tag{136}$$

whenever the max over K exists, where $\mathcal{P}_K \triangleq \{\mathbf{p} \in \mathcal{P} : |\bigcup_{d=1}^D \operatorname{Supp}(p^{(d)})| = K\}$.

The case $D \geq 2$ is more complicated than the case $D = 1$ because the D support sets $(\operatorname{Supp}(p^{(d)}))_{d=1}^D$ can overlap partially. If $\mathcal{X}$ is the joint support set – that is, $\mathcal{X} = \bigcup_{d=1}^D \operatorname{Supp}(p^{(d)})$ – then we can associate with each symbol $x \in \mathcal{X}$ an indicator function $\phi_d(x) = \mathbb{1}(x \in \operatorname{Supp}(p^{(d)}))$, so there are $2^D - 1$ possible values $(\phi_d(x))_{d=1}^D$ for each $x \in \mathcal{X}$ [9].

9. 1 less than 2^D since $\phi_d(x) = 0 \; \forall d$ implies $x \notin \mathcal{X}$.

We could then be more ambitious and estimate not just the joint support set size $K = |\mathcal{X}|$, but also the optimal value $(\phi_d(x))_{d=1}^D$ for all symbols. Our proposed approximate PML approximation scheme for $D \geq 2$ is less ambitious and only supports estimation of $|\mathcal{X}|$, but not of the finer structure of the joint support set $\mathcal{X}$ in terms of D partially overlapping support sets $(\mathrm{Supp}(p^{(d)}))_{d=1}^D$.

H.3. Approximate PML in D dimensions: the case of known support set size

In this Section we assume the support set size $K = |\mathcal{X}|$ is known. Analogously to the notation in the $D = 1$ case of Section 4.3 and the proof of the permanent lower bound for $D = 1$ in Appendix F, define the level sets $\mathcal{X}_{\mathbf{u}}(\mathbf{p})$ of $\mathbf{p}$ indexed by D-dimensional real vector $\mathbf{u} = (u_d)_{d=1}^D$:

$$\mathcal{X}_{\mathbf{u}}(\mathbf{p}) \triangleq \{x \in \mathcal{X} : (p_x^{(d)})_{d=1}^D = \mathbf{u}\} \tag{137}$$

the unique D-tuple entries of $\mathbf{p}$ by $\mathcal{U}(\mathbf{p})$:

$$\mathcal{U}(\mathbf{p}) \triangleq \{(p_x^{(d)})_{d=1}^D : x \in \mathcal{X}\} \tag{138}$$

and the partition $\mathcal{A}(\mathbf{p})$ of $\mathcal{X}$ into level sets of $\mathbf{p}$:

$$\mathcal{A}(\mathbf{p}) \triangleq \{\mathcal{X}_{\mathbf{u}}(\mathbf{p}) : \mathbf{u} \in \mathcal{U}(\mathbf{p})\} \tag{139}$$

We assume $K = |\mathcal{X}|$ is known, so $\mathbf{p}^*$ (135) is a maximizer of

$$V(\mathbf{p}) \triangleq \log(\mathrm{perm}(Q)) \tag{140}$$

As for the $D = 1$ case, we do not know how to computationally efficiently maximize $V(\mathbf{p})$ over $\mathbf{p}$, so we settle for maximizing a lower bound $\bar{V}(\mathbf{p})$, analogous to (31) and motivated by our empirical observations for the case $D = 2$ in Section 4.2:

$$V(\mathbf{p}) \geq \bar{V}(\mathbf{p}) \triangleq \log\left(\sum_{\sigma \in \mathcal{S}_{\mathcal{X},\mathbf{p}}} \prod_{x \in \mathcal{X}} \prod_{d=1}^D (p_x^{(d)})^{n_d \hat{p}_{\sigma(x)}^{(d)}}\right) \tag{141}$$

$$= -\sum_{d=1}^D n_d \left(D(\hat{p}^{(d)} || p^{(d)}) + H(\hat{p}^{(d)})\right) + \sum_{\alpha \in \mathcal{A}(\mathbf{p})} \log(|\alpha|!) \tag{142}$$

where $\mathcal{S}_{\mathcal{X},\mathbf{p}}$ is analogous to (32) – a subgroup of $\mathcal{S}_{\mathcal{X}}$ consisting of all permutations that exchange only those alphabet symbols that are in the same level set of $\mathbf{p}$:

$$\mathcal{S}_{\mathcal{X},\mathbf{p}} \triangleq \{\sigma \in \mathcal{S}_{\mathcal{X}} : \sigma\mathbf{p} = (\sigma p^{(d)})_{d=1}^D = \mathbf{p}\} \tag{143}$$

(142) follows from (141) by an analogous argument to the one given in in Appendix F for the case $D = 1$.

As in the $D = 1$ case, the intuition is that we approximate the permanent by summing over only those permutations that contribute "a lot" to the value, and then maximize this lower bound $\bar{V}(\mathbf{p})$ over $\mathbf{p}$. Our approximate PML distributions are the D-tuple:

$$\bar{\mathbf{p}}^* = (\bar{p}^{(d)*})_{d=1}^D \triangleq \operatorname*{argmax}_{\mathbf{p} \in \mathcal{P}} \bar{V}(\mathbf{p}) \tag{144}$$

As in the 1-D case, the first term in (142) encourages $\bar{\mathbf{p}}^*$ to clump many symbols together, while the second term encourages $\bar{\mathbf{p}}^*$ to be similar to $\hat{\mathbf{p}}$, dominating as $n \to \infty$.

Note that optimizing $\bar{V}(\mathbf{p})$ over $\mathbf{p}$ is not equivalent to solving D independent optimization problems because the D distributions "interact" through the second summation in expression (142) (the summation over $\mathcal{A}(\mathbf{p})$).

As in the 1-D case, we can show that for all d the approximate PML distribution $\bar{p}^{(d)*}$ (144) satisfies the averaging property (37). Therefore as in the 1-D case, $\bar{\mathbf{p}}^*$ is determined by its partition of $\mathcal{X}$ into level sets. Thus we seek to maximize the quantity

$$\bar{V}(\mathcal{A}) \triangleq \sum_{\alpha \in \mathcal{A}} \bar{V}(\alpha) \tag{145}$$

where for $\alpha \subset \mathcal{X}$

$$\bar{V}(\alpha) \triangleq \log(|\alpha|!) + |\alpha| \sum_{d=1}^D n_d \hat{p}_\alpha^{(d)} \log(\hat{p}_\alpha^{(d)}) \tag{146}$$

The optimal partition is

$$\bar{\mathcal{A}}^* \triangleq \operatorname*{argmax}_{\mathcal{A}\text{: partition of } \mathcal{X}} \bar{V}(\mathcal{A}) \tag{147}$$

Finally $\bar{\mathbf{p}}^*$ is obtained by averaging the empirical distributions $\hat{\mathbf{p}}^*$ within each partition element $\alpha \in \bar{\mathcal{A}}^*$.

As in the 1-dimensional case, the approximate PML partition $\bar{\mathcal{A}}^*$ has the iso-clumping property (generalizing (41) to D dimensions):

$$(\hat{p}_x^{(d)})_{d=1}^D = (\hat{p}_{x'}^{(d)})_{d=1}^D \Rightarrow \alpha(x) = \alpha(x') \tag{148}$$

for all $x, x' \in \mathcal{X}$, where $\alpha(x), \alpha(x') \in \bar{\mathcal{A}}^*$ are the partition elements containing x, x', respectively. This can be shown by adapting the proof for the 1-D case in Appendix G.1.

We do not know if $\bar{\mathcal{A}}^*$ satisfies a convexity property analogous to the 1-D case (42).

H.4. A greedy heuristic to approximately maximize the permanent for $D \geq 2$ distributions

Unlike the $D = 1$ case, we do not solve the optimization problem (144) exactly for $D \geq 2$. The difficulty is that there is no natural ordering on $\mathbb{N}^D$ for $D \geq 2$, so we can no longer define a structured collection of subsets of $\mathcal{X}$ from which to build the approximate PML partition $\bar{\mathcal{A}}^*$ (147) $\mathcal{X}$, so we do not offer a dynamic programming algorithm like the one described in Section 4.3. Our search for an optimal partition of $\mathcal{X}$ is similar in flavor to a clustering problem, so the added difficulty in $D \geq 2$ dimensions makes some sense. For example, k-means clustering in 1 dimension is exactly solvable by a dynamic programming algorithm (Wang and Song, 2011) similar to the one we present, but no such algorithm is available for dimension greater than 1 (indeed k-means is NP-hard for dimension greater than 1).

We settle for a greedy heuristic to approximately maximize $\bar{V}(\mathcal{A})$, and call the resulting approximate optimizer $\bar{\bar{\mathcal{A}}}^*$:

$$\bar{\bar{\mathcal{A}}}^* \approx \operatorname*{argmax}_{\mathcal{A}\text{: partition of } \mathcal{X}} \bar{V}(\mathcal{A}) \tag{149}$$

the double bar signaling the two levels of approximation (an approximation to the approximate PML partition $\bar{\mathcal{A}}^*$ (144)). The doubly-approximate PML distribution $\bar{\bar{\mathbf{p}}}^*$ is obtained by averaging the empirical distributions $\hat{\mathbf{p}}^*$ within each partition element $\alpha \in \bar{\bar{\mathcal{A}}}^*$.

The greedy heuristic used to compute $\bar{\bar{\mathcal{A}}}^*$ is given in algorithm (1). In words, we start with initial partition $\mathcal{A}(\hat{\mathbf{p}})$ of $\mathcal{X}$ into the level sets of the empirical distributions $\hat{\mathbf{p}}$ (we clump equally empirically frequent symbols into the same level sets since we know that $\bar{\mathcal{A}}^*$ satisfies the iso-clumping property; see Section H.3) and then iteratively merge partition elements until it is no longer profitable to do so in terms of the objective function $\bar{V}$ increasing. At each step we merge two partition elements that most boost the value of the objective function.

Algorithm 1 Approximation to level set partition of the approximate D-dimensional PML distribution

1: **function** $\bar{\bar{\mathcal{A}}}^*(\hat{\mathbf{p}}, \mathbf{n}, |\mathcal{X}|)$ ▷ Input: D empirical distributions, D sample sizes, and assumed support set size $|\mathcal{X}|$
2: $\quad \mathcal{A} \leftarrow \mathcal{A}(\hat{\mathbf{p}})$ ▷ Initial partition into level sets of empirical distribution $\hat{\mathbf{p}}$
3: $\quad$ **while** $\max_{\alpha_1,\alpha_2 \in \mathcal{A}:\alpha_1 \neq \alpha_2} \left(\bar{V}(\alpha_1 \cup \alpha_2) - (\bar{V}(\alpha_1) + \bar{V}(\alpha_2))\right) > 0$ **do**
4: $\quad\quad (\alpha_1^*, \alpha_2^*) \leftarrow \operatorname{argmax}_{\alpha_1,\alpha_2 \in \mathcal{A}:\alpha_1 \neq \alpha_2} \left(\bar{V}(\alpha_1 \cup \alpha_2) - (\bar{V}(\alpha_1) + \bar{V}(\alpha_2))\right)$
5: $\quad\quad \mathcal{A} \leftarrow (\mathcal{A} \cup \{\alpha_1^* \cup \alpha_2^*\}) \setminus \{\alpha_1^*, \alpha_2^*\}$ ▷ Merge two level sets into one
6: $\quad$ **end while**
7: $\quad$ **return** $\mathcal{A}$ ▷ Output: $\bar{\bar{\mathcal{A}}}^*(\hat{\mathbf{p}})$
8: **end function**

The inputs $\hat{\mathbf{p}}$ and $\mathbf{n}$ in algorithm (1) are used when evaluating $\bar{V}(\alpha)$ (146) for $\alpha \subset \mathcal{X}$. The input of the support set size $|\mathcal{X}|$ is used in line (2) to initially assign the $|\mathcal{X}| - |\mathcal{X}(\hat{\mathbf{p}})|$ unobserved symbols to a partition element. The loop in algorithm (1) always terminates since we only merge distinct partition elements of the finite support set $\mathcal{X}$. We can implement algorithm (1) efficiently by observing that for $\alpha_1, \alpha_2 \subset \mathcal{X}$ $\alpha_1 \neq \alpha_2$:

$$\hat{p}_{\alpha_1 \cup \alpha_2} = \frac{|\alpha_1| \hat{p}_{\alpha_1} + |\alpha_2| \hat{p}_{\alpha_2}}{|\alpha_1| + |\alpha_2|} \tag{150}$$

Relation (150) lets us evaluate $\bar{V}(\alpha_1 \cup \alpha_2)$ without summing over all points in $\alpha_1 \cup \alpha$, so we can reuse the work we did when we previously computed $\bar{V}(\alpha_1)$, $\bar{V}(\alpha_2)$.

We obtain the doubly-approximate set of D distributions $\bar{\bar{\mathbf{p}}}^*$ by setting for all $x \in \mathcal{X}$, $d \in \{1, \ldots, D\}$

$$\bar{\bar{p}}_x^{(d)*} = \hat{p}_{\alpha(x)}^{(d)} \tag{151}$$

where $\alpha(x) \in \bar{\bar{\mathcal{A}}}^*$ is the partition element containing x, and $\hat{p}_\alpha^{(d)}$ (36) is the average of $\hat{p}^{(d)}$ over $\alpha \subset \mathcal{X}$.

We know that the partition $\bar{\bar{\mathcal{A}}}^*$ computed by algorithm (1) is suboptimal (that is, does not maximize $\bar{V}$ (147)) because it only tries to merge pairs of existing partition elements rather than triplets or larger collections; it is possible to construct an example where the optimal partition consists of a single set (all of $\mathcal{X}$) but no pair of partition elements of $\mathcal{A}(\hat{\mathbf{p}})$ is "worth" merging in the sense of satisfying the loop condition in line (3).

The running time of the heuristic (1) is $O(|\operatorname{Supp}(\mathcal{F})|^2) = O(n)$, where $n = \sum_{d=1}^D n_d$: In line (2), we have $|\mathcal{A}(\hat{\mathbf{p}})| = |\operatorname{Supp}(\mathcal{F})| = O(\sqrt{n})$. To evaluate the max over $\alpha_1, \alpha_2 \in \mathcal{A}$ for the first time in line (3) we consider $O(|\mathcal{A}(\hat{\mathbf{p}})|^2) = O(n)$ pairs of partition elements. On every subsequent iteration of the loop, we add one new partition element $(\alpha_1^* \cup \alpha_2^*)$ to $\mathcal{A}$ and remove two partition elements (α_1^* and α_2^*) from $\mathcal{A}$, so there are $O(|\mathcal{A}|) = O(\sqrt{n})$ new pairs of distinct partition elements to check in computing the max in line (3). The size of $|\mathcal{A}|$ decrements by one each time the loop is run, so the loop is run at most $O(|\mathcal{A}(\hat{\mathbf{p}})|) = O(\sqrt{n})$ times, so the overall complexity of algorithm 1 is $O(n) + O(\sqrt{n})O(\sqrt{n}) = O(n)$.

If the support set size $|\mathcal{X}|$ is unknown, then we vary the assumed support set size $|\mathcal{X}|$ to optimize $\bar{V}(\bar{\bar{\mathcal{A}}}^*(\hat{\mathbf{p}}, \mathbf{n}, |\mathcal{X}|))$ using a bisection search on the range $[|\mathcal{X}(\hat{\mathbf{p}})|, \max_{d \in [1,D]} n_d^2]$, increasing the overall running time of the algorithm (optimizing over both the level set partition $\mathcal{A}$ and the support set size) to $O(n \log(n))$. Numerical experiments show that $\bar{V}(\bar{\bar{\mathcal{A}}}^*(\hat{\mathbf{p}}, \mathbf{n}, |\mathcal{X}|))$ is not unimodal in $|\mathcal{X}|$, so this heuristic might choose a suboptimal support set size.

The $O(n)$ worst-case running time of algorithm (1) turns out to be slow for examples of practical interest (in making the performance plots of Section 6), so we introduce another heuristic to speed things up at the cost of adding some tunable knobs to the algorithm. The heuristic is to avoid checking all pairs of partition elements $\alpha_1, \alpha_2 \in \mathcal{A}$, instead checking only the "promising" pairs. A pair $\alpha_1, \alpha_2 \in \mathcal{A}$ is called *promising* if the point $(\hat{p}_{\alpha_1}^{(d)})_{d=1}^D \in \mathbb{R}^D$ is one of the k nearest neighbors of the point $(\hat{p}_{\alpha_2}^{(d)})_{d=1}^D \in \mathbb{R}^D$ in Euclidean distance[10]. This heuristic is motivated by the observation that the level sets that get merged in the loop of algorithm (1) tend to correspond to nearby points in Euclidean distance. Then we modify algorithm (1) to first compute the set of k nearest neighbors (doable in time $O(|\operatorname{Supp}(\mathcal{F})| \log(|\operatorname{Supp}(\mathcal{F})|)) = O(\sqrt{n} \log(\sqrt{n})) = O(\sqrt{n} \log(n))$) and then to update the set of nearest neighbors after each loop execution. This updating can be done heuristically by taking the k nearest neighbors of $\hat{p}_{\alpha_1 \cup \alpha_2}^{(d)}$ to be the k nearest points chosen from among only the union of the k nearest neighbors of $(\hat{p}_{\alpha_1}^{(d)})_{d=1}^D$ and the k nearest neighbors of $(\hat{p}_{\alpha_2}^{(d)})_{d=1}^D$ (rather than the k nearest points chosen from among $(\hat{p}_\alpha^{(d)})_{d=1}^D$ for all $\alpha \in \mathcal{A}$). The loop is executed $O(\sqrt{n})$ times, so the overall running time of the algorithm is $O(\sqrt{n} \log(n) + k\sqrt{n})$. The time to compute the empirical histograms $\hat{\mathbf{p}}$ is $O(n)$, asymptotically in n dominating the runtime of our algorithm for large n, but tends to take less time in our use cases than running algorithm 1 modified with our k nearest neighbor heuristic. For the performance plots of Section 6 we used $k = 5$.

Figures 2 and 14 show the doubly-approximate PML level set partition $\bar{\bar{\mathcal{A}}}^*$ (149) for $D = 2$ and $D = 3$, respectively.

Appendix I. Mutual and Lautum information

For the case of estimating the mutual information $I(X;Y)$, defined as

$$I(X;Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p_{x,y} \log\left(\frac{p_{x,y}}{p_x p_y}\right), \tag{152}$$

10. The nearest neighbor relation is not symmetric, so α_1, α_2 being promising does not imply that α_2, α_1 is a promising pair.

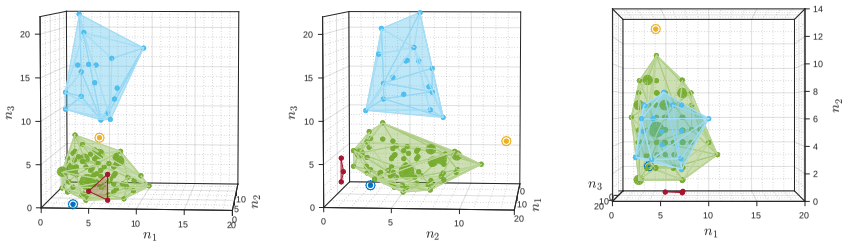


Figure 14: Computing the doubly-approximate PML distribution level set partition $\bar{\bar{\mathcal{A}}}^*$ (149) by clumping entries of a 3-D fingerprint $\mathcal{F}_{i,j,k}$ using algorithm 1 modified with the 5-nearest neighbors heuristic as described in Section H.4, shown in three projections. The underlying distributions are $p^{(1)} = p^{(2)}$ uniform on $\{1, \ldots, 100\}$ and $p^{(3)}$ is a mixture of two uniforms, with half its mass uniform on $\{1, \ldots, 20\}$ and the other half uniform on $\{21, \ldots, 100\}$. The sample size is 500 for all three empirical distributions $(\hat{p}^{(d)})_{d=1}^3$. Colored convex hulls correspond to symbols assigned to the same clump (partition element). The marker size of point (i, j, k) is roughly proportional to $\mathcal{F}_{i,j,k}$. We can see two large clumps, corresponding to symbols $\{1, \ldots, 20\}$ and $\{21, \ldots, 100\}$.

or lautum information (Palomar and Verdú, 2008), defined as

$$L(X;Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p_x p_y \log \left(\frac{p_x p_y}{p_{x,y}} \right), \tag{153}$$

we have n samples from a joint distribution p on $\mathcal{X} \times \mathcal{Y}$ with joint histogram $\hat{p}$. Here $p_x = \sum_y p_{x,y}, p_y = \sum_x p_{x,y}$ are the marginal distributions of X and Y, respectively.

I.1. Product of permutation groups: one candidate for PML

It is clear that the mutual information and the lautum information are invariant to the product of two permutation groups $\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y}$, acting on p as:

$$((\sigma_\mathcal{X}, \sigma_\mathcal{Y})p)_{x,y} \triangleq p_{\sigma_\mathcal{X}(x), \sigma_\mathcal{Y}(y)} \tag{154}$$

for all $(\sigma_\mathcal{X}, \sigma_\mathcal{Y}) \in \mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y}$. Let $(\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y})p$ denote the orbit of p under the group action. Following the general profile maximum likelihood methodology for the product of these two permutation groups, the PML distribution p^* maximizes the following quantity over a set

of distribution $\mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$:

$$\begin{aligned}
\mathbb{P}_p((\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y})\hat{p}) &= \sum_{\hat{p}' \in (\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y})\hat{p}} \mathbb{P}_p(\hat{p}') && (155)\\
&\overset{\text{(a)}}{=} \frac{|(\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y})\hat{p}|}{|\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y}|} \sum_{(\sigma_\mathcal{X}, \sigma_\mathcal{Y}) \in \mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y}} \mathbb{P}_p((\sigma_\mathcal{X}, \sigma_\mathcal{Y})\hat{p}) && (156)\\
&= \left(\prod_{i \geq 0} \frac{1}{\mathcal{F}_i!} \right) \binom{n}{n\hat{p}} \sum_{(\sigma_\mathcal{X}, \sigma_\mathcal{Y}) \in \mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y}} \prod_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p_{x,y}^{n\hat{p}_{\sigma_{(\sigma_\mathcal{X}, \sigma_\mathcal{Y})\mathcal{X}}(x), \sigma_\mathcal{Y}(y)}} && (157)\\
&= \left(\prod_{i \geq 0} \frac{1}{\mathcal{F}_i!} \right) \binom{n}{n\hat{p}} \operatorname{perm}\left(\left(p_{x,y}^{n\hat{p}_{x',y'}} \right)_{(x,y),(x',y') \in \mathcal{X} \times \mathcal{Y}} \right), && (158)
\end{aligned}$$

where in (a) the prefactor corrects for multiple-counting points of the orbit $(\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y})\hat{p}$ when we sum over $(\sigma_\mathcal{X}, \sigma_\mathcal{Y}) \in (\mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y})$, and where $(\mathcal{F}_i)_{i \geq 0} = (|\{(x,y) \in \mathcal{X} \times \mathcal{Y} : n\hat{p}_{x,y} = i\}|)_{i \geq 0}$ is the fingerprint of the distribution when one treats $\mathcal{X} \times \mathcal{Y}$ as a single alphabet. Note that in this setting $(\mathcal{F}_i)_{i \geq 0}$ is not partially sufficient for estimating the mutual (lautum) information: the mutual (lautum) information cannot be written as a functional of the sorted probabilities of $p_{x,y}$.

In this setting, it follows from Theorem 6 that the partially sufficient statistic could be defined via the following equivalence class representation: for any empirical distribution $\hat{p}^{(1)} = (\hat{p}^{(1)}_{x,y}), \hat{p}^{(2)} = (\hat{p}^{(2)}_{x,y})$, we say $\hat{p}^{(1)}$ is equivalent to $\hat{p}^{(2)}$ if there exists an element $g \in \mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y}$ such that $g\hat{p}^{(1)} = \hat{p}^{(2)}$. The partially sufficient statistic is the corresponding equivalent classes.

In this PML formulation, we analyze the cardinality of the corresponding partially sufficient statistic below. This number appeared in (OEIS sequence A007716), and its precise asymptotics is obtained in the following theorem.

Theorem 12 *For any two empirical distributions $\hat{p}^{(1)} = (\hat{p}^{(1)}_{x,y}), \hat{p}^{(2)} = (\hat{p}^{(2)}_{x,y})$ with identical sample size n, we say $\hat{p}^{(1)}$ is equivalent to $\hat{p}^{(2)}$ if there exists an element $(\sigma_\mathcal{X}, \sigma_\mathcal{Y}) \in \mathcal{S}_\mathcal{X} \times \mathcal{S}_\mathcal{Y}$ such that $(\sigma_\mathcal{X}, \sigma_\mathcal{Y})\hat{p}^{(1)} = \hat{p}^{(2)}$, where $\mathcal{S}_\mathcal{X}, \mathcal{S}_\mathcal{Y}$ are the permutation groups on $\mathcal{X}, \mathcal{Y}$, respectively. Then, the number of equivalence classes is given by $e^{(1+o(1))n \log n}$.*

Theorem 12 shows that the cardinality of the partially sufficient statistic is no longer sub-exponential. This negative result immediately renders Theorem 3 not applicable, which leaves us with no guarantee on the performance of this type of PML on estimating mutual information or lautum information.

I.2. Three permutation groups: the other candidate for the PML

The mutual information $I(X;Y)$ can be rewritten as

$$\begin{aligned}
I(X;Y) &= H(P_X) + H(P_Y) - H(P_{XY}) && (159)\\
&= \sum_{x \in \mathcal{X}} p_x \log \left(\frac{1}{p_x} \right) + \sum_{y \in \mathcal{Y}} p_y \log \left(\frac{1}{p_y} \right) - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p_{x,y} \log \left(\frac{1}{p_{x,y}} \right), && (160)
\end{aligned}$$

where $H(P_X) = \sum_x p_x \log \frac{1}{p_x}$ denotes the Shannon entropy of discrete distribution P_X.

Following the discussion in Section 3.1, we introduce

$$(\mathcal{F}_i^{XY})_{i\geq 0} = (|\{(x,y) \in \mathcal{X} \times \mathcal{Y} : n\hat{p}_{x,y} = i\}|)_{i\geq 0} \tag{161}$$

$$(\mathcal{F}_i^{X})_{i\geq 0} = (|\{x \in \mathcal{X} : n\hat{p}_{x} = i\}|)_{i\geq 0} \tag{162}$$

$$(\mathcal{F}_i^{Y})_{i\geq 0} = (|\{y \in \mathcal{Y} : n\hat{p}_{y} = i\}|)_{i\geq 0}, \tag{163}$$

and it is clear that $\left(\mathcal{F}_i^{XY}\right)_{i\geq 0}$ is partially sufficient for $H(P_{XY})$, $\left(\mathcal{F}_i^{X}\right)_{i\geq 0}$ is partially sufficient for $H(P_X)$, and $\left(\mathcal{F}_i^{Y}\right)_{i\geq 0}$ is partially sufficient for $H(P_Y)$. Since $I(X;Y) = H(P_X) + H(P_Y) - H(P_{XY})$, it motivates the following PML definition:

$$p^* \triangleq \operatorname*{argmax}_{p\in\mathcal{P}} \mathbb{P}_p\left(\mathcal{F}^{XY}, \mathcal{F}^X, \mathcal{F}^Y\right). \tag{164}$$

In other words, the PML distribution $(p^*_{x,y})$ on $\mathcal{X}\times\mathcal{Y}$ maximizes the probability of observing the three joint fingerprints $\left(\mathcal{F}^{XY}, \mathcal{F}^X, \mathcal{F}^Y\right)$. This PML formulation, which only applies to mutual information but not lautum information, does not suffer from the super-exponential cardinality problem. Applying Theorem 10, it follows that the cardinality of the statistic is at most

$$\left(e^{\pi\sqrt{\frac{2n}{3}}}\right)^3 = e^{\pi\sqrt{6n}}, \tag{165}$$

which is still sub-exponential. However, it seems to be a non-trivial task to even approximately solve the PML formulation (164). Moreover, this PML formulation does not apply to lautum information.

Note that there may not exist a group G (a subgroup of the symmetric group $\mathcal{S}_{\mathcal{X}\times\mathcal{Y}}$) acting on the joint empirical distributions $\hat{p}$ such that the set of distinct triplets $(\mathcal{F}^{XY}, \mathcal{F}^X, \mathcal{F}^Y)$ is in bijection with the set of distinct orbits under the action of G. For example, let $\hat{p}$, $\hat{p}'$, and $\hat{p}''$ be three empirical distributions on $\mathcal{X} = \{x_1, x_2\}$, $\mathcal{Y} = \{y_1, y_2, y_3\}$ with sample size $n = 1110$:

$$n\hat{p} = \begin{pmatrix} 1 & 4 & 100 \\ 2 & 3 & 1000 \end{pmatrix} \qquad n\hat{p}' = \begin{pmatrix} 2 & 3 & 100 \\ 1 & 4 & 1000 \end{pmatrix} \qquad n\hat{p}'' = \begin{pmatrix} 2 & 100 & 3 \\ 1 & 1000 & 4 \end{pmatrix} \tag{166}$$

where rows (columns) correspond to elements of $\mathcal{X}$ ($\mathcal{Y}$). $\hat{p}$, $\hat{p}'$, and $\hat{p}''$ have the same $\mathcal{X}$- and $\mathcal{Y}$-marginal distributions ($(105, 1005)/1110$ and $(3, 7, 1100)/1110$, respectively) and the same set of entries, so their fingerprint triplets $(\mathcal{F}^{XY}, \mathcal{F}^X, \mathcal{F}^Y)$ are the same.

Suppose there exists a group $G \subset \mathcal{S}_{\mathcal{X}\times\mathcal{Y}}$ such that the orbit $G\hat{p}$ contains exactly those distributions with the same fingerprint triplet $(\mathcal{F}^{XY}, \mathcal{F}^X, \mathcal{F}^Y)$. Then $\hat{p}', \hat{p}'' \in G\hat{p}$. Let $\sigma \in G$, where $\hat{p}' = \sigma\hat{p}$. Then σ switches x_1 and x_2 iff $y \neq y_3$:

$$(\sigma p)_{x,y} = \begin{cases} p_{x_2,y} & : x = x_1,\ y \neq y_3 \\ p_{x_1,y} & : x = x_2,\ y \neq y_3 \\ p_{x,y} & : \text{otherwise} \end{cases} \tag{167}$$

On the other hand,

$$n(\sigma\hat{p}'') = \begin{pmatrix} 1 & 1000 & 3 \\ 2 & 100 & 4 \end{pmatrix} \tag{168}$$

has $\mathcal{X}$-marginal $(1004, 106)/1110$, and thus a different $\mathcal{F}^X$, so the action of σ does not preserve the triplet $(\mathcal{F}^{XY}, \mathcal{F}^X, \mathcal{F}^T)$, so group G does not exist.

Appendix J. Estimating the support set size of the approximate PML distribution

We first state our results in the notation introduced in Sections 4.5 and 4.6.

Theorem 13 *Let $\bar{p}^*$ be the approximate PML distribution. Then $\mathcal{X}_{0:i^*} \in \mathcal{A}(\bar{p}^*)$ is a level set of $\bar{p}^*$, where*

$$i^* \triangleq \operatorname*{argmax}_{i\in[1,F_+]} \left(\sup_{\mathcal{F}_0\in\mathbb{N}} \left(-\log(\mathcal{F}_0!) + \bar{V}(\mathcal{X}_{0:i})\right) + \bar{V}_{i+1}\right) \tag{169}$$

For each value of $i \in [1, F_+]$ the supremum over $\mathcal{F}_0$ in (169) is finite and we compute it below.

Let $K_i \triangleq |\mathcal{X}_{1:i}|$ and $N_i \triangleq \sum_{x\in\mathcal{X}_{1:i}} n\hat{p}_x$.

If $i = 1$ and $\mathcal{F}_1 \geq 1$, then $N_1 = K_1 = \mathcal{F}_1 \geq 1$ is the number of symbols seen exactly once in the sample, and the supremum in (169) is equal to $N_1 \log\left(\frac{N_1}{n}\right)$. If $\mathcal{F}_1 > 1$, then the supremum is approached as $\mathcal{F}_0 \to \infty$ and is not achieved by any $\mathcal{F}_0 \in \mathbb{N}$. If $\mathcal{F}_1 = 1$, then the supremum is achieved by all $\mathcal{F}_0 \in \mathbb{N}$, so we arbitrarily choose $\mathcal{F}_0^{(1)} = 0$ as an achiever.

If $i > 1$ or $\mathcal{F}_1 = 0$, then $N_i > K_i$ and the supremum in (169) is achieved by $\mathcal{F}_0^{(i)} \in \mathbb{N}$ where

$$\mathcal{F}_0^{(i)} \triangleq \operatorname*{argmax}_{\mathcal{F}_0\in\mathbb{N}} \left(\log\left(\frac{(K_i+\mathcal{F}_0)!}{\mathcal{F}_0!}\right) - N_i \log\left(K_i + \mathcal{F}_0\right)\right) \begin{cases} = 0 & : N_i \geq K_i H_{K_i} \\ \in [0, \frac{K_i^2 - N_i}{N_i - K_i} + 1] & : K_i H_{K_i} > N_i > K_i \end{cases} \tag{170}$$

where $H_k \triangleq \sum_{j=1}^k \frac{1}{j}$ is the k-th harmonic number.

If the supremum in (169) is achieved by finite $\mathcal{F}_0 = \mathcal{F}_0^{(i^*)}$ (this is the case unless $i^* = 1$ and $\mathcal{F}_1 > 1$) then the approximate PML distribution $\bar{p}^*$ has finite support set size $\bar{K}^* = \mathcal{F}_0^{(i^*)} + \hat{K}$, where $\hat{K}$ is the support set size of the empirical distribution $\hat{p}$. Then we compute $\bar{p}^*$ as in Section 4.5 with known support set size $\bar{K}^*$. If the supremum is not achieved by finite $\mathcal{F}_0$ (this is the case if $i^* = 1$ and $\mathcal{F}_1 > 1$), then we say $\bar{p}^*$ has a continuous part of mass N_1/n and discrete part $\bar{p}^*_{\mathrm{d}}$ of mass $1 - N_1/n$ supported on $\mathcal{X}_{2:F_+} = \mathcal{X} \setminus \mathcal{X}_{0:1}$. Then we compute $\bar{p}^*_{\mathrm{d}}$ as in Section 4.5 with known support set size $|\mathcal{X}_{2:F_+}|$.

Since the function in the argmax in (170) is unimodal in $\mathcal{F}_0$ and $0 \leq \mathcal{F}_0^{(i)} \leq \frac{K_i^2 - N_i}{N_i - K_i} + 1 \leq N_i^2 + 2$ (where the last inequality follows from $N_i \geq K_i + 1$), we can compute the optimizing integer $\mathcal{F}_0^{(i)}$ in $O(\log(N_i))$ iterations of a bisection search. We must do this bisection search

$F_+ = |\operatorname{Supp}(\mathcal{F}_+)|$ times. As remarked earlier, since $|\operatorname{Supp}(\mathcal{F}_+)| \le \sqrt{2n}+1$ and $N_i \le n$ for all i, then the total run time of our support set size optimization scheme is $O(\sqrt{n}\log(n))$.

Thus to estimate the support set size of the approximate PML distribution we solve F_+ optimization problems, corresponding to estimation of the support set size of a uniform distribution (since all symbols in $\mathcal{X}_{0:i}$ are assigned to a single level set) given a sample of size N_i with K_i distinct symbols.

Proof

Optimizing over the support set size K, the approximate PML distribution $\bar{p}^*$ satisfies:

$$\log(\mathbb{P}_{\bar{p}^*}(\mathcal{F})) = \sup_K \Big(-\log((K-\hat{K})!) + \max_{p\in\mathcal{P}_K} \big(\bar{V}(p)\big)\Big) \tag{171}$$

$$= \sup_{\mathcal{F}_0} \Big(-\log(\mathcal{F}_0!) + \max_{p\in\mathcal{P}_{\hat{K}+\mathcal{F}_0}} \big(\bar{V}(p)\big)\Big) \tag{172}$$

$$\overset{(a)}{=} \sup_{\mathcal{F}_0} \Big(-\log(\mathcal{F}_0!) + \bar{V}_0\Big) \tag{173}$$

$$\overset{(b)}{=} \sup_{\mathcal{F}_0} \Big(-\log(\mathcal{F}_0!) + \max_{i\in[1,F_+]} \big(\bar{V}(\mathcal{X}_{0,i}) + \bar{V}_{i+1}\big)\Big) \tag{174}$$

$$\overset{(c)}{=} \max_{i\in[1,F_+]} \Big(\bar{V}_{i+1} + \sup_{\mathcal{F}_0} \big(\bar{V}(\mathcal{X}_{0,i}) - \log(\mathcal{F}_0!)\big)\Big) \tag{175}$$

$$\overset{(d)}{=} \max_{i\in[1,F_+]} \left(\bar{V}_{i+1} + N_i \log\left(\frac{N_i}{n}\right) + \sup_{\mathcal{F}_0} \underbrace{\left(\log\left(\frac{(K_i+\mathcal{F}_0)!}{\mathcal{F}_0!}\right) - N_i \log\left(K_i+\mathcal{F}_0\right)\right)}_{f(\mathcal{F}_0,K_i,N_i)}\right) \tag{176}$$

where (a) follows from (48), (b) follows from (46), (c) follows since $\bar{V}_i$ is independent of $\mathcal{F}_0$ (and hence K) for $i \ge 1$, and (d) from (39) and (50). We use $\sup_{\mathcal{F}_0}$ (rather than $\max_{\mathcal{F}_0}$) since a maximizing value of $\mathcal{F}_0$ might not exist. We later show that the supremum in (176) is finite, so we interchange the order of the max and supremum. The function $f(\mathcal{F}_0, K, N)$ is defined for non-negative arguments as in (176). Note that we do not allow $i = 0$ in the optimization (176) in order to exclude $\mathcal{X}_{0,0}$ (the set of unseen symbols) as a level set of p; otherwise, due to the averaging-over-clumps property (37), we would have p vanish on $\mathcal{X}_{0,0}$, violating the assumption that the support set size of p is K. It is possible to have $\bar{K}^* = \hat{K} = |\operatorname{Supp}(\hat{p})|$, corresponding to the case $\mathcal{F}_0 = 0$, so that $\mathcal{X}_{0,i} = \mathcal{X}_{1,i}$ for all i.

Let's compute the supremization in (176). We can check that $f(\mathcal{F}_0, N, N) < 0$ for all $N > 1$ and $\mathcal{F}_0 \ge 0$, and

$$\lim_{\mathcal{F}_0\to\infty} f(\mathcal{F}_0, N, N) = 0 \tag{177}$$

so if $i = 1$ and $\mathcal{F}_1 > 1$, then $N_1 = K_1 = \mathcal{F}_1 > 1$ and the supremum in (176) is equal to 0, is approached as $\mathcal{F}_0 \to \infty$, and is not achieved by any $\mathcal{F}_0 \in \mathbb{N}$.

We can check that $f(\mathcal{F}_0, 1, 1) = 0$ for all $\mathcal{F}_0$, so if $i = 1$ and $\mathcal{F}_1 = 1$, then $N_1 = K_1 = \mathcal{F}_1 = 1$ and the supremum in (176) is equal to 0 and is achieved by all $\mathcal{F}_0 \in \mathbb{N}$, so we arbitrarily choose $\mathcal{F}_0^{(1)} = 0$ as an achiever.

If $i > 1$ or $\mathcal{F}_1 = 0$, then $N_i > K_i$. When $N > K$ the function $f(\mathcal{F}_0, K, N)$ is continuous in $\mathcal{F}_0$ [11] and unimodal with a global maximum. Solving

$$\frac{\partial}{\partial \mathcal{F}_0} f(\mathcal{F}_0, K, N^*) = 0 \tag{178}$$

for N^* we find $N^* = KH_K$, where H_K is the K-th harmonic number. Thus if $N_i \ge K_i H_{K_i}$, then the supremum in (176) is achieved by $\mathcal{F}_0 = 0$. If $N_i < K_i H_{K_i}$, then we can upper bound the supremizing value $\mathcal{F}_0$ by finding the unique inflection point of f in $\mathcal{F}_0$ beyond the global maximum. Setting the "discrete" second derivative to 0 (we take the discrete derivative because the function $\frac{\partial f}{\partial \mathcal{F}_0}$ is easier to work with for integer $\mathcal{F}_0$):

$$\frac{\partial}{\partial \mathcal{F}_0} \left(f(\mathcal{F}_0+1, K, N) - f(\mathcal{F}_0, K, N)\right)\Big|_{\mathcal{F}_0=\mathcal{F}_0^*} = 0 \tag{179}$$

we find $\mathcal{F}_0^* = \frac{K^2-N}{N-K}$. If $K < N < KH_K$ then $K > 1$ (since $1 = H_1$) and $N < K^2$, so $\mathcal{F}_0^* > 0$. Thus we can upper bound the inflection point of f by $\frac{K^2-N}{N-K}+1$, so if $K_i < N_i < K_i H_{K_i}$, then $\mathcal{F}_0^{(i)} \in [0, \frac{K_i^2-N_i}{N_i-K_i}+1]$.

■

References

Jayadev Acharya, Alon Orlitsky, and Shengjun Pan. The maximum likelihood probability of unique-singleton, ternary, and length-7 patterns. In *Information Theory (ISIT), 2009 IEEE International Symposium on*, pages 1135–1139. IEEE, 2009.

Jayadev Acharya, Hirakendu Das, Ashkan Jafarpour, Alon Orlitsky, and Shengjun Pan. Competitive closeness testing. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 47–68, 2011.

Jayadev Acharya, Hirakendu Das, Alon Orlitsky, and Ananda Theertha Suresh. A unified maximum likelihood approach for estimating symmetric properties of discrete distributions. In *International Conference on Machine Learning*, pages 11–21, 2017.

Dragi Anevski, Richard D Gill, and Stefan Zohren. Estimating a probability mass function with unknown labels. *The Annals of Statistics*, 45(6):2708–2735, 2017.

FC Auluck. On partitions of bipartite numbers. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 49, pages 72–83. Cambridge Univ Press, 1953.

I Bárány and AM Vershik. On the number of convex lattice polytopes. *Geometric and Functional Analysis*, 2(4):381–393, 1992.

D Basu. On partial sufficiency: A review. In *Selected Works of Debabrata Basu*, pages 291–303. Springer, 2011.

11. Replacing $x!$ with $\Gamma(x+1)$, where Γ is the gamma function.

Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 259–269. IEEE, 2000.

Allan Birnbaum. On the foundations of statistical inference. *Journal of the American Statistical Association*, 57(298):269–306, 1962.

David A Blackwell and Meyer A Girshick. *Theory of games and statistical decisions*. Courier Corporation, 1979.

Yuheng Bu, Shaofeng Zou, Yingbin Liang, and Venugopal V Veeravalli. Estimation of kl divergence: Optimal minimax rate. *arXiv preprint arXiv:1607.02653*, 2016.

Chun Lam Chan, Winston Fernandes, Navin Kashyap, and Majunath Krishnapur. Phase transitions for the uniform distribution in the pattern maximum likelihood problem and its Bethe approximation. *SIAM J. Discrete Math*, 31:597–631, 2015.

Bradley Efron. Maximum likelihood and decision theory. *The Annals of Statistics*, pages 340–356, 1982.

Winston Fernandes and Navin Kashyap. A phase transition for the uniform distribution in the pattern maximum likelihood problem. In *ITW*, 2013.

Jaroslav Hájek. On basic concepts of statistics. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probabilities*, volume 1, pages 139–162, 1967.

Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Minimax rate-optimal estimation of divergences between discrete distributions. *arXiv preprint arXiv:1605.09124*, 2016a.

Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Minimax rate-optimal estimation of KL divergence between discrete distributions. In *Information Theory and Its Applications (ISITA), 2016 International Symposium on*, pages 256–260. IEEE, 2016b.

Godfrey H Hardy and Srinivasa Ramanujan. Asymptotic formulaæ in combinatory analysis. *Proceedings of the London Mathematical Society*, 2(1):75–115, 1918.

David A Harville. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72(358):320–338, 1977.

Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. *Information Theory, IEEE Transactions on*, 61(5): 2835–2885, 2015.

Jiantao Jiao, Kartik Venkat, and Tsachy Weissman. Relations between information and estimation in discrete-time Lévy channels. *IEEE Transactions on Information Theory*, 63(6):3579–3594, 2017.

Jiantao Jiao, Yanjun Han, and Tsachy Weissman. Minimax estimation of the L_1 distance. *IEEE Transactions on Information Theory*, 64(10):6672–6706, 2018.

A Kolmogoroff. Sur lestimation statistique des parametres be la loi de gauss. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 6(1):3–32, 1942.

L Le. Sufficiency and approximate sufficiency. *The Annals of Mathematical Statistics*, pages 1419–1455, 1964.

Lucien Le Cam. Maximum likelihood: an introduction. *Statistics Branch, Department of Mathematics, University of Maryland*, 1979.

Erich Leo Lehmann and George Casella. *Theory of point estimation*, volume 31. Springer, 1998.

Susan A Murphy and Aad W Van der Vaart. On profile likelihood. *Journal of the American Statistical Association*, 95(450):449–465, 2000.

OEIS Foundation Inc. The online encyclopedia of integer sequences (OEIS)–A007716. `https://oeis.org/A007716`. Accessed: 2017-05-16.

Alon Orlitsky, Sajama, Narayana Santhanam, Krishnamurthy Viswanathan, and Junan Zhang. Algorithms for modeling distributions over large alphabets. In *Information Theory (ISIT), 2004 IEEE International Symposium on*, page 304. IEEE, 2004a.

Alon Orlitsky, Narayana P Santhanam, Krishnamurthy Viswanathan, and Junan Zhang. On modeling profiles instead of values. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 426–435. AUAI Press, 2004b.

Daniel P Palomar and Sergio Verdú. Lautum information. *IEEE transactions on information theory*, 54(3):964–975, 2008.

Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15 (6):1191–1253, 2003.

H Desmond Patterson and Robin Thompson. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554, 1971.

Dmitri S. Pavlichin, Jiantao Jiao, and Tsachy Weissman. Approximate profile maximum likelihood (software), 2017. URL `https://doi.org/10.5281/zenodo.1043617`. URL `https://github.com/dmitrip/PML`.

Aditi Raghunathan, Greg Valiant, and James Zou. Estimating the unseen from multiple populations. *arXiv preprint arXiv:1707.03854*, 2017.

Gregory Valiant and Paul Valiant. The power of linear estimators. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 403–412. IEEE, 2011a.

Gregory Valiant and Paul Valiant. Estimating the unseen: An $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *the 43rd ACM Symposium on Theory of Computing (STOC)*, 2011b.

Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. In *Advances in Neural Information Processing Systems*, pages 2157–2165, 2013.

L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

Paul Valiant. *Testing Symmetric Properties of Distributions*. PhD thesis, Massachusetts Institute of Technology, 2008.

SS Vallender. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786, 1974.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Pascal O. Vontobel. The Bethe approximation of the pattern maximum likelihood distribution. In *ISIT*, 2012.

Abraham Wald. *Statistical decision functions*. Wiley, 1950.

Haizhou Wang and Mingzhou Song. Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R Journal*, 3(2):29–33, 2011.

Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *arXiv preprint arXiv:1504.01227*, 2015.

Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6): 3702–3720, 2016.

Journal of Machine Learning Research 20 (2019) 1-26 Submitted 4/18; Revised 2/19; Published 5/19

ADMMBO: Bayesian Optimization with Unknown Constraints using ADMM

Setareh Ariafar SARIAFAR@ECE.NEU.EDU
Electrical and Computer Engineering Department
Northeastern University
Boston, MA 02115, USA

Jaume Coll-Font JCOLLFONT@GMAIL.COM
Computational radiology Laboratory
Boston Children's Hospital
Boston, MA 02115, USA

Dana Brooks BROOKS@ECE.NEU.EDU
Electrical and Computer Engineering Department
Northeastern University
Boston, MA 02115, USA

Jennifer Dy JDY@ECE.NEU.EDU
Electrical and Computer Engineering Department
Northeastern University
Boston, MA 02115, USA

Editor: Bayesian Optimization Special Issue

Abstract

There exist many problems in science and engineering that involve optimization of an unknown or partially unknown objective function. Recently, Bayesian Optimization (BO) has emerged as a powerful tool for solving optimization problems whose objective functions are only available as a black box and are expensive to evaluate. Many practical problems, however, involve optimization of an unknown objective function subject to unknown constraints. This is an important yet challenging problem for which, unlike optimizing an unknown function, existing methods face several limitations. In this paper, we present a novel constrained Bayesian optimization framework to optimize an unknown objective function subject to unknown constraints. We introduce an equivalent optimization by augmenting the objective function with constraints, introducing auxiliary variables for each constraint, and forcing the new variables to be equal to the main variable. Building on the Alternating Direction Method of Multipliers (ADMM) algorithm, we propose ADMM-Bayesian Optimization (ADMMBO) to solve the problem in an iterative fashion. Our framework leads to multiple unconstrained subproblems with unknown objective functions, which we then solve via BO. Our method resolves several challenges of state-of-the-art techniques: it can start from infeasible points, is insensitive to initialization, can efficiently handle 'decoupled problems' and has a concrete stopping criterion. Extensive experiments on a number of challenging BO benchmark problems show that our proposed approach outperforms the state-of-the-art methods in terms of the speed of obtaining a feasible solution and convergence to the global optimum as well as minimizing the number of total evaluations of unknown objective and constraints functions.

©2019 Setareh Ariafar, Jaume Coll-Font, Dana Brooks, Jennifer Dy..

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-227.html.

Keywords: Bayesian Optimization, Gaussian Processes, ADMM, Expected Improvement

1. Introduction

Bayesian optimization (BO) has been shown to be a powerful tool for solving optimization problems whose objective functions are unknown and expensive to evaluate (Brochu et al., 2010a; Martinez-Cantin et al., 2007; Hutter et al., 2011; Torn and Zilinskas, 1989). For example, in drug design (Azimi et al., 2012; Scott, 2010; Brochu et al., 2010b), where the goal is to maximize the efficacy of a drug, the evaluation of the objective function, i.e., drug efficacy, across multiple drug formulations requires producing and testing new drugs, which would be subject to resource and cost limitations. As another example, minimizing the validation error of a machine learning model, such as hyperparameter tuning of a deep neural network (LeCun et al., 2015), involves many evaluations of the objective function, i.e., the validation error, where each evaluation requires training and evaluating a new model (Bergstra et al., 2011; Hoffman et al., 2014; Snoek et al., 2012; Swersky et al., 2013).

In many real-world problems, the desired solution, in addition to optimizing the objective function, must satisfy constraints that are also unknown and expensive to evaluate (Shahriari et al., 2016). For example, in the drug design problem, the goal is often to maximize the drug efficacy while limiting its side effects. In the hyperparameter tuning problem in machine learning, the optimal hyperparameters not only must minimize the validation error, but also must ensure that the prediction time of the learned model is sufficiently short. The majority of existing work on BO has focused on the unknown-objective problem (Jones et al., 1998; Kushner, 1964; Lizotte, 2008; Jones, 2001; Hernández-Lobato et al., 2014; Cox and John, 1992; Wu et al., 2017), while only a few recent reports have addressed the problem in the unknown-objective unknown-constraint setting (Snoek, 2013; Gelbart et al., 2014; Gardner et al., 2014; Bernardo et al., 2011; Hernández-Lobato et al., 2015; Picheny et al., 2016; Gramacy et al., 2016; Picheny, 2014), (see Section 4 for a review).

1.1. Existing Challenges & Paper Contributions

In this subsection, we describe some of the limitations of the state-of-the-art constrained Bayesian optimization methods. First, several of these methods, including Expected Improvement with Constraints (EIC) (Schonlau et al., 1998; Snoek, 2013; Gelbart et al., 2014), Integrated Conditional Expected Improvement (IECI) (Bernardo et al., 2011) and Expected Volume Reduction (EVR) (Picheny, 2014), require a feasible initial point. However, in practice, a feasible point is often not available; hence, these algorithms require finding a feasible point first, which consumes part of the computational budget. Second, most existing constrained BO methods, including EIC, IECI, EVR, Augmented Lagrangian for Bayesian Optimization (ALBO) (Gramacy et al., 2016) and Slacked-augmented Lagrangian for Bayesian Optimization (Slack-AL) (Picheny et al., 2016), require joint evaluation of the objective function and all constraints at a candidate point in each step, in order to quantify its utility for finding a global optimum. However, for a large class of problems, labeled as "decoupled", the objective function and the constraints can each be evaluated independently. This can be especially advantageous in settings where some functions might be significantly more expensive to evaluate than the others (Gelbart, 2015). For example, in the hyperparameter tuning problem, evaluating the prediction time may not require training

the model and is often much cheaper than evaluating the validation error. Thus, methods that require joint evaluation of all unknown functions, including EIC, IECI, EVR, ALBO, and Slack-AL could increase the overall cost of solving decoupled problems more than might be necessary. Third, the majority of existing methods, including IECI, EVR, ALBO and Predictive Entropy Search with Constraints (PESC) (Hernández-Lobato et al., 2015), do not have closed form expressions for the so-called 'acquisition function', which is a key step in the BO algorithm. Thus, these methods need to approximate the acquisition function, typically via algorithms such as Expectation Propagation (Minka, 2001) or Monte-Carlo Sampling (Picheny et al., 2013), which often suffer from implementation difficulty and slow execution time, or may cause instabilities (Picheny et al., 2016; Gelbart, 2015). Finally, most of the BO methods fix a computational budget in terms of either wall-clock time or the number of function evaluations, and stop when the budget is exhausted. However, this budget is an additional parameter which must be hand tuned, and the performance of the BO method typically is highly dependent on it. A value that is too small may result in missing easy improvement while one that is too large might incur additional cost for an insignificant gain. Thus, having an automatic stopping criterion is highly desirable while many BO methods, including EIC, IECI, EVR and PESC, lack such a criterion.

In this paper, we propose a novel constrained BO framework for optimizing an unknown objective function subject to unknown constraints that resolves all the aforementioned challenges. First, we reformulate the problem into an equivalent unconstrained optimization. Since the joint (Bayesian) optimization of the unconstrained problem over the unknown objective function and unknown augmented constraints is challenging, we introduce auxiliary variables, one per constraint, and then force these variables to be equal to the the original variable, resulting in an equivalent constrained formulation, where the constraints are now known. The new formulation allows to perform the (Bayesian) optimization over each term independently, decoupling the objective function optimization from constraint satisfaction. To efficiently solve our proposed optimization, we adopt the Alternating Direction Method of Multipliers (ADMM) framework (Boyd et al., 2011; Hong and Luo, 2017; Parikh et al., 2014), which leads to solving an 'optimality subproblem', and a 'feasibility subproblem' for each constraint, at each iteration. The optimality subproblem minimizes the objective function close to current solutions of the feasibility subproblems, while each feasibility subproblem searches for a feasible solution of its constraint close to the current solution of the optimality subproblem. Our framework, which we call Alternating Direction Method of Multipliers for Bayesian Optimization (ADMMBO), provides the following advantages compared to the state-of-the-art methods (see Table 1 for a summary).

– Unlike many existing methods, ADMMBO can start from an infeasible initial point and gradually move towards a feasible point via solving the feasibility subproblems.

– Due to its separation of the optimizations over each expensive to evaluate function, i.e., objective function and each constraint, ADMMBO can handle decoupled problems efficiently, without requiring joint evaluation of all such functions at each candidate point.

– Because it decomposes the overall problem into separate subproblems, each involving only one expensive to evaluate function, resulting in simpler BO steps, ADMMBO may be able to facilitate derivation of closed-form acquisition functions.

Advantages	Infeasible initialization	Decoupled problems	Closed-form acquisition func.	Stopping criterion
EIC	–	–	✓	–
IECI	–	–	–	–
PESC	✓	✓	–	–
EVR	–	–	–	–
ALBO	✓	–	–	✓
Slack-AL	✓	–	✓	✓
ADMMBO	✓	✓	✓	✓

Table 1: Advantages of ADMMBO with respect to the state-of-the-art methods.

– ADMMBO offers a well-defined stopping criterion, inherited from ADMM, which in practice avoids unnecessary function evaluations. The stopping criterion is satisfied when the solutions of the optimality and feasibility subproblems converge to each other.

– Our experiments empirically show that ADMMBO achieves good solutions significantly faster than the state-of-the-art methods, is relatively insensitive to initialization, and requires fewer function evaluations to find desirable solutions. Moreover, our results suggest that ADMMBO's performance does not depend on whether the optimal solution lies on the boundary of or inside the feasible region, and is also insensitive to the relative volume of the feasibility region.

1.2. Paper Organization

In Section 2, we review both BO and the ADMM algorithm that we build upon. We motivate and introduce our proposed reformulation of the constrained problem and present our ADMMBO algorithm to solve this reformulated optimization in Section 3. In Section 4, we discuss existing related work on constrained BO that handles unknown-objective unknown-constraint optimization problems. We present experimental results on synthetic and real data in Section 5. Finally, in Section 6, we discuss our results and open avenues for future research and conclude the paper.

2. Background

In this section, we review the underlying components of our proposed method: Bayesian Optimization in its standard settings, with a focus on EI as the acquisition function, and the ADMM algorithm.

2.1. Bayesian Optimization

Bayesian optimization (Shahriari et al., 2016; Brochu et al., 2010b) addresses the problem of finding a global minimum (or maximum), x^*, of an objective function $f(x)$ over a bounded box $\mathcal{B} \in \mathbb{R}^d$, where f is unknown but available to evaluate pointwise via computationally costly queries. Thus, the goal is to find x^* with as few evaluations of $f(x)$ as possible. Given a collection of initial points in $\mathcal{B}$ and their observed objective values, denoted by $\mathcal{F} = \{(x_l, f(x_l))\}_{l=1}^n$, BO methods iteratively suggest the next best candidate x_{l+1}, and

evaluate the corresponding objective value at iteration $l+1$. More specifically, to find x_{l+1}, BO first assumes a prior probability model for the unknown function f, denoted by $p\big[f(x)\big]$, and then uses the observed data, $\mathcal{F}$, to update the posterior probability $p\big[f(x)|\mathcal{F}\big]$. This posterior is then used to build an *acquisition function*, denoted by $\alpha(x)$, which provides an estimate of the *optimization usefulness* of any candidate point $x \in \mathcal{B}$ if it is chosen as x_{l+1} for the next function evaluation. In contrast to $f(x)$, the acquisition function $\alpha(x)$ has a known form and can be maximized over $x \in \mathcal{B}$, using analytical or numerical optimization techniques. The optimum of the acquisition function provides a recommendation for x_{l+1} that is used to evaluate $f(x_{l+1})$ and then to update the data $\mathcal{F}$ accordingly. BO iteratively repeats this process, guiding the search towards sampling a global minimizer of f.

Many BO methods assume that the unknown function $f(x)$ is a Lipschitz continuous bounded function over $\mathcal{B}$, and then model $p\big[f(x)\big]$ as a Gaussian Process (GP) (Močkus, 1975; Jones et al., 1998; Hernández-Lobato et al., 2014; Kushner, 1964; Cox and John, 1992). GPs are non-parametric Bayesian models which are widely used in the Bayesian optimization literature since they provide a flexible fit for modeling unknown functions. Moreover given GP models, some acquisition functions give closed-form expressions, which can be efficiently optimized with numerical optimization techniques (Rasmussen and Williams, 2006; Houlsby et al., 2012). As an example, which we will then employ in the exposition of our method below, we describe a popular acquisition function called Expected Improvement (Jones et al., 1998; Brochu et al., 2010b).

Given data $\mathcal{F}$, let f^+ denote the best objective value achieved by the points in $\mathcal{F}$. Then the improvement of any given point x, denoted by $I(x)$, corresponds to the improvement of $f(x)$ relative to f^+, i.e., $I(x) \triangleq \max(0, f^+ - f(x))$. An efficient strategy in BO would be to choose the next candidate for function evaluation by finding a point x which offers the largest improvement (Jones et al., 1998). However, since $f(x)$ is unknown and expensive to evaluate pointwise, it is difficult to calculate improvement $I(x)$. Alternatively, Expected Improvement of x, denoted by $EI(x)$, is an acquisition function which computes the expectation of $I(x)$ with respect to $p\big[f(x)|\mathcal{F}\big]$. Močkus (1975); Jones et al. (1998); Brochu et al. (2010b) has shown that assuming a GP model for $p\big[f(x)\big]$, the Expected Improvement can be computed using the closed-form expression

$$EI(x) = E_{f|\mathcal{F}}\big[I(x)\big] = \sigma_f(x)\left[\frac{m_f(x)-f^+}{\sigma_f(x)}\,\Phi\big(\frac{m_f(x)-f^+}{\sigma_f(x)}\big) + \phi\big(\frac{m_f(x)-f^+}{\sigma_f(x)}\big)\right], \tag{1}$$

where the expectation is computed with respect to the posterior probability $p\,[f(x)|\mathcal{F}]$. Here, $\Phi(\cdot)$ denotes the normal cumulative distribution function, $\phi(\cdot)$ is the standard normal probability density function, and $m_f(x)$ and $\sigma_f(x)$ are the posterior mean and standard deviation of $p\big[f(x)|F\big]$, respectively. All four of these components can be cheaply and easily evaluated for any given x. Thus, one advantage of $EI(x)$ over $I(x)$ is that $EI(x)$ can be cheaply evaluated pointwise without requiring evaluation of $f(x)$. Moreover, according to (1), $EI(x)$ can be efficiently maximized using numerical optimization techniques to suggest the most promising point (from the perspective of EI) for function evaluation.

2.2. Alternating Direction Method of Multipliers (ADMM) Optimization

Our reformulation of the optimization of an unknown-objective unknown-constraint problem allows us to build a framework based on a popular numerical optimization technique, ADMM (Boyd et al., 2011; Hong and Luo, 2017; Parikh et al., 2014), which we briefly review here. Consider the problem of minimizing $f(x) + g(x)$ with respect to x, where $x \in \mathbb{R}^d$ and $f, g : \mathbb{R}^d \to \mathbb{R}$. Specifically, consider the case where separately minimizing $f(x)$ and $g(x)$ is relatively easy, while optimizing their sum is challenging. For example in the Least Absolute Shrinkage and Selection Operator (LASSO) problem (Tibshirani, 1996; Mota et al., 2013), we are interested in minimizing $\|Ax - b\|_2^2 + \lambda\|x\|_1$ with respect to x, with an overdetermined dictionary, A. While each term can be easily minimized, it is much harder to minimize the sum of the two terms. ADMM is a powerful numerical optimization method which handles such cases (Boyd et al., 2011; Hong and Luo, 2017).

In order to minimize $f(x) + g(x)$, ADMM first defines an auxiliary variable $z \in \mathbb{R}^d$ for the function g, and considers the following optimization, which is equivalent to the original minimization problem,

$$\min_{x,z} \;\; f(x) + g(z) \quad \text{s.t.} \quad x = z. \tag{2}$$

To solve (2), ADMM first builds the augmented Lagrangian function (ALF) for (2), where the ALF provides an unconstrained surrogate function for the constrained problem. Specifically, ALF augments the objective function of a constrained problem with terms penalizing the infeasibility of the constraints. These penalty terms include the product of the feasibility gap with a dual variable vector, also called a Lagrange multiplier vector, and the squared Euclidean norm of the feasibility gap. More specifically, ALF for (2) is given by

$$L_\rho(x, z, y) \triangleq f(x) + g(z) + y^T(x - z) + \frac{\rho}{2}\big\|x - z\big\|_2^2, \tag{3}$$

where $y \in \mathbb{R}^d$ denotes the Lagrange multiplier vector corresponding to the constraint, $x - z$ is the feasibility gap, and ρ is a positive penalty parameter.

Starting from an initial value for y, z, ADMM iteratively updates the values of variables x, y, z by minimizing the ALF, until convergence. Let x^k, z^k, y^k denote the values of variables at iteration k. At iteration $k+1$, ADMM solves two optimization problems, one over x while fixing $z = z^k$ and $y = y^k$ and one over z while fixing $x = x^{k+1}$ and $y = y^k$, and updates the Lagrange multiplier vector afterwards. More specifically, at iteration $k+1$, ADMM solves

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x \; L_\rho(x, z^k, y^k) = \operatorname*{argmin}_x \; f(x) + (y^k)^T(x - z^k) + \frac{\rho}{2}\big\|x - z^k\big\|_2^2, \\ z^{k+1} &= \operatorname*{argmin}_z \; L_\rho(x^{k+1}, z, y^k) = \operatorname*{argmin}_z \; g(z) + (y^k)^T(x^{k+1} - z) + \frac{\rho}{2}\big\|x^{k+1} - z\big\|_2^2, \\ y^{k+1} &= y^k + \rho\,(x^{k+1} - z^{k+1}). \end{aligned} \tag{4}$$

The primal residual is defined as $r^{k+1} \triangleq x^{k+1} - z^{k+1}$, i.e., the gap between the main variable x and the auxiliary variable z, and the dual residual can be shown to be $s^{k+1} \triangleq -\rho(z^{k+1} - z^k)$ (Boyd et al., 2011; Hong and Luo, 2017). Assuming f and g are closed, proper and convex, and also that the unaugmented Lagrangian $L_\rho(x, z, y) - \frac{\rho}{2}\big\|x - z\big\|_2^2$ has a saddle point, Boyd et al. (2011) proves that as $k \to \infty$, $r^k \to 0$, $f(x^k) + g(z^k) \to p^*$, and

$y^k \to y^*$ where p^* is the optimal objective value of primal problem (2) and y^* is the dual optimal point. The necessary and sufficient optimality conditions for the ADMM problem are primal feasibility and dual feasibility, and they are effecively met in practice when the ℓ_2-norm of both the primal and dual residuals of (2) fall below an appropriately small tolerance.

Boyd et al. (2011) shows that ADMM can be extended to problems optimizing sum of more than two functions. In this situation, ADMM defines a distinct auxiliary variable z_i for each additional function $g_i,\ i = 1, \ldots, N$, and enforces each such variable to be equal to the main variable x. The rest of the algorithm naturally follows. See (Boyd et al., 2011), chapter 7 for a detailed discussion.

3. Constrained Bayesian Optimization via ADMMBO

In this section, we describe our proposed framework, which we refer to as ADMMBO, for solving the Bayesian optimization problem under unknown constraints. More specifically, we consider the constrained optimization problem of

$$\begin{aligned} \min_{x \in \mathcal{B}} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) \leq 0,\ i = 1, \ldots, N, \end{aligned} \tag{5}$$

where, $\mathcal{B} \subset \mathbb{R}^d$ is a bounded domain and $f, c_i : \mathbb{R}^d \to \mathbb{R}$ are unknown functions which can be evaluated pointwise. However, such evaluations are expensive. Our goal is to determine a sampling procedure for x that sequentially approaches a global optimum, x^*, with as few function queries from f and all c_i's as possible.

To tackle the problem, we first reformulate (5) into the unconstrained optimization

$$\min_{x \in \mathcal{B}} \quad f(x) + \sum_{i=1}^{N} M\, 1(c_i(x) > 0), \tag{6}$$

where $1(\cdot)$ is an indicator function, which is one when its argument is true and is zero otherwise, and M is a positive constant (Boyd et al., 2011). For a sufficiently large M, the constrained problem in (5) will be equivalent to the unconstrained one in (6).

Proposition 1. Given Lipschitz continuity of f and compactness of $\mathcal{B}$, f is bounded for every x in $\mathcal{B}$. Let η_ℓ and η_u denote, respectively, the lower and upper bound of f, i.e., $\eta_\ell \leq f(x) \leq \eta_u,\ \forall x \in \mathcal{B}$. Assume the feasible region of (5) is non-empty. For $M > \eta_u - \eta_\ell$, the unconstrained optimization in (6) will be equivalent to (5).

Proof. Let $J(x)$ denote the value of the objective function of (6). For any infeasible point of (5) $x_i \in \mathcal{B}$, we have $J(x_i) \geq \eta_\ell + M$, since the minimum value that f can attain is η_ℓ and the second term in (6) will be at least M, as x_i is infeasible for at least one constraint. On the other hand, for any feasible point $x_f \in \mathcal{B}$ of (5), we have $J(x_f) \leq \eta_u$. Since $M > \eta_u - \eta_\ell$, we always have $J(x_f) < J(x_i)$, hence (6) always finds a feasible solution, which makes the second term of the objective function vanish. As a result, the minimization in (6), obtains the minimizer of f, which satisfies all the constraints, hence being equivalent to (5).

A key observation in our proposed framework is that while jointly minimizing the objective function in (6) is difficult, individually minimizing each term of the objective function using Bayesian optimization allows independent function evaluations for f and each c_i. More specifically, we can minimize $f(x)$ with respect to x by assuming a GP model for $p\big[f(x)\big]$ and using BO afterwards. Similarly, we can minimize $1(c_i(x) > 0)$ with respect to x by assuming a GP model for $p\big[c_i(x)\big]$ and using it to build a Bernoulli random variable with parameter $\theta_i \triangleq p\big[c_i(x) > 0\big]$ to represent $1(c_i(x) > 0)$, and then applying BO. In contrast, optimizing the entire objective function in (6) is difficult and also may require joint function evaluations for f and every c_i. To take advantage of the simplicity of individually optimizing each term in the objective function of (6), we introduce N auxiliary variables, one per constraint function, and consider the following optimization problem

$$\begin{aligned} \min_{x, z_1, \ldots, z_N \in \mathcal{B}} \quad & f(x) + \sum_{i=1}^{N} M\, 1(c_i(z_i) > 0) \\ \text{s.t.} \quad & x = z_i,\ i = 1, \ldots, N. \end{aligned} \tag{7}$$

which clearly is equivalent to (6). Notice that in contrast to the unknown-objective unknown-constraint problem in (5), in (7) the equality constraints are known (deterministic) and only the objective function is unknown. Moreover, each of the unknown terms in the objective function of (7) is defined over a different variable, leading to a variable separation property which we will take advantage of. Next, we describe how ADMMBO combines Bayesian optimization with an ADMM-inspired framework to solve (7) efficiently.

3.1. ADMMBO Formulation

In this section, we describe our approach to combine the ADMM algorithm with BO steps to solve the proposed equivalent reformulation in (7). We first need to build the ALF for the optimization in (7), which is given by

$$\begin{aligned} L_\rho(x, z_i, y_i) &= f(x) + \sum_{i=1}^{N} \Big[M \ \ 1(c_i(z_i) > 0) + y_i^T (x - z_i) + \frac{\rho}{2} \big\| x - z_i \big\|_2^2 \Big] \\ &= f(x) + \sum_{i=1}^{N} \Big[M \ \ 1(c_i(z_i) > 0) + \frac{\rho}{2} \big\| x - z_i + \frac{y_i}{\rho} \big\|_2^2 - \frac{\rho}{2} \big\| y_i \big\|_2^2 \Big], \end{aligned} \tag{8}$$

where $y_i \in \mathbb{R}^d$ is a Lagrange multiplier vector, and ρ is a positive penalty parameter. Note that the second line in (8) follows from the first line by completing the square and reordering terms. Having formed the ALF, to apply the ADMM algorithm, as discussed in Section 2.2, we initialize the unknown variables and the Lagrange multiplier vectors and at ADMM iteration k, we solve for each variable in turn, using BO, having fixed the rest.

More specifically, for (8), the kth ADMM iteration will become

$$\begin{aligned} x^{k+1} &= \underset{x\in\mathcal{B}}{\operatorname{argmin}} \;\; f(x) + \sum_{i=1}^{N} \frac{\rho}{2}\|x - z_i^k + \frac{y_i^k}{\rho}\|_2^2, \\ z_i^{k+1} &= \underset{z_i\in\mathcal{B}}{\operatorname{argmin}} \;\; M\,1(c_i(z_i) > 0) + \frac{\rho}{2}\|x^{k+1} - z_i + \frac{y_i^k}{\rho}\|_2^2, \;\forall i = 1,\dots,N \\ y_i^{k+1} &= y_i^k + \rho\,(x^{k+1} - z_i^{k+1}), \;\forall i = 1,\dots,N. \end{aligned} \tag{9}$$

The x update, which we refer to as the *optimality subproblem*, minimizes the unconstrained objective function of the original problem in (5), f, plus a sum of quadratic terms that force the solution to be close to the feasible region. On the other hand, each z_i update, which we refer to as *feasibility subproblems*, looks for a feasible point of the constraint c_i that is also close to the unconstrained optimum found in the optimality subproblem.

Since both the optimality and feasibility subproblems involve unknown objectives, we solve each of them using Bayesian optimization with *unconstrained* acquisition functions. Thus, in ADMMBO there are two levels of iteration: ADMM iterations (from now on referred to as main loop iterations), and BO iterations, which are performed to solve each subproblem during each main loop iteration. ADMMBO's general framework allows it to incorporate *any* unconstrained acquisition function, including EI, Predictive Entropy Search (PES)(Hernández-Lobato et al., 2014), and Knowledge Gradient (KG)(Wu et al., 2017), as best fits a given problem. For example, while PES is reported to outperform EI by Hernández-Lobato et al. (2014), but has also been reported to be relatively slow due to its need to sample x^* and compute expectation propagation approximations (Hernández-Lobato et al., 2016). EI has a closed-form solution which, in practice, may make it faster than PES (Jones et al., 1998). The choice of acquisition function for each subproblem in any main loop iteration of ADMMBO is a matter of user preference and does not change ADMMBO's structure. In this paper we chose to use EI to solve both the optimality and feasibility subproblems because of its wide popularity and because its structure more easily leads to closed form solutions. In addition, while we could have modeled the objective function of each subproblem with a single GP, this would have ignored available partial knowledge about the structure of these objectives. Instead, we designed a specific Bayesian model for each subproblem objective that takes advantage of this knowledge to better guide the optimization. We show that EI still maintains a closed-form solution given these new Bayesian models.

3.1.1. **Expected Improvement for the Optimality Subproblem**

For the kth main loop iteration, the optimality subproblem associated with (9) requires optimizing the sum of the unknown objective function, f, and a known function, i.e.,

$$\min_{x\in\mathcal{B}} \;\; u^k(x), \quad \text{where} \;\; u^k(x) \triangleq f(x) + \sum_{i=1}^{N} \frac{\rho}{2}\|x - z_i^k + \frac{y_i^k}{\rho}\|_2^2. \tag{10}$$

As $f(x)$ is an unknown function, we solve (10) via BO by assuming that f follows a GP prior. Since z_i^k and y_i^k are given and fixed, the second term in the definition of $u^k(x)$ is

constant for any given x. Thus, we can still model $p\big[u^k(x)\big]$ as a GP. Given observed data $\mathcal{F} = \{\big(x_l, f(x_l)\big)\}_{l=1}^n$, $x_l \in \mathcal{B}$, we compute $\mathcal{U}^k = \{\big(x_l, u^k(x_l)\big)\}_{l=1}^n$ and denote the best objective value of (10) so far by u^{k+}. Then, similar to the standard EI, we compute the Expected Improvement for the optimality subproblem, which will be

$$\begin{aligned} EI(x) &= E_{u^k|\mathcal{U}^k}\big[\max\big(0, u^{k+} - u^k(x)\big)\big] \\ &= \sigma_{u^k}(x)\left[\frac{m_{u^k}(x) - u^{k+}}{\sigma_{u^k}(x)}\,\Phi\big(\frac{m_{u^k}(x) - u^{k+}}{\sigma_{u^k}(x)}\big) + \phi\big(\frac{m_{u^k}(x) - u^{k+}}{\sigma_{u^k}(x)}\big)\right], \end{aligned} \tag{11}$$

where $m_{u^k}(x), \sigma_{u^k}(x)$ are, respectively, the mean and standard deviation of the posterior distribution $p\big[u^k(x)|\mathcal{U}^k\big]$. Thus, for any given x, we can calculate its EI via (11).

3.1.2. **Expected Improvement for the Feasibility Subproblem**

For k^{th} main loop iteration, the ith feasibility subproblem associated with (9) requires optimizing the sum of an unknown function and a known function, i.e,

$$\min_{z_i\in\mathcal{B}} \;\; h_i^k(z_i), \quad \text{where} \;\; h_i^k(z_i) \triangleq 1(c_i(z_i) > 0) + \frac{\rho}{2M}\|x^{k+1} - z_i + \frac{y_i^k}{\rho}\|_2^2. \tag{12}$$

Let us call $q_i^k(z_i) = \frac{\rho}{2M}\|x^{k+1} - z_i + \frac{y_i^k}{\rho}\|_2^2$. Since $c_i(z_i)$ is unknown, we solve (12) via BO by assuming that c_i follows a GP prior. Then, we model $1(c_i(z_i) > 0)$ as a Bernoulli random variable with the parameter $\theta_i \triangleq p\big[c_i(z_i) > 0\big]$. Since x_i^{k+1} and y_i^k are given and fixed, $q_i^k(z_i)$ will be constant for any given z_i. Thus, we model $h_i^k(z_i)$ as a shifted Bernoulli random variable, again with the parameter θ_i, which is equal to $q_i^k(z_i) + 1$ with probability θ_i, and equal to $q_i^k(z_i)$ with probability $1 - \theta_i$. Note that $1 - \theta_i$ for any z_i is a Gaussian Cumulative Distribution Function (CDF) based on the marginal Gaussianity of GPs (Houlsby et al., 2012; Gardner et al., 2014; Rasmussen and Williams, 2006). Given $\mathcal{C}_i = \{\big(z_{l,i}, c_i(z_{l,i})\big)\}_{l=1}^{m_i}, z_{l,i} \in \mathcal{B}$, we generate $\mathcal{H}_i^k = \{\big(z_{l,i}, h_i^k(z_{l,i})\big)\}_{l=1}^{m_i}$ using $\mathcal{C}_i$, and denote the best objective value of (12) by h_i^{k+}. We then compute the Expected Improvement for the ith feasibility subproblem, which is given by

$$\begin{aligned} EI(z_i) = E_{h_i^k|\mathcal{H}_i^k}\big[\max\big(0, h_i^{k+} - h_i^k(z_i)\big)\big] &= \max\big(0, h_i^{k+} - q_i^k(z_i) - 1\big)\theta_i \\ &\quad + \max\big(0, h_i^{k+} - q_i^k(z_i)\big)\big(1 - \theta_i\big), \end{aligned} \tag{13}$$

Given any z_i, if $h_i^{k+} - q_i^k(z_i)$ is non-positive, then $EI(z_i)$ is zero. If $h_i^{k+} - q_i^k(z_i)$ lies between zero and one, the first term in (13) is zero while the second term has a positive value. When $h_i^{k+} - q_i^k(z_i)$ is larger than one, both terms are positive. Thus, we can simplify (13) to

$$EI(z_i) = \begin{cases} 0, & \text{if} \quad h_i^{k+} - q_i^k(z_i) \le 0 \\ \max\big(0, h_i^{k+} - q_i^k(z_i)\big)\big(1 - \theta_i\big), & \text{if} \quad 0 < h_i^{k+} - q_i^k(z_i) \le 1 \\ \max\big(0, h_i^{k+} - q_i^k(z_i)\big)\big(1 - \theta_i\big) + \max\big(0, h_i^{k+} - q_i^k(z_i) - 1\big)\theta_i, & \text{else.} \end{cases} \tag{14}$$

Notice that according to the closed-form expressions in (11) and (14) , EIs can be cheaply evaluated pointwise at any given point. As a result, these acquisition functions can be maximized with standard numerical optimization methods such as DIRECT or stochastic gradient descent (Finkel, 2003; Bottou, 2010)

Algorithm 3.1 ADMMBO

1: **Input**: $\mathcal{B}, n, m_i, \delta, K, \alpha^k, \beta_i^k, \rho, \epsilon, M, y_i^1, z_i^1;\ \forall i = 1, \ldots, N,\ \forall k = 1, \ldots, K$
2: Randomly generate $\{x_l \in \mathcal{B}\}_{l=1}^{n}$ and $\{z_{l,i} \in \mathcal{B}\}_{l=1}^{m_i},\ \forall i = 1, \ldots, N$
3: Initialize: $k{=}1, \mathcal{F}^1{=}\{(x_l, f(x_l))\}_{l=1}^{n}, \mathcal{C}_i^1{=}\{(z_{l,i}, c_i(z_{l,i}))\}_{l=1}^{m_i}$, S = False;
4: **while** $(k \leq K)$ and $(S == False)$ **do**
5: $\quad [x^{k+1}, \mathcal{F}^{k+1}] \leftarrow \mathrm{OPT}(\mathcal{F}^k, \mathcal{B},\ \alpha^k,\ z_i^k,\ y_i^k)$ (See Algorithm 3.2)
6: $\quad$ **for** $i = 1, \ldots, N$ **do**
7: $\quad\quad [z_i^{k+1}, \mathcal{C}_i^{k+1}] \leftarrow \mathrm{FEAS}(\mathcal{C}_i^k, \mathcal{B},\ \beta_i^k,\ x^{k+1},\ y_i^k)$ (See Algorithm 3.3)
8: $\quad\quad y_i^{k+1} = y_i^k + \rho(x^{k+1} - z_i^{k+1})$
9: $\quad\quad r^{k+1}[i] = x^{k+1} - z_i^{k+1}$
10: $\quad\quad s^{k+1}[i] = -\rho(z_i^{k+1} - z_i^k)$
11: $\quad$ **end for**
12: $\quad$ **if** $\left(\left\|r^{k+1}\right\|_2 \leq \epsilon\right)$ and $\left(\left\|s^{k+1}\right\|_2 \leq \epsilon\right)$ **then**
$\quad\quad$ S = True
13: $\quad$ **end if**
14: $\quad k \leftarrow k + 1$
15: **end while**
16: **if** S==True **then**
$\quad$ **Output:** x^{k+1}
17: **elseOutput:** $\underset{x \in \mathcal{F}^K \cup \mathcal{C}_1^K \cup \cdots \cup \mathcal{C}_N^K}{\operatorname{argmin}} \quad E_{f|\mathcal{F}^K}\left[f(x)\right] \quad \text{s.t.} \quad p\left[c_i(x) \leq 0\right] \geq 1 - \delta$
18: **end if**

3.2. ADMMBO Algorithm

Algorithm 3.1 summarizes the steps of ADMMBO. The parameters to the algorithm are the search space $\mathcal{B}$, the coefficient M, the number of initial function evaluations for the objective function n, number of initial function evaluations for each constraint m_i for $i = 1, \ldots, N$, the maximum number of ADMM iterations K, the ADMM's penalty parameter ρ, and the total BO iteration budget, the maximum number of function evaluations throughout the algorithm. We distribute this budget among main loop where at iteration k, α^k denotes the BO budget for the optimality subproblem and β_i^k is the BO budget for i^{th} feasibility subproblem, the tolerances for the stopping criterion ϵ, and a confidence parameter δ to determine the final solution returned in the case that the budget is exhausted before convergence.

Algorithm 3.1 works as follows: first in order to build the initial datasets $\mathcal{F}$ and $\mathcal{C}_i$, the algorithm randomly generate n and m_i samples in the search space $\mathcal{B}$, and then evaluate f and c_i at the corresponding points (lines $2-3$). After initializing the parameters (line 3), ADMMBO iterates through its main loop until it reaches the total BO iteration budget or the stopping criterion is satisfied (line 4). At the kth iteration of the main loop, ADMMBO solves the optimality subproblem with BO given budget α^k. Specifically, this step is done by calling the algorithm 3.2, denoted OPT, which outputs a desirable solution of the optimality subproblem and the updated dataset $\mathcal{F}$ (line 5). Then, for each constraint i, ADMMBO performs the following steps: first it solves the corresponding feasibility subproblem with BO given budget α_i^k, by calling the algorithm 3.3, denoted FEAS, which

Algorithm 3.2 OPT

1: **Input:** $\mathcal{F} = \{\big(x_l, f(x_l)\big)\}_{l=1}^{n}, \mathcal{B}, \alpha, z_i, y_i;\quad i = 1, \ldots, N$
2: Initialize: $\mathcal{F}^1 = \mathcal{F}$
3: **for** $t = 1, \ldots, \alpha$ **do**
4: $\quad$ Given $\mathcal{F}^t$, compute $\mathcal{U}^t = \{(x_l, f(x_l) + \sum_{i=1}^{N} \frac{\rho}{2}\left\|x_l - z_i + \frac{y_i}{\rho}\right\|_2^2\}_{l=1}^{n}$
5: $\quad$ Update the GP posterior $p[u(x)|\ \mathcal{U}^t]$
6: $\quad x^t \leftarrow \arg\max_{x \in \mathcal{B}} EI(x)$ (use expression (11) for $EI(x)$)
7: $\quad \mathcal{F}^{t+1} = \mathcal{F}^t \cup \{\big(x^t, f(x^t)\big)\}$
8: $\quad n \leftarrow n + 1$
9: **end for**
10: $x_{min} = \underset{x \in \mathcal{F}^\alpha}{\operatorname{argmin}} \quad f(x) + \sum_{i=1}^{N} \frac{\rho}{2}\left\|x - z_i + \frac{y_i}{\rho}\right\|_2^2$
11: **Output:** $[x_{min},\ F^\alpha]$

outputs a good solution of the ith feasibility subproblem and the updated dataset $\mathcal{C}_i$ (line 7). Then, ADMMBO updates the corresponding Lagrange multipliers and components of the primal and dual residuals (lines $8-10$). Afterwards, at the end of each main loop iteration, it checks the stopping criterion, i.e. whether the ℓ_2-norms of the primal and dual residuals are smaller than or equal to a chosen tolerance (line 12). If the stopping criterion is satisfied, the algorithm stops and reports the most recent x as the desirable solution for the unknown-objective unknown-constraint problem (5) (line 17). Otherwise, it keeps iterating. After reaching the maximum number of total iterations without meeting stopping criterion, ADMMBO reports a final recommendation for the desirable solution of (5). This recommendation is the point belonging to the merged data $\mathcal{F} \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_N$ which has the lowest expected objective value subject to the posterior probability of satisfying the constraints being at least $1 - \delta$, where δ is a parameter representing the maximum acceptance probability that a final solution is infeasible.

Algorithm 3.2, denoted by OPT, solves the optimality subproblem with BO under a budget α. For α iterations, OPT repeats the following steps: Given y_i, z_i, and dataset $\mathcal{F}$, it computes $\mathcal{U}$ and updates the GP posterior $p[u(x)|\mathcal{U}]$. Then, OPT uses this posterior to compute $EI(x)$ using equation (11) and maximizes it over $x \in \mathcal{B}$. It evaluates the objective function f at the global optimum of $EI(x)$, and updates data $\mathcal{F}$ accordingly. After α iterations, OPT gives a final recommendation for the solution of the optimality subproblem, and outputs the most updated data $\mathcal{F}$.

Algorithm 3.3, denoted by FEAS, solves each feasibility subproblem with BO under a budget β_i. For β_i iterations, FEAS repeats the following steps: Given x, y_i, and data $\mathcal{C}_i$, it computes $\mathcal{H}_i$. Then, FEAS updates the GP posterior $p[c_i(z_i)|\mathcal{C}_i]$ and use this posterior and $\mathcal{H}_i$ to compute $EI(z_i)$ using equation (13). Next, it maximizes $EI(z_i)$ over $z_i \in \mathcal{B}$, evaluate the constraint c_i at the optimum of the $EI(z_i)$ and updates $\mathcal{C}_i$ accordingly. After β_i iterations, the algorithm gives a final recommendation of the solution for the feasibility subproblem, and outputs the most updated data $\mathcal{C}_i$.

ADMMBO has the potential to be parallelized at two different levels: first, the feasibility subproblems are independent and can be solved independently, in parallel (lines $6-11$ in Algorithm 3.1). Second, BO iterations at OPT and FEAS (lines $3-9$ in Algorithms 3.2 and

Algorithm 3.3 FEAS

Input: $\mathcal{C}_i = \{(z_{l,i}, c_i(z_{l,i}))\}_{l=1}^{m_i}, \mathcal{B}, \beta_i, x, y_i$;
Initialize: $\mathcal{C}_i^1 = \mathcal{C}_i$
for $t = 1, \ldots, \beta_i$ **do**
  Given $\mathcal{C}_i^t$, compute $\mathcal{H}_i^t = \{(z_{l,i}, 1(c_i(z_{l,i}) > 0) + \frac{\rho}{2M}\|x - z_{l,i} + \frac{y_i}{\rho}\|_2^2)\}_{l=1}^{m_i}$
  Update the GP posterior $p[c_i(z_i)|\ \mathcal{C}_i^t]$
  $z_i^t \leftarrow \arg\max_{z_i \in \mathcal{B}} EI(z_i)$ (use expression (13) for $EI(z_i)$)
  $\mathcal{C}_i^{t+1} = \mathcal{C}_i^t \cup \{(z_i^t, c_i(z_i^t))\}$
  $m_i \leftarrow m_i + 1$
end for
$z_{min} = \underset{z_i \in \mathcal{C}_i^{\beta_i}}{\operatorname{argmin}} \quad 1(c_i(z_i) > 0) + \frac{\rho}{2M}\|x - z_i + \frac{y_i}{\rho}\|_2^2$
Output: $[z_{min},\ \mathcal{C}_i^{\beta_i}]$

3.3) can be parallelized according to a scheme suggested by Snoek et al. (2012) where a new candidate location is selected according to not only the observed data, but also the locations of pending function evaluations. Both parallelization will lead to speed up of ADMMBO.

3.2.1. Hyperparameter Tuning for ADMMBO

ADMMBO has two sets of parameters: BO-dependent parameters, which are commonly used by other constrained BO methods, and ADMM-dependent parameters, which lend themselves to the ADMM framework of ADMMBO. BO-dependent parameters are $\mathcal{B}$, a box defining the search space, n and m_i, the number of initial random samples at which to evaluate f and each c_i, respectively, δ, the parameter used if ADMMBO does not converge, and a total BO iteration budget.

ADMM-dependent parameters are K, the maximum number of iterations in the main loop, along with α^k and β_i^k, the BO iteration budgets for the optimality and i^{th} feasibility subproblems during the kth main loop iteration. These three hyperparameters should be jointly set in a way that $\sum_{k=1}^{K}\left(\alpha^k + \sum_{i=1}^{N}\beta_i^k\right)$ equals to the total BO iteration budget. ρ is the ADMM penalty parameter, and ϵ is a small tolerance which controls the stopping rule for the algorithm. We provide guidance on how to set these hyperparameters in practice in section 5.1. Finally, M controls the penalization of infeasibility. In Proposition 1, we suggest to set the value of M based on the range of f over $\mathcal{B}$. The unconstrained range of f is known for many user-defined objectives. For example, if we define f as the validation error of a machine learning model, this range equals, the maximum possible error, 1, minus the minimum possible error 0, $1 - 0 = 1$. Since the unconstrained range of f is an upper bound for the range of f over $\mathcal{B}$, we can use it to set M. However, this range might be unknown in practice. In such cases, we recommend setting the value of M equal to a large number with respect to any likely value of f, which can be chosen based on the application domain. As long as M is reasonably large, its precise value does not affect the performance of ADMMBO in practice. In section 5.7, we illustrate that ADMMBO's performance is not sensitive to the choice of M over a very wide range of values.

3.2.2. Convergence in Practice

Convergence guarantees for ADMM only hold for convex problems (Boyd et al., 2011). However, here only limited information is available about the objective function and the feasible set and thus often the convexity of the problem is unknown. If f is a convex function and the feasible set is a convex set, ADMM has convergence guarantees given each subproblem is solved exactly. In ADMMBO, however, the subproblems have unknown objectives which the algorithm solves using BO methods. These methods offer exact solutions only given an unlimited budget, which is not realistic in practice. For a limited budget, BO methods find approximate solutions for the subproblems, and thus similar to the rest of the BO state-of-the-art, ADMMBO cannot offer convergence guarantees.

However, in fact, we have chosen ADMM precisely to build upon the many studies that have found that ADMM exhibits a good empirical performance even if the convergence conditions are not satisfied (Xu et al., 2016; Wang et al., 2015; Hong et al., 2016). We report in section 5 that ADMMBO converged for the non-convex problems we tested.

4. Related Work

Two general strategies have been introduced to extend Bayesian optimization to constrained Bayesian optimization with unknown constraints. One strategy is to modify the acquisition function within the Bayesian optimization framework, that the acquisition function simultaneously takes into account the feasibility of a candidate point along with its objective value. Most previous work falls into this category, including EIC, IECI, EVR, and PESC (Schonlau et al., 1998; Snoek, 2013; Gelbart et al., 2014; Gardner et al., 2014; Bernardo et al., 2011; Picheny, 2014; Hernández-Lobato et al., 2015).

The second strategy merges Bayesian optimization with numerical optimization techniques which are designed to deal with constrained optimization problems. To the best of our knowledge, to date there is only one such approach in this category for BO, ALBO, along with its Slack-AL variant, (Gramacy et al., 2016; Picheny et al., 2016). We describe some existing methods in both categories next.

4.1. Constrained BO using Modified Acquisition Functions

Several proposed acquisition functions for BO problems with unknown constraints are extensions of EI (Jones et al., 1998). One such extension, Expected Improvement with Constraints is reported by Schonlau et al. (1998); Snoek (2013); Gelbart et al. (2014), and Gardner et al. (2014). Given a point x, EIC calculates the expectation of the improvement of the objective value of x over the best observed objective value evaluated at a *feasible* point so far, and then weight it by the probability of feasibility of x. Assuming independent Gaussian process models for the objective function and constraints, EIC yields closed-form solutions based on the standard normal CDF and PDF. Bernardo et al. (2011) proposed another variation of EI, Integrated Expected Conditional Improvement, based on a one-step look-ahead strategy. Given a candidate point x, IECI measures how much reduction in the improvement of the objective value of a typical point x' is expected, if we have previously evaluated the objective value of x and augmented our data with it. IECI does not have a closed-form solution, and thus its integral is computed numerically over a grid of x'. This

limits the application of IECI to small dimensional problems (Hernández-Lobato et al., 2015; Shahriari et al., 2016).

In addition to EI-based methods, there is a class of information-based acquisition functions designed to reduce a chosen measure of uncertainty about the location of the global optimum. Thus, for a candidate point, such methods evaluate the reduction in their uncertainty measure which will be obtained by evaluating its objective value. Expected Volume Reduction proposed by Picheny (2014) uses the expected volume of the feasible region as its measure of uncertainty. For a point x, EVR first computes the probability that, for any given point x', $f(x')$ is less than the minimum of the best observed f so far corresponds to a feasible point and $f(x)$. It then integrates that probability against the probability of feasibility of x' over all x'. Another information-based acquisition function, Predictive Entropy Search with Constraints (PESC) uses entropy as its uncertainty measure. Specifically, PESC first calculates the differential entropy of the posterior of the global optimum and then for a point x, measures how much reduction is expected in this entropy if we evaluate the objective function and constraints at point x (Hernández-Lobato et al., 2015).

4.2. Constrained BO using Numerical Optimization

In addition to the approaches based on BO with a modified acquisition function, there is a second category that solves the unknown constraint problem using ideas from the field of numerical optimization. Many numerical optimization algorithms tackle a constrained problem by reformulating it into two or more coupled unconstrained problems, which are generally easier to handle, and then solving them via alternating iterations (Nocedal and Wright, 2006). Here, where the constrained problem involves unknown functions, the idea is to define unconstrained surrogate problems using numerical optimization techniques, and then solve these problems, which still involve unknown functions, with BO. The first, and to-date only, methods in this category are based on the augmented Lagrangian method.

Gramacy et al. (2016) proposed the Augmented Lagrangian for BO, ALBO, which uses the Augmented Lagrangian Function (ALF) to formulate unconstrained surrogate problems, and then solves them using EI as acquisition function. The challenge in the proposed approach is that ALF of the original problem involves a complicated mixture of unknown functions. Thus, the previous calculations for the EI, which assumed a single GP model, do not hold any more. Building a probabilistic model for this mixture objective and recalculating EI based on it is a challenging task. As a result, EI calculations in ALBO do not result in closed form solutions, and so this method relies on Monte-Carlo approximation. To address this issue, Picheny et al. (2016) introduced Slack-AL by modifying the original problem to include a slack variable and then applying the augmented Lagrangian method on the modified problem. The authors optimized the modified ALF with EI iterations. It turns out that the modified ALF in Slack-AL is easier to solve than the ALF in ALBO. As a result, the Expected Improvement in Slack-AL, in contrast to ALBO, has a closed-form expression, and may be evaluated with library routines.

5. Experiments

In this section, we evaluate the performance of ADMMBO on several synthetic problems, studied in the constrained Bayesian optimization literature (Gardner et al., 2014; Gramacy et al., 2016; Picheny et al., 2016), as well as on the problem of hyperparameter tuning for a fast neural network on the MNIST digit recognition dataset (LeCun, 1998; Hernández-Lobato et al., 2015). We compare ADMMBO with four state-of-the-art constrained Bayesian optimization methods[1]: EIC (Gelbart et al., 2014; Gardner et al., 2014), ALBO (Gramacy et al., 2016), Slack-AL (Picheny et al., 2016) and PESC (Hernández-Lobato et al., 2015).

5.1. Implementation Details

In all the synthetic problems, discussed below, similar to (Hernández-Lobato et al., 2015; Picheny et al., 2016; Gramacy et al., 2016), we assume that f and c_i follow independent GP priors with zero mean and squared exponential kernels. For the problem of hyperparameter tuning in Neural Networks on the MNIST dataset, we assume that f and c_i , follow independent GP priors with zero mean and with Matérn 5/2 kernels (Hernández-Lobato et al., 2015). For ADMMBO, in all the experiments we set $M \in \{20, 50\}$, $\rho = 0.1$, $\epsilon = 0.01$, $\delta = 0.05$ and initialize y_i^1 and z_i^1 with the bounds of $\mathcal{B}$. Further, in all the experiments, we set the total BO iteration budget to $100(N+1)$, where N is the number of constraints of the optimization. We empirically observed that ADMMBO performed best when we assign a higher BO budget for the first iteration of the algorithm. Thus, we set $\alpha^1 = \beta_i^1 \in \{10, 20, 50\}$ for the first iteration and $\alpha^k = \beta_i^k \in \{2, 5\}$ for the rest. Considering total BO budget and the budgets for the optimality and feasibility subproblems, we set K accordingly. We initialize datasets $\mathcal{F}$ and $\mathcal{C}_i$ with $n = m_i = 2$ points. Notice that initial points are randomly generated and will not necessarily be feasible.

The convergence speed of ADMM in practice depends on the value of the penalty parameter ρ (Boyd et al., 2011). Specifically, a large value of ρ imposes a large penalty on violating the primal feasibility and thus encourages small primal residuals. On the other hand, a small value of ρ increases the penalty on the dual residual, encouraging it to be small, but at the same time also reduces the penalty on primal feasibility, resulting in a larger primal residual. To improve the convergence speed of ADMMBO in practice and to make the performance less sensitive to the choice of the penalty parameter ρ, following (Boyd et al., 2011), we use the penalty ρ^k at iteration k, where

$$\rho^{k+1} = \begin{cases} \tau^{incr}\rho^k & \text{if } \left\|r^k\right\|_2 > \mu\left\|s^k\right\|_2 \\ \rho^k/\tau^{decr} & \text{if } \left\|s^k\right\|_2 > \mu\left\|r^k\right\|_2 \\ \rho^k & \text{otherwise.} \end{cases} \tag{15}$$

We set $\mu = 10$ and $\tau^{incr} = \tau^{decr} = 2$ similar to (Boyd et al., 2011; Hong and Luo, 2017). Please see our opensource code available at https://github.com/SetarehAr/ADMMBO for more details on each experiment.

5.2. Performance Metrics

To test the sensitivity of different algorithms to the initialization of $\{\mathcal{F}, \mathcal{C}_1, \ldots, \mathcal{C}_N\}$, we run each algorithm for each synthetic problem with 100 random initializations and for the hyperparameter tuning problem for 5 random initializations.

1. We used the open source codes provided by Baldi et al. (2015); Gardner et al. (2014); Gramacy et al. (2016); Picheny et al. (2016).

For each method after each additional function evaluation, we report the median of the best observed objective value at a feasible point, over all random initializations. This median is shown by a solid curve in our figures (Figures 1 to 6). For each method, we start to report results (show the median curve) once all 100 runs have found a feasible point. The budget at which each method attains such a point over 100 runs is denoted by a dashed vertical line in our figures. Moreover, the variability of the performance is illustrated after different number of function evaluations by reporting the 25/75 percentiles of the best feasible objective value over the 100 runs. Moreover, in Figures 1 to 4, we depict the feasible region of our 2-dimensional problems, their global and local optima, as well as the final recommendation provided by each method given a specific budget, among 100 runs.

5.3. Test Problem with a Small Feasible Region

Consider the following optimization problem, studied also in (Gardner et al., 2014),

$$\begin{aligned} \min_{x\in\mathcal{B}} \quad & \sin(x_1) + x_2 \\ \text{s.t.} \quad & \sin(x_1)\sin(x_2) + 0.95 \leq 0, \end{aligned} \tag{16}$$

where $\mathcal{B} = [0,6]^2$. This is a challenging problem since both the objective function and the constraint are highly non-linear. Moreover, the feasible region with respect to the bounded parameter space $\mathcal{B}$ is small, hence, finding a feasible point is difficult.

The left plot in Figure 1 shows the feasible region of (16) and its global and local optima, while the right figure shows the median of the objective value of the best feasible point, obtained by each method, among 100 runs as a function of total number of function evaluations. As the results demonstrate, ADMMBO outperforms EIC, ALBO, Slack-AL and PESC in terms of finding the global optimum at a lower budget. Moreover, ADMMBO is the first method to find a feasible point in all 100 runs, followed by PESC second and then the others. Only ADMMBO, ALBO, and Slack-AL have defined stopping criteria, and of those three only ADMBBO reaches its criterion and stops before the pre-set budget is exhausted. Figure 2 shows the best points obtained by ADMMBO, ALBO, and Slack-AL after 100 function evaluations, over 100 runs. Over all runs, ADMMBO has consistently found a feasible solution very close to the global optimum (black star in the left figure in Figure 1). However, the best points obtained by ALBO and Slack-AL are scattered throughout the entire feasible region and not necessarily close to the global optimum. Note that in a few runs, the best solutions found by these two methods are outside the feasible region, and thus are infeasible. We observe that ALBO and Slack-AL require a higher budget in order to converge to the global optimum of (16).

5.4. Test Problem with Multiple Constraints

Our second benchmark problem is a toy two dimensional problem introduced first in (Gramacy et al., 2016), and used later in (Picheny et al., 2016; Hernández-Lobato et al., 2015). We have chosen this problem specifically to test our ADMMBO on a problem with multiple

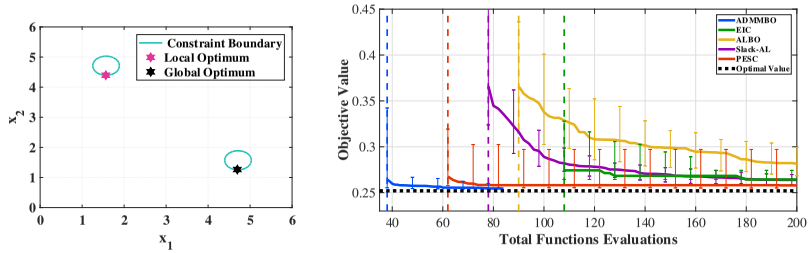


Figure 1: Left: feasible region of (16) consists of two oval regions. The pink and black stars show, respectively, the local and global optimizer. Right: the curve of the median and 25/75 percentiles of the best objective value found by each method, among 100 runs that obtain a feasible solution, as a function of the total budget for function evaluation. We report the results of each method for a budget once all of its 100 runs obtain a feasible solution.

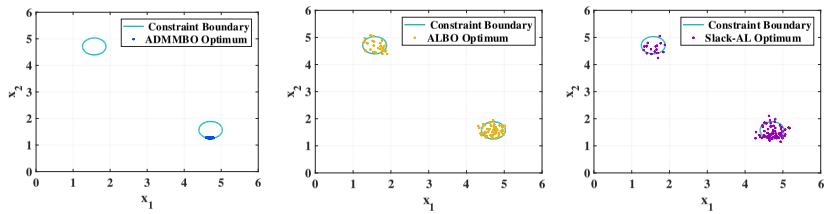


Figure 2: Feasible region of (16) and the best solutions obtained by ADMMBO (left), ALBO (middle) and Slack-AL (right) after 100 function evaluations, over 100 runs. For each method, each point represents the final solution of that method over one run, after 100 total function evaluations.

constraints. More specifically, for $\mathcal{B} = [0,1]^2$, we consider the optimization problem

$$\begin{aligned} \min_{x\in\mathcal{B}} \quad & x_1 + x_2 \\ \text{s.t.} \quad & 0.5\ \sin(2\pi(x_1^2 - 2x_2)) + x_1 + 2x_2 + 1.5 \leq 0, \quad -x_1^2 - x_2^2 + 1.5 \leq 0. \end{aligned} \tag{17}$$

The left plot in Figure 3 shows the feasible region of (17) and its global and local optimizers, and the right plot shows the performance of different methods as a function of the number of function evaluation budget. The layout is the same as for the previous figure. Again, ADMMBO achieves the best performance in terms of converging to the global optimum at a lower budget, followed by PESC. We believe this is due to the fact that both ADMMBO and PESC can handle decoupled problems, including this example, via single function evaluations, while EIC, ALBO and Slack-AL enforce joint function evaluations at each step. Moreover, as the plot demonstrates, ADMMBO and ALBO are the first methods that arrive at a feasible point over 100 runs at a lower number of function evaluations.

Figure 4 shows the best points obtained by ADMMBO, EIC, and PESC after 50 function evaluations, over 100 runs. Over all runs, ADMMBO and PESC have found solutions very close to the global or local optima (black and pink stars in the left figure in Figure 3). However, EIC was unable to narrow down its search toward such an optimum, and thus

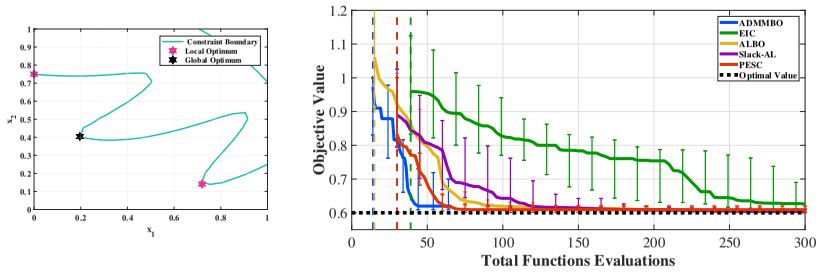


Figure 3: Left: feasible region of the optimization problem (17). Right: performance of different methods as a function of the total budget for function evaluation.

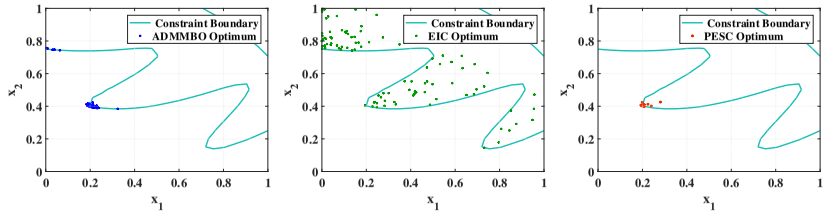


Figure 4: The feasible region of (17) and the best solutions obtained by ADMMBO (left), EIC (middle) and PESC (right) after 50 function evaluations, over 100 runs. For each method, each point represents the final solution of that method over one run, after 50 total function evaluations.

its proposed solutions are scattered throughout the entire feasible region. According to the Figure 3, all methods, including EIC, ultimately converge to the global optimum. However, ADMMBO and PESC achieve this sooner and at a lower budget.

5.5. Test Problem in Higher Dimensions

We chose the first two problems, defined over a two-dimensional space, in order to be able to visualize the feasible regions, global and local optima, as well as the final solution of different algorithms. Here, we evaluate the performance of ADMMBO on a test problem which is defined over a higher dimensional space. This is a modification of the example in (Picheny et al., 2016), where the constraint is the Hartman 4-dimensional function. This function was used as an equality constraint in (Picheny et al., 2016), however, we modified it as an inequality constraint to follow the format of (5). More specifically, for $\mathcal{B} = [0,1]^4$,

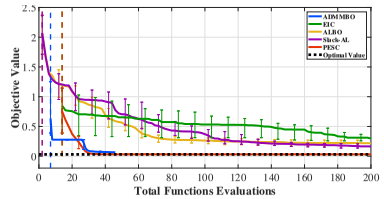


Figure 5: Performance of different algorithms solving (18) as a function of the total budget for function evaluation.

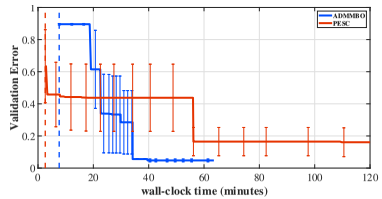


Figure 6: Comparison between ADMM and PESC for hyperparameter tuning for a neural network with short prediction time.

we consider the optimization problem

$$\begin{aligned} \min_{x \in \mathcal{B}} \quad & \sum_{i=1}^{4} x_i \\ \text{s.t.} \quad & 1.1923 \Big[\sum_{i=1}^{4} E_i \exp\big(- \sum_{j=1}^{4} A_{ji} (x_j - P_{ji})^2 \big) - 1.1 \leq 0 \Big], \end{aligned} \tag{18}$$

where A_{ji}, E_i and P_{ji} denote, respectively, the entries of

$$A = \begin{bmatrix} 10.00 & 0.05 & 3.00 & 17.00 \\ 3.00 & 10.00 & 3.50 & 8.00 \\ 17.00 & 17.00 & 1.70 & 0.05 \\ 3.50 & 0.10 & 10.00 & 10.00 \end{bmatrix}, \quad E = \begin{bmatrix} 1.0 \\ 1.2 \\ 3.0 \\ 3.2 \end{bmatrix}, \quad P = \begin{bmatrix} 0.131 & 0.232 & 0.234 & 0.404 \\ 0.169 & 0.413 & 0.145 & 0.882 \\ 0.556 & 0.830 & 0.352 & 0.873 \\ 0.012 & 0.373 & 0.288 & 0.574 \end{bmatrix}.$$

As Figure 5 shows, ADMMBO and PESC compared to EIC, ALBO, and Slack-AL achieve lower value of the objective function after a smaller number of function evaluation. However, similar to other examples, ADMMBO arrives at a feasible point after fewer number of functions evaluations compared to PESC. As an interesting observation, in the budget range of $[5, 25]$, ADMMBO shows a flat curve, which we speculate is due to being at a local minima, however, finally ADMMBO escapes this local minimizer. Again, as this figure shows, an advantage of ADMMBO compared to existing work is its efficient stopping criterion that allows our algorithm to terminate before consuming the total budget, hence, avoiding unnecessary function evaluations.

5.6. Tuning a Fast Neural Network

In our last experiment, we tune the hyperparameters of a three-hidden-layers fully connected neural network for a multiclass classification task using MNIST dataset (LeCun, 1998; LeCun et al., 2015). A version of this experiment was previously reported in (Hernández-Lobato et al., 2015). We train the network using stochastic gradient descent with momentum. We consider the optimization problem of finding a set of hyperparameters that minimize the validation error subject to the prediction time being smaller than or equal

to 0.045 second on NVIDIA Tesla K80 GPU. [2] Here, we focus on eleven hyperparameters: learning rate, decay rate, momentum parameter, two drop out probabilities for the input layer and the hidden layers as well as two regularization parameters for the weight decay, the weight maximum value, the number of hidden units in each of the 3 hidden layers, and the choice of activation function (RELU or sigmoid). We define $\mathcal{B} = [0\ 1; 0\ 1; 0\ 1; -4\ 1; 0\ 100; -4\ 0; -3\ 0; 50\ 500; 50\ 500; 50\ 500; 0\ 1]$. We build our network using Keras with TensorFlow backends (Chollet et al., 2015; Abadi et al., 2016). We compute the prediction time as the average time of 1000 predictions, over a batch size of 128 (Hernández-Lobato et al., 2015). Note that as mentioned in section 1, evaluating the prediction time may not require training the model and could be cheaply done using arbitrary weights.

We compare ADMMBO only with PESC, since as previously reported (Hernández-Lobato et al., 2015, 2016) (and also consistent with our results on the synthetic experiments), PESC typically outperforms EIC and ALBO. Moreover, among state-of-the-art methods, ADMMBO and PESC are the only ones capable of handling decoupled problems, and thus are a good fit for this experiment. Note that since the computational cost of evaluating the validation error and the prediction time are significantly different, we show the results in terms of total wall-clock time rather than the total number of function evaluations.

As the results in Figure 6 show, PESC performed better at first. PESC found the first feasible set of hyperparameters slightly faster than ADMMBO, and also was able to find hyperparameters with lower validation error compared to the hyperparameters suggested by ADMMBO. However, around 18 minutes after initializing the algorithms, ADMMBO's performance started to improve and outperformed PESC from minute 22 on. For example, at minute 40, ADMMBO found a desirable set of hyperparameters resulting in 0.05 validation error and less than 0.045 seconds prediction time. After the same time, PESC's suggested hyperparameter result in a shorter prediction time less than 0.045 seconds, but their validation error was around 0.45. One interesting observation is that ADMMBO terminated after around one hour, satisfying its stopping criterion, avoiding extra expensive evaluations.

5.7. Sensitivity Analysis on M and ρ

In this section, we report on an evaluation of the sensitivity of ADMMBO to the hyperparameters M and ρ. In the first set of experiments, we set the value of M to 20 and ran ADMMBO for fifteen uniformly distributed initial values of $\rho \in [0.0001, 2]$, while keeping the rest of the hyperparameters as in 5.1. In Figure 7 we report on some selected cases, to avoid cluttering the figure. The figure illustrates that ADMMBO's performance was not very sensitive to the initial value of ρ. In particular, for initial $\rho = 2$, ADMMBO attained a feasible point over all 100 runs after no more than 27 function evaluations, while for other values of ρ, the same was achieved after roughly 15 evaluations. Even with $\rho = 2$, at budget 15, 91 out of 100 runs had already found a feasible solution. The vertical dashed line in Figure 7 shows the budget at which the last run found a feasible solution.

2. We chose the value 0.045 empirically, as it resulted in an active trade off between the objective and the constraint, while also ensuring consideration of limitations on the resources that happened to be available to us in our implementation. Clearly this choice will be highly implementation dependent, both in terms of problem and computational platform.

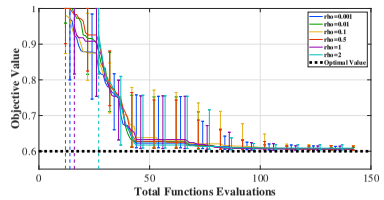


Figure 7: Performance of ADMMBO solving (17) given $M = 20$ and different values of ρ.

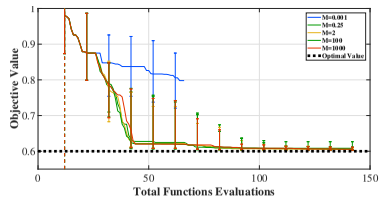


Figure 8: Performance of ADMMBO solving (17) given $\rho = 0.1$ and different values of M.

In the second set of experiments, we set the value of ρ to 0.1 and ran ADMMBO for ten uniformly distributed values of $M \in [0.001, 2]$ as well as $M = 100, 1000$. As Figure 8 shows, again reporting on a subset of the values tested for clarity, the performance of ADMMBO did not depend strongly on the precise value of M as long as it was large enough. Note that in problem (17), the range of f over $\mathcal{B}$ was 2. Even for $M < 2$, ADMMBO found good solutions, failing only when M was 4 order of magnitude smaller than the bound. Also, ADMMBO with different values of M found a feasible point over 100 runs at the same budget. Finally, given all different combinations of M and ρ, ADMMBO always converged before spending the total iteration budget of 300.

6. DISCUSSION

In this paper, we address the problem of solving an optimization whose objective function and constraints are unknown and available to evaluate pointwise, but at high computational cost. We proposed a novel constrained Bayesian optimization algorithm, called ADMMBO, which merges ADMM, a powerful tool from numerical optimization, with Bayesian optimization techniques. ADMMBO defines a set of unconstrained subproblems, over the modified objective function and over modified constraints, and iteratively solves them using Bayesian optimization on each subproblem. Some key advantages of ADDMBO are its ability to start from an infeasible point, its ability to effectively handle decoupled problems, the ability to find closed-form acquisition functions, and its stopping criterion. We showed the effectiveness of ADMMBO through experiments on benchmark problems and the problem of hyperparameter tuning for a fast neural network for digit recognition. ADMMBO consistently outperformed existing methods and obtained the feasible optimum with the fewest number of black-box evaluations. We speculate that the reason behind this rapid convergence is that ADMMBO typically first finds the unconstrained optimum of the problem, and then looks for the closest point to that optimum which belongs to the feasible set, which turns out to be an effective search strategy.

There are several ways in which we believe ADMMBO could be extended or improved. One would be to enable ADMMBO to handle problems with unknown *equality* constraints. A straightforward approach would be to rewrite an equality constraint as a pair of inequality constraints and employ our current algorithm. Since ADMMBO is highly parallelizable (lines $6 - 11$ in Algorithm 3.1 can be implemented in parallel), increasing the number of constraints may not be a significant computational barrier. However, investigating more

principled ways of handling equality constraints is a topic for future investigation. Another extension of clear interest is to speed up ADMMBO through augmenting the problem with environmental variables which directly affect the duration of function evaluations (Klein et al., 2016). For example, in tuning the hyperparameters of a neural network, the size of the training set or the number of epochs can be regarded as such variable. Then, similar to Snoek et al. (2012), one can penalize ADMMBO's acquisition functions with the inverse duration of function evaluations. We speculate that this may lead to finding a good solution for the unknown optimization problem as quickly as possible. A further extension concerns developing a comprehensive budget management strategy. This will be useful specifically in cases where we have partial knowledge about the unknown functions, in particular an estimate of the relative computational complexity of the objective and constraint functions. In this setting, we may be able to leverage such estimates to more efficiently distribute the budget. Another direction worth exploring is the flexibility within the ADMMBO framework to adopt different random process models (beyond independent GPs) and different acquisition functions (beyond EI). In particular using Predictive Entropy as acquisition function with approximations potentially similar to PESC-F would seem to be a promising approach to explore, given both the results in the literature and the relative success of PESC in our experiments. With any new acquisition function we would face the challenge of efficiently optimizing it. Acquisition functions are often multi-modal and complex, and it is an open question how best to carry out this particular optimization step.

Finally, we mention one current limitation of ADMMBO, which is the number of hyperparameters. Good values will clearly speed up the optimization time of ADMMBO. In our experiments here, we followed the default initialization suggested in (Boyd et al., 2011; Hong and Luo, 2017) for the ADMM-related parameters and were able to obtain favorable results. However, for more complex problems, an adaptive initialization policy, potentially similar to ρ's adaption rule based on primal and dual residuals suggested by Boyd et al. (2011), might make the algorithm less sensitive to the possibility of a poor parameter setting.

Acknowledgements

This project was supported by NIH grant R01CA199673 from NCI.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

Javad Azimi, Ali Jalali, and Xiaoli Fern. Hybrid batch bayesian optimization. *arXiv preprint arXiv:1202.5597*, 2012.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Enhanced higgs boson to $\tau+\tau-$ search with deep learning. *Physical review letters*, 114(11):111801, 2015.

James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West. Optimization under unknown constraints. *Bayesian Statistics 9*, 9:229, 2011.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Eric Brochu, Tyson Brochu, and Nando de Freitas. A bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112. Eurographics Association, 2010a.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010b.

François Chollet et al. Keras, 2015.

Dennis D Cox and Susan John. A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, pages 1241–1246. IEEE, 1992.

Daniel E Finkel. Direct optimization algorithm user guide. *Center for Research in Scientific Computation, North Carolina State University*, 2, 2003.

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, pages 937–945, 2014.

Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

Michael Adam Gelbart. *Constrained Bayesian Optimization and Applications*. PhD thesis, 2015.

Robert B Gramacy, Genetha A Gray, Sébastien Le Digabel, Herbert KH Lee, Pritam Ranjan, Garth Wells, and Stefan M Wild. Modeling an augmented lagrangian for blackbox constrained optimization. *Technometrics*, 58(1):1–11, 2016.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive entropy search for bayesian optimization with unknown constraints. In *International Conference on Machine Learning*, pages 1699–1707, 2015.

José Miguel Hernández-Lobato, Michael A Gelbart, Ryan P Adams, Matthew W Hoffman, and Zoubin Ghahramani. A general framework for constrained bayesian optimization using information-based search. 2016.

Matthew Hoffman, Bobak Shahriari, and Nando Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Artificial Intelligence and Statistics*, pages 365–374, 2014.

Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1): 337–364, 2016.

Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Jose M Hernández-Lobato. Collaborative gaussian processes for preference learning. In *Advances in Neural Information Processing Systems*, pages 2096–2104, 2012.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. *arXiv preprint arXiv:1605.07079*, 2016.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Daniel James Lizotte. *Practical bayesian optimization*. University of Alberta, 2008.

Ruben Martinez-Cantin, Nando de Freitas, Arnaud Doucet, and José A Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and Systems*, volume 3, pages 334–341, 2007.

Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

J Močkus. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer, 1975.

Joao FC Mota, Joao MF Xavier, Pedro MQ Aguiar, and Markus Puschel. D-admm: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

Victor Picheny. A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial Intelligence and Statistics*, pages 787–795, 2014.

Victor Picheny, David Ginsbourger, Yann Richet, and Gregory Caplin. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55(1):2–13, 2013.

Victor Picheny, Robert B Gramacy, Stefan Wild, and Sebastien Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian. In *Advances in Neural Information Processing Systems*, pages 1435–1443, 2016.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

Matthias Schonlau, William J Welch, and Donald R Jones. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25, 1998.

Steven L Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016.

Jasper Snoek. *Bayesian optimization and semiparametric models with applications to assistive technology*. PhD thesis, Citeseer, 2013.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task bayesian optimization. In *Advances in neural information processing systems*, pages 2004–2012, 2013.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Aimo Torn and Antanas Zilinskas. *Global optimization*. Springer-Verlag New York, Inc., 1989.

Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, pages 1–35, 2015.

Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, pages 5273–5284, 2017.

Zheng Xu, Soham De, Mario Figueiredo, Christoph Studer, and Tom Goldstein. An empirical study of admm for nonconvex problems. *arXiv preprint arXiv:1612.03349*, 2016.

Journal of Machine Learning Research 20 (2019) 1-62 Submitted 5/18; Revised 8/19; Published 8/19

Deep Exploration via Randomized Value Functions

Ian Osband IOSBAND@GOOGLE.COM
DeepMind

Benjamin Van Roy BVR@STANFORD.EDU
Stanford University

Daniel J. Russo DJR2174@GSB.COLUMBIA.EDU
Columbia University

Zheng Wen ZWEN@ADOBE.COM
Adobe Research

Editor: Peter Auer

Abstract

We study the use of randomized value functions to guide deep exploration in reinforcement learning. This offers an elegant means for synthesizing statistically and computationally efficient exploration with common practical approaches to value function learning. We present several reinforcement learning algorithms that leverage randomized value functions and demonstrate their efficacy through computational studies. We also prove a regret bound that establishes statistical efficiency with a tabular representation.

Keywords: Reinforcement learning, exploration, value function, neural network

1. Introduction

Reinforcement learning might provide the basis for an artificial intelligence that can manage a wide range of systems and better serve the needs of society. To date, its potential has primarily been assessed through learning in simulated systems, where data generation is relatively unconstrained and algorithms are routinely trained over millions to trillions of episodes. Real systems, where data collection is costly or constrained by the physical context, call for a focus on statistical efficiency. A key driver of statistical efficiency is how the agent explores its environment.

The design of reinforcement learning algorithms that efficiently explore intractably large state spaces remains an important challenge. Though a substantial body of work addresses efficient exploration, most of this focusses on tabular representations in which the number of parameters learned and the quantity of data required scale with the number of states. Despite valuable insights that have been generated through design and analysis of tabular reinforcement learning algorithms, they are of limited practical import because, due to the curse of dimensionality, state spaces in most contexts of practical interest are enormous. There is a need for algorithms that generalize across states while exploring intelligently to learn to make effective decisions within a reasonable time frame.

In this paper, we develop a new approach to exploration that serves this need. We build on value function learning, which underlies the most popular and successful approaches to reinforcement learning. In common value function learning approaches, the agent maintains

©2019 Ian Osband, Benjamin Van Roy, Daniel J. Russo and Zheng Wen.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-339.html.

a point estimate of a function mapping state-action pairs to expected cumulative future reward. This estimate typically takes a parameterized form, such as a linear combination of features or a neural network, with parameters fit to past observations. The estimate approximates the agent's prevailing expectation of the true value function, and can be used to guide action selection. As actions are applied and new observations gathered, parameters are adapted to fit the growing data set. The hope is that this process quickly converges on a mode in which the agent selects near optimal actions and new observations reinforce prevailing value estimates.

In using the value function estimate to guide actions, the agent could operate according to a greedy policy, which at any given state, applies the action that maximizes estimated value. However, such a policy does not investigate poorly-understood actions that are assigned unattractive point estimates. This can forgo enormous potential value; it is worthwhile to experiment with such an action since the action could be optimal, and learning that can provide cumulating future benefit over subsequent visits to the state. Thoughtful exploration can be critical to effective learning.

The simplest and most widely used approaches to exploration perturb greedy actions with random *dithering*. An example is ϵ-greedy exploration, which selects the greedy action with probability $1 - \epsilon$ and otherwise selects uniformly at random from all currently available actions. Dithering induces the experimentation required to learn about actions with unattractive point estimates. However, such approaches waste much exploratory effort because they do not "write-off" actions that are known to be inferior. This is because exploratory actions are selected without regard to the level of uncertainty associated with value estimates. Clearly, it is only worth experimenting with an action that is expected to be undesirable if there is sufficient uncertainty surrounding that assessment. As we will discuss further in Section 4, this inefficiency can result in learning times that grow exponentially with the number of states.

A more sophisticated approach might only experiment with an action when applying the action will reveal useful information. We refer to such approaches as *myopic*, since they do not account for subsequent learning opportunities made possible by taking an action. Though myopic approaches do "write off" actions where dithering approaches fail to, as we will discuss in Section 4, myopic exploration can also require learning times that grow exponentially with the number of states or even entirely fail to learn.

Reliably efficient reinforcement learning calls for *deep exploration*. By this we mean that the exploration method does not only consider immediate information gain but also the consequences of an action on future learning. A deep exploration method could, for example, choose to incur losses over a sequence of actions while only expecting informative observations after multiple time periods. Dithering and myopic approaches do not exhibit such strategic pursuit of information.

In this paper, we develop a new approach to deep exploration. The idea is to apply actions that are greedy with respect to a randomly drawn statistically plausible value function. Roughly speaking, we aim to sample from a proxy of the posterior distribution over value functions. Such randomized value functions incentivize experimentation with actions of highly uncertain value, since this uncertainty translates into variance in the sampled value estimate. This randomness often generates positive bias and therefore induces exploration.

There is much more to be said about the design of algorithms that leverage randomized value functions, and we cover some of this ground in Section 5. It is worth mentioning here, though, that this concept is abstract and broadly applicable, transcending specific algorithms. Randomized value functions can be synthesized with the multitude of useful algorithmic ideas in the reinforcement learning literature to produce custom approaches for specific contexts.

To provide insight into the efficacy of randomized value functions, in Section 6, we establish a strong bound on the Bayesian regret of a tabular algorithm. This is not the first result to establish strong efficiency guarantees for tabular reinforcement learning. However, previous algorithms that have been shown to satisfy similar regret bounds do not extend to contexts involving generalization via parameterized value functions. In this regard, the approach we present is the first to satisfy a strong regret bound with tabular representations while also working effectively with the wide variety of practical value function learning methods that generalize over states and actions.

Section 7 presents computational results guided by randomized value functions that synthesize efficient exploration with generalization. Experiments with a family of simple toy examples demonstrate dramatic efficiency gains relative to dithering approaches for exploration and that our randomized approaches are compatible with linearly parameterized generalizing value functions. We also consider a cart-pole balancing problem that requires both deep exploration and generalization. We address this problem through a combination of randomization and deep learning, with value functions represented by neural networks.

2. Literature review

The Bayes-optimal policy serves as a gold standard for statistically efficient exploration in reinforcement learning (RL). Given a prior distribution over Markov decision processes, one can formulate a problem to maximize expected cumulative reward by taking an action at each future time contingent on the prevailing posterior distribution. A policy that attains this maximum is Bayes-optimal, and to do this it must explore judiciously. Unfortunately, for problems of practical interest, computing a Bayes-optimal policy is intractable; the computational requirements grow exponentially in the problem parameters (Szepesvári, 2010). We introduce an approach based on *randomized value functions* that offers a computationally tractable approach to statistically efficient reinforcement learning. Exploration via randomized value functions is not generally Bayes-optimal but, as we will argue, offers a practical approach to deep exploration, which common exploration schemes fail to address, sometimes at enormous cost to statistical efficiency.

There is a substantial body of work on simultaneously computationally and statistically efficient exploration in tabular RL. This begins with the seminal work of Kearns and Singh (Kearns and Singh, 2002), which identified the necessity of multi-period exploration strategies – for which we adopt the term *deep exploration* – to polynomial-time learning and established a polynomial-time learning guarantee for a particular tabular algorithm. Subsequent papers proposed and analyzed alternative tabular algorithms that carry out deep exploration with varying degrees of efficacy (Brafman and Tennenholtz, 2002; Auer and Ortner, 2006; Strehl et al., 2006; Jaksch et al., 2010; Osband et al., 2013; Dann and Brunskill, 2015; Azar et al., 2017). None of these algorithms are Bayes-optimal, but they do

bound the level of sub-optimality by some polynomial function of states and/or planning horizon. By contrast, popular schemes such as ϵ-greedy and Boltzmann exploration can require learning times that grow exponentially in the number of states and/or the planning horizon (see, e.g., Kakade (2003); Strehl (2007)). We discuss this phenomenon further in Section 4.

The design and analysis of tabular algorithms has generated valuable insights, but the resultant algorithms are of little practical importance since, for practical problems the state space is typically enormous (due to the curse of dimensionality). To learn effectively, practical RL algorithms must generalize across states to make effective decisions with limited data. The literature offers a rich collection of such algorithms (e.g. Bertsekas and Tsitsiklis (1996); Sutton and Barto (2018); Szepesvári (2010); Powell and Ryzhov (2011) and references therein). Though algorithms of this genre have achieved impressive outcomes, notably in games such as backgammon (Tesauro, 1995), Atari arcade games (Mnih et al., 2015), and go (Silver et al., 2016, 2017), they use naive exploration schemes that can be highly inefficient. Possibly for this reason, these applications required enormous quantities of data. In the case of Silver et al. (2016), for example, neural networks were trained over hundreds of billions to trillions of simulated games.

The design of reinforcement learning algorithms that efficiently explore intractably large state spaces remains an important challenge. Model learning algorithms exploit generalization in an underlying model of the environment (Kearns and Koller, 1999; Abbasi-Yadkori and Szepesvári, 2011; Ibrahimi et al., 2012; Ortner and Ryabko, 2012; Osband and Van Roy, 2014a,b; Gopalan and Mannor, 2015). However, these are typically restricted to simple model classes and become statistically or computationally intractable for problems of practical scale. Policy learning algorithms, and the closely-related 'evolutionary' algorithms identify high-performers among a set of policies (Kakade, 2003; Wierstra et al., 2008; Deisenroth et al., 2013; Plappert, 2017). These algorithms can perform well, particularly when the space of possible optimal policies is parameterized to be small. However, in a typical problem the space of policies is exponentially large; existing works either entail overly restrictive assumptions or do not make strong efficiency guarantees.

Value function learning has the potential to overcome computational challenges and offer practical means for synthesizing efficient exploration and effective generalization. A relevant line of work establishes that efficient reinforcement learning with value function generalization reduces to efficient "knows what it knows" (KWIK) online regression (Li and Littman, 2010; Li et al., 2008). However, it is not known whether the KWIK online regression problem can be solved efficiently. In terms of concrete algorithms, there is optimistic constraint propagation (OCP) (Wen and Van Roy, 2013), a provably efficient reinforcement learning algorithm for exploration and value function generalization in deterministic systems, and C-PACE (Pazis and Parr, 2013), a provably efficient reinforcement learning algorithm that generalizes using interpolative representations. These contributions represent important developments, but OCP is not suitable for stochastic systems and is highly sensitive to model misspecification, and generalizing effectively in high-dimensional state spaces calls for methods that extrapolate.

In this paper, we leverage randomized value functions to explore efficiently while generalizing via parameterized value functions. Algorithms of the kind we consider, which we refer to collectively as randomized least-squares value iteration (RLSVI), were first introduced in

Wen (2014). Prior reinforcement learning algorithms that generalize via parameterized value functions require, in the worst case, learning times exponential in the number of model parameters and/or the planning horizon. RLSVI aims to overcome these inefficiencies. While RLSVI operates in a manner similar to well-known approaches such as least-squares value iteration (LSVI) and SARSA (see, e.g. Sutton and Barto (2018)), what fundamentally distinguishes RLSVI is exploration through randomly sampling statistically plausible value functions. Alternatives such as LSVI and SARSA are typically applied in conjunction with action-dithering schemes such as Boltzmann or ϵ-greedy exploration, which lead to highly inefficient learning.

This paper aims to establish the use of *randomized value functions* as a promising approach to tackling a critical challenge in reinforcement learning: synthesizing efficient exploration and effective generalization. The only preceding work that advocates exploration through random samples of the value function comes from Dearden et al. (1998). This paper proposes a tabular algorithm that resamples every timestep and so does not perform deep exploration. A preliminary version of part of this work appeared in a short conference paper (Osband et al., 2016b). While that paper proposed a specific algorithm that is compatible with linear function approximation, this paper develops the concept of deep exploration via randomized value functions in much greater depth and generality. We provide a general template for building algorithms that perform randomized value function learning and propose several specific instantiations of this idea. These algorithms are evaluated in entirely new simulations experiments, including an example in which neural networks are used for function approximation. While a proof sketch for a regret bound was given in (Osband et al., 2016b), this paper gives a full and careful proof and develops new recursive stochastic dominance arguments that simplify the analysis. Following (Osband et al., 2016b), several papers have proposed adaptations to neural network function approximation via bootstrap sampling (Osband et al., 2016a), linear final layer approximation (Azizzadenesheli et al., 2018) or variational inference (Lipton et al., 2018; Fortunato et al., 2018).

The mathematical analysis we present in Section 6 establishes a bound on expected regret for a tabular version of RLSVI applied to an episodic finite-horizon problem, where the expectation is taken with respect to a particular uninformative distribution. We view this result as a sanity check that, although it is designed for exploration with generalization, RLSVI recovers state-of-the-art efficiency guarantees in the simple tabular setting. Our bound is $\tilde{O}(H\sqrt{SAHL})$, where S and A denote the cardinalities of the state and action spaces, L denotes the number of episodes elapsed, and H denotes the episode duration. The lower bound of Jaksch et al. (2010) can be adapted to the episodic finite-horizon context to produce a $\Omega(H\sqrt{SAL})$ lower bound that applies to any algorithm. This differs from our upper bound by a factor of $\sqrt{H}$, though this is not an apples-to-apples comparison, since the lower bound applies to a maximum over Markov decision processes and may not hold for the expectation over Markov decision processes, taken with respect to a prior distribution we posit. Follow up work by Russo (2019) shows that RLSVI also satisfies worst-case regret bounds in tabular environments.

In recent years there has been significant interest in alternative methods to incentivize exploration. One popular method uses a density model or "pseudocount" to assign a bonus to states that have been visited infrequently (Bellemare et al., 2016; Tang et al., 2016). These methods can perform well, but only when the generalization of the density model

is aligned with the task objective. Crucially, this generalization is not learned from the task and, unlike the optimal value function, "counts" are generated by the agent's choices so there is no single target function to learn. Further, these approaches add uncertainty bonus that is uncoupled across states, which can lead to a substantial negative impact on statistical efficiency, as discussed in (Osband and Van Roy, 2017; O'Donoghue et al., 2017).

Exploration via randomized value functions is inspired by Thompson sampling (Thompson, 1933; Russo et al., 2018). In particular, when generating a randomized value function, the aim is to approximately sample from the posterior distribution of the optimal value function. There are problems where Thompson sampling is in some sense near-optimal (Agrawal and Goyal, 2012, 2013a,b; Russo and Van Roy, 2013, 2014a; Gopalan and Mannor, 2015). Further, the theory suggests that "well-designed" upper-confidence-bound-based approaches, which appropriately couple uncertainties across state-action pairs, but are often computationally intractable, are similarly near-optimal (statistically) and competitive with Thompson sampling in such contexts (Russo and Van Roy, 2013, 2014a). On the other hand, for some problems with more complex information structures, it is possible to explore much more efficiently than do Thompson sampling or upper-confidence-bound methods (Russo and Van Roy, 2014b). As such, for some RL problems and value function representations, the randomized value function approaches we put forth will leave substantial room for improvement.

At a high level, randomized value functions replaces a point estimate of the value function by a distribution of plausible value functions. Recently, another approach called "distributional RL" also suggests replacing a scalar value estimate by a distribution (Bellemare et al., 2017). Although both might reasonably claim to offer a distributional perspective on reinforcement learning, the meaning and utility of the two distributions are quite distinct. Randomized value functions aim to sample from a distribution that captures the Bayesian uncertainty in the unknown optimal value function; this concentrates around the true value function as more data is gathered. By contrast, "distributional RL" fits a distribution to the realized value under stochastic outcomes. For efficient exploration of *unknown* rather than stochastic outcomes, it is important to use the correct notion of "distributional RL".

3. Reinforcement learning problem

We consider a reinforcement learning problem in which an agent interacts with an unknown environment over a sequence of episodes. We model the environment as a Markov decision process, identified by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho)$. Here, $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $\mathcal{R}$ is a reward model, $\mathcal{P}$ is a transition model, and $\rho \in \mathcal{S}$ is an initial state distribution. For each s, $\rho(s)$ is the probability that an episode begins in state s. For any $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$, $\mathcal{R}_{s,a,s'}$ is a distribution over real numbers and $\mathcal{P}_{s,a}$ is a sub-distribution over states. In particular, $\mathcal{P}_{s,a}(s')$ is the conditional probability that the state transitions to s' from state s and action a. Similarly, $\mathcal{R}_{s,a,s'}(dr)$ is the conditional probability that the reward is in the set dr. By *sub-distribution*, we mean that the sum can be less than one. The difference $1 - \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,a}(s')$ represents the probability that the process terminates upon transition.

We will denote by r_t^ℓ, s_t^ℓ, a_t^ℓ the state, action, and reward observed at the start of the tth time period of the ℓth episode. In each ℓth episode, the agent begins in a random

state $s_0^\ell \sim \rho$ and selects an action $a_0^\ell \in \mathcal{A}$. Given this state-action pair, a reward and transition are generated according to $r_1^\ell \sim \mathcal{R}_{s_0^\ell, a_0^\ell, s_1^\ell}$ and $s_1^\ell \sim \mathcal{P}_{s_0^\ell, a_0^\ell}$. The agent proceeds until termination, in each tth time period observing a state s_t^ℓ, selecting an action a_t^ℓ, and then observing a reward r_{t+1}^ℓ and transition to s_{t+1}^ℓ. Let τ_ℓ denote the random time at which the process terminates, so that the sequence of observations made during episode ℓ is $\mathcal{O}_\ell = \left(s_0^\ell, a_0^\ell, r_1^\ell, s_1^\ell, a_1^\ell, \ldots, s_{\tau_\ell-1}^\ell, a_{\tau_\ell-1}^\ell, r_{\tau_\ell}^\ell\right)$.

We define a *policy* to be a mapping from $\mathcal{S}$ to a probability distribution over $\mathcal{A}$, and denote the set of all policies by Π. We will denote by $\pi(a|s)$ the probability that π assigns to action a at state s. Without loss of generality, we will consider states and actions to be integer indices, so that $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$ and $\mathcal{A} = \{1, \ldots, |\mathcal{A}|\}$. As such, we can define a substochastic matrix whose (s, s')th element is $\sum_{a \in \mathcal{A}} \pi(a|s)\mathcal{P}_{s,a}(s')$. We make the following assumption to ensure finite episode duration:

Assumption 1 *For all policies $\pi \in \Pi$, if each action a_t is sampled from $\pi(\cdot|s_t)$, then the MDP $\mathcal{M}$ almost surely terminates in finite time. In other words, $\lim_{t\to\infty} P_\pi^t = 0$, where P_π is the matrix whose (s, s')th element is $\sum_{a \in \mathcal{A}} \pi(a|s)\mathcal{P}_{s,a}(s')$.*

For any MDP $\mathcal{M}$ and policy $\pi \in \Pi$, we define a value function $V_{\mathcal{M}}^\pi : \mathcal{S} \mapsto \mathbb{R}$ by

$$V_{\mathcal{M}}^\pi(s) = \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=1}^{\tau} r_t \;\middle|\; s_0 = s\right],$$

where r_t, s_t, a_t, and τ denote rewards, states, actions, and termination time of a generic episode, and the subscripts of the expectation indicate that actions are sampled according to $a_t \sim \pi(\cdot|s_t)$ and transitions and rewards are generated by the MDP $\mathcal{M}$. Further, we define an optimal value function:

$$V_{\mathcal{M}}^\star(s) = \max_{\pi \in \Pi} V_{\mathcal{M}}^\pi(s).$$

The agent's behavior is governed by a reinforcement learning algorithm alg. Immediately prior to the beginning of episode L, the algorithm produces a policy $\pi^L = \text{alg}(\mathcal{S}, \mathcal{A}, \mathcal{H}_{L-1})$ based on the state and action spaces and the history $\mathcal{H}_{L-1} = (\mathcal{O}_\ell : \ell = 1, \ldots, L-1)$ of observations made over previous episodes. Note that alg may be a randomized algorithm, so that multiple applications of alg may yield different policies.

In episode ℓ, the agent enjoys a cumulative reward of $\sum_{t=1}^{\tau_\ell} r_t^\ell$. We define the *regret* over episode ℓ to be the difference between optimal expected value and the expected value under algorithm alg. This can be written as $\mathbb{E}_{\mathcal{M},\text{alg}}\left[V^\star(s_0^\ell) - V^{\pi^\ell}(s_0^\ell))\right]$, where the subscripts of the expectation indicate that each policy π^ℓ is produced by algorithm alg and state transitions and rewards are generate by MDP $\mathcal{M}$. Note that this expectation integrates over all initial states, actions, state transitions, rewards, and any randomness generated within alg, while the MDP $\mathcal{M}$ is fixed. We denote *cumulative regret* over L episodes by

$$\text{Regret}(\mathcal{M}, \text{alg}, L) = \sum_{\ell=1}^{L} \mathbb{E}_{\mathcal{M},\text{alg}}\left[V^\star(s_0^\ell) - V^{\pi^\ell}(s_0^\ell))\right].$$

We will generally refer to *cumulative regret* simply as *regret*.

When used as a measure for comparing algorithms, one issue with regret is its dependence on $\mathcal{M}$. One way of addressing this is to assume that $\mathcal{M}$ is constrained to a pre-defined set and to design algorithms with an aim of minimizing worst-case regret over this set. This tends to yield algorithms that behave in an overly conservative manner when faced with representative MDPs. An alternative is to aim at minimizing an average over representative MDPs. The distribution over MDPs can be thought of as a prior, which captures beliefs of the algorithm designer. In this spirit, we define *Bayesian regret*:

$$\text{BayesRegret}(\text{alg}, L) = \mathbb{E}\left[\text{Regret}(\mathcal{M}, \text{alg}, L)\right].$$

Here, the expectation integrates with respect to a prior distribution over MDPs.

It is easy to see that minimizing regret or Bayesian regret is equivalent to maximizing expected cumulative reward. These measures are useful alternatives to expected cumulative reward, however, because for reasonable algorithms, $\text{Regret}(\mathcal{M}, \text{alg}, L)/L$ and $\text{BayesRegret}(\text{alg}, L)/L$ should converge to zero. When it is not feasible to apply an optimal algorithm, comparing how quickly these values diminish and how that depends on problem parameters can yield insight.

To denote our prior distribution over MDPs, as well as distributions over any other randomness that is realized, we will use a probability space $(\Omega, \mathbb{F}, \mathbb{P})$. With this notation, the probability that $\mathcal{M}$ takes values in a set $\mathbb{M}$ is written as $\mathbb{P}(\mathcal{M} \in \mathbb{M})$. In fact, the probability of any measurable event $\mathcal{E}$ is written as $\mathbb{P}(\mathcal{E})$.

4. Deep exploration

Reinforcement learning calls for a sophisticated form of exploration that we refer to as *deep exploration*. This form of exploration accounts not only for information gained upon taking an action but also for how the action may position the agent to more effectively acquire information over subsequent time periods. We will use the following simple example to illustrate the critical role of deep exploration as well as how common approaches to exploration fall short on this front.

Example 1 (Deep-sea exploration)
Consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho)$ with $|\mathcal{S}| = N^2$ states, each of which can be thought of as a square cell in an $N \times N$ grid, as illustrated in Figure 1. The action space is $\mathcal{A} = \{1, 2\}$. At each state, one of the actions represents "left" and the other represents "right," with the indexing possibly differing across states. In other words, for a pair of distinct states $s, s' \in \mathcal{S}$, action 1 could represent "left" at state s and "right" at state s'. Any transition from any state in the lowest row leads to termination of the episode. At any other state, the "left" action transitions to the cell immediately to the left, if possible, and below. Analogously, the "right" action transitions to the cell immediately to the right, if possible, and below. The agent begins every episode in the upper-left-most state (where her boat sits). Note that, given the dynamics we have described, each episode lasts exactly N time periods.

From any cell along the diagonal, there is a cost of $0.01/N$ incurred each time the "right" action is chosen. No cost is incurred for the left action. The only other situation that leads to an additional reward or cost arises when the agent is in the lower-right-most cell, where there is a chest. There is an additional reward of 1 (treasure) or cost of 1 (bomb) when the

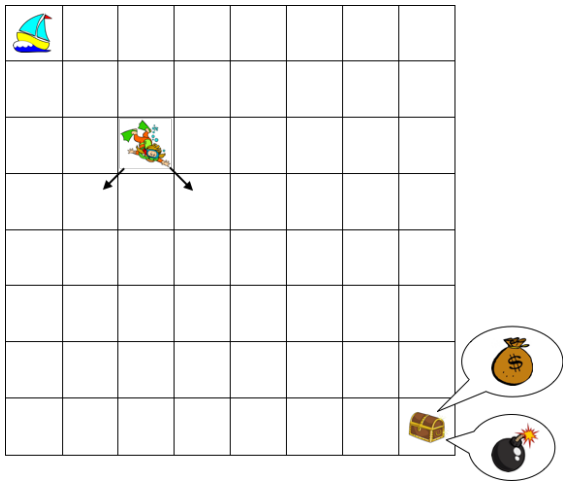


Figure 1: Deep-sea exploration: a simple example where deep exploration is critical.

"right" action is selected at that cell. Conditioned on the $\mathcal{M}$, this reward is deterministic, so once the agent discovers whether there is treasure or a bomb, she knows in subsequent episodes whether she wants to reach or avoid that cell. In particular, given knowledge of $\mathcal{M}$, the optimal policy is to select the "right" action in every time period if there is treasure and, otherwise, to choose the "left" action in every time period. Doing so accumulates a reward of 0.99 *if there is treasure and* 0 *if there is a bomb. It is interesting to note that a policy that randomly explores by selecting each action with equal probability is highly unlikely to reach the chest. In particular, the probability such a policy reaches that cell in any given episode is* $(1/2)^N$. *Hence, the expected number of episodes before observing the chest's content is* 2^N. *Even for a moderate value of* $N = 50$, *this is over a quintillion episodes.*

Let us now discuss the agent's beliefs, or state of knowledge, about the MDP $\mathcal{M}$, prior to the first episode. The agent knows everything about $\mathcal{M}$ except:

- *Action associations. At each state, the agent does not know which action index is associated with "right" or "left", and assigns equal probability to either association. These associations are independent across states.*
- *Reward. The agent does not know whether the chest contains treasure or a bomb and assigns equal probability to each of these possibilities.*

Before learning action associations and rewards, the distribution over optimal value at the initial state is given by $\mathbb{P}(V^*_{\mathcal{M}}(s_0) = 0.99) = \mathbb{P}(V^*_{\mathcal{M}}(s_0) = 0) = 1/2$. *Because the MDP is deterministic, when an agent transitions from any state, she learns the action associations for that state, and when the agent selects the "right" action at the lower-right-most state, she learns whether there is treasure or a bomb.*

Note that the reinforcement learning problem presented in this example is easy to address. In particular, it is straightforward to show that the minimal expected time to learn an optimal policy is achieved by an agent who chooses the "right" action whenever she knows which action that is, and otherwise, applies a random action, until she discovers the content of the chest, at which point she knows an optimal policy. This algorithm identifies an optimal policy within N episodes, since in each episode, the agent learns how to move right from at least one additional cell along the diagonal. Further, the expected learning time is $(N+1)/2$ episodes, since whenever at a state that has not previously been visited, the agent takes the wrong action with probability 1/2. Unfortunately, this algorithm is specialized to Example 1 and does not extend to other reinforcement learning problems. For our purposes, this example will serve as a sanity check and context for illustrating flaws and features of algorithms designed for the general reinforcement learning problem.

To facilitate our discussion, it is useful to define a couple of concepts. The first is that of an *optimal state-action value function*, defined by $Q^*_{\mathcal{M}}(s,a) = \mathbb{E}_{\mathcal{M}}[r + V^*_{\mathcal{M}}(s')]$, where r and s' represent the reward and transition following application of action a in state s. Second, for any $Q : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, the *greedy policy* with respect to Q selects an action that maximizes Q, sampling randomly among alternatives if there are multiple:

$$a \sim \texttt{unif}\left(\operatorname*{argmax}_{\alpha \in \mathcal{A}} Q(s,\alpha)\right). \tag{4.1}$$

Note that the greedy policy with respect to $Q^*_{\mathcal{M}}$ is optimal for the MDP $\mathcal{M}$. This policy depends on the random MDP $\mathcal{M}$, and therefore can not be applied in the process of learning.

The first reinforcement learning algorithm we consider is **pure-exploitation** and aims to maximize expected reward in the current episode, ignoring benefits of active exploration. This algorithm estimates a "best guess" MDP $\hat{\mathcal{M}}_L$ based upon the data it has gathered up until episode L. To offer a representative approach, we will take $\hat{\mathcal{M}}_L$ to be the MDP with rewards and transition probabilities given by their expectations conditioned on the data. The pure-exploitation algorithm then follows the policy that would be greedy with respect to $\hat{Q}_L = Q^*_{\hat{\mathcal{M}}_L}$ during episode L. While this algorithm is applicable to any reinforcement learning problem, its behavior in Example 1 reveals severe inefficiencies. Note that the algorithm is indifferent about finding the chest, since the expected reward associated with that is 0. Further, since moving toward the chest incurs cost, the algorithm avoids that, and therefore never visits the chest. As such, the algorithm is unlikely to ever learn an optimal policy.

Dithering approaches explore by selecting actions that randomly perturb what a pure-exploitation algorithm would do. As an example, one form of dithering, known as *Boltzmann exploration* selects actions according to

$$a_t \sim \texttt{multinomial}\left(\frac{\exp\left(\hat{Q}(s_t,\cdot)/\eta\right)}{\sum_{a\in\mathcal{A}} \exp\left(\hat{Q}(s_t,a)/\eta\right)}\right). \tag{4.2}$$

Here, η represents a "temperature" parameter. As η approaches zero, actions become the same as those that would be selected by a pure-exploitation algorithm. As η increases, the selection becomes noisier, eventually converging to a uniform distribution over actions. In Example 1, a dithering algorithm is biased against moving toward the chest because of the

associated cost. Only the random perturbations can lead the agent to the chest. As such, the expected learning time is $\Theta(2^N)$.[1]

It is well known that dithering can be highly inefficient, even for bandit learning. A key shortcoming is that dithering algorithms do not write-off bad actions. In particular, even when observations make clear that a particular action is not worthwhile, dithering approaches can sample that action. Despite this understanding, dithering is the most widely used exploration method in reinforcement learning. The primary reason for this has been lack of computationally efficient approaches that adequately address the complex problems that arise in practical contexts. This paper aims to fill that need.

Bandit learning can be thought of as a special case of reinforcement learning for which actions bear no delayed consequences. The bandit learning literature offers sophisticated methods that overcome shortcomings of dithering. Such methods write-off bad actions, only selecting an action when it is expected to generate desirable reward or yield useful information or both. A naive way of applying such an algorithm to a reinforcement learning problem involves selecting an action a_t only if the expected value $\hat{Q}_L(s_t, a_t)$ is large or the observed reward and/or transition are expected to provide useful information. We call this approach **myopic exploration**, since it incentivizes exploration over a single timestep. However, applying this approach to Example 1 would once again avoid moving toward the chest as soon as it had learned the action associations in the initial state. This is because there is a cost to moving right, but once the action associations are learned, there is no immediate benefit to applying the "right" action. As such, myopic exploration is unlikely to ever learn an optimal policy.

Myopic exploration does not adequately address reinforcement learning because, in reinforcement learning, there is an additional motivation that should not be overlooked: an action can be desirable even if expected to yield no value or immediate information if the action may place the agent in a state that leads to subsequent learning opportunities. This is the essence of *deep exploration*; the agent needs to consider how actions influence downstream learning opportunities. Viewed in another way, when considering how to explore, the agent should probe *deep* in his decision tree.

Optimism serves as another guiding principle in much of the bandit learning literature and can provide a basis for deep exploration as well. In Example 1, if the agent takes most optimistic plausible view, it would assume that the chest offers treasure rather than a bomb, so long as this hypothesis has not been invalidated. In each Lth episode, the agent follows a greedy policy with respect to a value function Q_L that assigns to each state-action pair the maximal expected value under this assumption. When at a cell along the diagonal of the grid, this policy selects the "right" action whenever the agent knows which that is. Hence, this optimistic algorithm learns the optimal policy within N episodes.

The optimistic algorithm attains its strong performance in Example 1 through carrying out deep exploration. In particular, by assuming treasure rather than a bomb, the agent is incentivized to move right whenever it can, since that is the only way to obtain the posited treasure. This exploration strategy is deep since the agent does not seek only immediate information but also a learning opportunity that will only arise after consecutively moving right over multiple time periods.

1. A similar observation holds true for another popular dithering strategy: ϵ-greedy. This approach selects random actions according to probability ϵ and greedy (pure-exploitation) otherwise.

There are reasonably effective optimistic algorithms that apply to reinforcement learning problems with small (tractably enumerated) state and action spaces. However, the design of such algorithms that adequately address reinforcement learning problems of practical scale in a computationally tractable manner remains a challenge.

An alternative approach studied in the bandit learning literature involves randomly sampled instead of optimistic estimates. A focus of this paper is to extend this approach – known as Thompson sampling – to accommodate deep exploration in complex reinforcement learning problems. Applied to Example 1, this **randomized** approach would sample before each episode a random estimate $\tilde{Q}_L$ from the agent's posterior distribution over $Q^*_{\mathcal{M}}$, conditioned on observations made over previous episodes, or an approximation of this posterior distribution. Before the agent's first visit to the chest, she assigns equal probability to treasure and a bomb, and therefore, the sample $\tilde{Q}_L$ has an equal chance of being optimistic or pessimistic. The agent selects actions according to the greedy policy with respect to $\tilde{Q}_L$ and therefore on average explores over half of the episodes in a manner similar to an optimistic algorithm. As such, the randomized algorithm can expect to learn the optimal policy within $2N$ episodes.

As applied to Example 1, there is no benefit to using a randomized rather than optimistic approach. However, in the face of in complex reinforcement learning problems, the randomized approach can lead to computationally tractable algorithms that carry out deep exploration where the optimistic approach does not.

Table 1 summarizes our discussion of learning times of various exploration methods applied to Example 1. The minimal time required to learn an optimal policy, which is achieved by an agent who moves right whenever she knows how to, is $\Theta(N)$ episodes. The pure-exploitation algorithm avoids *any* active exploration and requires $\Theta(2^N)$ episodes to learn. Dithering does not help for our problem. Though more sophisticated, myopic approaches do not carry out deep exploration, and as such, still require $\Theta(2^N)$ episodes. Optimistic and randomized approaches require only $\Theta(N)$ episodes.

exploration method	expected episodes to learn
optimal	$\Theta(N)$
pure exploitation	∞
myopic	∞
dithering	$\Theta(2^N)$
optimistic	$\Theta(N)$
randomized	$\Theta(N)$

Table 1: Expected number of episodes required to learn an optimal policy for Example 1.

5. Algorithms

The field of reinforcement learning has produced a substantial body of algorithmic ideas that serve as ingredients to mix, match, and customize in tailoring solutions to specific applications. Such ideas are well-summarized in the textbooks of Bertsekas and Tsitsiklis (1996) and Sutton and Barto (2018), among others. The aim of this paper is to contribute to this corpus a new approach to exploration based on randomized value functions, with

the intention that this additional ingredient will broadly enable computationally efficient deep exploration.

Much of the literature and most notable applications build on value function learning. This involves fitting a parameterized value function to observed data in order to estimate the optimal value function. The algorithms we present will be of this genre. As a starting point, in Section 5.2, we will describe least-squares value iteration (LSVI), which is perhaps the simplest of value function learning algorithms. In Section 5.3, we consider modifying LSVI by injecting randomness in a manner that incentivizes deep exploration. This gives rise to a new class of algorithms, which we will refer to as randomized least-squares value iteration (RLSVI), and which offer computationally tractable means to deep exploration.

LSVI plays a foundational role in the sense that most popular value function learning algorithms can be interpreted as variations designed to improve computational efficiency or robustness to mis-specification of the parameterized value function. The reinforcement learning literature presents many ideas that address such practical considerations. In Section 5.4, we will discuss how such ideas can be brought to bear in tandem with RLSVI.

5.1. Value function learning

Before diving into specific reinforcement learning algorithms, let us discuss general concepts that apply to all of them. Value function learning algorithms make use of a family $\mathcal{Q}$ of state-action value functions indexed by $\theta \in \mathbb{R}^d$. Each $\mathcal{Q}_\theta : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ identifies a state-action value function.[2] As a simple example of such a family, consider representing value functions as linear combinations of fixed features. In particular, if $\phi(s,a) \in \mathbb{R}^d$ is a vector of features designed to capture salient characteristics of the state-action pair (s,a), it is natural to consider the family of functions taking the form $\mathcal{Q}_\theta(s,a) = \theta^\top \phi(s,a)$, with $\theta \in \mathbb{R}^d$.

Algorithm 1 (`live`) provides a template for reinforcement learning algorithms we will consider. It operates over an endless sequence of episodes, accumulating observations, learning value functions, and applying actions. We use a Pythonic pseudocode, with an object-oriented division into `agent` and `environment`. We use

$$\texttt{transition} = \texttt{NamedTuple}(\texttt{old_state}, \texttt{action}, \texttt{reward}, \texttt{new_state}, \texttt{timestep}),$$

to describe the evolution of the system. Where convenient, we will alternatively write $\texttt{transition} = (s_t, a_t, r_t, s'_t, t)$. We highlight three key methods the `agent` must implement:

- `act` – select actions given its internal value estimates, (e.g. greedy action selection).
- `update_buffer` – incorporate observations to its memory buffer, (e.g. append to list).
- `learn_from_buffer` – update value estimate given the data in the buffer, (e.g. LSVI).

The agents that we discuss will be distinguished through their implementation of these methods, which we will now outline.

The simplest form of `act` is given by the greedy strategy `act_greedy`. An agent that uses this approach will select actions that maximize its estimated state-action value. If multiple actions attain the maximum, one is sampled uniformly from among them.[3]

2. We adopt notation that for all $\mathcal{Q}$, parameter $\theta_0 = \textbf{null}$ indicates $\mathcal{Q}_{\theta_0} \equiv 0$.
3. We might also consider action selection via `act_epsilon_greedy` or `act_boltzmann`, as described in Section 4. As we saw before, these dithering approaches do not perform *deep exploration* and thus can lead to exponentially slower learning.

Algorithm 1 `live`

Input: `agent` methods `act`, `update_buffer`, `learn_from_buffer`
`environment` methods `reset`, `step`

```
for ℓ in (1, 2, ...) do
    agent.learn_from_buffer()
    transition ← environment.reset()
    while transition.new_state is not null do
        action ← agent.act(transition.new_state)
        transition ← environment.step(action)
        agent.update_buffer(transition)
```

Similarly, the simplest form of `update_buffer` is to simply accumulate all observed data `update_buffer_queue`. Our next two sections will investigate agents that store all observed data and take greedy actions; we investigate the effects of `learn_from_buffer` and explain why training least-squares value iteration on randomly perturbed versions of the data can offer a computationally tractable means to deep exploration.

5.2. Least-squares value iteration

Given an MDP $\mathcal{M}$, one can apply the value iteration algorithm (Algorithm 2) to compute an arbitrarily close approximation to Q^*. The algorithm takes $\mathcal{M}$ and a planning horizon H as input and computes Q^*_H, the optimal value over the next H time periods of the episode as a function of the current state and action. The computation is recursive: given Q^*_h, the algorithm computes Q^*_{h+1} by taking the expected sum of immediate reward and Q^*_h, evaluated at the next state, maximized over actions. Under Assumption 1, the mapping from Q^*_h to Q^*_{h+1} is a weighted-maximum-norm contraction mapping (Bertsekas and Tsitsiklis, 1996), and as such, Q^*_h converges to Q^* at a geometric rate. Hence, for any $\mathcal{M}$ satisfying Assumption 1 and sufficiently large H, the greedy policy with respect to Q^*_H is optimal.

Algorithm 2 `vi`

Input: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho)$ MDP
$H \in \mathbb{N}$ planning horizon
Output: Q^*_H optimal value function for H-period problem

$Q^*_0 \leftarrow 0$
for h in $(0, \ldots, H-1)$ **do**
  $Q^*_{h+1}(s,a) \leftarrow \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,a}(s') \left(\int r \mathcal{R}_{s,a,s'}(dr) + \max_{a' \in \mathcal{A}} Q^*_h(s',a') \right) \ \forall s, a \in \mathcal{S} \times \mathcal{A}$
return Q^*_H

Value iteration offers an idealized approach to evaluating Q^*, given knowledge of the underlying MDP $\mathcal{M}$ and the necessary computational power. Least-squares value iteration (LSVI) adapts `vi` to an RL setting with imperfect statistical knowledge and limited computation. For a value function family $\mathcal{Q} = \{\mathcal{Q}_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$, observed data $\mathcal{D} = \{(s_t, a_t, r_t, s'_t, t)\}$ and target parameters θ^- we define the empirical temporal difference

(TD) loss:

$$\mathcal{L}(\theta;\theta^-,\mathcal{D}) := \sum_{t\in\mathcal{D}}\left(r_t + \max_{a'\in\mathcal{A}} \mathcal{Q}_{\theta^-}(s'_t,a') - \mathcal{Q}_\theta(s_t,a_t)\right)^2. \tag{5.1}$$

Note that, if $\mathcal{Q}$ spans the true value function Q^* and the data $\mathcal{D}$ matches the distribution of $\mathcal{M}$ then the minimizer of $\mathcal{L}$ matches the solution of Algorithm 2; for more information see Sutton and Barto (2018).

Algorithm 3 (`learn_lsvi`) describes the `learn_from_buffer` method for LSVI, whcih approximates the operations carried out by value iteration. The algorithm successively minimizes the empirical temporal difference loss (5.1) plus a regularization term:

$$\mathcal{R}(\theta;\theta^p) := \frac{v}{\lambda}\|\theta^p - \theta\|_2^2. \tag{5.2}$$

Here θ^p can be interpreted as a prior for θ and $\frac{v}{\lambda}$ determines the strength of the regularization coefficient. In a linear system these correspond to a prior belief $\theta \sim N(\theta^p, \lambda I)$ with observations $y_t = x_t\theta + z_t$ for $z_t \sim N(0,v)$. Similarly to `vi`, `learn_lsvi` computes a sequence of value functions $(\mathcal{Q}_{\tilde\theta_h} : h = 0,\dots,H)$, reflecting optimal expected rewards over an expanding horizon. However, while value iteration computes optimal values using full knowledge of the MDP, LSVI produces estimates based only on observed data. In each iteration, for each observed transition (s,a,r,s'), `learn_lsvi` regresses the sum of immediate reward r and the value estimate $\max_{a'\in\mathcal{A}} \mathcal{Q}_{\tilde\theta_h}(s',a')$ at the next state onto the value estimate $\mathcal{Q}_{\tilde\theta_{h+1}}(s,a)$ for the current state-action pair.

Algorithm 3 `learn_lsvi`

Agent:	$\mathcal{L}(\theta{=}\cdot;\theta^-{=}\cdot,\mathcal{D}{=}\cdot)$	TD error loss function
	$\mathcal{R}(\theta{=}\cdot;\theta^p{=}\cdot)$	regularization function
	`buffer`	memory buffer of observations
	`prior`	prior distribution of θ
	$H \in \mathbb{N}$	planning horizon
Updates:	$\tilde\theta$	agent value function estimate

1: $\tilde\theta_0 \leftarrow$ **null**
2: Data $\tilde{\mathcal{D}} \leftarrow$ `buffer.data()`
3: Prior parameter $\tilde\theta^p \leftarrow$ `prior.mean()`
4: **for** h in $(0,\dots,H-1)$ **do**
5: $\quad \tilde\theta_{h+1} \leftarrow \operatorname*{argmin}_{\theta\in\mathbb{R}^D}\left(\mathcal{L}(\theta;\tilde\theta_h,\tilde{\mathcal{D}}) + \mathcal{R}(\theta;\tilde\theta^p)\right)$
6: update value function estimate $\tilde\theta \leftarrow \tilde\theta_H$

In the event that the parameterized value function is flexible enough to represent every function mapping $\mathcal{S}\times\mathcal{A}$ to $\mathbb{R}$, it is easy to see that, for any $\overline{\theta}$ and any positive λ and v, as the observed history grows to include an increasing number of transitions from each state-action pair, value functions $\mathcal{Q}_{\tilde\theta_H}$ produced by LSVI converge to Q^*_H. However, in practical contexts, the data set is finite and the parameterization is chosen to be less flexible in order to enable generalization. As such, $\mathcal{Q}_{\tilde\theta_H}$ and Q^*_H can differ greatly.

In addition to inducing generalization, a less flexible parameterization is critical for computational tractability. In particular, the compute time and memory requirements of value iteration scale linearly with the number of states, which, due to the curse of dimensionality, grows intractably large in most practical contexts. LSVI sidesteps this scaling, instead requiring compute time and memory that scale polynomially with the dimension of the parameter vector $\tilde\theta$, the number of historical observations, and the time required to maximize over actions at any given state.

An LSVI `agent` may also be paired with some dithering strategy for exploration, such as ϵ-greedy or Boltzmann exploration (4.2) in place of `act_greedy`. As discussed in Section 4, randomly perturbing greedy actions – or dithering – does not achieve deep exploration and so can lead to exponentially poor performance. Our next subsection introduces randomized value function estimates as an alternative.

5.3. Randomized least-squares value iteration

At a high level, the idea is to randomly sample an imagined optimal parameter $\tilde\theta$ according to the probability that it is optimal. This approach is inspired by Thompson sampling, an algorithm widely used in bandit learning (Thompson, 1933). In the context of a multi-armed bandit problem, Thompson sampling maintains a belief distribution over models that assign mean rewards to arms. As observations accumulate, this belief distribution evolves according to Bayes rule. When selecting an arm, the algorithm samples a model from this belief distribution and then selects the arm to which this model assigns largest mean reward.

To address a reinforcement learning problem, one could in principle apply Thompson sampling to value function learning. This would involve maintaining a belief distribution over candidates for the optimal value function. Before each episode, we would sample a function from this distribution and then apply the associated greedy policy over the course of the episode. This approach could be effective if it were practically viable, but distributions over value functions are complex to represent and exact Bayesian inference would likely prove computationally intractable.

Randomized least-squares value iteration (RLSVI) is modeled after this Thompson sampling approach and serves as a computationally tractable method for sampling value functions. RLSVI was first introduced in Wen (2014), and subsequent work has examined variations of RLSVI and their performance with both linear and nonlinear function approximation (Osband et al., 2016b,a; Osband, 2016). RLSVI does not explicitly maintain and update belief distributions and does not optimally synthesize information, as a coherent Bayesian method would. Regardless, as we will later establish through computational and mathematical analyses, RLSVI can offer an effective approach to deep exploration.

5.3.1. Randomization via Gaussian noise

We first consider a version of RLSVI that induces exploration through injecting Gaussian noise into calculations of the form carried out by LSVI. To understand the role of this noise, let us first consider a conventional linear regression problem. Suppose we wish to estimate a parameter vector $\theta \in \mathbb{R}^D$, with $N(\overline{\theta}, \lambda I)$ prior and data $\mathcal{D} = ((x_n, y_n) : n = 1,\dots,N)$, where each "feature vector" x_n is a row vector with K components and each "target value" y_n is scalar. Given the parameter vector θ and feature vector x_n, the target y_n is generated

according to $y_n = x_n\theta + w_n$, where w_n is independently drawn from $N(0, v)$. Conditioned on $\mathcal{D}$, θ is Gaussian with

$$\mathbb{E}[\theta|\mathcal{D}] = \operatorname*{argmin}_{\theta\in\mathbb{R}^D}\left(\frac{1}{v}\sum_{n=1}^{N}(y_n - x_n\theta)^2 + \frac{1}{\lambda}\|\overline{\theta}-\theta\|^2\right) = \left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1}\left(\frac{1}{v}X^\top y + \frac{1}{\lambda}\overline{\theta}\right) \tag{5.3}$$

and

$$\mathrm{Cov}[\theta|\mathcal{D}] = \left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1},$$

where $X \in \mathbb{R}^{N\times D}$ is a matrix whose nth row is x_n and $y \in \mathbb{R}^n$ is a vector whose nth component is y_n.

One way of generating a random sample $\tilde{\theta}$ with the same conditional distribution as θ is simply to sample from $\tilde{\theta} \sim N(\mathbb{E}[\theta|\mathcal{D}], \mathrm{Cov}[\theta|\mathcal{D}])$. An alternative construction is given by

$$\tilde{\theta} \leftarrow \operatorname*{argmin}_{\theta\in\mathbb{R}^D}\left(\frac{1}{v}\sum_{n=1}^{N}(y_n + z_n - x_n\theta)^2 + \frac{1}{\lambda}\|\hat{\theta}-\theta\|^2\right) = \left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1}\left(\frac{1}{v}X^\top(y+z) + \frac{1}{\lambda}\hat{\theta}\right), \tag{5.4}$$

where $\hat{\theta} \sim N(\overline{\theta}, \lambda I)$ and $z_n \sim N(0, v)$ are sampled independently. To see why this $\tilde{\theta}$ takes on the same distribution, first note that $\tilde{\theta}$ is Gaussian, since it is affine in $\overline{\theta}$ and z. Further, $\tilde{\theta}$ exhibits the appropriate mean and covariance matrix, since

$$\mathbb{E}[\tilde{\theta}|\mathcal{D}] = \left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1}\left(\frac{1}{v}X^\top(y + \mathbb{E}[z|\mathcal{D}]) + \frac{1}{\lambda}\mathbb{E}[\hat{\theta}|\mathcal{D}]\right) = \mathbb{E}[\theta|\mathcal{D}],$$

and

$$\begin{aligned}
Cov[\tilde{\theta}|\mathcal{D}] &= \left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1}\left(\frac{1}{v^2}X^\top\mathbb{E}[zz^\top|\mathcal{D}]X + \frac{1}{\lambda^2}\mathbb{E}[\hat{\theta}\hat{\theta}^\top|\mathcal{D}]\right)\left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1} \\
&= \left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1}\left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)\left(\frac{1}{v}X^\top X + \frac{1}{\lambda}I\right)^{-1} \\
&= \mathrm{Cov}[\theta|\mathcal{D}].
\end{aligned}$$

Equation (5.4) is signficant, since it allows us to understand Bayesian linear regression through a purely computational perspective. For the linear setting, we see that training a least-squares solution on perturbed versions of the data is equivalent to conjugate Bayesian posterior. This suggests that, in order to generate approximate posterior samples for Q^* we can replace the least-square computation of Algorithm 3 with an alternative value iteration that trains on randomly perturbed versions of the data. We call this algorithm *randomized least-squares value iteration*, which we now outline.

Algorithm 4 `learn_rlsvi`

Agent:	$\mathcal{L}(\theta=\cdot;\theta^-=\cdot,\mathcal{D}=\cdot)$	TD error loss function
	$\mathcal{R}(\theta=\cdot;\theta^p=\cdot)$	regularization function
	`buffer`	memory buffer of observations
	`prior`	prior distribution of parameters
	$H \in \mathbb{N}$	planning horizon
Updates:	$\tilde{\theta}$	agent value function estimate

$\tilde{\theta}_0 \leftarrow$ **null**
Data $\tilde{\mathcal{D}} \leftarrow$ `buffer.sample_perturbed_data()`
Prior parameter $\tilde{\theta}^p \leftarrow$ `prior.sample()`
for h in $(0, \ldots, H-1)$ **do**
  $\tilde{\theta}_{h+1} \leftarrow \operatorname*{argmin}_{\theta\in\mathbb{R}^D}\left(\mathcal{L}(\theta;\tilde{\theta}_h,\tilde{\mathcal{D}}) + \mathcal{R}(\theta;\tilde{\theta}^p)\right)$
update value function estimate $\tilde{\theta} \leftarrow \tilde{\theta}_H$

Note that `learn_rlsvi` is identical to `learn_lsvi` except that the optimization happens over randomized versions of the underlying data and prior. For correspondence with the Gaussian derivation above we would implement:

$$\texttt{buffer.sample_perturbed_data()} := [(s_t, a_t, r_t + z_t, s'_t, t)\ \forall t \in \texttt{buffer},\ z_t \sim N(0, v)]. \tag{5.5}$$

We call this version of RLSVI with additive Gaussian noise `learn_grlsvi`, indicating the specific choice of data randomization and prior $\theta \sim N(0, \lambda I)$. In Section 6 we prove that this method recovers a polynomial regret bound when used with linear value functions and tabular representation. This is significant, because LSVI with any dithering action selection scheme can not recover such a bound (Section 4). Before we jump to analysis, we provide some intuition for how this simple modification can lead to deep exploration.

5.3.2. How does RLSVI drive deep exploration?

To understand the role of injected noise and how this gives rise to deep exploration, let us discuss a simple example, involving an MDP $\mathcal{M}$ with four states $\mathcal{S} = \{1, 2, 3, 4\}$ and two actions $\mathcal{A} = \{up, down\}$. Let $\mathcal{H}$ be a list of all transitions we have observed, and partition this into sublists $\mathcal{H}_{s,a} = ((\tilde{s}, \tilde{a}, r, s') \in \mathcal{H} : (\tilde{s}, \tilde{a}) = (s, a))$, each containing transitions from a distinct state-action pair. Suppose that $|\mathcal{H}_{(4,down)}| = 1$, while for each state-action pair $(s, a) \neq (4, down)$, $|\mathcal{H}_{s,a}|$ is virtually infinite. Hence, we are highly uncertain about the expected immediate rewards and transition probabilities at $(4, down)$ but can infer these quantities with extreme precision for every other state-action pair.

Given our uncertainty about $\mathcal{M}$, Q^*_H for each planning horizon H is a random variable. Figure 2 illustrates our uncertainty in these values. Each larger triangle represents a pair (s, h), where h is the horizon index. Note that these triangles represent possible future states, and h represents the number of periods between a visit to the state and the end of the planning horizon. Each of these larger triangles is divided into two smaller ones, associated with *up* and *down* actions. The dotted lines indicate plausible transitions, except at $(4, down)$, where we are highly uncertain and any transition is plausible. The shade of

each smaller triangle represents our degree of uncertainty in the value of $Q_h^*(s,a)$. To be more concrete, take our measure of uncertainty to be the variance of $Q_h^*(s,a)$.

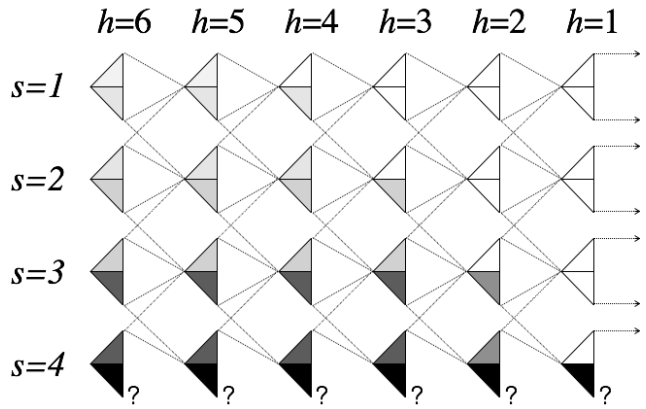


Figure 2: Illustration of how `learn_grlsvi` achieves deep exploration.

For the case of $h = 1$, only immediate rewards influence Q_1^*, and as such we are only uncertain about $Q_1^*(4, down)$. Stepping back to $h = 2$, in addition to being highly uncertain about $Q_2^*(4, down)$, we are somewhat uncertain about $Q_2^*(4, up)$ and $Q_2^*(3, down)$, since these pairs can transition to $(4, down)$ and be exposed to the uncertainty associated with that state-action pair. We are not as uncertain about $Q_2^*(4, up)$ and $Q_2^*(3, down)$ as we are about $Q_2^*(4, down)$ because from $(4, up)$ and $(3, down)$, there is reasonable chance that we will never see $(4, down)$. Continuing to work our way leftward in the diagram, it is easy to visualize how uncertainty propagates as h increases.

Let us now turn our attention to the variance of samples $\mathcal{Q}_{\tilde{\theta}_H}(s,a)$ generated by `learn_grlsvi`, which, for reasons we will explain, tend to grow and shrink with the variance of $Q_H^*(s,a)$. To keep things simple, assume $\lambda = \infty$ and that we use an exhaustive – or "tabular" – representation of value functions. In particular, each component of the parameter vector $\theta \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ encodes the value $\mathcal{Q}_\theta(s,a)$ of a single state-action pair. This parameterized value function can represent any mapping from $\mathcal{S} \times \mathcal{A}$ to $\mathbb{R}$.

Under our simplifying assumptions, it is easy to show that

$$\mathcal{Q}_{\tilde{\theta}_{h+1}}(s,a) = \frac{1}{|\tilde{\mathcal{H}}_{s,a}|} \sum_{(\tilde{s},\tilde{a},r,s',z)\in\tilde{\mathcal{H}}_{s,a}} \left(r + \max_{a'\in\mathcal{A}} \mathcal{Q}_{\tilde{\theta}_h}(s',a') + z \right).$$

The right-hand-side is an average of target values. Recall that, for any $(s,a) \neq (4, down)$, $|\tilde{\mathcal{H}}_{s,a}|$ is so large that any sample average is extremely accurate, and therefore, $\mathcal{Q}_{\tilde{\theta}_{h+1}}(s,a)$ is essentially equal to $\mathbb{E}_{\mathcal{M}}[r_{t+1} + \max_{\alpha\in\mathcal{A}} \mathcal{Q}_{\tilde{\theta}_h}(s_{t+1},\alpha)|s_t = s, a_t = a]$. For the distinguished case of $(4, down)$, $|\tilde{\mathcal{H}}_{4,down}| = 1$, and the average target value may therefore differ substantially from its expectation $\mathbb{E}[r + \max_{a'\in\mathcal{A}} \mathcal{Q}_{\tilde{\theta}_h}(s',a')|\tilde{\theta}_h, \mathcal{M}]$. Notably, the noise term z does not

average out as it does for other state-action pairs and should contribute variance v to the sample $\mathcal{Q}_{\tilde{\theta}_{h+1}}(4, down)$.

Based on this observation, for the case of $h = 1$, for $(s,a) \neq (4, down)$, $\mathcal{Q}_{\tilde{\theta}_1}(s,a)$ is virtually equal to Q_1^*, while for $\mathcal{Q}_{\tilde{\theta}_1}(4, down)$ exhibits variance of at least v. For $h = 2$, $\mathcal{Q}_{\tilde{\theta}_2}(4, down)$ again exhibits variance of at least v, but unlike the case of $h = 1$, $\mathcal{Q}_{\tilde{\theta}_2}(4, up)$ and $\mathcal{Q}_{\tilde{\theta}_2}(3, down)$ also exhibit non-negligible variance since these pairs can transition to $(4, down)$ and therefore depend on the noise-corrupted realization of $\mathcal{Q}_{\tilde{\theta}_1}(4, down)$. Working leftward through Figure 2, we can see that noise propagates and influences value estimates in a manner captured by the shading in the figure. Hence, samples $\mathcal{Q}_{\tilde{\theta}_h}(s,a)$ exhibit high variance where the variance of $Q_h^*(s,a)$ is large.

This relationship drives deep exploration. In particular, a high variance sample $\mathcal{Q}_{\tilde{\theta}_H}(s,a)$ will be overly optimistic in some episodes. Over such episodes, the agent will be incentivized to try the associated action. This is appropriate because the agent is uncertain about the optimal value $Q_H^*(s,a)$ over the planning horizon. Note that this incentive is not only driven by uncertainty concerning the immediate reward and transition. As illustrated in Figure 2, uncertainty propagates to offer incentives for the agent to pursue information even if it will require multiple time periods to arrive at an informative observation. This is the essence of deep exploration.

It is worth commenting on a couple subtle properties of `learn_grlsvi`. First, given θ_h and $\mathcal{H}$, θ_{h+1} is sampled from a Gaussian distribution. However, given the inputs to `learn_grslvi`, the output $\tilde{\theta}$ is not Gaussian. This is because at each step θ_{h+1} depends nonlinearly on θ_h due to the maximization over actions in the TD loss (5.1). Second, it is important that `learn_grlsvi` uses the same noise samples z in across iterations of the for loop in line 5. To see why, suppose `learn_grlsvi` used independent noise samples in each iteration. Then, when applied to the example of Figure 2, in some iterations, we would be optimistic about the reward at $(4, down)$, while in other iterations, we would be pessimistic about that. Now consider a sample $\mathcal{Q}_{\tilde{\theta}_H}(1, up)$ for large H. This sample would be perturbed by a combination of optimistic and pessimistic noise terms influencing assessments at $(4, down)$ to the right. The resulting averaging effect could erode the chances that $\mathcal{Q}_{\tilde{\theta}_H}(1, up)$ is optimistic.

5.3.3. Randomization via statistical bootstrap

With `learn_grlsvi`, the Gaussian distribution of noise serves as a coarse model of errors between targets $r + \max_{a'\in\mathcal{A}} \mathcal{Q}_{\tilde{\theta}_h}(s',a')$ and expectations $\mathbb{E}[r + \max_{a'\in\mathcal{A}} \mathcal{Q}_{\tilde{\theta}_h}(s',a')|\tilde{\theta}_h, \mathcal{M}]$. The statistical bootstrap[4] offers an alternative approach to randomization which may often more accurately capture characteristics of the generating process. In its classic form, the bootstrap takes a dataset $\mathcal{D}$ of size N and generates a sampled dataset $\tilde{\mathcal{D}}$ also of size N drawn uniformly with replacement from $\mathcal{D}$ (Efron and Tibshirani, 1994). The bootstrap serves as a form of data-based simulation and, in certain cases, recovers strong convergence guarantees (Bickel and Freedman, 1981; Fushiki, 2005).

4. We are overloading the term *bootstrap* here. In reinforcement learning, bootstrapping is commonly used to refer to the calculation of a state-action value estimate based on value estimates at states to which the agent may transition. Here, we refer to the *statistical* bootstrap of data-based simulation. The most common form of statistical bootstrap uses the sample data as an approximation to its generating distribution (Efron, 1982).

`learn_brlsvi` is a version of RLSVI that makes use of the bootstrap in place of additive Gaussian noise. This algorithm follows `learn_rlsvi` (Algorithm 4) and implements

$$\texttt{buffer.sample_perturbed_data()} := \texttt{bootstrap_sample(buffer.data())}. \tag{5.6}$$

Bootstrap sampling for value function randomization may present several benefits over additive Gaussian noise. First, most bootstrap resampling schemes do not require a 'noise variance' as input, which simplifies the algorithm from a user perspective. Related to this point, the bootstrap can effectively induce a state-dependent and heteroskedastic randomization which may be more appropriate in complex environments. More generally, we can consider bootstrapped RLSVI as a non-parameteric randomization for the value function estimate and this opens a wide range of potential bootstrap variants and prior mechanisms that could be employed with RLSVI (Osband and Van Roy, 2015).

Eckles and Kaptein (2019) were the first to propose using bootstrap samples as an approximation to the posterior samples used in the Thompson sampling algorithm. Unfortunately, bootstrapping does not provide meaningful uncertainty estimates in early periods. If applied without modification, the algorithms in Eckles and Kaptein (2019) incur regret that scales linearly in the time horizon. Value function estimates in `learn_brlsvi` are instead randomized not just through bootstrap sampling, but through regularizing toward a random prior sample. The effect of the random prior sample vanishes as diverse data is collected, but it is critical to driving exploration in early periods.

5.4. Practical variants of RLSVI

In this section, we will present variants of RLSVI designed to address the important practical considerations of computational efficiency and robustness to mis-specification of the parameterized value function. There are many ideas in the reinforcement learning literature that can be brought to bear for these purposes, and we will by no means cover an exhaustive list. Rather, we will present a mix of ideas that lead to a particular algorithm that effectively addresses a broad range of complex problems. This algorithm also serves to illustrate the many degrees of freedom in mixing and matching ingredients from the reinforcement learning literature when randomized value functions are part of the recipe.

5.4.1. Finite buffer experience replay

The use of a buffer of past observations to fit a value function is sometimes referred to as *experience replay*. The algorithms we have presented so far use an infinite buffer and thus require memory and compute time that grow linearly in the number of observations. For complex problems that require substantial learning times, such a requirement becomes onerous. To overcome this, we can restrict the buffer to some finite size, treating it as a FIFO queue.

Computational requirements aside, there can be other substantial benefits to using a finite buffer. In particular, the agent may learn to make more effective decisions within fewer episodes (Adam et al., 2012). This is likely due to model mis-specification. In particular, if Q^* can not be represented by $\mathcal{Q}_\theta$ for any θ, it is helpful to restrict attention to the most relevant data when regressing, as this focusses on minimizing errors at relevant states and actions. Restricting the buffer to recent observations may serve as a reasonable heuristic

here. Recent work has also demonstrated benefit from more sophisticated prioritization of data for storage in a finite buffer (Schaul et al., 2015).

5.4.2. Discounted TD and incremental learning

Both LSVI and RLSVI take a planning horizon $H \in \mathbb{N}$ as an argument. These approaches can be wasteful in that they compute separate estimates $\tilde{\theta}_h$ for each $h = 1, .., H$. Further, these algorithms are "batch," in the sense that they require computation over all observed data at the start of each episode; this leads to computational costs that grow with time. In this subsection we introduce a discounted formulation that admits an incremental computational approach.

Let $\gamma \in (0, 1)$ be a discount factor that induces a time preference over future rewards. A discount γ approximates an effective planning horizon $H \simeq \frac{1}{1-\gamma}$, but affords a solution to the discounted Bellman equation $Q^*_\gamma(s, a) = \mathbb{E}_{\mathcal{M}}\left[R(s, a) + \gamma \max_{a'} Q^*_\gamma(s', a')\right]$ (Blackwell, 1965). Inspired by this relationship we define the γ-discounted empirical TD loss:

$$\mathcal{L}\gamma(\theta; \theta^-, \mathcal{D}) := \sum_{t \in \mathcal{D}} \left(r_t + \gamma \max_{a' \in \mathcal{A}} \mathcal{Q}_{\theta^-}(s'_t, a') - \mathcal{Q}_\theta(s_t, a_t) \right)^2. \tag{5.7}$$

Algorithm 5 (`learn_online_lsvi`) presents an incremental variant of LSVI. Rather than recompute $\tilde{\theta}$ from scratch each episode, `learn_online_lsvi` updates its previous estimate by gradient descent over a subset of the data. This algorithm is a form of temporal difference learning (Sutton, 1988) and, at a high level, this approach broadly describes famous approaches such as TD-gammon (Tesauro, 1995) and DQN (Mnih et al., 2015). For more background and discussion of this family of algorithms we refer to Sutton and Barto (2018).

Algorithm 5 `learn_online_lsvi`

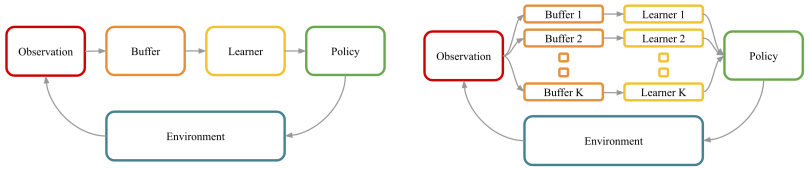
Agent:	$\tilde{\theta}$	parameter estimate
	$\tilde{\theta}^p$	prior mean of parameter estimate
	$\mathcal{L}_\gamma(\theta{=}\cdot; \theta^-{=}\cdot, \mathcal{D}{=}\cdot)$	TD error loss function
	$\mathcal{R}(\theta{=}\cdot; \theta^p{=}\cdot)$	regularization function
	`buffer`	memory buffer of observations
	α	learning rate
Updates:	$\tilde{\theta}$	agent value function estimate

1: Data $\tilde{\mathcal{D}} \leftarrow$ `buffer.sample_minibatch()`
2: $\delta \leftarrow$ `buffer.minibatch_size` / `buffer.size`
3: $\tilde{\theta} \leftarrow \tilde{\theta} - \alpha \nabla_{\theta|\theta=\tilde{\theta}} \left(\mathcal{L}_\gamma(\theta; \tilde{\theta}, \tilde{\mathcal{D}}) + \delta \mathcal{R}(\theta; \tilde{\theta}^p) \right)$

5.4.3. Randomization via ensemble sampling

Algorithm 5 (`learn_online_lsvi`) presents an incremental version of LSVI. However, RLSVI in its purest form calls for an estimate trained on randomly perturbed data at the beginning of each episode $l = 1, 2, ..$; this is not immediately amenable to such an incremental algorithm. Instead, our solution approximates the effects of RLSVI in an online algorithm via *ensemble sampling*.

Ensemble sampling approximates the distribution induced by RLSVI through an ensemble of $K \in \mathbb{N}$ estimates trained in parallel $(\tilde{\theta}_1, .., \tilde{\theta}_K)$ (Osband and Van Roy, 2015; Lu and Van Roy, 2017). At the start of any episode l, we can approximate the distribution of $\tilde{\theta}$ under RLSVI via a sample of $\tilde{\theta}_k$ for $k \sim \text{Unif}(1, .., K)$. Figure 3 presents an illustration of such a system.



(a) learning a single value function (b) learning multiple value functions in parallel

Figure 3: RLSVI via ensemble sampling, each member produced by LSVI on perturbed data.

One method to maintain K estimates in parallel is to implement an ensemble memory buffer `ensemble_buffer`. An ensemble buffer should function similarly to `buffer`, but maintain K distinct perturbations of the observed data $\tilde{\mathcal{D}}_k$, each associated with their appropriate ensemble value estimate $\tilde{\theta}_k$. We use a Pythonic notation that `ensemble_buffer[k]` is a `buffer` for each `k = 1, .., K`; although in a practical implementation it will be sensible to share appropriate parts of the memory requirements. For an online variant of `learn_grlsvi` (additive Gaussian noise) we can store K distinct samples of additive noise, as described in Algorithm 6. Similarly, for an online approximation to bootstrap sampling we might use "double-or-nothing" sampling per Algorithm 7 (Owen and Eckles, 2012).

Algorithm 8 (`learn_ensemble_rlsvi`) presents the ensemble RLSVI algorithm. An agent that employs `learn_ensemble_rlsvi` maintains ensemble parameter estimates $\tilde{\theta}_1, .., \tilde{\theta}_K$ with random prior samples $\tilde{\theta}^p_1, .., \tilde{\theta}^p_K$. The regulatization effects of $\mathcal{R}(\theta; \tilde{\theta}^p_k)$ can play an important role in exploration. Although we have suggested a particular form in (5.2), we would consider alternative approaches based on prior observations as studied in *off-policy learning* (Precup et al., 2001) and *transfer learning* (Taylor and Stone, 2009).

We note that, when used off-policy, minimizing the TD loss may lead to unstable learning and even cause the value function estimate to diverge (Tsitsiklis and Van Roy, 1997). This instability can be exacerbated by algorithms such as `learn_ensemble_rlsvi`, which may lead to value estimates computed from data more off-policy than a single value estimate. In this context, it may be beneficial to replace the naive TD loss with an alternative designed for off-policy learning (Sutton et al., 2009; Munos et al., 2016).

Algorithm 6 `ensemble_buffer.update_gaussian_noise(·)`

Input:	`transition`	(s_t, a_t, r_t, s'_t, t)
Agent:	v	noise variance
Updates:	`ensemble_buffer`	replay buffer of K-parallel perturbed data

for k in $(1, \ldots, K)$ **do**
  `ensemble_buffer[k].enqueue(`$(s_t, a_t, r_t + z^k_t, s'_t, t)$`)` where $z^k_t \sim N(0, v)$

Algorithm 7 `ensemble_buffer.update_bootstrap(·)`

Input:	`transition`	(s_t, a_t, r_t, s'_t, t)
Updates:	`ensemble_buffer`	replay buffer of K-parallel perturbed data

for k in $(1, \ldots, K)$ **do**
  if $m^k_t \sim \text{Unif}(\{0, 1\}) = 1$ **then**
    `ensemble_buffer[k].enqueue(`(s_t, a_t, r_t, s'_t, t)`)`

Algorithm 8 `learn_ensemble_rlsvi`

Agent:	$\tilde{\theta}_1, .., \tilde{\theta}_K$	ensemble parameter estimates
	$\tilde{\theta}^p_1, .., \tilde{\theta}^p_K$	prior samples of parameter estimates
	$\mathcal{L}_\gamma(\theta = \cdot; \theta^- = \cdot, \mathcal{D} = \cdot)$	TD error loss function
	$\mathcal{R}(\theta = \cdot; \theta^p = \cdot)$	regularization function
	`ensemble_buffer`	replay buffer of K-parallel perturbed data
	α	Learning rate
Updates:	$\tilde{\theta}$	agent value function estimate

for k in $(1, \ldots, K)$ **do**
  Data $\tilde{\mathcal{D}}_k \leftarrow$ `ensemble_buffer[k].sample_minibatch()`
  $\delta \leftarrow$ `buffer.minibatch_size / buffer.size`
  $\tilde{\theta}_k \leftarrow \tilde{\theta}_k - \alpha \nabla_{\theta | \theta = \tilde{\theta}_k} \left(\mathcal{L}_\gamma(\theta; \tilde{\theta}_k, \tilde{\mathcal{D}}_k) + \mathcal{R}(\theta; \tilde{\theta}^p_k) \right)$
update $\tilde{\theta} \leftarrow \tilde{\theta}_j$ for $j \sim \text{Unif}(1, .., K)$

6. Regret bound

This section provides a regret analysis of RLSVI in a particularly simple type of decision problem (Section 3). We consider an RLSVI `agent` with an infinite buffer, greedy actions and `learn_grlsvi` (Algorithm 4 with additive Gaussian noise (5.5)) and a tabular representation.[5] The bound we establish applies to a tabular time-inhomogeneous MDP with transition kernel drawn from a Dirichlet prior. This stylized setting provides rigorous confirmation that RLSVI is capable of performing provably efficient deep exploration in tabular environments. In addition, we hope this analysis provides a framework for establishing more general guarantees – for example those applying to RLSVI with linearly parameterized value functions. Several intermediate lemmas used in the analysis hold under much less restrictive assumptions, and could be useful beyond the setting studied here.

6.1. Formulation of a time-inhomogenous MDP

We consider a class of *finite-horizon time-inhomogeneous MDPs*. This can be formulated as a special case the paper's general formulation as follows. Assume the state space factorizes as $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{H-1}$ where the state always advances from some state $s_t \in \mathcal{S}_t$ to $s_{t+1} \in \mathcal{S}_{t+1}$ and the process terminates with probability 1 in period H. For notational convenience, we assume each set $\mathcal{S}_0, ..., \mathcal{S}_{H-1}$ contains an equal number of elements. This

5. A tabular representation for RLSVI means that $\mathcal{Q}_\theta = \theta \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ and $\mathcal{Q}_\theta(s, a) = \theta_{s,a}$.

is stated formally in the next assumption, which is maintained for all statements in this section.

Assumption 2 (Finite-horizon time-inhomogeneous MDP)
The state space factorizes as $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{H-1}$ where $|\mathcal{S}_0| = \cdots = |\mathcal{S}_{H-1}| < \infty$. For any MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho)$,

$$\sum_{s' \in \mathcal{S}_{t+1}} \mathcal{P}_{s,a}(s') = 1 \qquad \forall t \in \{0, ..., H-2\},\ s \in \mathcal{S}_t,\ a \in \mathcal{A},$$

and

$$\sum_{s' \in \mathcal{S}} \mathcal{P}_{s,a}(s') = 0 \qquad \forall s \in \mathcal{S}_{H-1},\ a \in \mathcal{A}.$$

Each state $s \in \mathcal{S}_t$ can be written as a pair $s = (t, x)$ where $t \in \{0, ..., H-1\}$ and $x \in \mathcal{X} = \{1, ..., |\mathcal{S}_0|\}$. Similarly, a policy $\pi : \mathcal{S} \to \mathcal{A}$ can be viewed as a sequence $\pi = (\pi_0, ..., \pi_{H-1})$ where $\pi_t : x \mapsto \pi((t, x))$. Our notation can be specialized to this time-inhomogenous problem, writing transition probabilities as $\mathcal{P}_{t,x,a}(x') \equiv \mathcal{P}_{(t,x),a}((t+1, x'))$ and reward probabilities as $\mathcal{R}_{t,x,a,x'}(r) \equiv \mathcal{R}_{(t,x),a,(t+1,x')}(r)$. For consistency, we also use different notation for the optimal value function, writing

$$V^{\pi}_{\mathcal{M},t}(x) \equiv V^{\pi}_{\mathcal{M}}((t, x))$$

and define $V^{*}_{\mathcal{M},t}(x) := \max_{\pi} V^{\pi}_{\mathcal{M},t}(x)$. Similarly, we can define the state-action value function under the MDP at timestep $t \in \{0, ..., H-1\}$ by

$$Q^{*}_{\mathcal{M},t}(x, a) = \mathbb{E}[r_{t+1} + V^{*}_{\mathcal{M},t+1}(x_{t+1}) \mid \mathcal{M}, x_t = x, a_t = a] \qquad \forall x \in \mathcal{X}, a \in \mathcal{A}.$$

This is the expected reward accrued by taking action a in state x and proceeding optimally thereafter.

Upon choosing an action, the algorithm observes a pair $o = (x', r)$ consisting of a state transition and a reward. We will refer to this pair o as an *outcome* of the decision. Assumptions about the distribution of rewards and state-transitions can be more compactly written as an assumption about outcome distributions. We study the regret of RLSVI under the following Bayesian model for the MDP $\mathcal{M}$. This assumption is not required for some of the results in this section, and we will specify when it is needed.

Assumption 3 (Independent Dirichlet prior for outcomes)
Rewards take values in $\{0, 1\}$ and so the cardinality of the outcome space is $|\mathcal{X} \times \{0, 1\}| = 2|\mathcal{X}|$. For each, $(t, x, a) \in \{0, ..., H-2\} \times \mathcal{X} \times \mathcal{A}$, the outcome distribution is drawn from a Dirichlet prior

$$\mathcal{P}^{O}_{t,x,a}(\cdot) \sim \text{Dirichlet}(\alpha_{0,t,x,a})$$

for $\alpha_{0,t,x,a} \in \mathbb{R}^{2|\mathcal{X}|}_{+}$ and each $\mathcal{P}^{O}_{t,x,a}$ is drawn independently across (t, x, a). Assume there is $\beta \geq 3$ such that $\mathbb{1}^T \alpha_{0,t,a,x} = \beta$ for all (t, x, a).

6.2. Bayesian regret bound

The following theorem is the main result of this section, and establishes a polynomial bound on the Bayesian regret of RLSVI.

Theorem 1 (Bayesian regret bound for RLSVI)
Consider an RLSVI `agent` *with an infinite buffer, greedy actions and* `learn_grlsvi` *with tabular representation.*[6] *Under Assumption 3 with $\beta \geq 3$, if this version of RLSVI is applied with planning horizon H, and parameters $v = 3H^2$, $\bar{\theta} = H\mathbb{1}$ and $v/\lambda = \beta$, then for all $L \in \mathbb{N}$,*

$$\text{BayesRegret}(\text{RLSVI}_{\bar{\theta},v,\lambda}, L) \leq 6H^2\sqrt{\beta|\mathcal{X}||\mathcal{A}|L\log_{+}(1 + |\mathcal{X}||\mathcal{A}|HL)}\log_{+}\left(1 + \frac{L}{|\mathcal{X}||\mathcal{A}|}\right), \tag{6.1}$$

and

$$\begin{aligned}\text{BayesRegret}(\text{RLSVI}_{\bar{\theta},v,\lambda}, L) \quad \leq \quad & 5\beta H^3|\mathcal{X}||\mathcal{A}|\sqrt{\log_{+}(1 + |\mathcal{X}||\mathcal{A}|HL)}\log_{+}\left(1 + \frac{L}{|\mathcal{X}||\mathcal{A}|}\right) \\ & + 2H^2\sqrt{6|\mathcal{X}||\mathcal{A}|L\log(|\mathcal{X}||\mathcal{A}|)}\end{aligned} \tag{6.2}$$

where $\log_{+}(x) = \max\{1, \log(x)\}$.

Let us focus on the first bound given in equation (6.1). The parameter β governs the relative strength of prior mean $\bar{\theta}$ in the Q-functions sampled by RLSVI. We typically think of β as a constant, reflecting situations with weak prior knowledge of the optimal value function that does not grow with variables H, S, A, L. In this case, this regret bound is $\tilde{O}(H^2\sqrt{|\mathcal{X}||\mathcal{A}|L})$ where $\tilde{O}$ ignores poly-logarithmic factors. Note that since $|\mathcal{S}_0| = ... = |\mathcal{S}_{H-1}| = |\mathcal{X}|$ then $|\mathcal{S}| = |\mathcal{X}|H$ and for $T = LH$ denoting the number of periods,

$$\text{BayesRegret}(\text{RLSVI}_{\bar{\theta},v,\lambda}, L) = \tilde{O}(H\sqrt{|\mathcal{S}||\mathcal{A}|T}).$$

This bound reveals that RLSVI requires a number of episodes that is just linear in the number of states to reach near optimal performance. Indeed, it is possible to guarantee cumulative Bayesian regret less than $L\epsilon$ with a value of L that scales with $|\mathcal{X}|/\epsilon^2$. In general, at least order $|\mathcal{X}|^2$ samples are required to learn the transition kernel $\mathcal{P}_{t,x,a}$. Therefore, for large $|\mathcal{X}|$ we prove that RLSVI learns to make near-optimal decisions using fewer samples than would be required to learn the transition dynamics of the MDP.

It is interesting to compare this Bayesian regret bound with bounds that have been established for other tabular reinforcement learning algorithms. The results of Bartlett and Tewari (2009) and Jaksch et al. (2010) are not directly comparable to the bound established for RLSVI, as they develop bounds on minimax, rather than Bayesian regret, and study classes of MDPs satisfying recurrence assumptions, rather than episodic MDPs. However, it is worth noting that because these algorithms attempt to represent each transition probability $\mathcal{P}_{t,x,a}(x')$ accurately, applying their analysis to our problem yields a regret bound of $\tilde{O}(H^2|\mathcal{X}|\sqrt{|\mathcal{A}|L})$, which has is larger dependence on $|\mathcal{X}|$.

The second bound given in equation (6.2) reveals the dependence of regret on β more precisely. This bound is $\tilde{O}(\beta H^3|\mathcal{X}||\mathcal{A}| + H^2\sqrt{|\mathcal{X}||\mathcal{A}|L})$. The first term in this regret bound

6. `learn_grlsvi` is `learn_rlsvi` (Algorithm 4) with additive Gaussian reward noise of variance v (5.5). A tabular representation for RLSVI means that $\mathcal{Q}_{\theta} = \theta \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ and $\mathcal{Q}_{\theta}(s, a) = \theta_{s,a}$.

can roughly be thought of as a bound on the regret incurred throughout an initial phase of the algorithm, during which it gathers data that overwhelms the prior mean. When the number of episodes L is large, the dominant term is the second one, which is $\tilde{O}(H^2\sqrt{|\mathcal{X}||\mathcal{A}|L})$ and has no dependence on β.

6.3. Stochastic Bellman operators

Any state-action value function $Q \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ induces a value function $V(x) = \max_{a\in\mathcal{A}} Q(x,a)$ that maps each state to a real number. To simplify the analysis, it is useful to introduce nonstandard notation for the value function over outcomes $o = (r,x)$.

Definition 2 (Induced value function)
For a state-action value function $Q \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ define the corresponding value function $V_Q \in \mathbb{R}^{2|\mathcal{X}|}$ over outcomes by $V_Q(r,x') := r + \max_{a\in\mathcal{A}} Q(x',a)$ for all $x' \in \mathcal{X}$ and $r \in \{0,1\}$.

It is useful to also keep notation for the empirical distribution over observed outcomes. Let

$$D_{\ell-1}(t,x,a) = \{(r^k_{t+1}, x^k_{t+1}) : k < \ell, x^k_t = x, a^k_t = a\}$$

be the set of data observed up to episode ℓ when action a was chosen in (t,x), and set $n_\ell(t,x,a) = |D_{\ell-1}(t,x,a)|$ to be number of past observations of the triple (t,x,a). For ease of notation we will write y for the timestep, state, and action $y := (t,x,a)$. Denote by $\hat{P}^O_{\ell,y}(r',x')$ the empirical distribution over outcomes (r',x') in the dataset $D_{\ell-1}(y)$.

This section introduces the Bellman operator underlying the MDP $\mathcal{M}$ and a notion of a Bellman operator that underlies the recursion defining RLSVI. Due to the randomness in $\mathcal{M}$ under Assumption 3 and the Gaussian noise added by RLSVI iterations both of these can be viewed as *stochastic Bellman operators*, as applying one of these operators to a state action value function $Q \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ generates a random state-action value function as output.

True Bellman Operator. For $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ the true Bellman operator at timestep t applied to Q is defined by

$$\begin{aligned} F_{\mathcal{M},t}Q(x,a) &= \mathbb{E}[r_{t+1} + \max_{a'\in\mathcal{A}} Q(x_{t+1},a') \mid \mathcal{M}, x_t = x, a_t = a] \\ &= \mathbb{E}[V_Q(r_{t+1}, x_{t+1}) \mid \mathcal{M}, x_t = x, a_t = a] \\ &= V_Q^T \mathcal{P}^O_{t,x,a}. \end{aligned}$$

Applying $F_{\mathcal{M},t}$ backward in time produces a sequence of optimal state-action value functions satisfying $Q^*_{\mathcal{M},H} = 0$ and the Bellman equation $Q^*_{\mathcal{M},t} = F_{\mathcal{M},t}Q^*_{\mathcal{M},t+1}$ for $t < H$. Under Assumption 3, this can be viewed as a randomized Bellman operator due to the randomness in the MDP $\mathcal{M}$.

Under Assumption 3, the posterior transition probabilities are distributed as

$$\mathcal{P}^O_y(\cdot)|\mathcal{H}_{\ell-1} \sim \text{Dirichlet}(\alpha_{\ell,y})$$

where

$$\alpha_{\ell,y} = \alpha_{0,y} + n_\ell(y)\hat{P}^O_{\ell,y} \in \mathbb{R}^{2|\mathcal{X}|} \tag{6.3}$$

for any triple $y = (t,x,a)$. These determine the posterior mean of $\mathcal{P}^O_y$ as a weighted linear combination of the prior and the empirical observations:

$$\mathbb{E}[\mathcal{P}^O_y \mid \mathcal{H}_{\ell-1}] = \frac{\alpha_{0,y} + n_\ell(y)\hat{P}^O_{\ell,y}}{\beta + n_\ell(y)}.$$

Bellman operator of RLSVI. In episode ℓ, we can define a notion of a Bellman operator underlying the recursion of RLSVI. Define

$$F_{\ell,t}Q(x,a) := \sigma^2_\ell(t,x,a)\left(\frac{\bar{\theta}_{t,x,a}}{\lambda} + \frac{1}{v}\left(\sum_{\substack{(r,x')\in \\ D_{\ell-1}(t,x,a)}} r + \max_{a'\in\mathcal{A}} Q(x',a')\right)\right) + w_\ell(t,x,a)$$

$$\sigma^2_\ell(t,x,a) = \left(\frac{1}{\lambda} + \frac{n_\ell(t,x,a)}{v}\right)^{-1} = \frac{v}{n_\ell(t,x,a) + v/\lambda}$$

$$w_\ell(t,x,a) \mid \mathcal{H}_{\ell-1} \sim N(0, \sigma^2_\ell(t,x,a))$$

where $w_\ell(y)/\sigma_\ell(y) \sim N(0,1)$ is drawn independently across episodes ℓ and triples $y = (t,x,a)$.

In episode ℓ RLSVI generates a sequence of state-action value functions $Q_{\ell,1}, ..., Q_{\ell,H}$ where $Q_{\ell,H} = 0 \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ consists of all zeros and for all $t < H$, $Q_{\ell,t} = F_{\ell,t}Q_{\ell,t+1}$. RLSVI chooses actions greedily with respect to this sequence of state-action value functions. Note that because of the Gaussian sampling noise, the action $\arg\max_{a\in\mathcal{A}} Q_\ell(x,a)$ is unique with probability one for any x and ℓ. Therefore the policy applied by RLSVI in an episode is completely determined by the state-action value functions it samples.

We can also express the RLSVI Bellman update in a simple way using the empirical distribution $\hat{P}^O_{\ell,y}$ over past outcomes resulting from $y = (t,x,a)$. We have

$$\sum_{(r,x')\in D_{\ell-1}(y)} \left(r + \max_{a'\in\mathcal{A}} Q(x',a')\right) = n_\ell(y) V_Q^T \hat{P}^O_{\ell,y}.$$

Direct calculation gives the following alternate expression

$$F_{\ell,t}Q(x,a) = \frac{(v/\lambda)\bar{\theta} + n_\ell(y)V_Q^T\hat{\mathcal{P}}^O_{\ell,y}}{(v/\lambda) + n_\ell(y)} + w_\ell(y) \qquad \forall y = (t,x,a). \tag{6.4}$$

This shows that the Bellman update of RLSVI differs from the empirical Bellman update $V_Q^T\hat{\mathcal{P}}^O_{\ell,y}$ in two ways: there is slight regularization toward the prior mean $\bar{\theta}$, and more importantly, RLSVI adds independent Gaussian noise to each update.

6.4. Optimism and regret decompositions

The next lemma forms a crucial element of the proof.

Lemma 3 (Planning Error to On Policy Bellman Error)
Let $Q_0, Q_1, Q_2, ..., Q_H \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ be any sequence with $Q_H = 0 \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ and take $\pi = (\pi_0, \pi_1, ..., \pi_{H-1})$

to be a policy with $\pi_t(x) \in \arg\max_{a\in\mathcal{A}} Q_t(x,a)$ for all x. Then for any MDP $\mathcal{M}$ and initial state $x \in \mathcal{X}$,

$$Q_0(x,\pi_0(x)) - V^{\pi}_{\mathcal{M},0}(x) = \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=0}^{H-1}(Q_t - F_{\mathcal{M},t}Q_{t+1})(x_t,a_t) \mid x_0 = x\right].$$

Remark 4 *To interpret this lemma, consider an algorithm that generates a sequence of state-action value functions $Q_0,...,Q_H \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ and chooses actions greedily with respect to this sequence. We can interpret $Q_0(x,\pi_0(x))$ to be the algorithm's estimate of the value of following this greedy policy throughout the episode from a starting state x, while $V^{\pi}_{\mathcal{M},0}(x)$ denotes the true expected value. One can interpret $Q_t - F_{\mathcal{M}}Q_{t+1}$ as the error in Bellman's equation at stage t. The right hand side of of the equation in Lemma 3 measures Bellman error on policy, i.e. at the states and actions that the agent is expected to sample by following the policy throughout the episode. This lemma says that the prediction $Q_0(x,\pi_0(x))$ can be far from the true value function only when on policy Bellman error is large.*

Lemma 3 is a powerful tool for studying the regret of optimistic algorithms. The regret of the policy π in Lemma 3 incurred in a single episode can always be decomposed as

$$V^*_{\mathcal{M},0}(x) - V^{\pi}_{\mathcal{M},0}(x) = \left(\max_{a\in\mathcal{A}} Q^*_{\mathcal{M},0}(x,a) - \max_{a\in\mathcal{A}} Q_0(x,a)\right) + \left(\max_{a\in\mathcal{A}} Q_0(x,a) - V^{\pi}_{\mathcal{M},0}(x)\right),$$

where we have used the fact that $V^*_{\mathcal{M},0}(x) = \max_{a\in\mathcal{A}} Q^*_{\mathcal{M},0}(x,a)$. The second term in this decomposition can be rewritten using Lemma 3. In particular, for any sequence $Q_0,...,Q_H \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ with $Q_H = 0$ and policy $\pi = (\pi_0,...,\pi_{H-1})$ under which actions $\pi_t(x) = \arg\max_{a\in\mathcal{A}} Q_t(x,a)$ are chosen greedily with respect these Q–functions, regret can be decomposed as follows:

$$\begin{aligned} V^*_{\mathcal{M},0}(x) - V^{\pi}_{\mathcal{M},0}(x) = \quad & \max_{a\in\mathcal{A}} Q^*_{\mathcal{M},0}(x,a) - \max_{a\in\mathcal{A}} Q_0(x,a) && \text{(pessimism of } Q_0) \\ (6.5) \qquad + \quad & \mathbb{E}_{\mathcal{M},\pi}\left[\textstyle\sum_{t=0}^{H-1}(Q_t - F_{\mathcal{M},t}Q_{t+1})(x_t,a_t) \mid x_0 = x\right] && \text{(on policy Bellman error).} \end{aligned}$$

If the function Q_0 is optimistic at an initial state x, in the sense that $\max_a Q_0(x,a) \geq \max_a Q^*_{\mathcal{M},0}(x,a)$, then regret in the episode is bounded by on policy Bellman error under $(Q_0,...,Q_H)$.

One can apply this regret decomposition to study RLSVI by taking $(Q_0,...,Q_H)$ to be the sequence $(Q_{\ell,0},...,Q_{\ell,H})$ generated by RLSVI in some episode ℓ. On policy Bellman error can be simplified further by plugging in $Q_{\ell,t} = F_{\ell,t}Q_{\ell,t+1}$. The next corollary of Lemma 3 then follows by taking expectations on both sides of equation (6.5).

Corollary 5 (Optimistic regret bounds)
For any episode $\ell \in \mathbb{N}$, if

$$\mathbb{E}\left[\max_{a\in\mathcal{A}} Q_{\ell,0}(x^{\ell}_0,a)\right] \geq \mathbb{E}\left[\max_{a\in\mathcal{A}} Q^*_{\mathcal{M},0}(x^{\ell}_0,a)\right] \tag{6.6}$$

then

$$\mathbb{E}\left[V^*_{\mathcal{M},0}(x^{\ell}_0) - V^{\pi_\ell}_{\mathcal{M},0}(x^{\ell}_0)\right] \leq \mathbb{E}\left[\sum_{t=0}^{H}(F_{\ell,t}Q_{\ell,t+1} - F_{\mathcal{M},t}Q_{\ell,t+1})(x^{\ell}_t,a^{\ell}_t)\right].$$

Corollary 5 forms the core of our analysis. The next section establishes that (6.6) holds in every episode $\ell \in \mathbb{N}$. We then complete the proof by bounding the cumulative on policy Bellman error throughout L episodes.

6.5. Stochastic optimism

Our goal is to show equation 6.6 holds when RLSVI is applied with appropriate parameters. We will instead prove that under Assumption 3 the stronger condition that

$$\mathbb{E}\left[\max_{a\in\mathcal{A}} Q_{\ell,0}(x^{\ell}_0,a) \mid \mathcal{H}_{\ell-1}\right] \geq \mathbb{E}\left[\max_{a\in\mathcal{A}} Q^*_{\mathcal{M},0}(x^{\ell}_0,a) \mid \mathcal{H}_{\ell-1}\right]$$

holds for any history $\mathcal{H}_{\ell-1}$. By the tower property of conditional expectation, this clearly implies equation (6.6).

As highlighted in Subsection 6.3, both $Q_{\ell,0} = F_{\ell,0}\cdots F_{\ell,H-1}0$ and $Q^*_{\mathcal{M},0} = F_{\mathcal{M},0}\cdots F_{\mathcal{M},H-1}0$ are calculated through recursive backward application of stochastic Bellman operators. The distributions of $Q_{\ell,0}$ and $Q^*_{\mathcal{M},0}$ generated in this fashion is complicated and difficult to study directly. Instead, we study properties of the stochastic Bellman operators themselves. We establish a strong sense in which $F_{\ell,t}$ generates random Q-functions $F_{\ell,t}Q$ that are optimistic compared to those generated by applying $F_{\mathcal{M},t}$ to Q. We then show this optimism is preserved under recursive application of the stochastic Bellman operators, which will imply the optimism of the final iterate $Q_{\ell,0}$. This strong notion of optimism is defined below.

Definition 6 (Stochastic optimism)
A random variable X is stochastically optimistic with respect to another random variable Y, written $X \geq_{SO} Y$, if for all convex increasing functions $u : \mathbb{R} \to \mathbb{R}$

$$\mathbb{E}[u(X)] \geq \mathbb{E}[u(Y)]. \tag{6.7}$$

This definition closely mirrors that of "second order stochastic dominance", which is widely used in decision theory (Hadar and Russell, 1969). A random payout X is second order stochastically dominant with respect to Y if (6.7) holds for all *concave* increasing function u. This means that any rational *risk-averse* agent prefers X to Y, while $X \geq_{SO} Y$ implies that any rational *risk-loving* agent prefers X to Y. Intuitively, this requirement means that draws of X generate payouts that are larger and noisier than Y. Our goal then is to show if RLSVI is applied with appropriate parameters, it generates iterates that are larger and noisier than the true $Q-$*functions*.

Example 2 (Stochastic optimism in Gaussian random variables)
If $X \sim N(\mu_X,\sigma^2_X)$ and $Y \sim N(\mu_Y,\sigma^2_Y)$ then $X \geq_{SO} Y$ if and only if $\mu_X \geq \mu_Y$ and $\sigma^2_X \geq \sigma^2_Y$.

The following observation is key to our analysis.

Lemma 7 (Preservation of optimism under convex operations)
For any two collections $(X_1,...,X_n)$ and $(Y_1,...,Y_n)$ of independent random variables with $X_i \geq_{SO} Y_i$ for each $i \in \{1,...n\}$ and any convex increasing function $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(X_1,...,X_n) \geq_{SO} f(Y_1...,Y_n).$$

Two special cases of Lemma 7 imply that the partial ordering of stochastic optimism is preserved under convolution and maximization. In particular, for any independent random

variables (X, Y, Z) if $X \succeq_{SO} Y$ we can conclude[7] $X + Z \succeq_{SO} Y + Z$. For two pairs of independent random variables (X_1, X_2) and (Y_1, Y_2) with $X_1 \succeq_{SO} Y_1$ and $X_2 \succeq_{SO} Y_2$,

$$\max\{X_1, X_2\} \succeq_{SO} \max\{Y_1, Y_2\}.$$

This implies the following monotonicity property of the Bellman operator $F_{\ell,t}$ underlying RLSVI. This will later enable us to show that if initial iterates of RLSVI are stochastically optimistic, then this optimism is preserved under recursive application of the stochastic Bellman operators $F_{\ell,0} \cdots F_{\ell,H-1}$.

Lemma 8 (Monotonicity)
Fix two random Q functions $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$. Suppose that conditioned on $\mathcal{H}_{\ell-1}$, for each $i = 1, 2$ the entries of $Q_i(x, a)$ are drawn independently across x, a, and drawn independently of the RLSVI noise terms $w_\ell(t, x, a)$. Then

$$Q_1(x, a) \mid \mathcal{H}_{\ell-1} \succeq_{SO} Q_2(x, a) \mid \mathcal{H}_{\ell-1} \qquad \forall (x, a) \in \mathcal{X} \times \mathcal{A}$$

implies

$$F_{\ell,t} Q_1(x, a) \mid \mathcal{H}_{\ell-1} \succeq_{SO} F_{\ell,t} Q_2(x, a) \mid \mathcal{H}_{\ell-1} \qquad \forall (x, a) \in \mathcal{X} \times \mathcal{A},\ t \in \{0, ..., H-1\}.$$

Proof Conditioned on $\mathcal{H}_{\ell-1}$,

$$F_{\ell,t} Q(x, a) = \frac{\sigma_\ell^2(t, x, a) \bar{\theta}_{t,x,a}}{\lambda} + \frac{\sigma_\ell^2(t, x, a)}{v} \left(\sum_{\substack{(r,x') \in \\ D_{\ell-1}(t,x,a)}} r + \max_{a' \in \mathcal{A}} Q(x', a') \right) + w_\ell(t, x, a)$$

is a convex function of $(Q(x', a'))_{x' \in \mathcal{X}, a' \in \mathcal{A}}$ convolved with the independent noise term $w_\ell(t, x, a)$. The result therefore follows by Lemma 7. ■

Consider the random variable $Y = P^T V$ where $V \in \mathbb{R}^n$ and $P \sim \text{Dirichlet}(\alpha)$. Then, Y has mean $V^T \alpha / \mathbb{1}^T \alpha$. The size of its fluctuations depends on how concentrated P is around its mean, captured by the pseudocount $\mathbb{1}^T \alpha = \sum_{i=1}^n \alpha_i$, and spread of the elements in V, captured by $\text{Span}(V) \equiv \max_i V_i - \min_j V_j$. The next lemma shows that a Gaussian random variable large enough mean and variance is stochastically optimistic with respect to Y. This result is established in the appendix.

Lemma 9 (Gaussian vs Dirichlet optimism)
Let $Y = P^T V$ for $V \in \mathbb{R}^n$ fixed and $P \sim \text{Dirichlet}(\alpha)$ with $\alpha \in \mathbb{R}_+^n$ and $\sum_{i=1}^n \alpha_i \geq 3$. Let $X \sim N(\mu, \sigma^2)$ with $\mu \geq \frac{\sum_{i=1}^n \alpha_i V_i}{\sum_{i=1}^n \alpha_i}$, $\sigma^2 \geq 3 \left(\sum_{i=1}^n \alpha_i\right)^{-1} \text{Span}(V)^2$, then $X \succeq_{SO} Y$.

With Lemma 9 in place, we can now establish a sense in which the Bellman operator underlying RLSVI is stochastically optimistic relative to the true Bellman operator. Recall definition 2, which defines the value over outcomes (r, x') under Q by $V_Q(r, x') \equiv r + \max_{a' \in \mathcal{A}} Q(x', a')$.

7. This follows from Lemma 7 by looking at the pairs (X, Z) and (Y, Z) and taking $f : \mathbb{R}^2 \to \mathbb{R}$ to be $f(x_1, x_2) = x_1 + x_2$.

Lemma 10 (Stochastically optimistic operators)
Suppose Assumption 3 holds and RLSVI is applied with parameters $(\bar{\theta}, v, \lambda)$ satisfying $(v/\lambda) = \beta$. Then for any episode ℓ with history $\mathcal{H}_{\ell-1}$, time $t \in \{0, ..., H-1\}$, and pair $(x, a) \in \mathcal{X} \times \mathcal{A}$,

$$F_{\ell,t} Q(x, a) \mid \mathcal{H}_{\ell-1} \succeq_{SO} F_{\mathcal{M},t} Q(x, a) \mid \mathcal{H}_{\ell-1}$$

for any fixed $Q \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ such that $v \geq 3\text{Span}(V_Q)^2$ and $\max_{x \in \mathcal{X}} V_Q(x) \leq \min_{t,x,a} \bar{\theta}_{t,x,a}$.

Remark 11 *When $Q(x, a) \geq 0$ for all (x, a), $\text{Span}(V_Q) \leq \|V_Q\|_\infty \leq \|Q\|_\infty + 1$. Therefore it suffices that $v \geq 3(1 + \|Q\|_\infty)^2$ and $\min_y \bar{\theta}_y \geq \|Q\|_\infty + 1$.*

Proof Recall the Bellman update of Q under the true MDP $\mathcal{M}$ is

$$F_{\mathcal{M},t} Q(x, a) = V_Q^T \mathcal{P}_{t,x,a}^O$$

For each $y = (t, x, a)$, $\mathcal{P}_y^O | \mathcal{H}_{\ell-1} \sim \text{Dirichlet}(\alpha_{\ell,y})$ with $\alpha_{\ell,y} = \alpha_{0,y} + n_\ell(y) \hat{P}_{\ell,y}^O \in \mathbb{R}^{2|\mathcal{X}|}$. Similarly, for each $y = (t, x, a)$, plugging in $\beta = v/\lambda$ we have

$$F_{\ell,t} Q(x, a) \mid \mathcal{H}_{\ell-1} \sim N(\mu_y, \sigma_y^2)$$

where

$$\mu_y \equiv \frac{\beta \bar{\theta}_y + n_\ell(y) V_Q^T \hat{\mathcal{P}}_{\ell,y}^O}{\beta + n_\ell(y)} \qquad \sigma_y^2 \equiv \frac{v}{n_\ell(y) + \beta}.$$

The result follows from Lemma 9 if we establish $\sigma_y^2 \geq (\mathbb{1}^T \alpha_{\ell,y})^{-1} \text{Span}(V_Q)^2$ and $\mu_y \geq V_Q^T \alpha_{\ell,y} / \mathbb{1}^T \alpha_{\ell,y}$. We have

$$\frac{3 \cdot \text{Span}(V_Q)^2}{\mathbb{1}^T \alpha_{\ell,y}} = \frac{3 \cdot \text{Span}(V_Q)^2}{\beta + n_\ell(y)} \leq \sigma_y^2$$

because of the assumption that $v \geq 3 \cdot \text{Span}(V_Q)^2$. Next we have

$$\frac{V_Q^T \alpha_{\ell,y}}{\mathbb{1}^T \alpha_{\ell,y}} = \frac{V_Q^T \alpha_{0,y} + n_\ell(y) V_Q^T \hat{\mathcal{P}}_y^O}{\beta + n_\ell(y)} \leq \frac{\beta \max_{x \in \mathcal{X}} V_Q(x) + n_\ell(y) V_Q^T \hat{\mathcal{P}}_y^O}{\beta + n_\ell(y)} \leq \mu_y$$

because of the assumption that $V_Q(x) \leq \min_y \bar{\theta}_y$ for all x. ■

Lemmas 8 and 10 together imply the stochastic optimism of the state-action value functions $Q_{\ell,0}$ generated by RLSVI.

Corollary 12 *If Assumption 3 holds and RLSVI is applied with parameters $(\bar{\theta}, v, \lambda)$ satisfying $(v/\lambda) = \beta$, $v \geq 3H^2$ and $\min_y \bar{\theta}_y \geq H$,*

$$Q_{\ell,0}(x, a) \mid \mathcal{H}_{\ell-1} \succeq_{SO} Q_{\mathcal{M},0}^*(x, a) \mid \mathcal{H}_{\ell-1}$$

for any history $\mathcal{H}_{\ell-1}$ and state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$.

Proof To reduce notation, we prove this for episode $\ell = 1$, but the proof follows identically for general ℓ by conditioning on the history $\mathcal{H}_{\ell-1}$ at every step. Recall that $Q_{1,0} = F_{1,0}F_{1,1}\cdots F_{1,H-1}0$ and $Q^*_{\mathcal{M},0} = F_{\mathcal{M},0}F_{\mathcal{M},1}\cdots F_{\mathcal{M},H-1}0$.

By Lemma 10,

$$(F_{1,H-1}0)(x,a) \geq_{SO} (F_{\mathcal{M},H-1}0)(x,a) \qquad \forall x, a.$$

Proceeding by induction, suppose for some $t \leq H-1$

$$(F_{1,t+1}F_{1,t+2}\cdots F_{1,H-1}0)(x,a) \geq_{SO} (F_{\mathcal{M},t+1}F_{\mathcal{M},t+2}\cdots F_{\mathcal{M},H-1}0)(x,a) \qquad \forall x, a.$$

Combining this with Lemma 8 shows

$$\begin{aligned} F_{1,t}(F_{1,t+1}F_{1,t+2}\cdots F_{1,H-1}0)(x,a) &\geq_{SO} F_{1,t}(F_{\mathcal{M},t+1}F_{\mathcal{M},t+2}\cdots F_{\mathcal{M},H-1}0)(x,a) \\ &\geq_{SO} F_{\mathcal{M},t}(F_{\mathcal{M},t+1}F_{\mathcal{M},t+2}\cdots F_{\mathcal{M},H-1}0)(x,a) \end{aligned}$$

where the final step uses Lemma 10 combined with the fact that for any $t \in \{0,..,H-1\}$,

$$Q \equiv F_{\mathcal{M},t+1}F_{\mathcal{M},t+2}\cdots F_{\mathcal{M},H-1}0$$

satisfies $3 \cdot \mathrm{Span}(V_Q) \leq 3H \leq v$ and $Q \leq \bar{\theta}$. ■

6.6. Analysis of on-policy Bellman error: proof of Theorem 1

The proof relies on the following bound. For standard Gaussian random variables $X_1, ..., X_n$, a basic Gaussian maximal inequality implies $\mathbb{E}[\max_i X_i] \leq \sqrt{2\log(n)}$. The next lemma is a generalization of this result, which can be seen by taking $J = \arg\max_j X_j$. This lemma is implied by Proposition A.1. of (Russo and Zou, 2015).

Lemma 13 *Let (X, J) be jointly distributed random variables where $X \in \mathbb{R}^n$ follows a multivariate Gaussian distribution with $X_j \sim N(0, \sigma_j^2)$ and $J \in \{1, ...n\}$ is a random index. Then*

$$\mathbb{E}[X_J] \leq \sqrt{2\log(n)\mathbb{E}[\sigma_J^2]}.$$

Applying this leads to two bounds that are used in our analysis. The first bounds the noise terms $w_\ell(t, x_t, a_t)$ of RLSVI at the state and action visited by RLSVI, and the second bounds the norm of the value function sampled by RLSVI.

Corollary 14 *For each $t \leq H$ and $\ell \leq L$*

$$\mathbb{E}[w_\ell(t, x_t, a_t)] \leq \sqrt{2\log(|\mathcal{A}||\mathcal{X}|)\mathbb{E}[\sigma_\ell(t, x_t, a_t)^2]}.$$

Corollary 15 *If RLSVI is applied with parameters $(\lambda, v, \bar{\theta})$ with $v/\lambda = \beta \geq 3$, $v = 3H^2$ and $\bar{\theta} = H\mathbb{1}$,*

$$\mathbb{E}[\max_{\ell \leq L, t < H} \|V_{Q_{\ell,t+1}}\|_\infty] \leq 2H + H^2\sqrt{2\log(1 + |\mathcal{X}||\mathcal{A}|HL)}.$$

A proof of this corollary is provided in the appendix. We now complete the regret analysis of RLSVI and establish Theorem 1.

Proof Set

$$\Delta_\ell = V^*_{\mathcal{M},0}(x_0^\ell) - V^{\pi_\ell}_{\mathcal{M},0}(x_0^\ell).$$

By Corollary 5 and Corollary 12,

$$\mathbb{E}\left[\sum_{\ell=1}^{L} \Delta_\ell\right] \leq \mathbb{E}\left[\sum_{\ell=1}^{L}\sum_{t=0}^{H-1}(F_{\ell,t}Q_{\ell,t+1} - F_{\mathcal{M},t}Q_{\ell,t+1})(x_t^\ell, a_t^\ell)\right].$$

The posterior-mean Bellman update of Q under $\mathcal{M}$ is

$$\mathbb{E}[F_{\mathcal{M},t}Q(x,a)|\mathcal{H}_{\ell-1}] = V_Q^T\mathbb{E}[\mathcal{P}^O_{t,x,a}|\mathcal{H}_{\ell-1}].$$

Recall as well that for each $y = (t, x, a)$, $\mathcal{P}^O_y|\mathcal{H}_{\ell-1} \sim \mathrm{Dirichlet}(\alpha_{\ell,y})$ with

$$\alpha_{\ell,y} = \alpha_{0,y} + n_\ell(y)\hat{P}^O_{\ell,y} \in \mathbb{R}^{2|\mathcal{X}|}.$$

Since the prior over $\mathcal{P}^O_{t,x,a}(\cdot)$ is distributed independently across states and actions (t, x, a), and (t, x, a) cannot be visited prior to period t in any episode, we have also that

$$\mathcal{P}^O_{t,x,a}|\mathcal{H}_{\ell-1}, x_0^\ell, a_0^\ell, .., x_t^\ell, a_t^\ell \sim \mathrm{Dirichlet}(\alpha_{\ell,y}).$$

As a result

$$\begin{aligned} \mathbb{E}[F_{\mathcal{M},t}Q(x,a)|\mathcal{H}_{\ell-1}, x_0^\ell, a_0^\ell, .., x_t^\ell, a_t^\ell] &= \mathbb{E}[F_{\mathcal{M},t}Q(x,a) \mid \mathcal{H}_{\ell-1}] \\ &= \frac{V_Q^T\alpha_{0,y} + n_\ell(y)V_Q^T\hat{P}^O_{\ell,y}}{\beta + n_\ell(y)} \\ &\geq \frac{-\beta\|V_Q\|_\infty}{\beta + n_\ell(y)} + \frac{n_\ell(y)V_Q^T\hat{P}^O_{\ell,y}}{\beta + n_\ell(y)}. \end{aligned}$$

By equation (6.4), we find

$$F_{\ell,t}Q(x,a) - \mathbb{E}[F_{\mathcal{M},t}Q(x,a)|\mathcal{H}_{\ell-1}, x_1^\ell, a_1^\ell, .., x_t^\ell, a_t^\ell] \leq \frac{\beta(\|\bar{\theta}\|_\infty + \|V_Q\|_\infty)}{\beta + n_\ell(y)} + w_\ell(y).$$

Then,

$$\begin{aligned} \mathbb{E}[\Delta_\ell] &\leq \mathbb{E}\left[\sum_{t=0}^{H-1}(F_{\ell,t}Q_{\ell,t+1} - F_{\mathcal{M},t}Q_{\ell,t+1})(x_t^\ell, a_t^\ell)\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{H-1}((F_{\ell,t}Q_{\ell,t+1})(x_t^\ell, a_t^\ell) - \mathbb{E}[F_{\mathcal{M},t}Q_{\ell,t+1}(x_t^\ell, a_t^\ell) \mid \mathcal{H}_{\ell-1}, x_1^\ell, a_1^\ell, .., x_t^\ell, a_t^\ell])\right] \\ &\leq \mathbb{E}\left[\sum_{t=0}^{H-1}\frac{\beta(\|\bar{\theta}\|_\infty + \|V_{Q_{\ell,t+1}}\|_\infty)}{\beta + n_\ell(t, x_t^\ell, a_t^\ell)} + w_\ell(t, x_t^\ell, a_t^\ell)\right] \end{aligned}$$

where the second inequality uses that $Q_{\ell,t+1}$ and $F_{\mathcal{M},t}$ are independent conditioned on $\mathcal{H}_{\ell t}$. Summing over episodes $\ell \in \{1, .., L\}$ implies

$$\mathbb{E}\sum_{\ell=1}^{L}\Delta_\ell \leq \mathbb{E}\left[\beta\left(\|\bar{\theta}\|_\infty + \max_{\ell \leq L, t < H}\|V_{Q_{\ell,t+1}}\|_\infty\right)\sum_{t<H,\ell\leq L}\frac{1}{\beta + n_\ell(t, x_t^\ell, a_t^\ell)} + \sum_{\ell\leq L, t\leq H} w_\ell(t, x_t^\ell, a_t^\ell)\right].$$

Each term can be bounded separately. By Corollary 14

$$\begin{aligned}\mathbb{E}\sum_{\ell\le L,t<H} w_\ell(t,x_t^\ell,a_t^\ell) \le \mathbb{E}\sum_{\ell\le L,t<H} \sigma_\ell(t,x_t^\ell,a_t^\ell)\sqrt{2\log(|\mathcal{X}||\mathcal{A}|)} &= \mathbb{E}\sum_{\ell\le L,t<H}\sqrt{\frac{2v\log(|\mathcal{X}||\mathcal{A}|)}{\beta+n_\ell(t,x_t^\ell,a_t^\ell)}}\\ &\overset{(a)}{\le} 2\sqrt{2vH^2|\mathcal{X}|\mathcal{A}|L\log(|\mathcal{X}||\mathcal{A}|)}\\ &= 2H^2\sqrt{6|\mathcal{X}|\mathcal{A}|L\log(|\mathcal{X}||\mathcal{A}|)}\end{aligned}$$

where the second to last inequality is proved in Lemma 16, provided below. The other term can be bounded as,

$$\begin{aligned}&\mathbb{E}\left[\beta\left(\|\bar\theta\|_\infty+\max_{\ell\le L,t\le H}\|V_{Q_{\ell,t+1}}\|_\infty\right)\sum_{t\le T,\ell\le L}\frac{1}{\beta+n_\ell(t,x_t^\ell,a_t^\ell)}\right]\\ \overset{(b)}{\le}\ &\beta\left(\|\bar\theta\|_\infty+\mathbb{E}\left[\max_{\ell\le L,t\le H}\|V_{Q_{\ell,t+1}}\|_\infty\right]\right)H|\mathcal{X}||\mathcal{A}|\log\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)\\ \overset{(c)}{\le}\ &\beta\left(H+2H+H^2\sqrt{2\log(1+|\mathcal{X}||\mathcal{A}|HL)}\right)H|\mathcal{X}||\mathcal{A}|\log\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)\\ \le\ &5\beta H^3|\mathcal{X}||\mathcal{A}|\sqrt{\log_+(1+|\mathcal{X}||\mathcal{A}|HL)}\log\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)\end{aligned}$$

where the bound on the sum in inequality (b) is from Lemma 16 (proof in the Appendix), and inequality (c) applies Corollary 15.

Lemma 16 *If $\beta \ge 2$ then with probability 1,*

$$\sum_{\ell\le L}\sum_{t\le H}\frac{1}{\beta+n_\ell(t,x_t^\ell,a_t^\ell)} \le H|\mathcal{X}||\mathcal{A}|\log\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)$$

and

$$\sum_{\ell\le L}\sum_{t\le H}\sqrt{\frac{1}{\beta+n_\ell(t,x_t^\ell,a_t^\ell)}} \le 2\sqrt{H^2|\mathcal{X}||\mathcal{A}|L}.$$

Together, the calculations above yield the regret bound

$$\mathbb{E}\sum_{\ell=1}^L \Delta_\ell \le 5\beta H^3|\mathcal{X}||\mathcal{A}|\sqrt{\log_+(1+|\mathcal{X}||\mathcal{A}|HL)}\log\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)+2H^2\sqrt{6|\mathcal{X}|\mathcal{A}|L\log(|\mathcal{X}||\mathcal{A}|)}.$$

Unfortunately, this alone does not yield the desired bound of order $\tilde O(H^2\sqrt{|\mathcal{X}||\mathcal{A}|L})$. To complete the proof, we consider two cases. First suppose $L \ge 25\beta H^2|\mathcal{X}||\mathcal{A}|$. Then

$$\begin{aligned}\mathbb{E}\sum_{\ell=1}^L\Delta_\ell &\le H^2\sqrt{|\mathcal{X}|\mathcal{A}|L\log_+(1+|\mathcal{X}||\mathcal{A}|HL)}\left(2\sqrt{6}+5\beta H\sqrt{|\mathcal{X}||\mathcal{A}|/L}\log\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)\right)\\ &\le H^2\sqrt{|\mathcal{X}||\mathcal{A}|L\log_+(1+|\mathcal{X}||\mathcal{A}|HL)}\left(2\sqrt{6}+\sqrt{\beta}\log_+\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)\right)\\ &\overset{(*)}{\le} 6H^2\sqrt{\beta|\mathcal{X}||\mathcal{A}|L\log_+(1+|\mathcal{X}||\mathcal{A}|HL)}\log_+\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right)\end{aligned}$$

which is the desired bound. When $L \le 25H^2|\mathcal{X}||\mathcal{A}|$, we use the naive bound

$$\mathbb{E}\sum_{\ell=1}^L\Delta_\ell \le HL \le H\sqrt{L}\sqrt{25H^2|\mathcal{X}||\mathcal{A}|} = 5H^2\sqrt{\beta|\mathcal{X}||\mathcal{A}|L},$$

which is also less than the term in $(*)$. This completes the proof of Theorem 1. ■

7. Computational studies

In Section 6 we established formal guarantees for a tabular version of RLSVI. This result serves as a sanity check, demonstrating that RLSVI carries out efficient deep exploration, but the tabular nature and prior structure of the setting limits the scope of our theoretical results. Perhaps most importantly, the results do not apply when parameterized representations are used to generalize across states and actions. In this section, we present computational results that offer further assurances. In particular, we discuss results from a series of experiments designed to enhance insight into the workings of RLSVI beyond the scope of our theoretical analysis. The focus of these experiments is to improve understanding, rather than to solve challenging problems. Nevertheless, we believe that observations from these didactic examples will prove valuable toward the design of practical systems that require the synthesis of efficient deep exploration with effective generalization.

7.1. Deep-sea exploration

We begin our computational experiments with an empirical study of the "deep-sea exploration" problem from Example 1. This offers a simple illustration of the importance of deep exploration. Although the associated MDP has only N^2 states, dithering schemes require a number of episodes that grows exponentially in N to effectively explore the environment. Deep exploration approaches, on the other hand, can effectively explore the environment within a sub-exponential number of episodes. Our results verify the efficacy randomized value functions and that RLSVI carries out deep exploration.

7.1.1. Tabular representation

We begin with an investigation into RLSVI with a tabular representation. Our goal will be to study the behavior of RLSVI in a simple setting similar to that addressed by Theorem 1. To do this, we generate random "deep-sea" environments according to Example 1 and empirically evaluate performance over many simulations.

We apply RLSVI `agent` with infinite buffer, greedy actions and `learn_grlsvi`[8] with a tabular representation. Specifically, each component of the parameter vector $\theta \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ provides a value estimate for one state-action pair. We set the tuning parameters to $v = H^2/25$, $\bar\theta = 0$, and $\lambda = v$. Note that, compared to the setting specified in Theorem 1 we rescaled v by a constant in order to accelerate learning in the deterministic deep-sea environment. Further our prior parameter $\bar\theta$ is not optimistic. This choice is not particularly important in terms of performance on this task, in fact setting $\bar\theta = H\mathbb{1}$ leads to almost

8. `learn_grlsvi` is `learn_rlsvi` (Algorithm 4) with additive Gaussian reward noise of variance v (5.5)

identical results. The reason we do not rely on an optimistic prior is to highlight the practical efficacy of RLSVI even without optimistic prior. This will be important in later sections where we study RLSVI in domains with generalization, for which the notion of an universally 'optimistic prior' does not carry over from the tabular setting (Osband et al., 2016b).

We compare the performance of RLSVI against two well-studied reinforcement learning algorithms specifically designed to explore efficiently with tabular representations: UCRL2 (Jaksch et al., 2010) and PSRL (Osband et al., 2013). We similarly modify UCRL2 and PSRL that accelerate learning in the deterministic deep-sea environment[9]. For each of the algorithms our modifications reduce learning times but do not affect rates at which learning times scale with problem size.

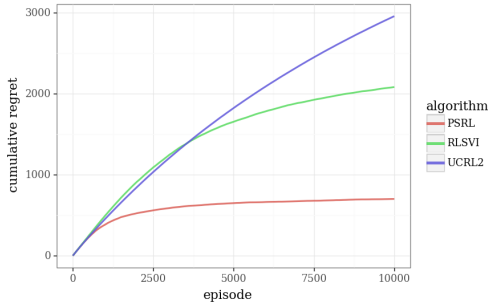


Figure 4: RLSVI is competitive with algorithms designed for tabular exploration ($N = 10$).

Figure 4 plots the average regret realized by RLSVI (specifically, `learn_grlsvi`), UCRL2 and PSRL, over five seeds with a bomb and five seeds with treasure. Among the algorithms, PSRL offers the lowest level of regret, followed by RLSVI, and then UCRL2. Hence, RLSVI is competitive with these algorithms, which are designed to yield efficient exploration with tabular representations.

One natural question is how this performance scales with the size N of the problem. To answer this we study the "learning time," defined to be the first episode where the average regret per episode is less than 0.5. Formally,

$$\text{Learning time}(\mathcal{M}^*, \text{alg}) := \min\left\{ L > 1 \,\middle|\, \frac{\text{Regret}(\mathcal{M}^*, \text{alg}, L)}{L} \leq 0.5 \right\}, \tag{7.1}$$

This quantity is random, as it depends on the realization of $\mathcal{M}^*$. For an algorithm with regret bound $\text{Regret}(\mathcal{M}^*, \text{alg}, L) \leq \sqrt{BL}$ we would expect the learning time to be $\tilde{O}(B)$.

The results of Theorem 1 suggests an $\tilde{O}(\sqrt{H^3SAL})$ *average* scaling when the environment is drawn from a symmetric Dirichlet distribution. We can contrast this to existing performance guarantees for UCRL2 which, when adapted to this setting, provide a

9. Specifically, we update the confidence sets for PSRL and UCRL2 as if each observed transition (s, a, r, s') occured identically 10 times repeatedly. We also further rescale the confidence sets for UCRL2 to be 10 times smaller than prescribed by the analysis.

$\tilde{O}(\sqrt{H^3S^2AL})$ regret bound. For the deep-sea problem, $H = N$, $S = N^2$ and $A = 2$, and the bounds therefore suggests that learning times scale as $\tilde{O}(N^5)$ for RLSVI and $\tilde{O}(N^7)$ for UCRL2. Figure 5 shows that observed performance to a large degree matches performance suggested by these theoretical results. The best known bound for PSRL also suggests a $\tilde{O}(N^5)$ scaling. However, recent work suggests that this bound is loose (Osband and Van Roy, 2017, 2016), and the associated plot in Figure 5 strengthens the case for that hypothesis.

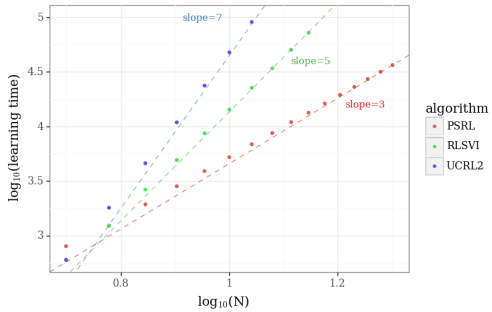


Figure 5: Scaling with tabular learning.

7.1.2. Linearly parameterized value functions

Section 7.1.1 presents evidence of the efficacy of RLSVI with a tabular representation. However, the value of RLSVI lies in its ability to function well with parameterized value functions that generalize across states and actions. Model-based algorithms such as UCRL2 or PSRL do not accommodate this form of generalization.

In this subsection, we continue our investigation of the "deep-sea" environment, but now using linear parameterized representations. To do this, we generate a random subspace of dimension D that is specifically designed to include the true optimal value function of the deep-sea environment irrespective of whether there is treasure or a bomb. We then generate a random basis of D unit vectors $\phi_1, \ldots, \phi_D \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ that span the space. Each vector ϕ_d can be thought of as representing a feature that assigns a numerical value to each state-action pair. As such, the representation can be thought of as a linear combination of features, with a dimensional parameter vector $\theta \in \mathbb{R}^D$ encoding feature weighs: $\tilde{Q}_\theta = \sum_{d=1}^D \theta_d \phi_d$.

To facilitate efficient computation in the deep-sea problem we restrict these D-dimensional features so that each is nonzero only at state-action pairs corresponding to one row of the grid of states. We only consider values of D that are multiples of N, and for each row assign $M = D/N$ features to generate nonzero values. With this representation, RLSVI learns a separate M-dimensional representation for each row, avoiding a costly dimension D inversion. Figure 6 plots realized regret generated by `learn_grlsvi` with $\lambda = 100$, $v = 0.01$, and $N = 50$. Once again, we simulate this problem for five random seeds with treasure and five random seeds with a bomb and report the average regret. These results demonstrate that

per-episode regret vanishes much faster than any dithering method, which would expect at least $2^{50} \simeq 10^{15}$ episodes to even reach the chest!

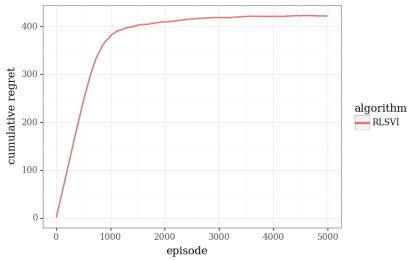
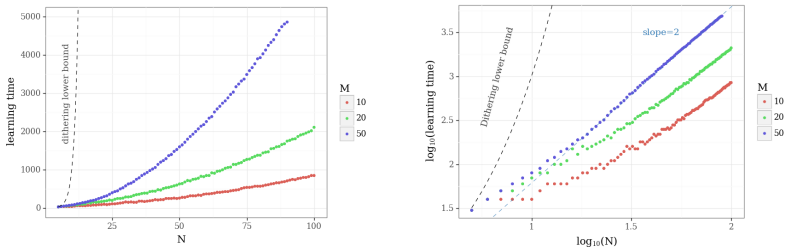


Figure 6: Regret with $N = 50$ and $M = 50$

It is the cases with treasure rather than a bomb that bind regret and learning times. For this reason we will only present results associated with the former case from here on to save on computation. Figure 7(a) plots learning times as a function of N for different numbers of features M per row. As one would expect, the learning time increases with the number of features. Importantly, this scaling with chain length N is graceful and grows much more slowly than even the lower bound for dithering methods $O(2^N)$. Figure 7(b) plots the same data on a log-log scale to highlight this sub-exponential growth. We can see empirically that the slope on this scale is approximately two, implying that learning time scales approximately quadratically in N.



(a) Raw values (b) Log-log plot shows scaling

Figure 7: Effect of problem size N on learning time.

For another perspective on scaling, Figure 8 presents plots of learning times as a function of the number of features M, for several values of N. In each case, the learning time appears to grow linearly in the number of features up until some threshold and then increase much more slowly beyond this point. The vertical dotted lines in Figure 8 appear at $M = 2N$. Empirically, this seems to be the point beyond which the incremental learning time incurred with additional features is small. Intuitively, one might speculate that this is reasonable because $2N$ is equal to the maximum number of states-action pairs which can be observed in any time period. Beyond this point, additional features must be linearly dependent.

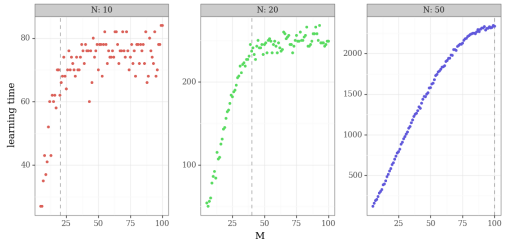


Figure 8: Scaling with number of features.

7.1.3. Misspecified representations

Let us now consider a more realistic setting in which the value function representation is mis-specified in the sense that Q^* is not equal to $\tilde{Q}_\theta$ for any vector θ. We experiment with a setting completely analogous to that of the previous section, except we add to each feature vector ϕ_d a random vector $\eta_d \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$. The random vector η_d is nonzero only only at state-action pairs associated with the feature ϕ_d. Each nonzero noise component is sampled from $N(0, \psi I)$. Hence, we make use of a representation of the form $\tilde{Q}_\theta = \sum_{d=1}^{D} \theta_d(\phi_d + \eta_d)$. As the parameter ψ increases, the representation becomes increasingly misspecified.

Figure 9 plots cumulative regret of `learn_grlsvi` with varying numbers of features and degrees of misspecification over 5000 episodes. Our results are the average of 20 seeds for each value of the noise scale ψ. These results indicate that RLSVI remains robust to some degree of misspecification. However, at some point the model-misspecification becomes too severe as the value of ψ increases depending upon the number of basis functions M. The power of the representation increases with the number of features, and this enables RLSVI to tolerate larger values of ψ. In the case $M \geq 2N$ random basis functions will span the true value function with high probability. As expected, we observe that for $M = 40, N = 20$ RLSVI performs similarly well irrespective of ψ.

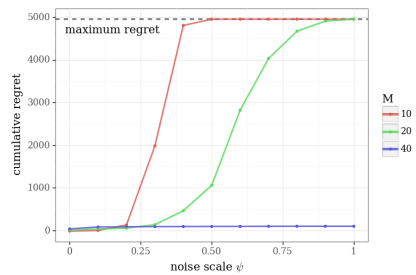


Figure 9: Robustness to misspecification with $N = 20$.

7.1.4. Parameter tuning

The computational results we have present in Sections 7.1.2 and 7.1.3 make use of particular settings for the prior and noise variance parameters $\lambda = 100, v = 0.01$. In this section, we study the dependence of results on these parameter settings. The deep-sea problem we have considered is in some sense degenerate because each problem instance is deterministic. In order to offer a more representative set of results pertaining to variance parameter tuning, we will also consider a modified version of the deep sea problem where all reward observations are corrupted by some $N(0, 1)$ noise.

Figure 10 plots the cumulative average regret after 5000 episodes over 10 random seeds for various choices of prior and noise variance with $N = 20$ and $M = 10$. In both settings with and without stochastic rewards, and for all choices of noise randomization, we can see that prior variance which is too small can prohibit learning. In this problem, where our prior $\overline{\theta} = 0$ is not informative, choosing even very large λ does not degrade performance.

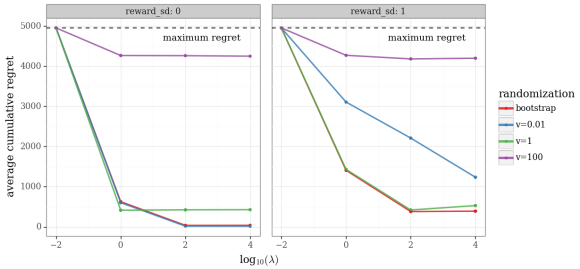


Figure 10: Robustness to prior and noise variance parameters.

Figure 11 takes the same data as Figure 10 but investigates the sensitivity of RLSVI to the noise randomization, for the choice $\lambda = 100$. We see that choice of the best-performing noise variance v is largely dependent upon the scale of the noise in the actual environment. When the underlying environment is deterministic there is no benefit to adding noise and low values of v perform best. However, when the environment is stochastic choosing v on the order of the variance of the noise in the environment is necessary to not fall victim to unlucky observations. In both settings, bootstrapping performs competitively with the ex-ante "best" choice of v but does not need to be specified in advance.

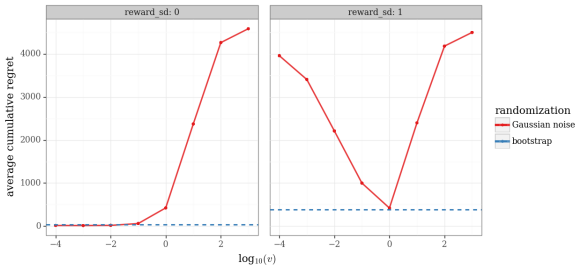


Figure 11: Bootstrap is competitive with the best choice of v across levels of noise.

To gain some more intuition for this parameter tuning we take this same data as Figure 11 and present the realized regret by random seed in Figure 12. We see v smaller than the noise in the problem can lead to premature and sub-optimal convergence that never opens the chest (and so leads to linear regret). Choices of v which are too large lead to slower learning and more exploration, but do not lead to linear regret. We note that RLSVI with $v > 0$ does not seem to significantly degrade in performance for stochastic rewards with variance up to v. Once again, we see that randomization by bootstrap is competitive with the best ex-ante choice of v but with one fewer parameter to tune.

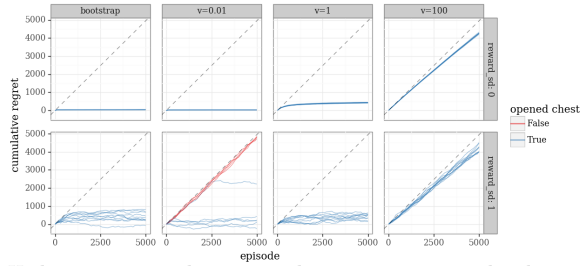
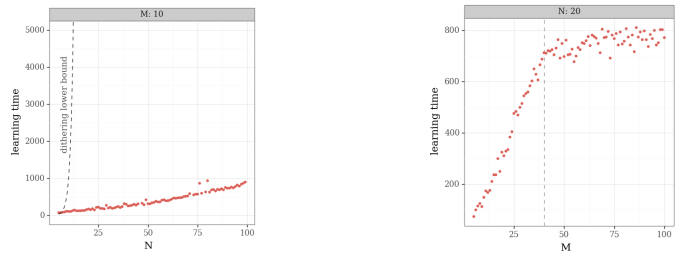


Figure 12: Higher v is more robust to stochastic environments but learns more slowly.

The bootstrap learns the "right" noise variance, and beyond that, can even learn how it should vary over states and actions. Further, Figure 13 plot learning times from applying the `learn_brlsvi` in the same settings to which `learn_grlsvi` was applied to generate Figures 6 and 7. These results suggest that learning times of `learn_brlsvi` scale similarly to those of `learn_grlsvi`. In general, our results suggest that bootstrapping offers a natural approach to setting an appropriate form of randomizing noise without prior knowledge.



(a) Scaling with N (b) Scaling with number of features.

Figure 13: Performance of the bootstrap scales similarly to `learn_grlsvi`.

7.2. Deep exploration with deep learning

The experiments of Section 7.1 are designed to highlight several key properties of RLSVI in a simple setting. These results demonstrate that RLSVI can successfully synthesize efficient

exploration with generalization. However, the context was a "toy" example in that the underlying system involved a tractable number of states. Further, the algorithms we evaluate in that section are not practical for large-scale learning problems for two reasons. First, they recompute the whole history of data each episode and so have computational costs that grow with the amount of data collected. Second, their performance is highly reliant upon an accurate linear basis for the value function whereas many of the recent breakthroughs in the field have come from so-called "deep" RL that uses deep neural networks for function approximation in RL.

In this section we present computational results for deep exploration with a practical variant `learn_ensemble_rlsvi` together with neural network function approximation. We begin with more experiments on the "deep sea" problem and show that a parallel RLSVI strategy can recover performance qualitatively similar to full batch resampling, but at a dramatically lower computational cost. Next we investigate the scaling properties of RLSVI with nonlinear neural network representations and find that the performance can successfully synthesize exploration with generalization in this setting. Finally, we apply these findings to a difficult task in continuous control. We find that `learn_ensemble_rlsvi` successfully demonstrates deep exploration together with complex nonlinear generalization.

7.2.1. RLSVI via ensemble sampling

In this section we apply `learn_ensemble_rlsvi` (Algorithm 8) for $K = 1, 5, 10, 20, 40$ and with an `ensemble_buffer` that stores the most recent 10^5 transitions. For `update` we use `update_bootstrap` (Algorithm 7) to approximate a "double or nothing" online bootstrap (Owen and Eckles, 2012). We use a discounted TD loss with $\gamma = 0.99$, learning rate $\alpha = 10^{-3}$ and minibatch size of 128. For our value function family $\mathcal{Q}$ we consider two-layer MLP with 50 rectified linear units in each layer.

In place of explicit prior regularization $\mathcal{R}$ we evaluate $\tilde{Q}_{\theta_k} = f^{\text{MLP}}_{\theta_k} + f^{\text{MLP}}_{\theta^0_k}$ where f^{MLP} is a 2-layer MLP and the parameters θ_k, θ^0_k are sampled independently from Glorot initialization and henceforth θ^0_k is held fixed (Glorot and Bengio, 2010; Osband et al., 2018). We found that this randomization plus SGD training provide sufficient regularization for deep learning without use of weight decay (Zhang et al., 2016; Bartlett et al., 2017). Using two separate networks, one with fixed weights, is useful so that the SGD training cannot easily learn to ignore the state input and learn $Q = 0$ as a degenerate global solution. More detail on this specific prior mechanism for deep neural networks is available in Osband et al. (2018).

We apply this algorithm to "deep sea" problem with a raw pixel representation in $[0, 1]^{N \times N}$ where the diver's position is given by a 1 and all other entries zero. This problem is by nature tabular and does not necessitate complex representation via neural network however we use it as a simple example to investigate the importance of ensemble size K. Figure 14 presents the performance for $N = 25$ averaged over 20 seeds. We see that even a relatively small number of parallel estimates K can direct deep exploration and that, for a problem of this size, the marginal benefits seem to plateau around $K = 20$. The computational savings of this parallel approach can be quite significant. We compare the cost of $K = 20$ parallel Q-networks each with $O(1)$ computation per episode against a naive `learn_brlsvi` that computes $H = 25$ value functions each with $\Omega(L)$ computation per episode.

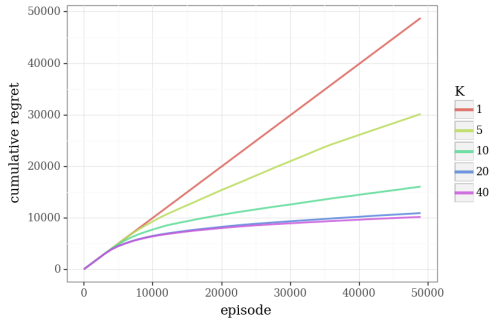


Figure 14: Investigating ensemble size K for `learn_ensemble_rlsvi` on deep sea $N = 25$.

Our next set of experiments investigates the scalability of `learn_ensemble_rlsvi` with different feature representations. We repeat the experiment of Figure 14 with $K = 20$ varying $N = 5, .., 50$ under three separate representations and averaged over 20 seeds. First we consider the raw pixel representation as above; this is effectively a tabular problem. Next, we consider the informative linear basis of Section 7.1.2 with $M = 10$; this learning can be expedited through generalization. Finally, we consider the raw pixel representation but alter the problem formulation so that the action 1 is always "go right"; the optimal value function for this setting takes a particularly simple form and an optimal policy can easily be happened upon through random weight initialization and without any learning. Figure 15 shows that RLSVI with neural network architecture can exploit these feature representations where they are present, but defaults to an approximately tabular learning approach when they are not.

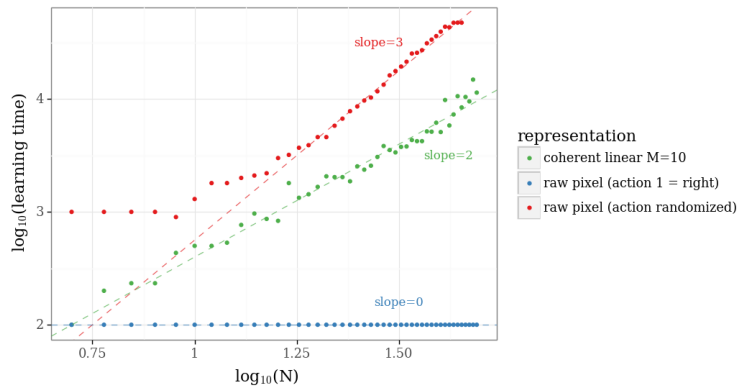


Figure 15: Log-log plot shows empirical scalings with different feature representations.

7.2.2. Cartpole swing up

In this section we consider the classic "cartpole" problem of a cart attached to a pole on a frictionless rail. We modify the problem so that, as in previous sections, deep exploration is crucial to finding rewarding states and thus learning the optimal policy. However, unlike "deep sea" the underlying dynamics are not governed by a small finite MDP. The cart is of mass $M = 1$ and the pole is mass $m = 0.1$ and length $l = 1$, with acceleration due to gravity $g = 9.8$. At each timestep the agent can apply a horizontal force F_t to the cart. The dynamics for this system are given by a second order differential equation in x_t, the horizontal position of the cart and θ_t, the angle of the pole from vertically upright at $\theta = 0$,

$$\tau_t = \frac{F_t + \frac{l}{2}\dot{\theta}_t^2 \sin(\theta_t)}{m+M}, \quad \ddot{\theta}_t = \frac{g\sin(\theta_t) - \cos(\theta_t)\tau_t}{\frac{l}{2}\left(\frac{4}{3} - \frac{m}{m+M}\cos(\theta_t)^2\right)}, \quad \ddot{x} = \tau - \frac{m\frac{l}{2}\ddot{\theta}_t \cos(\theta_t)}{m+M}. \tag{7.2}$$

Unlike the traditional cartpole problem, where the agent begins with the pole stood upright and must learn to balance it; our agent begins each episode with the pole hanging down and has to learn to swing it up. Concretely we interact with the environment through the state $s_t := (\theta_t, \dot{\theta}_t, x_t, \dot{x}_t, t) \in \mathbb{R}^5$. Each episode begins with $s_0 = (\pi, 0, 0, 0) + w$ for $w_i \sim \text{Unif}([-0.05, 0.05])$ i.i.d. in each component. We discretize the evolution of (7.2) with timescale $\Delta t = 0.01$ and present the choice of actions $F_t \in \{-10, 0, 10\}$ for all t. The reward structure of this task is specifically designed to necessitate deep exploration; each timestep the agent pays a cost $\frac{|F_t|}{1000}$ for its action but can receive a reward of 1 if the pole is balanced upright and steady in the middle[10]. The ends of the rail at $x = -5, 5$ are rigid and immovable; an episode ends whenever $t > 10$.

Figure 16 presents results for DQN and a 50-50-MLP with rectified linear units with linear annealing ϵ-greedy dithering from 1 to 0 over varying number of episodes. Irrespective of annealing schedule, DQN is unable to gather informative data since it does not perform deep exploration. In this environment, dithering strategies for exploration are insufficient to gain information beyond the locally-attractive policy to remain motionless and receive a reward of zero.

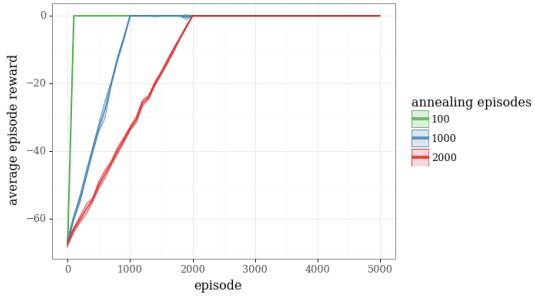


Figure 16: DQN with ϵ-greedy exploration simply learns to stay motionless.

10. Reward +1 received if $\cos(\theta) > 0.95$ with all of $|\dot{\theta}|, |x|, |\dot{x}| \leq 1$.

Figure 17 presents the average episodic reward for an ensemble approach to RLSVI with $K = 20$ and the same algorithmic approach as Section 7.2.1. We note that, unlike DQN with ϵ-greedy exploration, RLSVI is able to learn a successful swing up policy with an identical network architecture. In addition, we note that RLSVI implemented with linear or single-layer Q-value functions is unable to learn a successful swing up policy. This demonstrates the importance of *both* deep exploration and deep representation learning in order for a successful application of deep RL in this setting. The results of Figures 16 and 17 are averaged over 20 seeds, with confidence intervals at 1 standard error of the mean. We present visualization of this performance `https://youtu.be/ia72VyW5MfI`.

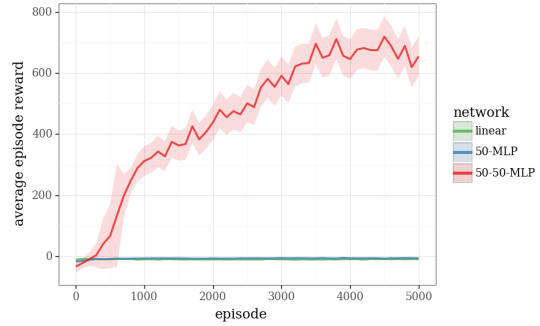


Figure 17: RLSVI with 2-layer neural network is able to learn a near-optimal policy.

The computational results we present in this paper are tailored to be simple and interpretable, with a clear focus on the importance of deep exploration and the compatibility of this approach with linear and nonlinear value function learning. Related work investigates scaling up this approach in several arcade games including Tetris, Angry Birds, Atari 2600 as well as a model of recommendation systems (Osband et al., 2016b; Ibarra et al., 2016; Osband et al., 2016a). We consciously choose to keep our empirical investigation concise and the key results sanitary, but look forward to pushing the boundaries of large-scale applications of "deep RL" via randomized value functions in future work.

8. Closing remarks

Much of the applied RL literature focuses on simulated systems and learning a good final policy, potentially over billions or trillions of episodes. Assessed in this manner, performance is driven largely by the investment of computational resources and simulation time; not just how effectively a reinforcement learning algorithm makes decisions and interprets observations. In many real systems, data collection is costly or constrained by the physical context, and this calls for a focus on statistical efficiency. In these contexts it may be more appropriate, for example, to evaluate algorithms over a fixed number of episodes.

Exploration is a key driver of statistical efficiency. As discussed in Section 4, there can be an exponentially large difference in data requirements between an agent that explores via dithering, as has commonly been done in past applications of reinforcement learning, and

an agent that carries out deep exploration. In this paper, we have developed randomized value functions as a concept that enables efficient deep exploration in conjunction with value function learning methods commonly used in reinforcement learning.

Acknowledgements

This work was generously supported by a research grant from Boeing, a Marketing Research Award from Adobe, and Stanford Graduate Fellowships, courtesy of PACCAR, Burt and Deedee McMurty, and ST Microelectronics. We thank Emma Brunskill, Hamid Reza Maei, and Rich Sutton for helpful discussions and Vikranth Dwaracherla, Xiuyuan Lu, Shuhui Ku, and Kuang Xu for pointing out errors and ambiguities in earlier drafts, and more broadly, students who participated in Stanford University's 2017 and 2018 offerings of Reinforcement Learning, for feedback and stimulating discussions on this work. We also thank John Aslanides, Albin Cassirer, Alex Pritzel, Charles Blundell and the rest of the team at DeepMind for help with experiments, infrastructure and an engaging work environment.

APPENDIX

Appendix A. Proofs of technical lemmas

A.1. Proof of Lemma 3

Lemma 1 (Planning Error to Bellman Error) *Let $Q_0, Q_1, Q_2, ..., Q_H \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ be any sequence with $Q_H = 0 \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ and take $\pi = (\pi_0, \pi_1, ...)$ to be the policy $\pi_t(x) = \arg\max_{a \in \mathcal{A}} Q_t(x,a)$ for all a, x. Then for any MDP $\mathcal{M}$ and initial state $x \in \mathcal{X}$,*

$$Q_0(x, \pi_0(x)) - V^\pi_{\mathcal{M}}(x) = \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=0}^{H} ((Q_t - F_{\mathcal{M},t}Q_{t+1})(x_t, a_t))\,|x_0 = x)\right] \tag{A.1}$$

Proof Define the operator $F^\pi_{\mathcal{M},t}$ at time t for the MDP $\mathcal{M}$ and policy π by

$$F^\pi_{\mathcal{M},t}Q(x,a) = \mathbb{E}[r_{t+1} + Q(x_{t+1}, \pi_{t+1}(x_{t+1}))|\mathcal{M}, x_t = x, a_t = a].$$

Let $Q^\pi_{\mathcal{M},0}, ..., Q^\pi_{\mathcal{M},H} \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$ be defined according to $Q^\pi_{\mathcal{M},H} = 0$ and

$$Q^\pi_{\mathcal{M},t} = F^\pi_{\mathcal{M},t}Q^\pi_{\mathcal{M},t+1} \quad t \in \{0, ..., H-1\}.$$

Then $Q^\pi_{\mathcal{M},0}(x, \pi_0(x)) = V^\pi_{\mathcal{M}}(x)$ and, since $\pi_t(x) = \arg\max_a Q_{t+1}(x,a)$, $F^\pi_{\mathcal{M},t}Q_{t+1} = F_{\mathcal{M},t}Q_{t+1}$ for all t. We can therefore rewrite (A.1) as

$$Q_0(x, \pi_0(x)) - Q^\pi_{\mathcal{M},0}(x, \pi_0(x)) = \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=0}^{H-1} \left((Q_t - F^\pi_{\mathcal{M},t}Q_{t+1})(x_t, a_t)\right)|x_0 = x)\right].$$

We have

$$\begin{aligned} Q_0 - Q^\pi_{\mathcal{M},0} &= Q_0 - F^\pi_{\mathcal{M},0}Q_1 + F^\pi_{\mathcal{M},0}Q_1 - Q^\pi_{\mathcal{M},0} \\ &= Q_0 - F^\pi_{\mathcal{M},0}Q_1 + F^\pi_{\mathcal{M},0}Q_1 - F^\pi_{\mathcal{M},0}Q^\pi_{\mathcal{M},1}. \end{aligned}$$

By definition, this means

$$(Q_0 - Q^\pi_{\mathcal{M},0})(x, \pi_0(x)) = \left(Q_0 - F^\pi_{\mathcal{M},0}Q_1\right)(x, \pi_0(x)) + \mathbb{E}_{\mathcal{M},\pi}[\left(Q_1 - Q^\pi_{\mathcal{M},1}\right)(x_1, a_1)|x_0 = x].$$

The result follows by iterating this relation. ■

A.2. Proof of Lemma 7

Lemma 2 (Preservation under convex operations) *For two collections $(X_1, ..., X_n)$ and $(Y_1, ..., Y_n)$ of independent random variables with $X_i \geq_{SO} Y_i$ for each $i \in \{1, ...n\}$ and any convex increasing function $f : \mathbb{R}^n \to \mathbb{R}$,*

$$f(X_1, ..., X_n) \geq_{SO} f(Y_1 ..., Y_n).$$

Proof The proof proceeds by induction on n. First consider the base case $n = 1$. Fix any convex increasing function $u : \mathbb{R} \to \mathbb{R}$. Then $u \circ f$ is convex increasing and

$$\mathbb{E}[u(f(X_1))] \geq \mathbb{E}[u(f(Y_1))].$$

Now suppose the result holds for any collection of $n-1$ random variables. Fix any convex increasing $u : \mathbb{R} \to \mathbb{R}$: Define the convex increasing functions $U_X : \mathbb{R} \to \mathbb{R}$ and $U_Y : \mathbb{R} \to \mathbb{R}$ by

$$\begin{aligned} U_X(z) &\equiv \mathbb{E}[u(f(z, X_2, ..., X_n))] \\ U_Y(z) &\equiv \mathbb{E}[u(f(z, Y_2, ..., Y_n))]. \end{aligned}$$

For each fixed $z \in \mathbb{R}$, $U_X(z) \geq U_Y(z)$ since

$$U_X(z) = \mathbb{E}[u(f_z(X_2, .., X_n))] \geq \mathbb{E}[u(f_z(Y_2, .., Y_n))] = U_Y(z)$$

where $f_z : \mathbb{R}^{n-1} \to \mathbb{R}$ is the convex increasing function $f_z(x_2, ..., x_n) = f(z, x_2, ..., x_n)$ and $f_z(X_2, ...X_n) \succeq_{SO} f_z(Y_2, ...Y_n)$ by the inductive hypothesis. We conclude

$$\mathbb{E}[u(f(X_1, ..., X_n))] = \mathbb{E}[U_X(X_1)] \geq \mathbb{E}[U_Y(X_1)] \geq \mathbb{E}[U_Y(Y_1)] = \mathbb{E}[u(f(X_1, ..., X_n))]$$

where the first and last equality use the independence of $(X_1, ..., X_n)$ and $(Y_1, ..., Y_n)$ along with the Fubini–Tonelli theorem. The final inequality uses the definition of stochastic optimism. ■

A.3. Proof of Corollary 15

Corollary 1 *If RLSVI is applied with parameters $(\lambda, v, \bar{\theta})$ with $v/\lambda = \beta \geq 3$, $v = 3H^2$ and $\bar{\theta} = H\mathbb{1}$,*

$$\mathbb{E}[\max_{\ell \leq L, t < H} \|V_{Q_{\ell,t+1}}\|_\infty] \leq 2H + H^2\sqrt{2\log(|\mathcal{X}||\mathcal{A}|HL)}$$

Proof To begin, we observe a basic fact about the maximum of Gaussian random variables. Fix independent Gaussian random variables $X_0, X_1, ..., X_n \sim N(0, 1)$. Let $f : (x_0, ..., x_n) \mapsto \max_i x_i$ be the maximum function, so $\mathbb{E}[f(X_0, ..., X_n)] \leq \sqrt{2\log(n+1)}$ by a standard Gaussian maximum inequality. Then by Jensen's inequality,

$$\begin{aligned} \mathbb{E}\left[\left(\max_{i \in \{1,...,n\}} X_i\right)_+\right] = \mathbb{E}[f(0, X_1, ..., X_n)] &= \mathbb{E}[f(\mathbb{E}[(X_0, X_1, ..., X_n)|X_1, ..., X_n])] \\ &\leq \mathbb{E}[f(X_0, X_1, ..., X_n)] \\ &\leq \sqrt{2\log(n+1)}. \end{aligned} \tag{A.2}$$

For every state action value function $Q \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$, $\|V_Q\|_\infty \leq 1 + \|Q\|_\infty$. Therefore, by equation 6.4, for every episode ℓ and period t,

$$\begin{aligned} \|F_{\ell,t}Q\|_\infty &\leq \max\{\|\theta\|_\infty, \|V_Q\|_\infty\} + \max_{x \in \mathcal{X}, a \in \mathcal{A}} w_\ell(t, x, a) \\ &\leq \max\{\|\theta\|_\infty, \|Q\|_\infty\} + 1 + \max_{x \in \mathcal{X}, a \in \mathcal{A}} w_\ell(t, x, a) \\ &\leq \max\{\|\theta\|_\infty, \|Q\|_\infty\} + 1 + w_{\max} \end{aligned}$$

where $w_{\max} \triangleq \left(\max_{t \leq H, \ell \leq L, a \in \mathcal{A}, x \in \mathcal{X}} w_\ell(t, x, a)\}\right)_+$. This implies

$$\|Q_{\ell,H-1}\|_\infty = \|F_{\ell,H-1}0\|_\infty \leq \|\theta\|_\infty + 1 + w_{\max}.$$

Then

$$\|Q_{\ell,H-2}\|_\infty = \|F_{\ell,H-2}Q_{\ell,H-1}\|_\infty \leq \max\{\|\theta\|_\infty, \|Q_{\ell,H-1}\|_\infty\} + 1 + w_{\max} \leq \|\theta\|_\infty + 2(1 + w_{\max}).$$

Repeating this by backward induction shows,

$$\max_{t<H} \|\|Q_{\ell,t+1}\|_\infty \leq \|\theta\|_\infty + (H-1)(1 + w_{\max})$$

Therefore

$$\max_{t<H} \|V_{Q_{\ell,t+1}}\|_\infty \leq 1 + \max_{t<H} \|Q_{\ell,t+1}\|_\infty \leq \|\bar{\theta}\|_\infty + H\,(1 + w_{\max})\,.$$

In addition

$$\begin{aligned} \mathbb{E}[w_{\max}] = \mathbb{E}\left[\left(\max_{t \leq H, \ell \leq L, a \in \mathcal{A}, x \in \mathcal{X}} w_\ell(t, x, a)\right)_+\right] &= \mathbb{E}\left[\left(\max_{t<H, \ell \leq L, a \in \mathcal{A}, x \in \mathcal{X}} \sigma_\ell(t, x, a)\frac{w_\ell(t, x, a)}{\sigma_\ell(t, x, a)}\right)_+\right] \\ &\leq \sqrt{\frac{v}{\beta}}\mathbb{E}\left[\left(\max_{t<H, \ell \leq L, a \in \mathcal{A}, x \in \mathcal{X}} \frac{w_\ell(t, x, a)}{\sigma_\ell(t, x, a)}\right)_+\right] \\ &\leq \sqrt{2v/\beta\log(1 + |\mathcal{X}||\mathcal{A}|HL)} \end{aligned}$$

where the last step uses equation (A.2). Combining these results implies,

$$\mathbb{E}[\max_{\ell \leq L, t<H} \|V_{Q_{\ell,t+1}}\|_\infty] \leq \|\bar{\theta}\|_\infty + H + H\mathbb{E}[w_{\max}] \leq \|\bar{\theta}\|_\infty + H + H\sqrt{2(v/\beta)\log(1 + |\mathcal{X}||\mathcal{A}|HL)}.$$

The result then follows by plugging in for $\beta \geq 3$, $v = 3H^2$, and $\bar{\theta} = H$. ■

A.4. Proof of Lemma 16

Lemma 3 *If $\beta \geq 2$, with probability 1,*

$$\sum_{\ell \leq L}\sum_{t \leq H} \frac{1}{\beta + n_\ell(t, x_t, a_t)} \leq H|\mathcal{X}||\mathcal{A}|\log\left(\frac{1 + L}{|\mathcal{X}||\mathcal{A}|}\right)$$

and

$$\sum_{\ell \leq L}\sum_{t \leq H} \sqrt{\frac{1}{\beta + n_\ell(t, x_t, a_t)}} \leq 2H\sqrt{|\mathcal{X}||\mathcal{A}|L}.$$

Proof Set $\mathcal{Y} = \{0, ..., H-1\} \times \mathcal{X} \times \mathcal{A}$ to be the set of valid period, state, action triples $y = (t, x, a) \in \mathcal{Y}$. Note that $|\mathcal{Y}| = H|\mathcal{X}||\mathcal{A}|$. We have

$$\begin{aligned}
\sum_{t \le H, \ell \le L} \frac{1}{\beta + n_\ell(t, x_t, a_t)} = \sum_{y \in \mathcal{Y}} \sum_{i=0}^{n_L(y)-1} \frac{1}{\beta + i} &\le \sum_{y \in \mathcal{Y}} \int_{\beta-1}^{n_L(y)+\beta-1} \frac{1}{z} dz \\
&= \sum_{y \in \mathcal{Y}} \log\left(\frac{\beta - 1 + n_L(y)}{\beta - 1}\right) \\
&\le \sum_{y \in \mathcal{Y}} \log\left(1 + n_L(y)\right) \\
&\le |\mathcal{Y}| \log\left(\frac{\sum_{y \in \mathcal{Y}}(1 + n_L(y))}{|\mathcal{Y}|}\right) \\
&= |\mathcal{Y}| \log\left(1 + \frac{LH}{|\mathcal{Y}|}\right). \\
&= H|\mathcal{X}||\mathcal{A}| \log\left(1 + \frac{L}{|\mathcal{X}||\mathcal{A}|}\right).
\end{aligned}$$

In addition

$$\begin{aligned}
\sum_{t \le H, \ell \le L} \sqrt{\frac{1}{\beta + n_\ell(t, x_t, a_t)}} = \sum_{y \in \mathcal{Y}} \sum_{i=0}^{n_L(y)-1} \sqrt{\frac{1}{\beta + i}} &\le \sum_{y \in \mathcal{Y}} \int_{z=0}^{n_L(y)} \frac{1}{(\beta - 1 + z)^{1/2}} dz \\
&\le \sum_{y \in \mathcal{Y}} \int_{z=0}^{n_L(y)} \frac{1}{z^{1/2}} dz \\
&= \sum_{y \in \mathcal{Y}} 2\sqrt{n_L(y)} \\
&\le 2\sqrt{|\mathcal{Y}| \sum_{y \in \mathcal{Y}} n_L(y)} \\
&= 2H\sqrt{|\mathcal{X}||\mathcal{A}|L}.
\end{aligned}$$

■

A.5. Proof of Lemma 9

This section establishes Lemma 9 through a sequence of results. First, Lemma 17 provides general conditions for stochastic optimism. Next, we use these properties to show a stochastic dominance relation between Dirichlet and Beta distributions in Lemma 18 and between Beta and Gaussian distributions in Lemma 19. The final proof of Lemma 9, given at the end of this section, uses a simple combination of these results.

We begin with Lemma 17, which reproduces several sufficient conditions for stochastic optimism. Conditions (1.) and (2.) are classic results in the theory of second order

stochastic dominance (SSD) and we refer to (Hadar and Russell, 1969; Hanoch and Levy, 1969) for proofs. [11] For a clever and explicit construction of coupled random variables per condition (1.) see (Machina and Pratt, 1997). In fact, both (1.) and (2.) are also necessary conditions for stochastic optimism, although our results do not rely upon this.

Condition (3.) is less widely stated, but is a simple consequence of (2.) and has been known since Hanoch and Levy (1969). To our knowledge, condition (4.) is a new result that provides a sufficient condition for stochastic optimism that is much easier to verify than condition (3.). Some intuition for this result is captured by Figure 18a. We provide a detailed proof that analytically establishes many of the properties observable in Figure 18.

Lemma 17 (Sufficient Conditions for Stochastic Optimism) *Consider two integrable random variables X and Y with probability density functions f and g. Let $F(s) = \int_{-\infty}^{s} f(x)dx$ and $G(s) = \int_{-\infty}^{s} g(x)dx$ denote the corresponding cumulative distribution functions. Then $X \succeq_{SO} Y$ if any of the following properties hold:*

1. *One can construct random variables $(\tilde{X}, \tilde{Y}, \tilde{W})$ on a joint probability space such that $\tilde{X}$ has marginal distribution F, $\tilde{Y}$ has marginal distribution G, $\mathbb{E}[\tilde{W}|\tilde{Y}] \ge 0$ and*
$$\tilde{X} = \tilde{Y} + \tilde{W}.$$
2. *For all $a \in \mathbb{R}$,*
$$\int_a^\infty (G(s) - F(s))\, ds \ge 0.$$
3. *$\mathbb{E}[X] \ge \mathbb{E}[Y]$ and there exists $a \in \mathbb{R}$ such that*
$$G(s) \ge F(s) \iff s \ge a.$$
4. *$\mathbb{E}[X] \ge \mathbb{E}[Y]$ and $C := \{x \in \mathbb{R} : g(x) > f(x)\}$ is convex.*

Proof (1.) follows easily by the tower property of conditional expectation and the conditional Jensen inequality. We have $\mathbb{E}[\tilde{X} \mid \tilde{Y}] \ge \tilde{Y}$ and hence for any convex increasing function $u : \mathbb{R} \to \mathbb{R}$,

$$\mathbb{E}[u(\tilde{X})] = \mathbb{E}[\mathbb{E}[u(\tilde{X}) \mid \tilde{Y}]] \ge \mathbb{E}[u\left(\mathbb{E}[\tilde{X} \mid \tilde{Y}]\right)] \ge \mathbb{E}[u\left(\tilde{Y}\right)].$$

(2.) follows from integration by parts and (3.) is a fairly simple consequence of (2.). See Hanoch and Levy (1969) for a proof. The claim (4.) is new and established here.

Define the function

$$D(s) = G(s) - F(s) = \int_{-\infty}^{s} (g(x) - f(x))dx$$

11. The literature on SSD generally considers *concave* and increasing utility functions whereas we consider *convex* and increasing utility functions. Results on SSD can be easily translated into results about stochastic optimism through swapping positive and negative signs.

and note $C = \{x : g(x) - f(x) > 0\} = \{x : D'(x) > 0\}$. We have $\lim_{t\to\infty} D(t) = 0$. Two distributions yielding convex C and the corresponding functions $D'(s)$ and $D(s)$ are pictured in Figure 18.

Integration by parts shows (Hanoch and Levy, 1969, Lemma 1)

$$\int_{-\infty}^{\infty} D(s)ds = \mathbb{E}[X] - \mathbb{E}[Y] \geq 0. \tag{A.3}$$

We first show $\sup\{C\} < \infty$. Note that we cannot have $C = (-\infty, \infty)$, as this would imply $\int g(x)dx > \int f(x)dx$, contradicting that g and f are probability density functions. Now suppose for contradition that $C = (\underline{c}, \infty)$ for some $\underline{c} \in \mathbb{R}$. Then, for $t \leq \underline{c}$ we have $D(t) = \int_{-\infty}^{t}(g(s) - f(s))ds < 0$. Since $\lim_{t\to\infty} D(t) = 0$, we have that for $t \geq \underline{c}$, $D(t) = -\int_t^{\infty}(g(s) - f(s))ds < 0$. Thus we have showed that $D(t) < 0$ for all $t \in \mathbb{R}$ contradicting (A.3).

Next, suppose $C = (-\infty, \overline{c})$ for $\overline{c} \in \mathbb{R}$. Then for $s \in C$, $D'(s) > 0$ by definition of C, which shows $D(t) = \int_{-\infty}^{t} D'(s)ds > 0$ for $t \in C$. But since $D'(s) < 0$ for all $s > \overline{c}$ and $\lim_{t\to\infty} D(t) = 0$, we must have $D(t) > 0$ also for all $t \geq \overline{c}$. Since D is non-negative, we have $\int_a^{\infty} D(s) \geq 0$ for all $a \in \mathbb{R}$ and hence $X \geq_{SO} Y$. [12]

Finally, suppose $C = (\underline{c}, \overline{c})$ and refer to Figures 18c and 18d for visual guidance. For $s < \underline{c}$, $D'(s) \leq 0$ and hence $D(t) = \int_{-\infty}^{t} D'(s)ds \leq 0$ for $t \leq \underline{c}$. For $s \in C$, $D'(s) > 0$. For $s > \overline{c}$, $D'(s) \leq 0$, which implies $D(t) \geq 0$ for all $t \geq \overline{c}$ since $\lim_{t\to\infty} D(t) = 0$. Since we have shown $D(\underline{c}) \leq 0 \leq D(\overline{c})$ there must exists a single crossing point $a \in (\underline{c}, \overline{c})$ with $D(a) = 0$. For this value of a, we have

$$G(s) - F(s) \iff s \geq a,$$

implying $X \geq_{SO} Y$. This completes the proof of condition (4.). ■

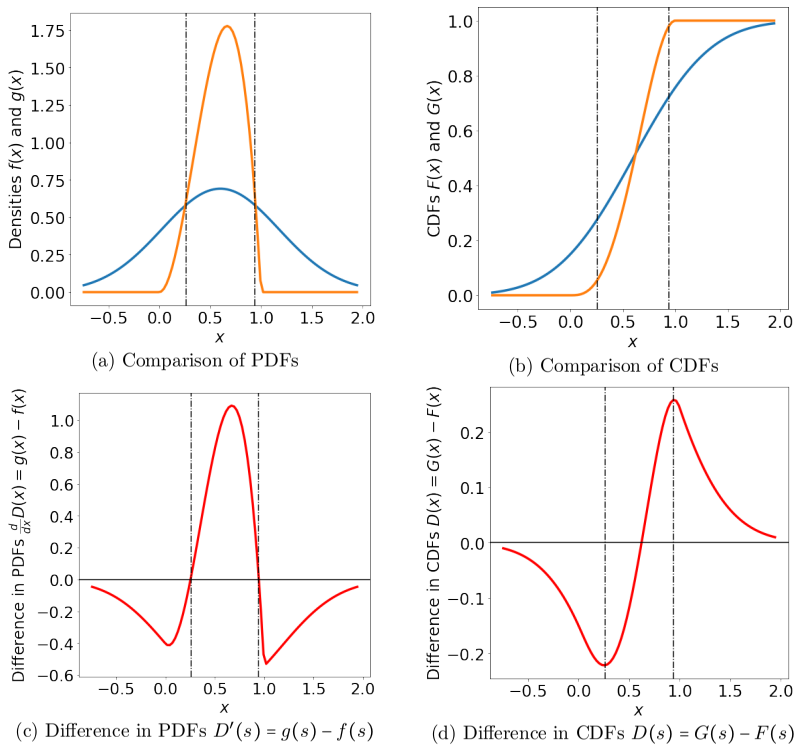
Our next result uses several relationships between Gamma, Beta, and Dirichlet random variables to establish a stochastic optimism relationship between specific matched distributions. First, for two independent Gamma distributed random variables $\gamma \sim \text{Gamma}(\alpha, 1)$ and $\gamma' \sim \text{Gamma}(\beta, 1)$, we have that $\frac{\gamma}{\gamma+\gamma'} \sim \text{Beta}(\alpha, \beta)$. Next, $\gamma + \gamma' \sim \text{Gamma}(\alpha + \beta, 1)$ and $\mathbb{E}[\gamma \mid \gamma + \gamma'] = (\gamma + \gamma')\left(\frac{\alpha}{\alpha+\beta}\right)$. Finally, for a collection of independent Gamma random variables $\gamma_i \sim \text{Gamma}(\alpha_i, 1)$, the random probability vector $P = \left(\frac{\gamma_1}{\sum_{j=1}^n \gamma_j}, \ldots \frac{\gamma_n}{\sum_{j=1}^n \gamma_j}\right)$ follows a Dirichlet(α) distribution. Lemma 18 compares the distribution of the inner product $P^\top V$ between a Dirichlet random variable P and a fixed vector V to an appropriate Beta random variable X.

Lemma 18 (Beta-Dirichlet optimism) *If $V \in \mathbb{R}^n$ where $0 = V_1 \leq \cdots \leq V_n = 1$, $P \sim$ Dirichlet(α) where $\alpha \in \mathbb{R}_+^n$, and $X \sim \text{Beta}(\sum_{i=1}^n \alpha_i V_i, \sum_{i=1}^n \alpha_i(1 - V_i))$ then $X \geq_{SO} P^\top V$.*

Proof Our proof constructs coupled Dirichlet and and Beta random variables with the marginal distributions described in the lemma's statement. Consider independent Gamma-distributed random variables $(\gamma_1^0, \gamma_1^1, \ldots, \gamma_n^0, \gamma_n^1)$ where

$$\gamma_i^0 \sim \text{Gamma}(\alpha_i V_i, 1) \quad \text{and} \quad \gamma_i^1 \sim \text{Gamma}(\alpha_i(1 - V_i), 1) \qquad i = 1, \ldots, n.$$

12. In fact, X first-order stochastically dominates Y in this case.



(a) Comparison of PDFs (b) Comparison of CDFs

(c) Difference in PDFs $D'(s) = g(s) - f(s)$ (d) Difference in CDFs $D(s) = G(s) - F(s)$

Figure 18: Comparison of Beta$(3, 2)$ and $N(3/5, 1/3)$ distributions. The dashed vertical lines indicate the boundaries of set C.

Set $\gamma_i = \gamma_i^0 + \gamma_i^1$ so that $\gamma_i \sim \text{Gamma}(\alpha_i, 1)$. Then, set

$$P = \frac{(\gamma_1, \ldots, \gamma_n)}{\sum_{i=1}^n \gamma_i} \sim \text{Dirichlet}(\alpha)$$

and

$$X = \frac{\sum_{i=1}^n \gamma_i^0}{\sum_{i=1}^n \gamma_i} = \frac{\sum_{i=1}^n \gamma_i^0}{\sum_{i=1}^n \gamma_i^0 + \sum_{i=1}^n \gamma_i^1} \sim \text{Beta}\left(\sum_{i=1}^n \alpha_i V_i, \sum_{i=1}^n \alpha_i(1 - V_i)\right).$$

Our result follows from the fact that $\sum_{i=1}^n \gamma_i^0 \sim \text{Gamma}(\sum \alpha_i V_i, 1)$ and $\sum_{i=1}^n \gamma_i^1 \sim \text{Gamma}(\sum_{i=1}^n \alpha_i(1 - V_i), 1)$ and the ratio of Gamma distributed random variables follows a Beta distribution. Now, we have that

$$\mathbb{E}[\gamma_i^0 | \gamma_i] = \gamma_i \left(\frac{\mathbb{E}[\gamma_i^0]}{\mathbb{E}[\gamma_i^0] + \mathbb{E}[\gamma_i^1]}\right) = \gamma_i \left(\frac{\alpha_i V_i}{\alpha_i}\right) = \gamma_i V_i.$$

This implies,

$$\mathbb{E}[X|\gamma] = \mathbb{E}\left[\frac{\sum_{i=1}^n \gamma_i^0}{\sum_{i=1}^n \gamma_i}\middle|\gamma\right] = \frac{\sum_{i=1}^n \mathbb{E}[\gamma_i^0 \mid \gamma_i]}{\sum_{i=1}^n \gamma_i} = \frac{\sum_{i=1}^n \gamma_i V_i}{\sum_{i=1}^n \gamma_i} = P^\top V.$$

Set $Y = P^\top V$ and $W = X - Y$ so that $X = Y + W$. We have shown $\mathbb{E}[W \mid \gamma] = 0$ and so $\mathbb{E}[W \mid Y] = \mathbb{E}[\mathbb{E}[W \mid \gamma] \mid Y] = 0$ and the result follows from part 1 of Lemma 17. ■

Our next result, Lemma 19 shows that a Gaussian distribution with sufficiently large mean and varaince stochastically dominates a Beta distribution. The proof shows this by looking at the ratio of their probability density functions and concluding this is a quasi-concave function. The basic properties of quasi-concave functions used here are in (Boyd and Vandenberghe, 2004, Section 3.4).

Lemma 19 (Gaussian-Beta optimism) *Let $X \sim N(\mu, \sigma)$ and $Y \sim \text{Beta}(\alpha, \beta)$. If $\alpha + \beta \geq 3$, $\mu = \alpha/(\alpha+\beta)$ and $\sigma^2 = (\alpha+\beta-2)^{-1}$, then $X \succeq_{SO} Y$.*

Proof Let $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$ and $g(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$ denote the probability density functions corresponding to the $N(\mu, \sigma)$ and $\text{Beta}(\alpha, \beta)$ distributions. We show $g(x)/f(x)$ is quasi-concave, which implies the super-level set $C = \{x : g(x)/f(x) \geq 1\}$ is convex. The result then follows by Lemma 17 and condition (4.).

Let $\ell(x) = \log(g(x)/f(x))$. Since $g(x)/f(x) = e^{\ell(x)}$ is a monotone function of ℓ, it is quasi-concave a long as $\ell(x)$ is quasi-concave. We show ℓ is quasi-concave by considering two cases. First, if $\alpha > 1$ and $\beta > 1$, we show ℓ is concave (and therefore quasi-concave). Differentiation shows

$$\ell''(x) = -\left(\frac{\alpha-1}{x^2} + \frac{\beta-1}{(1-x)^2}\right) + \frac{1}{\sigma^2} \leq -(\alpha+\beta-2) + \frac{1}{\sigma^2} \leq 0$$

where the last step follows from our choice of σ^2. In the case where either $\alpha \in (0,1)$ or $\beta \in (0,1)$, we find that ℓ is monotone (and therefore quasi-concave). Assume $\beta \in (0,1)$, in which case $\alpha \geq 2$. Then,

$$\ell'(x) = \frac{\alpha-1}{x} - \frac{\beta-1}{1-x} + \frac{x-\mu}{\sigma^2} \geq \frac{\alpha-1}{x} + \frac{x-\mu}{\sigma^2}.$$

This is non-negative for $x \geq \mu$. For $x < \mu$, plugging in $\mu = \alpha/(\alpha+\beta)$ and $\sigma^2 \geq 1/(\alpha+\beta)$ gives

$$\ell'(x) \geq \frac{\alpha-1}{x} + \frac{x-\mu}{\sigma^2} \geq \frac{\alpha-\alpha/2}{x} + x(\alpha+\beta) - \alpha \geq \frac{\alpha}{2x} + \alpha(x-1) \geq 0$$

where the last inequality can be verified by solving $\min_{x>0} \frac{2}{x} + x - 1 = \frac{2}{\sqrt{1/2}} + \sqrt{1/2} - 1 \geq 0$. ■

With these results we are now ready to prove Lemma 9, which we restate below.

Lemma 4 *Let $Y = P^T V$ for $V \in \mathbb{R}^n$ fixed and $P \sim \text{Dirichlet}(\alpha)$ with $\alpha \in \mathbb{R}_+^n$ and $\sum_{i=1}^n \alpha_i \geq 3$. Let $X \sim N(\mu, \sigma^2)$ with $\mu \geq \frac{\sum_{i=1}^n \alpha_i V_i}{\sum_{i=1}^n \alpha_i}$, $\sigma^2 \geq 3\left(\sum_{i=1}^n \alpha_i\right)^{-1} \text{Span}(V)^2$, then $X \succeq_{SO} Y$.*

Proof [Proof of Lemma 9] Without loss of generality, assume $V_1 \leq \ldots \leq V_n$. Set $V_i' = (V_i - V_1)/(V_n - V_1)$ for $i = 1, \ldots, n$. If $X \sim \text{Beta}(\sum_{i=1}^n \alpha_i V_i', \sum_{i=1}^n \alpha_i(1-V_i'))$, then $X \succeq_{SO} P^\top V'$ by Lemma 18. Let $Z \sim N(\mu, \sigma^2)$ where $\mu = \mathbb{E}[X] = \sum_i \alpha_i V_i'/(\sum_j \alpha_j)$ and $\sigma^2 = (\sum_{i=1}^n \alpha_i - 2)^{-1}$. Then $Z \succeq_{SO} X$ by Lemma 19. Combining these results gives

$$P^\top V = V_1 + (V_n - V_1)P^\top V' \preceq_{SO} V_1 + (V_n - V_1)X \preceq_{SO} V_1 + (V_n - V_1)Z.$$

We have that $V_1 + (V_n - V_1)Z$ is normally distributed with

$$\mathbb{E}[V_1 + (V_n - V_1)Z] = V_1 + (V_n - V_1)\mu = V_1 + (V_n - V_1)\left(\frac{\sum_i \alpha_i V_i'}{\sum_j \alpha_j}\right) = \frac{\sum_{i=1}^n \alpha_i V_i}{\sum_{i=1}^n \alpha_i}$$

and

$$\text{Variance}\,(V_1 + (V_n - V_1)Z) = (V_n - V_1)^2\sigma^2 = \text{Span}(V)^2\left(\sum_{i=1}^n \alpha_i - 2\right)^{-1} \leq 3 \cdot \text{Span}(V)^2\left(\sum_{i=1}^n \alpha_i\right)^{-1}.$$

■

References

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. *Journal of Machine Learning Research - Proceedings Track*, 19: 1–26, 2011.

Sander Adam, Lucian Busoniu, and Robert Babuska. Experience replay for real-time reinforcement learning control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):201–212, 2012.

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *Artificial Intelligence and Statistics*, pages 99–107, 2013a.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th Annual International Conference on Machine Learning*, pages 127–135, 2013b.

Peter Auer and Ronald Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems 19*, pages 49–56, 2006.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th Annual International Conference on Machine Learning*, 2017.

Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through Bayesian deep q-networks. *arXiv preprint arXiv:1802.04412*, 2018.

Peter L. Bartlett and Ambuj Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI2009)*, pages 35–42, June 2009.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30*, pages 6241–6250, 2017.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems 29*, pages 1471–1479. 2016.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Advances in Neural Information Processing Systems 30*, 2017.

Dimitri P. Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, September 1996.

Peter J Bickel and David A Freedman. Some asymptotic theory for the bootstrap. *The Annals of Statistics*, pages 1196–1217, 1981.

David Blackwell. Discounted dynamic programming. *The Annals of Mathematical Statistics*, 36(1):226–235, 1965.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems 28*, pages 2818–2826. 2015.

Richard Dearden, Nir Friedman, and Stuart J. Russell. Bayesian Q-learning. In *AAAI Conference on Artificial Intelligence*, pages 761–768, 1998.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Dean Eckles and Maurits Kaptein. Bootstrap thompson sampling and sequential decision problems in the behavioral sciences. *SAGE Open*, 9(2):2158244019851675, 2019.

Bradley Efron. *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM, 1982.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *International Conference on Learning Representations*, 2018.

Tadayoshi Fushiki. Bootstrap prediction and bayesian prediction under misspecified models. *Bernoulli*, pages 747–758, 2005.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th international conference on artificial intelligence and statistics*, pages 249–256, 2010.

Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized Markov decision processes. In *Proceedings of the 28th Annual Conference on Learning Theory*, 2015.

Josef Hadar and William R Russell. Rules for ordering uncertain prospects. *The American Economic Review*, pages 25–34, 1969.

G Hanoch and H Levy. The efficiency analysis of choices involving risk. *The Review of Economic Studies*, 36(3):335–346, 1969.

Imanol Arrieta Ibarra, Bernardo Ramos, and Lars Roemheld. Angrier birds: Bayesian reinforcement learning. *arXiv preprint arXiv:1601.01297*, 2016.

Morteza Ibrahimi, Adel Javanmard, and Benjamin V Roy. Efficient reinforcement learning for high dimensional linear quadratic systems. In *Advances in Neural Information Processing Systems 25*, pages 2636–2644, 2012.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Sham Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

Michael J. Kearns and Daphne Koller. Efficient reinforcement learning in factored MDPs. In *IJCAI*, pages 740–747, 1999.

Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Lihong Li and Michael L Littman. Reducing reinforcement learning to kwik online regression. *Annals of Mathematics and Artificial Intelligence*, 58(3-4):217–237, 2010.

Lihong Li, Michael L Littman, and Thomas J Walsh. Knows what it knows: a framework for self-aware learning. In *Proceedings of the 25th international conference on Machine learning*, pages 568–575. ACM, 2008.

Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. BBQ-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. *AAAI Conference on Artificial Intelligence*, 2018.

Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3258–3266. 2017.

Mark Machina and John Pratt. Increasing risk: some direct constructions. *Journal of Risk and Uncertainty*, 14(2):103–127, 1997.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems 29*, pages 1046–1054, 2016.

Brendan O'Donoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. The uncertainty Bellman equation and exploration. In *Proceedings of the 35th Annual International Conference on Machine Learning*, 2017.

Ronald Ortner and Daniil Ryabko. Online regret bounds for undiscounted continuous reinforcement learning. In *Advances in Neural Information Processing Systems 25*, pages 1763–1771, 2012.

Ian Osband. *Deep Exploration via Randomized Value Functions*. PhD thesis, Stanford University, 2016.

Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems 27*, pages 1466–1474, 2014a.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored MDPs. In *Advances in Neural Information Processing Systems 27*, pages 604–612, 2014b.

Ian Osband and Benjamin Van Roy. Bootstrapped Thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *Proceedings of the 34th International Conference on Machine Learning*, pages 2701–2710, 2017.

Ian Osband, Dnaiel Russo, and Benjamin Van Roy. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems 26*, pages 3003–3011. 2013.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances In Neural Information Processing Systems 29*, pages 4026–4034, 2016a.

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2377–2386, 2016b.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8625–8637. 2018.

Art B Owen and Dean Eckles. Bootstrapping data arrays of arbitrary order. *The Annals of Applied Statistics*, pages 895–927, 2012.

Jason Pazis and Ronald Parr. PAC optimal exploration in continuous space Markov decision processes. In *AAAI Conference on Artificial Intelligence*. Citeseer, 2013.

Matthias Plappert. *Parameter Space Noise for Exploration in Deep Reinforcement Learning*. PhD thesis, Karlsruhe Institute of Technology, 2017.

Warren Powell and Ilya Ryzhov. *Optimal Learning*. John Wiley and Sons, 2011.

Doina Precup, Richard Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *Proceedings of The 18th International Conference on Machine Learning*, pages 417–424, 2001.

Daniel Russo. Worst-case regret bounds for exploration via randomized value functions. *arXiv preprint arXiv:1906.02870*, 2019.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems 26*, pages 2256–2264. 2013.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014a.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems 27*, pages 1583–1591. 2014b.

Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *arXiv preprint arXiv:1511.05219*, 2015.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on Thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1): 1–96, 2018.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *CoRR*, abs/1511.05952, 2015.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Alexander L Strehl. *Probably approximately correct (PAC) exploration in reinforcement learning*. PhD thesis, Rutgers University-Graduate School-New Brunswick, 2007.

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *Proceedings of the 23rd Annual International Conference on Machine Learning*, pages 881–888, 2006.

Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction, Second Edition*. MIT Press, 2018.

Richard Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000. ACM, 2009.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. *CoRR*, abs/1611.04717, 2016.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

Gerald Tesauro. Temporal difference learning and TD-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

Zheng Wen. *Efficient reinforcement learning with value function generalization*. PhD thesis, Stanford University, 2014.

Zheng Wen and Benjamin Van Roy. Efficient exploration and value function generalization in deterministic systems. In *Advances in Neural Information Processing Systems 26*, pages 3021–3029, 2013.

Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. Natural evolution strategies. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 3381–3387. IEEE, 2008.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016.

Journal of Machine Learning Research 20 (2019) 1-5 Submitted 6/18; Revised 8/19; Published 8/19

ORCA: A Matlab/Octave Toolbox for Ordinal Regression

Javier Sánchez-Monedero SANCHEZ-MONEDEROJ@CARDIFF.AC.UK
School of Journalism, Media and Culture
Cardiff University
Cardiff CF10 1FS, Wales, United Kingdom

Pedro A. Gutiérrez PAGUTIERREZ@UCO.ES
Department of Computer Science and Numerical Analysis
University of Córdoba
Córdoba, 14071, Spain

María Pérez-Ortiz MARIA.PEREZ@UCL.AC.UK
Department of Computer Science
University College London
London WC1E 6EA, United Kingdom

Editor: Cheng Soon Ong

Abstract

Ordinal regression, also named ordinal classification, studies classification problems where there exist a natural order between class labels. This structured order of the labels is crucial in all steps of the learning process in order to take full advantage of the data.

ORCA (Ordinal Regression and Classification Algorithms) is a Matlab/Octave framework that implements and integrates different ordinal classification algorithms and specifically designed performance metrics. The framework simplifies the task of experimental comparison to a great extent, allowing the user to: (i) describe experiments by simple configuration files; (ii) automatically run different data partitions; (iii) parallelize the executions; (iv) generate a variety of performance reports and (v) include new algorithms by using its intuitive interface. Source code, binaries, documentation, descriptions and links to data sets and tutorials (including examples of educational purpose) are available at `https://github.com/ayrna/orca`.

Keywords: Ordinal regression, ordinal classification, Matlab, Octave, threshold models

1. Introduction

The terms ordinal regression and ordinal classification refer to those supervised learning problems where labels show an ordinal arrangement (Gutiérrez et al., 2016), e.g. in an age estimation problem where the categories are: {*baby, child, teenager, adult*}. The aim in this case is not just to improve standard accuracy, but to reduce the magnitude of misclassification errors, in such a way that the order relation between labels is considered during model construction and evaluation. In the problem of age estimation, if true label is *baby*, predicted label *adult* should entail a more severe misclassification error than predicted label *child*. Exploiting the ordinal disposition of the labels has proven to yield better performance than simply treating them as nominal categories.

©2019 Javier Sánchez-Monedero, Pedro A. Gutiérrez, and María Pérez-Ortiz.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/18-349.html`.

In this paper, we introduce ORCA (Ordinal Regression and Classification Algorithms), a Matlab/Octave framework that gathers an extensive collection of recent ordinal machine learning methods and ordinal performance metrics. It also presents a general framework to automate experiments that can be used both from an API or by describing experiments with configuration files. Online documentation includes extensive tutorials to introduce users to ordinal classification and the pipeline of the experimental framework. We also provide several small data sets for code testing purposes and a list of 44 ordinal data sets of different characteristics that can be used for algorithm comparison.

There are some alternative software toolboxes that implement certain features for ordinal regression. Those include: i) `mord`[1] in Python, ii) `ordinal`[2] in R, iii) `vgam`[3] in R, iv) `bmrm`[4] in R and v) `ocapis`[5] in Scala. `mord` and `ordinal` focus on well-established simple statistical methods (i.e. ordinal logistic regression, which is also included in our toolbox) rather than on novel machine learning approaches. `vgam` and `bmrm` focus on vector generalised linear and additive models and regularised empirical minimisation respectively, having a setting to deal with ordinal classification problems. Finally, `ocapis` implements 4 of the 15 methods that ORCA includes but lacks the experimental and parallelisation framework.

2. Architecture and Features

The main features can be grouped in the following categories:

Methods. The `Algorithm` class defines an API including different methods such as `fit` and `predict`, which process a pair of train and test partitions for a specific hyper-parameter configuration of the selected classifier. Adding a new method is easy, the user only needs to add a new class implementing the `fit` and `predict` methods.

Metrics. All metrics implement a `calculateMetric` method, which can be run using the true and predicted labels or the confusion matrix.

Automatic experiment running. ORCA automates the process of running one or many methods for a set of data sets. It processes every experiment by fitting a model with the training data (including model selection through cross-validation), evaluating the test error and creating performance reports of all the performance metrics, training time and hyper-parameter values.

Experiment configuration. ORCA can be used by directly interacting with the API or by describing experiments with configuration files in INI format. ORCA checks the consistency of these files by analysing the corresponding algorithm class, in such a way that, if a new method is developed, the researcher does not need to modify the INI parser.

Experiment parallelisation. When presenting new classifier proposals, researchers typically run several methods over a set of partitions. For example, if we perform 30 repetitions of a hold-out design for 10 data sets, we need at least 300 calls to the `fit` and `predict`

1. `https://pythonhosted.org/mord/`
2. `https://cran.r-project.org/web/packages/ordinal/index.html`
3. `https://cran.r-project.org/web/packages/VGAM/index.html`
4. `https://cran.r-project.org/web/packages/bmrm/index.html`
5. `https://arxiv.org/abs/1810.09733`

Method	Reference
Ordinal methods	
Support vector regression (SVR)	Gutiérrez et al. (2016)
Cost-sensitive support vector classifier (CSSVC)	Hsu and Lin (2002)
Support vector machine with ordered partition (SVMOP)	Waegeman and Boullart (2009)
Ordinal extreme learning machine (ELMOP)	Deng et al. (2010)
Linear logistic regression for ordinal data (POM)	McCullagh (1980)
Ordinal SVM with explicit constraints (SVOREX)	Chu and Keerthi (2007)
Ordinal SVM with implicit constraints (SVORIM)	Chu and Keerthi (2007)
Linear SVORIM	Chu and Keerthi (2007)
Kernel discriminant analysis for ordinal regression (KDLOR)	Sun et al. (2010)
Neural network based on the POM (NNPOM)	Mathieson (1996)
Neural network with ordered partitions (NNOP)	Cheng et al. (2008)
Reduction applied to SVM (REDSVM)	Lin and Li (2012)
Ordinal regression boosting (ORBoost)	Lin and Li (2006)
Ordinal projection based ensemble (OPBE)	Pérez-Ortiz et al. (2014)
Partial order methods	
Hierarchical Partial Order Label Decomposition (HPOLD)	Sánchez-Monedero et al. (2018)
Nominal methods	
Multi-class nominal SVM with 1vs1 formulation (SVC1V1)	Hsu and Lin (2002)
Multi-class nominal SVM with 1vsAll formulation (SVC1VA)	Hsu and Lin (2002)
Matlab wrapper for LIBLINEAR (LIBLINEAR)	Fan et al. (2008)

Table 1: Ordinal and nominal methods available in ORCA.

functions, which can be run in parallel. ORCA exploits this by using Matlab and Octave parallelisation toolboxes. In addition, a set of scripts is provided to perform the parallelisation in the HTCondor[6] distributed computing environment.

3. Implemented Methods and Performance Metrics

ORCA collects an extensive list of ordinal classification methods including naïve approaches, ordinal binary decompositions and threshold models (see Table 1). Further details of the methods can be found in Gutiérrez et al. (2016), together with running time of the algorithms. From this analysis, it was concluded that ELMOP, SVORLin and POM are the best option if computational cost is a priority. The training time of neural network methods (NNPOM and NNOP) and GPOR is generally the highest. This cost can be assumed for GPOR, since it obtains very good performance for balanced ordinal data sets, while neural network-based methods are generally beaten by the ordinal SVM variants. Concerning scalability, the experimental setup in Gutiérrez et al. (2016) also included some relatively large data sets, so the practitioner could check the time it took to train one of those models with the ORCA framework. In general, linear models such as POM and SVORLin perform very well in these scenarios where there is plenty of data while still having a reasonably low

6. http://research.cs.wisc.edu/htcondor/

running time (e.g. around 10 seconds for cross-validating, training and testing on a data set of almost 22.000 patterns).

ORCA provides a collection of performance metrics, which can also be used for hyper-parameter selection. Given that ordinal problems usually show a skewed class distribution specific imbalance classification metrics are also included in the framework. For more details, we refer the reader to the project documentation and ordinal metrics review in (Cruz-Ramírez et al., 2014).

4. Sample Code

The ORCA API includes methods to directly configure, train and test an ordinal classifier:

```
% Create an Algorithm object
addpath('src/Algorithms/')
kdlorAlgorithm = KDLOR();
% Load data set
load exampledata/1-holdout/toy/matlab/train_toy.0
load exampledata/1-holdout/toy/matlab/test_toy.0
train.patterns = train_toy(:,1:(size(train_toy,2)-1));
train.targets = train_toy(:,size(train_toy,2));
test.patterns = test_toy(:,1:(size(test_toy,2)-1));
test.targets = test_toy(:,size(test_toy,2));
% Fit the model and predict with test data
result = kdlorAlgorithm.fitpredict(train,test);
% Evaluate performance metrics
addpath('src/Measures/')
CCR.calculateMetric(result.predictedTest,test.targets)
MAE.calculateMetric(result.predictedTest,test.targets)
```

Moreover, as previously discussed, ORCA can be used to run a batch of experiments specified in an INI file:

```
Utilities.runExperiments('tutorial/config-files/pom.ini')
```

Acknowledgments

We would like to thank Professor Peter Tiňo and the three anonymous reviewers for their valuable feedback. Javier Sánchez-Monedero and María Pérez-Ortiz started the development of this software while working at the University of Córdoba (Spain). This work has been subsidized by the projects TIN2017-85887-C2-1-P and TIN2017-90567-REDT of the Spanish Ministry of Economy and Competitiveness (MINECO) and EU FEDER funds.

References

Jianlin Cheng, Zheng Wang, and Gianluca Pollastri. A neural network approach to ordinal regression. In *IEEE International Joint Conference on Neural Networks*, pages 1279–1284. IEEE Press, 2008.

Wei Chu and S. Sathiya Keerthi. Support Vector Ordinal Regression. *Neural Computation*, 19(3):792–815, 2007.

Manuel Cruz-Ramírez, César Hervás-Martínez, Javier Sánchez-Monedero, and Pedro A. Gutiérrez. Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. *Neurocomputing*, 135:21–31, July 2014.

Wan-Yu Deng, Qing-Hua Zheng, Shiguo Lian, Lin Chen, and Xin Wang. Ordinal extreme learning machine. *Neurocomputing*, 74(1–3):447–456, 2010.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Pedro A. Gutiérrez, María Pérez-Ortiz, Javier Sánchez-Monedero, Francisco Fernandez-Navarro, and César Hervás-Martínez. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146, 2016.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multi-class support vector machines. *IEEE Transaction on Neural Networks*, 13(2):415–425, 2002.

Hsuan-Tien Lin and Ling Li. Large-margin thresholded ensembles for ordinal regression: Theory and practice. In José L. Balcázar, Philip M. Long, and Frank Stephan, editors, *International Conference on Algorithmic Learning Theory*, volume 4264, pages 319–333. Springer-Verlag, October 2006.

Hsuan-Tien Lin and Ling Li. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367, 2012.

Mark J. Mathieson. Ordinal models for neural networks. In J. Moody A.-P. N. Refenes, Y. Abu-Mostafa and A. Weigend, editors, *International Conference on Neural Networks in the Capital Markets*, pages 523–536. World Scientific, 1996.

Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):109–142, 1980.

María Pérez-Ortiz, Pedro A. Gutiérrez, and César Hervás-Martínez. Projection-based ensemble learning for ordinal regression. *IEEE Transactions on Cybernetics*, 44(5):681–694, May 2014.

Javier Sánchez-Monedero, María Pérez-Ortiz, Aurora Saez, Pedro A. Gutiérrez, and César Hervás-Martínez. Partial order label decomposition approaches for melanoma diagnosis. *Applied Soft Computing*, 64:341–355, March 2018.

Bing-Yu Sun, Jiuyong Li, Desheng Dash Wu, Xiao-Ming Zhang, and Wen-Bo Li. Kernel discriminant learning for ordinal regression. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):906–910, 2010.

Willem Waegeman and Luc Boullart. An ensemble of weighted support vector machines for ordinal regression. *International Journal of Computer Systems Science and Engineering*, 3(1):47–51, 2009.

Journal of Machine Learning Research 20 (2019) 1-45 Submitted 6/18; Revised 3/19; Published 7/19

Learning Representations of Persistence Barcodes

Christoph D. Hofer CHR.DAV.HOFER@GMAIL.COM
Department of Computer Science
University of Salzburg
Salzburg, Austria

Roland Kwitt ROLAND.KWITT@GMAIL.COM
Department of Computer Science
University of Salzburg
Salzburg, Austria

Marc Niethammer MN@CS.UNC.EDU
Department of Computer Science
University of North Carolina
Chapel Hill, NC, USA

Editor: Michael Mahoney

Abstract

We consider the problem of supervised learning with summary representations of topological features in data. In particular, we focus on persistent homology, the prevalent tool used in topological data analysis. As the summary representations, referred to as barcodes or persistence diagrams, come in the unusual format of multi sets, equipped with computationally expensive metrics, they can not readily be processed with conventional learning techniques. While different approaches to address this problem have been proposed, either in the context of kernel-based learning, or via carefully designed vectorization techniques, it remains an open problem how to leverage advances in representation learning via deep neural networks. Appropriately handling topological summaries as input to neural networks would address the disadvantage of previous strategies which handle this type of data in a *task-agnostic* manner. In particular, we propose an approach that is designed to learn a *task-specific* representation of barcodes. In other words, we aim to learn a representation that adapts to the learning problem while, at the same time, preserving theoretical properties (such as stability). This is done by projecting barcodes into a finite dimensional vector space using a collection of parametrized functionals, so called structure elements, for which we provide a generic construction scheme. A theoretical analysis of this approach reveals sufficient conditions to preserve stability, and also shows that different choices of structure elements lead to great differences with respect to their suitability for numerical optimization. When implemented as a neural network input layer, our approach demonstrates compelling performance on various types of problems, including graph classification and eigenvalue prediction, the classification of 2D/3D object shapes and recognizing activities from EEG signals.

Keywords: Topological data analysis, persistent homology, topological summary, supervised learning, deep learning

©2019 Christoph D. Hofer, Roland Kwitt, Marc Niethammer.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-358.html.

1. Introduction

Over the past decade, concepts from the field of algebraic topology have evolved into computationally efficient algorithms to analyze data. Methods from this field are now succinctly summarized under the term *topological data analysis (TDA)* (Carlsson, 2009) and have found a broad range of applications, ranging from studying activity patterns of the visual cortex (Singh et al., 2008), breast cancer (Nicolau et al., 2011), the manifold of natural image patches (Carlsson et al., 2008), to analyzing brain artery trees (Bendich et al., 2016), 3D surfaces (Reininghaus et al., 2015; Li et al., 2014; Hofer et al., 2017b), clustering (Chazal et al., 2013b) and the recognition of 2D object shapes (Turner et al., 2014b). Arguably, the most prevalent method used in practice is *persistent homology* (see Edelsbrunner et al., 2002; Zomorodian and Carlsson, 2004) which offers a concise summary representation of topological features in data. In short, persistent homology tracks topological changes as we analyze data at multiple "scales". As the scale changes, topological features (i.e., connected components, holes, etc.) appear and disappear. Persistent homology associates a lifespan to these features, resulting in a multi set of $(birth, death)$ tuples, typically visualized as a *barcode* or a *persistence diagram*. These *topological signatures* can offer complementary information that is not easily extractable by other methods. This opens up novel pathways to address learning problems based on topological information.

Despite the advantages of TDA in capturing topological invariants of data, the field is still largely disconnected from recent developments in machine learning. With respect to persistent homology, this can be attributed to the unusual data structure underlying topological signatures, that is, multi sets, and the associated (computationally intensive) metrics in that space. In fact, Mileyko et al. (2011) and Turner et al. (2014a) have investigated the theoretical properties of this metric space and shown that it is not easily amenable to statistical computations (for example, there is no unique Fréchet mean), or machine learning for that matter. Nevertheless, several works (see Section 3) have recently shown advances towards bridging the gap between machine learning and TDA; either in the context of kernel-based learning, or via suitable vectorization techniques for persistence diagrams / barcodes, based on algebraic ideas. However, both strategies typically come at the cost of computational complexity. In case of vectorization techniques, computational bottlenecks typically arise when computing the vectorization itself. In case of kernels, it is well-known that kernel-based learning (such as SVMs) scales poorly with sample size and sometimes even the kernel computation itself can be computationally challenging. Importantly, both strategies rely on an a-priori *fixed* representation of topological signatures.

With respect to the latter issue, the success of deep neural networks in vision (see Krizhevsky et al., 2012; He et al., 2016; Huang et al., 2017), or natural language processing (see Graves, 2013; Sutskever et al., 2014), has shown that it is preferable to learn *task-specific representations*, instead of hand-crafting them. While neural networks exhibit remarkable performance on many types of problems, input data with strong geometric structure, such as graphs or manifold-valued objects, pose considerable algorithmic and theoretical challenges. Topological summaries fall exactly into this category because of their nature as multi sets and the associated metrics. This has, so far, largely prevented principled approaches to use barcodes as input(s) to neural networks.

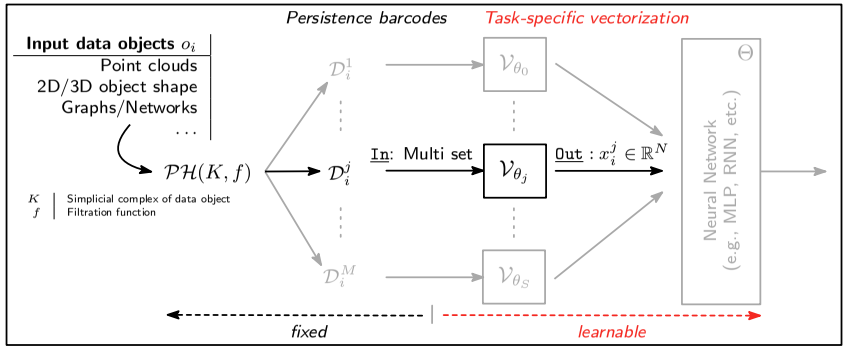


Figure 1: Illustration of *learnable, task-specific vectorizations of barcodes*. Inputs are barcodes $\mathcal{D}_i^j$, obtained by (1) constructing a simplicial complex K from data object o_i and (2) computing persistent homology ($\mathcal{PH}$) of a filtration of K. The superscript j in $\mathcal{D}_i^j$ denotes that we can have multiple barcodes per o_i, e.g., considering homology groups of different dimension, or from multiple filtrations. Each barcode is a *multi set*, fed to one (or more) parametrized (by θ_j) input layers that implement a mapping $\mathcal{V}_{\theta_j}$ to $\mathbb{R}^N$. These vectorizations are then fed to a neural network implementing, e.g., a discriminant classifier.

Contributions. This work extends Hofer et al. (2017a), where we introduced a neural network layer that can handle barcodes in a principled manner. The core idea is to project points in a barcode by a collection of parametrized functionals, so called structure elements. The parametrization is learned during training and allows to obtain a task-specific vectorization of barcodes (see Figure 1). In this work, we conduct an in-depth theoretical analysis of this approach. In particular, we prove that such a construction leads to an induced mapping of barcodes that is continuous with respect to the p-Wasserstein distance, although it *cannot* be stable with respect to p-Wasserstein for $p > 1$. This complements a similar recent result of Reininghaus et al. (2015) in the context of kernels. Further, we show that the original strategy of Hofer et al. (2017a) to enforce the required conditions on structure elements has limitations and present an alternative allowing a broader range of functional families to be considered. This then enables us to introduce new structure elements with desirable properties for learning. Finally, we present experiments on a variety of problems, including graph classification, eigenvalue prediction of (normalized) graph Laplacian matrices, 2D/3D shape recognition and the classification of EEG signals.

Organization. Section 2 introduces the required background material on persistent homology. Section 3 reviews related work on machine learning with topological signatures and highlights connections to our approach. Section 4 presents our main analysis of the construction scheme and Section 5 discusses its practical and theoretical aspects. Finally, Section 6 presents experimental results and Section 7 concludes the paper with a discussion of the most relevant points.

2. Background

For brevity, we only provide a brief overview of the mathematical concepts relevant to this work and refer the reader to Hatcher (2002) or Edelsbrunner and Harer (2010) for details.

Homology. The key idea of homology theory is to study the properties of some object X by means of (commutative) algebra. In particular, we assign to X a sequence of modules C_n which are connected by homomorphisms ∂_n, that is,

$$\cdots \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \mathbf{0}$$

with

$$\partial_n : C_n \to C_{n-1} \quad \text{such that} \quad \operatorname{im} \partial_n \subseteq \ker \partial_{n-1} \ .$$

A structure of this form is called a *chain complex* and by studying its homology groups

$$H_n = \ker \partial_n / \operatorname{im} \partial_{n+1}$$

we can derive properties of X.

A prominent example of a homology theory is *simplicial homology*. Throughout this work, it is the used homology theory and hence we will now concretize the already presented ideas. Let K be a simplicial complex and K_n its n-skeleton. Then, we set $C_n(K)$ as the vector space generated (freely) by K_n over $\mathbb{Z}/2\mathbb{Z}$[1]. The connecting homomorphisms

$$\partial_n : C_n(K) \to C_{n-1}(K)$$

are called boundary operators. For a n-simplex $\sigma = [x_0, \ldots, x_n] \in K_n$, we define them as

$$\partial_n(\sigma) = \sum_{i=0}^{n} [x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$$

and linearly extend this to $C_n(K)$, that is, $\partial_n(\sum \sigma_i) = \sum \partial_n(\sigma_i)$.

Persistent homology. Let K be a simplicial complex. A sequence of simplicial complexes, $(K^i)_{i=0}^m$, such that

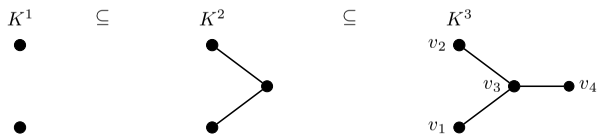
$$\emptyset = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^m = K$$

is called a *filtration* of K. If we use the extra information provided by the filtration of K, we obtain the following sequence of chain complexes

$$\begin{tikzcd}
\cdots \arrow[r, "\partial_3"] & C_2^1 \arrow[r, "\partial_2"] \arrow[d, "\iota"] & C_1^1 \arrow[r, "\partial_1"] \arrow[d, "\iota"] & C_0^1 \arrow[r, "\partial_0"] \arrow[d, "\iota"] & \mathbf{0} \\
\cdots \arrow[r, "\partial_3"] & C_2^2 \arrow[r, "\partial_2"] \arrow[d, dotted, "\iota"] & C_1^2 \arrow[r, "\partial_1"] \arrow[d, dotted, "\iota"] & C_0^2 \arrow[r, "\partial_0"] \arrow[d, dotted, "\iota"] & \mathbf{0} \\
\cdots \arrow[r, "\partial_3"] & C_2^m \arrow[r, "\partial_2"] & C_1^m \arrow[r, "\partial_1"] & C_0^m \arrow[r, "\partial_0"] & \mathbf{0}
\end{tikzcd}$$

1. Simplicial homology is not specific to $\mathbb{Z}/2\mathbb{Z}$, but it's a typical choice, since it allows us to interpret n-chains as sets of n-simplices.

where $C_n^i = C_n(K_n^i)$, ι denotes the inclusion and $\mathbf{0}$ is the trivial group. To illustrate this more conveniently, we provide a simple *example*:



The chain groups in this example are as follows:

For K^1	For K^2	For K^3
$C_0^1 = [[v_1],[v_2]]_{\mathbb{Z}_2}$	$C_0^2 = [[v_1],[v_2],[v_3]]_{\mathbb{Z}_2}$	$C_0^3 = [[v_1],[v_2],[v_3],[v_4]]_{\mathbb{Z}_2}$
$C_1^1 = \mathbf{0}$	$C_1^2 = [[v_1,v_3],[v_2,v_3]]_{\mathbb{Z}_2}$	$C_1^3 = [[v_1,v_3],[v_2,v_3],[v_3,v_4]]_{\mathbb{Z}_2}$
$C_2^1 = \mathbf{0}$	$C_2^2 = \mathbf{0}$	$C_2^3 = \mathbf{0}$

This then leads to the concept of *persistent homology groups*, defined by

$$H_n^{i,j} = \ker \partial_n^i / (\operatorname{im} \partial_{n+1}^j \cap \ker \partial_n^i) \quad \text{for} \quad i \leq j \ .$$

The ranks, $\beta_n^{i,j} = \operatorname{rank} H_n^{i,j}$, of these homology groups (that is, the *n-th persistent Betti numbers*), capture the number of homological features of dimensionality n (for example, connected components for $n = 0$, holes for $n = 1$, etc.) that persist from i to (at least) j. In fact, according to (Edelsbrunner and Harer, 2010, Fundamental Lemma of Persistent Homology), the quantities

$$\mu_n^{i,j} = (\beta_n^{i,j-1} - \beta_n^{i,j}) - (\beta_n^{i-1,j-1} - \beta_n^{i-1,j}) \quad \text{for} \quad i < j \tag{1}$$

encode all the information about the persistent Betti numbers of dimension n. Before we continue to introduce barcodes, we take a short detour and discuss the concept of multi sets, a data structure that appears naturally when encoding information captured by persistent homology.

Multi sets. We focus on multi sets over a fixed domain. Informally, multi sets extend the concept of ordinary sets by allowing elements to occur more than once. For example, in the multi set $M = \{1, 1, 2\}$, the element 1 is contained 2 times. For multi sets over a fixed domain, D, this concept can be formalized by following the idea that *ordinary* sets over a fixed domain can be represented by their indicator function. Analogously, we can represent a multi set by its multiplicity function. Hence, a multi set M can be interpreted as a function

$$M : D \to \mathbb{N}_0^\infty = \mathbb{N} \cup \{0, \infty\} \ ,$$

where $M(x) = n$ denotes that x is contained in M n-times. However, we will write

$$\operatorname{mult}_M(x) \text{ instead of } M(x)$$

to address the multiplicity of x in M, since it is not common to write sets as functions. Based on this representation, we introduce the multi set operations needed in the subsequent parts of this work.

Definition 1 *Let M, N be multi sets and A an ordinary set over the domain D. We adhere to the following conventions:*

(M1) $\operatorname{supp}(M) = \operatorname{supp}(\operatorname{mult}_M) = \{x \in D : \operatorname{mult}_M(x) > 0\}$ — *Support*

(M2) $x \in M \Leftrightarrow x \in \operatorname{supp}(M)$ — *Set membership*

(M3) $|M| = \sum_{x \in \operatorname{supp}(M)} \operatorname{mult}_M(x)$ — *Cardinality*

(M4) $(M \uplus N)(x) = \operatorname{mult}_{M \uplus N}(x) = \operatorname{mult}_M(x) + \operatorname{mult}_N(x)$ — *Union*

(M5) $(M \cap A)(x) = \operatorname{mult}_{M \cap A}(x) = \operatorname{mult}_M(x) \cdot \mathbb{1}_A(x)$ — *Intersection with set*

(M6) $(M \setminus A)(x) = \operatorname{mult}_{M \setminus A}(x) = \operatorname{mult}_M(x) \cdot \mathbb{1}_{D \setminus A}(x)$ — *Difference to set*

(M7) Let $f : D \to \mathbb{R}^n$; then, $\sum_{x \in M} f(x) = \sum_{x \in \operatorname{supp}(M)} \operatorname{mult}_M(x) \cdot f(x)$ — *Function evaluation*

If necessary, we can always interpret an ordinary set as a multi set, since the characteristic function can be interpreted as a multiplicity function with values in $\{0, 1\}$.

Topological signatures. A typical way to obtain a filtration of a simplicial complex K is to consider sub level sets of a real valued function, $f : K \to \mathbb{R}$, defined on K such that for $\rho \in K$ and the corresponding boundary operator, ∂, it holds that

$$f(\sigma) \leq f(\rho) \text{ for } \sigma \in \partial(\rho) \ .$$

Let

$$a_1 < \cdots < a_m \text{ with } m = |f(K)|$$

be the sorted sequence of values of $f(K)$. Then, we obtain $(K^i)_{i=0}^m$ by setting

$$K^0 = \emptyset \text{ and } K^i = f^{-1}\big((-\infty, a_i]\big) \text{ for } 1 \leq i \leq m \ .$$

If we construct a multi set such that, for $i < j$, the point (a_i, a_j) is inserted with multiplicity $\mu_n^{i,j}$, see Eq. (1), we effectively encode the persistent homology of dimension n with respect to the sub level set filtration induced by f.

Definition 2 (Barcode) *Let $\Omega = \{(b, d) \in \mathbb{R}^2 : d > b\}$ the upper-diagonal part of the real plane. A* barcode, *$\mathcal{D}$, is a multi set, over the domain Ω. We denote by $\mathbb{D}$ the set of all barcodes with* finite *cardinality.*

Remark 3 *While, in theory, barcodes are not necessarily finite, in any practical setup where data is analyzed, we deal with barcodes of finite cardinality. Thus, in all subsequent parts of this work, we assume $\mathcal{D} \in \mathbb{D}$.*

For a given complex K of dimension $n_{\max}$ and a function f (of the discussed form), we can interpret persistent homology ($\mathcal{PH}$) as a mapping

$$(K, f) \xrightarrow{\mathcal{PH}} (\mathcal{D}^0, \ldots, \mathcal{D}^{n_{\max}-1}) \ ,$$

where $\mathcal{D}^i$ is the barcode of dimension i and $n_{\max}$ is the dimension of K.

Equipping $\mathbb{D}$ with a metric structure, allows studying the sensitivity of $\mathcal{PH}$ to perturbations in K or f. Before we can give the definitions of the metrics usually used for this purpose, we have to introduce the concept of *relative bijective matchings* between two multi sets.

Definition 4 (Δ-relative bijective matchings between finite multi sets) *Let M, N be finite multi sets over the domain D. Further, let $\Delta \subset D$ such that*

$$\operatorname{supp}(M) \cap \Delta = \emptyset \text{ and } \operatorname{supp}(N) \cap \Delta = \emptyset \ .$$

A Δ-relative bijective matching *between M, N is a multi set, φ, over $D \times D$ such that*

(C1) $\operatorname{supp}(\varphi) \cap (\Delta \times \Delta) = \emptyset$,

(C2) $\forall x \in \operatorname{supp}(M) : \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_\varphi\big((x,y)\big) = \operatorname{mult}_M(x)$, *and*

(C3) $\forall y \in \operatorname{supp}(N) : \sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_\varphi\big((x,y)\big) = \operatorname{mult}_N(y)$.

Intuitively, $\operatorname{mult}_\varphi\big((x,y)\big) = n$ means that x is matched to y n-times. Condition (C1) ensures that there are no matchings from Δ to Δ, while (C2) and (C3) ensure that x, y are used in a matching exactly as often as they are contained in M, N.

Definition 5 (Bottleneck, Wasserstein distance) *Let $(\mathbb{R}^2, \delta)$ be a metric space and let*

$$\Delta = \{(x,y) \in \mathbb{R}^2 : x = y\}$$

be the diagonal of the real plane. For two barcodes $\mathcal{D}, \mathcal{E} \in \mathbb{D}$, the Bottleneck ($\mathrm{w}_\infty^\delta$) and Wasserstein ($\mathrm{w}_p^\delta$) distances (with respect to the metric δ) are defined by

$$\mathrm{w}_\infty^\delta(\mathcal{D}, \mathcal{E}) = \inf_\varphi \sup_{(x,y) \in \operatorname{supp}(\varphi)} \delta(x,y)$$

and

$$\mathrm{w}_p^\delta(\mathcal{D}, \mathcal{E}) = \inf_\varphi \left(\sum_{(x,y) \in \operatorname{supp}(\varphi)} \operatorname{mult}_\varphi\big((x,y)\big) \cdot \delta(x,y)^p \right)^{\frac{1}{p}} \text{ with } p \in [1, \infty) \ ,$$

where the infimum is taken over all Δ-relative bijective matchings φ between $\mathcal{D}$ and $\mathcal{E}$.

Remark 6 *In the case where δ is the metric induced by the q-norm, $\|\cdot\|_q$, we write w_∞ for $\mathrm{w}_\infty^{\|\cdot\|_\infty}$ and w_p^q for $\mathrm{w}_p^{\|\cdot\|_q}$.*

We also remark that introducing the Wasserstein distances as in Definition 5, using Δ-relative bijective matchings, deviates from the original formulation in Cohen-Steiner et al. (2007). However, this variant simplifies any of our subsequent analyses of stability/continuity properties involving those metrics. We refer the reader to (Cohen-Steiner et al., 2007, 2010; Chazal et al., 2009) for a selection of existing stability results in a broader context and further details.

Remark 7 *By setting $\mu_n^{i,\infty} = \beta_n^{i,m} - \beta_n^{i-1,m}$, we extend Eq. (1) to features which never disappear, also referred to as "essential". This change can be lifted to $\mathbb{D}$ by setting $\Omega = \{(b,d) \in \mathbb{R} \times (\mathbb{R} \cup \{\infty\}) : d > b\}$. However, if the filtration of K is defined by the sub level sets of a function f, a more pragmatic way of handling essential features is to map their death time to the maximum of the function f. In many cases, for example height filtrations, this has a more natural interpretation (see Section 6.3).*

3. Related work

In order to deal with the practical inconveniences associated to a (direct) handling of barcodes in machine learning problems, several strategies have been proposed over the last couple of years. On a high level, these approaches can be categorized into (1) *kernel-based* techniques and (2) approaches that aim for a *vectorization*. Next, we summarize these techniques, draw connections between them, and elaborate on how they relate to our work. We additionally discuss related research on learning with (multi) set data structures and highlight important differences to our approach.

Kernel-based techniques. In short, the idea of kernel-based learning techniques (Schölkopf and Smola, 2001) is to rely on a positive-definite *kernel* function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that realizes an inner product $\langle \phi(x), \phi(y) \rangle_\mathcal{G} = k(x,y)$, for $x, y \in \mathcal{X}$, in a Hilbert space $\mathcal{G}$ for some (possibly unknown) feature mapping $\phi : \mathcal{X} \to \mathcal{G}$. A kernel can either be constructed by (1) defining a function k that captures some notion of similarity between two input objects and showing its positive-definiteness, or (2) by explicitly constructing ϕ and using the inner product in $\mathcal{G}$ as a kernel (that is positive-definite by construction). Both strategies have found application in the context of kernel-based learning with topological signatures.

As a representative of the latter strategy, Reininghaus et al. (2015) introduced the *persistence scale-space (PSS) kernel*. The construction is based on first representing a diagram as a sum of Dirac deltas and then using this representation as the initial condition of a heat-diffusion process with a Dirichlet boundary condition on the diagonal Δ. The solution of this partial differential equation (at time t) resides in $L^2(\Omega)$ and serves as an explicit feature mapping. The inner-product in $L^2(\Omega)$ is then used as a kernel function. A different, yet conceptually similar approach is taken by Bubenik (2015) to construct *persistence landscapes*, that is, functions of the form $\lambda : \mathbb{N} \times \mathbb{R} \to \mathbb{R} \cup \{+\infty, -\infty\}$. Upon the definition of a suitable norm $\|\cdot\|_p$ on $\mathbb{N} \times \mathbb{R}$, landscapes map barcodes into a (separable) Banach space $L^p(\mathbb{N} \times \mathbb{R})$. Notably, for $p = 2$, we get the Hilbert space $L^2(\mathbb{N} \times \mathbb{R})$ which then facilitates to use the inner-product again to define a kernel. In Kwitt et al. (2015), it is further shown that under some mild restrictions on the barcodes (that is, birth-death boundedness and an upper-bound on the multiplicities of points), an exponentiated version of the persistence scale-space kernel is universal[2] (Steinwart and Christmann, 2008, Def. 4.52). A similar argumentation would give a universal kernel constructed from persistence landscapes, for $p = 2$. When working in this setting, statistical computations (see Gretton et al., 2012) become feasible, although, for landscapes many theoretical properties for statistics have already been developed (Chazal et al., 2013a, 2014; Fasy et al., 2014).

2. that is, the RKHS associated to the kernel is dense in the space of continuous functions $\mathbb{D} \to \mathbb{R}$.

A different kernel-based technique is introduced by Kusano et al. (2016), where the authors leverage the theory of reproducing kernel Hilbert space (RKHS) embeddings of probability measures (see Berlinet and Thomas-Agnan, 2004). Different to Reininghaus et al. (2015), a diagram is represented as a weighted measure, that is, a weighted sum of Dirac deltas, centered at each point of the diagram. The weighting function accounts for the different persistence of each point. Notably, the authors show stability of the kernel-induced distance between barcodes with respect to the Hausdorff distance (for two compact metric spaces embedded in the same metric space). Essentially, this is achieved via the aforementioned weighting function. Contrary to that, Reininghaus et al. (2015) achieve stability by enforcing the Dirichlet boundary condition on the diagonal. In the latter case, this only leads to stability of the kernel-induced distance with respect to w_1^q and it is shown that no kernel for which $k(\mathcal{F} \uplus \mathcal{G}, \mathcal{D}) = k(\mathcal{F}, \mathcal{D}) + k(\mathcal{G}, \mathcal{D})$ holds (with $\mathcal{F}, \mathcal{G}, \mathcal{D} \in \mathbb{D}$), is stable with respect to w_p^q for $p > 1$.

The recently proposed kernel approach of Carriére et al. (2017) follows the strategy of directly defining k, instead of explicitly constructing a feature map ϕ. While there is strong indication (via counterexamples) that the negative of the w_p^q distance cannot be used in a construction of the form $k(\mathcal{F}, \mathcal{G}) = \exp(-\mathrm{w}_p^q(\mathcal{F}, \mathcal{G}))$, as w_p^q is not negative semi-definite, the authors circumvent this problem by introducing the *sliced Wasserstein distance* which can be shown to be negative semi-definite. The distance induced by the resulting kernel, termed the *sliced Wasserstein kernel*, is strongly equivalent to w_1^∞.

Vectorization techniques. A conceptually different line of developments is to "coordinatize" the barcode space which allows vectorization. In (Adcock et al., 2016), for example, the authors define a ring of algebraic functions on the space of barcodes. A similar approach with appealing properties is based on tropical algebraic geometry (Kališnik Verovšek, 2018). In particular, the coordinatizations in the latter work are stable with respect to w_p^q and w_∞ (which is not the case for the approach by Adcock et al. (2016)) and facilitates the development of sufficient statistics for barcodes (Monod et al., 2017). While this presents a promising approach, the challenge is to construct suitable polynomials in a computationally efficient manner.

Another instance of a *vectorization* technique is presented in Adams et al. (2017), where barcodes are mapped to so-called *persistence surfaces*. This is done by computing a weighted sum of normalized (isotropic) Gaussians, evaluated at each point in the diagram. Upon discretization of this *persistence surface*, one obtains the *persistence image* that can then be vectorized and fed to, for example, a linear SVM. Notably, a similar approach, motivated from a statistical point of view, was introduced by Chen et al. (2015) based on earlier ideas presented in Edelsbrunner et al. (2012). Similar to Kusano et al. (2016), stability of persistence images (with respect to w_1^∞) is achieved via a continuous piecewise-differentiable weighting function that evaluates to zero on the diagonal Δ. By taking a measure-theoretic point of view, Chazal and Divol (2018) have recently shown that, in a wide range of situations, the persistence surface is a kernel density estimator for the expected barcode.

A different vectorization scheme is presented by Bendich et al. (2016), where the authors obtain a fixed-dimensional representation of barcodes by extracting the N longest bars. Based on this representation, it is then possible to apply standard statistical techniques, for

example, PCA, or a discriminant classifier. However, it is unclear how the hyper-parameter N should be chosen appropriately, a choice that will most likely depend on the application.

In summary, most vectorization and kernel-based techniques retain certain stability properties of persistence diagrams / barcodes with respect to the common metrics in the field of persistent homology. Yet, they also share one common *drawback*: the mapping of the topological signatures to a representation that is compatible with existing learning techniques is *pre-defined*. While this is, a-priori, not a disadvantage for statistical computations, it can be undesirable for learning, as the representation is *agnostic* to the learning task. In fact, the success of deep neural networks (see Krizhevsky et al., 2012; He et al., 2016) has shown that *learning* representations is a preferable approach. It is worth pointing out that algebraic constructions (Adcock et al., 2016; Kališnik Verovšek, 2018; Monod et al., 2017) might be amenable to learning task-specific representations, but it is unclear if this is computationally feasible. Furthermore, techniques based on kernels, such as (Reininghaus et al., 2015; Kwitt et al., 2015; Kusano et al., 2016; Carriére et al., 2017), additionally suffer scalability issues as training kernel SVMs scales poorly with the number of samples (Chapelle, 2007) and even evaluating the kernel function itself can be computationally challenging. In the spirit of end-to-end training, we therefore aim for a computationally efficient approach that allows to learn a *task-specific* representation.

Learning with sets. As mentioned earlier, learning with barcodes requires to appropriately handle *multi sets* and to respect the topology induced by the metrics. In the context of deep learning, using *multi sets* as input has gained limited attention so far, except for approaches where the multi set is the prediction target (Welleck et al., 2017). On the other hand, learning with *sets* as input to neural networks has spawned considerable research interest recently. In Qi et al. (2017a,b), for instance, the authors primarily focus on handling point clouds in a way that is permutation invariant with respect to the points. Zaheer et al. (2017) present a characterization theorem for valid set functions that allow the design of appropriate neural networks layers. In particular, it is shown that if set elements are from a *countable* domain, a necessary condition for a function to be a valid set function is to have a decomposition of the form $\rho(\sum_{x \in S} \phi(x))$, for a suitable choice of ρ and ϕ. In the *uncountable* case, this was only shown to hold for sets of fixed size, but it is conjectured that the previous characterization holds even for sets of arbitrary size. In this work, we are exactly in the latter setting, as we consider multi sets over an uncountable subset of $\mathbb{R}^2$ as input. Moreover, the fact that we are dealing with multi sets *and* a topology induced by the metrics in Definition 5 is crucial. It is, for example, straightforward to show (see Section 4.2) that blind application of the approach in Zaheer et al. (2017) does neither lead to a stable representation, nor is it clear how to appropriately handle points with higher multiplicities. Our construction scheme is specifically designed to avoid these problems.

4. Learning vectorizations of barcodes

This section is divided into two major parts. In the first part, we develop the intuition of the proposed (learnable) vectorization technique. Our intention is to give the reader a deeper

insight into the core ideas that motivate the construction scheme which we then introduce and analyze in the second part.

Motivation. As mentioned in Section 2, *barcodes* are multi sets residing over the upper diagonal part of the real plane, Ω. From both a computational and an implementation perspective, the handling of (multi) sets poses considerable challenges, simply because there is no notion of order.

While many previous works have resorted to a mapping of barcodes to function spaces (with linear structure) in order to facilitate learning techniques, our strategy is to construct a learnable vectorization $\mathcal{V} : \mathbb{D} \to \mathbb{R}^N$ which operates on (*finite*) barcodes and respects the topology induced by the metrics in Definition 5.

4.1. Vectorization of multi sets

We start with a rather basic strategy to handle multi sets and then refine this idea through an analysis of the desired properties in the context of barcodes. First, consider a multi set M over some fixed domain D. Then, a natural finite vectorization is to fix a finite subset of D, say $\{\mu_1, \dots, \mu_N\}$, and set

$$\mathcal{V}(M) = \big(\mathrm{mult}_M(\mu_1), \dots, \mathrm{mult}_M(\mu_N)\big) \ .$$

While possible, such a strategy is undesirable for multiple reasons. *First*, it is a rather strict representation, as for a multi set M where $M \cap \{\mu_1, \dots, \mu_N\} = \emptyset$, we would obviously get $\mathcal{V}(M) = (0, \dots, 0)$. *Second*, and possibly more fundamental, such a mapping would be inherently discontinuous, as it can not capture continuous changes of the points in M (if D is equipped with a topology). This is easy to see by considering the example of $D = \mathbb{R}$. In that case, we have

$$\mathcal{V}(\{\mu_1, \mu_1 + \varepsilon\}) = (1, 0, \dots, 0) \text{ for } \varepsilon > 0$$

if $(\mu_1 + \varepsilon) \notin \{\mu_1, \dots, \mu_N\}$, however,

$$\mathcal{V}(\{\mu_1, \mu_1\}) = (2, 0, \dots, 0) \ .$$

At first sight, given the obvious drawbacks of $\mathcal{V}$, continuing along this direction does not seem promising. However, the basic problem essentially resides in the (rigorously local) way of how elements of the multi set are represented by the multiplicity function. It is therefore interesting, to study a relaxed (that is, less local) version of this idea. In particular, consider a functional such that, for some metric δ residing on D,

$$s_{\mu_i} : D \to [0, 1] \ , \ s_{\mu_i}(\mu_i) = 1 \text{ and } \lim_{\delta(\mu_i, x) \to \infty} s_{\mu_i}(x) = 0 \tag{2}$$

for $1 \le i \le N$. Under this functional, we get $\mathrm{mult}_M(\mu_i) = \sum_{x \in M} s_{\mu_i}(x)$ as a boundary case. By further requiring that s_{μ_i} is continuous with respect to δ and by controlling the convergence speed of

$$s_{\mu_i}(x) \to 0 \quad \text{as} \quad \delta(\mu_i, x) \to \infty \tag{3}$$

we can reformulate a *relaxed*, but *continuous*[3] version of $\mathcal{V}$ as

$$\mathcal{V}(M) = \left(\sum_{x \in M} s_{\mu_1}(x), \dots, \sum_{x \in M} s_{\mu_N}(x) \right) \ . \tag{4}$$

Revisiting our initial example, we now obtain a more reasonable mapping of $\{\mu_1, \mu_1\}$ and $\{\mu_1, \mu_1 + \varepsilon\}$ as

$$\mathcal{V}(\{\mu_1, \mu_1\}) = (2, \varepsilon_2, \dots, \varepsilon_N) \text{ for } \varepsilon_i > 0$$

and

$$\mathcal{V}(\{\mu_1, \mu_1 + \varepsilon\}) = (2 - \varepsilon'_1, \varepsilon'_2, \dots, \varepsilon'_N) \text{ for } \varepsilon'_i > 0 \ ,$$

respectively[4]. In other words, the mapping changes continuously with respect to the points in M. However, this comes at the cost of precision with respect to the multiplicity function.

4.2. From multi sets to barcodes

So far, we have introduced a first strategy for vectorizing multi sets which could already be applied to barcodes. However, $\mathbb{D}$ is not simply a collection of multi sets, but a *metric space* (with respect to, for example, w^δ_∞ or w^δ_p) and hence has a topological structure. By keeping in mind that typical stability results in persistent homology are formulated with respect to those metrics, it seems imperative that the mapping $\mathcal{V}$ should be stable to at least a selection of those metrics. To highlight this, consider the following sequence of barcodes

$$\mathcal{D}_j = \{x_j\} \text{ with } x_j \in \Omega \text{ and } \lim_{j \to \infty} x_j \in \Delta \ .$$

For every choice of the proposed barcode distances, the empty barcode, $\emptyset$, is the limit of this sequence. A vectorization, $\mathcal{V}$, of the proposed form and continuous with respect to the metrical structure of the barcodes, $\mathbb{D}$, should yield

$$\mathcal{V}(\mathcal{D}_j) \to (0, \dots, 0) = \mathcal{V}(\emptyset) \text{ for } j \to \infty \ .$$

Hence, in addition to *continuity* of s on $\Omega \cup \Delta$, we should demand that

$$s(x) = 0 \text{ for } x \in \Delta \ .$$

Remark 8 *As mentioned in Section 3, a similar (multi) set vectorization method is introduced in the "deep sets" approach of Zaheer et al. (2017) by setting*[5]

$$\mathcal{V}(M) = \sum_{x \in M} \phi(x) \text{ for } M \subset \mathbb{R}^m \ ,$$

where $\phi : \mathbb{R}^m \to \mathbb{R}^n$ is a mapping implemented by a neural network (possibly through multiple layers). We highlight that this approach does not intrinsically respect the topological structure of $\mathbb{D}$. The reason is that ϕ is not constrained (or constructed) to vanish on the diagonal Δ. Hence, we have no guarantee that the vectorization is continuous with respect to the introduced metrics.

3. Continuity, at this point, is meant as continuity in the points of M.
4. For simplicity, we assume that the limit process in Eq. (3) is monotone.
5. In detail, we would have $\rho(\sum_{x \in M} \phi(x))$, but ρ does not affect our argument.

4.3. Learnable vectorizations

Until now, we have developed a first idea how a *fixed* vectorization $\mathcal{V}$ could be constructed. However, most of the progress in supervised learning over the last years can be attributed to the fact that state-of-the-art approaches (such as deep neural networks) do not rely on fixed representations. Instead, they operate on a *family of representations* and aim to find a task-specific one for the problem at hand. This is achieved by back-propagating the error under a suitable loss function and adjusting the representation parameters to minimize the loss. The sought-for vectorization $\mathcal{V}$ (as introduced in the previous section) can be interpreted as such a representation. Hence, it seems beneficial to define a family of mappings, $\mathcal{V}_\theta$, and let the learner determine a suitable parametrization θ. When learning via gradient descent, this demands (sub-)differentiability in the parameters θ. If $\theta = (\theta_i)$ for a defined set of (real) parameters, we have to put an additional constraint on a practically useful vectorization, that is, the existence of the partial (sub-)derivative with respect to θ_i,

$$\frac{\partial}{\partial \theta_i}\big(s_\theta(\mathcal{D})\big) \text{ for } \mathcal{D} \in \mathbb{D} \ .$$

4.4. Construction & Theoretical analysis

We propose a construction, based on parametrized functionals on $\Omega \cup \Delta$, for a vectorization that possesses the properties outlined in Sections 4.1 to 4.3. Our two main results, in Theorems 12 and 13, establish sufficient conditions such that the induced vectorization is (1) Lipschitz continuous with respect to w_1^δ (that is, *stable* in the sense of Cohen-Steiner et al. (2010)) and (2) *continuous* with respect to w_p^δ. Further, we show that stability with respect to w_p^δ for $p > 1$ is not possible. We begin by defining the notion of a *structure element*.

Definition 9 (Structure element) *Let s be a family of continuous functionals, parametrized over some parameter space $\Theta \subset \mathbb{R}^d$, that is,*

$$\Theta \ni \theta \overset{s}{\mapsto} s_\theta$$

such that

$$s_\theta : \Omega \cup \Delta \to \mathbb{R} \text{ with } s_\theta(x) = 0 \text{ for } x \in \Delta \ .$$

We call s_θ a structure element over the parameter space Θ.

Now, we introduce a mapping of barcodes with respect to s_θ.

Definition 10 (Induced mapping of barcodes) *Let s be a family of functionals as in Definition 9. Then, for each s_θ, we define by*

$$s_\theta : \mathbb{D} \to \mathbb{R} \qquad \mathcal{D} \mapsto \sum_{x \in \mathcal{D}} s_\theta(x)$$

the s_θ-induced mapping of barcodes.

The question is, if the induced mapping from Definition 10 allows to guarantee stability with respect to the Wasserstein distances, or a subset of them. To answer this, we provide a technical result (in Lemma 11) that allows us to shorten the proofs of Theorems 12 and 13.

Lemma 11 *Let $\mathcal{D}, \mathcal{E} \in \mathbb{D}$ and let φ be a Δ-relative bijective matching between $\mathcal{D}$ and $\mathcal{E}$. Further, let $f : \Omega \cup \Delta \to \mathbb{R}$ with $f(x) = 0$ for $x \in \Delta$. Then,*

$$\sum_{x \in \mathcal{D}} f(x) - \sum_{y \in \mathcal{E}} f(y) = \sum_{(x,y) \in \mathrm{supp}(\varphi)} \mathrm{mult}_\varphi\big((x,y)\big)\big(f(x) - f(y)\big) \ .$$

Proof see Appendix B.1

Theorem 12 (w_1^δ stability of the induced mapping) *Let s be a family of functionals as in Definition 9 and s_θ such that*

$$s_\theta \text{ is Lipschitz continuous w.r.t. } \delta \text{ and constant } K_{s_\theta} \ .$$

Then, for two barcodes $\mathcal{D}, \mathcal{E} \in \mathbb{D}$, it holds that

$$|s_\theta(\mathcal{D}) - s_\theta(\mathcal{E})| \leq K_{s_\theta} \cdot \mathrm{w}_1^\delta(\mathcal{D}, \mathcal{E}) \ . \tag{5}$$

Proof Let φ be a Δ-relative bijective matching between $\mathcal{D}$ and $\mathcal{E}$, realizing $\mathrm{w}_1^\delta(\mathcal{D}, \mathcal{E})$. As $\mathcal{D}, \mathcal{E}$ are assumed to be of finite cardinality it holds that $|\mathrm{supp}(\varphi)| < \infty$. Thus,

$$\begin{aligned}
|s_\theta(\mathcal{D}) - s_\theta(\mathcal{E})| &= \left| \sum_{x \in \mathcal{D}} s_\theta(x) - \sum_{y \in \mathcal{E}} s_\theta(y) \right| \\
&= \left| \sum_{(x,y) \in \mathrm{supp}(\varphi)} \mathrm{mult}_\varphi\big((x,y)\big) \cdot \big(s_\theta(x) - s_\theta(y)\big) \right| && \text{(by Lemma 11)} \\
&\leq \sum_{(x,y) \in \mathrm{supp}(\varphi)} \mathrm{mult}_\varphi\big((x,y)\big) \cdot \big|s_\theta(x) - s_\theta(y)\big| && \text{(by the triangle inequality)} \\
&\leq K_{s_\theta} \cdot \sum_{(x,y) \in \mathrm{supp}(\varphi)} \mathrm{mult}_\varphi\big((x,y)\big) \cdot \delta(x,y) && \text{(by Lipschitz continuity of } s_\theta) \\
&= K_{s_\theta} \cdot \mathrm{w}_1^\delta(\mathcal{D}, \mathcal{E}) \ .
\end{aligned}$$

■

Next, we show that the mapping induced by a structure element s_θ can not achieve stability for w_p^δ with $p > 1$ (including $p = \infty$, that is, Bottleneck stability). Our argumentation is similar to Reininghaus et al. (2015, Theorem 3), where the authors characterize kernels on barcodes that cannot be stable with respect to w_p^q for $p > 1$. To this end, let s_θ be non-trivial, that is, there is some $x \in \Omega$ such that $s_\theta(x) > 0$ and define a sequence of barcodes based on $\mathcal{D} = \{x\}$ by

$$\mathcal{D}_j = \biguplus_{i=1}^{j} \mathcal{D} \text{ with } j \geq 1 \ .$$

Now consider

$$|s_\theta(\mathcal{D}_j) - s_\theta(\emptyset)| = \left| \sum_{x \in \mathcal{D}_j} s_\theta(x) - 0 \right| = j \cdot \left| \sum_{x \in \mathcal{D}} s_\theta(x) \right| = j \cdot |s_\theta(x)| \tag{6}$$

and observe that

$$\mathrm{w}_p^\delta(\mathcal{D}_j, \emptyset) = \begin{cases} \sqrt[p]{j} \cdot \mathrm{w}_p^\delta(\mathcal{D}_1, \emptyset), & p < \infty, \\ 1 \cdot \mathrm{w}_\infty(\mathcal{D}_1, \emptyset), & p = \infty \ . \end{cases} \tag{7}$$

As we can see from Eq. (6), the order of growth is linear. For Eq. (7), this only holds in case $p = 1$. Hence, for $p > 1$, we cannot find a constant such that Eq. (6) is bounded from above by Eq. (7). In particular,

$$|s_\theta(\mathcal{D}_j) - s_\theta(\emptyset)| > K \cdot \mathrm{w}_p^\delta(\mathcal{D}_j, \emptyset)$$

for $K \in \mathbb{R}$ and j sufficiently large. While this is obviously a *negative* result, it does not necessarily mean that this adversely effects us in a learning setting . Nevertheless, although we cannot guarantee stability for $p > 1$, we can find sufficient conditions for a weaker property, namely *continuity* with respect to w_p^δ.

Theorem 13 (w_p^δ continuity of the induced barcode mapping) *Let s be a family of functionals as in Definition 9 such that s_θ satisfies the growth condition*

$$|s_\theta(x)| \leq \kappa_{s_\theta} \cdot \delta(x, y)^p \text{ for } x \in \Omega, y \in \Delta \text{ and } \kappa_{s_\theta} > 0 \ . \tag{8}$$

Then, the induced barcode mapping is continuous with respect to w_p^δ.

Proof Let $\mathcal{D} \in \mathbb{D}$ be arbitrary but fixed. Consider a sequence of barcodes, $(\mathcal{D}_j) \in \mathbb{D}$, converging to $\mathcal{D}$, that is,

$$\lim_{j \to \infty} \mathrm{w}_p^\delta(\mathcal{D}, \mathcal{D}_j) = 0 \ .$$

We have to show that $s_\theta(\mathcal{D}_j) \to s_\theta(\mathcal{D})$ for $j \to \infty$. Let φ_j be a Δ-relative bijective matching that realizes $\mathrm{w}_p^\delta(\mathcal{D}, \mathcal{D}_j)$. We get

$$\lim_{j \to \infty} \mathrm{w}_p^\delta(\mathcal{D}, \mathcal{D}_j)^p = \lim_{j \to \infty} \sum_{(x_j, y_j) \in \mathrm{supp}(\varphi_j)} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \delta(x_j, y_j)^p = 0 \ . \tag{9}$$

Thus, for $(x_j, y_j) \in \mathrm{supp}(\varphi_j)$, $\delta(x_j, y_j)$ converges to 0 for $j \to \infty$. In the following, we show that for $\varepsilon > 0$ it holds that $|s_\theta(\mathcal{D}) - s_\theta(\mathcal{D}_j)| \leq \varepsilon$. In particular, consider

$$\begin{aligned} |s_\theta(\mathcal{D}) - s_\theta(\mathcal{D}_j)| &= \left| \sum_{x \in \mathcal{D}} s_\theta(x) - \sum_{y \in \mathcal{D}_j} s_\theta(y) \right| \\ &= \left| \sum_{(x_j, y_j) \in \mathrm{supp}(\varphi_j)} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \big(s_\theta(x_j) - s_\theta(y_j)\big) \right| && \text{(by Lemma 11)} \\ &\leq \sum_{(x_j, y_j) \in \mathrm{supp}(\varphi_j)} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \big|s_\theta(x_j) - s_\theta(y_j)\big| && \text{(triangle inequality)} \\ &= C_j \ . \end{aligned}$$

To proceed, we split the matching φ_j into two disjoint parts, that is,

$$A_j = \mathrm{supp}(\varphi_j) \cap \big(\Omega \times (\Omega \cup \Delta)\big) \text{ and } B_j = \mathrm{supp}(\varphi_j) \cap \big(\Delta \times (\Omega \cup \Delta)\big) \ .$$

A_j contains the matchings from $\mathcal{D}$ to either points in $\mathcal{D}_j$ or Δ, while B_j contains the remaining matchings from Δ to $\mathcal{D}_j$. This allows us to represent C_j by A_j and B_j, that is,

$$C_j = \underbrace{\sum_{(x_j, y_j) \in A_j} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \big|s_\theta(x_j) - s_\theta(y_j)\big|}_{D_j} + \underbrace{\sum_{(x_j, y_j) \in B_j} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \big|s_\theta(x_j) - s_\theta(y_j)\big|}_{E_j} \ .$$

First, we see that

$$\begin{aligned} D_j &\leq \sum_{(x_j, y_j) \in A_j} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \max_{(x_j, y_j) \in A_j} \big|s_\theta(x_j) - s_\theta(y_j)\big| \\ &= |\mathcal{D}| \cdot \max_{(x_j, y_j) \in A_j} \big|s_\theta(x_j) - s_\theta(y_j)\big| && \text{(by (C2) of Definition 4)} \\ &\leq \varepsilon/2 \end{aligned}$$

for j sufficiently large, as $\delta(x_j, y_j) \to 0$ and s_θ is continuous on Ω (Definition 9). Second, to bound E_j, we observe the following: it holds that

$$\begin{aligned} E_j &= \sum_{(x_j, y_j) \in B_j} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \big|0 - s_\theta(y_j)\big| && \text{(as } x_j \in \Delta) \\ &\leq \sum_{(x_j, y_j) \in B_j} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \kappa_{s_\theta} \cdot \delta(x_j, y_j)^p && \text{(by Eq. (8))} \\ &\leq \kappa_{s_\theta} \cdot \sum_{(x_j, y_j) \in \mathrm{supp}(\varphi_j)} \mathrm{mult}_{\varphi_j}\big((x_j, y_j)\big) \cdot \delta(x_j, y_j)^p && \text{(as } B_j \subset \mathrm{supp}(\varphi_j)) \\ &= \kappa_{s_\theta} \cdot \mathrm{w}_p^\delta(\mathcal{D}, \mathcal{D}_j)^p && \text{(by Definition 5)} \\ &\leq \varepsilon/2 && \text{(by Eq. (9))} \end{aligned}$$

for j sufficiently large. Overall, we get

$$|s_\theta(\mathcal{D}) - s_\theta(\mathcal{D}_j)| \leq D_j + E_j \leq \varepsilon$$

for j sufficiently large, which concludes the proof.

■

Overall, our theoretical analysis of the proposed construction scheme shows that the induced mapping of barcodes has favorable properties, as long as the discussed conditions on the structure element are satisfied. Based on these developments, it is now straightforward to introduce a neural network module that can handle barcodes as input and extend this scheme to multiple families of functionals.

Definition 14 (Neural network input layer) *Let s be a family of functionals over the parameter space Θ and $\theta = (\theta_1, \dots, \theta_N)$ a vector of parameter realizations in Θ^N. Then,*

$$\begin{aligned} \mathcal{V}_s : \Theta^N &\to \{f : \mathbb{D} \to \mathbb{R}^N\} \\ \theta &\mapsto \mathcal{V}_{s,\theta} \end{aligned}$$

with

$$\begin{aligned} \mathcal{V}_{s,\theta} : \mathbb{D} &\to \mathbb{R}^N \\ \mathcal{D} &\mapsto \big(s_{\theta_1}(\mathcal{D}), \dots, s_{\theta_N}(\mathcal{D})\big) \end{aligned}$$

gives a mapping that is implementable as a neural network (input) layer.

Notably, Lipschitz continuity of the input layer $\mathcal{V}_{s,\theta}$ with respect to w_1^δ is a simple consequence of Theorem 12.

In the following part of the paper, we take a closer look at the practical aspects of the proposed construction scheme.

5. From theory to practice

In addition to the conditions on the structure elements, we now additionally require that the elements are (sub-)differentiable in their parameter space to facilitate numerical optimization in a gradient descent manner (see Section 4.3). In the context of neural networks, this allows to learn a task-specific representation of barcodes via error backpropagation. Taking this into consideration and setting $\delta = \|\cdot\|_q$, we arrive at the following conditions for a *practical* structure element s_θ:

(P1) $s_\theta(x) = 0$ for $x \in \Delta$

(P2) s_θ is Lipschitz continuous with respect to $\|\cdot\|_q$

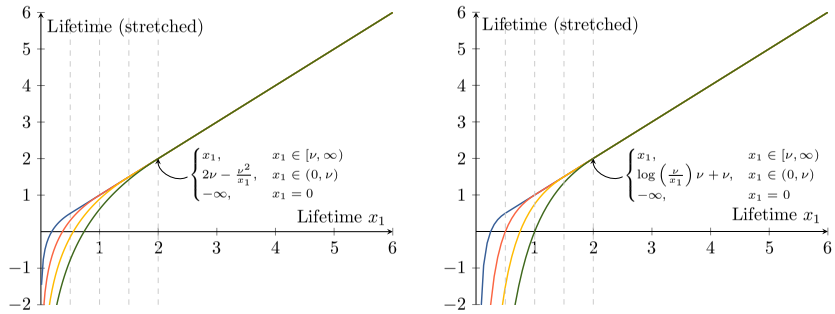
(P3) For $\mathcal{D} \in \mathbb{D}$, we require $s_\theta(\mathcal{D})$, $\theta \in \Theta$, to be differentiable on the parameter space Θ

As shown in the previous section, conditions (P1) and (P2) suffice to guarantee that mapping $\mathcal{V}_{s,\theta}$, stated in Definition 14, is stable[6] with respect to w_1^q. To develop structure elements and to verify the conditions, it will be helpful to rearrange and transform the coordinate axes of the barcode. We introduce this next.

5.1. Birth-lifetime coordinates

The intention of rearranging the coordinate axes to a *birth-lifetime* coordinate frame is to facilitate a straightforward analytic construction of structure elements that fulfill requirements (P1) to (P3). First, note that in the traditional notation we have $(x_0, x_1) \in \Delta \Leftrightarrow x_0 = x_1$. Hence, the predicate for (x_0, x_1) to be contained in Δ depends on both x_0 and x_1. Shifting the coordinate system such that (x_0', x_1') lies in the shifted version of $\Delta \Leftrightarrow x_1' = x_1 - x_0 = 0$

6. For $p > 1$, to guarantee continuity with respect to w_p^q, the growth condition from Eq. (8) needs to hold.



(a) (b)

Figure 2: Stretching the *lifetime* axis of a birth-lifetime transformed (cf. Def. 15) barcode via (a) the transform proposed in Eq. (10) and (b) the log-stretching previously proposed by Hofer et al. (2017a). Different values of ν are marked by dashed lines. While, the differences appear subtle, the stretching as in (a) allows to guarantee Lipschitz continuity of *all* proposed structure elements (Section 5.4), whereas (b) only works for exponential structure elements (Section 5.4.1).

results in a notable simplification. We summarize this step in the following definition of a *birth-lifetime* coordinate transform, similar to (Adams et al., 2017).

Definition 15 (Birth-lifetime coordinate transform) *We define*

$$\rho : \Omega \cup \Delta \to \mathbb{R} \times [0, \infty) \qquad (b, d) \mapsto (b, d - b) \ .$$

as the birth-lifetime coordinate transform of a barcode.

The transform ρ enables representing barcodes as multi sets over the upper half-plane of $\mathbb{R}^2$.

5.2. Rationally stretched birth-lifetime coordinates

Second, we introduce a transform, τ, on top of ρ, such that we can represent barcodes as multi sets over $\mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$. We use the convention that for some function $f : \mathbb{R} \to \mathbb{R}$, $f(-\infty) = \lim_{x \to -\infty} f(x)$, if the limit exists and it is not defined otherwise.
The advantage of splitting the structure element into a composition of ρ, τ and t_θ is that is allows to use parametrized functionals t_θ defined on $\mathbb{R} \times \mathbb{R}$ as a starting point and then require that t_θ vanishes on $\mathbb{R} \times \{-\infty\} = \tau \circ \rho(\Delta)$. For example, most continuous probability density functions are subject to this constraint. As a consequence, we can use the rich pool of such functions to find possible candidates for t.

Definition 16 (Rationally stretched birth-lifetime transform) *For $0 < \nu$, we define the birth-lifetime transform*

$$\tau_\nu : \mathbb{R} \times [0, \infty) \to \mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$$

by

$$\tau_\nu((x_0,x_1)) = \begin{cases} (x_0,x_1), & x_1 \in [\nu,\infty), \\ (x_0, 2\nu - \frac{\nu^2}{x_1}), & x_1 \in (0,\nu), \\ (x_0,-\infty), & x_1 = 0 \ . \end{cases} \tag{10}$$

Note that τ_ν is continuous on $\mathbb{R}^2$ as

$$\lim_{x\to\nu} x = \nu = \lim_{x\to\nu} 2\nu - \frac{\nu^2}{x} \ .$$

As a consequence, we can define a parametrized functional, say

$$t : \Theta \to \{f : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}\} \qquad \theta \mapsto t_\theta \ ,$$

on $\mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$ such that

$$s = t \circ \tau_\nu \circ \rho : \Theta \to \{f : \Omega \cup \Delta \to \mathbb{R}\} \qquad \theta \mapsto s_\theta = t_\theta \circ \tau_\nu \circ \rho$$

is a structure element iff $t\big((x_0,-\infty)\big) = 0$. A visualization of the proposed birth-lifetime transform is shown in Figure 2b, in comparison to the original (but limited) variant introduced in Hofer et al. (2017a).

5.3. Lipschitz continuity

As we are especially interested in a stability preserving vectorization, we want to construct structure elements which are Lipschitz continuous, see Section 4. While the transform $\tau_\nu \circ \rho$ facilitates the selection of t, it also comes at a price. A closer examination reveals that τ_ν is not continuous on Δ and hence not Lipschitz continuous. As a consequence, Lipschitz continuity of some functional t is not enough to guarantee that $t \circ \tau_\nu \circ \rho$ is Lipschitz continuous. The core of the problem is that $\tau_\nu \circ \rho$ diverges for $(x_0,x_1) \to (x_0,0)$. In order to preserve Lipschitz continuity, a chosen functional t has to *converge rapidly enough* to dominate this divergence. In the following, we provide the essential argumentation to verify the Lipschitz condition for $t \circ \tau_\nu \circ \rho$. We start with two remarks to recall some properties of Lipschitz continuous functions.

Remark 17 *To show that a multivariate function $f : \mathbb{R}^N \to \mathbb{R}$ is Lipschitz continuous with respect to $\|\cdot\|_q$, $1 \le q \le \infty$, it is sufficient that Lipschitz continuity with respect to $\|\cdot\|_q$ holds in each coordinate. The reason is that coordinate-wise Lipschitz continuity with respect to $\|\cdot\|_q$ leads to Lipschitz continuity with respect to $\|\cdot\|_1$ by taking the maximum of the coordinate-wise Lipschitz constants. As all q-norms are equivalent, that is, $c\|\cdot\|_{q'} \le \|\cdot\|_q \le C\|\cdot\|_{q'}$ for $1 \le q \le q' \le \infty$ and $0 < c \le C$, this is sufficient. Additionally, concatenation preserves Lipschitz continuity, that is, if f, g are Lipschitz continuous with respect to to some metric, so is $f \circ g$.*

Our strategy to verify that a construction of the form $t \circ \rho_\nu \circ \tau$ is Lipschitz continuous will be as follows: *First*, let $\nu > 0$ and $t : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be Lipschitz continuous. Then, it suffices to verify Lipschitz continuity of $t \circ \tau_\nu$ on $\mathbb{R} \times [0,\infty)$, as ρ is trivially Lipschitz continuous. *Second*, for all following variants of t in Section 5.4, it will be clear that $\mathbb{R} \times [\varepsilon,\infty), \varepsilon > 0$ poses no problem and it remains to study the behavior of $t \circ \tau_\nu$ on $\mathbb{R} \times [0,\varepsilon)$. Now, as all variants of t will also be Lipschitz continuous in the first coordinate, x_0, only the second coordinate, x_1, needs to be considered (Remark 17) and our strategy boils down to show

$$\left| \frac{\partial}{\partial x_1} t \circ \tau_\nu \right| < C \in \mathbb{R} \text{ on } [0,\varepsilon) \ .$$

In all of the following cases, this will be equivalent to

$$\lim_{x\to 0} \left| \left(\frac{\partial}{\partial x_1} t \circ \tau_\nu \right)(x) \right| < C \in \mathbb{R} \ ,$$

as the partial derivative of $t \circ \tau_\nu$ will be continuous on $(0,\varepsilon)$.

5.4. Structure elements

Next, we use the previous ideas to introduce (radial) structure elements which result in w_1^q stable vectorizations. In particular, we start with a structure element based on the exponential function, analyze its properties, and then successively refine this element by switching to rational functions for better numerical stability and compatibility of the parameter gradients. Our main motivation is, to build structure elements that integrate well into a deep learning framework.

Remark 18 *This section iteratively refines the structure element originally proposed by Hofer et al. (2017a), eventually resulting in the rational hat structure element of Section 5, Definition 23. We choose this type of presentation, as it elucidates certain design choices. However, those (intermediate) steps are not necessary to understand the final form of the refined version and may be skipped.*

5.4.1. Exponential structure element(s)

Exponential functions have been previously used by Reininghaus et al. (2015), Kusano et al. (2016) and Adams et al. (2017) to weight points in barcodes. In (Reininghaus et al., 2015), the exponential function arises as the solution of a heat-diffusion problem, whereas in (Kusano et al., 2016) this is motivated by the idea of embedding measures into a reproducing kernel Hilbert space. The crucial difference to these approaches is that our structure elements *are not* at *fixed* locations (that is, one element per point of the particular barcode), but their locations and scales are adjustable.

Definition 19 (Exponential structure element) *We set $\Theta = \mathbb{R}^2 \times \mathbb{R}^2$ with $((\mu_0,\mu_1),(\sigma_0,\sigma_1)) = (\mu,\sigma) \in \Theta$ and define*

$$\begin{aligned} t^{exp} &: \Theta \to \{f : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}^+\} \\ &(\mu,\sigma) \mapsto t^{exp}_{(\mu,\sigma)} \end{aligned}$$

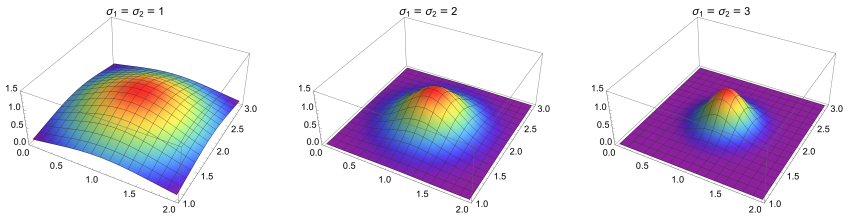


Figure 3: Illustration of an *exponential structure element*, centered at $\mu = (1, 2)$.

with

$$t^{exp}_{(\mu,\sigma)} : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}^+$$
$$(x_0, x_1) \mapsto e^{-(\sigma_0^2(\mu_0 - x_0)^2 + \sigma_1^2(\mu_1 - x_1)^2)} \quad .$$

We call $s^{exp}_{\mu,\sigma} = t^{exp}_{\mu,\sigma} \circ \tau_\nu \circ \rho$ *an exponential structure element.*

Note that $s^{\mathtt{exp}}_{\mu,\sigma}$ is well-defined, that is, it is a structure element in the sense of Definition 9. This is obvious, as

$$s^{\mathtt{exp}}_{\mu,\sigma}(\Delta) = t^{\mathtt{exp}}_{\mu,\sigma} \circ \tau_\nu \circ \rho(\Delta) = t^{\mathtt{exp}}_{\mu,\sigma} \circ \tau_\nu(\mathbb{R} \times \{0\}) = t^{\mathtt{exp}}_{\mu,\sigma}(\mathbb{R} \times \{-\infty\}) = \{0\}$$

holds by design. An illustration of an exponential structure element, centered at $\mu = (1, 2)$ is shown in Figure 3. The following lemma establishes Lipschitz continuity of this element.

Lemma 20 $s^{exp}_{(\mu,\sigma)}$ *is Lipschitz continuous with respect to* $\|\cdot\|_q$ *on* $\Omega \cup \Delta$.

Proof see Appendix B.2

With respect to the differentiability of $s^{\mathtt{exp}}_{(\mu,\sigma)}$, it is easy to see that $t^{\mathtt{exp}}_{(\mu,\sigma)}(x)$ is differentiable in Θ for $x \in \mathbb{R}^2$. Therefore, so is $s^{\mathtt{exp}}_{(\mu,\sigma)}(x)$ for $x \in \Omega \cup \Delta$. As the differential operator is additive, it follows that the induced barcode mapping, $\sum_{x \in \mathcal{D}} s^{\mathtt{exp}}_{(\mu,\sigma)}(x)$, is differentiable.

5.4.2. Rational structure element(s)

The second structure element is inspired by rational functions of the form $(1 + x)^{-n}$. Our main intention is to simplify $s^{\mathtt{exp}}$ such that we circumvent the exponential dependency of the diagram points. This is desirable in terms of numerical stability.

Definition 21 (Rational structure element) *We set* $\Theta = \mathbb{R}^2 \times \mathbb{R}^2 \times [1, \infty)$ *with* $((\mu_0, \mu_1), (\sigma_0, \sigma_1), \alpha) = (\mu, \sigma, \alpha) \in \Theta$ *and define*

$$t^{rat} : \Theta \to \{f : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}^+\}$$
$$(\mu, \sigma) \mapsto t^{rat}_{(\mu,\sigma,\alpha)}$$

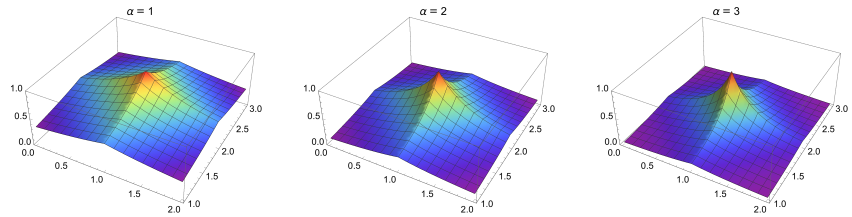


Figure 4: Illustration of a *rational structure element* for $\sigma_{0,1} = 1$ and various choices of α.

with

$$t^{rat}_{(\mu,\sigma,\alpha)} : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}$$
$$(x_0, x_1) \mapsto \frac{1}{\left(1 + |\sigma_0| \cdot |x_0 - \mu_0| + |\sigma_1| \cdot |x_1 - \mu_1|\right)^\alpha} \quad .$$

We call $s^{rat}_{(\mu,\sigma,\alpha)} = t^{rat}_{(\mu,\sigma,\alpha)} \circ \tau_\nu \circ \rho$ *a rational structure element.*

Lemma 22 $s^{rat}_{(\mu,\sigma,\alpha)}$ *is Lipschitz continuous with respect to* $\|\cdot\|_q$.

Proof see Appendix B.3

An illustration of a rational structure element, centered at location $\mu = (1, 2)$ is shown in Figure 4. Similar to the exponential structure element, we see that $s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}$ is differentiable in μ_i and σ_i for $i = 1, 2$. However, in case of α, we have to introduce a slight restriction as $s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}$ is only differentiable for $\alpha \in (1, \infty)$. Regarding the partial derivatives, it is worth pointing out that

$$\frac{\partial}{\partial \alpha} s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}(x) \quad \text{and} \quad \frac{\partial}{\partial \theta} s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}(x) \quad \text{for } \theta \in \{\sigma_0, \sigma_1, \mu_0, \mu_1\}$$

are scaled differently. More precisely, let $f(\mu, \sigma, x)$ denote the denominator, that is,

$$s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}(x) = \frac{1}{f(\mu, \sigma, x)^\alpha} \quad .$$

Then, we see that

$$\frac{\partial}{\partial \alpha} s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}(x) = \ln\left(\frac{1}{f(\mu, \sigma, x)^\alpha}\right) \cdot \frac{1}{f(\mu, \sigma, x)^\alpha}$$

differs from

$$\frac{\partial}{\partial \theta} s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}(x) = -\alpha \cdot \frac{1}{f(\mu, \sigma, x)^{\alpha+1}} \cdot \frac{\partial}{\partial \theta} f(\mu, \sigma, x) \quad \text{for } \theta \in \{\sigma_0, \sigma_1, \mu_0, \mu_1\}$$

with respect to the order of growth, depending on f. This yields the conjecture that it may be difficult to optimize both parameter groups $\{\alpha\}$ and $\{\sigma_0, \sigma_1, \mu_0, \mu_1\}$ simultaneously. A similar scaling of the parameter gradients is preferable, as it simplifies the optimization process.

5.4.3. Rational hat structure element(s)

A closer look at $s^{\mathtt{rat}}$ reveals that the purpose of the parameters σ and α is to control the slope of $s^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}$ around μ. To obtain a similar effect and to reduce the number of parameters, we gather this functionality into a single parameter r. We will see that this strategy yields a more balanced gradient scaling behavior. In detail, the idea is to use a uniformly scaled rational structure element and to subtract an antipode whose maximum is reached when the distance of a point to μ is exactly r. This leads to the following definition.

Definition 23 (Rational hat structure element) *We set $\Theta = \mathbb{R}^2 \times \mathbb{R} \times \mathbb{N}$ with $((\mu_0, \mu_1), r, q) = (\mu, r, q) \in \Theta$, $x = (x_0, x_1) \in \mathbb{R} \times (\mathbb{R} \cup \{-\infty\})$ and define*

$$\begin{aligned} t^{\mathtt{rat_hat}} : \Theta &\to \{f : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) \to \mathbb{R}\} \\ (\mu, r, q) &\mapsto t^{\mathtt{rat_hat}}_{(\mu,r,q)} \end{aligned}$$

with

$$\begin{aligned} t^{\mathtt{rat_hat}}_{(\mu,r,q)} : \mathbb{R} \times (\mathbb{R} \cup \{-\infty\}) &\to \mathbb{R} \\ x &\mapsto \frac{1}{1 + \|x-\mu\|_q} - \frac{1}{1 + |\,|r| - \|x-\mu\|_q|} \quad . \end{aligned}$$

We call $s^{\mathtt{rat_hat}}_{(\mu,\sigma,\alpha)} = t^{\mathtt{rat_hat}}_{(\mu,\sigma,\alpha)} \circ \tau_\nu \circ \rho$ rational hat structure element.

To see that this structure element has the desired gradient behavior, lets consider the partial derivatives with respect to r and μ_i, that is,

$$\frac{\partial}{\partial r}\left(t^{\mathtt{rat_hat}}_{(\mu,r,q)}(x)\right)(r) = \frac{\operatorname{sgn}(|r| - \|x-\mu\|_q|)}{(1 + |\,|r| - \|x-\mu\|_q|)^2} \cdot \operatorname{sgn}(r)$$

and

$$\frac{\partial}{\partial \mu_i}\left(t^{\mathtt{rat_hat}}_{(\mu,r,q)}(x)\right)(\mu_i) = (-1) \cdot \left(\frac{1}{(1+\|x-\mu\|_q)^2} - \frac{\operatorname{sgn}(|r|-\|x-\mu\|_q|)}{(1+|\,|r|-\|x-\mu\|_q|)^2}\right) \cdot \frac{\partial}{\partial \mu_i}(\|x-\mu\|_q)(\mu_i) \quad .$$

We observe that both terms have similar order of growth in r and μ_i, respectively. However, this comes at the price that we no longer have $s^{\mathtt{rat_hat}}_{(\mu,\sigma,\alpha)}(\mu) = 1$. While this could be repaired by a multiplicative scaling factor

$$c_r = \frac{1 + |r|}{|r|} \quad ,$$

a closer look at

$$\frac{\partial}{\partial r} c_r(r) = -\frac{\operatorname{sgn}(r)}{|r|^2}$$

reveals that this should be avoided, as it would lead to an undesirable quadratic dependency of the denominator with respect to r.

Lemma 24 *$t^{\mathtt{rat_hat}}_{(\mu,\sigma,\alpha)}$ is Lipschitz continuous with respect to $\|\cdot\|_q$.*

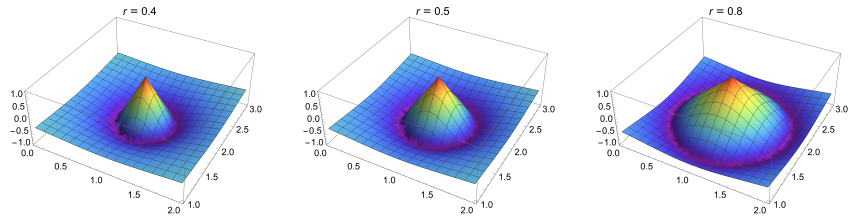


Figure 5: Illustration of a *rational hat structure element* for $\|\cdot\|_2$ and various choices of r.

We omit the proof for brevity, but note that it is similar to the proof of Lemma 22. Furthermore, $s^{\mathtt{rat_hat}}_{(\mu,\sigma,\alpha)}(\mathcal{D})$ is differentiable in its parameters. An illustration of this structure element, centered at $\mu = (1, 2)$, for different parameter settings is shown in Figure 5.

Overall, the advantages of $s^{\mathtt{rat_hat}}$ over $s^{\mathtt{exp}}$ and $s^{\mathtt{rat}}$ in terms of (1) the scaling of the derivatives and (2) its straightforward implementation, are particularly appealing for learning in the context of neural networks. Consequently, we use this element in all experiments that follow.

6. Experiments

We present a diverse set of experiments for supervised learning with different types of data objects. In particular, we show results for 2D/3D shape recognition and the classification of social network graphs, evaluated on standard benchmark data sets. Additionally, we present two *exploratory* experiments for the problems of predicting the eigenvalue distribution of normalized graph Laplacian matrices and activity recognition from EEG signals. These experiments demonstrate the versatility and predictive power of persistent homology. Our goal is not to necessarily outperform approaches that are specifically tailored to a problem, but to demonstrate that persistent homology combined with deep learning delivers competitive performance in a very generic setup.

All experiments were implemented in `PyTorch`[7], using `DIPHA`[8] (Bauer et al., 2014) and `Perseus`[9] (Mischaikow and Nanda, 2013) for persistent homology computations. Source code to fully reproduce the experiments is publicly-available[10].

6.1. Network architecture

We focus on a generic neural network architecture, shown in Figure 6, that only needs to be slightly adjusted per experiment.

7. https://github.com/pytorch/pytorch
8. https://bitbucket.org/dipha/dipha
9. http://people.maths.ox.ac.uk/nanda/perseus
10. https://github.com/c-hofer/jmlr_2019.git

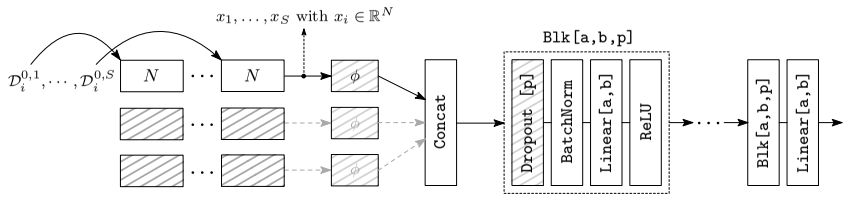


Figure 6: Illustration of our generic *network architecture*. Building blocks that are hatched are optional. One processing path (solid arrows) for handling S barcodes $\mathcal{D}_i^{0,1}, \ldots, \mathcal{D}_i^{0,S}$ from *one* input object o_i are shown. In particular, $\mathcal{D}_i^{0,j}$ denotes the S-th 0-dimensional barcode for o_i. Handling homology groups of higher dimension simply requires more input layers. The hyperparameter N denotes the number of structure elements per input layer.

For each filtration and homology dimension, we need one input layer. In Figure 6, an example with S 0-dimensional barcodes, $\mathcal{D}_i^{0,1}, \ldots, \mathcal{D}_i^{0,S}$, obtained from one input object o_i, is illustrated. Each input layer can have a varying number, N, of structure elements. We use the rational hat structure element (see Definition 23) for the following reasons: *First*, we only need to learn two parameters per structure element, that is, μ and r. *Second*, the gradients with respect to (μ, r) only differ by a linear term which we consider beneficial for learning. *Third*, commonly used building blocks of neural networks, for instance, batch normalization or rectified linear units (ReLU), expect both negative and positive inputs which is a property that only the rational hat element possesses. Overall, the output of one input layer is a vectorization in $\mathbb{R}^N$ that is then concatenated with the output of other input layers and fed into a multilayer perceptron (MLP). For optimization, we use stochastic gradient descent (SGD) with momentum and cross-entropy loss (if not mentioned otherwise). We additionally include a non-linearity $\phi : \mathbb{R} \to \mathbb{R}$ (in our case $\phi = \tanh$) after the input layer(s) to squash values in situations where points occur with high multiplicity; otherwise, $\phi = \mathrm{id}$. The parameter ν for the birth-lifetime transform of Definition 16 is fixed to 0.01 over all experiments[11]. We refer the reader to Table 6 for a full specification of the network configurations. Regarding initialization of the structure elements, we run k-means++ (Arthur and Vassilvitskii, 2007) clustering on all points from diagrams in the training portion of each dataset with $k = N$. The N cluster centers are then used to initialize the position of the structure elements. The parameter r is set to $1/4$ of the maximum lifetime of barcode elements in the training corpus. Different initializations of r (within a reasonable range) only had a negligible impact on our results.

6.2. Baselines & Competitors

In all experiments on *benchmark data* (Sections 6.3 to 6.5), we compare against two baselines. *First*, we include a straightforward vectorization approach, following Bendich et al. (2016).

11. In principle, ν could be learned as well, but we did not explore that direction in this work.

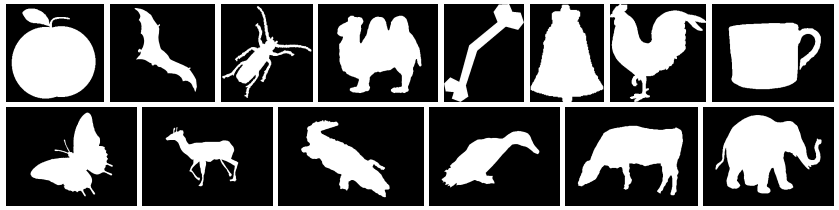


Figure 7: Some examples from the `MPEG-7` (*top*) and `Animal` (*bottom*) 2D shape datasets.

For each point (b, d) in a barcode $\mathcal{D}$, we calculate its *persistence* $d - b$, sort the calculated persistences by magnitude from high to low and take the first N values. Hence, for each barcode, a vector of dimension N (if $|\mathcal{D} \setminus \Delta| < N$, we pad with zero) is obtained. Then, a linear SVM (denoted as $\texttt{SVM}^{\texttt{lin}}$) is trained with N determined by cross-validation on the training portion of each dataset (we denote this as *Baseline* in all experiments). *Second*, we include a baseline-like approach where the parametrization of the structure elements in our approach is *only initialized* via k-means++ and then frozen during optimization. This allows to assess the impact of learning the parametrization of the structure elements.

We additionally compare against the *persistence images* approach from Adams et al. (2017) in two different variants. In its first incarnation, persistence images are computed on a 20×20 grid, vectorized (as originally suggested) and fed to a linear SVM. We use Gaussians centered at each point with the standard deviation determined via cross-validation (over $\sigma \in \{0.1, 0.5, 1.0\}$). In its second incarnation, we compute persistence images for the three choices of σ listed above and then use these images as a *multi-channel* input (that is, each σ yields an input channel) to a convolutional neural network (`NN`) with one convolution layer (using 3×3 kernels and a stride of 2), followed by two fully-connected layers (interleaved with ReLUs). This setup is similar to Cang and Wei (2017) where barcodes are vectorized and fed to a 1D convolutional neural network. We use Adam (Kingma and Ba, 2014) for optimization, train for 100 epochs with a batch size of 32, and use a learning rate of 0.01 (dropping to 0.001 after 50 epochs). This establishes a strong baseline and closely resembles our approach. For fair comparison, the number of (trainable) parameters in this network is comparable to the number of parameters in our architecture. We also found that including additional convolution layers did not further improve the results (in fact, unlike natural images, persistence images do not possess a compositional structure which would justify multiple convolution layers).

Remark 25 *It is interesting to note that the construction of persistence images, in particular the construction of the persistence surface, is conceptually similar to the feature map constructed in the PSS kernel of Reininghaus et al. (2015). The key difference is that the feature map in the PSS kernel evaluates to zero on the diagonal by mirroring the Gaussians (placed on top of each point in the barcode), whereas persistence images use a suitable weighting function to achieve this. From this perspective, feeding persistence images to a convolutional neural network is akin to an approach that would feed a discretized (on a*

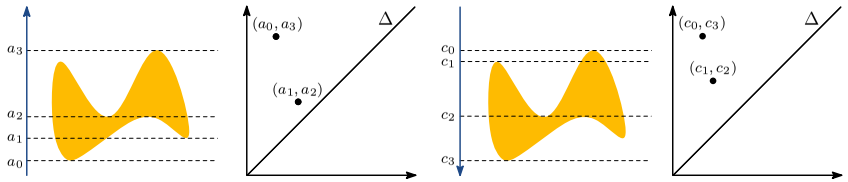


Figure 8: *Height function filtration* of a 2D shape from two directions, that is, $\xi = (0, 1)$ (*left*) and $\xi' = (0, -1)$ (*right*), with their corresponding 0-dimensional barcodes (visualized as persistence diagrams).

grid) version of the PSS kernel feature map (or multiple feature maps, computed with varying PSS kernel scale parameter).

6.3. Recognition of 2D shapes

We first test our approach on two different datasets of 2D object shape recognition: (1) the `Animal` dataset, introduced by Bai et al. (2009), which consists of 20 different animal classes, 100 samples each, and (2) the `MPEG-7` dataset which consists of 70 classes of different object/animal shapes, 20 samples each; see Latecki et al. (2000) for more details. For illustration, Figure 7 shows a selection of 2D object shapes from both datasets.

Filtration. The requirements to use persistent homology on 2D shapes are twofold: *First*, we need to assign a simplicial complex to each shape; *second*, we need to appropriately filter the complex. While, in principle, we could analyze contour features, such as curvature, and compute a sub level set filtration, such a strategy requires substantial preprocessing of the discrete data (for example, smoothing). Instead, we choose to work with the raw pixel data and leverage the *persistent homology transform*, introduced by Turner et al. (2014b). The filtration in that case is based on sub level sets of the *height function*, computed from multiple directions, illustrated in Figure 8. Practically, this means that we *directly construct a simplicial complex from the binary image*. We set K_0 as the set of all pixels which are contained in the object. A 1-simplex $[p_0, p_1]$ is in the 1-skeleton K_1 iff p_0 and p_1 are 4–neighbors on the pixel grid. To filter the complex, we denote by β the barycenter of the object and by r the radius of its bounding circle around β. For $[p] \in K_0$ and $\xi \in \mathbb{S}^1$, the filtration function is then given by

$$f([p]) = \frac{1}{r} \cdot \langle p - \beta, \xi \rangle \ ,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbb{R}^2$. The filtration is lifted to K_1 by taking the maximum over the two vertices of each edge. Letting ξ_i denote 16 equidistantly distributed directions in $\mathbb{S}^1$, starting from $(1, 0)$, we get a vector of barcodes $(\mathcal{D}^{0,j})_{j=1}^{16}$. Here, $\mathcal{D}^{0,j}$ is the barcode for 0-dimensional homology groups, obtained by filtration along ξ_j. As most objects do not differ in homology groups of *higher* dimensions (> 0), we did not use the corresponding barcodes. Furthermore, we only obtain *one* essential feature per direction, as

		`MPEG-7`	`Animal`
‡Skeleton paths (Bai et al., 2009)		86.7	67.9
‡Class segment set (Sun and Super, 2005)		90.9	69.7
†ICS (Bai et al., 2009)		96.6	78.4
†BCF (Wang et al., 2014)		**97.2**	**83.4**
Baseline (Bendich et al., 2016)	$\texttt{SVM}^{\texttt{lin}}$	82.3	50.0
PSS kernel (Reininghaus et al., 2015)	$\texttt{SVM}^{\texttt{kernel}}$	94.5	73.6
Persistence images (Adams et al., 2017)	$\texttt{SVM}^{\texttt{lin}}$	85.1	61.1
Persistence images (Adams et al., 2017)	`NN`	92.3	69.4
Ours (init. only, via k-means++)	`NN`	92.2	72.9
Ours (learned)	`NN`	92.7	74.6

Table 1: Average cross-validation accuracies (in %) on the `Animal` and `MPEG-7` benchmark datasets, compared to the two best (†) and two worst (‡) results reported in Wang et al. (2014), as well as different approaches that use persistent homology: our baselines from Section 6.2 and the PSS kernel from Reininghaus et al. (2015).

all objects only have one connected component. For practical reasons, the death time of those essential points is set to the maximum filtration value of the corresponding direction. This has a natural geometric interpretation, as the persistence of these essential points is the diameter of the object along a direction.

Network configuration/training. We use the architecture of Figure 6 with $\phi = \mathrm{id}$ and train with an initial learning rate of 0.01, a momentum of 0.9 and a batch size of 100 for 200 epochs. The learning rate is halved every 40-th epoch.

Results. Table 1 lists the average cross-validation accuracies over 10 random 90/10 splits. Regarding approaches that also leverage information from persistent homology, we include the approaches of Section 6.2, as well as the *PSS kernel* from Reininghaus et al. (2015). For the PSS kernel, we train a kernel SVM using the averaged kernel matrices obtained for each filtration direction. In case of persistence images, multiple filtration directions simply increase the dimensionality of the vectorized input to the linear SVM, or the number of input channels in the neural network variant, respectively. When using SVMs, the cost parameter C is also determined via cross-validation on the training portion of each random split. Regarding non-topological strategies, we report the two best (†) and two worst (‡) results from Wang et al. (2014). While using topological signatures is below the top result on both datasets, we remark that *no* specific data preprocessing is required. This is in contrast to BCF or ICS which require contour extraction and are tailored to this type of data. Notably, in case of `MPEG-7`, the PSS kernel actually performs better than our approach. This can be attributed to the fact that the dataset is relatively small and we have to learn the parameters of our structure elements for all 16 input layers. In comparison, on the larger `Animal` dataset, learning a representation of barcodes already exhibits superior performance. We highlight that the proposed architecture readily generalizes to 3D with the only difference that, in this case, the filtration directions ξ_i are in $\mathbb{S}^2$. Finally, we observe that persistence images

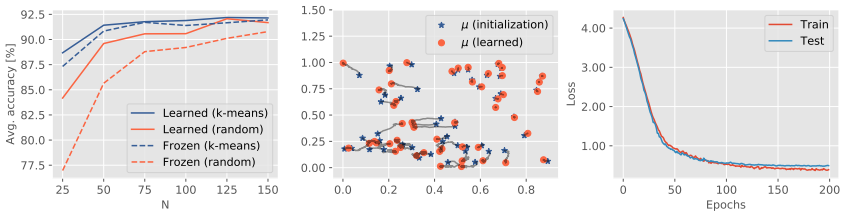


Figure 9: *Left*: Comparison, in terms of average cross-validation accuracy, of different initialization schemes, as a function of the number of structure elements, N, for two scenarios: (1) freezing the initial parameters and (2) learning the parametrization. *Middle*: Example of the change in location of the rational hat structure elements during training (for $N = 25$) and *random initialization*. *Right*: Training *vs.* testing loss over the training epochs for the same setup.

combined with a convolutional neural network, perform better than simple vectorization combined with a linear SVM. Also, initialization via k-means++ exhibits good performance on `MPEG-7`, but is inferior to learning the parametrization of the input layer on the `Animal` dataset.

6.3.1. Effect of learning a task-specific vectorization

On `MPEG-7`, we conduct a more detailed study on the impact of *learning* the parametrization of the proposed input layer *vs.* different initialization schemes and no learning. This means that we compare against a parametrization that is *only initialized*, frozen during training, and only the classification part (that is, everything after the `Concat` layer in Figure 6) of the network is trained. Notably, by construction, even a random initialization outputs a vectorization that is 1-Wasserstein stable, however it is not optimized for the learning task. We experimented with different numbers of structure elements N, as well as two initialization schemes: (1) random initialization and (2) initialization via k-means++ clustering (our default choice with $k = N$) of the points from all training diagrams. Figure 9 (*left*) shows the average cross-validation accuracy on `MPEG-7` as a function of the number of structure elements. *First*, we observe that initialization via k-means++ is superior to random initialization, no matter if the input layer is frozen or not. *Second*, we see that learning consistently improves performance, although, on this dataset and for small N, k-means++ initialization exhibits performance that is superior to random initialization and learning. This can be explained by the observation that random initialization typically requires a substantial change with respect to location which might not be achievable due to local minima in the energy landscape. Figure 9 (*middle*) shows an example of how the centers of each rational hat structure element change over the training epochs for $N = 25$ and randomly initialized locations. Finally, Figure 9 (*right*) shows an example of training vs. testing loss for the same parameter settings, revealing good generalization ability, as the testing loss closely tracks the training loss.

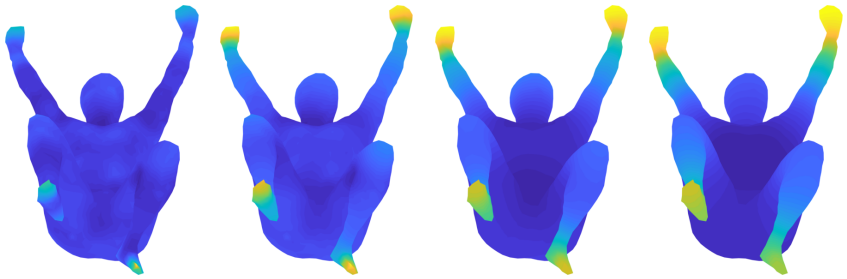


Figure 10: 3D surface meshes for one example object from the `SHREC14` evaluation benchmark (Pickup, D. et al., 2014). Vertices x are colored by the heat-distribution $h(x, t)$ at increasing values of t (from *left* to *right*). This relates to the scalar curvature of x at scale t.

6.4. Recognition of 3D shapes

In this experiment, we consider the problem of 3D object recognition, based on features of the object's surface mesh. We use the real watertight 3D surface meshes that are part of the `SHREC14` benchmark and replicate the evaluation setup of Reininghaus et al. (2015). In particular, this data set contains 400 watertight meshes, split into 40 classes with 10 samples per class. Each class represents one human in different poses.

Filtration. For each mesh, we compute the heat-kernel signature (HKS) of Sun et al. (2009) at 10 discrete time points t_i, which captures the local heat distribution $h(x, t_i)$ at mesh vertex x at scale t_i. Figure 10 illustrates the heat-distribution for one example object of the `SHREC14` benchmark as t_i increases (from *left* to *right*). Similar to Section 6.5, we lift the heat-distribution function to higher-dimensional simplices by taking the maximum over all faces and compute persistent homology of the sub level set filtration. Notably, in the kernel-based approach of Reininghaus et al. (2015), the optimal choice of t_i is determined via cross-validation and only diagrams for homology groups of dimension 0 are used[12]. In contrast, we use 0- *and* 1-dimensional homology and process the diagrams for each t_i directly via 20 parallel input layers (that is, 10 for dimension 0 and 10 for dimension 1). This is a natural choice and allows the network to focus on information relevant to the learning task.

Network configuration/training. We use the generic architecture of Figure 6 with $\phi = \mathrm{id}$ and 20 input branches, that is, two input branches per HKS scale t_i. The network is trained for 300 epochs with a batch size of 20 and an initial learning rate of 0.5. The learning rate is halved every 20-th epoch.

Results. Table 2 lists cross-validation accuracies in comparison to specifically-tailored approaches, reported in Pickup, D. et al. (2014). We remark that results listed as `APT`, `supDLtrainR` and `R-BiHDM-s` are obtained in a leave-one-out cross-validation setup using a 1-nearest neighbor classifier, due to the fact that Pickup, D. et al. (2014) assess *retrieval*

12. In particular, it is mentioned that 0- and 1-dimensional homology produced comparable results.

		SHREC14 (*Real*)
‡APT		**84.5**
‡supDLtrainR		79.3
‡R-BiHDM-s		68.5
Baseline (Bendich et al., 2016)	SVM^{lin}	41.3
†Persistence landscapes (Bubenik, 2015)	$\text{SVM}^{\text{kernel}}$	51.7
†PSS kernel (Reininghaus et al., 2015)	$\text{SVM}^{\text{kernel}}$	62.7
Sparse-TDA/Persistence images (Guo et al., 2018)	SVM^{lin}	68.8
Persistence images Adams et al. (2017)	NN	69.0
Ours (init. only, via k-means++)	NN	70.5
Ours (learned)	NN	72.2

Table 2: Average cross-validation accuracies (in %) on SHREC14 (*Real*), compared to the approaches discussed in Section 6.2, the best results (†) reported by Reininghaus et al. (2015) and a recent Sparse-TDA approach by Guo et al. (2018). For reference, we also list the top-3 approaches (‡) of Pickup, D. et al. (2014).

performance. This is obviously the extreme case of making maximum use of the training data, as only one sample is left out per cross-validation run. In comparison, all other results (using persistent homology) are averaged over 10 random 90/10 splits. While, leave-one-out evaluation would be computationally impractical in a neural network regime, we expect that our results could be better in such a setup due to an increased amount of available training data. As we replicate the evaluation setup from Reininghaus et al. (2015), we list the (best) results for the PSS kernel and persistence landscapes reported in this work. Additionally, we compare against the approaches listed in Section 6.2 and the Sparse-TDA approach of Guo et al. (2018) which is based on persistence images (Adams et al., 2017). Table 2 clearly shows that learning a task-specific vectorization of barcodes is beneficial.

6.5. Classification of social network graphs

Next, we consider the problem of classifying unlabeled and undirected graphs. A graph $\mathcal{G}$ is given by $\mathcal{G} = (V, E)$, where V denotes the set of vertices and E denotes the set of edges. This is already a valid simplicial complex. We evaluate our approach on the challenging task of classifying social networks, using the two largest benchmark datasets from Yanardag and Vishwanathan (2015), that is, reddit-5k (5 classes, ≈5k graphs) and reddit-12k (11 classes, ≈12k graphs). Each sample in these datasets represents a Reddit discussion graph and the classes indicate *subreddits* (for example, *worldnews*, *video*, etc.).

Filtration. The construction of a simplicial complex from $\mathcal{G} = (V, E)$ is straightforward: we set $K_0 = \{[v] \in V\}$ and $K_1 = \{[v_0, v_1] : \{v_0, v_1\} \in E\}$. We choose a simple filtration based on the *vertex degree*, that is, the number of incident edges to a vertex $v \in V$. Hence, for $[v_0] \in K_0$, we get $f([v_0]) = \deg(v_0)$ and again lift f to K_1 by taking the maximum. Note

that chain groups are trivial[13] for dimension > 1, hence, all features in dimension 1 are *essential*. Figure 11 (*top*) shows an illustration of the basic filtration principle and three exemplary sub level sets of a randomly selected sample from reddit-5k (*bottom*).

Network configuration/training. In this experiment, the network of Figure 6 has three input branches: two for 0-dimensional features (non-essential and essential) and one for essential 1-dimensional features. We use $\phi = \tanh$, as we observe many points with high multiplicity, due to the fact that the filtration values are natural numbers. Empirically, using tanh exhibits beneficial optimization behavior compared to, for example, normalizing the filtration function by the maximum degree as in Hofer et al. (2017a). We train the network for 300 epochs with a batch size of 100 and an initial learning rate of 0.5. The learning rate is halved every 20-th epoch.

Results. Table 3 compares our strategy to various state-of-the-art approaches from the literature. In particular, we compare against (1) the graphlet kernel (GK) and deep graphlet kernel (DGK) results from Yanardag and Vishwanathan (2015), (2) the Weisfeiler-Lehmann (WL) graph kernel (Shervashidze et al., 2011), (3) the Patchy-SAN (PSCN) results from Niepert et al. (2016) and (4) a graph-feature + random forest approach (RF) from Barnett et al. (2016). As we can see, using information from persistent homology in our proposed setting considerably outperforms the current state-of-the-art on both datasets by a large margin. This is an interesting observation, as PSCN (Niepert et al., 2016), for instance, also relies on vertex degrees and an extension of the convolution operation to graphs. Regarding approaches that use persistent homology, we compare against the PSS kernel and the approaches of Section 6.2. Additionally, we evaluate our approach *with* and *without* using essential features. The reason for this is to establish a fair comparison, as essential features are typically discarded and it is unclear how to properly include them in other methods. From a computational perspective, the PSS kernel did not terminate within a reasonable time frame (< 1 day), due to the large number of points per barcode and the fact that the computing *one entry* of the PSS kernel matrix scales with $\mathcal{O}(N \cdot M)$, where M, N denote the cardinalities of the two multi sets involved. As multiple kernel matrices need to be computed to cross-validate the PSS scale parameter, this becomes even more of a bottleneck. We remark that the results reported in Table 3 for our approach *without* essential features are different to (Hofer et al., 2017a); in fact, they are lower by ≈ 5 points. As, in practice, one would use *all* available information (that is, *including* essential features), we focused on this case in terms of network design. The same architecture was then used for the comparative experiments (that is, *without* essential features), but could be adjusted to obtain results similar to (Hofer et al., 2017a), in particular, 49% (reddit-5k) and 38.5% (reddit-12k).

6.6. Predicting the eigenvalue distribution of normalized graph Laplacians

As a first exploratory experiment, we investigate the problem of predicting the distribution of eigenvalues from normalized graph Laplacian matrices. Technically, this problem is similar to the social network classification experiment of Section 6.5, but differs in the type of learning target. We consider this a practically relevant task, as for large graphs, exact computation

13. The reason for this is that there are no 2-simplices in this setting.

		reddit-5k	reddit-12k
GK (Yanardag and Vishwanathan, 2015)		41.0	31.8
DGK (Yanardag and Vishwanathan, 2015)		41.3	32.2
PSCN (Niepert et al., 2016)		49.1	41.3
RF (Barnett et al., 2016)		**50.9**	**42.7**
WL graph kernel (Shervashidze et al., 2011)		47.7	38.5
Baseline (Bendich et al., 2016)	$\texttt{SVM}^{\texttt{lin}}$	45.2	31.6
PSS kernel (Reininghaus et al., 2015)	$\texttt{SVM}^{\texttt{kernel}}$	n/a	n/a
Persistence images (Adams et al., 2017)	$\texttt{SVM}^{\texttt{lin}}$	44.9	34.6
Persistence images (Adams et al., 2017)	NN	46.7	35.1
Ours (w/o essential, learned)	NN	44.5	34.1
Ours (w/ essential, init. only, via k-means++)	NN	53.5	45.8
Ours (w/ essential, learned)	NN	54.8	46.0

Table 3: Average cross-validation accuracies (in %) on the `reddit-5k` and `reddit-12k` benchmark datasets. The *middle/bottom* part of the table lists approaches that use persistent homology. For the PSS kernel, `n/a` denotes that computation of the Gram matrices (required to cross-validate the PSS scale parameter) did not terminate within a reasonable time window ($<$ 1 day).

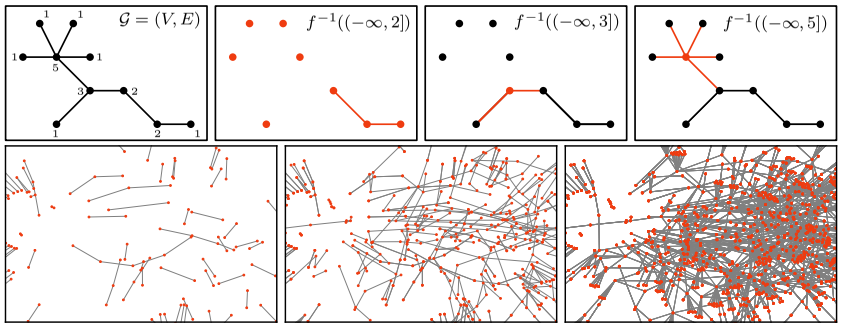


Figure 11: *Top:* Principle of graph filtration by vertex degree on a toy graph. Simplices that are added as the filtration value a_i increases are marked in red. *Bottom:* one real `reddit_5k` graph for filtration values $a_i \in \{2, 5, 100\}$ (from *left* to *right*).

of all eigenvalues is computationally intensive and time-consuming. A relevant question in this context is, whether it is possible to learn on small graphs and predict on larger graphs. Accurate prediction of the eigenvalue spectrum could constitute a lightweight approach to assess the conformity of graph generation approaches with respect to real world data.

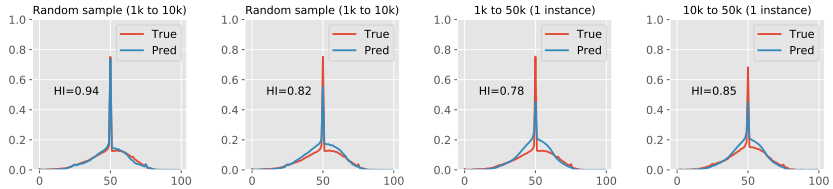


Figure 12: *Left*: two exemplary results for predicting the distribution (discretized to 100 bins) of normalized graph Laplacian eigenvalues. *Right*: prediction results for training on graphs of with $|V| = 10^3$ and $|V| = 10^4$ vertices and testing on a graph with $|V| = 5 \cdot 10^4$ vertices. HI denotes the value of the (symmetric) histogram intersection metric.

Dataset. To explore this problem, we use the `Pokec` (Takak and Zabovsky, 2012) social network graph that is part of the `SNAP` repository[14]. The full `Pokec` graph $\mathcal{G}(V, E)$ contains $|V| = 1,632,803$ vertices with $|E| = 30,622,564$ edges. We extracted two collections of subgraphs using breadth-first search (BFS), starting from randomly selected vertices of V with the constraint that 1-hop neighbors of vertices are not picked as starting points for BFS to reduce subgraph overlap. The first collection, S_1 contains 5000 subgraphs with 10^3 vertices, the second collection, S_2, contains 3000 subgraphs with 10^4 vertices. For both, S_1 and S_2 exact eigenvalue computation is time-consuming, but still possible. For exploration, we also computed the eigenvalue spectrum for *one* subgraph, $\mathcal{G}^*$, with $5 \cdot 10^4$ vertices in the same manner (with computation time $\approx$ 1-2 days). Our goal is to predict a histogram of normalized graph Laplacian eigenvalues λ_i (with $\lambda_i \in [0, 2]$). We chose a fixed binning of the eigenvalues into 100 bins, based on the eigenvalues of the largest graph and the Freedman-Diaconis rule for bin size (Freedman and Diaconis, 1981).

Filtration. The filtration is done as in Section 6.5, that is, by vertex degree. We only consider 0-dimensional persistent homology without essential features. The reason for this is that, due to the construction of the subgraphs (via BFS), we would only have one essential feature in H_0. Also, we were unable to handle 1-dimensional features, due to the vast number of cycles produced and the memory limitations of our GPU(s). Also, we did not want to introduce arbitrary thresholds for points in the barcode.

Network training/configuration. In this setup, we only have one input branch in the network of Figure 6 and $\phi = \tanh$. Different to previous sections, the loss function needs to be chosen differently. In particular, we decided to use the (symmetrized) histogram intersection (HI) metric (with range $[0, 1]$) as a loss function. A value of 0 indicates no overlap, a value of 1 indicates perfect overlap of two histograms.

Results. Three scenarios were evaluated: *First*, training on S_1 and testing on S_2, for which we obtain an average HI score of 0.85. This indicates remarkable performance, considering that the test set S_2 contains graphs that are one magnitude larger (in terms of vertices) than

14. https://snap.stanford.edu/data/soc-pokec.html

graphs in S_1. Figure 12 (*left*) shows two example prediction results. *Second*, for comparison, we trained on S_2 and tested on S_2 in a 90/10 cross-validation setup. In that case, the average HI increases to 0.94, indicating the larger graphs contain more structure that is relevant to the prediction task. When training on S_1, or S_2, respectively, and testing on $\mathcal{G}^*$, we obtain HI scores of 0.78 and 0.85, illustrated in Figure 12 (*right*). We consider this an encouraging result, as computing the filtration as well as one forward pass through the trained model only takes a fraction of the time required to compute the eigenvalues of $\mathcal{G}^*$.

6.7. Recognizing activity from EEG signals

As a second exploratory experiment, we investigate the problem of recognizing human activity, using recorded electroencephalography (EEG) signals. From an application point of view, this is a challenging problem and a lot of effort is put into developing brain-computer-interfaces (BCI) which rely solely on brain activity to, for example, facilitate robotic assistance systems for people with disabilities.

Dataset. The dataset we use in this experiment was collected by the Department of Neurology, Paracelsus Medical University Salzburg, Austria. It consists of 22 healthy participants as well as 7 participants in need of a wheelchair after spinal chord injury. The subjects were asked to execute 7 different tasks: *moving* of a hand/foot (2 tasks), *imagining to move* a hand/foot (2 tasks), *observing* a movie showing movement of a hand/foot (2 tasks) and *staying* in a resting condition, that is, no movement at all (1 task). In terms of recording the EEG signals, a geodesic sensor net with 256 channels was used. Consequently, each data point in the dataset is a collection of 256 time-intensity curves in combination with the corresponding task label. The EEG signal during each task was recorded over 25 trials where each trial was divided in 5 sub-runs, that is, repetitions of the given task. In total the dataset contains $\approx$ 31k such curves (some of the patients had more than one recording). Figure 13 shows example EEG signals of one sensor for three different actions.

Filtration. As persistent homology (partly) emerged from Morse theory, its application on time-intensity curves is straightforward. In our case, we interpreted the time-intensity sequences as a polyline, that is, edges are inserted between time-intensity points which are neighbors on the time axis. As a preprocessing step, intensities were mean and standard-deviation normalized. The resulting polylines were filtered along the intensity axis from top *and* bottom, leading to *two* barcodes for each channel of the EEG signal. For computational reasons, we used a sub-selection of 20 channels. This sub-selection represents a low-resolution sensor configuration. As only homology groups of dimension 0 are non-trivial for this data, we obtain two diagrams per sensor.

Network configuration/training. We use the architecture of Figure 6 with two input branches per sensor and $\phi = \mathrm{id}$. The network is trained for 100 epochs with a batch size of 300 and an initial learning rate of 0.1. The learning rate halved after every 10-th epoch.

Results. Table 4 lists the average cross-validation accuracies for 10 random 90/10 splits of the dataset. We chose to compare against our *Baseline*, as well as a linear SVM trained on mean and standard deviation normalized EEG sequences, reduced to 250 dimensions via

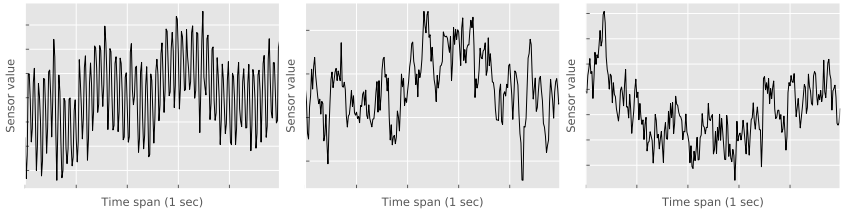


Figure 13: Exemplary EEG signals from one (out of 256) sensor for three different actions.

PCA (250 components already cover > 99% of the variance). The number of samples in this classification problem did not allow us to evaluate the PSS kernel within reasonable computation time (< 1 day). For persistence images, concatenation of the discretized persistence surface(s), even for a coarse 20×20 grid, leads to 16000 dimensional input vectors, which also posed computational problems. Hence, only persistence images in combination with a convolutional neural network (see Section 6.2) are included in Table 4 as a strong baseline. As we can see, the proposed approach exhibits considerably better performance than all competitors and, most importantly, easily handles datasets of this size with multiple parallel input paths.

		EEG sequences
PCA	$\mathtt{SVM}^{\mathtt{lin}}$	**30.5**
Baseline (Bendich et al., 2016)	$\mathtt{SVM}^{\mathtt{lin}}$	21.1
PSS kernel (Reininghaus et al., 2015)	$\mathtt{SVM}^{\mathtt{kernel}}$	n/a
Persistence images (Adams et al., 2017)	$\mathtt{SVM}^{\mathtt{lin}}$	n/a
Persistence images (Adams et al., 2017)	NN	30.4
Ours	NN	37.4

Table 4: Average cross-validation accuracies (in %) on the EEG dataset, compared to a non persistent homology approach (PCA + linear SVM), our *Baseline*, as well as persistence images combined with a convolutional neural network, see Section 6.2. Approaches marked as n/a did not finish within 1 day of computation time.

7. Discussion

We presented an in-depth theoretical analysis of a recently proposed (Hofer et al., 2017a) approach to learn task-specific vectorizations of barcodes. This analysis has not only highlighted shortcomings of the original concept, but also revealed alternatives to mitigate these shortcomings. We summarize and discuss the main contributions next.

Learnable vectorizations. We started by studying the problem of how to vectorize multi sets over $\mathbb{R}^2$ in a parametrized manner. This led to the concept of structure elements which

induce a real-valued mapping of multi sets. In particular, N structure elements lead to an N-dimensional vectorization upon concatenation. Different to existing approaches in the context of machine learning compatible representations of persistence diagrams, this mapping is not *fixed* a-priori, but can be optimized, for example, within a neural network regime. Our experimental study on datasets from different domains confirms that learning a vectorization of persistence diagrams is preferable over non-data driven approaches. Additionally, our results demonstrate that topological features of data can be beneficial in many learning tasks, not necessarily to replace existing inputs, but rather as a complementary source of discriminative information.

Continuity properties. We subsequently introduced conditions on the structure elements under which important continuity properties (such as stability) of persistence diagrams can be retained by the induced mapping. Our analysis also revealed (1) that we cannot achieve stability stronger than 1-Wasserstein (w_1^δ), but (2) the induced mapping is continuous with respect to w_p^δ if a specific growth-condition is satisfied. We consider this particularly relevant, as it might open up the path to learn a filtration for a given simplicial complex. For example, in case of graphs, using the degree function is an ad-hoc choice. From a learning perspective, one would rather favor a strategy that learns a task-specific mapping from vertex meta-data $x \in \mathbb{R}^n$ to $\mathbb{R}$ which can then be used to compute sub level set filtration. This is certainly an interesting research direction for future work.

Neural network integration. From a practical perspective, it is imperative that (1) the required conditions on new structure elements can be verified easily and (2) that any new layer integrates well into existing network architectures. To address the first issue, we studied a decomposition of structure elements into a composition of two fixed coordinate transforms (τ_ν and ρ) and a function t defined on $\mathbb{R}^2$. This is beneficial, as new structure elements can be easily constructed through different choices of t. Further, verifying w_1^q-stability of the induced mapping boils down to control the behavior of $t \circ \tau_\nu$ on a small part of the domain. Notably, this decomposition turned out to be equally beneficial for studying parameter gradients, as one can focus on the parameters of t. Starting from the popular Gaussian radial basis function (as our initial choice for t) and the observation that its parameter gradients greatly differ in terms of scaling, we introduced two radial alternatives with fewer parameters and gradients with comparable order of growth.

Open problems. While the proposed approach facilitates to leverage information from persistent homology in deep learning, it is unclear how to appropriately choose the number of structure elements. Although, empirically, we observe fairly stable performance with a reasonable number of elements, it would be desirable to make this decision in a more informed manner. Additionally, the particular functional form of the structure elements can most likely be further improved (or simplified), possibly with even more linear parameter gradients. Finally, we consider the problem of filtration learning to be of great interest, especially in cases where a fixed simplicial complex (for example, a graph or mesh) is given and the learner decides on an suitable filtration function for persistent homology.

Acknowledgements

This work was partially funded by the Austrian Science Fund (FWF project KLI 00012 and FWF project P 31799), the Spinal Cord Injury and Tissue Regeneration Center Salzburg (SCI-TReCS), Paracelsus Medical University, Salzburg and NSF grant NSF EECS-1610762. We also thank Robert Elsässer and Gregor Bankhamer for providing the subgraphs for the experiments in Section 6.6, funded by the Austrian Science Fund (FWF project P 27613).

Appendix A. Notation

Symbol	Meaning
Δ	Diagonal $\Delta = \{(x_0, x_1) \in \mathbb{R}^2 : x_0 = x_1\}$
Ω	Upper-diagonal part of $\mathbb{R}^2$, that is, $\Omega = \{(x_0, x_1) \in \mathbb{R}^2 : x_1 > x_0\}$
$\mathcal{D}_i^{n,j}$	Barcode of n-th homology group, i-th data object and j-th filtration
$\mathbb{D}$	Set of barcodes with finite cardinality
w_p^q, w_∞	p-Wasserstein distance (using $\|\cdot\|_q$) and Bottleneck distance
$\langle x, y\rangle$	Inner product in $\mathbb{R}^n$ for $x, y \in \mathbb{R}^n$
$\mathbb{S}^n$	$(n+1)$ dimensional unit sphere
H_n	n-th homology group
$[v_0, \ldots, v_n]$	n-dimensional simplex
K_n	n-skeleton of a simplicial complex K
$\mathcal{G}(V, E)$	Undirected graph with vertex set V and edge set E
$[a, b], [a, b), (b, a], (a, b)$	Closed, left-open, right-open and open interval for $a < b$, $a, b \in \mathbb{R}$

Table 5: Commonly used symbols used throughout the manuscript.

Appendix B. Proofs

B.1. Proof of Lemma 11

Proof To show this claim, we leverage the definition of a Δ-relative bijective matching (cf. Definition 4) and a beneficial re-arrangements of the involved sums.

$$\begin{aligned}
\sum_{x\in\mathcal{D}} f(x) - \sum_{y\in\mathcal{E}} f(y) &= \sum_{x\in\mathrm{supp}(\mathcal{D})} \mathrm{mult}_{\mathcal{D}}(x) f(x) - \sum_{y\in\mathrm{supp}(\mathcal{E})} \mathrm{mult}_{\mathcal{E}}(x) f(y) \\
&= \sum_{x\in\mathrm{supp}(\mathcal{D})} \left(\sum_{y\in\mathrm{supp}(\mathcal{E})} \varphi(x,y) \right) f(x) - \sum_{y\in\mathrm{supp}(\mathcal{E})} \left(\sum_{y\in\mathrm{supp}(\mathcal{D})} \varphi(x,y) \right) f(y) \\
&= \sum_{(x,y)\in\mathrm{supp}(\mathcal{D})\times\mathrm{supp}(\mathcal{E})} \mathrm{mult}_{\varphi}\big((x,y)\big) f(x) - \sum_{(y,x)\in\mathrm{supp}(\mathcal{E})\times\mathrm{supp}(\mathcal{D})} \mathrm{mult}_{\varphi}\big((x,y)\big) f(y) \\
&= \sum_{(x,y)\in\mathrm{supp}(\varphi)} \mathrm{mult}_{\varphi}\big((x,y)\big) f(x) - \sum_{(x,y)\in\mathrm{supp}(\varphi)} \mathrm{mult}_{\varphi}\big((x,y)\big) f(y) \\
&= \sum_{(x,y)\in\mathrm{supp}(\varphi)} \mathrm{mult}_{\varphi}\big((x,y)\big) \big(f(x) - f(y)\big) \ . \qquad (\text{as } |\mathrm{supp}(\varphi)| < \infty)
\end{aligned}$$

■

B.2. Proof of Lemma 20

Proof As $t^{\mathtt{exp}}_{(\sigma,\mu)}$ is Lipschitz continuous on $\mathbb{R}\times\mathbb{R}$, we follow the argumentation of Section 5.3. That is, it suffices to show that

$$\lim_{x\to 0} \left| \left(\frac{\partial}{\partial x_1} t^{\mathtt{exp}}_{(\sigma,\mu)} \circ \tau_\nu \right)(x) \right| < C \tag{11}$$

holds for $C \in \mathbb{R}$. Let $x < \varepsilon < \nu$. We get

$$\begin{aligned}
\left| \left(\frac{\partial}{\partial x_1} t^{\mathtt{exp}}_{(\sigma,\mu)} \circ \tau_\nu \right)(x) \right| &= \left| \left(\frac{\partial}{\partial x_1} e^{-\sigma_0^2(x_0-\mu_0)^2 - \sigma_1^2(2\nu-\nu^2/x_1-\mu_1)^2} \right)(x) \right| \\
&= |c_0| \cdot \left| \left(\frac{\partial}{\partial x_1} e^{-\sigma_1^2(2\nu-\nu^2/x_1-\mu_1)^2} \right)(x) \right| \\
&= |c_0| \cdot \left| e^{-\sigma_1^2(2\nu-\nu^2/x-\mu_1)^2} \cdot 2\,(-\sigma_1^2)(2\nu-\nu^2/x-\mu_1) \cdot (-1)\frac{\nu^2}{x^2} \right| \\
&= |c_0| \cdot \left| c_1\, e^{(\cdots)} \cdot \frac{1}{x^2} + c_2\, e^{(\cdots)} \cdot \frac{1}{x^3} + c_3\, e^{(\cdots)} \cdot \frac{1}{x^2} \right| \\
&\leq 3 \cdot |c_0| \cdot \max\{|c_1|, |c_2|, |c_3|\} \cdot \left| e^{-\sigma_1^2(2\nu-\nu^2/x-\mu_1)^2} \cdot \frac{1}{x^3} \right| \\
&= c_4 \cdot \underbrace{\left| \frac{\frac{1}{x^3}}{e^{\sigma_1^2(2\nu-\nu^2/x-\mu_1)^2}} \right|}_{\mathrm{I}}
\end{aligned}$$

for x small enough (we aggregated constants into c_i, $0 \leq i \leq 4$ for readability). Applying the de l'Hôpital's rule iteratively on I yields $\mathrm{I} \to 0$. Consequently, Eq. (11) is satisfied. ■

B.3. Proof of Lemma 22

Proof Similar to the proof of Lemma 20, we follow the argumentation of Section 5.3 and note that $t^{\mathtt{rat}}_{(\sigma,\mu,\alpha)}$ is Lipschitz continuous on $\mathbb{R}\times\mathbb{R}$.

Let $\varepsilon < \nu$ and $2\nu - \nu^2/\varepsilon < \mu_1$. Then, for $x < \varepsilon$

$$\left| 2\nu - \frac{\nu^2}{x} - \mu_1 \right| = \mu_1 - 2\nu + \frac{\nu^2}{x}$$

and we circumvent the fact that $|\cdot|$ is not differentiable at 0. Now consider

$$
\begin{aligned}
\left|\left(\frac{\partial}{\partial x_1} t^{\mathtt{rat}}_{(\mu,\sigma,\alpha)}\right)(x)\right| &= \left|\left(\frac{\partial}{\partial x_1}\left(1+|\sigma_0|\cdot|x_0-\mu_0|+|\sigma_1|\cdot\left(\mu_1-2\nu+\frac{\nu^2}{x_1}\right)\right)^{-\alpha}\right)(x)\right| \\
&= \left|\left(\frac{\partial}{\partial x_1}\left(c_0+\frac{|\sigma_1|\nu^2}{x_1}\right)^{-\alpha}\right)(x)\right| \\
&= \left|-\alpha\cdot\left(c_0+\frac{|\sigma_1|\nu^2}{x}\right)^{-\alpha-1}\cdot(-1)\frac{|\sigma_1|\nu^2}{x^2}\right| \\
&= c_1\cdot\left|\left(c_0+\frac{|\sigma_1|\nu^2}{x}\right)^{-\alpha-1}\cdot\frac{1}{x^2}\right| \\
&\overset{*}{\leq} c_1\cdot\left|\left(c_0+\frac{|\sigma_1|\nu^2}{x}\right)^{-1-1}\cdot\frac{1}{x^2}\right| \\
&= c_1\cdot\left|\frac{\frac{1}{x^2}}{\left(c_0^2+2\cdot c_0\cdot\frac{|\sigma_1|\nu^2}{x}+\frac{|\sigma_1|^2\nu^4}{x^2}\right)}\right| \xrightarrow{x\to 0} \frac{c_1}{|\sigma_1|^2\nu^4}
\end{aligned}
$$

where we collapse constant terms into c_0, c_1 and $(*)$ holds for $x \in (0,\varepsilon)$ small enough, as $y^{-(\alpha+1)} \leq y^{-2}$ for $\alpha \geq 1$ and y large enough $\Rightarrow$ the partial derivative is bounded on $(0,\varepsilon)$. ■

Appendix C. Neural network configurations

Dataset		H_0	H_0^e	H_1^e	ϕ	MLP
MPEG-7	N	150	-	-	id	Blk[2400,600,0.3]
	S	16	-	-		Blk[600,150,0.2]
						Lin[150,70,0.1]
Animal-7	N	100	-	-	id	Blk[1600,400,0.3]
	S	16	-	-		Blk[400,100,0.2]
						Lin[100,20,0.1]
Reddit-5k	N	150	5	50	tanh	Blk[205,410,0.2]
	S	1	1	1		Blk[410,104,0.2]
						Blk[104,25,-]
						Lin[25,5,-]
Reddit-12k	N	150	5	50	tanh	Blk[205,410,0.2]
	S	1	1	1		Blk[410,104,0.2]
						Blk[104,25,-]
						Lin[25,12,-]
Pokec	N	100	-	-	tanh	Blk[100,100,-]
	S	1	-	-		Blk[100,100,0.2]
						Lin[100,100,-]
SHREC 2014	N	40	-	-	id	Blk[800,800,-]
	S	10	-	-		Blk[800,200,-]
						Lin[200,40,-]

Table 6: Configuration of the neural network in Figure 6 per experiment/dataset.

References

H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *JMLR*, 18(8):1–35, 2017.

A. Adcock, E. Carlsson, and G. Carlsson. The ring of algebraic functions on persistence bar codes. *Homol. Homotopy Appl.*, 18:381–402, 2016.

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *SODA*, 2007.

X. Bai, W. Liu, and Z. Tu. Integrating contour and skeleton for shape classification. In *ICCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*, 2009.

I. Barnett, N. Malik, M.L. Kuijjer, P.J. Mucha, and J.-P. Onnela. Feature-based classification of networks. *CoRR*, 2016. https://arxiv.org/abs/1610.05868.

U. Bauer, M. Kerber, and J. Reininghaus. Distributed computation of persistent homology. In *ALENEX*, 2014.

P. Bendich, J.S. Marron, E. Miller, A. Pieloch, and S. Skwerer. Persistent homology analysis of brain artery trees. *Ann. Appl. Stat.*, 10(2), 2016.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, London, UK, 2004.

P. Bubenik. Statistical topological data analysis using persistence landscapes. *JMLR*, 16(1): 77–102, 2015.

Z. Cang and G.-W. Wei. Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput. Biol.*, 13(7):e1005690, 2017.

G. Carlsson. Topology and data. *Bull. Amer. Math. Soc.*, 46:255–308, 2009.

G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian. On the local behavior of spaces of natural images. *IJCV*, 76:1–12, 2008.

M. Carriére, M. Cuturi, and S. Outot. Sliced Wasserstein kernel for persistence diagrams. In *ICML*, 2017.

O. Chapelle. Training a support vector machine in the primal. *Neural Comput.*, 19(5): 1155–78, 2007.

F. Chazal and V. Divol. The density of expected persistence diagrams and its kernel based estimation. In *SoCG*, 2018.

F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot. Gromov-Hausdorff stable signatures for shapes using persistence. *Comput. Graph. Forum*, 28(5):13931403, 2009.

F. Chazal, B.T. Fasy, F. Lecci, A. Rinaldo, A. Singh, and L. Wasserman. On the bootstrap for persistence diagrams and landscapes. *Model. Anal. Inform. Sist.*, 20(6):111–120, 2013a.

F. Chazal, L.J. Guibas, S.Y. Oudot, and P. Skraba. Persistence-based clustering in Riemannian manifolds. *J. ACM*, 60(6):41–79, 2013b.

F. Chazal, B.T. Fasy, F. Lecci, A. Rinaldo, and L. Wassermann. Stochastic convergence of persistence landscapes and silhouettes. *JoCG*, 6(2):140161, 2014.

Y.-C. Chen, D. Wang, A. Rinaldo, and L. Wasserman. Statistical analysis of persistence intensity functions. *CoRR*, 2015. https://arxiv.org/abs/1510.02502.

D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, 2007.

D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have L_p-stable persistence. *Found. Comput. Math.*, 10(2):127–139, 2010.

H. Edelsbrunner and J. L. Harer. *Computational Topology : An Introduction*. American Mathematical Society, 2010.

H. Edelsbrunner, D. Letcher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28(4):511–533, 2002.

H. Edelsbrunner, A. Ivanov, and R. Karasev. Current open problems in discrete and computational geometry. *Modelirovanie i Analiz Informats. Sistem*, 19(5):5–17, 2012.

B. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh. Confidence sets for persistence diagrams. *Ann. Statist.*, 42(6):2301–2339, 2014.

D. Freedman and P. Diaconis. On the histogram as a density estimator: l_2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57:453–476, 1981.

A. Graves. Generating sequences with recurrent neural networks. *CoRR*, 2013. https://arxiv.org/abs/1308.0850.

A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *JMLR*, 13:723–773, 2012.

W. Guo, K. Manohar, S.L. Brunton, and A.G. Banerjee. Sparse-TDA: Sparse realization of topological data analysis for multi-way classification. *IEEE Trans. Knowl. Data Eng.*, 2018.

A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2002.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep learning with topological signatures. In *NIPS*, 2017a.

C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Constructing shape spaces from a topological perspective. In *IPMI*, 2017b.

G. Huang, Z. Liu, L. van der Maaten, and K. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

S. Kališnik Verovšek. Tropical coordinates on the space of persistent barcodes. *Found. Comput. Math.*, pages 1–29, 2018.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

G. Kusano, K. Fukumizu, and Y. Hiraoka. Persistence weighted Gaussian kernel for topological data analysis. In *ICML*, 2016.

R. Kwitt, S. Huber, M. Niethammer, W. Lin, and U. Bauer. Statistical topological data analysis - a kernel perspective. In *NIPS*, 2015.

L. Latecki, R. Lakamper, and T. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, 2000.

C. Li, M. Ovsjanikov, and F. Chazal. Persistence-based structural recognition. In *CVPR*, 2014.

Y. Mileyko, S. Mukherjee, and J. Harer. Probability measures on the space of persistence diagrams. *Inverse Probl.*, 27(12), 2011.

K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete Comput. Geom.*, 50(2):330–353, 2013.

A. Monod, S. Kališnik-Verovšek, J.A. Patiño Galindo, and L. Crawford. Tropical sufficient statistics for persistent homology. *CoRR*, 2017. https://arxiv.org/abs/1709.02647.

M. Nicolau, A.J. Levine, and G. Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *PNAS*, 108(17): 7265–7270, 2011.

M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, 2016.

Pickup, D. et al. SHREC '14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, EG 3DOR'14. Eurographics Association, 2014.

C.R. Qi, H. Su, K. Mo, and L.J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017a.

C.R. Qi, L. Yi, H. Su, and L.J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017b.

R. Reininghaus, U. Bauer, S. Huber, and R. Kwitt. A stable multi-scale kernel for topological machine learning. In *CVPR*, 2015.

B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

N. Shervashidze, P. Schweitzer, E.J. van Leeuwen, K. Mehlhorn, and K.M Borgwardt. Weisfeiler-lehmann graph kernels. *JMLR*, 12:2539–2561, 2011.

G. Singh, F. Memoli, T. Ishkhanov, G. Sapiro, G. Carlsson, and D.L. Ringach. Topological analysis of population activity in visual cortex. *J. Vis.*, 8(8), 2008.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 1st edition, 2008.

J. Sun, M. Ovsjanikov, and L. Guibas. A concise and probably informative multi-scale signature based on heat diffusion. In *SGP*, 2009.

K. Sun and B. Super. Classification of contour shapes using class segment sets. In *CVPR*, 2005.

I. Sutskever, O. Vinyals, and Q.V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

L. Takak and M. Zabovsky. Data analysis in public social networks. In *International Scientific Conference & International Workshop Present Day Trends of Innovations*, 2012.

K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet means for distributions of persistence diagrams. *Discrete Comput. Geom.*, 52(1):44–70, 2014a.

K. Turner, S. Mukherjee, and D. M. Boyer. Persistent homology transform for modeling shapes and surfaces. *Inf. Inference*, 3(4):310–344, 2014b.

X. Wang, B. Feng, X. Bai, W. Liu, and L.J. Latecki. Bag of contour fragments for robust shape classification. *Pattern Recognit.*, 47(6):2116–2125, 2014.

S. Welleck, K. Cho, and Z. Zhang. Saliency-based sequential image attention with multiset prediction. In *NIPS*, 2017.

P. Yanardag and S.V.N. Vishwanathan. Deep graph kernels. In *KDD*, 2015.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. J. Smola. Deep sets. In *NIPS*, 2017.

A. Zomorodian and G. Carlsson. Computing persistent homology. In *SCG*, 2004.

Journal of Machine Learning Research 20 (2019) 1-32 Submitted 6/18; Revised 6/19; Published 8/19

Causal Learning via Manifold Regularization

Steven M. Hill* STEVEN.HILL@MRC-BSU.CAM.AC.UK
MRC Biostatistics Unit
University of Cambridge
Cambridge, CB2 0SR, UK

Chris J. Oates* CHRIS.OATES@NCL.AC.UK
School of Mathematics, Statistics and Physics
Newcastle University
Newcastle-upon-Tyne, NE1 7RU, UK

Duncan A. Blythe DUNCANBLYTHE@GOOGLEMAIL.COM
Sach Mukherjee* SACH.MUKHERJEE@DZNE.DE
German Center for Neurodegenerative Diseases
53127 Bonn, Germany

Editor: Peter Spirtes

Abstract

This paper frames causal structure estimation as a machine learning task. The idea is to treat indicators of causal relationships between variables as 'labels' and to exploit available data on the variables of interest to provide features for the labelling task. Background scientific knowledge or any available interventional data provide labels on some causal relationships and the remainder are treated as unlabelled. To illustrate the key ideas, we develop a distance-based approach (based on bivariate histograms) within a manifold regularization framework. We present empirical results on three different biological data sets (including examples where causal effects can be verified by experimental intervention), that together demonstrate the efficacy and general nature of the approach as well as its simplicity from a user's point of view.

Keywords: causal learning, manifold regularization, semi-supervised learning, interventional data, causal graphs

1. Introduction

Causal structure learning is concerned with learning causal relationships between variables. Such relationships are often represented using directed graphs with nodes corresponding to the variables of interest. Consider a set of p variables or nodes indexed by $V = \{1, \ldots, p\}$. The aspect we focus on in this paper is to determine, for each (ordered) pair $(i, j) \in V \times V$, whether or not node i exerts a causal influence on node j. In particular, our focus is on the binary 'detection' problem (of learning whether or not node i exerts a causal influence on node j) rather than estimation of the magnitude of any causal effect.

Methods for learning causal structures can be usefully classified according to whether the graph is intended to encode direct or total (ancestral) causal relationships. For example

*. These authors contributed equally to this work.

©2019 Steven Hill, Chris Oates, Duncan Blythe, Sach Mukherjee.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-383.html.

if variable A acts on B which in turn acts on C, A has an ancestral effect on C (via B). Here, the graph of direct effects has edges $A \to B \to C$, while the graph of total or ancestral effects has in addition the edge $A \to C$. Methods based on (causal) directed acyclic graphs (DAGs) are a natural and popular choice for causal discovery (Spirtes et al., 2000; Pearl, 2009). The PC algorithm (Spirtes et al., 2000) is an important example of such a method. Using a sequence of tests of conditional independence, the PC algorithm estimates an underlying causal DAG. Due to the fact that the graph may not be identifiable, the output is an equivalence class of DAGs (encoded as a completed partially directed acyclic graph or CPDAG). Here the estimand is intended to encode direct influences. IDA (Intervention calculus when the DAG is Absent; Maathuis et al., 2009) uses the PC output to bound the quantitative total causal effect of any node i on any other node j. These estimated effects can be thresholded to provide a set of edges. FCI (Fast Causal Inference; Spirtes et al., 2000) and RFCI (Really Fast Causal Inference; Colombo et al., 2012) consider a type of ancestral graph as estimand and allow for latent variables. Greedy Interventional Equivalence Search (GIES; Hauser and Bühlmann, 2012) is a score-based approach that allows for the inclusion of interventional data.

Methods for learning causal structures (such as those above) are often rooted in data-generating causal models. In a quite different vein, there have been some interesting recent efforts in the direction of labelling pairs of variables as causal or otherwise, such as in Lopez-Paz et al. (2015) and Mooij et al. (2016). These approaches are 'discriminative' in spirit, in the sense that they need not be rooted in an explicit data-generating model; rather the emphasis is on learning how to tell causal and non-causal apart. Our work is in this latter vein. We address a specific aspect of causal learning—that of estimating edges in a graph encoding causal relationships between a defined set of vertices—but via a machine learning approach that allows the inclusion of any available information concerning known cause-effect relationships. The output of our method is a directed graph that need not be acyclic (see Spirtes, 1995; Richardson, 1996; Hyttinen et al., 2012, for discussion of cyclic causality) and whose edges may encode either direct or total/ancestral relationships, as discussed below. The main differences between our work and previous work on labelling causal pairs (Lopez-Paz et al., 2015; Mooij et al., 2016) are the specific methods and associated theory that we put forward, the manifold regularization framework, and the empirical examples.

In general terms the idea is as follows: let $\mathcal{D}$ denote the available data and Φ denote any available knowledge on causal relationships among the variables indexed in V (e.g., based on background knowledge or experimental intervention). We view the causal learning task in terms of constructing an estimator of the form $\hat{G}(\mathcal{D}, \Phi)$, where $\hat{G}$ is a directed graph with vertex set V and edge set $E(\hat{G})$, with $(i, j) \in E(\hat{G})$ corresponding to the claim that variable i has a causal influence on variable j. To put this another way: entries in a binary adjacency matrix encoding causal relationships are treated as 'labels' in a machine learning sense. From this point of view, the task of constructing the estimator $\hat{G}(\mathcal{D}, \Phi)$ is essentially one of learning these labels from available data and from any *a priori* known labels (derived from Φ). Thus, a key difference with respect to a number of existing methods is the nature of the inputs needed: our approach requires causal background information Φ as an input while several existing methods (such as PC) use only observational data. The casual background information Φ need not be interventional data *per se*, but must encode knowledge on some causal relationships in the system (we consider both scenarios in empirical examples below).

Note also that in our approach the causal status of multiple pairs is coupled via the learning scheme: loosely speaking (see below for technical details), it is the position of a test pair on a classification manifold (relative to other pairs) that determines its status.

Our approach differs in several ways from graphical model-based methods. In our approach, the same framework can be used to estimate either direct or ancestral causal relationships, depending on the precise input (we show real data examples of both tasks below). This is because the classifier can be agnostic to the label semantics: provided the Bayes' risk for the label of interest is sufficiently low, these labels can in principle be learned. In contrast to much of the literature, our approach does not try to provide a full data-generating model of the causal system but instead focuses on the specific problem of learning edges encoding causal relationships. As we see in experiments below, this can lead to good empirical performance, but the output is in a sense less rich than a full causal model (see the Discussion). Our work is motivated by scientific problems where good performance with respect to this narrower task can be useful in reducing the hypothesis space and targeting future work.

The remainder of the paper is organized as follows. We first introduce some notation and discuss in more detail how causal learning can be viewed as a semi-supervised task. We then discuss a specific instantiation of the general approach, based on manifold regularization using a simple bivariate featurization. Using this specific approach—which we call Manifold Regularized Causal Learning (MRCL)—we present empirical results using three biological data sets. The results cover a range of scenarios and include examples with explicitly interventional data.

2. Methods

2.1. Notation

Let $V = \{1, \ldots, p\}$ index a set of variables whose mutual causal relationships are of interest. Let G denote a directed graph with vertex set V and edge set E; where useful, we use $V(G), E(G)$ to denote its vertex and edge sets and $A(G)$ to denote the corresponding $p \times p$ binary adjacency matrix. To make the connection between causal relationships and machine learning more transparent, we introduce linear indexing by $[k]$ of the pairs $(i, j) \in V \times V$. Where needed, we make the correspondence explicit, denoting by $(i[k], j[k])$ the variable pair corresponding to linear index $[k]$ and by $[k(i, j)]$ the linear index for pair (i, j). Suppose A is the adjacency matrix of the unknown graph of interest. Let $y_{[k]} \in \{-1, +1\}$ be a binary variable (for convenience mapped onto $\{-1, +1\}$) corresponding to the entry $(i[k], j[k])$ in A; these $y_{[k]}$'s are the labels or outputs to be learned. Available data are denoted $\mathcal{D}$. Available *a priori* knowledge about causal relationships between the variables V is denoted Φ.

2.2. Causal Semantics

Given data $\mathcal{D}$ and background knowledge Φ we aim to construct an estimate $\hat{G}$, the latter being a directed graph that need not be acyclic. The information in Φ guides the learner. Two main cases arise, both of which we consider in experiments below:

- *Total or ancestral effects.* Here, Φ contains information on total effects—for example via interventional experiments as performed in biology—and the edges in the estimate $\hat{G}$ are intended to describe such effects. This means that an edge $(i, j) \in E(\hat{G})$ is interpreted to mean that node i is inferred to be a causal ancestor of node j.
- *Direct effects.* Here, Φ contains information on direct effects (relative to the variable set V) and the edges in the estimated graph $\hat{G}$ are intended to describe direct effects. Then, an edge $(i, j) \in E(\hat{G})$ is interpreted to mean that i is inferred to be a direct cause of j (relative to the variable set V).

Our immediate motivation comes from the experimental sciences and we focus in particular on causal influences that can, at least in principle, be experimentally verified (even in the presence of latent variables) and where causal cycles are possible (as is often the case in biology or economics, see e.g., Hyttinen et al., 2012). Accordingly, we do not demand acyclicity. In our empirical work in biology, the nature of the underlying chemical/physical systems means that there are many small magnitude causal effects that are essentially irrelevant in the scientific context and this is a characteristic of many problem settings in the natural and social sciences. This motivates a pragmatic approach assuming that estimated graphs are not very dense or fully connected nor necessarily transitive[1].

2.3. Semi-Supervised Causal Learning

With the notation above, the task is to learn the $y_{[k]}$'s using $\mathcal{D}$ and Φ. This is done using a semi-supervised estimator $\hat{y}_{[k]}(\mathcal{D}, \Phi)$ (we make the connection to semi-supervised learning explicit shortly). For now assume availability of such an estimator (we discuss one specific approach below). Then from the $\hat{y}_{[k]}$ we have an estimate of the graph of interest as $\hat{G}(\mathcal{D}, \Phi) = (V, E(\hat{G}(\mathcal{D}, \Phi)))$ (recall that the vertex set V is known) with the edge set specified via the semi-supervised learner as

$$(i, j) \in E(\hat{G}(\mathcal{D}, \Phi)) \iff \hat{y}_{[k(i,j)]}(\mathcal{D}, \Phi) = 1. \tag{1}$$

Background knowledge Φ could be based on relevant science or on available interventional data. For example, in a given scientific setting, certain cause-effect information may be known from previous work or theory. Alternatively, if some interventional data are available in the study at hand, this gives information on some causal relationships. Whatever the source of the information, assume that it is known that certain pairs (i, j) are either causal pairs (positive information) or not causal pairs (negative information). Using the notation above, this amounts to knowing, for some pairs $[k]$, the value of $y_{[k]}$. In semi-supervised learning terms, the pairs whose causal status is known correspond to the labelled objects and the remaining pairs are the unlabelled objects.

For each pair $[k]$, some of the data, or some transformation thereof will be used as predictors or inputs, denote these generically as $g_{[k]}(\mathcal{D})$. That is, $g_{[k]}$ is a featurization of the data, with the featurization specific to variables $(i[k], j[k])$. Let $\mathcal{K}$ be the set of linear indices (i.e., $[k] \in \mathcal{K}$ is a variable pair), $\mathcal{L} \subset \mathcal{K}$ be the variable pairs with labels available

1. We emphasize that these are pragmatic assumptions motivated by the nature of experimental data and scientific applications, and not intended to be fundamental statements about causality. For example, Hyttinen et al. (2012) make the point that cycles can be removed by considering time-varying data on a suitable time scale, but that nevertheless cycles are common in causal scientific models in economics, engineering and biology due to the fact that measurements are usually taken at wider intervals.

(via Φ) and $\mathcal{U} = \mathcal{K} \setminus \mathcal{L}$ be the set of unlabelled pairs. Let $\mathbf{y}^{\mathcal{L}}$ be a binary vector comprising the $m_{\mathcal{L}} = |\mathcal{L}|$ available labels and $\mathbf{y}^{\mathcal{U}}$ be an unknown binary vector of length $m_{\mathcal{U}} = |\mathcal{U}|$. The available labels are determined by the background information Φ and we can write $\mathbf{y}^{\mathcal{L}}(\Phi)$ to make this explicit. A semi-supervised learner gives estimates for the unlabelled objects, given the data and available labels. That is, an estimate of the form $\hat{\mathbf{y}}^{\mathcal{U}}(g(\mathcal{D}), \mathbf{y}^{\mathcal{L}}(\Phi))$. With these in hand we have estimates for all labels and therefore for all edges via (1).

Formulated in this way, it is clear that essentially any combination of featurization g and semi-supervised learner could be used in this setting. Below, as a practical example, we explore graph-based manifold learning (following Belkin et al., 2006) combined with a simple bivariate featurization.

2.4. A Bivariate Featurization

For distance-based learning, we require a distance measure between objects (here, variable pairs) $[k], [k'] \in \mathcal{K}$. The simplest candidate distance between variable pairs $[k], [k']$ is based only on the bivariate distribution for the variables comprising the pairs (we make this notion precise below). Proofs of propositions appearing in this Section are provided in Appendix A.

2.4.1. Distance between variable pairs

Let $\mathbf{Z}$ denote the p-dimensional random variable whose n realizations $\mathbf{z}^{(l)}$, $l = 1, \ldots, n$, comprise the data set $\mathcal{D}$. Assume $\mathbf{Z} \in \mathcal{Z}_p = [z_{\min}, z_{\max}]^p$ and that $\mathcal{Z}_p$ is endowed with the Borel σ-algebra $\mathcal{B}_p = \mathcal{B}(\mathcal{Z}_p)$. Let $\mathcal{P}$ be the set of all twice continuously differentiable probability density functions, generically denoted π, with respect to Lebesgue measure Λ_2 on $(\mathcal{Z}_2, \mathcal{B}_2)$. Let $\Pi_{[k]}$ be the bivariate (marginal) distribution for components $i[k], j[k] \in V$ of $\mathbf{Z}$.

Assumption 1 *Each $\Pi_{[k]}$ admits a density function $\pi_{[k]} \in \mathcal{P}$.*

If available, the densities $\pi_{[k]}, \pi_{[k']}$ could be used to define a distance between the pairs $[k], [k']$. Let $d_{\mathcal{P}} : \mathcal{P} \times \mathcal{P} \to [0, \infty)$ denote a pseudo-metric[2] on $\mathcal{P}$. Since we do not have access to the underlying probability density functions, we construct an analogue using the available data $\mathcal{D}$. Let $\mathcal{S}_n := [z_{\min}, z_{\max}]^{2n}$ denote the space of possible bivariate samples (the sample size is n) and $S_{[k]} \in \mathcal{S}_n$ denote the subset of the data for the variable pair $[k]$. That is, $S_{[k]} = \{(z^{(l)}_{i[k]}, z^{(l)}_{j[k]})\}_{l=1,\ldots,n} \subset \mathcal{Z}_2$.

Let $\kappa : \mathcal{S}_n \to \mathcal{P}$ be a density estimator (DE). We consider sample quantities of the form $d_{\mathcal{S}} = d_{\mathcal{P}} \circ (\kappa \times \kappa)$. That is, given data $S_{[k]}, S_{[k']} \in \mathcal{S}_n$ on two pairs $[k], [k']$, the DE is applied separately to produce density estimates $\kappa(S_{[k]})$ and $\kappa(S_{[k']})$, that are compared using $d_{\mathcal{P}}$ to give $d_{\mathcal{S}}(S_{[k]}, S_{[k']}) = d_{\mathcal{P}}(\kappa(S_{[k]}), \kappa(S_{[k']}))$. This construction ensures that $d_{\mathcal{S}}$ is a pseudo-metric without assumptions on the DE κ:

Proposition 1 *Assume that $d_{\mathcal{P}}$ is a pseudo-metric on $\mathcal{P}$. Then $d_{\mathcal{S}}$ is a pseudo-metric on $\mathcal{S}_n$. If, in addition, κ is injective and $d_{\mathcal{P}}$ is a metric on $\mathcal{P}$, then $d_{\mathcal{S}}$ is a metric on $\mathcal{S}_n$.*

2. Recall that a pseudo-metric d satisfies all of the properties of a metric with the exception that $d(x, y) = 0 \not\Rightarrow x = y$.

2.4.2. Choice of distance

For semi-supervised learning we need a notion of distance under which causal pairs are relatively 'close' to each other. For a measurable space $\mathcal{X}$ equipped with a measure ρ we let $\|f\|_{L^q(\rho)} := \left(\int_{\mathcal{X}} |f|^q \mathrm{d}\rho\right)^{\frac{1}{q}} < \infty$. The notion of distance that we consider is

$$d_{\mathcal{P}}(\pi, \tilde{\pi}) := \|\pi - \tilde{\pi}\|_{L^2(\Lambda_2)}.$$

The right hand side exists since the integrand is continuous on a compact set and thus bounded. This can be contrasted with the kernel embedding that was proposed for supervised causal learning in Lopez-Paz et al. (2015).

Proposition 2 *$d_{\mathcal{P}}$ is a metric on $\mathcal{P}$.*

The main requirement that we have of the DE is that it provides consistent estimation in the $\|\cdot\|_{L^2(\Lambda_2)}$ norm when $\pi \in \mathcal{P}$. Specifically, consider a sequence $S^{(n)}$ in $\mathcal{S}_n$ indexed by the number n of data points. In particular, suppose that $S^{(n)}$ is built from n independent data points whose distribution is Π (the shorthand notation $S^{(n)} \overset{\text{i.i.d.}}{\sim} \Pi$ will be used). Let π be the density function for Π. Then κ is said to be "consistent" if $\|\pi - \kappa(S^{(n)})\|_{L^2(\Lambda_2)} = o_P(1)$ holds for $S^{(n)} \overset{\text{i.i.d.}}{\sim} \Pi$ whenever $\pi \in \mathcal{P}$.

Proposition 3 *Suppose κ is consistent and that $\Pi, \tilde{\Pi}$ admit densities $\pi, \tilde{\pi} \in \mathcal{P}$. Then, for $S^{(n)} \overset{\text{i.i.d.}}{\sim} \Pi$, $\tilde{S}^{(n)} \overset{\text{i.i.d.}}{\sim} \tilde{\Pi}$, where $S^{(n)}$ and $\tilde{S}^{(n)}$ are not necessarily independent, we have that $d_{\mathcal{S}}(S^{(n)}, \tilde{S}^{(n)}) = d_{\mathcal{P}}(\pi, \tilde{\pi}) + o_P(1)$.*

Thus $d_{\mathcal{S}}$ approximates the idealized metric $d_{\mathcal{P}}$ in the limit of draws from Π and $\tilde{\Pi}$. Note that, in our intended use case, the $S^{(n)}$ and $\tilde{S}^{(n)}$ will correspond to bivariate scatter plots $S_{[k]}$, $S_{[k']}$ generated from the same underlying $\mathbf{z}^{(l)}$, $l = 1, \ldots, n$, and hence $S^{(n)}$ and $\tilde{S}^{(n)}$ will not be independent.

For the experiments in this paper, motivated by computational ease, we used a simple bivariate histogram as the DE κ. To this end, partition $\mathcal{Z}_2$ into an $M \times M$ regular grid whose (m_1, m_2)th element is denoted B_{m_1,m_2}. The standard bandwidth notation $h = M^{-1}$ will also be used. For a scatter plot $S \in \mathcal{S}_n$, let x_{m_1,m_2} denote the number of elements that belong to the set B_{m_1,m_2}. Then the histogram estimator is

$$\kappa(S)(\mathbf{z}') = \sum_{m_1,m_2=1}^{M} \frac{x_{m_1,m_2}}{n} \frac{1}{h^2} \mathbb{I}[\mathbf{z}' \in B_{m_1,m_2}], \qquad \mathbf{z}' \in \mathcal{Z}_2. \tag{2}$$

This DE is consistent in the sense of Proposition 3. Indeed:

Proposition 4 *Let the bandwidth parameter h of the histogram estimator κ be chosen such that $nh^2 \to \infty$. Then κ is consistent. Moreover, an optimal choice of $h \asymp n^{-1/4}$ leads to $\|\pi - \kappa(S^{(n)})\|_{L^2(\Lambda_2)} = O_P(n^{-1/4})$ whenever $S^{(n)} \overset{\text{i.i.d.}}{\sim} \Pi$ and $\pi \in \mathcal{P}$.*

We note that this histogram DE is not rate optimal for the class $\mathcal{P}$ (for comparison, kernel DEs attain a rate of $O_P(n^{-2/3})$ over the same class $\mathcal{P}$ of twice continuously differentiable bivariate densities considered here, see Wand and Jones, 1994). However, an important advantage of the histogram DE is that the subsequent evaluation of $\kappa(S)$ is $O(1)$, compared with $O(n)$ for the kernel DE.

2.4.3. Implementation of the DE

The above arguments support the use of a bivariate histogram to provide a simple featurization for variable pairs. In practice, for all examples below, the data were standardized, then truncated to $[-3,3]^2$, following which a bivariate histogram with bins of fixed width 0.2 was used. The dimension of the resulting feature matrix was then reduced (to 100) using PCA.

2.5. Manifold Regularization

Recall that the goal is to estimate binary labels $\mathbf{y}^{\mathcal{U}}$ for a subset $\mathcal{U} \subset \mathcal{K}$ of variable pairs given available data $\mathcal{D}$ and known labels $\mathbf{y}^{\mathcal{L}}(\Phi)$ for a subset $\mathcal{L} = \mathcal{K} \setminus \mathcal{U}$ (these are taken to be obtained from available interventional experiments and/or background knowledge). For any two pairs $[k],[k'] \in \mathcal{K}$, we also have available a distance $d_{\mathcal{S}}(S_{[k]}, S_{[k']})$. This is a task in semi-supervised learning (see e.g., Belkin et al., 2006; Fergus et al., 2009) and a number of formulations and methods could be used for estimation in this setting. Here we describe a specific approach in detail, using manifold regularization methods discussed in Belkin et al. (2006).

Let $\mathbf{x}_{[k]}$ denote a vector whose entries are the bin-counts $x_{i,j}$, $1 \le i,j \le M$, appearing in (2), for scatter plot $S_{[k]}$. Let $\mathcal{X} = \bigtimes_{1\le i,j\le M}[0,n]$ and note that $\mathbf{x}_{[k]} \in \mathcal{X}$. Then we make the observation that, for the histogram estimator,

$$d_{\mathcal{S}}(S_{[k]}, S_{[k']}) \propto \|\mathbf{x}_{[k]} - \mathbf{x}_{[k']}\|_2 \,.$$

This perspective emphasizes that $g_{[k]}(\mathcal{D}) = \mathbf{x}_{[k]}$ is the featurization that underpins this work, and that the classification task can be considered as the construction of a map $c : \mathcal{X} \to \{-1,+1\}$. To develop an approach to semi-supervised classification in the manner of Belkin et al. (2006), let $\rho_{\mathcal{X}}$ be a reference measure on $\mathcal{X}$ and let $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ be a Mercer kernel; i.e., continuous, symmetric and positive semi-definite. The reproducing kernel Hilbert space, $\mathcal{H}_K$, associated to K can be defined via the integral operator $\Sigma_K : L^2(\rho_{\mathcal{X}}) \to L^2(\rho_{\mathcal{X}})$ where

$$\Sigma_K f(\mathbf{x}) = \int K(\mathbf{x},\tilde{\mathbf{x}}) f(\tilde{\mathbf{x}}) \mathrm{d}\rho_X(\tilde{\mathbf{x}}).$$

From the fact that K is a Mercer kernel it follows that Σ_K is self-adjoint, positive semi-definite and compact. In particular, Σ_K^{α} is well-defined for $\alpha \in (0,\infty)$. The reproducing kernel Hilbert space is defined as $\mathcal{H}_K = \Sigma_K^{\frac{1}{2}} L^2(\rho_{\mathcal{X}})$ and its norm is $\|f\|_{\mathcal{H}_K} := \|\Sigma_K^{-\frac{1}{2}} f\|_{L^2(\rho_{\mathcal{X}})}$; c.f. Corollary 4.13 in Cucker and Zhou (2007).

Recall that $m_{\mathcal{L}} = |\mathcal{L}|$ is the number of available labels and $m_{\mathcal{U}} = |\mathcal{U}|$ the number of unlabelled pairs. Let $m = m_{\mathcal{U}} + m_{\mathcal{L}}$ $(= |\mathcal{K}|)$ be the total number of pairs. Using the distance function $d_{\mathcal{S}}$ we first define an $m \times m$ similarity matrix $\mathbf{W}$ with entries

$$W_{[k],[k']} = \exp\left(-\frac{1}{2\sigma_1^2}\|\mathbf{x}_{[k]} - \mathbf{x}_{[k']}\|_2^2\right) \tag{3}$$

where $\sigma_1 > 0$ must be specified. The squared-exponential form is motivated by an analytic connection between the heat kernel and the Laplace-Beltrami operator, which will be

exploited in Section 2.5.1. We will use a partition of the matrix corresponding to the sets $\mathcal{U}, \mathcal{L}$ as follows

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}^{\mathcal{LL}} & \mathbf{W}^{\mathcal{LU}} \\ \mathbf{W}^{\mathcal{UL}} & \mathbf{W}^{\mathcal{UU}} \end{bmatrix}$$

where we have assumed, without loss of generality, that the variable pairs are ordered so that the labelled pairs appear in the first $m_{\mathcal{L}}$ places, followed by the $m_{\mathcal{U}} = m - m_{\mathcal{L}}$ unlabelled pairs. Correspondingly let

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}^{\mathcal{L}} \\ \mathbf{y}^{\mathcal{U}} \end{bmatrix} \in \{-1,+1\}^m$$

denote a label matrix, where $+1$ indicate those pairs $[k]$ for which $y_{[k]} = 1$. The vector $\mathbf{y}^{\mathcal{U}}$ is unknown and is the object of estimation.

Let $\mathbf{D}$ be the $m \times m$ diagonal matrix with diagonal entries $D_{[k],[k]} = \sum_{[k']\in\mathcal{K}} W_{[k],[k']}$. Define $\mathbf{L} = \mathbf{D} - \mathbf{W}$ (i.e., the un-normalized graph Laplacian; all matrices with $O(m^2)$ entries are denoted as bold capitals to emphasize the potential bottleneck that is associated with storage and manipulation of these matrices). Let

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}^{\mathcal{L}} \\ \mathbf{f}^{\mathcal{U}} \end{bmatrix} \in \mathbb{R}^m$$

be a vector corresponding to a classification function $f : \mathcal{X} \to \mathbb{R}$ evaluated at the m variable pairs $\mathcal{K}$, with the superscripts indicating correspondence with the labelled and unlabelled pairs. Intuitively, we want the sign of $\mathbf{f}$ to agree with the known labels $\mathbf{y}^{\mathcal{L}}$ and also to take account of the manifold structure encoded in $\mathbf{L}$.

In this work we consider a classifier of the form $\hat{c}(\mathbf{x}) = \mathrm{sign}(\hat{f}(\mathbf{x}))$ where $\hat{f}$ arises from the *Laplacian-regularized least squares* method

$$\hat{f} = \underset{f\in\mathcal{H}_K}{\arg\inf} \frac{\|\mathbf{y}^{\mathcal{L}} - \mathbf{f}^{\mathcal{L}}\|_2^2}{m_L} + \lambda_1 \frac{\mathbf{f}^\top \mathbf{L}\mathbf{f}}{m} + \lambda_2 \|f\|_{\mathcal{H}_K}^2 \,, \tag{4}$$

following Section 4.2 of Belkin et al. (2006). Here the first term relates the known labels to the values of the function f. The second term imposes 'smoothness' on the label assignment in the sense of encouraging solutions where the labels do not change quickly with respect to the distance metric. The third term is principally to ensure that the infimum remains well-defined and unique in the situation where there is insufficient data for the first penalty alone to be sufficient (see Remark 2 in Belkin et al., 2006).

Remark 5 (Choice of loss) *It is important to comment on our choice of a squared-error loss function in* (4)*, which differs from the more natural approach of using hinge loss for a binary classification task. Our motivation here is principally computational expedience; the computational burden associated with the $m = O(p^2)$ different scatter plots requires that a light-weight estimation procedure is used. However, we note that we are not the first to propose the use of squared-error loss in the classification context; it is in fact a standard approach to classification in the situation when the number of classes is > 2 (e.g., Wang et al., 2008).*

2.5.1. Consistency of the Classifier

As explained in Remark 5, the use of a squared-error loss function in a classification context is somewhat unnatural. It is therefore incumbent on us to establish consistency of the proposed method.

To this end, we exploit the specific form of the similarity matrix used in (3). Indeed, if we re-write

$$\frac{\mathbf{f}^\top \mathbf{L}\mathbf{f}}{m} = \frac{1}{2m} \sum_{[k],[k']\in\mathcal{K}} (f(\mathbf{x}_{[k]}) - f(\mathbf{x}_{[k']}))^2 W_{[k],[k']} \tag{5}$$

then it can be established (under certain regularity conditions) that, if input data $\mathbf{x}$ are independently drawn from $\rho_{\mathcal{X}}$, then (5) converges to the quantity $\int f(\mathbf{x})\Delta_{\mathcal{M}} f(\mathbf{x}) \mathrm{d}\rho_{\mathcal{X}}^2(\mathbf{x})$ (up to proportionality), a smoothness penalty based on weighted Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ on the manifold $\mathcal{M}$ induced by $\rho_{\mathcal{X}}$ (Grigor'yan, 2006). The convergence occurs as $m, \sigma_1^2 m^{d+2} \to \infty$ (Theorem 3.1 of Belkin and Niyogi, 2008).

This convergence of the graph Laplacian to the Laplace-Beltrami operator underlies existing consistency results for semi-supervised regression (e.g., Cao and Chen, 2012) and is exploited again to establish the consistency of our classifier $\hat{c}(\mathbf{x}) = \mathrm{sign}(\hat{f}(\mathbf{x}))$ in Appendix B. In summary, the ability to assign the correct label to an unlabelled pair $[k] \in \mathcal{L}$ depends on both the intrinsic predictability of the label as a function of the scatter plot $S_{[k]}$, as quantified by the Bayes risk, and the smoothness of the Bayes classifier f_ρ as quantified by the largest value $\alpha \in (0, 1]$ such that $\Sigma_K^{-\frac{\alpha}{2}} f_\rho \in L^2(\rho_{\mathcal{X}})$; see Corollary 9 in Appendix B for full detail.

2.5.2. Implementation of the Classifier

Given training labels $\mathbf{y}^{\mathcal{L}}$, label estimates $\hat{\mathbf{y}}^{\mathcal{U}} = \mathrm{sign}(\hat{\mathbf{f}}^{\mathcal{U}})$ are obtained by minimizing the objective function described above, as explained in Equation 8 in Belkin et al. (2006). This gives

$$\hat{\mathbf{f}}^{\mathcal{U}} = \mathbf{K}_{\mathcal{U},\mathcal{K}} \left(\begin{bmatrix} \mathbf{I}_{m_{\mathcal{L}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{K}_{\mathcal{K},\mathcal{K}} + \lambda_2 m_{\mathcal{L}} \mathbf{I}_m + \frac{\lambda_1 m_{\mathcal{L}}}{m^2} \mathbf{L}\mathbf{K}_{\mathcal{K},\mathcal{K}} \right)^{-1} \begin{bmatrix} \mathbf{y}^{\mathcal{L}} \\ \mathbf{0} \end{bmatrix} \tag{6}$$

where $\mathbf{K}_{\mathcal{U},\mathcal{K}}$ is the $m_{\mathcal{U}} \times m$ kernel matrix based on the unlabeled $\mathcal{U}$ and total $\mathcal{K}$ data, $\mathbf{K}_{\mathcal{K},\mathcal{K}}$ is the $m \times m$ kernel matrix based on the total data $\mathcal{K}$ and $\mathbf{I}_m$ denotes an m-dimensional identity matrix.

Here $\hat{\mathbf{y}}^U$ provides a point estimate for the unknown labels while $\hat{\mathbf{f}}^{\mathcal{U}}$ is real-valued and can be used to rank candidate pairs if required. The linear system in (6) can be solved at a naive computational cost of $O(m^3)$. Computation for large-scale semi-supervised learning has been studied in the literature (see e.g., Fergus et al., 2009) and a number of approaches could be used to scale up to larger problems, but were not pursued in this work.

For experiments reported below we employed a similarity matrix (with length scale σ_1 as in (3)) and a kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2\sigma_2^2} \|\mathbf{x} - \mathbf{x}'\|_2^2 \right)$$

whose length-scale parameter σ_2 was set equal to σ_1 in the absence of prior knowledge about the manifold $\mathcal{M}$. The scale σ_1 was set to the average distance to the nearest 50 points in the feature space (in practice estimated via a subsample).

The two penalty parameters in (4) were set to small positive values ($\lambda_1 = \lambda_2 = 0.001$; we found results were broadly insensitive to this choice). Following common practice we worked with the normalized graph Laplacian $\tilde{\mathbf{L}} := \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$ in place of $\mathbf{L}$ (see Remark 3 of Belkin et al., 2006).

3. Empirical Results

We tested our approach using three data sets with different characteristics. The key features of each data set are outlined below, with a full description of each data set appearing in the respective subsection. In all cases performance was assessed using either held-out interventional data or scientific knowledge.

- **D1: Yeast knockout data.** Here, we used a data set due to Kemmeren et al. (2014), previously considered for causal learning in Peters et al. (2016); Meinshausen et al. (2016). The data consist of a large number of gene deletion experiments with corresponding gene expression measurements.
- **D2: Kinase intervention data from human cancer cell lines.** These data, due to Hill et al. (2017), involve a small number of interventions on human cells, with corresponding protein measurements over time.
- **D3: Protein data from cancer patient samples.** These data arise from The Cancer Genome Atlas (TCGA) and are presented in Akbani et al. (2014). There are no interventional data, but the data pertain to relatively well-understood biological processes allowing inferences to be checked against causal scientific knowledge.

An appealing feature of MRCL is the simplicity with which it can be applied to diverse problems. In each case below, we simply concatenate available data to form the data set $\mathcal{D}$ and available knowledge/interventions to form Φ, then directly apply the methods as described.

3.1. General Problem Set-Up

The basic idea in all three problems was as follows: given data on a set of variables, for each (ordered) pair (i, j) of variables we sought to determine whether or not i has a causal effect on j. In the case of data sets **D1** and **D2** the results were assessed against the outcome of experiments involving explicit interventions. As discussed above, such experiments reveal ancestral relationships (that need not be direct) and the goal in these examples was to learn such relationships. The availability of a large number of interventions in **D1** allowed a wider range of experiments, whereas **D2** is a smaller data set (but from human cells), allowing only a relatively limited assessment. In the case of **D3**, where interventional data (i.e., interventions on the same biological material that give rise to the training data) were not available but the relevant biological mechanisms are relatively well understood, we compared results to a reference mechanistic graph derived from the domain literature. The

literature itself is in effect an encoding of extensive interventional experiments combined with biochemical and biophysical knowledge. This gives information on direct edges and here the edges learned are intended to represent direct causes (relative to the set of observed variables). Within the semi-supervised set-up, a subset of pairs were labelled at the outset and the remaining pairs were unlabelled. All empirical results below are for unlabelled pairs; that is, in all cases assessment is carried out with respect to causal (and non-causal) relationships that were not used to train the models.

3.2. Data Set D1: Yeast Gene Expression

Data. The data consisted of gene expression levels (log ratios) for a total of $p_{\text{total}} = 6170$ genes. Some of the data samples were measurements after knocking out a specific gene (interventional data) and the other samples were without any such intervention (observational data), with sample sizes of $n^{\text{int}} = 1479$ and $n^{\text{obs}} = 153$ respectively. Each of the genes intervened on was one of the p_{total} genes. Let $t(l)$ be the index of the gene targeted by the lth intervention. That is, the lth interventional sample was an experiment in which gene $t(l)$ was knocked out. Let $T = \{t(1), \ldots, t(n^{\text{int}})\}$ be the subset of genes that were the target of an interventional experiment.

Problem set-up. Our problem set-up was as follows. We sampled a subset $C \subset T$ of the genes that were intervened upon, with $|C| = 50$, and treated this as the vertex set of interest (i.e., setting $V = C$ and $p = |C| = 50$). The goal was to uncover causal relationships between these p variables.

Since by design interventional data were available for all variables $j \in C$, we used these data to define an interventional 'gold standard'. To this end we used a robust z-score that considered the change in a variable of interest under intervention, relative to its observational variation. Let Z_{ij}^{int} denote the expression level of gene j following intervention on gene i. For any pair of genes $i, j \in C$ we say that gene i has a causal effect on gene j if and only if $\zeta_{ij} = |Z_{ij}^{\text{int}} - M_j^{\text{obs}}|/\text{IQR}_j^{\text{obs}} > \tau$, where M_j^{obs} is the median level of gene j (calculated using half of the observational data samples; the remaining samples were used as training data—see below), $\text{IQR}_j^{\text{obs}}$ the corresponding inter-quartile range and $\tau = 5$ was a fixed threshold. That is, we say there is an (experimentally verified) causal relationship between gene i and gene j if and only if $\zeta_{ij} > \tau$. An absence of causal effects precludes estimation of true positive rates; hence we sampled C subject to a sparsity condition (that at least 2.5% of gene pairs show an effect).

Let $A(C)$ be a $p \times p$ binary matrix encoding the causal effects as described in the foregoing (i.e., $A(C)_{ij} = 1$ indicates that i has an experimentally verified causal effect on j). Then, given data on genes C, we set up the learning problem as follows. We treated a fraction ρ of the entries in $A(C)$ as the available labels Φ. Thus, here $m = p^2 = 2500$, $m_{\mathcal{L}} = \lfloor \rho m \rfloor$ and $m_{\mathcal{U}} = m - m_{\mathcal{L}}$. Using these labels and data on the variables C, we learned causal edges as described. This gave estimates for the remaining (unseen) entries in $A(C)$, which we compared against the corresponding true values. The data set $\mathcal{D}$ comprised expression measurements for the genes in C for $n_{\text{train}}^{\text{obs}} = 76$ observational data samples (those samples not used to calculate the robust z-scores), plus $n_{\text{train}}^{\text{int}}$ interventional data samples where genes outside the set of interest were intervened upon; that is, a subset of the 1429 genes in $T \backslash C$. This set-up ensured that $\mathcal{D}$ include neither any of the interventional nor observational

data that was used to obtain the ground-truth matrix $A(C)$. The total amount of training data is denoted by $n_{\text{train}} = n_{\text{train}}^{\text{obs}} + n_{\text{train}}^{\text{int}}$. We considered $n_{\text{train}} = 200, 500$ and 1000 (corresponding to $n_{\text{train}}^{\text{int}} = 124, 424$ and 924 respectively, sampled at random).

Results. We compared the proposed Manifold Regularized Causal Learning (MRCL) approach with the following approaches:

- Penalized regression with an ℓ_1 penalty (**Lasso**; Tibshirani, 1996). Each variable $j \in C$ was regressed on all other variables $i \in C, i \neq j$ to obtain regression coefficients. This is not a causal approach as such, but is included as a simple multivariate baseline.
- Intervention-calculus when the DAG is absent (**IDA**; Maathuis et al., 2009, 2010). A lower bound for the total causal effect of variable i on variable j was estimated for each pair $i, j \in C, i \neq j$.
- The PC algorithm (**PC**; Spirtes et al., 2000). This provides a CPDAG estimate for the variables C.
- GIES (**GIES**; Hauser and Bühlmann, 2012). This provides an essential graph estimate for the variables C, and allows inclusion of interventional data in a principled manner.

As simple baselines, we also included Pearson and Kendall correlation coefficients (**Pearson** and **Kendall**) and, following a suggestion from a referee, a simple k-nearest neighbor approach based on the featurization introduced above (**k-NN**).

We note that the causal methods compared against here differ in various ways from MRCL in the nature of their inputs and outputs and should not be regarded as direct competitors. Rather, the aim of the experiments is to investigate how MRCL performs on real data, whilst providing a set of baselines corresponding to well-known causal tools and standard correlation measures.

For the methods resulting in a score s_{ij} for all pairs $i, j \in C, i \neq j$ (i.e., correlation or regression coefficients, total causal effects, or, for MRCL, the real-valued $\hat{\mathbf{f}}$ in (6)), the scores were thresholded and pairs (i, j) whose absolute values of the score fell above the threshold were labelled as 'causal'. Varying the threshold and calculating true positives and false positives with respect to the binary *unseen* entries in the matrix $A(C)$ resulted in a receiver operating characteristic (ROC) curve.

Figure 1 shows the area under the ROC curve (AUC) as a function of the proportion ρ of entries in $A(C)$ that were observed, for the three sample sizes. Results were averaged over 25 iterations. MRCL showed good performance relative to the other approaches for all 12 considered combinations of n_{train} and ρ (for the other methods shown in Figure 1, any variation in performance with ρ was solely due to the changing test set as these methods do not use the background knowledge Φ). Results for PC, which provides a point estimate of a graphical object, are shown as points on the ROC plane for the 12 different regimes in Appendix C (Fig. 6). We considered also the transitive closure (motivated by the nature of the experimental data) and exploiting the background information Φ via additional constraints. MRCL performs well relative to the other methods in all regimes (see also the Discussion).

In the above results the pairs whose causal relationship was to be predicted were chosen at random (i.e., the set of unlabelled pairs was a random subset of the set of all pairs). In

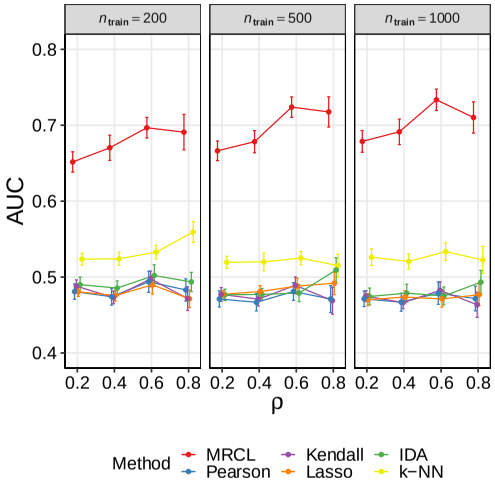


Figure 1: Results for data set D1 (yeast data), random sampling. Area under the ROC curve (AUC; with respect to causal relationships determined from unseen interventional data), as a function of the fraction ρ of labels available (labels were sampled at random). Results are shown for three training data sample sizes n_{train}. Results are mean values over 25 iterations and error bars indicate standard error of the mean. Additional results for the PC algorithm appear in Appendix C (see text for details).

contrast, in some settings it may be relevant to predict the effect of intervening on variable i, without knowing the effect of intervening on i on *any* other variable. For this setting, the unlabelled set should comprise entire rows of the causal adjacency matrix $A(C)$. Figure 2 considers this case. To ensure a sufficient number of rows were non-empty, we imposed the additional restriction on the gene subset C that at least half of the rows had at least one causal effect. Results for PC are shown in Appendix C (Fig. 7) as points on the ROC plane. As for the random sampling case above, MRCL offers an improvement over the other methods. k-NN also performs well relative to the other approaches here.

We additionally compared MRCL with GIES. GIES and MRCL differ in terms of their required inputs: In addition to data $\mathcal{D}$, MRCL requires binary labels on causal relationships via background information Φ, while GIES requires the interventional data itself and metadata specifying the intervention targets. For row-wise sampling, to allow for a reasonable comparison, we ran GIES providing the interventional data corresponding to the rows whose labels are provided to MRCL. The same data was also provided as input to the other approaches, including in data set $\mathcal{D}$ for MRCL. This means the data matrices differ

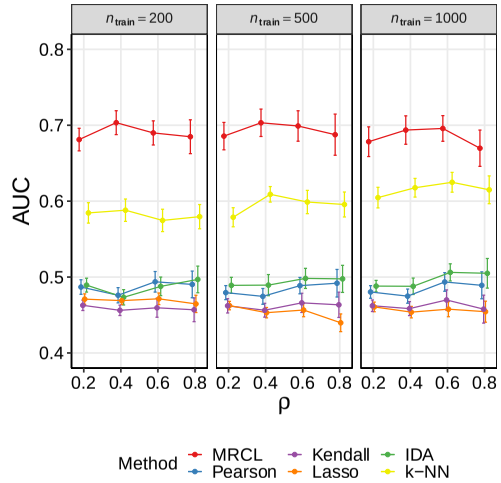


Figure 2: Results for data set D1 (yeast data), row-wise sampling. As Figure 1, except the subset of labels available to the learner were obtained by sampling entire rows of the causal adjacency matrix. As before, a proportion ρ were sampled. The remaining rows were then used as test data. Additional results for the PC algorithm appear in Appendix C (see text for details).

from those above, with sample size dependent on ρ, and for MRCL, $\mathcal{D}$ now includes data that was used to obtain background information Φ (train/test validity is preserved since it remains the case that all testing is done with respect to entirely unseen interventions). Results appear in Figure 3, with PC and GIES shown as a points on the ROC plane. MRCL appears to offer an improvement relative to the other methods (see also the Discussion). Note that GIES is not directly applicable to the random sampling setting above since it requires the interventional data with respect to all other variables (and not just a subset thereof).

3.3. Data Set D2: Protein Time-Course Data

Data. The data consisted of protein measurements for $p = 35$ proteins measured at seven time points in four different 'cell lines' (BT20, BT549, MCF7 and UACC812; these are laboratory models of human cancer) and under eight growth conditions. The proteins under study act as kinases (i.e., catalysts for a biochemical process known as phosphorylation) and interventions were carried out using kinase inhibitors that block the kinase activity of specific proteins. A total of four intervention regimes were considered, plus a control regime with no interventions. The data used here were a subset of the complete data set reported

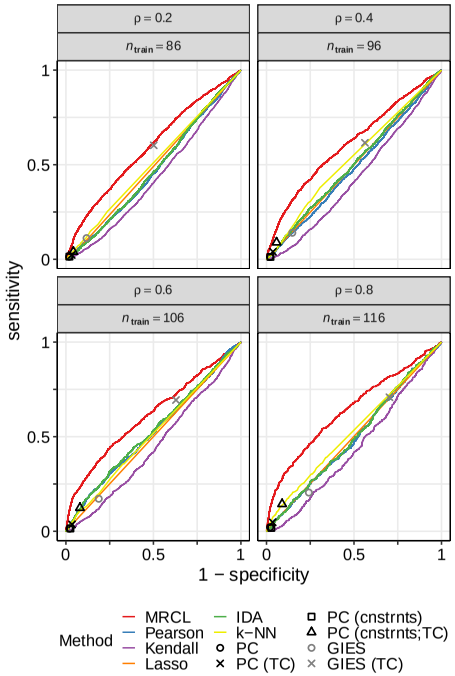


Figure 3: Results for data set D1 (yeast data), comparison including GIES, row-wise sampling. ROC curves are shown with respect to causal relationships determined from unseen interventional data. "TC" indicates use of a transitive closure operation and "cnstrnts" indicates that the background information Φ was included via input constraints. Results for PC and GIES are shown as points on the ROC plane. Note that due to the nature of input required by GIES the data matrices in this example differ from the row-wise sampling example in Figure 2 (see text for details). Results are averages over 25 iterations.

in detail in Hill et al. (2017) and were also previously used in a Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenge on learning causal networks (Hill et al., 2016).

Problem set-up. Treating each cell line as a separate, independent problem, the intervention regimes were used to define an interventional 'gold standard', in a similar vein as for data set **D1**. This followed the procedure described in detail in Hill et al. (2016) with an additional step of taking a majority vote across growth conditions to give a causal gold

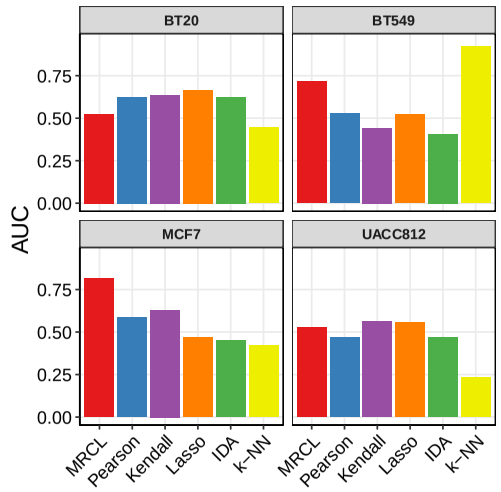


Figure 4: Results for data set D2 (protein time course data). Each panel is a different cell line, with its own training and (interventional) test data. AUC is with respect to an entirely held-out intervention. See text for details.

standard for each cell line c. For each cell line c, we formed a data matrix $\mathbf{Z}_c$ consisting of all available data for the $p = 35$ proteins except for one of the intervention regimes. The intervention regime not included was a kinase inhibitor targeting the protein mTOR. This intervention was entirely held out and used to provide the test labels. As background knowledge Φ_c we took as training labels causal effects under the other interventions. With this set-up, the task was to determine the (ancestral) causal effects of the entirely unseen intervention. Note that each cell line c was treated as an entirely different data set and task, with its own data matrix, background knowledge and interventional test data.

Results. Figure 4 shows AUCs (with respect to changes seen under the test intervention) for each of the four cell lines and each of the methods. There was no single method that outperformed all others across all four cell lines. MRCL performed particularly well relative to the other methods for cell lines BT549 and MCF7 (k-NN also performed well for BT549), was competitive for cell line UACC812, but performed less well for cell line BT20. We note also that, for cell lines BT549 and MCF7, the performance of MRCL was competitive with the best performers in the DREAM challenge and with an analysis reported in Hill et al. (2017). The latter involved a Bayesian model specifically designed for such data. In contrast, MRCL was applied directly to a data matrix comprising all training samples simply collected together.

3.4. Data Set D3: Human Cancer Data

Data. The data consisted of protein measurements for $p = 35$ proteins measured in $n = 820$ human breast cancer samples (from biopsies). The data originate from The Cancer Genome Atlas (TCGA) Project, are described in Akbani et al. (2014) and were retrieved from The Cancer Proteome Atlas (TCPA) data portal (Li et al., 2013, `https://tcpaportal.org`; data release version 4.0; Pan-Can 19 Level 4 data). Data for many cancer types are available, but here we focus on a single type (breast cancer) to minimize the potential for confounding by cancer type. It is at present difficult to carry out interventions in biopsy samples of this kind. However, we focused on the same 35 proteins as in data set **D2**, whose mutual causal relationships are relatively well-understood, and used a reference causal graph for these proteins based on the biochemical literature (as reported in Hill et al., 2017).

Problem set-up. We formed a data set $\mathcal{D}$ consisting of measurements for the $p = 35$ proteins for three different sample sizes: (i) $n_{\text{train}} = 200$, (ii) $n_{\text{train}} = 500$ or (iii) all $n_{\text{train}} = 820$ patient samples. For (i) and (ii) patient samples were selected at random. We then used a random fraction ρ of the reference graph as background knowledge, testing output on the (unseen) remainder.

Results. Figure 5 shows AUCs (with respect to the held-out causal labels) as a function of the proportion ρ of causal labels that were observed, for each of the methods and for the three sample sizes. Results were averaged over 25 iterations. MRCL performed well relative to the other methods, with performance improving with ρ. Results were qualitatively similar for the three sample sizes, with increases in AUC for $n_{\text{train}} = 820$ and $n_{\text{train}} = 500$ relative to $n_{\text{train}} = 250$. Results for PC are shown in Appendix C (Fig. 8) as points on the ROC plane.

4. Discussion

In this paper, we showed how a key aspect of causal structure learning can be framed as a machine learning task. Although many available approaches, including those based on DAGs and related graphical models, offer a well-studied framework, we think it may be fruitful to revisit some questions in causality using machine learning tools.

In our experiments, based on three real data sets, we found that MRCL performed well relative to a range of graphical model-based approaches. However, two points should be noted regarding these comparative results. First, the various methods differ with respect to their required inputs and the nature of their outputs. This means that in some cases specific methods may not be an ideal fit to the context of the specific data/task (as detailed when presenting the empirical results above). Second, the biological systems underlying these data sets are likely to have features (such as causal insufficiency and cycles) that violate one or more of the assumptions of some of these methods. That said, we think biological data sets of the kind we focused on here offer perhaps the best opportunity at present to empirically study causal learning methods and that causal learning tasks of the kind addressed here are highly relevant in many applications, in biology and beyond. Hence, we think that pursuing empirical work on such data is valuable both from methodological and applied points of view. As more interventional data become available in the future, it will

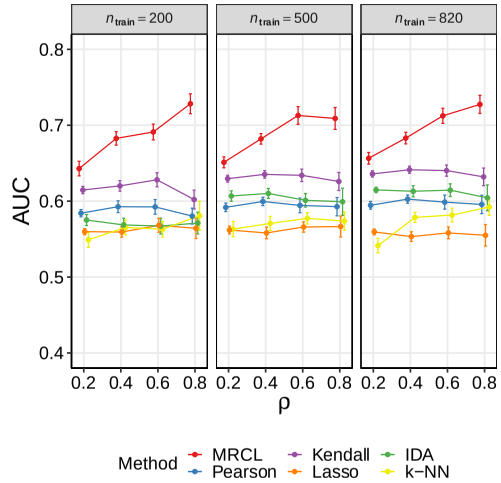


Figure 5: Results for data set D3 (human cancer data). Data are protein measurements from breast cancer patient samples from The Cancer Genome Atlas (TCGA). AUC is with respect to a reference graph based on the (causal) biochemical literature. Results are mean values over 25 iterations and error bars indicate standard error of the mean. See text for details. Additional results appear in Appendix C.

be important to carry out similar analyses in other contexts, in order to better understand the extent to which our findings generalize to other scientific settings.

An open question from a theoretical point of view is to understand conditions on data-generating processes needed to permit a discriminative approach as pursued here and we think this will be an interesting direction for future work. One point of view—analogous to that used in practical applications of classification—is to estimate the risk of the learner and thereby report an estimate of (causal) efficacy without having to directly consider requirements on the underlying system. We think this approach is acceptable when some causal information is available, since one can then empirically test problem-specific efficacy (as in our examples above). This then gives confidence with respect to generalization to new interventions on the system of interest (but does not address the broader theoretical question).

In our approach, information on multiple variable pairs is coupled via the classifier but not by global constraints on the graph. In the scientific settings we focused on we did not consider further coupling via global constraints but such constraints (e.g. enforcing transitivity) could be relevant in some applications and an interesting direction for further work. The main advantage of our approach is that it allows regularities in the data to emerge via learning, rather than having to be encoded via an explicit causal or mechanistic model. It also naturally provides some uncertainty quantification, in the sense of scores

that can be used to guide decisions or future experimental work. The main disadvantage relative to methods rooted in DAGs and related graphical models is the lack of a full causal model. Albeit under relatively strong assumptions, DAG-based models, once estimated, can be used to shed light on a huge range of questions concerning causal relationships, including direct and ancestral effects, and details of post-intervention distributions. In contrast, our approach in itself provides only estimates of binary causal relationships. That said, given the efficacy and simplicity of our approach, we think it would be fruitful to consider coupling it to established causal tools in a two-step approach, with our methods used to learn an edge structure in a data-driven manner and this structure used to inform a full analysis in a second step. Such an approach would require some care to avoid bias, and sample splitting techniques that have been studied in high-dimensional statistics could be relevant (Wasserman and Roeder, 2009; Städler and Mukherjee, 2017).

5. Code Availability

All computational analysis was performed in R (R Core Team, 2018). Source code for MRCL and scripts to generate the empirical results presented in Section 3 are available at `https://github.com/Steven-M-Hill/MRCL`.

Acknowledgments

The authors are grateful to the Editor and Reviewers for their constructive feedback on an earlier version of the manuscript. The authors are grateful to Umberto Noè for input on the empirical work and to Oliver Crook for feedback on an earlier version of the manuscript. This work was supported by the UK Medical Research Council (University Unit Programme number MC_UU_00002/2). CJO was supported by the ARC Centre of Excellence for Mathematics and Statistics, Australia, and the Lloyd's Register Foundation programme on data-centric engineering at the Alan Turing Institute, UK. The results shown here are in part based upon data generated by the TCGA Research Network: `https://www.cancer.gov/tcga`.

Author contributions: CJO, DB and SM developed methodology based on original ideas by SM. SMH performed the computational work and CJO contributed theory. CJO and SM wrote the paper with input from SMH and DB.

Appendix A. Proof of Results in the Main Text

In this appendix we provide proofs for the theoretical results in the main text.

Proof [Proposition 1] Given $S, S' \in \mathcal{S}_n$, we obtain from the pseudo-metric properties of $d_{\mathcal{P}}$ each of (i) positivity $d_{\mathcal{S}}(S,S') = d_{\mathcal{P}}(\kappa(S),\kappa(S')) \geq 0$, (ii) pseudo-identity $d_{\mathcal{S}}(S,S) = d_{\mathcal{P}}(\kappa(S),\kappa(S)) = 0$, (iii) symmetry $d_{\mathcal{S}}(S,S') = d_{\mathcal{P}}(\kappa(S),\kappa(S')) = d_{\mathcal{P}}(\kappa(S'),\kappa(S)) = d_{\mathcal{S}}(S',S)$ and (iv) the triangle inequality $d_{\mathcal{S}}(S,S'') = d_{\mathcal{P}}(\kappa(S),\kappa(S'')) \leq d_{\mathcal{P}}(\kappa(S),\kappa(S')) + d_{\mathcal{P}}(\kappa(S'),\kappa(S'')) = d_{\mathcal{S}}(S,S') + d_{\mathcal{S}}(S',S'')$, for all $S, S', S'' \in \mathcal{S}_n$.

Suppose now that κ is injective and $d_{\mathcal{P}}$ is a metric on $\mathcal{P}$. Then if it holds that $d_{\mathcal{S}}(S,S') = d_{\mathcal{P}}(\kappa(S),\kappa(S')) = 0$, it follows that $\kappa(S) = \kappa(S')$ which (from assumption on κ) implies $S = S'$ in $\mathcal{S}_n$. Thus under these additional assumptions, $d_{\mathcal{S}}$ is a metric on $\mathcal{S}_n$. ■

Proof [Proposition 2] The non-negativity, symmetry and sub-additivity properties of $d_{\mathcal{P}}$ are clear, so all that remains is to establish that $d_{\mathcal{P}}(\pi,\pi') = 0$ implies $\pi = \pi'$. From the definition of $\mathcal{P}$, both π and π' are continuous on $\mathcal{Z}_2$. The result is then immediate from the fact that, since π and π' are continuous and $\mathcal{Z}_2$ is compact, then $\int_{\mathcal{Z}_2} |\pi(\mathbf{z}') - \pi'(\mathbf{z}')|^2 \, \mathrm{d}\Lambda_2(\mathbf{z}') = 0$ implies π and π' must be identical as functions on $\mathcal{Z}_2$. ■

Proof [Proposition 3] Observe that, using Prop. 2 for sub-additivity of the metric $d_{\mathcal{P}}$,

$$\begin{aligned} d_{\mathcal{S}}(S^{(n)}, \tilde{S}^{(n)}) &= d_{\mathcal{P}}(\kappa(S^{(n)}), \kappa(\tilde{S}^{(n)})) \\ &\leq d_{\mathcal{P}}(\kappa(S^{(n)}), \pi) + d_{\mathcal{P}}(\pi, \tilde{\pi}) + d_{\mathcal{P}}(\tilde{\pi}, \kappa(\tilde{S}^{(n)})) \\ &= d_{\mathcal{P}}(\pi, \tilde{\pi}) + \|\pi - \kappa(S^{(n)})\|_{L^2(\Lambda_2)} + \|\tilde{\pi} - \kappa(\tilde{S}^{(n)})\|_{L^2(\Lambda_2)} \end{aligned}$$

Since κ is consistent we have $\|\pi - \kappa(S^{(n)})\|_2 = o_P(1)$ and $\|\tilde{\pi} - \kappa(\tilde{S}^{(n)})\|_2 = o_P(1)$. This completes the proof. ■

Proof [Proposition 4] This proof extends the simpler proof given for the univariate case in Theorem 6.11 of Wassermann (2006). For convenience, and without loss of generality, we suppose that $\mathcal{Z}_2 = [0,1]^2$. It will be convenient in this section to re-assign the notation $\mathbf{z}$ as a dummy variable in $\mathcal{Z}_2$ (instead of in $\mathcal{Z}_p$). Let

$$p_{i,j} = \int_{B_{i,j}} \pi \mathrm{d}\Lambda_2$$

be the probability mass assigned to

$$\begin{aligned} B_{i,j} = &\left[z_{\min} + (z_{\max} - z_{\min})\frac{i-1}{M}, z_{\min} + (z_{\max} - z_{\min})\frac{i}{M} \right) \\ &\times \left[z_{\min} + (z_{\max} - z_{\min})\frac{j-1}{M}, z_{\min} + (z_{\max} - z_{\min})\frac{j}{M} \right), \end{aligned}$$

so that, from binomial properties, the mean and variance of the histogram estimator $\kappa(S^{(n)})(\mathbf{z})$ at the point $\mathbf{z} \in \mathcal{Z}_2$ are

$$\begin{aligned} m(\mathbf{z}) &= \frac{p_{i,j}}{h^2} \\ v(\mathbf{z}) &= \frac{p_{i,j}(1-p_{i,j})}{nh^4}. \end{aligned}$$

Let $b(\mathbf{z}) = m(\mathbf{z}) - \pi(\mathbf{z})$ denote the bias of the histogram estimator. The mean square of the error $\pi(\mathbf{z}) - \kappa(S^{(n)})(\mathbf{z})$ at a point $\mathbf{z}' \in \mathcal{Z}_2$ can be bias-variance decomposed:

$$\mathbb{E}\{[\pi(\mathbf{z}) - \kappa(S^{(n)})(\mathbf{z})]^2\} = b(\mathbf{z})^2 + v(\mathbf{z})$$

The aim is to obtain independent bounds on both the bias and variance terms next.

To bound the bias term, Taylor's theorem gives that, for $\mathbf{z}, \mathbf{z}' \in B_{i,j}$,

$$\pi(\mathbf{z}') = \pi(\mathbf{z}) + (\mathbf{z}' - \mathbf{z})^\top \cdot \nabla\pi(\mathbf{z}) + \frac{1}{2}(\mathbf{z}' - \mathbf{z})^\top \; R_{i,j}(\mathbf{z}) \; (\mathbf{z}' - \mathbf{z}) \tag{7}$$

where the remainder term satisfies

$$\begin{aligned} \|R_{i,j}(\mathbf{z})\|_{\max} &\leq \sup_{\mathbf{z}'' \in B_{i,j}} \|\nabla\nabla^\top \pi(\mathbf{z}'')\|_{\max} \quad \text{(Taylor)} \\ &\leq \sup_{\mathbf{z}'' \in \mathcal{Z}_2} \|\nabla\nabla^\top \pi(\mathbf{z}'')\|_{\max} < \infty \quad \text{(continuous on compact domain).} \end{aligned}$$

Here $\|M\|_{\max} = \max\{M_{i,j}\}$ and $\nabla\nabla^\top\pi$ denotes the Hessian, which exists since π is twice continuously differentiable in $\mathcal{Z}_2$. Thus for $\mathbf{z} \in B_{i,j}$, integrating (7):

$$\int_{B_{i,j}} \pi(\mathbf{z}')\mathrm{d}\Lambda_2(\mathbf{z}') = h^2\pi(\mathbf{z}) + h^2\left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) + E_{i,j}(\mathbf{z})$$

where the new remainder term can be bounded:

$$\begin{aligned} |E_{i,j}(\mathbf{z})| &= \left| \frac{1}{2}\int_{B_{i,j}} (\mathbf{z}' - \mathbf{z})^\top \; R_{i,j}(\mathbf{z}) \; (\mathbf{z}' - \mathbf{z})\mathrm{d}\Lambda_2(\mathbf{z}') \right| \\ &\leq \frac{1}{2}\int_{B_{i,j}} \|\mathbf{z}' - \mathbf{z}\|_2^2 \mathrm{d}\Lambda_2(\mathbf{z}') \times \sup_{\mathbf{z}'' \in \mathcal{Z}_2} \|\nabla\nabla^\top\pi(\mathbf{z}'')\|_{\max} \\ &\leq 8h^4 \sup_{\mathbf{z}'' \in \mathcal{Z}_2} \|\nabla\nabla^\top\pi(\mathbf{z}'')\|_{\max} =: Ch^4 \end{aligned} \tag{8}$$

where the constant C is independent of $\mathbf{z}$ and i, j. The number 8 (which is not sharp) is obtained from trivial but tedious computation of the integral in (8) and bounding each term in the result. Now, for $\mathbf{z} \in B_{i,j}$, the bias is expressed using (7) as

$$\begin{aligned} b(\mathbf{z}) &= \frac{1}{h^2}\int_{B_{i,j}} \pi(\mathbf{z}')\mathrm{d}\Lambda_2(\mathbf{z}') - \pi(\mathbf{z}) \\ &= \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) + \frac{1}{h^2}E_{i,j}(\mathbf{z}). \end{aligned}$$

Now we integrate this expression over $\mathbf{x} \in B_{i,j}$:

$$\begin{aligned} \int_{B_{i,j}} b^2 \mathrm{d}\Lambda_2 &= \int_{B_{i,j}} \left\{ \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) + \frac{1}{h^2}E_{i,j}(\mathbf{z}) \right\}^2 \mathrm{d}\Lambda_2(\mathbf{z}) \\ &\leq \int_{B_{i,j}} \left\{ \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) \right\}^2 \mathrm{d}\Lambda_2(\mathbf{z}) \\ &\quad + 2\int_{B_{i,j}} \left| \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) \right| \frac{1}{h^2}|E_{i,j}(\mathbf{z})|\mathrm{d}\Lambda_2(\mathbf{z}) \\ &\quad + \int_{B_{i,j}} \frac{1}{h^4}E_{i,j}(\mathbf{z})^2 \mathrm{d}\Lambda_2(\mathbf{z}) \\ &\leq \int_{B_{i,j}} \left\{ \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) \right\}^2 \mathrm{d}\Lambda_2(\mathbf{z}) \\ &\quad + 2Ch^2 \int_{B_{i,j}} \left| \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) \right| \mathrm{d}\Lambda_2(\mathbf{z}) + C^2h^2 \end{aligned}$$

To bound these integrals we use Cauchy-Schwarz:

$$\begin{aligned} \int_{B_{i,j}} \left\{ \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) \right\}^2 \mathrm{d}\Lambda_2(\mathbf{z}) &\leq \int_{B_{i,j}} \left\| \frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z} \right\|_2^2 \|\nabla\pi(\mathbf{z})\|_2^2 \mathrm{d}\Lambda_2(\mathbf{z}) \\ &\leq \frac{h^2}{2}\int_{B_{i,j}} \|\nabla\pi(\mathbf{z})\|_2^2 \mathrm{d}\Lambda_2(\mathbf{z}) \end{aligned} \tag{9}$$

and

$$\begin{aligned} \int_{B_{i,j}} \left| \left(\frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z}\right) \cdot \nabla\pi(\mathbf{z}) \right| \mathrm{d}\Lambda_2(\mathbf{z}) &\leq \int_{B_{i,j}} \left\| \frac{h}{2}\begin{bmatrix} 2i-1 \\ 2j-1 \end{bmatrix} - \mathbf{z} \right\|_2 \|\nabla\pi(\mathbf{z})\|_2 \mathrm{d}\Lambda_2(\mathbf{z}) \\ &\leq \frac{h}{\sqrt{2}}\int_{B_{i,j}} \|\nabla\pi(\mathbf{z})\|_2 \mathrm{d}\Lambda_2(\mathbf{z}). \end{aligned} \tag{10}$$

Both expressions in (9) and (10) are finite since the integrand is continuous and the domain is compact. The total integrated bias is thus bounded as

$$\int_{\mathcal{Z}_2} b^2 \mathrm{d}\Lambda_2 \leq \frac{h^2}{2}\int_{\mathcal{Z}_2} \|\nabla\pi(\mathbf{z})\|_2^2 \mathrm{d}\Lambda_2(\mathbf{z}) + C^2h^2 + O(h^3)$$

To bound the variance term, from the integral form of the mean value theorem we have that, for some $\mathbf{z}_{i,j} \in B_{i,j}$,

$$p_{i,j} = \int_{B_{i,j}} \pi \mathrm{d}\Lambda_2 = h^2\pi(\mathbf{z}_{i,j}).$$

The application of the integral form of the mean value theorem is valid since π is continuous on $\mathcal{Z}_2$. Then:

$$\begin{aligned}
\int_{\mathcal{Z}_2} v^2 \mathrm{d}\Lambda_2 &= \sum_{i,j=1}^{M} \int_{B_{i,j}} v \mathrm{d}\Lambda_2 \\
&= \sum_{i,j=1}^{M} \int_{B_{i,j}} \frac{p_{i,j}(1-p_{i,j})}{nh^4} \mathrm{d}\Lambda_2 \\
&= \frac{1}{nh^2} - \frac{1}{nh^2} \sum_{i,j=1}^{M} p_{i,j}^2 \\
&= \frac{1}{nh^2} - \frac{h^2}{n} \sum_{i,j=1}^{M} \pi(\mathbf{z}_{i,j})^2 \\
&= \frac{1}{nh^2} - \frac{1}{n}\left(\int_{\mathcal{X}_2} \pi^2 \mathrm{d}\Lambda_2 + o(1)\right) \;=\; \frac{1}{nh^2} + O\left(\frac{1}{n}\right)
\end{aligned}$$

Putting this all together to obtain a bound:

$$\begin{aligned}
\mathbb{E}\|\pi - \kappa(S^{(n)})\|^2_{L^2(\Lambda_2)} &= \int_{\mathcal{X}_2} b^2 \mathrm{d}\Lambda_2 + \int_{\mathcal{Z}_2} v \mathrm{d}\Lambda_2 \quad \text{(Fubini)} \\
&\leq \frac{h^2}{2} \int_{\mathcal{Z}_2} \|\nabla \pi(\mathbf{z})\|_2^2 \mathrm{d}\Lambda_2(\mathbf{z}) + C^2 h^2 + O(h^3) + \frac{1}{nh^2} + O\left(\frac{1}{n}\right) \qquad (11)
\end{aligned}$$

where $\mathbb{E}$ denotes expectation with respect to sampling of the data $S^{(n)} \sim \Pi$. From inspection of (11), the estimator error vanishes provided that h is chosen such that $nh^2 \to \infty$. Since convergence in expectation implies convergence in probability, we have established that $\|\pi - \kappa(S^{(n)})\|_{L^2(\Lambda_2)} = o_P(1)$. The bandwidth h^*, which minimizes the upper bound in (11), is

$$h^* = \frac{1}{n^{1/4}} \left(\frac{2}{\int_{\mathcal{Z}_2} \|\nabla \pi(\mathbf{z})\|_2^2 \mathrm{d}\Lambda_2(\mathbf{z}) + 2C^2} \right)^{1/4}$$

and with this choice we have that $\mathbb{E}\|\pi - \kappa(S^{(n)})\|^2_{L^2(\Lambda_2)} = O_P(n^{-1/2})$. For $h = h^*$ we have thus established that $\|\pi - \kappa(S^{(n)})\|_{L^2(\Lambda_2)} = O_P(n^{-1/4})$. ■

Appendix B. Consistency of the Classifier

Let $\mathcal{X}$ be the compact metric space $\mathcal{X} = \bigtimes_{1 \leq i,j \leq M} [0, n]$ from the main text, where n (the number of points in each scatter plot) is fixed. Let $\mathcal{Y} = \mathbb{R}$, so that $\{-1, +1\} \subset \mathcal{Y}$. This section studies the performance of the classifier $\hat{c} : \mathcal{X} \to \{-1, +1\}$, $\hat{c}(\mathbf{x}) = \mathrm{sign}(\hat{f})$, where $\hat{f}$ is the Laplacian-regularized least squares method from (4) in the main text, trained on labelled data $\{(\mathbf{x}_{[k]}, y_{[k]}) : [k] \in \mathcal{L}\}$ and unlabelled data $\{\mathbf{x}_{[k]} : [k] \in \mathcal{U}\}$, where $\mathbf{x}_{[k]} \in \mathcal{X}$ and $y_{[k]} \in \mathcal{Y}$. To this end, we must establish a context in which the data pairs $(\mathbf{x}_{[k]}, y_{[k]})$ can be considered to be generated. Let $\rho_{\mathcal{X},\mathcal{Y}}$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$, with marginals

$\rho_{\mathcal{X}}, \rho_{\mathcal{Y}}$ and conditional $\rho_{\mathcal{Y}|\mathcal{X}}$. In this theoretical investigation we suppose that all data are generated independently from $\rho_{\mathcal{X},\mathcal{Y}}$, with the values $\{y_{[k]} : [k] \in \mathcal{U}\}$ being withheld.

For a generic classifier $c : \mathcal{X} \to \{-1, +1\}$, define the misclassification rate

$$\mathcal{R}(c) = \frac{1}{2} \int |y - c(\mathbf{x})| \mathrm{d}\rho_{\mathcal{X},\mathcal{Y}}(\mathbf{x}, y).$$

This is minimized by $c_\rho(\mathbf{x}) := \mathrm{sign}(f_\rho(\mathbf{x}))$ where $f_\rho : \mathcal{X} \to \mathcal{Y}$ is the (typically unavailable) regression function

$$f_\rho(\mathbf{x}) = \int y \; \mathrm{d}\rho_{\mathcal{Y}|\mathcal{X}}(y|\mathbf{x}).$$

Thus the quantity $\mathcal{R}(c_\rho)$ captures the intrinsic difficulty of the classification task. A classifier $\hat{c}$ is said to be *consistent* (either in expectation, with high probability, etc.) if $\mathcal{R}(\hat{c}) \to \mathcal{R}(c_\rho)$ in the limit $m_{\mathcal{L}} \to \infty$ of infinite labelled data (with convergence either in expectation, with high probability, etc.). Our consistency argument is based around the following straightforward bound:

Lemma 6 *Fix $\epsilon > 0$ and let $\mathcal{X}_\epsilon := \{\mathbf{x} \in \mathcal{X} : |f_\rho(\mathbf{x})| < \epsilon\}$. Then*

$$\mathcal{R}(\hat{c}) \leq \mathcal{R}(c_\rho) + \rho_{\mathcal{X}}(\mathcal{X}_\epsilon) + \frac{1}{2\epsilon} \|\hat{f} - f_\rho\|_{L^1(\rho_{\mathcal{X}})},$$

where $\rho_{\mathcal{X}}(\mathcal{X}_\epsilon)$ denotes the $\rho_{\mathcal{X}}$-measure of the set $\mathcal{X}_\epsilon$.

Proof For all $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, we have that

$$|y - \mathrm{sign}(\hat{f}(\mathbf{x}))| \leq |y - \mathrm{sign}(f_\rho(\mathbf{x}))| + |\mathrm{sign}(f_\rho(\mathbf{x})) - \mathrm{sign}(\hat{f}(\mathbf{x}))|$$

so in particular

$$\mathcal{R}(\hat{c}) \leq \mathcal{R}(c_\rho) + \frac{1}{2} \|\mathrm{sign}(f_\rho) - \mathrm{sign}(\hat{f})\|_{L^1(\rho_{\mathcal{X}})}. \qquad (12)$$

Now,

$$\|\mathrm{sign}(f_\rho) - \mathrm{sign}(\hat{f})\|_{L^1(\rho_{\mathcal{X}})} = \underbrace{\int_{\mathcal{X}_\epsilon} |\mathrm{sign}(f_\rho) - \mathrm{sign}(\hat{f})| \mathrm{d}\rho_{\mathcal{X}}}_{(*)} + \underbrace{\int_{\mathcal{X} \setminus \mathcal{X}_\epsilon} |\mathrm{sign}(f_\rho) - \mathrm{sign}(\hat{f})| \mathrm{d}\rho_{\mathcal{X}}}_{**}.$$

To bound $(*)$, we note that the integrand is trivially bounded by 2. To bound $(**)$, we note that if $|f_\rho(\mathbf{x})| > \epsilon$ then $\mathrm{sign}(f_\rho) \neq \mathrm{sign}(\hat{f})$ implies that $|\hat{f}(\mathbf{x}) - f_\rho(\mathbf{x})| > 2\epsilon$. Thus

$$(*) + (**) \;\leq\; 2\rho_{\mathcal{X}}(\mathcal{X}_\epsilon) + \int_{\mathcal{X} \setminus \mathcal{X}_\epsilon} \frac{|f_\rho(\mathbf{x}) - \hat{f}(\mathbf{x})|}{\epsilon} \mathrm{d}\rho_{\mathcal{X}}(\mathbf{x}) \;=\; 2\rho_{\mathcal{X}}(\mathcal{X}_\epsilon) + \frac{1}{\epsilon} \|\hat{f} - f_\rho\|_{L^1(\rho_{\mathcal{X}})} \qquad (13)$$

Combining (12) and (13) completes the proof. ■

Next we leverage an existing high-probability consistency result established in the regression (as opposed to classification) context:

Theorem 7 *Suppose f_ρ is non-constant and that $\Sigma_K^{-\frac{\alpha}{2}} f_\rho \in L^2(\rho_\mathcal{X})$ for some $\alpha \in (0,1]$. Let $\theta = \frac{1}{(1+\alpha)(1+s)}$. Take $\lambda_1 = m_\mathcal{U}^\theta$ and $\lambda_2 = m_\mathcal{L}^\theta$. Then there exists a finite constant C such that for any $\delta \in (0,1)$, and for $m_\mathcal{L}, m_\mathcal{U}$ sufficiently large, we have with probability at least $1-8\delta$ that*

$$\|\hat{f} - f_\rho\|_{L^1(\rho_\mathcal{X})} \leq C \log\left(\frac{2}{\delta}\right) m_\mathcal{L}^{-\alpha\theta}. \tag{14}$$

Proof This result is an immediate consequence of Theorem 5.6 in Cao and Chen (2012), whose bound on the $L^2(\rho_\mathcal{X})$ error clearly also implies a bound on the $L^1(\rho_\mathcal{X})$ error. In addition, since our intention in what follows is limited to establishing consistency of the proposed classification method, as opposed to a detailed convergence rate analysis, we have simplified the presentation by stating a slightly weaker but less-verbose upper bound. ■

Note how the "for $m_\mathcal{U}$ sufficiently large" condition in Theorem 7 will typically be automatically satisfied in our context, where the amount of unlabelled data is $m_\mathcal{U} = O(p^2)$. Thus the content of (14) is control over $\hat{f} - f_\rho$ as the number $m_\mathcal{L}$ of labeled data is increased.

Corollary 8 *Under the same assumptions as Theorem 7, we have with probability at least $1-8\delta$ that*

$$\mathcal{R}(\hat{c}) \leq \mathcal{R}(c_\rho) + \rho_\mathcal{X}(\mathcal{X}_\epsilon) + \frac{C}{2\epsilon} \log\left(\frac{2}{\delta}\right) m_\mathcal{L}^{-\alpha\theta}. \tag{15}$$

Corollary 8 makes explicit how the intrinsic difficulty of the classification task depends on the form of f_ρ, and in particular the extent to which $|f_\rho(\mathbf{x})| < \epsilon$ occurs in $\mathcal{X}$. For typical regression functions f_ρ with simple roots in $\mathcal{X}$, it will hold that $\rho_\mathcal{X}(\mathcal{X}_\epsilon) = O(\epsilon)$. An assumption of this form can therefore be used to complete a high probability consistency argument:

Corollary 9 (Consistency of the Classifier) *Suppose that $\rho_\mathcal{X}(\mathcal{X}_\epsilon) = O(\epsilon^\gamma)$ for some $\gamma > 0$. Under the same assumptions as Theorem 7, there exists a finite constant $\tilde{C}$ such that, with probability at least $1-8\delta$,*

$$\mathcal{R}(\hat{c}) \leq \mathcal{R}(c_\rho) + \tilde{C}\left(\log\left(\frac{2}{\delta}\right)\right)^{\frac{\gamma}{1+\gamma}} m_\mathcal{L}^{-\frac{\alpha\theta\gamma}{1+\gamma}}.$$

In particular, this establishes that the classifier $\hat{c}$ is (with high probability) consistent.

Proof From the hypothesis, $\exists B_1$, ϵ_1 such that $\rho_\mathcal{X}(\mathcal{X}_\epsilon) \leq B_1\epsilon^\gamma$ for all $\epsilon < \epsilon_1$. Thus, for $\epsilon < \epsilon_1$ the difference $\mathcal{R}(\hat{c}) - \mathcal{R}(c_\rho)$ can be bounded via (15) as

$$\begin{aligned}\mathcal{R}(\hat{c}) - \mathcal{R}(c_\rho) &\leq \rho_\mathcal{X}(\mathcal{X}_\epsilon) + \frac{C}{2\epsilon}\log\left(\frac{2}{\delta}\right) m_\mathcal{L}^{-\alpha\theta} \\ &\leq B_1\epsilon^\gamma + \frac{B_2}{\epsilon} =: J(\epsilon)\end{aligned}$$

where $B_2 = \frac{C}{2}\log\left(\frac{2}{\delta}\right) m_\mathcal{L}^{-\alpha\theta}$. Differentiating J and setting to zero reveals that J is minimized over $(0,\infty)$ at

$$\epsilon^* = \left(\frac{B_2}{\gamma B_1}\right)^{\frac{1}{1+\gamma}},$$

which satisfies $\epsilon^* < \epsilon_1$ for $m_\mathcal{L}$ sufficiently large (recall that $m_\mathcal{L}$ being sufficiently large was an assumption of Theorem 7). Thus, for $m_\mathcal{L}$ sufficiently large,

$$\mathcal{R}(\hat{c}) - \mathcal{R}(c_\rho) \;\leq\; J(\epsilon^*) \;=\; \left(\gamma^{-\frac{\gamma}{1+\gamma}} + \gamma^{\frac{1}{1+\gamma}}\right) B_1^{\frac{1}{1+\gamma}} B_2^{\frac{\gamma}{1+\gamma}}$$

which, upon substitution for B_2, yields the required result with the value for the constant $\tilde{C} = \left(\gamma^{-\frac{\gamma}{1+\gamma}} + \gamma^{\frac{1}{1+\gamma}}\right) B_1^{\frac{1}{1+\gamma}} \left(\frac{C}{2}\right)^{\frac{\gamma}{1+\gamma}}$. ■

Appendix C. Additional Figures

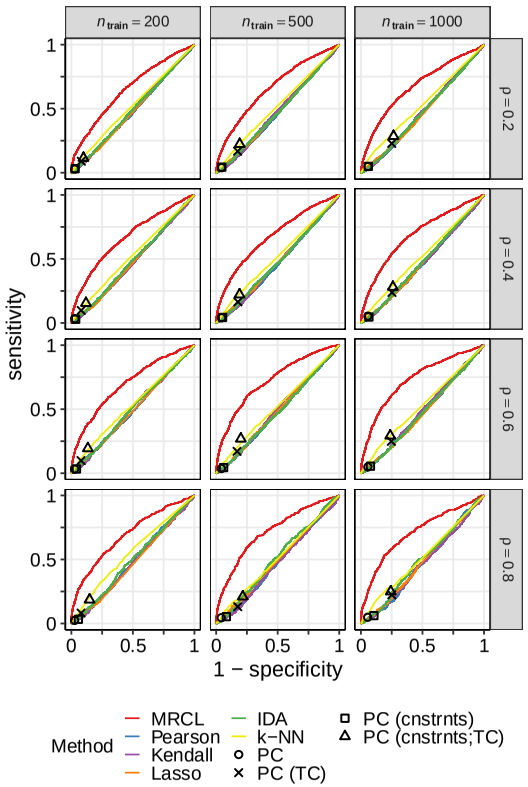


Figure 6: Data set D1 (yeast data), random sampling. ROC curves are shown with respect to causal relationships determined from unseen interventional data (see Main Text for details). Results for PC (which returns a point estimate) are shown as locations on the ROC plane. "TC" indicates use of a transitive closure operation and "cnstrnts" indicates that the background information Φ was included via input constraints. [Results shown are for significance level $\alpha = 0.01$ and for a lenient interpretation where possible edges are included. Results are averages over 25 iterations.]

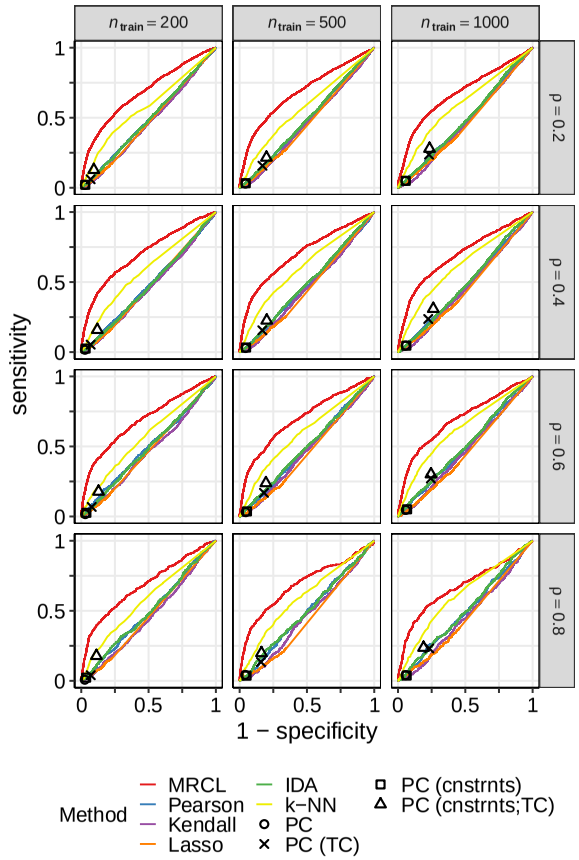


Figure 7: Data set D1 (yeast data), row-wise sampling. ROC curves are shown with respect to causal relationships determined from unseen interventional data (see Main Text for details). Results for PC (which returns a point estimate) are shown as locations on the ROC plane. "TC" indicates use of a transitive closure operation and "cnstrnts" indicates that the background information Φ was included via input constraints. [Results shown are for significance level $\alpha = 0.01$ and for a lenient interpretation where possible edges are included. Results are averages over 25 iterations.]

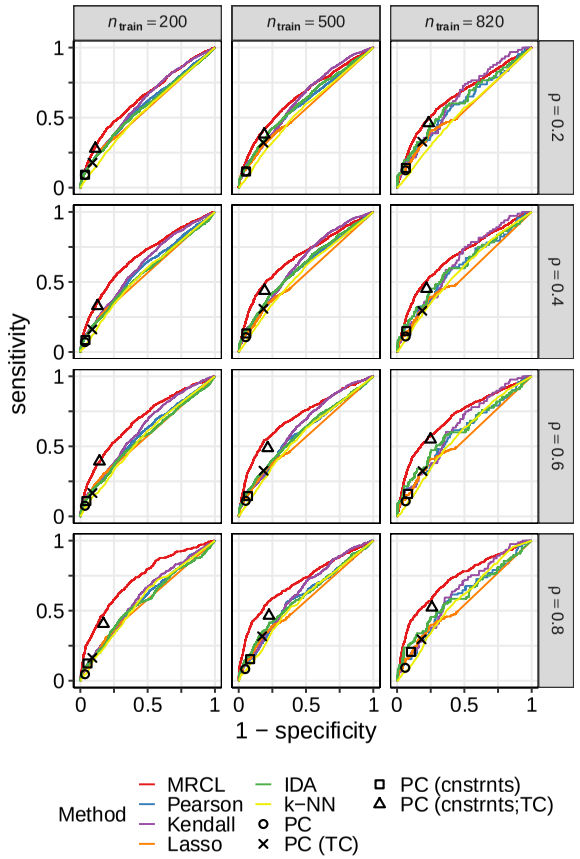


Figure 8: Data set D3 (cancer protein data). ROC curves are shown with respect to a reference graph determined from the scientific literature (see Main Text for details). Results for PC (which returns a point estimate) are shown as locations on the ROC plane. "TC" indicates use of a transitive closure operation and "cnstrnts" indicates that the background information Φ was included via input constraints. The "TC" results are included here for completeness, but we note that the reference graph here encodes direct, rather than ancestral, relationships. [Results shown are for significance level $\alpha = 0.01$ and for a lenient interpretation where possible edges are included. Results are averages over 25 iterations.]

References

R. Akbani, P.K.S. Ng, H.M. Werner, M. Shahmoradgoli, F. Zhang, Z. Ju, W. Liu, J.Y. Yang, K. Yoshihara, J. Li, S. Ling, E.G. Seviour, P.T. Ram, J.D. Minna, L. Diao, P. Tong, J.V. Heymach, S.M. Hill, F. Dondelinger, N. Städler, L.A. Byers, F. Meric-Bernstam, J.N. Weinstein, B.M. Broom, R.G.W. Verhaak, H. Liang, S. Mukherjee, Y. Lu, and G.B. Mills. A pan-cancer proteomic perspective on The Cancer Genome Atlas. *Nature Communications*, 5:3887, 2014.

M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Y. Cao and D. Chen. Generalization errors of Laplacian regularized least squares regression. *Science China Mathematics*, 55(9):1859–1868, 2012.

D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40 (1):294–321, 2012.

F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 522–530, 2009.

A. Grigor'yan. Heat kernels on weighted manifolds and applications. *Cont. Math*, 398: 93–191, 2006.

A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13: 2409–2464, 2012.

S.M. Hill, L.M. Heiser, T. Cokelaer, M. Unger, N.K. Nesser, D.E. Carlin, Y. Zhang, A. Sokolov, E.O. Paull, C.K. Wong, K. Graim, A. Bivol, H. Wang, F. Zhu, B. Afsari, L.V. Danilova, A.V. Favorov, W.S. Lee, D. Taylor, C.W. Hu, B.L. Long, D.P. Noren, A.J. Bisberg, The HPN-DREAM Consortium, G.B. Mills, J.W. Gray, M. Kellen, T. Norman, S. Friend, A.A. Qutub, Y. Fertig, E.J. Guan, M. Song, J.M. Stuart, P.T. Spellman, H. Koeppl, G. Stolovitzky, J. Saez-Rodriguez, and S. Mukherjee. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature Methods*, 13(4):310–318, 2016.

S.M. Hill, N.K. Nesser, K. Johnson-Camacho, M. Jeffress, A. Johnson, C. Boniface, S.E.F. Spencer, Y. Lu, L.M. Heiser, Y. Lawrence, N.T. Pande, J.E. Korkola, J.W. Gray, G.B. Mills, S. Mukherjee, and P.T. Spellman. Context specificity in causal signaling networks revealed by phosphoprotein profiling. *Cell Systems*, 4(1):73–83, 2017.

A. Hyttinen, F. Eberhardt, and P. O. Hoyer. Learning linear cyclic causal models with latent variables. *Journal of Machine Learning Research*, 13:3387–3439, 2012.

P. Kemmeren, K. Sameith, L.A. van de Pasch, J.J. Benschop, T.L. Lenstra, T. Margaritis, E. O'Duibhir, E. Apweiler, S. van Wageningen, C.W. Ko, S. van Heesch, M.M. Kashani, G. Ampatziadis-Michailidis, M.O. Brok, N.A.C.H. Brabers, A.J. Miles, D. Bouwmeester, S.R. van Hooff, H. Bakel, E. Sluiters, L.V. Bakker, B. Snel, P. Lijnzaad, D. van Leenen, M.J.A. Groot Koerkamp, and F.C.P. Holstege. Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors. *Cell*, 157(3):740–752, 2014.

J. Li, Y. Lu, R. Akbani, Z. Ju, P. L. Roebuck, W. Liu, J-Y. Yang, B.M. Broom, R.G.W. Verhaak, D.W. Kane, C. Wakefield, J.N. Weinstein, G.B. Mills, and H. Liang. TCPA: a resource for cancer functional proteomics data. *Nature Methods*, 10(11):1046–1047, 2013.

D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin. Towards a learning theory of causation. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1452–1461, 2015.

M.H. Maathuis, M. Kalisch, and P. Bühlmann. Estimating high-dimensional intervention effects from observational data. *Annals of Statistics*, 37(6A):3133–3164, 2009.

M.H. Maathuis, D. Colombo, M. Kalisch, and P. Bühlmann. Predicting causal effects in large-scale systems from observational data. *Nature Methods*, 7(4):247–248, 2010.

N. Meinshausen, A. Hauser, J.M. Mooij, J. Peters, P. Versteeg, and P. Bühlmann. Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences*, 113(27):7361–7368, 2016.

J.M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 17(32):1–102, 2016.

J. Pearl. *Causality*. Cambridge University Press, 2009.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B*, 78(5):947–1012, 2016.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `https://www.R-project.org/`.

T. Richardson. A discovery algorithm for directed cyclic graphs. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 454–461, 1996.

P. Spirtes. Directed cyclic graphical representations of feedback models. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 491–498, 1995.

P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2000.

N. Städler and S. Mukherjee. Two-sample testing in high dimensions. *Journal of the Royal Statistical Society: Series B*, 79(1):225–246, 2017.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

M.P. Wand and M.C. Jones. *Kernel Smoothing*. CRC Press, 1994.

J. Wang, T. Jebara, and S.-F. Chang. Graph transduction via alternating minimization. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1144–1151, 2008.

L. Wasserman and K. Roeder. High dimensional variable selection. *Annals of Statistics*, 37 (5A):2178, 2009.

L. Wassermann. *All of Nonparametric Statistics*. Springer, 2006.

Journal of Machine Learning Research 20 (2019) 1-73 Submitted 6/18; Revised 7/19; Published 8/19

Unsupervised Basis Function Adaptation for Reinforcement Learning

Edward Barker EBARKER@STUDENT.UNIMELB.EDU.AU
School of Mathematics and Statistics
University of Melbourne
Melbourne, Victoria 3010, Australia

Charl Ras CJRAS@UNIMELB.EDU.AU
School of Mathematics and Statistics
University of Melbourne
Melbourne, Victoria 3010, Australia

Editor: George Konidaris

Abstract

When using reinforcement learning (RL) algorithms it is common, given a large state space, to introduce some form of approximation architecture for the value function (VF). The exact form of this architecture can have a significant effect on an agent's performance, however, and determining a suitable approximation architecture can often be a highly complex task. Consequently there is currently interest among researchers in the potential for allowing RL algorithms to adaptively generate (i.e. to learn) approximation architectures. One relatively unexplored method of adapting approximation architectures involves using feedback regarding the frequency with which an agent has visited certain states to guide which areas of the state space to approximate with greater detail. In this article we will: (a) informally discuss the potential advantages offered by such methods; (b) introduce a new algorithm based on such methods which adapts a state aggregation approximation architecture online and is designed for use in conjunction with SARSA; (c) provide theoretical results, in a policy evaluation setting, regarding this particular algorithm's complexity, convergence properties and potential to reduce VF error; and finally (d) test experimentally the extent to which this algorithm can improve performance given a number of different test problems. Taken together our results suggest that our algorithm (and potentially such methods more generally) can provide a versatile and computationally lightweight means of significantly boosting RL performance given suitable conditions which are commonly encountered in practice.

Keywords: reinforcement learning, unsupervised learning, basis function adaptation, state aggregation, SARSA

1. Introduction

Traditional reinforcement learning (RL) algorithms such as TD(λ) (Sutton, 1988) or Q-learning (Watkins and Dayan, 1992) can generate optimal policies, when dealing with small state and action spaces, by exactly representing the value function (VF). However, when

©2019 Edward Barker and Charl Ras.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-392.html.

environments are complex (with large or continuous state or action spaces), using such algorithms directly becomes too computationally demanding. As a result it is common to introduce some form of architecture with which to approximate the VF, for example a parametrised set of functions (Sutton and Barto, 2018; Bertsekas and Tsitsiklis, 1996). One issue when introducing VF approximation, however, is that the accuracy of the algorithm's VF estimate, and as a consequence its performance, is highly dependent upon the exact form of the architecture chosen (it may be, for example, that no element of the chosen set of parametrised functions closely fits the VF). Accordingly, a number of authors have explored the possibility of allowing the approximation architecture to be *learned* by the agent, rather than pre-set manually by the designer—see Busoniu et al. (2010) for an overview. It is hoped that, by doing this, we can design algorithms which will perform well within a more general class of environment whilst requiring less explicit input from designers.[1]

A simple and perhaps, as yet, under-explored method of adapting an approximation architecture involves using an estimate of the frequency with which an agent has visited certain states to determine which states should have their values approximated in greater detail. We might be interested in such methods since, intuitively, we would suspect that areas which are visited more regularly are, for a number of reasons, more "important" in relation to determining a policy. Such a method can be contrasted with the more commonly explored method of explicitly measuring VF error and using this error as feedback to adapt an approximation architecture. We will refer to methods which adapt approximation architectures using visit frequency estimates as being *unsupervised* in the sense that no direct reference is made to reward or to any estimate of the VF.

Our intention in this article is to provide—in the setting of problems with large or continuous state spaces, where reward and transition functions are unknown, and where our task is to maximise reward—an exploration of unsupervised methods along with a discussion of their potential merits and drawbacks. We will do this primarily by introducing an algorithm, PASA, which represents an attempt to implement an unsupervised method in a manner which is as simple and obvious as possible. The algorithm will form the principle focus of our theoretical and experimental analysis.

It will turn out that unsupervised techniques have a number of advantages which may not be offered by other more commonly used methods of adapting approximation architectures. In particular, we will argue that unsupervised methods have (a) low computational overheads and (b) a tendency to require less sampling in order to converge. We will also argue that the methods can, under suitable conditions, (c) decrease VF error, in some cases significantly, with minimal input from the designer, and, as a consequence, (d) boost performance. The methods will be most effective in environments which satisfy certain conditions, however these conditions are likely to be satisfied by many of the environments we encounter most commonly in practice. We will also see that the principle of using state visit frequency can, perhaps counter-intuitively, interact favourably with the process of exploration.[2] The

1. Introducing the ability to adapt an approximation architecture is in some ways similar to simply adding additional parameters to an approximation architecture. However separating parameters into two sets, those adjusted by the underlying RL algorithm, and those adjusted by the adaptation method, permits us scope to, amongst other things, specify two distinct update rules.
2. Whilst policy changes will result in changes to visit frequency, in practice this co-dependency does not typically result in instability, but rather assists the agent to narrow in on strong policies. We explore the nature of this interaction more fully in Section 1.5 and Section 4 (in particular Section 4.4).

fact that unsupervised methods are cheap and simple, yet still have significant potential to enhance performance, makes them appear a promising, if perhaps somewhat overlooked, means of adapting approximation architectures.

1.1. Article Overview

Our article is structured as follows. Following some short introductory sections we will offer an informal discussion of the potential merits of unsupervised methods in order to motivate and give a rationale for our exploration (Section 1.5). We will then propose (in Section 2) our new algorithm, PASA, short for "Probabilistic Adaptive State Aggregation". The algorithm is designed to be used in conjunction with SARSA, and adapts a state aggregation approximation architecture on-line.

Section 3 is devoted to a theoretical analysis of the properties of PASA. Sections 3.1 to 3.3 relate to finite state spaces. We will demonstrate in Section 3.1 that PASA has a time complexity (considered as a function of the state and action space sizes, S and A) of the same order as its SARSA counterpart, i.e. $O(1)$. It has space complexity of $O(X)$ as a function of S, where X is the number of cells in the state aggregation architecture, and $O(1)$ as a function of A. This is compared to $O(X)$ and $O(A)$ respectively for its SARSA counterpart. This means that PASA is computationally cheap: it does not carry significant computational cost beyond SARSA with fixed state aggregation.

In Section 3.2 we investigate PASA in the context of where an agent's policy is held fixed and prove that the algorithm converges. This implies that, unlike non-linear architectures in general, SARSA combined with PASA will have the same convergence properties as SARSA with a fixed linear approximation architecture (i.e. the VF estimate may, assuming the policy is updated, "chatter", or fail to converge, but will never diverge).

In Section 3.3 we will use PASA's convergence properties to obtain a theorem, again where the policy is held fixed, regarding the impact PASA will have on VF error. This theorem guarantees that VF error will be arbitrarily low as measured by routinely used scoring functions provided certain conditions are met, conditions which require primarily that the agent spends a large amount of the time in a small subset of the state space. This result permits us to argue informally that PASA will also, assuming the policy is updated, improve performance given similar conditions.

In Section 3.4 we extend the finite state space concepts to continuous state spaces. We will demonstrate that, assuming we employ an initial arbitrarily complex discrete approximation of the agent's continuous input, all of our discrete case results have a straightforward continuous state space analogue, such that PASA can be used to reduce VF error (at low computational cost) in a manner substantially equivalent to the discrete case.

In Section 3.5 we outline some examples to help illustrate the types of environment in which the conditions relevant to Sections 3.3 and 3.4 are likely to be satisfied. We will see that, even for apparently highly unstructured environments where prior knowledge of the transition function is largely absent, the necessary conditions potentially exist to guarantee that employing PASA will result in low VF error. In a key example, we will show that for environments with large state spaces and where there is no prior knowledge of the transition function, PASA will permit SARSA to generate a VF estimate with error which is arbitrarily low with arbitrarily high probability provided the transition function

and policy are sufficiently close to deterministic and the algorithm has $X = X(S) \geq f(S)$ cells available in its adaptive state aggregation architecture, where f is $O(\sqrt{S} \ln S \log_2 S)$.

To corroborate our theoretical analysis, and to further address the more complex question of whether PASA will improve overall performance, we outline some experimental results in Section 4. We explore three different types of environment: a GARNET environment,[3] a "Gridworld" type environment, and an environment representative of a logistics problem.

Our experimental results suggest that PASA, and potentially, by extension, techniques based on similar principles, can significantly boost performance when compared to SARSA with fixed state aggregation. The addition of PASA improved performance in all but one of our experiments,[4] and regularly doubled or even tripled the average reward obtained. Indeed, in some of the environments we tested, PASA was also able to outperform SARSA with *no* state abstraction, the potential reasons for which we discuss in Section 4. This is despite minimal input from the designer with respect to tailoring the algorithm to each distinct environment type.[5] Furthermore, in each case the additional processing time and resources required by PASA are measured and shown to be minimal, as predicted.

1.2. Related Works

The concept of using visit frequencies in an unsupervised manner is not completely new however it remains relatively unexplored compared to methods which seek to measure the error in the VF estimate explicitly and to then use this error as feedback. We are aware of only three papers in the literature which investigate a method similar in concept to the one that we propose, though the algorithms analysed in these three papers differ from PASA in some key respects.

Moreover there has been little by way of theoretical analysis of unsupervised techniques. The results we derive in relation to the PASA algorithm are all original, and we are not aware of any published theoretical analysis which is closely comparable.

In the first of the three papers just mentioned, Menache et al. (2005) provide a brief evaluation of an unsupervised algorithm which uses the frequency with which an agent has visited certain states to fit the centroid and scale parameters of a set of Gaussian basis functions. Their study was limited to an experimental analysis, and to the setting of policy evaluation. The unsupervised algorithm was not the main focus of their paper, but rather was used to provide a comparison with two more complex adaptation algorithms which used information regarding the VF as feedback.[6]

In the second paper, Nouri and Littman (2009) examined using a regression tree approximation architecture to approximate the VF for continuous multi-dimensional state spaces. Each node in the regression tree represents a unique and disjoint subset of the state space.

3. An environment with a discrete state space where the transition function is deterministic and generated uniformly at random. For more details refer to Sections 3.5 and 4.1.
4. The single exception was likely to have been affected by experimental noise. See Section 4.1.
5. For each problem, with the exception of the number of cells available to the state aggregation architecture, the PASA parameters were left unchanged
6. Their paper actually found the unsupervised method performed unfavourably compared to the alternative approaches they proposed. However they tested performance in only one type of environment, a type of environment which we will argue is not well suited to the methods we are discussing here (see Section 3.5).

Once a particular node has been visited a fixed number of times, the subset it represents is split ("once-and-for-all") along one of its dimensions, thereby creating two new tree nodes. The manner in which the VF estimate is updated[7] is such that incentive is given to the agent to visit areas of the state space which are relatively unexplored. The most important differences between their algorithm and ours are that, in their algorithm, (a) cell-splits are permanent, i.e. once new cells are created, they are never re-merged and (b) a permanent record is kept of each state visited (this helps the algorithm to calculate the number of times newly created cells have already been visited). With reference to (a), the capacity of PASA to re-adapt is, in practice, one of its critical elements (see Section 3). With reference to (b), the fact that PASA does not retain such information has important implications for its space complexity. The paper also provides theoretical results in relation to the optimality of their algorithm. Their guarantees apply in the limit of arbitrarily precise VF representation, and are restricted to model-based settings. In these and other aspects their analysis differs significantly from our own.

In the third paper, which is somewhat similar in approach and spirit to the second (and which also considers continuous state spaces), Bernstein and Shimkin (2010) examined an algorithm wherein a set of kernels are progressively split (again "once-and-for-all") based on the visit frequency for each kernel. Their algorithm also incorporates knowledge of uncertainty in the VF estimate, to encourage exploration. The same two differences to PASA (a) and (b) listed in the paragraph above also apply to this algorithm. Another key difference is that their algorithm maintains a distinct set of kernels for each action, which implies increased algorithm space complexity. The authors provide a theoretical analysis in which they establish a linear relationship between policy-mistake count[8] and maximum cell size in an approximation of the VF in a continuous state space.[9] The results they provide are akin to other PAC ("probably approximately correct") analyses undertaken by several authors under a range of varying assumptions—see, for example, Strehl et al. (2009) or, more recently, Jin et al. (2018). Their theoretical analysis differs from ours in many fundamental respects. Unlike our theoretical results in Section 3, their results have the advantage that they are not dependent upon characteristics of the environment and pertain to performance, not just VF error. However, similar to Nouri and Littman (2009) above, they carry the significant limitation that there is no guarantee of arbitrarily low policy-mistake count in the absence of an arbitrarily precise approximation architecture, which is equivalent in this context to arbitrarily large computational resources.[10]

There is a much larger body of work less directly related to this article, but which has central features in common, and is therefore worth mentioning briefly. Two important threads of research can be identified.

First, as noted above, a number of researchers have investigated the learning of an approximation architecture using feedback regarding the VF estimate. Early work in this area includes Singh et al. (1995), Moore and Atkeson (1995) and Reynolds (2000). Such approaches most commonly involve either (a) progressively adding features "once-and-for-all"—see for example Munos and Moore (2002),[11] Keller et al. (2006) or Whiteson et al. (2007)—based on a criteria related to VF error, or (b) the progressive adjustment of a fixed set of basis functions using, most commonly, a form of gradient descent—see, for example, Yu and Bertsekas (2009), Di Castro and Mannor (2010) and Mahadevan et al. (2013).[12] Approaches which use VF feedback form an interesting array of alternatives for adaptively generating an approximation architecture, however such approaches can be considered as "taxonomically distinct" from the unsupervised methods we are investigating. The implications of using VF feedback compared to unsupervised adaptation, including some of the comparative advantages and disadvantages, are explored in more detail in our motivational discussion in Section 1.5. We will make the argument that unsupervised methods have certain advantages not available to techniques which use VF feedback in general.

Second, given that the PASA algorithm functions by updating a state aggregation architecture, it is worth noting that a number of principally theoretical works exist in relation to state aggregation methods. These works typically address the question of how states in a Markov decision process (MDP) can be aggregated, usually based on "closeness" of the transition and reward function for groups of states, such that the MDP can be solved efficiently. Examples of papers on this topic include Hutter (2016) and Abel et al. (2016) (the results of the former apply with generality beyond just MDPs). Notwithstanding being focussed on the question of how to create effective state aggregation approximation architectures, these works differ fundamentally from ours in terms of their assumptions and overall objective. Though there are exceptions—see, for example, Ortner (2013)[13]—the results typically assume knowledge of the MDP (i.e. the environment) whereas our work assumes no such knowledge. Moreover the techniques analysed often use the VF, or a VF estimate, to generate a state aggregation, which is contrary to the unsupervised nature of the approaches we are investigating.

It is worth, before concluding our discussion of related works, making brief mention of approaches which use gradient descent to optimise a fixed set of parameters of a complex and differentiable non-linear VF approximator.[14] Such approaches encompass (in part) what are known as deep RL methods, which have recently shown impressive results on an array of challenging problems (Mnih et al., 2015; Silver et al., 2018). When using this type of approach, additional techniques are often employed to encourage stability, since few convergence guarantees exist for non-linear VF approximation architectures in general. Our algorithm, PASA, is not incompatible with these types of approach fundamentally (the SARSA algorithm which PASA supports could, hypothetically, be replaced by a non-linearly parametrised RL algorithm). However the principles underlying unsupervised basis

7. The paper proposes more than one algorithm. We refer here to the fitted Q-iteration algorithm.
8. Defined, in essence, as the number of time steps in which the algorithm executes a non-optimal policy.
9. See, in particular, their Theorems 4 and 5.
10. Our results, in contrast, provide guarantees relating to maximally reduced VF error under conditions where resources may be limited.
11. The authors in this article investigate several distinct adaptation, or "splitting", criteria. However all depend in some way on an estimate of the value function.
12. Whilst less common, some approaches have been proposed, such as Bonarini et al. (2006), which fall somewhere in between (a) and (b). In the paper just cited the authors propose a method which involves employing a cell splitting rule offset by a cell merging (or pruning) rule.
13. This paper explores the possibility of aggregating states based on learned estimates of the transition and reward function, and as such the techniques it explores differ quite significantly from those we are investigating.
14. These approaches implicitly bear some similarity to basis function adaptation methods which use VF error as feedback.

function adaptation are very different from those underlying, for example, deep RL. The motivational discussion in Section 1.5 is helpful in drawing out the core differences between unsupervised methods and approaches, such as deep RL, which apply gradient descent in conjunction with a fixed, non-linear VF approximation architecture.

1.3. Formal Framework[15]

We assume that we have an agent which interacts with an environment over a sequence of iterations $t \in \mathbb{N}$. We will assume throughout this article (with the exception of Section 3.4) that we have a finite set $\mathcal{S}$ of *states* of size S (Section 3.4 relates to continuous state spaces and contains its own formal definitions where required). We also assume we have a discrete set $\mathcal{A}$ of *actions* of size A. Since S and A are finite, we can, using arbitrarily assigned indices, label each state s_i $(1 \leq i \leq S)$ and each action a_j $(1 \leq j \leq A)$.

For each t the agent will be in a particular state and will take a particular action. Each action is taken according to a *policy* π whereby the probability the agent takes action a_j in state s_i is denoted as $\pi(a_j|s_i)$.

The *transition function* P: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ defines how the agent's state evolves over time. If the agent is in state s_i and takes an action a_j in iteration t, then the probability it will transition to the state $s_{i'}$ in iteration $t+1$ is given by $P(s_{i'}|s_i, a_j)$. The transition function must be constrained such that $\sum_{i'=1}^{S} P(s_{i'}|s_i, a_j) = 1$.

Denote as $\mathscr{R}$ the space of all probability distributions defined on the real line. The *reward function* R: $\mathcal{S} \times \mathcal{A} \to \mathscr{R}$ is a mapping from each state-action pair (s_i, a_j) to a real-valued random variable $R(s_i, a_j)$, where each $R(s_i, a_j)$ is defined by a cumulative distribution function $F_R(s_i, a_j) : \mathbb{R} \to [0,1]$, such that if the agent is in state s_i and takes action a_j in iteration t, then it will receive a real-valued *reward* in iteration t distributed according to $R(s_i, a_j)$. Some of our key results will require that $|R(s_i, a_j)| \leq c$, where c is a constant such that $0 \leq c < \infty$, for all i and j, in which case we use R_{m} to denote the maximum magnitude of the expected value of $R(s_i, a_j)$ over all i and j.

Prior to the point at which an agent begins interacting with an environment, both P and R are taken as being unknown. However we may assume in general that we are given a prior distribution for both. Our overarching objective is to design an algorithm to adjust π during the course of the agent's interaction with its environment so that total reward is maximised over some interval (for example, in the case of our experiments in Section 4, this will be a finite interval towards the end of each trial).

1.4. Scoring Functions

Whilst our overarching objective is to maximise performance, an important step towards achieving this objective involves reducing error in an algorithm's VF estimate. This is based on the assumption that more accurate VF estimates will lead to better directed policy updates, and therefore better performance. A large part of our theoretical analysis in Section 3 will be directed at assessing the extent to which VF error will be reduced under different circumstances.

15. The formal framework we assume in this article is a special case of a Markov decision process. For more general MDP definitions see, for example, Chapter 2 of Puterman (2014).

Error in a VF estimate for a fixed policy π is typically measured using a *scoring function*. It is possible to define many different types of scoring function, and in this section we will describe some of the most commonly used types.[16] We first need a definition of the VF itself. We formally define the *value function* Q_γ^π for a particular policy π, which maps each of the $S \times A$ state-action pairs to a real value, as follows:

$$Q_\gamma^\pi(s_i, a_j) \coloneqq \mathrm{E}\left(\sum_{t=1}^{\infty} \gamma^{t-1} R\big(s^{(t)}, a^{(t)}\big) \middle| s^{(1)} = s_i, a^{(1)} = a_j \right),$$

where the expectation is taken over the distributions of P, R and π (i.e. for particular instances of P and R, not over their prior distributions) and where $\gamma \in [0,1)$ is known as a *discount factor*. We will generally omit the subscript γ. We have used superscript brackets to indicate dependency on the iteration t. Initially the VF is unknown.

Suppose that $\hat{Q}$ is an estimate of the VF. One commonly used scoring function is the squared error in the VF estimate for each state-action, weighted by some arbitrary function w which satisfies $0 \leq w(s_i, a_j) \leq 1$ for all i and j. We will refer to this as the *mean squared error* (MSE):

$$\mathrm{MSE}_\gamma \coloneqq \sum_{i=1}^{S} \sum_{j=1}^{A} w(s_i, a_j) \left(Q_\gamma^\pi(s_i, a_j) - \hat{Q}(s_i, a_j) \right)^2. \tag{1}$$

Note that the true VF Q_γ^π, which is unknown, appears in (1). Many approximation architecture adaptation algorithms use a scoring function as a form of feedback to help guide how the approximation architecture should be updated. In such cases it is important that the score is something which can be measured by the algorithm. In that spirit, another commonly used scoring function (which, unlike MSE, is not a function of Q_γ^π) uses T_γ^π, the *Bellman operator*, to obtain an approximation of the MSE. This scoring function we denote as L. It is a weighted sum of the *Bellman error* at each state-action:[17]

$$L_\gamma \coloneqq \sum_{i=1}^{S} \sum_{j=1}^{A} w(s_i, a_j) \left(T_\gamma^\pi \hat{Q}(s_i, a_j) - \hat{Q}(s_i, a_j) \right)^2,$$

where:

$$T_\gamma^\pi \hat{Q}(s_i, a_j) \coloneqq \mathrm{E}\left(R(s_i, a_j)\right) + \gamma \sum_{i'=1}^{S} P(s_{i'}|s_i, a_j) \sum_{j'=1}^{A} \pi(a_{j'}|s_{i'}) \hat{Q}(s_{i'}, a_{j'}).$$

The value L still relies on an expectation within the squared term, and hence there may still be challenges estimating L empirically. A third alternative scoring function $\tilde{L}$, which

16. Sutton and Barto (2018) provide a detailed discussion of different methods of scoring VF estimates. See, in particular, Chapters 9 and 11.
17. Note that this scoring function also depends on a discount factor γ, inherited from the Bellman error definition. It is effectively analogous to the constant γ used in the definition of MSE.

steps around this problem, can be defined as follows (F_R is defined in Section 1.3):

$$\tilde{L}_\gamma := \sum_{i=1}^{S}\sum_{j=1}^{A} w(s_i,a_j)\sum_{i'=1}^{S} P(s_{i'}|s_i,a_j)\sum_{j'=1}^{A}\pi(a_{j'}|s_{i'}) \times \int_{\mathbb{R}}\left(R(s_i,a_j)+\gamma\hat{Q}(s_{i'},a_{j'})-\hat{Q}(s_i,a_j)\right)^2 dF_R(s_i,a_j).$$

These three different scoring functions are arguably the most commonly used scoring functions, and we will state results in Section 3 in relation to all three. Scoring functions which involve a projection onto the space of possible VF estimates are also commonly used. We will not consider such scoring functions explicitly, however our results below will apply to these scoring functions, since, for the architectures we consider, scoring functions with and without a projection are equivalent.

We will need to consider some special cases of the weighting function w. Towards that end we define what we will term the *stable state probability* vector $\psi = \psi(\pi, s^{(1)})$, of dimension S, as follows:

$$\psi_i := \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} I_{\{s^{(t)}=s_i\}},$$

where I is the indicator function for a logical statement such that $I_A = 1$ if A is true. The value of the vector ψ represents the proportion of the time the agent will spend in each state as $t \to \infty$ provided it follows the fixed policy π. In particular ψ_i indicates the proportion of the time the agent spends in the state s_i. As implied by its definition, ψ is a function of the policy π and the agent's starting state $s^{(1)}$ (i.e. its state at $t = 1$). In the case where a transition matrix obtained from π and P is irreducible and aperiodic, ψ will be the stationary distribution associated with π. None of the results in this paper relating to finite state spaces require that a transition matrix obtained from π and P be irreducible, however in order to avoid possible ambiguity, we will assume unless otherwise stated that ψ, whenever referred to, is the same for all $s^{(1)}$.

Perhaps the most natural, and also most commonly used, weighting coefficient $w(s_i, a_j)$ is $\psi(s_i)\pi(a_j|s_i)$, such that each error term is weighted in proportion to how frequently the particular state-action occurs (Menache et al., 2005; Yu and Bertsekas, 2009; Di Castro and Mannor, 2010). A slightly more general set of weightings is made up of those which satisfy $w(s_i, a_j) = \psi_i\tilde{w}(s_i, a_j)$, where $0 \le \tilde{w}(s_i, a_j) \le 1$ and $\sum_{j'=1}^{A}\tilde{w}(s_i, a_{j'}) \le 1$ for all i and j. All of our theoretical results will require that $w(s_i, a_j) = \psi_i\tilde{w}(s_i, a_j)$, and some will also require that $\tilde{w}(s_i, a_j) = \pi(a_j|s_i)$.[18]

1.5. A Motivating Discussion

The principle we are exploring in this article is that frequently visited states should have their values approximated with greater precision. Why would we employ such a strategy? There is a natural intuition which says that states which the agent is visiting frequently are

18. It is worth noting that weighting by ψ and π is not necessarily the only valid choice for w. It would be possible, for example, to set $w(s_i, a_j) = 1$ for all i and j depending on the purpose for which the scoring function has been defined.

more important, either because they are intrinsically more prevalent in the environment, or because the agent is behaving in a way that makes them more prevalent, and should therefore be more accurately represented.

However it may be possible to pinpoint reasons related to efficient algorithm design which might make us particularly interested in such approaches. The thinking behind unsupervised approaches from this perspective can be summarised (informally) in the set of points which we now outline. Our arguments are based principally around the objective of minimising VF error (we will focus our arguments on MSE, though similar points could be made in relation to L or $\tilde{L}$). We will note at the end of this section, however, circumstances under which the arguments will potentially translate to benefits where policies are updated as well.

It will be critical to our arguments that the scoring function is weighted by ψ. Accordingly we begin by assuming that, in measuring VF error using MSE, we adopt $w(s_i, a_j) = \psi_i\tilde{w}(s_i, a_j)$, where $\tilde{w}(s_i, a_j)$ obeys the constraints noted in the previous subsection, is stored by the algorithm, and is not a function of the environment (for example, $\tilde{w}(s_i, a_j) = \pi(a_j|s_i)$ or $\tilde{w}(s_i, a_j) = 1/A$ for all i and j). Now consider:

1. Our goal is to find an architecture which will permit us to generate a VF estimate with low error. We can see, referring to equation (1), that we have a sum of terms of the form:
$$\psi_i\tilde{w}(s_i,a_j)\left(Q^\pi(s_i,a_j)-\hat{Q}(s_i,a_j)\right)^2.$$
Suppose $\hat{Q}_{\text{MSE}}$ represents the value of $\hat{Q}$ for which MSE is minimised subject to the constraints of a particular architecture. Assuming we can obtain a VF estimate $\hat{Q} = \hat{Q}_{\text{MSE}}$ (e.g. using a standard RL algorithm), each term in equation (1) will be of the form:
$$\psi_i\tilde{w}(s_i,a_j)\left(Q^\pi(s_i,a_j)-\hat{Q}_{\text{MSE}}(s_i,a_j)\right)^2.$$
In order to reduce MSE we will want to focus on ensuring that our architecture avoids the occurrence of large terms of this form. A term may be large either because ψ_i is large, because $\tilde{w}(s_i, a_j)$ is large, or because $Q^\pi(s_i, a_j) - \hat{Q}_{\text{MSE}}(s_i, a_j)$ has large magnitude. It is likely that any adaptation method we propose will involve directly or indirectly sampling one or more of these quantities in order to generate an estimate which can then be provided as feedback to update the architecture. Since $\tilde{w}(s_i, a_j)$ is assumed to be already stored by the algorithm, we focus our attention on the other two factors.

2. Whilst both ψ and $Q^\pi - \hat{Q}_{\text{MSE}}$ influence the size of each term, in a range of important circumstances generating an accurate estimate of ψ will be easier and cheaper than generating an accurate estimate of $Q^\pi - \hat{Q}_{\text{MSE}}$. We would argue this for three reasons:

   (a) An estimate of $Q^\pi - \hat{Q}_{\text{MSE}}$ can only be generated with accuracy once an accurate estimate of $\hat{Q}_{\text{MSE}}$ exists. The latter will typically be generated by the underlying RL algorithm, and may require a substantial amount of training time to generate, particularly if γ is close to one;[19]

19. Whilst the underlying RL algorithm will store an estimate $\hat{Q}$ of Q^π, having an estimate of Q^π is not the same as having an estimate of $Q^\pi - \hat{Q}$. If we want to estimate $Q^\pi - \hat{Q}$, we should consider it

(b) The value $Q^\pi(s_i, a_j) - \hat{Q}_{\text{MSE}}(s_i, a_j)$ may *also* depend on trajectories followed by the agent consisting of many states and actions (again particularly if γ is near one), and it may take many sample trajectories and therefore a long training time to obtain a good estimate, even once $\hat{Q}_{\text{MSE}}$ is known;

(c) For each single value ψ_i there are A terms containing distinct values for $Q^\pi - \hat{Q}_{\text{MSE}}$ in the MSE. This suggests that ψ can be more quickly estimated in cases where $\tilde{w}(s_i, a_j) > 0$ for more than one index j. Furthermore, the space required to store an estimate, if required, is reduced by a factor of A.

3. If we accept that it is easier and quicker to estimate ψ than $Q^\pi - \hat{Q}_{\text{MSE}}$, we need to ask whether measuring the former and not the latter will provide us with sufficient information in order to make helpful adjustments to the approximation architecture. If ψ_i is roughly the same value for all $1 \leq i \leq S$, then our approach may not work. However in practice there are many environments which (in some cases subject to the policy) are such that there will be a large amount of variance in the terms of ψ, with the implication that ψ can provide critical feedback with respect to reducing MSE. This will be illustrated most clearly through examples in Section 3.5.

4. Finally, from a practical, implementation-oriented perspective we note that, for fixed π, the value $Q^\pi - \hat{Q}_{\text{MSE}}$ is a function of the approximation architecture. This is not the case for ψ. If we determine our approximation architecture with reference to $Q^\pi - \hat{Q}_{\text{MSE}}$, we may find it more difficult to ensure our adaptation method converges.[20] This could force us, for example, to employ a form of gradient descent (thereby, amongst other things,[21] limiting us to architectures expressible via differential parameters, and forcing architecture changes to occur gradually) or to make "once-and-for-all" changes to the approximation architecture (removing any subsequent capacity for our architecture to adapt, which is critical if we expect, under more general settings, the policy π to change with time).[22]

To summarise, there is the possibility that in many important instances visit probability loses little compared to other metrics when assessing the importance of an area of the VF, and the simplicity of unsupervised methods allows for fast calculation and flexible implementation.

The above points focus on the problem of policy evaluation. All of our arguments will extend, however, to the policy *learning* setting, provided that our third point above consistently holds as each update is made. Whether this is the case will depend primarily on

in general as being estimated from scratch. The distinction is explored, for example, from a gradient descent perspective in Baird (1995). See also Chapter 11 in Sutton and Barto (2018).

20. This is because we are likely to adjust the approximation architecture so that the approximation architecture is capable of more precision for state-action pairs where $Q^\pi(s_i, a_j) - \hat{Q}_{\text{MSE}}(s_i, a_j)$ is large. But, in doing this, we will presumably remove precision from other state-action pairs, resulting in $Q^\pi(s_i, a_j) - \hat{Q}_{\text{MSE}}(s_i, a_j)$ increasing for these pairs, which could then result in us re-adjusting the architecture to give more precision to these pairs. This could create cyclical behaviour.

21. Gradient descent using the Bellman error is also known to be slow to converge and may require additional computational resources (Baird, 1995).

22. As we saw in Section 1.2, most methods which use VF feedback explored to date in the literature do indeed employ one of these two approaches.

the type of environment with which the agent is interacting. This will be explored further in Section 3.5 and Section 4.

Having now discussed, albeit informally, some of the potential advantages of unsupervised approaches to adapting approximation architectures, we would now like to implement the ideas in an algorithm. This will let us test the ideas theoretically and empirically in a more precise, rigorous setting.

2. The PASA Algorithm

Our Probabilistic Adaptive State Aggregation (PASA) algorithm is designed to work in conjunction with SARSA (though certainly there may be potential to use it alongside other, similar, RL algorithms). In effect PASA provides a means of allowing a state aggregation approximation architecture to be adapted on-line. In order to describe in detail how the algorithm functions it will be helpful to initially provide a brief review of SARSA, and introduce some terminology relating to state aggregation approximation architectures.

2.1. SARSA with Fixed State Aggregation

In its *tabular* form SARSA[23] stores an $S \times A$ array $\hat{Q}(s_i, a_j)$. It performs an update to this array in each iteration as follows:

$$\hat{Q}^{(t+1)}\big(s^{(t)}, a^{(t)}\big) = \hat{Q}^{(t)}\big(s^{(t)}, a^{(t)}\big) + \Delta\hat{Q}^{(t)}\big(s^{(t)}, a^{(t)}\big),$$

where:[24]

$$\Delta\hat{Q}^{(t)}\big(s^{(t)}, a^{(t)}\big) = \eta\left(R\big(s^{(t)}, a^{(t)}\big) + \gamma\hat{Q}^{(t)}\big(s^{(t+1)}, a^{(t+1)}\big) - \hat{Q}^{(t)}\big(s^{(t)}, a^{(t)}\big)\right) \tag{2}$$

and where η is a fixed step size parameter.[25] In the tabular case, SARSA has some well known and helpful convergence properties (Bertsekas and Tsitsiklis, 1996).

It is possible to use different types of approximation architecture in conjunction with SARSA. *Parametrised value function approximation* involves generating an approximation of the VF using a parametrised set of functions. The approximate VF is denoted as $\hat{Q}_\theta$, and, assuming we are approximating over the state space only and not the action space, this function is parametrised by a matrix of weights θ of dimension $X \times A$ (where, by assumption, $X \ll S$). Such an approximation architecture is *linear* if $\hat{Q}_\theta$ can be expressed in the form $\hat{Q}_\theta(s_i, a_j) = \varphi(s_i, a_j)^T\theta_j$, where θ_j is the jth column of θ and $\varphi(s_i, a_j)$ is a fixed vector of dimension X for each pair (s_i, a_j). The XA distinct vectors of dimension S given by $(\varphi(s_1, a_j)_k, \varphi(s_2, a_j)_k, \ldots, \varphi(s_S, a_j)_k)$ are called *basis functions* (where $1 \leq k \leq X$). It is common to assume that $\varphi(s_i, a_j) = \varphi(s_i)$ for all j, in which case we have only X distinct

23. The SARSA algorithm (short for "state-action-reward-state-action") was first proposed by Rummery and Niranjan (1994). It has a more general formulation SARSA(λ) which incorporates an eligibility trace. Any reference here to SARSA should be interpreted as a reference to SARSA(0).

24. Note that, in equation (2), γ is a parameter of the algorithm, distinct from γ as used in the scoring function definitions. However there exists a correspondence between the two parameters which will be made clearer below.

25. In the literature, η is generally permitted to change over time, i.e. $\eta = \eta^{(t)}$. However throughout this article we assume η is a fixed value.

basis functions, and $\hat{Q}_\theta(s_i, a_j) = \varphi(s_i)^T \theta_j$. If we assume that the approximation architecture being adapted is linear then the method of adapting an approximation architecture is known as *basis function adaptation*. Hence we refer to the adaptation of a linear approximation architecture using an unsupervised approach as *unsupervised basis function adaptation*.

Suppose that Ξ is a partition of $\mathcal{S}$, containing X elements, where we refer to each element as a *cell*. Indexing the cells using k, where $1 \leq k \leq X$, we will denote as $\mathcal{X}_k$ the set of states in the kth cell. A *state aggregation* approximation architecture—see, for example, Singh et al. (1995) and Whiteson et al. (2007)—is a simple linear parametrised approximation architecture which can be defined using any such partition Ξ. The parametrised VF approximation is expressed in the following form: $\hat{Q}_\theta(s_i, a_j) = \sum_{k=1}^{X} I_{\{s_i \in \mathcal{X}_k\}} \theta_{kj}$.

SARSA can be extended to operate in conjunction with a state aggregation approximation architecture if we update θ in each iteration as follows:[26]

$$\theta_{kj}^{(t+1)} = \theta_{kj}^{(t)} + \eta I_{\{s^{(t)} \in \mathcal{X}_k\}} I_{\{a^{(t)} = a_j\}} \left(R\big(s^{(t)}, a^{(t)}\big) + \gamma d^{(t)} - \theta_{kj}^{(t)} \right), \tag{3}$$

where:

$$d^{(t)} := \sum_{k'=1}^{X} \sum_{j'=1}^{A} I_{\{s^{(t+1)} \in \mathcal{X}_{k'}\}} I_{\{a^{(t+1)} = a_{j'}\}} \theta_{k'j'}^{(t)}. \tag{4}$$

We will say that a state aggregation architecture is *fixed* if Ξ (which in general can be a function of t) is the same for all t. For convenience we will refer to SARSA with fixed state aggregation as SARSA-F. We will assume (unless we explicitly state that π is held fixed) that SARSA updates its policy by adopting the ϵ-greedy policy at each iteration t.

Given a fixed state aggregation approximation architecture, if π is held fixed then the value $\hat{Q}_\theta$ generated by SARSA can be shown to converge—this can be shown, for example, using much more general results from the theory of stochastic approximation algorithms.[27] If, on the other hand, we allow π to be updated, then this convergence guarantee begins to erode. In particular, any policy update method based on periodically switching to an ϵ-greedy policy will not, in general, converge. However, whilst the values $\hat{Q}_\theta$ and π generated by SARSA with fixed state aggregation may oscillate, they will remain bounded (Gordon, 1996, 2001).

2.2. The Principles of How PASA Works

PASA is an attempt to implement the idea of unsupervised basis function adaptation in a manner which is as simple and obvious as possible without compromising computational efficiency. The underlying idea of the algorithm is to make the VF representation comparatively detailed for frequently visited regions of the state space whilst allowing the representation to be coarser over the remainder of the state space. It will do this by progressively

26. This algorithm is a special case of a more general variant of the SARSA algorithm, one which employs stochastic semi-gradient descent and which can be applied to any set of linear basis functions.

27. This is examined more formally in Section 3. Note that the same is true for SARSA when used in conjunction with any linear approximation architecture. Approximation architectures which are *non-linear*, by way of contrast, cannot be guaranteed to converge even when a policy is held fixed, and may in fact diverge. Often the employment of a non-linear architecture will demand additional measures be taken to ensure stability—see, for example, Mnih et al. (2015). Given that the underlying approximation architecture is linear, unsupervised basis function adaptation methods typically do not require any such additional measures.

updating $\Xi = \Xi^{(t)}$. Whilst the partition Ξ progressively changes it will always contain a fixed number of cells X, and every cell will always be non-empty. We will refer to SARSA combined with PASA as SARSA-P (to distinguish it from SARSA-F described above).

The algorithm is set out in Algorithms 1 and 2. Before describing the precise details of the algorithm, however, we will attempt to describe informally how it works. PASA begins with an initial partition $\Xi^{(1)}$. This initial partition is, to some extent, arbitrary. PASA will update Ξ only infrequently. We must choose the value of a parameter $\nu \in \mathbb{N}$, which in practice will be large (for our experiments in Section 4 we choose $\nu = 50{,}000$). In each iteration t such that $t \bmod \nu = 0$, PASA will update Ξ, otherwise Ξ remains fixed. As will become clearer below, the reason for updating Ξ infrequently is to allow certain weight vectors, which are used to update Ξ, to be progressively adapted in between updates to Ξ. Every time Ξ is updated (which involves a sequence of steps *performed in a single iteration*) it describes a new, complete partition of the state space with X cells.

PASA updates Ξ, at the regular intervals defined by ν, as follows. Each time it in effect completely rebuilds the partition Ξ using a much less granular fixed partition Ξ_0 as a starting point. Specifically, for each update it begins with a fixed set of X_0 *base cells*, with $X_0 < X$, which together form a partition Ξ_0 of $\mathcal{S}$ (the partition Ξ_0 is identical for every periodic update). Suppose we have an estimate of how frequently the agent visits each of these base cells based on its recent behaviour. We can define a new partition Ξ_1 by "splitting" the most frequently visited cell into two cells containing a roughly equal number of states (the notion of a cell "split" is described more precisely below). If we now have a similar visit frequency estimate for each of the cells in the newly created partition, we could again split the most frequently visited cell giving us yet another partition Ξ_2. If we repeat this process a total of $X - X_0$ times (which PASA is designed to do all in the space of a single iteration, whenever $t \bmod \nu = 0$), then we will have generated a partition Ξ of the state space with X cells. Moreover, provided our visit frequency estimates are accurate, those areas of the state space which are visited more frequently will have a more detailed representation of the VF. (To be clear: Ξ_1, for example, denotes the partition generated after the first step of the process just described, whereas $\Xi^{(1)}$, for example, denotes the actual partition used by the SARSA algorithm at $t = 1$.)

For this process to work effectively, PASA needs to have access to an accurate estimate of the visit frequency for each cell for each stage of the splitting process. We could, at a first glance, provide this by storing an estimate of the visit frequency of every individual state. We could then estimate cell visit frequencies by summing the estimates for individual states as required. However S is, by assumption, very large, and storing S distinct real values is implicitly difficult or impossible. Accordingly, PASA instead stores an estimate of the visit frequency of each base cell, and an estimate of the visit frequency of *one* of the two cells defined each time a cell is split. This allows PASA to calculate an estimate of the visit frequency of every cell in every stage of the process described in the paragraph above whilst storing only X distinct values. It does this by subtracting certain estimates from others (a process described in more detail below). In this way we can estimate cell visit frequencies efficiently, at the cost of only a small trade off which we describe in Section 2.4 below.

2.3. Some Additional Terminology Relating to State Aggregation

In this subsection we will introduce some formal concepts, including the concept of "splitting" a cell, which will allow us, in the next subsection, to formally describe the PASA algorithm.

Our formalism is such that S is finite.[28] This means that, for any problem, we can arbitrarily index each state from 1 to S. Suppose we have a partition $\Xi_0 = \{\mathcal{X}_{j,0} : 1 \leq j \leq X_0\}$ defined on $\mathcal{S}$ with X_0 elements. We will say that the partition Ξ_0 is *ordered* if every cell $\mathcal{X}_{j,0}$ can be expressed as an interval of the form:

$$\mathcal{X}_{j,0} := \{s_i : L_{j,0} \leq i \leq U_{j,0}\},$$

where $L_{j,0}$ and $U_{j,0}$ are integers and $1 \leq L_{j,0} \leq U_{j,0} \leq S$. Starting with an ordered partition Ξ_0, we can formalise the notion of *splitting* one of its cells $\mathcal{X}_{j,0}$, via which we can create a new partition Ξ_1. The new partition $\Xi_1 = \{\mathcal{X}_{j',1} : 1 \leq j' \leq X_1\}$ will be such that:

$$\begin{aligned}
&X_1 = X_0 + 1 \\
&\mathcal{X}_{j,1} = \{s_i : L_{j,0} \leq i \leq L_{j,0} + \lfloor (U_{j,0} - L_{j,0} - 1)/2 \rfloor\} \\
&\mathcal{X}_{X_0+1,1} = \{s_i : L_{j,0} + \lfloor (U_{j,0} - L_{j,0} - 1)/2 \rfloor < i \leq U_{j,0}\} \\
&\mathcal{X}_{j',1} = \mathcal{X}_{j',0} \text{ for all } j' \in \{1, \ldots, j-1\} \cup \{j+1, \ldots, X_0\}
\end{aligned}$$

The effect is that we are splitting the interval associated with $\mathcal{X}_{j,0}$ as near to the "middle" of the cell as possible. This creates two new intervals, the "lower" interval replaces the existing cell, and the "upper" interval becomes a new cell (with index $X_0 + 1$). The new partition Ξ_1 is also an ordered partition. Note that the splitting procedure is only defined for cells with cardinality of two or more. For the remainder of this subsection our discussion will assume that this condition holds every time a particular cell is split. When we apply the procedure in practice we will take measures to ensure that, whenever a split occurs, this condition is always satisfied.

Starting with any initial ordered partition, we can recursively reapply this splitting procedure as many times as we like. Note that each time a split occurs, we specify the index of the cell we are splitting. This means, given an initial ordered partition Ξ_0 (with X_0 cells), we can specify a final partition Ξ_n (with $X_n = X_0 + n$ cells) by providing a vector ρ of integers, or *split vector*, which is of dimension n and is a list of the indices of cells to split. The split vector ρ must be such that, for each $1 \leq k \leq n$, the constraint $1 \leq \rho_k \leq X_0 + k - 1$ is satisfied (so that each element of ρ refers to a valid cell index). Assuming we want a partition composed of X cells exactly (i.e. so that $X_n = X$), then ρ must be of dimension $X - X_0$. Parts (a) and (b) of Figure 1 provide a partial illustration of how a pair of values ρ and Ξ_0 can be used to define a new, more granular partition.

Before proceeding we require one more definition. For each partition Ξ_k defined above, where $0 \leq k \leq n$, we introduce a collection of subsets of $\mathcal{S}$ denoted $\bar{\Xi}_k = \{\bar{\mathcal{X}}_j : 1 \leq j \leq X_0 + k\}$. Each element of $\bar{\Xi}_k$ is defined as follows:

$$\bar{\mathcal{X}}_j := \begin{cases} \{s_i : s_i \in \mathcal{X}_{j,0}\} & \text{if } 1 \leq j \leq X_0 \\ \{s_i : s_i \in \mathcal{X}_{j,j-X_0}\} & \text{if } X_0 < j \leq X \end{cases}$$

28. In the case of continuous state spaces we assume that we have a finite set of "atomic cells" which are analogous to the finite set of states discussed here. See Section 3.4.

Each set $\bar{\mathcal{X}}_j$ is not a function of k. However, for $j > X_0$ the value of $\bar{\mathcal{X}}_j$ will only be available after $j - X_0$ steps in the sequence described above. The effect of the definition is that, for $0 \leq j \leq X_0$, we simply have $\bar{\mathcal{X}}_j = \mathcal{X}_{j,0}$ for all j, whilst for $X_0 < j \leq X$, $\bar{\mathcal{X}}_j$ will contain all of the states which are contained in $\bar{\mathcal{X}}_{j,j-X_0}$, which is the first cell created during the overall splitting process which had an index of j, as it was before any additional splitting of that cell. (In other words, all the base cells can be referred to by $\bar{\mathcal{X}}_j$ for $1 \leq j \leq X_0$, and, for $j > X_0$, each time a cell $\mathcal{X}_{i,j-X_0-1}$ is split, the newly created cell $\mathcal{X}_{j,j-X_0}$ is equal to $\bar{\mathcal{X}}_j$, while $\bar{\mathcal{X}}_i$ remains unchanged and still retains its value from the first time it was created.) Note that $\bar{\Xi}_k$ is *not* a partition, with the single exception of $\bar{\Xi}_0$ which is equal to Ξ_0. The notation just outlined will be important when we set out the manner in which PASA estimates the frequency with which different cells are visited.

2.4. Details of the Algorithm

We now add the necessary final details to formally define the PASA algorithm. We assume we have a fixed ordered partition Ξ_0 containing X_0 cells. The manner in which Ξ_0 is constructed does not need to be prescribed as part of the PASA algorithm, however we assume $|\mathcal{X}_{j,0}| \geq 1$ for all $1 \leq j \leq X_0$. In general, therefore, Ξ_0 is a parameter of PASA.[29] PASA stores a split vector ρ of dimension $X - X_0$. This vector in combination with Ξ_0 defines a partition Ξ, which will represent the state aggregation architecture used by the underlying SARSA algorithm. Recall that we used $\mathcal{X}_j$ to denote a cell in a state aggregation architecture in Section 2.1. In the context of SARSA-P, $\mathcal{X}_j$ will be a function of t, and we will use the natural convention that $\mathcal{X}_j = \mathcal{X}_{j,X-X_0}$. We also adopt the notation $\bar{\Xi} := \bar{\Xi}_{X-X_0}$.

The vector ρ, and correspondingly the partition Ξ, will be updated every $\nu \in \mathbb{N}$ iterations, where ν (as noted above) is a fixed parameter. The interval defined by ν permits PASA to learn visit frequency estimates, which will be used when updating ρ. Subject to the constraint noted in the previous subsection, ρ can be initialised arbitrarily, however we assume that each ρ_k for $1 \leq k \leq X - X_0$ is initialised so that no attempt will be made to split a cell containing only one state (a *singleton* cell).

To assist in updating ρ, the algorithm will store a vector $\bar{u}$ of real values of dimension X (initialised as a vector of zeroes). We update $\bar{u}$ in each iteration as follows (i.e. using a simple stochastic approximation algorithm):

$$\bar{u}_j^{(t+1)} = \bar{u}_j^{(t)} + \varsigma \left(I_{\{s^{(t)} \in \bar{\mathcal{X}}_j\}} - \bar{u}_j^{(t)} \right), \tag{5}$$

where $\varsigma \in (0, 1]$ is a constant step size parameter. In this way, $\bar{u}$ will record the approximate frequency with which each of the sets in $\bar{\Xi}$ have been visited by the agent.[30] We also store an X dimensional boolean vector Σ. As will be made clearer below, Σ keeps track of whether a particular cell has only one state, as we don't want the algorithm to try to split singleton cells.

29. The reason we do not simply take $X_0 = 1$ is that taking $X_0 > 1$ can help to ensure that the values stored by PASA tend to remain more stable. In practice, it often makes sense to choose a suitable value for X_0, then simply take Ξ_0 to be the ordered partition consisting of X_0 cells which are as close as possible to equal size. See Section 4.

30. Hence, when estimating how frequently the agent visits certain sets of states, the PASA algorithm implicitly weights recent visits more heavily using a series of coefficients which decay geometrically. The rate of this decay depends on ς.

To update ρ the PASA algorithm, whenever $t \bmod \nu = 0$, performs a sequence of $X - X_0$ operations. A temporary copy of $\bar{u}$ is made, which we call u. The vector u is intended to estimate the approximate frequency with which each of the cells in Ξ have been visited by the agent. The elements of u will be updated as part of the sequence of operations which we will presently describe. We set the entries of Σ to $I_{\{|\mathcal{X}_{k,0}|=1\}}$ for $1 \leq k \leq X_0$ at the start of the sequence (the remaining entries can be set to zero). At each stage $k \in \{1, 2, \ldots, X - X_0\}$ of the sequence we update ρ as follows:

$$\rho_k = \begin{cases} j & \text{if } (1 - \Sigma_{\rho_k}) u_{\rho_k} < \max\{u_i : i \leq X_0 + k - 1, \Sigma_i = 0\} - \vartheta \\ \rho_k & \text{otherwise} \end{cases} \tag{6}$$

where:

$$j = \arg\max_i \{u_i : i \leq X_0 + k - 1, \Sigma_i = 0\}$$

(if multiple indices satisfy the arg max function, we take the lowest index) and where $\vartheta > 0$ is a constant designed to ensure that a (typically small) threshold must be exceeded before ρ is adjusted. In this way, in each step k in the sequence the non-singleton cell $\mathcal{X}_{j,k-1}$ with the highest value u_j (over the range $1 \leq j \leq k - 1$, and subject to the threshold ϑ) will be identified, via the update to ρ, as the next cell to split. In each step of the sequence we also update u and Σ:

$$u_{\rho_k} = u_{\rho_k} - u_{X_0+k}$$
$$\Sigma_j = I_{\{|\mathcal{X}_{j,k}| \leq 1\}} \text{ for } 1 \leq j \leq X_0 + k - 1.$$

The reason we update u as shown above is because each time the operation is applied we thereby obtain an estimate of the visit frequency of $\mathcal{X}_{\rho_k,k}$, which is the freshly updated value of u_{ρ_k}, and an estimate of the visit frequency of the cell $\mathcal{X}_{X_0+k,k}$, which is $u_{X_0+k} = \bar{u}_{X_0+k}$ (since $u_{X_0+k} = \bar{u}_{X_0+k}$ at step k). We note that, for each of the cell visit frequency estimates u_{ρ_k} and u_{X_0+k} to be accurate, it is critical that both the original estimates u_{ρ_k} and u_{X_0+k} are accurate. Crucial to the operation of the algorithm is the fact that this dependence only flows in one direction. As estimates for larger cells tend to become more accurate, estimates for smaller cells which are a function of the estimates for larger cells also become more accurate. As a result we can depend upon accurate estimates for larger cells flowing through to accurate estimates for smaller cells. We might ask why we do not, for example, estimate the visit frequency of a newly created cell by simply dividing the visit frequency for the parent cell in two. This makes sense the first time a split is generated (and indeed doing this the first time a cell is split represents a sensible extension of the algorithm, see Section A.5), however, adopting the method we have described permits us to obtain estimates which will be exact in the limit as the algorithm is given time to converge.

Once ρ has been generated, we implicitly have a new partition Ξ. The PASA algorithm is outlined in Algorithm 1 and a diagram illustrating the main steps is at Figure 1. Note that the algorithm calls a procedure called Split, which is outlined in Algorithm 2.[31] Algorithm 1 operates such that the cell splitting process (to generate Ξ) occurs concurrently with the update to ρ, such that, as each element ρ_k of ρ is updated, a corresponding temporary partition Ξ_k is constructed. Also note that the algorithm makes reference to objects Ξ' and $\bar{\Xi}'$. To avoid storing each Ξ_k and $\bar{\Xi}_k$ for $1 \leq k \leq X - X_0$, we instead initialise Ξ' and $\bar{\Xi}'$ as Ξ_0 and then recursively update Ξ' and $\bar{\Xi}'$ such that $\Xi' = \Xi_k$ and $\bar{\Xi}' = \bar{\Xi}_k$ at the kth stage of the splitting process.

31. Note that the pseudo code describes the underlying principles but is not an efficient implementation. The step to determine cell end points, for example, can be implemented in a far more efficient way than explicitly calculating a minimum or maximum. A full description of such details is beyond our present scope. See Section 3.1 for related details.

In Section 2.2 we noted that estimating the visit probability of individual cells by subtracting estimates from one another (as PASA is designed to do) allows us to avoid storing visit probabilities for S individual states. There is a trade-off involved when estimating visit frequencies in such a way. Suppose that $t = n\nu$ for some $n \in \mathbb{N}$ and the partition $\Xi^{(t)}$ is updated and replaced by the partition $\Xi^{(t+1)}$. The visit frequency estimate for a cell in $\Xi^{(t+1)}$ is only likely to be accurate if the same cell was an element of $\Xi^{(t)}$, or if the cell is a union of cells which were elements of $\Xi^{(t)}$. Cells in $\Xi^{(t+1)}$ which do not fall into one of these categories will need time for an accurate estimate of visit frequency to be obtained (this will be shown more clearly in Section 3.2). The consequence is that it may take longer for the algorithm to converge (assuming fixed π) than would be the case if an estimate of the visit frequency of every state were available. The negative impact of this trade-off in practice does not appear to be significant (see Section 4), however the trade-off is essential to ensure we have an efficient algorithm from the perspective of space complexity.

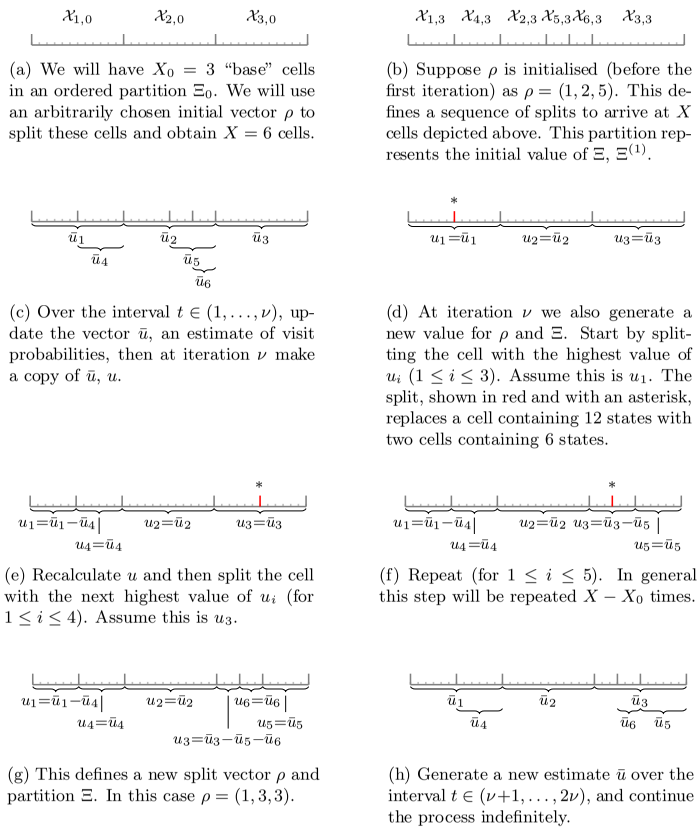
3. Theoretical Analysis

Having proposed the PASA algorithm we will now investigate some of its theoretical properties. With the exception of our discussion around complexity, these results will be constrained to problems of policy evaluation, where π is held fixed and the objective is to minimise VF error. A summary of all of the theoretical results can be found in Table 1.

3.1. Complexity of PASA

PASA requires only a very modest increase in computational resources compared to fixed state aggregation. In relation to time complexity, $\bar{u}$ can be updated in parallel with the SARSA algorithm's update of θ (and the update of $\bar{u}$ would not be expected to have any greater time complexity than the update to θ by SARSA, or indeed another standard RL algorithm such as Q-learning). Indeed, once a state has been mapped to a cell, the SARSA and PASA operations each have $O(1)$ time complexity in relation to S.

Assuming that we implement the mapping generated by Ξ using a tree-like structure—for details of such an approach, which can be applied equally to continuous and discrete state spaces, see, for example, Nouri and Littman (2009) or Munos and Moore (2002)—then the mapping from state to cell has, for discrete state spaces, a very low order of time complexity: $O(\log_2 S)$. In general the mapping from state to cell for a fixed partition will be of the same complexity, the only exception being when cells in the fixed partition are guaranteed to be above a certain size—for example we would have a minimum of $O(\log_2 X)$ for X fixed, equally-sized cells. The split vector ρ can be updated at intervals ν (and this update can also be run in parallel). In practice ν can be large because this allows time for $\bar{u}$ to converge. Hence, for all practical purposes, the increase in time complexity per iteration involved in the introduction of PASA is negligible.



(a) We will have $X_0 = 3$ "base" cells in an ordered partition Ξ_0. We will use an arbitrarily chosen initial vector ρ to split these cells and obtain $X = 6$ cells.

(b) Suppose ρ is initialised (before the first iteration) as $\rho = (1, 2, 5)$. This defines a sequence of splits to arrive at X cells depicted above. This partition represents the initial value of Ξ, $\Xi^{(1)}$.

(c) Over the interval $t \in (1, \ldots, \nu)$, update the vector $\bar{u}$, an estimate of visit probabilities, then at iteration ν make a copy of $\bar{u}$, u.

(d) At iteration ν we also generate a new value for ρ and Ξ. Start by splitting the cell with the highest value of u_i $(1 \leq i \leq 3)$. Assume this is u_1. The split, shown in red and with an asterisk, replaces a cell containing 12 states with two cells containing 6 states.

(e) Recalculate u and then split the cell with the next highest value of u_i (for $1 \leq i \leq 4$). Assume this is u_3.

(f) Repeat (for $1 \leq i \leq 5$). In general this step will be repeated $X - X_0$ times.

(g) This defines a new split vector ρ and partition Ξ. In this case $\rho = (1, 3, 3)$.

(h) Generate a new estimate $\bar{u}$ over the interval $t \in (\nu+1, \ldots, 2\nu)$, and continue the process indefinitely.

Figure 1: A simple example of the operation of the PASA algorithm, with $X = 6$, $X_0 = 3$ and $S = 36$. The horizontal line represents the set of all states, arranged in some arbitrary order, with the space between each short light grey tick representing a state (s_1 is the left-most state, with states arranged in order of their index up to s_{36}, the right-most state). The slightly longer vertical ticks represent the way in which states are distributed amongst cells: the space between two such longer ticks represents all of the states in a particular cell.

Algorithm 1 The PASA algorithm, called at each iteration t. We assume $\bar{u}$, ρ, Ξ_0, Ξ' and $\bar{\Xi}'$ are stored in memory. By definition $\bar{\Xi}_0 = \Xi_0$. The value of the partition Ξ' at the conclusion of the function call will be used at $t+1$ to determine the state aggregation approximation architecture employed by SARSA. The values ς, ϑ, and ν are constant parameters. Initialise c (a counter) at zero. Initialise each element of the vector $\bar{u}$ at zero. Denote $s = s^{(t)}$. Return is void.

function PASA(s)
  for $k \in \{1, \ldots, X\}$ **do**
    $\bar{u}_k \leftarrow \bar{u}_k + \varsigma(I_{\{s \in \bar{\mathcal{X}}_k\}} - \bar{u}_k)$ ▷ Update estimates of visit frequencies
  end for
  $c \leftarrow c + 1$
  if $c = \nu$ **then** ▷ Periodic updates to ρ and Ξ
    $c \leftarrow 0$
    $u \leftarrow \bar{u}$
    $\Xi' \leftarrow \Xi_0$
    $\bar{\Xi}' \leftarrow \Xi_0$
    for $k \in \{1, \ldots, X_0\}$ **do**
      $\Sigma_k \leftarrow I_{\{|\mathcal{X}_{k,0}|=1\}}$ ▷ Reset flags used to identify singular cells
    end for
    ▷ Iterate through sequence of cell splits
    for $k \in \{1, \ldots, X - X_0\}$ **do**
      ▷ Identify the cell with the highest visit probability estimate
      $u_{\max} \leftarrow \max\{u_i : i \leq X_0 + k - 1, \Sigma_i = 0\}$
      $i_{\max} \leftarrow \min\{i : i \leq X_0 + k - 1, u_i = u_{\max}, \Sigma_i = 0\}$
      ▷ If threshold ϑ is exceeded, update partition
      if $(1 - \Sigma_{\rho_k})u_{\rho_k} < u_{i_{\max}} - \vartheta$ **then**
        $\rho_k \leftarrow i_{\max}$ ▷ Reassign value for ρ
      end if
      $u_{\rho_k} \leftarrow u_{\rho_k} - u_{X_0+k}$ ▷ Update value of u
      $(\Xi', \bar{\Xi}', \Sigma) \leftarrow$ Split$(k, \rho_k, \Xi', \bar{\Xi}', \Sigma)$ ▷ Call function to split cell
    end for
  end if
end function

PASA does involve additional space complexity with respect to storing the vector $\bar{u}$: we must store X real values, such that PASA will have $O(X)$ space complexity when considered as a function of S. The values of Ξ and $\bar{\Xi}$ must also be stored, however Ξ must be stored for fixed state aggregation as well and, again when stored as a tree-like structure, each of Ξ and $\bar{\Xi}$ will only have $O(X)$ space complexity. Clearly PASA has space complexity of $O(1)$ when considered as a function of A (the number of actions available has no impact on the complexity or design of PASA). The SARSA component has space complexity of $O(X)$ when considered as a function of S, and space complexity of $O(A)$ when considered as a function of A (this reflects the $X \times A$ cell-action pairs). The implication is that the

Algorithm 2 Function to split selected cell in step k of sequence of cell splits. Called by PASA. The values Ξ'' and $\bar{\Xi}''$ are uninitialised and have $X_0 + k$ elements each. Use $\mathcal{X}'_j$, $\mathcal{X}''_j$, $\bar{\mathcal{X}}'_j$ and $\bar{\mathcal{X}}''_j$ to denote the jth element of Ξ', Ξ'', $\bar{\Xi}'$ and $\bar{\Xi}''$ respectively.

function Split(k,ρ_k,Ξ',$\bar{\Xi}'$,Σ)
  ▷ Set all but last element of Ξ'' and $\bar{\Xi}''$
  for $j \in \{1, \ldots, X_0 + k - 1\}$ **do**
    $\mathcal{X}''_j \leftarrow \mathcal{X}'_j$
    $\bar{\mathcal{X}}''_j \leftarrow \bar{\mathcal{X}}'_j$
  end for
  $L \leftarrow \min\{i : s_i \in \mathcal{X}_{\rho_k}\}$ ▷ Determine cell end points
  $U \leftarrow \max\{i : s_i \in \mathcal{X}_{\rho_k}\}$
  ▷ Update Ξ''
  $\mathcal{X}''_{\rho_k} \leftarrow \{s_i : L \leq i \leq L + \lfloor (U - L - 1)/2 \rfloor\}$
  $\mathcal{X}''_{X_0+k} \leftarrow \{s_i : L + \lfloor (U - L - 1)/2 \rfloor < i \leq U\}$
  ▷ Set last element of $\bar{\Xi}''$
  $\bar{\mathcal{X}}''_{X_0+k} \leftarrow \{s_j : L + \lfloor (U - L - 1)/2 \rfloor < i \leq U\}$ ▷ $\bar{\mathcal{X}}''_{\rho_k}$ does not change
  ▷ Identify singular cells (only need to update values for newly split cells)
  $\Sigma_{\rho_k} = I_{\{|\mathcal{X}''_{\rho_k}|=1\}}$
  $\Sigma_k = I_{\{|\mathcal{X}''_{X_0+k}|=1\}}$
  return $(\Xi'', \bar{\Xi}'', \Sigma)$ ▷ Return new partitions
end function

introduction of PASA as a pre-processing algorithm will not materially impact overall space requirements (in particular if A is large).

3.2. Convergence Properties of PASA

We would now like to consider some of the convergence properties of PASA. We will assume, for our next result, that π is held fixed for all t. There may be some potential to reformulate the result without relying on the assumption of fixed π, however our principle interest will be in this special case.

Our outline of PASA assumed a single fixed step size parameter ς. For our proof below it will be easier to suppose that we have a distinct fixed step size parameter ς_k for each element $\bar{u}_k$ of $\bar{u}$ ($1 \leq k \leq X$), each of which we can set to a different value (fixed as a function of t). For the remainder of this section ς should be understood as referring to this vector of step size parameters. We use $[1{:}k]$ to denote the set of indices from 1 to k, so that, for example, we can use $x_{[1:k]}$ to indicate a vector comprised of the first k elements of an arbitrary vector x. We will require some definitions.

Definition 1 *We will say that some function of t, $x = x^{(t)}$,* becomes ε-fixed over τ after T *provided T is such that, for all $T' > T$, the value x will remain the same for all t' satisfying $T' \leq t' \leq T' + \tau$ with probability at least $1 - \varepsilon$. We will similarly say, given a set of values $\mathcal{R}$, that x* becomes ε-fixed at $\mathcal{R}$ over τ after T *provided T is such that, for all $T' > T$, the value x is equal to a single element of $\mathcal{R}$ for all t' satisfying $T' \leq t' \leq T' + \tau$ with probability at least $1 - \varepsilon$.*

Definition 1 provides the sense in which we will prove the convergence of PASA in Proposition 4. It is weaker than most conventional definitions of convergence, however it will give us conditions under which we can ensure that ρ will remain stable (i.e. unchanging) for arbitrarily long intervals with arbitrarily high probability. This stability will allow us to call on well established results relating to the convergence of SARSA with fixed state aggregation. Such a convergence definition also permits a fixed step size parameter ς. This means, amongst other things, that the algorithm will "re-converge" if P or (more importantly) π change.

Definition 2 *We define $\mu_{j,k} := \sum_{i:s_i \in \mathcal{X}_{j,k}} \psi_i$. We also define $\tilde{\rho} = \tilde{\rho}(\pi)$ as the set of all split vectors which satisfy $\tilde{\rho}_k = j \Rightarrow \mu_{j,k-1} \geq \mu_{j',k-1}$ for all $1 \leq j' \leq X_0 + k - 1$ and all $1 \leq k \leq X - X_0$.*

The value $\mu_{j,k} = \mu_{j,k}(\pi)$ is the stable state probability of the agent visiting states in the cell $\mathcal{X}_{j,k}$ (assuming some policy π). This definition will be important as PASA progressively generates an estimate of this value, using the vectors u and $\bar{u}$. The set $\tilde{\rho}$ is the set of all split vectors which make the "correct" decision for each cell split (i.e. for some $\rho \in \tilde{\rho}$, the cell in Ξ_{k-1} with index ρ_k has the equal highest stable-state visit probability of all the cells in Ξ_{k-1} for $1 \leq k \leq X - X_0$). The set $\tilde{\rho}_{[1:k]}$ is defined such that $\rho_{[1:k]} \in \tilde{\rho}_{[1:k]}$ if and only if there exists $\rho' \in \tilde{\rho}$ such that $\rho'_{[1:k]} = \rho_{[1:k]}$. We require one final definition.

Definition 3 *If, for each $1 \leq i \leq X_0 + k$, and for all $\varepsilon > 0$, $h > 0$ and $\tau \in \mathbb{N}$, there exists $\varsigma_{[1:(X_0+k)]}$, H_i (a closed interval on the real line of length h which satisfies $\mu_{i,0} \in H_i$ for $i \leq X_0$ and $\mu_{i,i-X_0} \in H_i$ for $X_0 < i \leq X$) and T_i such that each*

$$I_i := I_{\{\bar{u}_i \in H_i\}}$$

is ε-fixed over τ after T_i we will say that $\bar{u}$ can be stabilised up to k. *Similarly if for all ε and τ there exists $\varsigma_{[1:(X_0+k-1)]}$, $\vartheta > 0$ and T such that $\rho_{[1:k]}$ is ε-fixed at $\tilde{\rho}_{[1:k]}$ over τ after T we will say that* ρ can be uniquely stabilised up to k.

The implications of this definition are as follows. If $\bar{u}$ can be stabilised up to k, it means we can find values $\varsigma_{[1:(X_0+k)]}$ (likely to be small) such that we can ensure that the value of each $\bar{u}_i$ for $1 \leq i \leq X_0 + k$ will, with high probability, eventually come to rest within a small interval H_i of the real line. Similarly, if ρ can be uniquely stabilised up to k, it implies that we can find values $\varsigma_{[1:(X_0+k-1)]}$ and ϑ (the value for ϑ is also likely to be small) so that $\rho_{[1:k]}$ will eventually remain at a fixed value with the properties of $\tilde{\rho}$ (i.e. such that each split is "correct") with high probability for a long period. The reason for introducing the index k into our definition is because our arguments below, wherein we establish that ρ will converge in the sense described in Definition 1, will depend on induction, and an interdependence between the elements of $\bar{u}$ and ρ.

Proposition 4 *For any particular instance of the pair (P, π), for every $\tau \in \mathbb{N}$ and $\varepsilon > 0$, there exists ς, $\vartheta > 0$ and T such that the vector $\rho^{(t)}$ generated by a PASA algorithm parametrised by ς and ϑ will become ε-fixed at $\tilde{\rho}$ over τ after T.*

Proof We will argue by induction. We will want to establish each of the following partial results, from which the result will follow: (1) $\bar{u}$ can be stabilised up to 0; (2) For $1 \leq k \leq X - X_0$, if $\bar{u}$ can be stabilised up to $k-1$, then ρ can be uniquely stabilised up to k; and, (3) For $1 \leq k \leq X - X_0$, if $\bar{u}$ can be stabilised up to $k-1$ then $\bar{u}$ can be stabilised up to k.

We begin with (1). We can argue this by relying on results regarding stochastic approximation algorithms with fixed step sizes. We rely on the following (much more general) result: Theorem 2.2 in Chapter 8 of Kushner and Yin (2003). We will apply the result to each $\bar{u}_i$ (for $1 \leq i \leq X_0$). The result requires that a number of assumptions hold (see Appendix A.1 where we state the assumptions and verify that they hold in this case) and states, in effect for our current purposes, the following.[32] For all $\delta > 0$, the fraction of iterations the value of $\bar{u}_i$ will stay within a δ-neighbourhood of the limit set of the ordinary differential equation (ODE) of the update algorithm for $\bar{u}_i$ over the interval $\{0, \ldots, T\}$ goes to one in probability as $\varsigma_i \to 0$ and $T \to \infty$. Recalling that, for all $1 \leq i \leq X_0$, $\bar{u}_i$ is initialised at zero, in our case the ODE is $\mu_{i,0}(1 - e^{-t})$ and the limit set is the point $\mu_{i,0}$ (this statement can be easily generalised to any initialisation of $\bar{u}_i$). The result therefore means that for any τ, h and ε we can, for each $\bar{u}_i$, choose T_i, ς_i and find $H_i \ni \mu_{i,0}$ such that I_i will be ε-fixed over τ after T_i, such that (1) holds.

We now look at (2). To see this holds, we elect arbitrary values τ' and ε' for which we need to find suitable values T', ϑ and $\varsigma'_{[1:(X_0+k-1)]}$. Suppose we set $0 < 2\vartheta \leq \min\{|\mu_{j,k'} - \mu_{j',k'}| : 0 \leq k' \leq k-1, 1 \leq j \leq X_0 + k', 1 \leq j' \leq X_0 + k', \mu_{j,k'} \neq \mu_{j',k'}\}$ (if $\mu_{j,k'}$ is the same for all j for all k', any value $\vartheta > 0$ can be chosen). Furthermore, using our assumption regarding $\bar{u}$, we select $h < \vartheta/2k$, noting that, if each $\bar{u}_i$ for $1 \leq i \leq k-1$ remains within an interval of size $\vartheta/2k$ for the set of iterations $T' \leq t \leq T'+\tau'$, then each u_i for $1 \leq i \leq X_0+k'$ (for each generated value of u in step k' of the sequence of updates of u for $k' \leq k-1$) will remain in an interval of length $\vartheta/2$ over the same set of iterations (since each u_i will be a set of additions of these values).

For any $k' \leq k-1$, define $i_{\max} := \arg\max_i\{\mu_{i,k'} : 1 \leq i \leq X_0 + k'\}$ (taking the lowest index if this is satisfied by more than one index). If, for any $1 \leq i' \leq X_0 + k'$, $\mu_{i',k'} \neq \mu_{i_{\max},k'}$, then, provided each $\bar{u}_i$ for $1 \leq i \leq X_0 + k'$ remains in an interval of length h over the iterations $T' \leq t \leq T' + \tau'$:

$$\begin{aligned} u^{(t)}_{i_{\max}} - u^{(t)}_{i'} &\geq \mu_{i_{\max},k'} - h(k'+1) - \big(\mu_{i',k'} + h(k'+1)\big) \\ &\geq 2\vartheta - 2h(k'+1) > \vartheta, \end{aligned}$$

for the value of u at step k' and for all t satisfying $T' \leq t \leq T' + \tau'$ so that $\rho_{k'} \in \tilde{\rho}_{k'}$ for $T' \leq t \leq T' + \tau'$ for all $1 \leq k' \leq k$, where $\tilde{\rho}_{k'+1}$ is the set of integers $\{m : \rho_{k'+1} = m, \rho \in \tilde{\rho}\}$.

Again for any $k' \leq k-1$, if, for some i', $\mu_{i_{\max},k'} = \mu_{i',k'}$, let's assume (without loss of generality) that $\rho^{(T')}_{k'+1} = i_{\max}$. We will have, again provided each $\bar{u}_i$ for $1 \leq i \leq X_0 + k'$ remains in an interval of length h over the iterations $T' \leq t \leq T' + \tau'$:

$$\begin{aligned} u^{(t)}_{i'} - u^{(t)}_{i_{\max}} &\leq \mu_{i',k'} + h(k'+1) - \big(\mu_{i_{\max},k'} - h(k'+1)\big) \\ &< \mu_{i',k'} + \frac{\vartheta}{2} - \left(\mu_{i_{\max},k'} - \frac{\vartheta}{2}\right) = \vartheta, \end{aligned}$$

32. The result is in fact stated with reference to a time-shifted interpolated trajectory of $\bar{u}_i$, where the trajectory of $\bar{u}_i$ is uniformly interpolated between each discrete point in the trajectory and then scaled by ς_i. The result as we state it in the body of the proof follows as a consequence.

for all t satisfying $T' \leq t \leq T'+\tau'$ which implies that $\rho_{k'+1}$ will not change for $T' \leq t \leq T'+\tau'$ for all $1 \leq k' \leq k$. As a result if we choose, from our assumed condition regarding $\bar{u}$, ε so that $(1-\varepsilon)^{X_0+k-1} \geq 1 - \varepsilon'$, and we choose $\tau = \tau' + \nu$ (to allow for the interval delay before ρ is updated), then (2) is satisfied, since we can choose $T' = \max_i T_i$ and $\varsigma'_{[1:(X_0+k-1)]} = \varsigma_{[1:(X_0+k-1)]}$ where $\varsigma_{[1:(X_0+k-1)]}$ and each T_i are chosen to satisfy our choices for ε and τ.

Finally, we examine (3). Suppose we choose the values τ, h and ε and must find suitable values ς_i, T_i and H_i (for $1 \leq i \leq X_0 + k$). By assumption, for each $\bar{u}_i$ for $1 \leq i \leq X_0 + k - 1$, we can find suitable values in order to satisfy the condition. However we also know, from the arguments in (2), that by selecting suitable values ε_1, h_1 and τ_1 to which our assumption regarding $\bar{u}_i$ for $1 \leq i \leq X_0 + k - 1$ applies, we can ensure, for any values of ε_2 and τ_2, that $\rho_{[1:k]}$ will be ε_2-fixed over τ_2 after T for some T. Note that if, for some i, I_i is ε-fixed over τ after T for some H_i of length h then I_i will also be ε'-fixed over τ' after T for some H'_i of length h' for any $\varepsilon' > \varepsilon$, $\tau' < \tau$ and $h' > h$. This last observation means that we can choose ε_0, h_0 and τ_0 so that $\varepsilon_0 < \varepsilon_1$, $\varepsilon_0 < \varepsilon$, $h_0 < h_1$, $h_0 < h$, $\tau_0 > \tau_1$ and $\tau_0 > \tau$, so that, for any ε_2, τ_2, ε, h and τ, we can find suitable values ς_i, T_i and H_i (for $1 \leq i \leq X_0 + k - 1$) so that all conditions are satisfied.

Again relying on the result from Kushner and Yin (2003), for any ε_3, h and τ there exists ς_{X_0+k}, H_{X_0+k} of length h and T'' which will ensure that $\bar{u}_{X_0+k}$ will remain in H_{X_0+k} with probability at least $1 - \varepsilon_3$ for all t such that $T'' \leq t \leq T'' + \tau$, provided the value of $\rho_{k'}$ for $1 \leq k' \leq k$ is held fixed for all $t \leq T'' + \tau$, for any starting value of $\bar{u}_k$ bounded by the interval $[-1, 1]$ (the limit set of the ODE is the same for all such starting values, and since the interval is compact we can choose the minimum value ς_{X_0+k} required to satisfy the condition for all starting values). Now, we can choose ε_2 and ε_3 such that $(1-\varepsilon) > (1-\varepsilon_2)(1-\varepsilon_3)$ and choose τ_2 such that $\tau_2 \geq T'' + \tau$. In this way, given the value of ς_{X_0+k} shown to exist above and $T_{X_0+k} \geq \max_i T_i + T''$ we will have the required values to ensure the necessary property also holds for $\bar{u}_{X_0+k}$. ■

Since ρ completely determines Ξ the result implies the convergence of Ξ to a partition $\Xi_{\lim}$. This fact means that SARSA-P will converge if π is held fixed (this is discussed in more detail below). Moreover a straightforward extension of the arguments in Gordon (2001) further implies that SARSA-P will not *diverge* even when π is updated. The result also implies that $\Xi_{\lim}$ will have the property that more frequently visited states will, in general, occupy smaller cells.

Note again that we have taken care to allow the vector ς to remain fixed as a function of t. This will, in principle, allow PASA to adapt to changes in π (assuming we allow π to change). We will use fixed step sizes in our experiments below. Whilst in our experiments in Section 4 we use only a single step size parameter (as opposed to a vector), the details of the proof above point to why there may be merit in using a vector of step size parameters as part of a more sophisticated implementation of the ideas underlying PASA (i.e. allowing ς_k to take on larger values for larger values of the index k, for $k > X_0$, may allow the algorithm to converge more rapidly).

The following related property of PASA will also be important for our subsequent analysis. The result gives conditions under which a guarantee can be provided that certain states

will occupy a singleton cell in $\Xi_{\lim}$. It will be helpful to define (i) as the index j which satisfies the condition $|\{k : \psi_k > \psi_j \text{ or } (\psi_k = \psi_j, k < j)\}| = i - 1$ (i.e. it is the index of the ith most frequently visited state, where we revert to the initial index ordering in case of equal values). Adopting this notation we therefore interpret, for example, $s_{(i)}$ as the ith most frequently visited state according to ψ. We continue to treat ς as a vector.

Proposition 5 *Suppose, given a particular instance of the pair (P, π) and a PASA algorithm parametrised by X, that, for some i satisfying $1 \leq i \leq S$, $X \geq i\lceil \log_2 S \rceil$ and $\psi_{(i)} > \sum_{j=i+1}^{S} \psi_{(j)}$. Then the mapping $\Xi_{\lim}$ obtained by applying Proposition 4 will be such that the state $s_{(i)}$ occupies a singleton cell.*

Proof Suppose that the conditions are satisfied for some index i but that the result does not hold. This implies that for at least one split in the sequence over $1 \leq l \leq X - X_0$ via which $\Xi_{\lim}$ is defined, PASA had the option to split a cell containing at least one state $s_{(i')}$ for $i' \leq i$, we will call this cell $\mathcal{X}_{k,l}$, and instead split an alternative cell $\mathcal{X}_{k',l}$ which contained no such state. This follows from the fact that $X \geq i\lceil \log_2 S \rceil$. As a result of our second assumption the cell $\mathcal{X}_{k,l}$ must be such that $\mu_{k,l} > \mu_{k',l}$. However this creates a contradiction to Proposition 4. ■

Finally, we turn to a further proposition, which applies to *sets* of pairs (P, π), and which will be important for our discussion in Section 3.5. A limitation of Proposition 4 is that, whilst we may be able to find parameters ϑ and ς to ensure that Ξ converges for a single pair (P, π), if, instead of a single pair (P, π), we are given an infinite set of pairs (we may, for example, only be given a prior distribution for P, which could be supported over a continuous range of values), then for certain pairs our chosen parameter values may not guarantee convergence in the sense we've defined.

We can address this by adjusting our arguments slightly, at the expense of some additional assumptions, and a slight (though relatively minor) weakening of the statement of the result. Appendix A.5 outlines some minor variants to the PASA algorithm (which are principally relevant to our experiments in Section 4). Crucially, for our next result, we need to assume that the PASA algorithm is altered in the sense indicated in points 3 and 4 of Appendix A.5. The effect of these changes will be that the order in which splits occur will not, for our current purposes, affect the final ordering of the cells in Ξ. This will permit us to guarantee, with the help of our next result, that Ξ will remain fixed, in a way which would be difficult otherwise. Note that these changes do not impact any of the discussion in Section 3.1, and are computationally very cheap to implement. Furthermore, all of the results stated in this article continue to hold (albeit with slightly adjusted proofs) in the presence of these changes. We need one more definition:

Definition 6 *Suppose that $\mathcal{Q}$ is a set and $\mathcal{R}$ is a set of subsets of $\mathcal{Q}$. Suppose also that $x = x^{(t)}$ and that $x^{(t)} \in \mathcal{Q}$ for all t. We will say that x* becomes ε-constrained at $\mathcal{R}$ over τ after T *provided T is such that, for all $T' > T$, with probability at least $1 - \varepsilon$ there is at least one element $r \in \mathcal{R}$ such that $x \in r$ for all t' satisfying $T' \leq t' \leq T' + \tau$.*

The definition allows us to loosen the concept of convergence slightly, such that we can define a set of subsets, and provide that a variable x will eventually remain in one of these

subsets for a long time with high probability. We don't know which subset the value will converge to in advance, and, once the value has converged to a subset, it may continue to change, however it will remain, with high probability, within that subset.

The proposition we now outline provides a guarantee that certain (high probability) states will occupy a singleton cell for all pairs (P, π) in a set, given some single pair of parameters ς and ϑ. Unlike Proposition 4, however, it does not guarantee that ρ will eventually be fixed at an element of $\tilde{\rho}$ per se. It instead guarantees that ρ becomes constrained to take a (possibly changing) value within a larger set of possible split vectors (containing $\tilde{\rho}$), characterised by certain states occupying singleton cells.

The result requires some further additional terminology which we set out within the result itself and its proof. The terminology is not used subsequently in this article.

Proposition 7 *Suppose that φ, ξ and κ are constants such that $\varphi > 0$, $\xi > 2\varphi$ and κ is a positive integer. Suppose also that $\mathcal{Z}$ is a set of pairs (P, π) each element of which is defined on the same set of states and actions. Suppose, finally, that for all pairs in $\mathcal{Z}$, there exists a subset of states $\mathcal{I}(P, \pi) \subset \mathcal{S}$ of size $|\mathcal{I}| \leq \kappa$ such that $\sum_{i: s_i \notin \mathcal{I}} \psi_i \leq \varphi$ and $s_i \in \mathcal{I} \Rightarrow \psi_i \geq \xi$. Then for all $\varepsilon > 0$ and $\tau \in \mathbb{N}$ there exists ς, ϑ and T such that, given a PASA algorithm (adjusted according to points 3 and 3 of Appendix A.5) with parameters ς, ϑ and $X \geq \kappa\lceil \log_2 S \rceil$, for all elements of $\mathcal{Z}$, ρ will become ε-constrained at $\hat{\rho}$ over τ after T, where $\hat{\rho} = \hat{\rho}(P, \pi)$ is a set of subsets of all possible split vectors, and where each element r of $\hat{\rho}$ is a set of split vectors where:*

1. *All split vectors in r correspond to the same final partition $\Xi(r)$; and,*
2. *The partition $\Xi(r)$ is such that each element of $\mathcal{I}$ occupies a singleton cell.*

Proof Begin by defining ρ^* as the set of split vectors which satisfy the constraint that, for $1 \leq l \leq X - X_0$, $\rho_l \in \{m : \mathcal{I} \cap \mathcal{X}_{m,l-1} \neq \emptyset, |\mathcal{X}_{m,l-1}| > 1\}$ whenever this last-mentioned set is non-empty (if it is empty, ρ_l has no constraints beyond those applicable to all split vectors). We then define $\hat{\rho}$ as the set of subsets defined if we partition ρ^* such that all elements of ρ^* which correspond to the same partition Ξ fall into the same element of $\hat{\rho}$. We can see that $\hat{\rho}$ will have the properties required by the result. Note that every split vector ρ can be mapped to a split vector $\rho_{[1:k]}$ by removing its last $X - X_0 - k$ values. If we map every split vector ρ to $\rho_{[1:k]}$ and preserve its membership of each subset of $\hat{\rho}$, we will have a new set of subsets of the set of all split vectors $\rho_{[1:k]}$, which we denote $\hat{\rho}_{[1:k]} = \hat{\rho}_{[1:k]}(P, \pi)$.

To obtain the result itself we proceed in much the same manner as Proposition 4. We first, however, introduce two new pieces of terminology. If $\bar{u}$ can be stabilised up to k for all elements of $\mathcal{Z}$ for the same values of $\varsigma_{[1:(X_0+k)]}$ and T_i for $1 \leq i \leq X_0 + k$, we will say that $\bar{u}$ *can be stabilised up to k for all $\mathcal{Z}$*. Furthermore, if for all ε and τ there exists T, ϑ and $\varsigma_{[1:(X_0+k-1)]}$ such that, for all elements of $\mathcal{Z}$, $\rho_{[1:k]}$ becomes ε-constrained at $\hat{\rho}_{[1:k]}$ over τ after T, we will say that ρ *can be effectively stabilised up to k for all $\mathcal{Z}$*.

We will argue by induction, relying on the following three claims: (1) $\bar{u}$ can be stabilised up to 0 for all $\mathcal{Z}$; (2) For $1 \leq k \leq X - X_0$, if $\bar{u}$ can be stabilised up to $k - 1$ for all $\mathcal{Z}$, then ρ can be effectively stabilised up to k for all $\mathcal{Z}$; and, (3) If $\bar{u}$ can be stabilised up to $k - 1$ for all $\mathcal{Z}$ then $\bar{u}$ can be stabilised up to k for all $\mathcal{Z}$. This will be enough to establish the result.

For (1), since the set of all possible values (P, π) is compact, and using an identical argument to that used in relation to statement (1) from the proof of Proposition 4, we can find $\varsigma_{[1:X_0]}$ and T_i for $1 \leq i \leq X_0$ such that $\bar{u}$ will be stabilised up to 0 for all elements of $\mathcal{Z}$.

For (2), suppose, for any $k' \leq k-1$, that $\mathcal{X}_{l,k'}$ is a cell which contains an element of $\mathcal{I}$, and $\mathcal{X}_{l',k'}$ and $\mathcal{X}_{l'',k'}$ are cells which contain no such element. We make two observations. First, for all $\mathcal{Z}$ over the iterations $T' \leq t \leq T' + \tau'$:

$$\begin{aligned} u_l^{(t)} - u_{l'}^{(t)} &\geq \sum_{i:s_i \in \mathcal{X}_{l,k'}} \psi_i - h(k'+1) - \left(\sum_{i:s_i \in \mathcal{X}_{l',k'}} \psi_i + h(k'+1) \right) \\ &\geq \mu_{l,k'} - hk - \big(\mu_{l',k'} + hk\big) \geq \xi - \varphi - 2hk. \end{aligned}$$

Second, again for all $\mathcal{Z}$ and over the iterations $T' \leq t \leq T' + \tau'$, suppose that $\Xi^{(T'-1)}$ contains at least one cell which is a strict subset of $\mathcal{X}_{l'',k'}$ (such that the split vector $\rho^{(T'-1)}$ splits $\mathcal{X}_{l'',k'}$ at some point):

$$\begin{aligned} u_{l'}^{(t)} - u_{l''}^{(t)} &\leq \mu_{l',k'} + h(k'+1) - \big(\mu_{l'',k'} - h(k'+1)\big) \\ &\leq \mu_{l',k'} + hk - \big(\mu_{l'',k'} - hk\big) \leq \varphi + 2hk. \end{aligned}$$

Now, since we can choose any value of h in relation to $\bar{u}_i$ for $1 \leq i \leq X_0 + k - 1$, and any value of ϑ, we choose values (which must exist) to satisfy the following inequality (recalling that ξ and φ are, by assumption, fixed for all elements of $\mathcal{Z}$, and that $\xi > 2\varphi$):

$$\varphi + 2hk \leq \vartheta < \xi - \varphi - 2hk.$$

Once this is done, we can see that, whenever at least one non-singleton cell is available containing a state in $\mathcal{I}$, it will be split before any cell *not* containing such a state is split. Furthermore, once all states in $\mathcal{I}$ occupy a singleton cell, the remaining splits which occur will remain unchanged each time Ξ is updated (though the order in which the splits occur may change). As a result, for any ε' and τ', we can choose h, ϑ, ε and τ in relation to our assumption regarding $\bar{u}$ so that h and ϑ satisfy our above assumption, so that $(1-\varepsilon)^{X_0+k-1} \geq 1 - \varepsilon'$, and so that $\tau = \tau' + \nu$. If the values $\varsigma_{[1:(X_0+k-1)]}$ and T_i for $1 \leq i \leq X_0 + k - 1$ are required to obtain h, ε and τ, then the values $T' = \max_i T_i$ and $\varsigma_{[1:(X_0+k-1)]}$ will obtain ε' and τ'. As a consequence ρ can be effectively stabilised up to k for all $\mathcal{Z}$, and (2) will hold.

Finally statement (3) follows in the same manner as statement (3) in Proposition 4, again using the compactness of the set of all possible values of (P, π) to ensure that we can find T, ς and ϑ such that the requirement will be satisfied for every element of $\mathcal{Z}$. ■

Implicitly, our interest in later sections will primarily be in cases where φ is small, and κ is relatively small compared to S, such that for each pair (P, π) a small subset of $\mathcal{S}$ has total visit probability close to one.

The implication of Proposition 7 is that, whilst ρ will converge to a set of values (and may alter within that set), since we have assumed that PASA is altered according to points 3 and 4 of Appendix A.5, the properties of $\hat{\rho}$ are such that the value Ξ will converge for all pairs in the set $\mathcal{Z}$, and in a way such that a set of states with high visit probability fall

into singleton cells. Note that, for states outside the set $\mathcal{I}$, many of our subsequent results will not actually require that Ξ converges with respect to its partitioning of these states. It seems likely that different arguments and slightly different assumptions could be made to obtain results which are not identical to, but carry many of the key implications of, the results as they are presented in this article.

3.3. Potential to Reduce Value Function Error Given Fixed π

The results in this subsection again apply to the case of policy evaluation only. In the tabular case, traditional RL algorithms such as SARSA provide a means of estimating Q^π which, assuming fixed π, will converge to the correct value as t becomes large. Once we introduce an approximation architecture, however, we no longer have any such guarantee, even if the estimate is known to converge. In relation to SARSA-F, there is little we can say which is non-trivial in relation to bounds on error in the VF estimate.[33]

Since the mapping generated by PASA converges to a fixed mapping (when suitably parametrised), then, assuming a fixed policy, if an RL algorithm such as SARSA is used to update $\hat{Q}_\theta$, the estimate $\hat{Q}_\theta$ will also converge (as noted in Section 2 above). If we know the convergence point we can then assess the limit of $\hat{Q}_\theta$ using an appropriate scoring function.

Our next result will provide bounds on the error in VF approximations generated in such a way. We will use PASA to guarantee that, under suitable conditions, a subset of the state space will be such that each element in that subset will fall into its own singleton cell. This has a powerful effect since, once this is the case, we can start to provide guarantees around the contribution to total VF error associated with those states.

The result will be stated in relation to the scoring functions defined in Section 1.4. We assume that the parameter γ selected for SARSA is the same as the parameter γ used to define each scoring function. We also assume SARSA has a fixed step size η. We will need four more definitions.

Definition 8 *For a particular transition function P and an arbitrary subset $\mathcal{I}$ of $\mathcal{S}$ define h (which is a function of π as well as of $\mathcal{I}$) as follows:*

$$h(\mathcal{I}, \pi) \coloneqq \sum_{i:s_i \in \mathcal{I}} \psi_i.$$

Note that $h(\mathcal{I}, \pi)$ is the proportion of time that the agent will spend in the subset $\mathcal{I}$ when following the policy π. It must take a value in the interval $[0, 1]$.

Definition 9 *For any $\delta \geq 0$, define (a) P, (b) R and (c) π respectively as being δ-*deterministic *if (a) P can expressed as follows: $P = (1-\delta)P_1 + \delta P_2$, where P_1 is a deterministic transition function and P_2 is an arbitrary transition function, (b) for all s_i and a_j, $\mathrm{Var}(R(s_i, a_j)) \leq \delta$, and (c) π can expressed as follows: $\pi = (1-\delta)\pi_1 + \delta\pi_2$, where π_1 is a deterministic policy and π_2 is an arbitrary policy.*

33. A result does exist which provides a bound on the extent to which the error in the VF estimate generated by SARSA-F exceeds the minimum possible error $\min_\theta \mathrm{MSE}(\theta)$ for all possible values of the matrix θ. See Bertsekas and Tsitsiklis (1996). Since SARSA generates its estimates using temporal differences, it will not generally attain this minimum. However, there is still little that we can say about the magnitude of $\min_\theta \mathrm{MSE}(\theta)$.

The three parts of the definition mean that, as δ moves closer to zero, a δ-deterministic transition function, a δ-deterministic reward function and a δ-deterministic policy respectively become "more deterministic".

Definition 10 *For a particular pair (P, π) and any $\delta \geq 0$, we define $\mathcal{I}$ as being* δ-constrained *provided that, for every state s_i in $\mathcal{I}$, $\sum_{j=1}^{A} \pi(a_j|s_i) \sum_{i':s_{i'} \notin \mathcal{I}} P(s_{i'}|s_i, a_j) \leq \delta$.*

This definition means that, for the pair (P, π), if $\mathcal{I}$ is δ-constrained, the agent will transition from a state in $\mathcal{I}$ to a state outside $\mathcal{I}$ with probability no greater than δ.

Definition 11 *For a particular triple (P, R, π) we also define the four values (a) δ_P, (b) δ_R, (c) δ_π and (d) $\delta_\mathcal{I}$ as (a) $\min\{\delta : P$ is δ-deterministic$\}$, (b) $\min\{\delta : R$ is δ-deterministic$\}$, (c) $\min\{\delta : \pi$ is δ-deterministic$\}$ and (d) $\min\{\delta : \mathcal{I}$ is δ-constrained$\}$ respectively.*

Each of δ_P, δ_R, δ_π and $\delta_\mathcal{I}$ must fall on the interval $[0, 1]$. Note that, for any $\mathcal{I}$ satisfying $\min\{\psi_i : s_i \in \mathcal{I}\} > 1 - h(\mathcal{I}, \pi)$, for any $\delta \geq 0$ there will exist $\delta' \geq 0$ such that if P and π are both δ'-deterministic, then $\mathcal{I}$ must be δ-constrained (since for any $\delta < 1$, if $\delta' = 0$, and $\mathcal{I}$ is not δ-constrained, then, for some i and i', where $s_{i'} \notin \mathcal{I}$ and $s_i \in \mathcal{I}$, s_i must deterministically transition to $s_{i'}$, and hence $\psi_{i'} \geq \psi_i$, contradicting our just-stated assumption with respect to $\mathcal{I}$).

Theorem 12 *For a particular instance of the triple (P, R, π), suppose that $|R(s_i, a_j)|$ is bounded for all i and j and that the constant R_m denotes the maximum of $|\mathrm{E}(R(s_i, a_j))|$ for all i and j. Take any subset $\mathcal{I}$ of $\mathcal{S}$. If $X \geq |\mathcal{I}|\lceil \log_2 S \rceil$ and $\min\{\psi_i : s_i \in \mathcal{I}\} > 1 - h(\mathcal{I}, \pi)$ then, for all $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ there exists T, η, ϑ and a parameter vector ς such that, provided $t \geq T$, with probability equal to or greater than $1 - \varepsilon_2$ the VF estimate $\hat{Q}^{(t)}$ generated by SARSA-P will be such that:*

1. *If $w(s_i, a_j) = \psi_i \pi(a_j|s_i)$ then:*

$$\mathrm{MSE} \leq \left(2(1-h) + \delta_\mathcal{I} + \frac{\delta_\mathcal{I}^2 \gamma^2}{1-\gamma} + \frac{\delta_\mathcal{I}^2 \gamma^4}{(1-\gamma)^2}\right) \frac{2R_\mathrm{m}^2}{(1-\gamma)^2} + \varepsilon_1;$$

2. *If $w(s_i, a_j) = \psi_i \tilde{w}(s_i, a_j)$ for an arbitrary function $\tilde{w}$ satisfying $0 \leq \tilde{w}(s_i, a_j) \leq 1$ and $\sum_{j'=1}^{A} \tilde{w}(s_i, a_{j'}) \leq 1$ for all i and j then:*

$$L \leq \frac{4(1-h)}{(1-\gamma)^2} R_\mathrm{m}^2 + \varepsilon_1;$$

and:

$$\tilde{L} \leq \left(4(1-h) + \gamma^2 \left(1 + 2(1-\delta_P)(1-\delta_\pi) - 3(1-\delta_P)^2(1-\delta_\pi)^2\right)\right) \frac{R_\mathrm{m}^2}{(1-\gamma)^2} + \delta_R + \varepsilon_1.$$

Remark 13 *The bounds just stated generally only become important when h is close to 1 (i.e. the agent spends a large proportion of its time in a subset $\mathcal{I}$ of $\mathcal{S}$ which satisfies the theorem's conditions), and (for* MSE *and $\tilde{L}$) when the environment and policy are "close to deterministic". In the case when h approaches 1 and the environment and policy are arbitrarily close to deterministic, all bounds become arbitrarily close to zero.*

Proof Propositions 4 and 5 mean we can guarantee that T', ϑ and ς exist such that, for any τ and $\varepsilon_2' > 0$, Ξ will be (a) fixed for τ iterations with probability at least $1 - \varepsilon_2'$ and (b) that Ξ will be such that each element of $\mathcal{I}$ will be in a singleton cell.

Our assumption with respect to R, as well as standard results relating to stochastic approximation algorithms—see Kushner and Yin (2003) and our brief discussion in Appendix A.1—allow us to guarantee for SARSA with fixed state aggregation that, for any $\varepsilon_1' > 0$ and $\varepsilon_2'' > 0$, there exists τ and η such that, provided we have a fixed partition Ξ' for the interval τ, $\hat{Q}(s_i, a_j)$ will be within ε_1' of a convergence point $\hat{Q}_\mathrm{lim}(s_i, a_j)$ associated with Ξ' for every i and j with probability at least $1 - \varepsilon_2''$.

This means that, for any ε_2 and ε_1', by choosing ε_2' and ε_2'' so that $(1-\varepsilon_2')(1-\varepsilon_2'') > 1 - \varepsilon_2$ we can find T (and η, ϑ and ς) such that, for all $t > T$, $|\hat{Q}(s_i, a_j) - \hat{Q}_\mathrm{lim}(s_i, a_j)| \leq \varepsilon_1'$ for all i and j with probability at least $1 - \varepsilon_2$, where $\hat{Q}_\mathrm{lim}$ is the limit point associated with the partition Ξ described above. We will use this fact in our proof of each of the three inequalities. We will also use the fact that, in general, for any state s_i in a singleton cell, and for each a_j, for $\hat{Q}_\mathrm{lim}$ we will have:

$$\begin{aligned}
&\mathrm{E}\left(\Delta\hat{Q}_\mathrm{lim}(s_i, a_j) \middle| s^{(t')} = s_i, a^{(t')} = a_j\right) \\
&\quad = \mathrm{E}\left(R(s_i, a_j) + \gamma \hat{Q}_\mathrm{lim}\left(s^{(t'+1)}, a^{(t'+1)}\right) - \hat{Q}_\mathrm{lim}(s_i, a_j) \middle| s^{(t')} = s_i, a^{(t')} = a_j\right) = 0,
\end{aligned}$$

where $\Delta\hat{Q}_\mathrm{lim}(s_i, a_j)$ is equal to $\hat{Q}^{(t'+1)}(s_i, a_j) - \hat{Q}^{(t')}(s_i, a_j)$ assuming $\hat{Q}^{(t')} = \hat{Q}_\mathrm{lim}$,[34] from which we can infer that:

$$\hat{Q}_\mathrm{lim}(s_i, a_j) = \mathrm{E}\left(R(s_i, a_j) + \gamma \hat{Q}_\mathrm{lim}\left(s^{(t'+1)}, a^{(t'+1)}\right) \middle| s^{(t')} = s_i, a^{(t')} = a_j\right).$$

For L, if $|\hat{Q}(s_i, a_j) - \hat{Q}_\mathrm{lim}(s_i, a_j)| \leq \varepsilon_1'$ for every i and j with probability at least $1 - \varepsilon_2$, then the equation above immediately implies that each term in L corresponding to a state in $\mathcal{I}$ will be less than $4\varepsilon_1'^2$ with probability of at least $1 - \varepsilon_2$. Furthermore, for states outside the set $\mathcal{I}$, the maximum possible value of each such term is:

$$\left(\sum_{t=1}^{\infty} \gamma^{t-1} 2R_\mathrm{m} + 2\varepsilon_1'\right)^2 \leq \frac{4R_\mathrm{m}^2}{(1-\gamma)^2} + \varepsilon_1'',$$

where for any $\varepsilon_1'' > 0$ there exists $\varepsilon_1' > 0$ so that the inequality is satisfied. By selecting ε_1' and ε_1'' such that $4\varepsilon_1'^2 + \varepsilon_1'' < \varepsilon_1$ this gives us the result for L (where we ignore the factor of h in relation to terms for states in $\mathcal{I}$, since this factor will only decrease the bound given that $h \leq 1$).

For $\tilde{L}$ we will use the temporary notation $\lambda \coloneqq (1-\delta_P)(1-\delta_\pi)$. Again, suppose that, with probability at least $1 - \varepsilon_2$ we have $|\hat{Q}(s_i, a_j) - \hat{Q}_\mathrm{lim}(s_i, a_j)| \leq \varepsilon_1'$ for every i and j. In the next equation, for notational convenience, we refer to $R(s_i, a_j)$ as $\tilde{R}$ and $\hat{Q}_\mathrm{lim}(s^{(t'+1)}, a^{(t'+1)})$ (which is a random variable conditioned on $s^{(t')} = s_i$ and $a^{(t')} = a_j$) as $\tilde{Q}'_\mathrm{lim}$. Given a state $s^{(t')} = s_i$ occupying a singleton cell, we will have the following, with probability at least

34. Note that it is straightforward to generalise here such that the implicit assumption of ergodicity is not required, however these details are omitted.

$1-\varepsilon_2$, for each a_j:

$$\begin{aligned}
&\mathrm{E}\Big(\tilde{R}+\gamma\hat{Q}\big(s^{(t'+1)},a^{(t'+1)}\big)-\hat{Q}(s_i,a_j)\Big)^2\\
&\quad\le \mathrm{E}\left(\tilde{R}+\gamma\tilde{Q}'_{\lim}-\mathrm{E}\big(\tilde{R}+\gamma\tilde{Q}'_{\lim}\big)+2\varepsilon'_1\right)^2\\
&\quad= \mathrm{E}\left(\big(\tilde{R}+\gamma\tilde{Q}'_{\lim}\big)^2\right)-\left(\mathrm{E}\big(\tilde{R}+\gamma\tilde{Q}'_{\lim}\big)\right)^2+4\varepsilon_1'^2\\
&\quad= \mathrm{E}\big(\tilde{R}^2\big)+2\gamma\mathrm{E}\big(\tilde{R}\tilde{Q}'_{\lim}\big)+\gamma^2\mathrm{E}\big(\tilde{Q}'^2_{\lim}\big)\\
&\qquad\quad-\big(\mathrm{E}(\tilde{R})\big)^2-2\gamma\mathrm{E}(\tilde{R})\mathrm{E}\big(\tilde{Q}'_{\lim}\big)-\gamma^2\left(\mathrm{E}\big(\tilde{Q}'_{\lim}\big)\right)^2+4\varepsilon_1'^2\\
&\quad= \mathrm{E}\big(\tilde{R}^2\big)-(\mathrm{E}(\tilde{R}))^2+\gamma^2\mathrm{E}\big(\tilde{Q}'_{\lim}\big)^2-\gamma^2\left(\mathrm{E}\big(\tilde{Q}'_{\lim}\big)\right)^2+4\varepsilon_1'^2,
\end{aligned}\tag{7}$$

where we've used the independence of R and P and of R and π. We can see that the first two terms together equal the variance of $R(s_i,a_j)$ and so are bounded above by δ_R. Suppose that $s_{i''}$ and $a_{j''}$ are the state-action pair corresponding to the deterministic transition and deterministic action following from the state and action s_i and a_j, i.e. according to the transition function P_1 and policy π_1 (which exist and are defined according to the definition of δ-deterministic for P and π respectively). We can then also expand the third and fourth terms in the final line of equation (7), temporarily omitting the factor of γ^2, to obtain:

$$\begin{aligned}
&\sum_{i'=1}^{S}\sum_{j'=1}^{A}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\hat{Q}_{\lim}(s_{i'},a_{j'})^2\\
&\qquad-\left(\sum_{i'=1}^{S}\sum_{j'=1}^{A}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\hat{Q}_{\lim}(s_{i'},a_{j'})\right)^2\\
&=\lambda\hat{Q}_{\lim}(s_{i''},a_{j''})^2+\sum\nolimits_{\Omega}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\hat{Q}_{\lim}(s_{i'},a_{j'})^2\\
&\qquad-\left(\lambda\hat{Q}_{\lim}(s_{i''},a_{j''})+\sum\nolimits_{\Omega}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\hat{Q}_{\lim}(s_{i'},a_{j'})\right)^2,
\end{aligned}$$

where $\Omega\coloneqq\{(i',j'):i'\neq i''\text{ or }j'\neq j''\}$.

Expanding relevant terms, and noting that $\sum_{\Omega}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\le 1-\lambda$, our statement becomes:

$$\begin{aligned}
&\lambda\hat{Q}_{\lim}(s_{i''},a_{j''})^2+\sum\nolimits_{\Omega}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\hat{Q}_{\lim}(s_{i'},a_{j'})^2-\lambda^2\hat{Q}_{\lim}(s_{i''},a_{j''})^2\\
&\qquad+2\lambda\hat{Q}_{\lim}(s_{i''},a_{j''})\sum\nolimits_{\Omega}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\hat{Q}_{\lim}(s_{i'},a_{j'})\\
&\qquad-\left(\sum\nolimits_{\Omega}P(s_{i'}|s_i,a_j)\pi(a_{j'}|s_{i'})\hat{Q}_{\lim}(s_{i'},a_{j'})\right)^2\\
&\le\left(\lambda-\lambda^2\right)\frac{R_{\mathrm{m}}^2}{(1-\gamma)^2}+(1-\lambda)\frac{R_{\mathrm{m}}^2}{(1-\gamma)^2}+2\lambda(1-\lambda)\frac{R_{\mathrm{m}}^2}{(1-\gamma)^2}=\left(1+2\lambda-3\lambda^2\right)\frac{R_{\mathrm{m}}^2}{(1-\gamma)^2}\\
&=\left(1+2(1-\delta_P)(1-\delta_\pi)-3(1-\delta_P)^2(1-\delta_\pi)^2\right)\frac{R_{\mathrm{m}}^2}{(1-\gamma)^2}\coloneqq D,
\end{aligned}$$

where we've used the fact that $R_{\mathrm{m}}/(1-\gamma)$ is an upper bound on the magnitude of $\hat{Q}_{\lim}(s_i,a_j)$ for all i and j, and where we ignore the last term in the first statement since its contribution must be less than zero. For those states not in $\mathcal{I}$ we argue in exactly the same fashion as for L, which gives us:

$$\tilde{L}\le\frac{4(1-h)}{(1-\gamma)^2}R_{\mathrm{m}}^2+4\varepsilon_1'^2+\varepsilon_1''+\gamma^2D+\delta_R$$

with probability at least $1-\varepsilon_2$. Accordingly, for any ε_1 we can select suitable ε'_1, ε''_1 so that the result is satisfied.

For MSE, we will decompose both Q^π and $\hat{Q}_{\lim}$ into different sets of sequences of states and actions. In particular, we will isolate the set of all finite sequences of states and actions which remain within the set $\mathcal{I}$, starting from a state in $\mathcal{I}$, up until the agent transitions to a state outside $\mathcal{I}$. For a single state $s_i\in\mathcal{I}$, and for all a_j, we will have:

$$\begin{aligned}
Q^\pi(s_i,a_j)=&\underbrace{\xi^{(1)}+\sum_{t'=2}^{\infty}\Pr\left(s^{(t'')}\in\mathcal{I}\text{ for }t''\le t'\middle|s^{(1)}=s_i,a^{(1)}=a_j\right)\xi^{(t')}}_{=:C}\\
&+\underbrace{\Pr\left(s^{(2)}\notin\mathcal{I}\middle|s^{(1)}=s_i,a^{(1)}=a_j\right)x^{(2)}}_{=:U}\\
&+\underbrace{\sum_{t'=3}^{\infty}\Pr\left(s^{(t'')}\in\mathcal{I}\text{ for }t''<t',s^{(t')}\notin\mathcal{I}\middle|s^{(1)}=s_i,a^{(1)}=a_j\right)x^{(t')}}_{=:V},
\end{aligned}$$

where $\xi^{(t')}$ is the expected reward at $t=t'$, conditioned upon $s^{(1)}=s_i$ and $a^{(1)}=a_j$, and conditioned upon the agent remaining within the set $\mathcal{I}$ for all iterations up to and including t'. The value $x^{(t')}$ is an expected discounted reward summed over all iterations following (and including) the first iteration t' for which the agent's state is no longer within the set $\mathcal{I}$ (each value is also conditioned upon $s^{(1)}=s_i$ and $a^{(1)}=a_j$). Each such value represents, in effect, a residual difference between the sum of terms involving ξ and $Q^\pi(s_i,a_j)$, and will be referred to below. It is for technical reasons that we separate out the term representing the case where $s^{(2)}\notin\mathcal{I}$. The reasons relate to the weighting $w(s_i,a_j)=\psi_i\pi(a_j|s_i)$ and will become clearer below. We will also have (by iterating the formula for $\hat{Q}_{\lim}$):

$$\begin{aligned}
\hat{Q}_{\lim}(s_i,a_j)=C&+\underbrace{\Pr\left(s^{(2)}\notin\mathcal{I}\middle|s^{(1)}=s_i,a^{(1)}=a_j\right)x'^{(2)}}_{=:U'}\\
&+\underbrace{\sum_{t'=3}^{\infty}\Pr\left(s^{(t'')}\in\mathcal{I}\text{ for }t''<t',s^{(t')}\notin\mathcal{I}\middle|s^{(1)}=s_i,a^{(1)}=a_j\right)x'^{(t')}}_{=:V'},
\end{aligned}$$

where each $x'^{(t')}$ similarly represents part of the residual difference between the summation over terms involving ξ and $\hat{Q}_{\lim}(s_i,a_j)$. We once again are permitted to assume, for sufficiently large t, that for any ε'_1 and ε_2 we can obtain $|\hat{Q}^{(t)}(s_i,a_j)-\hat{Q}_{\lim}(s_i,a_j)|\le\varepsilon'_1$ for all i and j with probability at least ε_2.

Again, we consider states inside and outside $\mathcal{I}$ separately and argue in exactly the same fashion as for L and $\tilde{L}$. This will leave us with (noting that the two values C, associated with Q^π and $\hat{Q}_{\text{lim}}$ respectively, will cancel out):

$$\begin{aligned}\text{MSE} &\leq \frac{4(1-h)R_{\text{m}}^2}{(1-\gamma)^2} + \varepsilon_1'' + \sum_{i:s_i\in\mathcal{I}} \psi_i \sum_{j=1}^{A} \pi(a_j|s_i)(U+V-U'-V'+\varepsilon_1')^2 \\ &= \frac{4(1-h)R_{\text{m}}^2}{(1-\gamma)^2} + \varepsilon_1'' \\ &\quad + \sum_{i:s_i\in\mathcal{I}} \psi_i \sum_{j=1}^{A} \pi(a_j|s_i)(U^2+V^2+U'^2+V'^2+\varepsilon_1'^2+UV-UU'-UV'+\ldots),\end{aligned}$$

where we have abbreviated the final line, omitting most of the terms in the expansion of the squared summand. We will derive bounds in relation to U^2, V^2 and UV. Similar bounds can be derived for all other terms (in a more-or-less identical manner, the details of which we omit) to obtain the result. First we examine U^2. Note that every value $|x^{(t')}|$ and value $|x'^{(t')}|$ is bound by $R_{\text{m}}^2/(1-\gamma)^2$. We have, for each $s_i \in \mathcal{I}$:

$$\sum_{j=1}^{A} \pi(a_j|s_i)U^2 \leq \frac{R_{\text{m}}^2}{(1-\gamma)^2} \sum_{j=1}^{A} \pi(a_j|s_i)\Pr\left(s^{(2)} \notin \mathcal{I} \middle| s^{(1)} = s_i, a^{(1)} = a_j\right) \leq \delta_{\mathcal{I}} \frac{R_{\text{m}}^2}{(1-\gamma)^2}.$$

(Note, in the inequality just stated, the importance of the weighting π.)

For V^2 we have:

$$\begin{aligned}V^2 &= \left(\sum_{t'=3}^{\infty} \Pr\left(s^{(t'')} \in \mathcal{I} \text{ for } t'' < t', s^{(t')} \notin \mathcal{I} \middle| s^{(1)} = s_i, a^{(1)} = a_j\right) x^{(t')}\right)^2 \\ &\leq \left(\sum_{t'=3}^{\infty} (1-\delta_{\mathcal{I}})^{t'-2}\delta_{\mathcal{I}}\gamma^{t'-1} \sum_{u=t'}^{\infty} \gamma^{u-t'} R_{\text{m}}\right)^2 \leq \left(\frac{\delta_{\mathcal{I}}(1-\delta_{\mathcal{I}})\gamma^2}{1-(1-\delta_{\mathcal{I}})\gamma}\right)^2 \frac{R_{\text{m}}^2}{(1-\gamma)^2}.\end{aligned}$$

Similar arguments will yield:

$$|UV| \leq \frac{\delta_{\mathcal{I}}^2(1-\delta_{\mathcal{I}})\gamma^2}{1-(1-\delta_{\mathcal{I}})\gamma} \frac{R_{\text{m}}^2}{(1-\gamma)^2}.$$

As noted, bounds on all other terms can be derived in the same fashion. We can also bound any term which involves ε_1', such that, for any ε_1, we can choose ε_1' and ε_1'' (which will itself be a function of ε_1') to finally obtain:

$$\text{MSE} \leq \left(4(1-h) + 2\delta_{\mathcal{I}} + \frac{2\delta_{\mathcal{I}}^2(1-\delta_{\mathcal{I}})\gamma^2}{1-(1-\delta_{\mathcal{I}})\gamma} + \frac{2\delta_{\mathcal{I}}^2(1-\delta_{\mathcal{I}})^2\gamma^4}{(1-(1-\delta_{\mathcal{I}})\gamma)^2}\right) \frac{R_{\text{m}}^2}{(1-\gamma)^2} + \varepsilon_1,$$

which holds with probability at least $1-\varepsilon_2$ provided $t > T$.

The final result is a simplified, less tight, version of the above inequality, with each instance of $1-\delta_{\mathcal{I}}$ replaced by 1. ■

We again note the fact that the scoring function being weighted by ψ is crucial for this result. The theorem suggests that using PASA will be of most value when an environment (P, R) and a policy π are such that a subset $\mathcal{I}$ exists which has the properties that:

1. $|\mathcal{I}|$ is small compared to S; and
2. $h(\mathcal{I}, \pi)$ is close to 1.

Indeed if a subset $\mathcal{I}$ exists such that h is arbitrarily close to one and $X \geq |\mathcal{I}|\lceil\log_2 S\rceil$, then we can in principle obtain a VF estimate using PASA which has arbitrarily low error according to L. When P, π and R are deterministic, or at least sufficiently close to deterministic, then we can also make equivalent statements in relation to $\tilde{L}$ and MSE (the latter follows from using the observation in the last paragraph before Theorem 12). Whilst results which guarantee low MSE are, in a sense, stronger than those which guarantee low L or $\tilde{L}$, the latter can still be very important. Some algorithms seek to minimise L or $\tilde{L}$ by using estimates of these values to provide feedback. Hence if PASA can minimise L or $\tilde{L}$ it will compare favourably with any algorithm which uses such a method. (Furthermore, the results for L and $\tilde{L}$ of course have weaker conditions.) We discuss the extension of these results to a continuous state space setting in Section 3.4.

It is worth stressing that, for MSE for example, assuming all of the conditions stated in Theorem 12 hold—and assuming $w(s_i, a_j) = \psi_i\pi(a_j|s_i)$—then, provided P and R are unknown, and given SARSA-F with *any* fixed state aggregation with $X < S$, it is impossible to guarantee MSE $< R_{\text{m}}^2/(1-\gamma)^2$ (i.e. the naïve upper bound).[35] This underscores the potential power of the result given suitable conditions.

The differences between each of the three bounds arise as a natural consequence of differences between each of the scoring functions. The bounds stated are not likely to be tight in general. It may also be possible to generalise the results slightly for different weightings $w(s_i, a_j)$ however any such generalisations are likely to be of diminishing value. All three bounds immediately extend to any projected form of any of the three scoring functions for the reason noted in Section 1.4.

Theorem 12 only ensures that, for a *particular* triple (P, R, π), we can find PASA parameters such that the result will hold. If we have an infinite set of triples (for example if we are drawing a random sample from a known prior for P), we cannot guarantee that there exists a single set of parameters such that the result will hold for all elements of the set.

However an alternative related theorem, which extends from Proposition 7, addresses this, and will be helpful for our discussion in Section 3.5. Suppose we have a set $\mathcal{Z}$ of triples (P, R, π). We can define $\delta_{P,\mathcal{Z}} := \sup\{\delta_P : (P, R, \pi) \in \mathcal{Z}\}$ and define $\delta_{R,\mathcal{Z}}$, $\delta_{\pi,\mathcal{Z}}$ and $\delta_{\mathcal{I},\mathcal{Z}}$ in an analogous way. It is possible to use identical arguments to Theorem 12, with each of δ_P, δ_R, δ_π and $\delta_{\mathcal{I}}$ replaced by $\delta_{P,\mathcal{Z}}$, $\delta_{R,\mathcal{Z}}$, $\delta_{\pi,\mathcal{Z}}$ and $\delta_{\mathcal{I},\mathcal{Z}}$ respectively, to obtain an equivalent theorem

35. This can be shown by constructing a simple example, which we briefly sketch. We know at least one cell exists with more than one state. We can assume that $\psi_i = 0$ (or is at least arbitrarily close to zero) for all but two states, both of which are inside this cell. Call these states s_1 and s_2, and assume $A = 1$ (this keeps the arguments simpler) so that we have a single action a_1. Suppose $R(s_1, a_1) = R_{\text{m}}$ w.p. 1 and $R(s_2, a_1) = -R_{\text{m}}$ w.p. 1, and the transition probabilities are such that s_1 transitions to s_2 with probability p, and s_2 transitions to s_1 *also* with probability p. We assume the prior for P is such that p may potentially assume an arbitrarily low value. For any fixed γ, by selecting p arbitrarily close to zero we will have MSE arbitrarily close to the non-zero lower bound we have just stated. Note that here we assume that $\eta = \eta(\gamma, p)$ is sufficiently small so that the VF estimate of SARSA converges to an arbitrarily small neighbourhood of zero. Similar arguments can be constructed to show that, for L and $\tilde{L}$, there is a similar minimum guaranteed lower bound of $R_{\text{m}}^2/2(1-\gamma)^2$ (we omit the details).

which will apply to any *set* of triples (P, R, π) that satisfies the conditions of Proposition 7. This is important because it means we can place bounds on VF error, for a single SARSA-P algorithm, given only a *prior* for (P, R). The trade-off is the presence of slightly stronger assumptions. The proof is omitted as it is identical to Theorem 12, except that we rely on Proposition 7 instead of Propositions 4 and 5. Since we are using Proposition 7 we must assume we have adopted the alterations to PASA in points 3 and 4 of Appendix A.5 (see also the related comments in Section 3.2):

Theorem 14 *Take some value $\varphi \in [0, 1]$. Suppose we have a set of triples $\mathcal{Z}$, each element of which is defined on the same set of states and actions, such that, for some $\xi > 2\varphi$, every element of $\mathcal{Z}$ contains a subset of states $\mathcal{I}$ of size $|\mathcal{I}| \leq \kappa$ which satisfies $\sum_{i:s_i \notin \mathcal{I}} \psi_i \leq \varphi$ and $s_i \in \mathcal{I} \Rightarrow \psi_i \geq \xi$. Suppose also that each $|R(s_i, a_j)|$ is bounded for all i and j for every element of $\mathcal{Z}$ by a single constant. We let R_{m} (another constant) denote the supremum of $|\mathrm{E}(R(s_i, a_j))|$ for all i and j and all $\mathcal{Z}$. Then for all $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ there exists T, η, ϑ and a parameter vector ς such that, provided $t \geq T$ and $X \geq \kappa\lceil\log_2 S\rceil$, for each element of $\mathcal{Z}$ the VF estimate $\hat{Q}^{(t)}$ generated by SARSA-P (where PASA incorporates changes 3 and 4 from Appendix A.5) will be such that, with probability equal to or greater than $1 - \varepsilon_2$, MSE, L and $\tilde{L}$ will satisfy the bounds stated in Theorem 12, where we replace $1 - h$ with φ, and where we replace each of δ_P, δ_R, δ_π and $\delta_\mathcal{I}$ by $\delta_{P,\mathcal{Z}}$, $\delta_{R,\mathcal{Z}}$, $\delta_{\pi,\mathcal{Z}}$ and $\delta_{\mathcal{I},\mathcal{Z}}$ respectively.*

As a final technical note, alterations can be made to the PASA algorithm (with no impact on computational complexity) that can remove the $\lceil\log_2 S\rceil$ factor in Propositions 5 and 7 and therefore also in Theorems 12 and 14 (the alteration involves, in effect, merging non-singleton cells in the partition Ξ). However such an alternative algorithm is more complex to describe and unlikely to perform noticeably better in a practical setting.

3.4. Extension to Continuous State Spaces

The results in Sections 3.1 to 3.3 can be extended to continuous state spaces, whilst retaining all of their most important implications. We will discuss this extension informally before introducing the necessary formal definitions.

It is typical, when tackling a problem with a continuous state space, to convert the agent's input into a discrete approximation, by mapping the continuous state space to the elements of some partition of the state space. Indeed, any computer simulation of an agent's input implicitly involves such an approximation.

It is not possible, in the absence of quite onerous assumptions (for example regarding continuity of transition, policy and reward kernels), to guarantee that, in the presence of such a discrete approximation, MSE, L or $\tilde{L}$, suitably redefined for the continuous case, are arbitrarily low.[36] However we can extend our discrete case analysis as follows. We assume that we begin with a discrete approximation of the continuous state space consisting of D *atomic cells*. SARSA with a fixed state aggregation approximation architecture corresponding to the D atomic cells will have associated values for MSE, L and $\tilde{L}$, which we

36. The extension of the formal definitions for MSE, L and $\tilde{L}$ to a continuous state space setting is reasonably self evident in each case. We do not provide definitions however some relevant details are at Appendix A.2.

denote MSE_0, L_0 and $\tilde{L}_0$. Each value MSE_0, L_0 and $\tilde{L}_0$ should be considered as a minimal "baseline" error, the minimum error possible assuming our initial choice of atomic cells.

As a rule of thumb, the finer our approximation, the lower MSE_0, L_0 and $\tilde{L}_0$ will typically be. To ensure low minimum error, we will often choose each atomic cell to be very small initially, with the implication that D will be very large (the value D can be considered, in a sense, as analogous to the value S in the discrete case). If D becomes too large, however, we may be in the position where applying an RL algorithm such as SARSA *directly* to the atomic cells becomes computationally intractable. In such a case we can employ PASA just as in the discrete case. All of our analysis in Sections 3.1 and 3.2 again holds. This allows us to derive a similar result to Theorem 12 above, however instead of guaranteeing arbitrarily low MSE, L or $\tilde{L}$, we instead guarantee MSE, L or $\tilde{L}$ which is arbitrarily close to the baseline values of MSE_0, L_0 or $\tilde{L}_0$.

Whilst the introduction of PASA does not remove the need to *a priori* select a discretisation of the state space, it gives us freedom to potentially choose a very fine discretisation (and, moreover, to choose a discretisation which is largely arbitrary) without necessarily demanding that we provide an underlying RL algorithm with a correspondingly large number of weights. This being the case, the application and advantages of PASA remain very similar to the case of finite state spaces.[37]

We will now formalise these ideas. Assume that the continuous state space $\mathcal{S}$ is a compact subset of d-dimensional Euclidean space $\mathbb{R}^d$ (noting that an extension of these concepts to more general state spaces is possible). We continue to assume that $\mathcal{A}$ is finite. Consistent with Puterman (2014), we redefine π, P and R as policy, transition and reward kernels respectively. As per our discussion above we assume that a preprocessing step maps every state in the continuous state space $\mathcal{S}$ to an element of a finite set $\mathcal{D}$ (a discrete approximation of the agent's input) of size D, via a mapping $m : \mathcal{S} \to \mathcal{D}$.

We will use $\Psi^{(t)}$ to represent the distribution of s at a particular time t given some starting point $s^{(1)}$. Given any fixed policy kernel π, we have a Markov chain with continuous state space. Provided that the pair (π, P) satisfies certain mild conditions we can rely on Theorem 13.0.1 of Meyn and Tweedie (2012) to guarantee that a stable distribution $\Psi^{(\infty)}$ exists, to which the distribution $\Psi^{(t)}$ will converge. We order the elements of $\mathcal{D}$ in some arbitrary manner and label each element as d_i $(1 \leq i \leq D)$. We now re-define the stable state probability vector ψ for the continuous case such that:

$$\psi_i = \int_{m^{-1}(d_i)} d\Psi^{(\infty)}(s).$$

It should be evident that all of the results in Sections 3.1 and 3.2 can be extended to the continuous case, with each result stated with reference to atomic cells, as opposed to

37. We can provide a more concrete example. We might imagine a bounded d-dimensional continuous state space. Assuming no prior knowledge of the problem, we assign states to atomic cells according to a uniform d-dimensional tiling. In order to ensure low values of MSE_0, L_0 and $\tilde{L}_0$, we elect to use extremely small tiles. However, as a result of the tiles being small, D—the number of individual tiles—is so large that SARSA cannot be applied directly. Hence we introduce PASA, in which case the problem becomes computationally tractable, and we have the benefit of the theoretical guarantees offered in this subsection.

individual states. We similarly redefine h as follows, where $\mathcal{I} \subset \mathcal{D}$:

$$h(\mathcal{I}, \pi) = \sum_{d_i \in \mathcal{I}} \psi_i.$$

The definition of δ-deterministic remains unchanged for π, as does the definition of δ_π. For the definitions of δ-deterministic for P and R, the definition of δ-constrained, and the definitions of δ_P, δ_R and $\delta_\mathcal{I}$, we redefine each of these in the obvious way with reference to atomic cells (and, as appropriate, $\Psi^{(\infty)}$ and π) as opposed to individual states.

Each scoring function in the continuous case can be defined with reference to an arbitrary weighting $w(s, a_j)$. As in the discrete case, however, we will need to assume, for our result, that either $w(s, a_j) = \pi(a_j|s) d\Psi^{(\infty)}(s)$ (for MSE) or $w(s, a_j) = \tilde{w}(m(s), a_j) d\Psi^{(\infty)}(s)$ (for L and $\tilde{L}$) for all s and j. In this context $\tilde{w} : \mathcal{D} \times \mathcal{A} \to [0, 1]$ is defined as an arbitrary function which must satisfy, in addition to $0 \leq \tilde{w}(d_i, a_j) \leq 1$ for all i and j, the constraint $\sum_{j'=1}^{A} \tilde{w}(d_i, a_{j'}) \leq 1$ for all i. Note that this definition of $\tilde{w}$ implies that the weighting is a constant function of s over each atomic cell. In general π may not satisfy this final constraint, however in practice usually we would expect to have $\pi(a_j, s') = \pi(a_j, s)$ for all s and s' such that $s \in d_i$ and also $s' \in d_i$ for some d_i. We can now generate an analogue to Theorem 12 for the case of continuous state spaces.

Theorem 15 *For a particular instance of the triple (P, R, π), suppose that each $|R(s, a_j)|$ for all s in $\mathcal{S}$ and for all j is bounded by a single constant. We let R_m (another constant) denote the supremum of $|\mathrm{E}(R(s, a_j))|$ for all j and $s \in \mathcal{S}$. For any subset $\mathcal{I} \subset \mathcal{D}$, if $X \geq |\mathcal{I}| \lceil \log_2 D \rceil$ and $\min\{\psi_i : d_i \in \mathcal{I}\} > 1 - h(\mathcal{I}, \pi)$ then, for all $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ there exists T, ϑ and a parameter vector ς such that, provided $t \geq T$, with probability equal to or greater than $1 - \varepsilon_2$ the VF estimate $\hat{Q}^{(t)}$ generated by SARSA-P (using the PASA algorithm parametrised by X, ϑ and ς) will be such that:*

1. *If $w(s, a_j) = \pi(a_j|s)\, d\Psi^{(\infty)}(s)$ then:*

$$\mathrm{MSE} - \mathrm{MSE}_0 \leq \left(4(1-h) + \frac{33\delta_\mathcal{I}}{1-\gamma}\right) \frac{R_\mathrm{m}^2}{(1-\gamma)^2} + \varepsilon_1;$$

2. *If $w(s, a_j) = \tilde{w}(m(s), a_j)\, d\Psi^{(\infty)}(s)$ for an arbitrary function $\tilde{w}$ satisfying $0 \leq \tilde{w}(d_i, a_j) \leq 1$ and $\sum_{j'=1}^{A} \tilde{w}(d_i, a_{j'}) \leq 1$ for all i and j then:*

$$L - L_0 \leq \frac{4(1-h) + 29(1 - (1-\delta_P)(1-\delta_\pi))}{(1-\gamma)^2} R_\mathrm{m}^2 + \varepsilon_1;$$

and:

$$\tilde{L} - \tilde{L}_0 \leq \left(2(1-h) + \gamma^2 \left(1 + 2(1-\delta_P)(1-\delta_\pi) - 3(1-\delta_P)^2(1-\delta_\pi)^2\right)\right) \frac{2R_\mathrm{m}^2}{(1-\gamma)^2} + \varepsilon_1.$$

The proof is similar to the proof of Theorem 12 and can be found at Appendix A.2. It should be clear that Remark 13 also applies, without significant change, to Theorem 15. A further theorem equivalent to Theorem 14 can also be generated in much the same way. Again the bounds (given, in particular, the details of the proof) are unlikely to be tight.

Note that each bound in Theorem 15 is, for technical reasons (see Appendix A.2), slightly different from the discrete case. However the implications of each bound are largely equivalent to the those in the discrete case. For example, a key implication of Theorem 15 is that, if $\mathcal{I}$ exists such that $\mathcal{I}$ consists of a small number of atomic cells and $h(\mathcal{I}, \pi)$ is large, we will be able to obtain arbitrarily low values for all three error functions compared to the baseline initially imposed by our mapping m. Similar to the discrete case this is of course implicitly subject to, in part, the values $\delta_\mathcal{I}$, δ_P and δ_π, depending on the scoring function of interest. The values of $\delta_\mathcal{I}$ and δ_P in the continuous case, defined, as they are, with reference to $\mathcal{D}$, $\Psi^{(\infty)}$ and π, are likely to depend on the continuity properties of P and π, as well as the size of the atomic cells. It may be possible to derive parts of Theorem 12 as a corollary to the result just outlined, however additional terms which appear in the continuous case bounds imply that this is not generally the case, at least insofar as the results have been stated here.

3.5. Application of Theorems 12 and 14 to Some Specific Examples

We can start to illustrate the consequences of Theorems 12 and 14 by examining some concrete examples. Again we are limiting our discussion to policy evaluation, though some of our observations will have consequences for policy improvement as well. Examples 1, 2 and 5 will show situations where we expect our results to guarantee that, for many policies which we might encounter, VF error can be substantially reduced by using PASA. Examples 3 and 4 will provide situations where our results will not be able to provide useful bounds on error (and by extension we cannot expect introducing PASA to have a meaningful impact). There is potential to extend our analysis far beyond the brief examples which we cover here.

Our first four examples are based on a "Gridworld" environment—see, for example, Sutton and Barto (2018)—although the principles we explore in each case will apply more generally. The agent is placed in a square $N \times N$ grid, where each point on the grid is a distinct state (such that $S = N^2$). It can select from four different actions: up, down, right and left. If the agent takes an action which would otherwise take it off the grid, its position remains unchanged. Certain points on the grid may be classed as "walls": points which, if the agent attempts to transition into them, its position will also remain unchanged.

Example 1 *Suppose that the agent can only occupy a small number n of points within a grid where N is very large (for example as a result of being surrounded by walls). See Figure 2a. Assume* we don't know in advance which points the agent can occupy. *Immediately we can use the Theorem 12 to guarantee for a particular P that, if $X \geq n \log_2 N^2$, we will have arbitrarily low VF error (for any of the three scoring functions subject to the stated conditions for w and subject, for $\tilde{L}$, to suitable constraints being placed on R) for any policy if we employ a suitably parametrised PASA algorithm. By employing Theorem 14 instead, we can extend this guarantee to a single PASA algorithm for all possible environments generated according to a suitably defined prior for P (one for which the number of accessible states will not exceed n).*

This example, though simple, is illustrative of a range of situations a designer may be presented with where only a small proportion of the state space is of any importance

to the agent's problem of maximising performance, but either exactly which proportion is unknown, or it is difficult to tailor an architecture to suitably represent this proportion.

As we illustrate in the next example, however, environments where there are no such constraints can be equally good potential candidates. This is because many (perhaps even most) commonly encountered environments have a tendency to "funnel" agents using deterministic or near-deterministic policies into relatively small subsets of the state space.

Example 2 *Consider a grid with no walls except those on the grid boundary. Suppose that π is deterministic (such that the same action is always selected from the same point in the grid) and selected uniformly at random. Starting from any point in the grid, the average number of states the agent will visit before revisiting a state it has already visited (and thereby creating a cycle) will be loosely bounded above by 5. This follows from considering a geometric distribution with parameter $p = 1/4$, which is the probability of returning to the state which was just left. The bound clearly also applies to the number of states in the cycle the agent ends up entering, and is true for any N.*

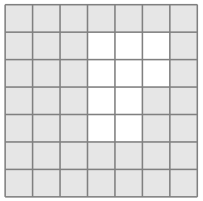
Hence, for any value of N, provided $X \geq 5 \log_2 N^2$, we can use Theorem 12 to guarantee that MSE, L and $\tilde{L}$ will all be arbitrarily low with a high probability (in the case of $\tilde{L}$ this is also conditional on the reward function R having low variance). Even if instead, for example, π is δ-deterministic where δ is small, the above arguments imply that the agent is still likely to spend extended periods of time in only a small subset of the state space.

PASA may fail to provide an improvement, however, where the nature of an environment and policy is such that the agent must consistently navigate through a large proportion of the state space, as illustrated by the next two examples. Note that, notationally, $g(x) = \Omega(f(x))$ is equivalent to $f(x) = O(g(x))$.

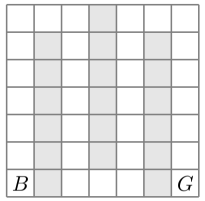
Example 3 *Suppose that the same environment in Example 2 now has a single "goal state". Whenever the agent takes an action which will mean it would otherwise enter the goal state, it instead transitions uniformly at random to another position on the grid and receives a reward of 1. All other state-action pairs have a reward of zero.*

Suppose a policy is such that the average reward obtained per iteration is β (which must of course be less than 1). Every state must then have a lower bound on the probability with which it is visited of $\beta/(S-1)$, since every state must be visited at at least the same rate as the goal state, divided by all states excluding the goal state. Accordingly, for any subset $\mathcal{I}$ of the state space of size n, we must have $h \leq 1-(S-1-n)\beta/(S-1) = n\beta/(S-1)+(1-\beta)$. Hence h is constrained to be small if β is large (i.e., if the policy performs well) and n is small compared to S. Accordingly this environment would not be well suited to employing PASA.

Example 4 *Suppose that the same environment in Example 3 now has its goal state in the bottom right corner of the grid, and that, instead of transitioning randomly, the agent is sent to the bottom left corner of the grid (the "start state") deterministically when it tries to enter the goal state (again receiving a reward of 1). Suppose that $N = 3 + 4k$ for some integer $k \geq 0$, and also that walls are placed in every second column of the grid as shown in Figure 2b (with "doors" placed in alternating fashion at the top and bottom of the grid). Clearly an optimal policy in this case will be such that for all subsets $\mathcal{I}$ of the state space of*



(a) Example of an environment type well suited to PASA (assuming much larger grid dimensions than illustrated). Only a comparitively small number of states (not known in advance to the designer) can be accessed by the agent.



(b) Example of an environment type *not* well suited to PASA (assuming much larger grid dimensions than illustrated). G is the goal state and B is the state transitioned to when the agent attempts to enter the goal state. The placement of the walls implies that an optimal policy will force the agent to regularly visit $O(S)$ states.

Figure 2: Example Gridworld diagrams for $N = 7$ (the case $N = 7$ is convenient to illustrate diagrammatically however our interest is in examples with much larger N). Each white square is a point in the grid. Grey squares indicate walls.

size n, $h < 2n/S$. Hence we would require $X = \Omega(S)$ in order to be able to apply Theorem 12 to the optimal policy.

In general, to find examples for deterministic P where this is the case requires some contrivance, however. If we removed the walls, for example, an optimal policy would be such that SARSA-P would obtain a VF estimate with arbitrarily low error provided $X(S) \geq f(S)$ for a particular function f which is $O(\sqrt{S}\log_2 S)$. This is true for any fixed start and goal states.

Whilst not all environments are good candidates for the approach we've outlined, very many commonly encountered environment types would appear to potentially be well suited to such techniques. To emphasise this, interestingly, even environments and policies with no predefined structure at all have the property that an agent will often tend to spend most of its time in only a small subset of the state space. Our next example will illustrate this. It is a variant of the GARNET problem.[38]

Example 5 *Consider a problem defined as follows. Take some $\delta > 0$. We assume: (1) P is guaranteed to be δ-deterministic, (2) P has a uniform* prior *distribution, in the sense that, according to our prior distribution for P, the random vector $P(\cdot|s_i, a_j)$ is independently distributed for all (i, j) and each random variable $P(s_{i'}|s_i, a_j)$ is identically distributed for all (i, j, i'), and (3) π is δ-deterministic.*

38. Short for "generic average reward non-stationary environment test-bench." This is in fact a very common type of test environment, though it does not always go by this name, and different sources define the problem slightly differently. See, for example, Di Castro and Mannor (2010).

Generally, condition (2) can be interpreted as the transition function being "completely unknown" to the designer before the agent starts interacting with its environment. It is possible to obtain the following result in relation to environments of this type:

Lemma 16 *For all $\varepsilon_1 > 0$, $K > 1$ and ε_2 satisfying $0 < \varepsilon_2 < K - 1$, there is sufficiently large S, sufficiently small $\delta > 0$ and sufficiently small $\delta' > 0$ such that, for an arbitrary policy π, with probability no less than $1 - 1/(K - \varepsilon_2 - 1)\ln S$ conditioned upon the prior for P, we will have a set $\mathcal{I} \subset \mathcal{S}$ such that $|\mathcal{I}| \leq K\sqrt{\pi S/8}\ln S$, $h(\mathcal{I}, \pi) > 1 - \varepsilon_1$ and $\min_{i \in \mathcal{I}} \psi_i > 2(1 - h) + \delta'$.*

Details of the proof are provided in Appendix A.3. The result is stated in the limit of large S, however numerical analysis can be used to demonstrate that S does not need to be very large before there is a high probability that $\mathcal{I}$ exists so that $|\mathcal{I}|$ falls within the stated bound. Now, with the help of Theorem 14 we can state the following result in relation to Example 5, which can be obtained by simply applying Theorem 14 in conjunction with Lemma 16. (Since we are relying on Theorem 14 we must assume we are using a version of PASA which incorporates the changes in points 3 and 4 of Appendix A.5.) In particular we note that, as a result of Lemma 16, for all S, we can, for any $\varphi > 0$, select δ and ξ such that we will have $\sum_{i:s_i \notin \mathcal{I}} \psi_i \leq \varphi$ and $s_i \in \mathcal{I} \Rightarrow \psi_i \geq \xi > 2\varphi$. We can also, for all S, select δ such that each of δ_P, δ_π and $\delta_\mathcal{I}$ is arbitrarily close to zero, where for $\delta_\mathcal{I}$ we rely on the observation made in the last paragraph before Theorem 12. As a result we will have:

Corollary 17 *Suppose that, for Example 5, the prior for R is such that R is guaranteed to satisfy $|\mathrm{E}(R(s_i, a_j))| \leq R_\mathrm{m}$ for all i and j, where R_m is a constant. Then for all $\varepsilon_1' > 0$, $\varepsilon_1'' > 0$, $K > 1$ and ε_2 satisfying $0 < \varepsilon_2 < K - 1$, there is sufficiently large S and sufficiently small δ such that there exist values T, ς and ϑ such that, provided $X \geq K\sqrt{\pi S/8}\ln S\lceil\log_2 S\rceil$, SARSA-P will (with a PASA component which incorporates the changes in points 3 and 4 of Appendix A.5 and which is parametrised by ς, ϑ and X) generate for $t > T$, with probability no less than $(1 - 1/(K - \varepsilon_2 - 1)\ln S)(1 - \varepsilon_1'')$ conditioned upon the prior for P, a VF estimate with $\mathrm{MSE} \leq \varepsilon_1'$ and $L \leq \varepsilon_1'$. If the prior for R is also such that R is guaranteed to be δ-deterministic, then the same result will hold for $\tilde{L}$.*

What the result tells us is that, even where there is no apparent structure at all to the environment, an agent's tendency to spend a large amount of time in a small subset of the state space can potentially be quite profitably exploited. Our experimental results (where we examine an environment type equivalent to that described by Example 5) will further confirm this.

The bound on X provided is clearly sub-linear to S, and may represent a significant reduction in complexity to the standard tabular case when S starts to take on a size comparable to many real world problems. (Whilst in practice, the key determinant of X will be available resources, the result of course implies the potential to deal with more complex problems—or complex representations—given fixed resources.) Furthermore the bound on X in Corollary 17 appears to be a loose bound and can likely be improved upon. Our result as stated only pertains to policies generated with no prior knowledge of P, however we can see that, for any R which is independent of P, for any policy π and for any $\varepsilon > 0$,

an optimal policy π^* will be such that:

$$\Pr(\min\{K : |\mathcal{I}| = K, h(\mathcal{I}, \pi^*) > 1 - \varepsilon\} < x) \geq \Pr(\min\{K : |\mathcal{I}| = K, h(\mathcal{I}, \pi) > 1 - \varepsilon\} < x).$$

So Corollary 17 will also apply to optimal policies.

An implication of the result is that, assuming our condition on X holds, then SARSA-P will have arbitrarily low VF error with arbitrarily high probability for sufficiently large S and sufficiently small δ. Our result was not stated in quite this way for a technical reason. Namely, whilst Corollary 17 does not apply to SARSA-F, *fixed* state aggregation will *also* tend to have zero error if $X \neq O(\sqrt{S}\ln S)$, S is sufficiently large, δ is sufficiently small and all our other assumptions continue to hold. This is because the probability of more than one state in a set $\mathcal{I}$ falling into a single cell tends to zero if the number of cells grows at a faster rate than the size of $\mathcal{I}$. However, since the arrangement of the states amongst cells will be uniformly random (since Ξ is arbitrary and P is uniform), then for SARSA-F to have only one element of the set $\mathcal{I}$ to which Lemma 16 refers in each cell will require both that the set $\mathcal{I}$ exists *and* that each state in $\mathcal{I}$ happens to fall into its own cell. So SARSA-F will always have an additional (generally very small, and therefore highly detrimental) factor contributing to the probability of arbitrarily low error. In effect, the probability of each element of a set $\mathcal{I}$ of states falling into a unique fixed cell increases slowly, so that any guarantee pertaining to SARSA-F can only made with much lower probability than for an equivalent guarantee for SARSA-P as S is increased.

There is some scope to extend Lemma 16 beyond uniform transition function priors. However the theoretical complexity involved with generating formal results can increase significantly as we add more complexity to our prior for P. Furthermore, in some special cases, knowledge about the value of P (i.e. a non-uniform prior) can mean our results will not apply. If we know, for example, that $P(s_{i+1}|s_i, a_j) = 1$ for all $1 \leq i \leq S - 1$ and all j, and $P(s_1|s_S, a_j) = 1$ for all j, then Lemma 16 clearly doesn't hold (much like in Example 4). There are some slightly more general sets of priors for which the result, or parts of the result, can be shown to hold. These minor extensions can be obtained, for example, by using the notion of Schur convexity. This is addressed in Appendix A.4.

Before considering our experimental results, we can summarise this subsection by observing, informally, that there appear to be two "pathologies" which P might suffer from which will result in PASA being likely to have little impact. The first is if P is subject to large degrees of randomness resulting in the agent being sent to a large proportion of the state space, as illustrated by Example 3 and the importance of the value δ in Example 5.[39] The second is if the prior for P has an exaggerated tendency to direct the agent through a large proportion of the state space, as illustrated by Example 4 and the example provided in the previous paragraph. These observations will be reinforced by our experimental results.[40]

39. High degrees of randomness in π can also create an issue, however in practice agents seeking to exploit learned knowledge of the environment will typically have near-deterministic policies.

40. Our examples have all focussed on discrete state spaces. Whilst we will not explore the details, Examples 1, 3 and 4 have ready continuous state space analogues. The same is true of Example 2 provided that certain assumptions hold around the continuity of P and π (so that the agent's policy is such that it is likely to return to a state close to one it has already visited). The implications which arise from each example in a continuous state space setting reflect closely the discrete state space case.

Table 1: Summary of main theoretical results.

Result	Description of result (omitting conditions)	Requires
Proposition 4	PASA parameters exist which guarantee, given a single pair (P, π), that Ξ will converge.	nil
Proposition 5	The convergence point of Ξ in Proposition 4 will be such that the agent will spend a large amount of time in singleton cells.	Proposition 4
Proposition 7	Equivalent to Propositions 4 and 5, however applies to a set of pairs (P, π), and the statement regarding the limit Ξ is slightly weaker.	nil
Theorem 12	The convergence point of Ξ in Proposition 4 is such that, given a single triple (P, R, π), the values of MSE, L and $\tilde{L}$ will be bounded. These bounds, under suitable conditions, will be arbitrarily close to zero.	Propositions 4 and 5
Theorem 14	Equivalent to Theorem 12, however the bounds on MSE, L and $\tilde{L}$ will apply to all triples (P, R, π) in a set.	Proposition 7
Theorem 15	Extension of Theorem 12 to continuous state spaces (we claim the existence of an equivalent extension to 14).	Propositions 4 and 5[a]
Lemma 16	Given a uniform prior for P, an agent will spend an arbitrary large proportion of the time in a subset of the state space containing $O(\sqrt{S} \ln S)$ states with high probability.	Results in Section A.3 (Lemmas 18 and 19)
Corollary 17	Given the same conditions as Lemma 16, MSE, L and $\tilde{L}$ will be arbitrarily low provided $X \geq K\sqrt{\pi S/8} \ln S \lceil \log_2 S \rceil$ for a constant K as S becomes large.	Lemma 16 and Theorem 14

[a] The result in fact requires more general versions of Propositions 4 and 5 applicable to continuous state spaces. Such extensions are straightforward in the context of our arguments however, and we do not provide these more general results formally.

4. Experimental Results

Our main objective in this section will be to determine via experiment the impact that PASA can have on actual performance. This will help clarify whether the theoretical properties of PASA which guarantee decreased VF error in a policy evaluation setting will translate to improved performance in practice. Whilst our principle intention here is to validate our theoretical analysis and demonstrate the core potential of PASA, more wide-ranging experimental investigation would comprise an interesting topic for further research.

We have examined three types of problem: (a) a variant of the GARNET class of problem (substantially equivalent to Example 5), (b) a Gridworld problem and (c) a logistics problem. All three of the environment types we define in terms of a prior on (P, R), which allows for some random variation between individual environments. In all of our experiments we compare the performance of SARSA-P to SARSA-F (both with the same number X of cells). In some cases we tested SARSA-F with more than one state aggregation, and in most cases we have also tested, for comparison, SARSA with *no* state aggregation.

We will see that in all cases (with the exception of one isolated result which appears to have been affected by experimental noise) SARSA-P exhibits better performance than SARSA-F, in some cases substantially so (for the GARNET problem we also demonstrate that, as predicted, SARSA-P results in lower MSE for randomly generated fixed policies). We will also see that in some key instances SARSA-P outperforms SARSA with no state aggregation. The parameters[41] of PASA were kept the same *for all environment types*, with the exceptions of X_0 and X (with X being changed for SARSA-F as well). The value of X_0 was always set to $X/2$. The parameters used are shown in Table 2.

Furthermore (as summarised in Table 5) SARSA-P requires only marginally greater computational time than SARSA-F, consistent with our discussion in Section 3.1. Whilst we have not measured it explicitly, the same is certainly true for memory demands.

Each experiment was run for 100 individual trials for both SARSA-P, SARSA-F and (where applicable) SARSA with no state aggregation, using the same sequence of randomly generated environments. Each trial was run over 500 million iterations.

For our experiments some minor changes have been made to the algorithm SARSA-P as we outlined it above (that is, changes which go beyond merely more efficiently implementing the same operations described in Algorithms 1 and 2). The changes are primarily designed to increase the speed of learning. These changes were not outlined above to avoid adding further complexity to both the algorithm description and parts of the theoretical analysis, however they in no way materially affect the manner in which the algorithm functions, and none of the changes affect the conclusions in Section 3.1. The changes are outlined in Appendix A.5. Whilst SARSA-P continues to outperform SARSA-F in the absence of these changes, without them SARSA-P tends to improve at a slower rate, making the difference observable from experiment less pronounced.

Unless otherwise stated, for SARSA-F, the state aggregation was generated arbitrarily, subject to cell sizes being as equally sized as possible. For SARSA-P, to generate the initial partition Ξ_0 we ordered states arbitrarily, then, starting with a partition containing a single

41. In this section we use the word "parameter" to refer to the handful of high-level values which can be optionally tuned for either an agent or environment (sometimes called "hyper-parameters"), such as η, ς, ϵ, γ and ϑ. It will not be used, for example, to refer to the matrix of values θ stored by SARSA (also sometimes referred to in the literature as "parameters").

cell containing every state, we recursively split the cells with indices indicated by the first X_0 elements of the following sequence:

$$(1, 1, 2, 1, 2, 3, 4, 1, 2, 3, 4, 5, 6, 7, 8, \ldots).$$

Doing so results in roughly equally sized cells, which in practice, assuming the absence of any specific information regarding the environment, is preferable.

Table 2: PASA parameters used in all experimental domains.

Parameter	Type[a]	Description	Value(s)
ϵ	S	ϵ-greedy parameter	0.01, 0.001[†]
γ	S	Discount rate	0.98
η	S	SARSA step size	3×10^{-4}
X	F	Number of cells	Differs by experiment[b]
X_0	P	Number of base cells	$X/2$
ς	P	PASA step size	1×10^{-8}
ν	P	PASA update frequency	50,000
ϑ	P	ρ update tolerance	0.9

[a] The values in this column reflect whether the parameter relates to the SARSA algorithm (S) (in which case the parameter applies to all three of the SARSA-P, SARSA-F and SARSA with no state aggregation experiments), both SARSA-F and SARSA-P (F), or the PASA algorithm only (P).
[b] Details can be found by referring to the discussion of each experiment.
[†] This parameter value was tested for the MSE experiments for the GARNET problem, but was not used for any of the performance experiments.

4.1. GARNET Problems

The prior for P for this environment is the same as that described in Example 5, however for the purposes of this experiment we have taken $\delta = 0$. We selected $s^{(1)}$ at random. Note that the transition function for a particular π for environments defined in such a way will not necessarily be irreducible. The prior for R is such that $R(s_i, a_j) = S$ with probability ζ/S for all (i, j) and is zero otherwise, where $\zeta > 0$ is a parameter of the environment. The way we have defined R is such that, as S increases or ζ decreases, state-action pairs with positive reward become more and more sparsely distributed (and therefore more difficult for the agent to find). However the expected reward associated with selecting actions uniformly at random remains constant for different values of S provided ζ remains constant.

We ran two sets of experiments. In the first set we selected π randomly and held it fixed, allowing us to measure comparative MSE for SARSA-P and SARSA-F. Each π generated was ϵ-deterministic (see Section 3.3). Specifically, with probability $1 - \epsilon$, a deterministic (however initially randomly selected) action is taken and, with probability ϵ, an action is selected uniformly at random. In the second set of experiments we allowed π to be updated and measured overall performance for SARSA-P, SARSA-F and SARSA with no state aggregation. For the second set in every iteration the policy π was selected to be ϵ-greedy with respect to the current VF estimate. We ran each set of experiments with different values of S and ζ as described in Table 3.

Table 3: GARNET experiment parameters.

Parameter	Description	Value(s)
S	Number of states	250, 500, 1K, 2K[*], 4K[*], 8K[*]
A	Number of actions	2
ζ	Expected state-action pairs with $R_{ij} = S$	30, 3[†]
X	Number of cells[a]	70, 100, 140, 200[*], 280[*], 380[*]

[a] This parameter pertains only to SARSA-P and SARSA-F. The individual values were arrived at heuristically, however with the guidance of Lemma 16.
[*] These parameter values were not tested for the MSE experiments.

The results of the experiments relating to MSE can be seen in Table 4. These results demonstrate that, as the results in Section 3 predict, PASA has the effect of reducing MSE for a fixed policy. The effect of applying PASA becomes more pronounced as ϵ is reduced, which is also consistent with our predictions. We obtained similar results (not reproduced) for $\zeta = 3$. (Note that, even for relatively low values of S, calculating MSE can quickly become prohibitively expensive from a computational standpoint, hence why we have limited these experiments to $S \leq 1{,}000$.)

In relation to actual performance, Figures 3 demonstrates that SARSA-P outperforms SARSA-F in every experiment (with the exception of one apparent anomaly, which we assume is due to noise), and in many cases by a substantial margin. Furthermore, as S increases and ζ decreases, that is, as the problem becomes more complex requiring the agent to regularly visit more states, the disparity in performance between SARSA-P and SARSA-F widens. Part of the reason for this widening may be because the VF estimates of SARSA-F will experience rapid decay in accuracy as cells are forced to carry more states, whereas SARSA-P (for the reasons we examined in Section 3.5) will be less affected by this issue.[42]

We have included, for comparison, the performance of SARSA with no VF approximation. In some cases SARSA-P compares favourably with the case where there is no state aggregation, although as S becomes large SARSA with no state aggregation starts to out-perform SARSA-P. The GARNET environment represents unique challenges for *any* RL algorithm which employs VF approximation (since every single state has behaviour completely independent of all other states), such that the performance increase we see for SARSA with no VF approximation compared to SARSA-P may not be completely surprising. SARSA-P, of course, requires significantly fewer weights, and, in general, as S gets very large using SARSA with no state aggregation may become impossible. This issue is highlighted more strongly in the third of the three environments, considered in Section 4.3. This inability of SARSA to scale is, of course, a key motivator behind proposing SARSA-P. Given this, the comparison between SARSA-P and SARSA-F may be more instructive in

42. An examination of π over individual trials (not shown) suggests that SARSA-P has a stronger tendency to converge than SARSA-F (i.e. in spite of the fact that convergence cannot be guaranteed in general for this problem). This may be in part because policies which are clearly represented tend to be associated with a higher value, which in turn makes such policies more likely to be selected and therefore remain stable.

Table 4: Comparative MSE of SARSA-P and SARSA-F.

Parameter settings			Average $\sqrt{\text{MSE}}$[a]		
S	ζ	ϵ	SARSA-P	SARSA-F	% reduction[b]
250	30	0.01	0.48 (±0.012)	0.825 (±0.022)	**41.8**%
500	30	0.01	0.4 (±0.014)	0.584 (±0.017)	**31.5**%
1,000	30	0.01	0.337 (±0.008)	0.477 (±0.011)	**37**%
250	30	0.001	0.438 (±0.012)	0.977 (±0.031)	**55.2**%
500	30	0.001	0.325 (±0.016)	0.673 (±0.024)	**51.7**%
1,000	30	0.001	0.259 (±0.007)	0.554 (±0.016)	**67.9**%

[a] Average over 100 independent trials. Figures are based on the average MSE over the final fifth of each trial. The MSE was calculated by first calculating the true value function for each random policy π. Confidence intervals (95%) for $\sqrt{\text{MSE}}$ are shown in brackets.
[b] The extent to which average $\sqrt{\text{MSE}}$ was reduced by introducing PASA, expressed as a percentage.

this instance as the two algorithms share the same number of cells as a function of S (as well as similar—as discussed in Section 3.1—computational complexity).

4.2. Gridworld Problems

We examined two different Gridworld-type problems. As in Examples 1 to 4 in Section 3.5 we confined the agent to an $N \times N$ grid, where the agent can move up, down, left or right. We singled out r positions on the grid, which we call *reward positions*. These positions will, if the agent attempts to move into them, result in a reward of 1, and the agent will transition to another (deterministic) position on the grid, which we call a *start position*. Each reward position transitions to its own unique start position. The problem is defined such that the reward and start positions are uniformly distributed over the grid (subject to the constraint that each start and reward position cannot share its location with any other start or reward position). Provided r is relatively small compared to N^2, solutions are likely to involve a reasonably complex sequence of actions.

We chose $N = 32$ (so that $S = 1{,}024$). We ran two separate experiments for $r = 8$ and $r = 24$. We also ran a third experiment, again with $r = 24$, however in this case we altered P to be non-deterministic, specifically such that each reward position transitioned to a completely random point on the grid (instead of to its corresponding start position), in a rough analogue to Example 3.

It is common in RL experiments of this form for the designer to choose basis functions which in some way reflect or take advantage of the spatial characteristics of the problem. In our case we have chosen our X_0 base cells for SARSA-P in an arbitrary manner, as we deliberately do not want to tailor the algorithm to this particular problem type. We tested SARSA-F with two different state aggregations. One arbitrary (in line with that chosen for SARSA-P) and one with cells which reflected the grid-like structure of the problem, where states were aggregated into as-near-as-possibly equally sized squares.[43] Only results for the former are shown, as these were superior (possibly since equating proximal states

43. This type of state aggregation is commonly referred to as "tile coding". See, for example, Sutton and Barto (2018).

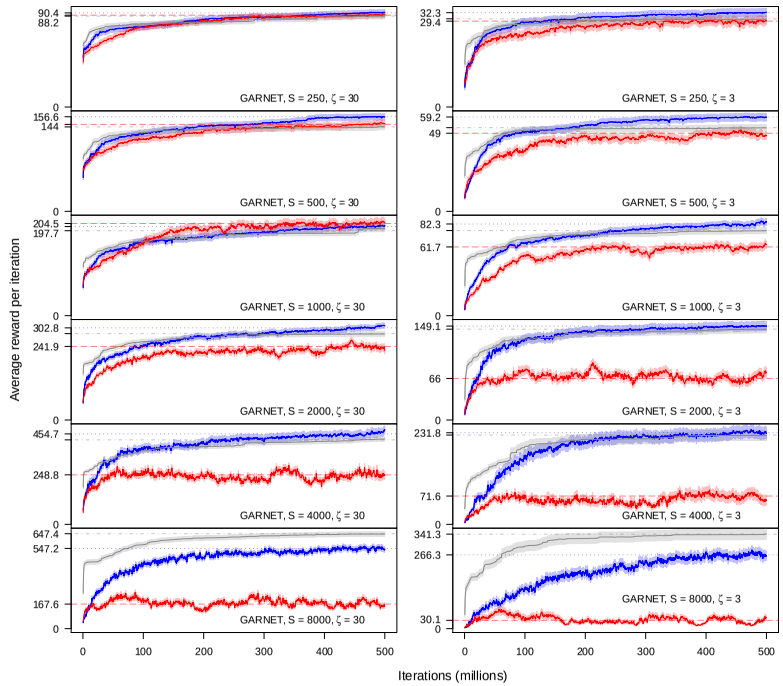


Figure 3: Comparative performance of SARSA-P (blue) and SARSA-F (red) as a function of t. Performance of SARSA with no state aggregation is also shown (grey). Horizontal lines indicate the average reward obtained over the final fifth of each trial. Notice that environments become progressively "more challenging" moving down or right through the figure. SARSA-P consistently outperformed SARSA-F (excluding the anomalous result for $S = 1{,}000$ and $\zeta = 30$, which appears to be affected by noise), and the disparity between the algorithms' performance increased as problem complexity increased. Slightly greater variance in reward obtained (for all three algorithms) can also be seen as problem complexity is increased.

can make the agent less likely to find rare short-but-optimal pathways through the state space). We also tested SARSA with no state aggregation. The algorithm parameter settings were $X_0 = 70$ and $X = 140$, with all other relevant parameters left unchanged from the GARNET problem settings.

The average performance of each algorithm as a function of time is shown in Figure 4. The results are significant insofar as SARSA-P has continued to outperform SARSA-F in a quite different, and more structured, setting than that encountered with the GARNET problem. This was despite some attempt to find an architecture for SARSA-F which was "tailored" to the problem (a process which, of course, implicitly requires an investment of time by the designer). The improved performance of SARSA-P was increased where $r = 8$. Hence the disparity in performance was even greater when r was set lower (i.e. when the problem requires more complex planning). Surprisingly, for this particular problem SARSA-P outperformed even SARSA with no state aggregation. We speculate that the reason for this is that the aggregation of states allows for a degree of generalisation which is not available to SARSA with no state aggregation.[44]

For the third experiment, consistent with our analysis in Section 3, SARSA-P's performance was well below its performance in the first experiment, despite the fact that policies exist in this environment which are likely to generate similar reward to that obtained in the first experiment. We would expect this drop in performance given this is not an environment well suited to using PASA (although the algorithm still performed far better than SARSA-F).

4.3. Logistics Related Problems

Many applications for RL come from problems in operations research (Powell, 2011; Powell et al., 2012). This is because such problems often involve planning over a sequence of steps, and maximising a single value which is relatively straightforward to quantify. Problems relating to logistics (e.g. supply chain optimisation) are a good example of this.

Our third environment involves a transport moving some type of stock from a depot to several different stores. We suppose that we have N stores, that each store (including the depot) has a storage rental cost (per unit stock held there), and that each store has a deterministic sales rate (the sales rate at the depot is zero). The agent has control of the transport (a truck), and also has the ability to order additional stock to the depot. There is a fixed cost associated with transporting from any given point to another. The following, therefore, are the agent's possible actions (only one of which is performed in each iteration): (a) ordering an additional unit of stock to the depot, (b) loading the truck, (c) moving the truck to one of the $N + 1$ locations (treated as $N + 1$ distinct actions), and (d) unloading the truck.

In what follows we will use $U(a, b)$, where $a < b$, $a \in \mathbb{R}$ and $b \in \mathbb{R}$ to denote a continuous uniform random variable in the range from a to b. In our experiment we have taken $N = 4$, capacities $= (12, 3, 4, 3, 6)$ (the first entry applies to the depot), transport cost $\sim$

44. More specifically, SARSA-P learns to associate states which have never (or have only very rarely) been visited with a slightly positive reward, due to its aggregation of many states over rarely visited areas of the state space. This encourages it to explore rarely-visited regions in favour of frequently visited regions with zero reward. SARSA with no state aggregation, in contrast, takes longer to revise its initial weights (in this case set to zero) due to the absence of generalisation between states.

$U(-1.2, -0.6)$, order cost $= -2$, sale revenue $= 7$, sales rate $= 1$ at all stores, and rent $= (U(-0.2, -0.05), U(-0.05, -0.01), U(-0.08, -0.03), U(-0.08, -0.01), U(-0.4, -0.001))$ (the first entry again applies to the depot). We stress that the randomness, where applicable, in these variables relates to the prior distribution of (P, R). There is no randomness once an instance of the environment is created. In arranging states into their initial cells for SARSA-P we have assumed we know nothing of the inherent problem structure (i.e. we selected an arbitrary cell arrangement). For SARSA-F we again trialled both arbitrary and tile coded state aggregations (where each stock level was divided into equal intervals; only the tile-coded results are shown, as these were superior, however the difference in performance was minimal).

We have kept the same agent parameters as in the GARNET and Gridworld problems, again with $X_0 = 70$ and $X = 140$. Note that the number of states which can be occupied is $13 \times 4 \times 5 \times 4 \times 7 \times 4 \times 2 = 72{,}800$. The majority of these states are likely to be rarely or never visited, or are of low importance based on the intrinsic structure of the problem. Complicating the problem further, the input provided to the agent is a binary string made up of 18 digits, consisting of the concatenation of each individual integer input variable converted to a binary expansion (which is a natural way to communicate a sequence of integers). As a result we effectively have $S = 2^{18} = 262{,}144$ (many of these states cannot be occupied by the agent, however the designer might be assumed to have no knowledge of the environment beyond the fact that each input consists of a binary string of length 18).

Over the 100 trials, SARSA-P obtained an average reward of 0.785 (which is close to optimal) compared to SARSA-F which was only able to obtain average reward of -0.019 (see Figure 4). Hence, for this problem, the disparity between the two algorithms was reasonably dramatic.

This is not a complex problem to solve by other means, despite the large state space.[45] However the example helps to further illustrate that SARSA-P can significantly outperform SARSA-F despite roughly equivalent computational demands and without requiring any prior information about the structure of the environment. Further experimentation could provide an indication as to how well SARSA-P performs on even more complex logistics problems, however the indications coming from this modest problem appear promising.

4.4. Comments on Experimental Results

Table 5 provides a summary of the experimental results. The table shows that, at a low computational cost, we are able to significantly improve performance by using PASA to update an otherwise naïve state aggregation architecture in a range of distinct problem types, and with minimal adjustment of algorithm parameters. The experiments where PASA had less impact were of a nature consistent with what we predicted in Section 3. Optimising parameters such as ς, ϵ or even the value of X might be expected to further increase the disparity between SARSA-P and SARSA-F.

It is interesting to consider the mechanics via which PASA appears able to increase performance, in addition to simply decreasing VF error. This is a complex question, and a rigorous theoretical analysis would appear to pose some challenges, however we still can attempt to provide some informal insight into what is occurring.

45. This is part of the reason why we have not tested SARSA with no state aggregation for this problem.

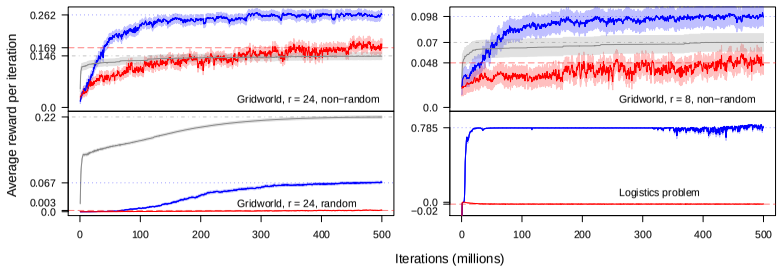


Figure 4: Comparative performance of SARSA-P (blue) and SARSA-F (red) as a function of t on Gridworld and logistics problems. Performance of SARSA with no state aggregation is also shown for Gridworld problems (grey). Horizontal lines indicate the average reward obtained over the final fifth of each trial. SARSA-P outperforms both SARSA-F and SARSA with no state aggregation by some margin on the Gridworld problem with deterministic start points. The disparity between SARSA-P and SARSA-F is greater when the number of available rewards is decreased. Introducing start points which are selected uniformly at random greatly diminishes the performance of SARSA-P (however it also diminishes the performance of SARSA-F). The logistics problem reveals a dramatic disparity in performance. SARSA-P consistently finds an optimal or near-optimal solution whilst SARSA-F fails to find a policy with average reward greater than zero.

SARSA-P is able to estimate the VF associated with a *current* policy with greater accuracy than SARSA-F. To some extent this is at the cost of less precise estimates of alternative policies, however this cost may be comparatively small, since adding additional error to an estimate which *already* has high error may have little practical impact. The fact that SARSA-F (unlike SARSA-P) is forced to share weights, even amongst only a small number of states (which is likely to be inevitable with *any* fixed linear approximation architecture, not just state aggregation, unless the designer makes strong initial assumptions regarding P and π), appears to rapidly decay VF estimate accuracy as problems become complex, and also makes the chattering described by Gordon (1996) particularly problematic, especially for complex environments. Observing individual trials across our experiments, it appears that SARSA-F does occasionally find strong policies, but that these are consistently lost as actions which deviate from this policy start to get distorted VF estimates.

The capacity of SARSA-P to place important states into very small cells is integral to its effectiveness. It potentially allows SARSA-P to perform a high precision search over localised regions of the policy space, whilst not being at a significant disadvantage to SARSA-F when searching (with much less precision) over the whole policy space. Furthermore, the addition of PASA creates a much greater tendency for strong policies, when they

are discovered, to remain stable (though not so stable that the agent is preventing from exploring the potential for improvements to its current policy).

Finally, as we saw in the Gridworld environments, there may be instances where the advantages of *generalisation* which arise from function approximation may be leveraged by employing PASA, without suffering to the same extent the consequences which typically come from a lack of precision in the VF estimate.

5. Conclusion

One of the key challenges currently facing RL remains understanding how to effectively extend core RL concepts and methodologies (embodied in many of the classic RL algorithms such as Q-learning and SARSA) to problems with large state or action spaces. As noted in the introduction, there is currently interest amongst researchers in methods which allow agents to learn VF approximation architectures, in the hope that this will allow agents to perform well, and with less supervision, in a wider range of environments. Whilst a number of different approaches and algorithms have been proposed towards this end, what we've termed "unsupervised" techniques of adapting approximation architectures remain relatively unexplored.

We have developed an algorithm which is an implementation of such an approach. Our theoretical analysis of the algorithm in Section 3 suggests that, in a policy evaluation setting, there are types of environment (which are likely to appear commonly in practice) in which our algorithm—and potentially such methods more generally—can on average significantly decrease error in VF estimates. This is possible despite minimal additional computational demands. Furthermore, our experiments in Section 4 suggest that this reduction in VF error can be relied upon to translate into improved performance. In our view, the theoretical and experimental evidence presented suggests that such techniques are a promising candidate for further research, and that the limited attention they have received to date may be an oversight.

Besides exploring improvements to PASA, or even alternative implementations of the principles and ideas underlying PASA, we consider that some of the more interesting avenues for further research would involve seeking to extend the same techniques to problems involving (a) large, or potentially continuous, action spaces, (b) factored Markov decision problems,[46] and (c) partially observable Markov decision problems.[47] The principles we have explored in this article suggest no obvious barriers to extending the techniques we have outlined to these more general classes of problem.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable and insightful comments.

46. Factored MDPs—see, for example, Boutilier et al. (1995)—in essence allow components of the environment to evolve independently.
47. Partially observable MDPs—see, for example, Kaelbling et al. (1998)—encompass problems where the agent does not have complete information regarding its current state $s^{(t)}$.

Table 5: Summary of results of SARSA-P and SARSA-F comparative performance.

Experiment	S	X	Comments	Aver. reward per iter. [a] SARSA-P	Aver. reward per iter. [a] SARSA-F	% incr.	Aver. μs per iter.[b] SARSA-P	Aver. μs per iter.[b] SARSA-F	% incr.
GARNET	250	70	$\zeta = 30$	90	87.8	**2.5**%	2.64	2.51	5.2%
	500	100	$\zeta = 30$	155.9	143.3	**8.8**%	2.45	2.41	1.8%
	1,000	140	$\zeta = 30$	196.8	203.5	−**3.3**%	2.5	2.41	3.4%
	2,000	200	$\zeta = 30$	301.4	240.8	**25.2**%	2.52	2.45	2.7%
	4,000	280	$\zeta = 30$	452.4	247.7	**82.6**%	2.56	2.49	2.8%
	8,000	380	$\zeta = 30$	544.6	166.7	**226.6**%	2.57	2.5	2.9%
	250	70	$\zeta = 3$	32.2	29.2	**10**%	2.43	2.39	1.7%
	500	100	$\zeta = 3$	58.9	48.8	**20.7**%	2.42	2.41	0.7%
	1,000	140	$\zeta = 3$	81.9	61.3	**33.4**%	2.48	2.42	2.3%
	2,000	200	$\zeta = 3$	148.4	65.6	**126**%	2.49	2.44	2%
	4,000	280	$\zeta = 3$	230.7	71.2	**223.9**%	2.52	2.47	2%
	8,000	380	$\zeta = 3$	264.9	30	**782.8**%	2.65	2.65	0%
Gridworld	1,024	140	$r = 24$	0.261	0.168	**55**%	2.76	2.62	5.2%
	1,024	140	$r = 8$, random trans.	0.098	0.048	**105.1**%	2.64	2.62	0.7%
	1,024	140	$r = 24$	0.066	0.003	> **2K**%	2.57	2.51	2.4%
Logistic	> 262K	140		**0.785**	−**0.019**	n/a[c]	1.95	1.8	8.3%

[a] We take the average reward per iteration over the last fifth of each trial.
[b] An estimate was generated of the microseconds required for a single iteration by both SARSA and, where relevant, PASA. The majority of experiments were run on an Intel(R) Xeon(R) CPU E5-4650 0 @ 2.70GHz for both algorithm variants. Timings are principally indicative, given that the implementations (of PASA in particular) haven't been optimised, and the measurements would have been affected by exogenous noise.
[c] Average reward in this environment (given, in particular, that it can assume negative values) doesn't lend itself to a percentage comparison.

List of symbols

$\mathcal{A}$	The action space, or set of all actions. 7
A	The size of the action space. 7
d	Value generated during update to SARSA. 13
$\mathcal{D}$	The set of atomic cells in some discretisation of a continuous state space. 36
D	The number of atomic cells in the set $\mathcal{D}$. 36
η	Learning rate for SARSA. 12
F_R	For a give state-action pair (s_i, a_j), the cumulative distribution function associated with the random variable $R(s_i, a_j)$. 7
$h(\mathcal{I}, \pi)$	The proportion of time that an agent spends in a set of states $\mathcal{I}$ when following the policy π. 28
I	The indicator function. Specifically, $I_A = 1$ if A is true. 9
L_γ	A scoring function. Estimates squared distance from VF using Bellman operator. 8
$\tilde{L}_\gamma$	A scoring function. Estimates squared distance from VF using temporal difference sampling. 8
m	A mapping from the points in a continuous state space to individual atomic cells. 36
Ψ	A probability distribution over a continuous state space. 36
MSE_γ	A scoring function. Measures mean squared difference from the true VF. 8
$\mu_{j,k}$	The sum of the stable state probabilities ψ_i of each state s_i in the cell $\mathcal{X}_{j,k}$ for a given policy π. 22
ν	PASA parameter, which determines the length of the interval between updates to ρ. 16
O	Suppose $f : D \to \mathbb{R}$ and $g : D \to \mathbb{R}$. If $f(x)$ is $O(g(x))$, or equivalently $f(x) = O(g(x))$, or further equivalently $g(x) = \Omega(f(x))$, then there exists $\alpha > 0$ and $\beta > 0$ such that $f(x) \leq \alpha g(x) + \beta$ for all $x \in D$. 3
Ω	Refer to definition of O. 39
P	The transition function, a mapping from each state-action pair to a probability distribution over $\mathcal{S}$. 7
π	The agent's policy. 7
ψ	A vector (of length S) of the stable state probability of visiting each state under a particular policy. 9
Q^π_γ	The value function for a given discount factor and fixed policy π. 7
$\hat{Q}$	An estimate of the VF stored by an RL algorithm. 8
R	The reward function, a mapping from each state-action pair to a random variable. 7

R_{m}	An assumed upper bound on the magnitude of the expected value of the reward function R. 7
ρ	Vector of integers (split vector) used by PASA to define a sequence of cell splits of an initial ordered state partition. 15
$\mathcal{S}$	The state space, or set of all states. 7
S	The size of the state space. 7
$[1{:}k]$	The set of integers j such $1 \leq j \leq k$. 21
Σ	Vector of boolean values stored by PASA to indicate whether a cell is a singleton cell. 16
T_γ^π	The Bellman operator. 8
θ	Matrix of parameters used in parametrised value function approximation. 12
ϑ	PASA parameter, which determines the threshold which must be exceeded before an update is made to ρ. 17
$\bar{u}$	Vector stored by PASA used to estimate the visit probability of each set of states in $\bar{\Xi}$. 16
u	Vector stored by PASA used to estimate the visit probability of each cell in the current partition Ξ. 16
w	Weighting used for each state-action pair when calculating total score. 8
$\tilde{w}$	Weighting used for each state-action pair when calculating total score, with additional constraints. 9
$\bar{\mathcal{X}}_j$	An element of $\bar{\Xi}$ with index j. 15
Ξ	A partition of $\mathcal{S}$, used to help define cells in a state aggregation architecture. 13
$\bar{\Xi}$	A set of subsets of $\mathcal{S}$, used by PASA to generate a partition Ξ. 15
$\mathcal{X}_k$	The cell in a state aggregation architecture with index k. 13
$\mathcal{X}_{j,k}$	In the PASA algorithm, the jth cell in the kth partition generated during the update of Ξ. 17
X	The number of basis functions in a parametrised value function approximation. Also the number of cells in a state aggregation architecture and the size of Ξ. 12
X_0	The number of initial base cells for PASA. 15

Appendix A.

A.1. Assumptions for Theorem 2.2, Chapter 8 of Kushner and Yin (2003)

The theorem relates—adopting the authors' notation, simplified somewhat, in the next equation—to stochastic processes over discrete time steps n of the form:

$$\theta_{n+1} = \Pi_H(\theta_n + \varepsilon Y_n).$$

where $\varepsilon > 0$ is a fixed constant and where Π_H is a truncation operator (note that θ as used in this subsection is distinct from θ as used in the definition of linear approximation architectures). In our case, equating n with t and ε with ς, and referring back to equation (5):

$$\theta_n \equiv \theta^{(t)} = \bar{u}_k^{(t)} \qquad \text{and} \qquad Y_n \equiv Y^{(t)} = \bar{x}_k^{(t)} - \bar{u}_k^{(t)}. \tag{8}$$

We are interested in applying the theorem in the context of Proposition 4. Note in what follows that, for all t, clearly $0 \leq \bar{u}_k^{(t)} \leq 1$ (such that $-1 \leq Y^{(t)} \leq 1$). This implies that we will be able to apply the form of Theorem 2.2 for which the trajectory of the estimate $\bar{u}_k$ is unbounded, such that we can ignore the truncation operator (i.e. treat it as the identity operator).

We address each of the relevant assumptions (using the authors' numbering) required for the result.[48] Assumption (A1.1) requires that $\{Y^{(t)}; t\}$ is uniformly integrable which holds. Assumptions (A1.2) and (A1.3) are not applicable. Assumption (A1.4) we will return to momentarily. Suppose that $\xi^{(t)}$ and $\beta^{(t)}$ are two sequences of random variables. Assumption (A1.5) requires that there exists a function g (measurable on the filtration defined by the sequence of values $\theta^{(u)}$, $Y^{(u-1)}$ and $\xi^{(u)}$ for $1 \leq u \leq t$) such that:

$$\mathrm{E}\left(Y^{(t)} \middle| \theta^{(u)}, Y^{(u-1)}, \xi^{(u)} \text{ for } 1 \leq u \leq t\right) = g\big(\theta^{(t)}, \xi^{(t)}\big) + \beta^{(t)},$$

which holds (we can ignore $\beta^{(t)}$ as in this special case it will always be zero) since the function for $Y^{(t)}$ stated above follows this form if the term $\bar{x}_k^{(t)}$ is interpreted as the random variable $\xi^{(t)}$. Our definition of $\beta^{(t)}$ implies that Assumption (A1.4), which we will not restate, trivially holds. Assumptions (A1.6)-(A1.9) we will not state in full but will be satisfied in our specific case provided g is a continuous function of $\bar{u}_k$, the sequence $\xi^{(t)}$ is bounded within a compact set, the sequence $Y^{(t)}$ is bounded, and provided:

$$\lim_{s\to\infty} \lim_{t\to\infty} \frac{1}{s} \sum_{t'=t}^{s} \mathrm{E}\Big(g\big(\theta, \xi^{(t')}\big)\Big) = \mu$$

for some μ for all $\theta \in [0, 1]$, all of which hold. Finally, assumption (A1.10) requires the sequence $\bar{u}_k$ is bounded with probability one, which is also satisfied given our observation immediately after equation (8) above.

The theorem can also be applied to demonstrate the convergence of SARSA with a fixed linear approximation architecture, fixed policy and a fixed step size parameter η. The special case of state aggregation architectures is of importance for Theorems 12 and 14, as well as Theorem 15.

48. See page 245 of Kushner and Yin (2003).

We will not examine the details, however it is relatively straightforward to examine each of the stated assumptions and demonstrate that each holds in this case. The principle differences are: (a) an additional step is required to solve the system of XA simultaneous equations formed by each of the formulae for $\theta_{kj}^{(t+1)} - \theta_{kj}^{(t)}$ for each pair of indices $1 \leq k \leq X$ and $1 \leq j \leq A$ (so as to find the limit set of the relevant ODE), (b) in the case of general linear approximation architectures, certain conditions must hold in relation to the basis functions—in particular to ensure a suitable limit set of the ODE exists, see page 45 of Kushner and Yin (2003)—which certainly do hold in the special case of state aggregation approximation architectures, and (c) caution needs to be exercised around the function R, since certain functions will violate the required assumptions, hence our assumption in Theorems 12, 14 and Theorem 15 that $|R(\cdot, a_j)|$ is uniformly bounded; weaker conditions exist which would be adequate—conditions which can be inferred from the assumptions stated in Kushner and Yin (2003) which we have referred to above.

Whilst we have not made the details explicit, an example of employing this type of approach (although here in the case where the step size parameter η is a function of t and is slowly decreased in size, and where a general linear approximation architecture is assumed) can be found at page 44 of Kushner and Yin (2003), where the authors describe the convergence of TD(λ) under quite general conditions on the state space (which, for our purposes, extend to both our discrete state space formalism and the continuous state space formalism we adopt in Section 3.4). Their discussion is readily applicable to SARSA with fixed state aggregation and fixed step sizes. See also, for a related discussion, Melo et al. (2008) and Perkins and Precup (2003).

A.2. Proof of Theorem 15

Assume that $\hat{Q}_0 = \hat{Q}_0^{(t)}$ is the VF estimate generated by SARSA with a fixed state aggregation approximation architecture corresponding to the D atomic cells (whilst $\hat{Q} = \hat{Q}^{(t)}$ is the VF estimate generate by SARSA-P).

Note that our discussion regarding the convergence of SARSA with fixed state aggregation and fixed step sizes in Appendix A.1 extends to continuous state spaces such as those described in our continuous state space formalism, provided a stable state distribution $\Psi^{(\infty)}$ exists on $\mathcal{S}$, which we've assumed. We can therefore define $\hat{Q}_{\text{lim}}$ in the same way as in Theorem 12. We can also argue in the same manner as the proof of Theorem 12 to establish that there exists T', η, ϑ and ς such that, for any $\varepsilon_1' > 0$ and $\varepsilon_2 > 0$, with probability at least $1 - \varepsilon_2$ we have $|\hat{Q}(s, a_j) - \hat{Q}_{\text{lim}}(s, a_j)| \leq \varepsilon_1'$ for every $s \in \mathcal{S}$ and $1 \leq j \leq A$ whenever $t > T'$. We can argue equivalently for $\hat{Q}_0$, in relation to a separate limit $\hat{Q}_{\text{lim},0}$, such that T'' and η'' exist so that $|\hat{Q}_0(s, a_j) - \hat{Q}_{\text{lim},0}(s, a_j)| \leq \varepsilon_1'$ holds for $t > T''$ with probability at least $1 - \varepsilon_2$. In this way there will exist T such that both inequalities will hold, with probability at least $1 - \varepsilon_2$, for $t > T$.

We now examine $L - L_0$.[49] We define $B = B(s, a_j)$ as follows:

$$B(s, a_j) \coloneqq \mathrm{E}\left(R\big(s^{(t)}, a^{(t)}\big) + \gamma \hat{Q}_{\text{lim}}\big(s^{(t+1)}, a^{(t+1)}\big) \Big| s^{(t)} = s, a^{(t)} = a_j\right) - \hat{Q}_{\text{lim}}(s, a_j).$$

And we define:

$$\begin{aligned} C(s, a_j, d_i) \coloneqq\ & \mathrm{E}\left(R\big(s^{(t)}, a^{(t)}\big) + \gamma \hat{Q}_{\text{lim}}\big(s^{(t+1)}, a^{(t+1)}\big) \Big| s^{(t)} = s, a^{(t)} = a_j\right) \\ & - \mathrm{E}\left(R\big(s^{(t)}, a^{(t)}\big) + \gamma \hat{Q}_{\text{lim}}\big(s^{(t+1)}, a^{(t+1)}\big) \Big| s^{(t)} \in d_i, a^{(t)} = a_j\right), \end{aligned}$$

where the expectation in the second term is over $\Psi^{(\infty)}$ (as well as over the distributions of R and P). We will have, for $t > T$:

$$\begin{aligned} L &\leq \int_{\mathcal{S}} \sum_{j=1}^{A} \tilde{w}(m(s), a_j) B(s, a_j)^2 \, d\Psi^{(\infty)}(s) + \varepsilon_1 \\ &\leq \frac{4(1-h)}{(1-\gamma)^2} R_{\text{m}}^2 + \sum_{i: d_i \in \mathcal{I}} \int_{d_i} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) C(s, a_j, d_i)^2 \, d\Psi^{(\infty)}(s) + \varepsilon_1, \end{aligned}$$

where for any ε_1 we can select ε_1' so that the inequality holds with probability at least $(1 - \varepsilon_2)^{1/2}$.

By equivalent reasoning, we will have exactly the same inequality for L_0, with $\hat{Q}_{\text{lim}}$ replaced by $\hat{Q}_{\text{lim},0}$ (due to the fact that the VF architecture for $\hat{Q}$ and $\hat{Q}_0$ are the same over the set $\mathcal{I}$). We will use the following shorthand (as used in the proof of Theorem 12): $\lambda = (1 - \delta_P)(1 - \delta_\pi)$. We define the event K as being the event that the agent transitions, after iteration t, to the "deterministic" next atomic cell d'' and action a'' (i.e. the agent transitions according to the transition function P_1 and policy π_1, the latter of which exists and is defined according to the definition in Section 3.3, and the former of which can be easily defined by extending the relevant Section 3.3 definition to transitions between atomic cells, consistent with the definition of δ-deterministic in the context of continuous state spaces). The value $\hat{Q}_{\text{lim}}$ associated with the pair (d'', a'') we denote as $\hat{Q}_{\text{lim}}''$. We will have:

$$\begin{aligned} C(s, a_j, d_i) \leq\ & \underbrace{\Big| \mathrm{E}\big(R(s^{(t)}, a_j) \big| s^{(t)} = s, a^{(t)} = a_j\big) - \mathrm{E}\big(R(s^{(t)}, a_j) \big| s^{(t)} \in d_i, a^{(t)} = a_j\big) \Big|}_{=:F} \\ & + \underbrace{\Big| \mathrm{P}\big(K \big| s^{(t)} = s, a^{(t)} = a_j\big) - \lambda \Big| \, \gamma |\hat{Q}_{\text{lim}}''|}_{=:G} \\ & + \underbrace{\mathrm{E}\left(I_{K^c} \gamma \big| \hat{Q}_{\text{lim}}(s^{(t+1)}, a^{(t+1)}) \big| \Big| s^{(t)} = s, a^{(t)} = a_j\right) + \gamma(1 - \lambda)|M|}_{=:H}, \end{aligned}$$

49. This bound differs the most of the three bounds from the discrete case. The reason for the additional terms is principally technical. Namely, we at no point guarantee that the value of $\hat{Q}_{\text{lim},0}$ is "close to" the value of $\hat{Q}_{\text{lim}}$, which demands that we must constrain the state-action pairs which occur as a result of a certain action in a cell contained in $\mathcal{I}$, in order to ensure that $L - L_0$ is small. The other noticeable difference from the bounds in Theorem 12 is the removal of δ_R from the bound relating to $\tilde{L}$. In the continuous case this term cancels since the bound pertains to $\tilde{L} - \tilde{L}_0$.

where $M = M(a_j, d_i)$ is a residual term (the magnitude of which we will bound in our discussion below). We need to evaluate terms in the right hand side of the following inequality:

$$\int_{d_i} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) C(s, a_j, d_i)^2 \, d\Psi^{(\infty)}(s)$$
$$\leq \int_{d_i} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) \left(F^2 + G^2 + H^2 + 2FG + 2FH + 2GH \right) \, d\Psi^{(\infty)}(s).$$

Our strategy is to argue that all of the terms involving F, G and H, with the exception of F^2, will become small provided λ is close to one. The value of F^2, moreover, will be identical for L_0 for every $d_i \in \mathcal{I}$. This will allow us to bound $L - L_0$. Note that terms involving G will be small if λ is near one because in such a case the VF estimate of $\hat{Q}_{\text{lim}}$ (for the initial state-action pair) closely resembles the temporal difference observed when the event K occurs. Terms involving H will be small if λ is near one because this implies that instances where the VF estimate does *not* closely resemble the temporal difference are rare. We will work through the terms in sequence. First G^2. We denote $J \coloneqq \mathrm{P}(K|s^{(t)} = s, a^{(t)} = a_j)$:

$$\int_{d_i} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) G^2 \, d\Psi^{(\infty)}(s) = \sum_{j=1}^{A} \tilde{w}(d_i, a_j) \int_{d_i} \left(J^2 - 2J\lambda + \lambda^2 \right) (\gamma \hat{Q}''_{\text{lim}})^2 \, d\Psi^{(\infty)}(s)$$
$$\leq \sum_{j=1}^{A} \tilde{w}(d_i, a_j) \psi_i \gamma^2 \frac{\lambda - \lambda^2}{(1-\gamma)^2} R_{\text{m}}^2 \leq \psi_i \gamma^2 \frac{\lambda - \lambda^2}{(1-\gamma)^2} R_{\text{m}}^2,$$

where in the first inequality we use the fact that $\psi_i \lambda^2 \leq \int_{d_i} \mathrm{P}(K|s^{(t)} = s)^2 \, d\Psi^{(\infty)}(s) \leq \psi_i \lambda$ for all i. Note the manner in which $\tilde{w}(d_i, a_j)$ was brought outside the integral (the properties of the integrand of course permit this). This can be done for every term, so we omit this step from now on.

For H^2, using $N \coloneqq \hat{Q}_{\text{lim}}(s^{(t+1)}, a^{(t+1)})$, and with the understanding that all expectations are conditioned on $s^{(t)} = s$ and $a^{(t)} = a_j$:

$$\int_{d_i} H^2 \, d\Psi^{(\infty)}(s)$$
$$= \int_{d_i} \left(\mathrm{E}(I_{K^c} \gamma |N|)^2 + 2\mathrm{E}(I_{K^c} \gamma |N|) \gamma (1-\lambda) |M| + \gamma^2 (1-\lambda)^2 M^2 \right) \, d\Psi^{(\infty)}(s)$$
$$\leq \frac{\gamma^2}{(1-\gamma)^2} R_{\text{m}}^2 \int_{d_i} \left(\mathrm{E}(I_{K^c})^2 + 2(1-\lambda)\mathrm{E}(I_{K^c}) + (1-\lambda)^2 \right) \, d\Psi^{(\infty)}(s)$$
$$\leq \frac{\gamma^2}{(1-\gamma)^2} R_{\text{m}}^2 \left((1-\lambda) + 2(1-\lambda)^2 + (1-\lambda)^2 \right) \leq \psi_i \gamma^2 \frac{4(1-\lambda)}{(1-\gamma)^2} R_{\text{m}}^2.$$

Similarly (using $|P(K') - \lambda| = P(K')(1-\lambda) + (1 - P(K'))\lambda$ for any event K'):

$$\int_{d_i} FG \, d\Psi^{(\infty)}(s) \leq \gamma \frac{2}{1-\gamma} R_{\text{m}}^2 \int_{d_i} \left| \mathrm{P}\left(K \middle| s^{(t)} = s, a^{(t)} = a_j\right) - \lambda \right| \, d\Psi^{(\infty)}(s)$$
$$\leq \psi_i \gamma \frac{4\lambda(1-\lambda)}{1-\gamma} R_{\text{m}}^2$$

and:

$$\int_{d_i} FH \, d\Psi^{(\infty)}(s) \leq \gamma \frac{2}{1-\gamma} R_{\text{m}}^2 \int_{d_i} \left(\mathrm{E}\left(I_{K^c} \middle| s^{(t)} = s, a^{(t)} = a_j\right) + (1-\lambda) \right) \, d\Psi^{(\infty)}(s)$$
$$\leq \psi_i \gamma \frac{4(1-\lambda)}{1-\gamma} R_{\text{m}}^2.$$

Finally, using the fact that $|\mathrm{P}(K|s^{(t)} = s, a^{(t)} = a_j) - \lambda| \leq 1$ for all j, and again with the understanding that all expectations are conditioned on $s^{(t)} = s$ and $a^{(t)} = a_j$:

$$\int_{d_i} GH \, d\Psi^{(\infty)}(s) \leq \gamma^2 \frac{2}{(1-\gamma)^2} R_{\text{m}}^2 \int_{d_i} |\mathrm{P}(K) - \lambda| \left(\mathrm{E}(I_{K^c}) + (1-\lambda) \right) \, d\Psi^{(\infty)}(s)$$
$$\leq \gamma^2 \frac{2}{(1-\gamma)^2} R_{\text{m}}^2 \int_{d_i} \left(\mathrm{E}(I_{K^c}) + (1-\lambda) \right) \, d\Psi^{(\infty)}(s) \leq \psi_i \gamma^2 \frac{4(1-\lambda)}{(1-\gamma)^2} R_{\text{m}}^2.$$

To simplify the inequality we multiply some terms by $1/(1-\gamma) \geq 1$, and replace λ and γ by one for others. This gives us, for each $d_i \in \mathcal{I}$:

$$\int_{d_i} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) C(s, a_j, d_i)^2 \, d\Psi^{(\infty)}(s) \leq \psi_i \frac{29(1-\lambda)}{(1-\gamma)^2} R_{\text{m}}^2 + \int_{d_i} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) F^2 \, d\Psi^{(\infty)}(s).$$

We can argue in an identical fashion for each $d_i \in \mathcal{I}$ for L_0, and obtain the same inequality (where F^2 will be an identical function of s). Therefore, for $t > T$, with probability at least $1 - \varepsilon_2$:

$$L - L_0 \leq \frac{4(1-h)}{(1-\gamma)^2} R_{\text{m}}^2 + \varepsilon_1 + \frac{29(1-\lambda)}{(1-\gamma)^2} R_{\text{m}}^2$$
$$\leq \frac{4(1-h) + 29(1 - (1-\delta_P)(1-\delta_\pi))}{(1-\gamma)^2} R_{\text{m}}^2 + \varepsilon_1,$$

where we have used the fact that each term in both L and L_0 which does not cancel must be greater than zero (such that coefficients are not doubled, they are the same as for L).

We now consider $\tilde{L} - \tilde{L}_0$. We define $\tilde{B}(s, a_j)$ as:

$$\tilde{B}(s, a_j) \coloneqq \mathrm{E}\left(R\left(s^{(t)}, a_j\right) + \gamma \hat{Q}_{\text{lim}}\left(s^{(t+1)}, a^{(t+1)}\right) - \hat{Q}_{\text{lim}}(s, a_j) \middle| s^{(t)} = s, a^{(t)} = a_j \right)$$

Using this definition, for any $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ we can select ε'_1 (defined above) such that for all $t > T$, with probability at least $(1 - \varepsilon_2)^{1/2}$, we have:

$$\tilde{L} \leq \int_{\mathcal{S}} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) \tilde{B}(s, a_j)^2 \, d\Psi^{(\infty)}(s) + \varepsilon_1$$
$$\leq \frac{4(1-h)}{(1-\gamma)^2} R_{\text{m}}^2 + \varepsilon_1 + \sum_{i: d_i \in \mathcal{I}} \sum_{j=1}^{A} \tilde{w}(d_i, a_j) \int_{d_i} \tilde{B}(s, a_j)^2 \, d\Psi^{(\infty)}(s)$$

Suppose we denote $\mathrm{E}_{d_i}(f(s)) \coloneqq \int_{d_i} f(s) \, d\Psi^{(\infty)}(s)$. The next step is much like that in the proof of the bound on $\tilde{L}$ in Theorem 12—see equation (7). Again we use the shorthand

$N = \hat{Q}_{\lim}(s^{(t+1)}, a^{(t+1)})$. For any d_i we can write, cancelling relevant terms:

$$\begin{aligned}
&\mathrm{E}_{d_i}\left(\tilde{B}(s,a_j)^2\right) \\
&= \underbrace{\mathrm{E}_{d_i}\left\{\mathrm{E}\big(R(s^{(t)},a_j)\big|s^{(t)}=s,a^{(t)}=a_j\big)^2\right\} - \left\{\mathrm{E}_{d_i}\mathrm{E}\big(R(s^{(t)},a_j)\big|s^{(t)}=s,a^{(t)}=a_j\big)\right\}^2}_{=:\tilde{A}} \\
&\quad + \underbrace{\gamma^2\mathrm{E}_{d_i}\left\{\mathrm{E}\big(N\big|s^{(t)}=s,a^{(t)}=a_j\big)^2\right\} - \gamma^2\left\{\mathrm{E}_{d_i}\mathrm{E}\big(N\big|s^{(t)}=s,a^{(t)}=a_j\big)\right\}^2}_{=:\tilde{C}}.
\end{aligned}$$

We can define $\tilde{B}_0$ in the same way as $\tilde{B}$, however with $\hat{Q}_{\lim}$ replaced by $\hat{Q}_{\lim,0}$. Proceeding via identical arguments we have $\mathrm{E}_{d_i}(\tilde{B}_0(s,a_j)^2) = \tilde{A} + \tilde{C}_0$, where $\tilde{C}_0$ is the same as $\tilde{C}$ however with $\hat{Q}_{\lim}$ replaced by $\hat{Q}_{\lim,0}$. We can reason in an identical manner to our proof of the bound on $\tilde{L}$ in Theorem 12 to conclude that:

$$|\tilde{C}| \leq \psi_i\gamma^2\frac{1+2(1-\delta_P)(1-\delta_\pi)-3(1-\delta_P)^2(1-\delta_\pi)^2}{(1-\gamma)^2}R_{\mathrm{m}}^2.$$

The same holds for $|\tilde{C}_0|$ of course. Hence we will have, with probability at least $1-\varepsilon_2$:

$$\begin{aligned}
\tilde{L} - \tilde{L}_0 &\leq \frac{4(1-h)}{(1-\gamma)^2}R_{\mathrm{m}}^2 + \varepsilon_1 + \sum_{j=1}^{A}\tilde{w}(d_i,a_j)\int_{\mathcal{S}}\left(\tilde{B}(s,a_j)^2 - \tilde{B}_0(s,a_j)^2\right)\,d\Psi^{(\infty)}(s) \\
&= \frac{4(1-h)}{(1-\gamma)^2}R_{\mathrm{m}}^2 + \varepsilon_1 + \sum_{j=1}^{A}\tilde{w}(d_i,a_j)\sum_{i=1}^{D}(\tilde{C}-\tilde{C}_0) \\
&\leq \frac{4(1-h)}{(1-\gamma)^2}R_{\mathrm{m}}^2 + \varepsilon_1 + \gamma^2\frac{2+4(1-\delta_P)(1-\delta_\pi)-6(1-\delta_P)^2(1-\delta_\pi)^2}{(1-\gamma)^2}R_{\mathrm{m}}^2 \\
&= \left(2(1-h)+\gamma^2\left(1+2(1-\delta_P)(1-\delta_\pi)-3(1-\delta_P)^2(1-\delta_\pi)^2\right)\right)\frac{2R_{\mathrm{m}}^2}{(1-\gamma)^2} + \varepsilon_1.
\end{aligned}$$

Finally we turn to $\mathrm{MSE}-\mathrm{MSE}_0$. As in the discussion for the other two scoring functions, there exists T such that, in this case, for $t > T$:

$$\begin{aligned}
\mathrm{MSE} &= \int_{\mathcal{S}}\sum_{j=1}^{A}\pi(a_j|s)\left(Q^\pi(s,a_j) - \hat{Q}(s,a_j)\right)^2\,d\Psi^{(\infty)}(s) \\
&\leq \int_{\mathcal{S}}\sum_{j=1}^{A}\pi(a_j|s)\left(Q^\pi(s,a_j) - \hat{Q}_{\lim}(s,a_j)\right)^2\,d\Psi^{(\infty)}(s) + \varepsilon_1
\end{aligned}$$

where for any ε_1 we can select ε_1' (defined above) so that the inequality holds with probability at least $(1-\varepsilon_2)^{1/2}$. We can write:

$$\mathrm{MSE} \leq \frac{4(1-h)}{(1-\gamma)^2}R_{\mathrm{m}}^2 + \sum_{i=1}^{D}\int_{d_i}\sum_{j=1}^{A}\pi(a_j|s)\left(Q^\pi(s,a_j) - \hat{Q}_{\lim}(s,a_j)\right)^2\,d\Psi^{(\infty)}(s) + \varepsilon_1 \qquad (9)$$

We now argue in a similar fashion to Theorem 12. Suppose we have some pair (s,a_j) such that $s \in d$ and $d \in \mathcal{I}$. We can separate the value $Q^\pi - \hat{Q}_{\lim}$ into those sequences of states and actions which remain in $\mathcal{I}$ and those which leave $\mathcal{I}$ at some point. We use the values $\xi^{(t')}$ and $\chi^{(t')}$ to represent the expected discounted reward obtained after exactly t' iterations conditioned on $a^{(1)} = a_j$, further conditioned on $s^{(t)} = s$ and $s^{(t)} \in d$ for some $d \in \mathcal{D}$ (with $s^{(t)}$ otherwise selected according to the distribution $\Psi^{(\infty)}$) respectively, and conditioned upon the agent remaining in $\mathcal{I}$ for all $t'' \leq t'$. We will have, for the pair (s,a_j):

$$\begin{aligned}
Q^\pi(s,a_j) &= \underbrace{\xi^{(1)} + \sum_{t'=2}^{\infty}\Pr\left(s^{(t'')} \in m^{-1}(\mathcal{I}) \text{ for } t'' \leq t' \middle| s^{(1)}=s, a^{(1)}=a_j\right)\xi^{(t')}}_{=:C_s} \\
&\quad + \underbrace{\Pr\left(s^{(2)} \notin m^{-1}(\mathcal{I})\middle|s^{(1)}=s,a^{(1)}=a_j\right)x^{(2)}}_{=:U_s} \\
&\quad + \underbrace{\sum_{t'=3}^{\infty}\Pr\left(s^{(t'')} \in m^{-1}(\mathcal{I}) \text{ for } t'' < t', s^{(t')} \notin m^{-1}(\mathcal{I})\middle|s^{(1)}=s,a^{(1)}=a_j\right)x^{(t')}}_{=:V_s}
\end{aligned}$$

And similarly, assuming s is distributed according to $\Psi^{(\infty)}$:

$$\begin{aligned}
\hat{Q}_{\lim}(s,a_j) &= \underbrace{\chi^{(1)} + \sum_{t'=2}^{\infty}\Pr\left(s^{(t'')} \in m^{-1}(\mathcal{I}) \text{ for } t'' \leq t' \middle| d^{(1)}=m(s), a^{(1)}=a_j\right)\chi^{(t')}}_{=:C_d} \\
&\quad + \underbrace{\Pr\left(s^{(2)} \notin m^{-1}(\mathcal{I})\middle|d^{(1)}=m(s),a^{(1)}=a_j\right)x'^{(2)}}_{=:U_d} \\
&\quad + \underbrace{\sum_{t'=3}^{\infty}\Pr\left(s^{(t'')} \in m^{-1}(\mathcal{I}) \text{ for } t'' < t', s^{(t')} \notin m^{-1}(\mathcal{I})\middle|d^{(1)}=m(s),a^{(1)}=a_j\right)x'^{(t')}}_{=:V_d}
\end{aligned}$$

Note that the values $x^{(t')}$ and $x'^{(t')}$ are, as was the case in Theorem 12, residual terms which reflect the expected total discounted future reward once the agent has left $\mathcal{I}$ for the first time. They are also conditioned on $a^{(1)} = a_j$, and conditioned on $s^{(t)} = s$ and $s^{(t)} \in d$ (with $s^{(t)}$ otherwise selected according to the distribution $\Psi^{(\infty)}$) respectively. Unlike ξ and χ, they reflect the sum of expected discounted reward over an infinite number of iterations. The significance of these values are that we are able to bound them, which we do below. The reason that $x^{(2)}$ and $x'^{(2)}$ are expressed separately as V_s and V_d respectively in the expression above is because (similar to the case in Theorem 12) the first action a_j is not guaranteed to be chosen according to the policy π. We now have:

$$\begin{aligned}
\left(Q^\pi(s,a_j) - \hat{Q}_{\lim}(s,a_j)\right)^2 &= (C_s - C_d + U_s - U_d + V_s - V_d)^2 \\
&= C_s^2 + C_d^2 + U_s^2 + U_d^2 + V_s^2 + V_d^2 - 2C_sC_d + \ldots
\end{aligned} \qquad (10)$$

Our strategy, similar to the case for L, will be to show that, in our formula for MSE, any term involving U_s, U_d, V_s or V_d will be small (provided $\delta_{\mathcal{I}}$ is near zero). The remaining terms (which will involve some combination of C_s and C_d only) will cancel when we consider MSE_0. Considering the quantity $|U_s|$, we will have:

$$\begin{aligned}
&\int_{d_i} \sum_{j=1}^{A} \pi(a_j|s)|U_s| \, d\Psi^{(\infty)}(s) \\
&= \int_{d_i} \sum_{j=1}^{A} \pi(a_j|s) \Pr\left(s^{(2)} \notin m^{-1}(\mathcal{I}) \middle| s^{(1)} = s, a^{(1)} = a_j\right) \left|x^{(2)}\right| d\Psi^{(\infty)}(s) \\
&\leq \frac{R_{\text{m}}}{1-\gamma} \int_{d_i} \sum_{j=1}^{A} \pi(a_j|s) \Pr\left(s^{(2)} \notin m^{-1}(\mathcal{I}) \middle| s^{(1)} = s, a^{(1)} = a_j\right) d\Psi^{(\infty)}(s) \leq \psi_i \frac{\delta_{\mathcal{I}} R_{\text{m}}}{1-\gamma}.
\end{aligned}$$

The argument is substantially the same for $|U_d|$ (the key difference being that the integrand in this latter case is independent of s). Now, focussing on $|V_s|$, we note that:[50]

$$\begin{aligned}
&\int_{d_i} \sum_{j=1}^{A} \pi(a_j|s)|V_s| \, d\Psi^{(\infty)}(s) \\
&\leq \int_{d_i} \sum_{j=1}^{A} \pi(a_j|s) \sum_{t'=3}^{\infty} \Pr\left(s^{(t')} \notin m^{-1}(\mathcal{I}) \middle| s^{(t'-1)} \in m^{-1}(\mathcal{I})\right) \underbrace{\left|x^{(t')}\right|}_{\leq \gamma^{t'} \frac{R_{\text{m}}}{1-\gamma}} d\Psi^{(\infty)}(s) \\
&\leq \frac{R_{\text{m}}}{1-\gamma} \sum_{t'=3}^{\infty} \gamma^{t'} \int_{d_i} \Pr\left(s^{(t')} \notin m^{-1}(\mathcal{I}) \middle| s^{(t'-1)} \in m^{-1}(\mathcal{I})\right) \sum_{j=1}^{A} \pi(a_j|s) \, d\Psi^{(\infty)}(s) \\
&\leq \psi_i \frac{R_{\text{m}}}{1-\gamma} \sum_{t'=3}^{\infty} \gamma^{t'} \delta_{\mathcal{I}} \leq \psi_i \frac{\delta_{\mathcal{I}} R_{\text{m}}}{(1-\gamma)^2}.
\end{aligned}$$

(Note that, compared to the discrete case arguments in Theorem 12, we have deliberately simplified this inequality by loosening the bound more than is strictly required, to make both the arguments and the equations more succinct.) The same bound is true for $|V_d|$ (again using a substantially equivalent argument). Since each of C_s, C_d, U_s, U_d, V_s and V_d must be bound (at least) by $R_{\text{m}}/(1-\gamma)$, the final contribution to MSE of every term in the last line of (10) which contains U_s, U_d, V_s or V_d is bound by $\delta_{\mathcal{I}} R_{\text{m}}^2/(1-\gamma)^3$. Now we simply observe that, for MSE_0, we will have the same equation as (9), except that U_s, U_d, V_s and V_d are replaced by the distinct values $U_{s,0}$, $U_{d,0}$, $V_{s,0}$ and $V_{d,0}$ respectively (all are subject to the same bounds discussed for MSE). The values C_s and C_d are the same for MSE_0. Not including C_s^2, C_d^2 and $C_s C_d$ there are 33 terms (including duplicates) in the expansion in (10). This gives us, with probability at least $1 - \varepsilon_2$:

$$\text{MSE} - \text{MSE}_0 \leq \left(4(1-h) + \frac{33\delta_{\mathcal{I}}}{1-\gamma}\right) \frac{R_{\text{m}}^2}{(1-\gamma)^2} + \varepsilon_1.$$

50. The properties of the integrand allow us to switch the order of the infinite sum and the integral in the third line.

A.3. Proof of Lemma 16

An introductory discussion will help us establish some of the concepts required for the proof. We can make the following observation. If π and P are deterministic, and we pick a starting state s_1, then the agent will create a path through the state space and will eventually revisit a previously visited state, and will then enter a cycle. Call the set of states in this path (including the cycle) $\mathcal{L}_1$ and call the set of states in the cycle $\mathcal{C}_1$. Denote as L_1 and C_1 the number of states in the path (including the cycle) and the cycle respectively. Of course $L_1 \geq C_1 \geq 1$.

If we now place the agent in a state s_2 (arbitrarily chosen) it will either create a new cycle or it will terminate on the path or cycle created from s_1. Call $\mathcal{L}_2$ and $\mathcal{C}_2$ the states in the second path and cycle respectively (and L_2 and C_2 the respective numbers of states, noting that $C_2 = 0$ is possible if the new path terminates on $\mathcal{L}_1$, and in fact that $L_2 = C_2 = 0$ is also possible, if $s_2 \in \mathcal{L}_1$). If we continue in this manner we will have S sets $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_S\}$. Call $\mathcal{C}$ the union of these sets and denote as C the number of states in $\mathcal{C}$. We denote as J_i the event that the ith path created in such a manner terminates on itself, and note that, if this does not occur, then $C_i = 0$. For the next result we continue to assume that π and P are deterministic and also that condition (2) in Example 5 holds.

Lemma 18 $\text{E}(C_1) = \sqrt{\pi S/8} + O(1)$ *and* $\text{Var}(C_1) = (32 - 8\pi)S/24 + O(\sqrt{S})$.

Proof Choose any state s_1. We must have (where probability is conditioned on the prior distribution for P):

$$\Pr(C_1 = i, L_1 = j) = \frac{S-1}{S}\frac{S-2}{S}\cdots\frac{S-j+1}{S}\frac{1}{S} = \frac{(S-1)!}{S^j(S-j)!}.$$

This means that, for large S, the expected value of C_1 can be approximately expressed, making use of Stirling's approximation $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, as:

$$\begin{aligned}
\text{E}(C_1) &= \sum_{j=1}^{S} j \sum_{k=j}^{S} \frac{(S-1)!}{S^k(S-k)!} = \sum_{j=1}^{S} \frac{j(j+1)}{2} \frac{(S-1)!}{S^j(S-j)!} \\
&= \frac{(S-1)!}{2} \sum_{j=1}^{S} \frac{(S-j+1)(S-j+2)}{S^{S-j+1}(j-1)!} \\
&= \frac{S!}{2S^{S+1}} \left(\sum_{j=0}^{S-1} \frac{(S-j)^2 S^j}{j!} + \sum_{j=0}^{S-1} \frac{(S-j)S^j}{j!} \right) \\
&= \frac{\sqrt{2\pi S}\left(\frac{S}{e}\right)^S}{2S^{S+1}} \left(\frac{Se^S}{2} + O\left(\sqrt{S}e^S\right) \right) = \sqrt{\frac{\pi}{8}} S + O(1).
\end{aligned}$$

We have used the fact that a Poisson distribution with parameter S will, as S becomes sufficiently large, be well approximated by a normal distribution with mean S and standard deviation $\sqrt{S}$. In this case we are taking the expectation of the negative of the distance from the mean, $S - j$, over the interval from zero to $-S + 1$. Hence we can replace the first and second sum in the third equality by the second and first raw moment respectively of a

half normal distribution with variance S. The second moment of a half normal is half the variance of the underlying normal distribution, here $S/2$. (The error associated with the Stirling approximation is less than order 1.)

Similarly for the variance, we first calculate the expectation of C_1^2:

$$\begin{aligned}\mathrm{E}(C_1^2) &= \sum_{j=1}^{S} j^2 \sum_{k=j}^{S} \frac{(S-1)!}{S^k(S-k)!} = \sum_{j=1}^{S}\left(\frac{j^3}{3}+\frac{j^2}{2}+\frac{j}{6}\right)\frac{(S-1)!}{S^j(S-j)!}\\ &= \frac{S!}{S^{S+1}}\sum_{j=0}^{S-1}\left(\frac{(S-j)^3}{3}+\frac{(S-j)^2}{2}+\frac{S-j}{6}\right)\frac{S^j}{j!}\\ &= \frac{\sqrt{2\pi S}\left(\frac{S}{e}\right)^S}{S^{S+1}}\left(\sqrt{\frac{2}{\pi}}\frac{2S^{\frac{3}{2}}}{3}e^S + O\left(Se^S\right)\right) = \frac{4}{3}S + O(\sqrt{S}).\end{aligned}$$

As a result:

$$\mathrm{Var}(C_1) = \frac{4}{3}S - \frac{\pi S}{8} + O(\sqrt{S}) = \left(\frac{32-8\pi}{24}\right)S + O(\sqrt{S}).$$

■

Note that the expectation can also be derived from the solution to the "birthday problem": the solution to the birthday problem[51] gives the expectation of L_1, and since each cycle length (less than or equal to L_1) has equal probability when conditioned on this total path length, we can divide the average by 2. Maintaining our assumptions from the previous result we have:

Lemma 19 $\mathrm{E}(C) < \mathrm{E}(C_1)(\ln S + 1)$ *and* $\mathrm{Var}(C) = O(S \ln S)$.

Proof We will have:

$$\begin{aligned}\mathrm{E}(C) &= \sum_{i=1}^{S}\mathrm{E}(C_i) = \sum_{i=1}^{S}\Pr(J_i)\sum_{j=1}^{S} j\Pr(C_i = j|J_i)\\ &\le \sum_{i=1}^{S}\frac{1}{i}\sum_{j=1}^{S} j\Pr(C_1 = j) < \mathrm{E}(C_1)(\ln S + 1).\end{aligned}$$

51. For a description of the problem and a formal proof see, for example, page 114 of Flajolet and Sedgewick (2009).

And for the variance:

$$\begin{aligned}\mathrm{Var}(C) &= \sum_{i=1}^{S}\mathrm{Var}(C_i) + 2\sum_{i=2}^{S}\sum_{j=1}^{i-1}\mathrm{Cov}(C_iC_j) \le \sum_{i=1}^{S}\mathrm{Var}(C_i)\\ &\le \sum_{i=1}^{S}\mathrm{E}(C_i^2) = \sum_{i=1}^{S}\Pr(J_i)\sum_{j=1}^{S} j^2\Pr(C_i = j|J_i)\\ &\le \sum_{i=1}^{S}\frac{1}{i}\sum_{j=1}^{S} j^2\Pr(C_1 = j) < \mathrm{E}(C_1^2)(\ln S + 1)\\ &= \left(\mathrm{Var}(C_1) + \mathrm{E}(C_1)^2\right)(\ln S + 1),\end{aligned}$$

where we have used the fact that the covariance term must be negative for any pair of lengths C_i and C_j, since if C_i is greater than its mean the expected length of C_j must decrease, and vice versa. We can substitute $\Pr(J_i)$ with $1/i$ in the sum based on the reasoning that, in the absence of any information regarding $\mathcal{L}_j$ for $1 \le j < i$, we must have $\Pr(L_j \ge k) \ge \Pr(L_i \ge k)$ for all j and $k \ge 0$, which implies that there is at least as much chance of the path $\mathcal{L}_i$ terminating on any other already generated path as of self-terminating for all i, so that $1/i$ is an upper bound on $\Pr(J_i)$. ■

Thus far the definitions of this subsection only apply to deterministic P and π. We can extend all the definitions to arbitrary P and π as follows. Define $P_{\mathrm{det}} = P_{\mathrm{det}}(P)$ as follows:

$$P_{\mathrm{det}}(s_{i'}|s_i, a_j) := \begin{cases}1 & \text{if } i' = \min\{i'' : P(s_{i''}|s_i, a_j) = \max_{i'''} P(s_{i'''}|s_i, a_j)\}\\ 0 & \text{otherwise}\end{cases}$$

and $\pi_{\mathrm{det}} = \pi_{\mathrm{det}}(\pi)$ as follows:

$$\pi_{\mathrm{det}}(a_j|s_i) := \begin{cases}1 & \text{if } j = \min\{j' : \pi(a_{j'}|s_i) = \max_{j''} \pi(a_{j''}|s_i)\}\\ 0 & \text{otherwise}\end{cases}$$

Both of these distributions can be most easily interpreted by considering them to be deterministic versions of their arguments, where the most probable transition or action of the argument is taken to be the deterministic transition or action (with, as an arbitrary rule, the lowest index chosen in case of more than one transition or action being equally most-probable). We now extend all relevant definitions introduced in this subsection such that, for example, $\mathcal{C}(P) := \mathcal{C}(P_{\mathrm{det}}(P))$. Our definitions can now be applied to pairs (P, π) as defined in Example 5. We can also now prove Lemma 16:

Proof of Lemma 16 Using Chebyshev's inequality, and Lemmas 18 and 19, for any $K > 1$ and ε_2 satisfying $0 < \varepsilon_2 < K-1$, we can choose S sufficiently high so that $C > K\sqrt{\pi S/8}\ln S$ with probability no greater than $1/(K - \varepsilon_2 - 1)\ln S$. To see this, take $Y := \sqrt{\pi S/8}\ln S$, $\mu := \mathrm{E}(C)$ and $\sigma := \sqrt{\mathrm{Var}(C)}$. Note that for any $\varepsilon' > 0$ we can obtain $\mu \le (1+\varepsilon')Y$ for sufficiently large S. Similarly for any $\varepsilon'' > 0$ we can obtain $(1+\varepsilon'')Y \ge \sqrt{\ln S}\sigma$ for

sufficiently large S. We have:

$$\begin{aligned}\Pr(C > KY) &\leq \Pr\left(|C-\mu| > KY-\mu\right) \leq \Pr\left(|C-\mu| > (K-\varepsilon'-1)Y\right)\\ &\leq \Pr\left(|C-\mu| > \frac{(K-\varepsilon'-1)\sqrt{\ln S}\sigma}{1+\varepsilon''}\right)\\ &\leq \frac{(1+\varepsilon'')^2}{(K-\varepsilon'-1)^2\ln S} \leq \frac{1}{(K-\varepsilon_2-1)^2\ln S},\end{aligned}$$

where in the first and second inequalities we assume S is sufficiently large so that all but the highest order terms in Lemma 19 can be ignored, and noting that, for any $\varepsilon_2 > 0$ and $K > 1$, we can find $\varepsilon' > 0$ and $\varepsilon'' > 0$ so that the final inequality is satisfied.

We now observe that, if conditions (1) and (3) stated in Example 5 hold, $\psi_i \leq 2\delta$ for all i for which $s_i \notin \mathcal{C}$. This is because ψ_i will be less than or equal to the probability of the agent transitioning from any state $s_{i''}$ to any state $s_{i'}$ where: (a) $s_{i'}$ is not the state which $s_{i''}$ transitions to as dictated by P_{det} and π_{det}, and (b) $s_{i'}$, according to P_{det} and π_{det}, will eventually transition (after any number of iterations) to s_i. This probability is less than or equal to 2δ (the factor of two arises since the agent may transition in a way which is not dictated by P_{det} and π_{det} either due to randomness in P or randomness in π). As a result for any given S we can set δ sufficiently low so that $\sum_{i:s_i\in\mathcal{C}}\psi_i$ is arbitrarily close to one uniformly over the possible values of C. If we define $\mathcal{I}$ as the set of states in $\mathcal{C}$, then this set $\mathcal{I}$ will have the three properties required for the result. ■

A.4. Some Minor Extensions to Corollary 17

We are interested in the possibility of generalising, albeit perhaps only slightly, Lemma 16 and, as a consequence, Corollary 17. We will make reference to notation introduced in Appendix A.3. We continue to assume a deterministic transition function, and a fixed deterministic policy π. In our discussion around uniform priors, we were also able to assume that the sequence of states $s_1, \ldots, s_S$ used to generate the sets $\mathcal{C}_1, \ldots, \mathcal{C}_S$ was selected arbitrarily. In a more general setting we may need to assume that this sequence is generated according to some specific probability distribution. (In contrast, since, in the discussion below, we will generally assume that the prior distribution of the transition probabilities is identical for different actions, we will be able to continue to assume that π is arbitrary, provided it is chosen with no knowledge of P.)

Appealing to techniques which make use of the notion of Schur convexity (Marshall et al., 2011), it's possible to show that, if the random vector $P(\cdot|s_i, a_j)$ is independently distributed for all (i, j), and $\Pr(P(s_{i''}|s_i, a_j) = 1) = \Pr(P(s_{i''}|s_{i'}, a_{j'}) = 1)$ for all (i, j, i', j', i''), then, given an arbitrary policy π, and assuming the sequence of starting states $s_1, \ldots, s_S$ are distributed uniformly at random, $\mathrm{E}(L_1)$ and $\mathrm{Var}(L_1)$ are minimised where the prior for P is uniform. Using this fact, Corollary 17 can be extended to such priors (we omit the details, though the proof uses arguments substantially equivalent to those used in Corollary 17). If we continue to assume an arbitrary policy and that the starting states are selected uniformly at random, we can consider a yet more general class of priors, using a result from Karlin and Rinott (1984). Their result can be used to demonstrate that, of the set of priors which

satisfy the following three conditions—(1) that $\Pr(P(s_{i'}|s_i, a_j) = 1) = \Pr(P(s_{i'}|s_i, a_{j'}) = 1)$ for all (i, j, i', j'), (2) that:

$$\begin{aligned}&\Pr(P(s_{i_3}|s_{i_1}, a_j) = 1) > \Pr(P(s_{i_4}|s_{i_1}, a_j) = 1)\\ &\qquad\qquad \Rightarrow \Pr(P(s_{i_3}|s_{i_2}, a_j) = 1) > \Pr(P(s_{i_4}|s_{i_2}, a_j) = 1)\end{aligned}$$

for all (i_1, i_2, i_3, i_4, j), and (3) the random vector $P(\cdot|s_i, a_j)$ is independently distributed for all (i, j)—the uniform prior will again maximise the values $\mathrm{E}(L_1)$ and $\mathrm{Var}(L_1)$. Since $C_i \leq L_i$, an equivalent result to Corollary 17 (though not necessarily with the same constant $\sqrt{\pi/8}$) can similarly be obtained for this even larger set of priors (we again omit the details and note that the arguments are substantially equivalent to those in Corollary 17).

Both these results assume a degree of similarity in the transition prior probabilities for each state. A perhaps more interesting potential generalisation is as follows. We can define a *balanced* prior for a deterministic transition function P as any prior such that the random vector $P(\cdot|s_i, a_j)$ is independently distributed for all (i, j), and we have $\Pr(P(s_{i'}|s_i, a_j) = 1) = \Pr(P(s_{i'}|s_i, a_{j'}) = 1)$, $\Pr(P(s_{i'}|s_i, a_j) = 1) = \Pr(P(s_i|s_{i'}, a_j) = 1)$ and $\Pr(P(s_i|s_i, a_j) = 1) \geq \frac{1}{S}$ for all (i, j, i', j'). In essence, the prior probability of transitioning from state s_i to $s_{i'}$ is the same as transitioning in the reverse direction from $s_{i'}$ to s_i. This sort of prior would be reflective of many real world problems which incorporate some notion of a geometric space with distances, such as navigating around a grid (examples 1, 2 and 5 all have balanced priors). The difference to the uniform prior is that we now have a notion of the "closeness" between two states (reflected in how probable it is to transition in either direction between them). Similar to the uniform prior there is no inherent "flow" creating cycles which have larger expected value than C in the uniform case.

It is not hard to conceive of examples where $\mathrm{E}(C_1)$ may be significantly reduced for a particular balanced prior compared to the uniform case. It would furthermore appear plausible that, amongst the set of all balanced priors, $\mathrm{E}(L_1)$ would be maximised for the uniform prior. Indeed investigation using numerical optimisation techniques demonstrates this is the case for $S \leq 8$, even when a fixed arbitrary starting state is selected relative to the balanced set of transition probabilities. The techniques used for the generalisations stated above, however, cannot be used to prove a similar result for balanced priors.[52] Notwithstanding this, we conjecture, based on our numerical analysis, that the uniform prior does maximise $\mathrm{E}(L_1)$ for all S, which would carry the implication, since $C_1 \leq L_1$, that Lemma 18 can be used to argue $\mathrm{E}(C_1) = O(\sqrt{S})$ and $\mathrm{Var}(C_1) = O(S)$.

Note that, even if this conjecture holds, we cannot extend Corollary 17 to balanced priors, which we can see with a simple example. Set $\Pr(P(s_i|s_i, a_j) = 1) \geq 1-\varepsilon$ for all (i, j) where ε is small. Provided that the transition matrix associated with π and P is irreducible, then $C = S$ for all S.

Notwithstanding that a formal result equivalent to Corollary 17 is unavailable for balanced priors, we should still be able to exploit the apparent tendency of the agent to spend

52. The earlier stated results follow in both cases from the stronger statement that $\Pr(L_1 > k)$ is maximised for all k by a uniform prior, from which our conclusions regarding the moments follow. In the case of balanced priors such a strong result does not hold, which can be seen by—for large S, and taking $k = S$—comparing a uniform prior to a prior where all transitions outside of a single fixed cycle covering all S states have probability zero, and where the prior probability of a transition in either direction along this cycle is $(1 - 1/S)/2$.

a majority of the time in a small subset of the state space. The main difference is that, much like in Example 2, this small subset may change slowly over time.

A.5. Minor Alterations to PASA for Experiments

The changes made were as follows:

1. The underlying SARSA algorithm is adjusted so that $d^{(t)}$ in equation (4) is weighted by the reciprocal of $\pi(s_i, a_j)$, with the effect that rarely taken actions will have a greater impact on changes to θ. This can compensate for a slowing-down of learning which can result from setting ϵ at small values;

2. The role of ϑ in PASA is adjusted slightly. The rule for updating ρ in equation (6) becomes:

$$\rho_k = \begin{cases} j & \text{if } (1-\Sigma_{\rho_k})u_{\rho_k} < \max\{u_i : i \leq X_0 + k - 1, \Sigma_i = 0\} \times \vartheta \\ \rho_k & \text{otherwise} \end{cases}$$

(the final operator in the first line, originally addition, has been replaced with multiplication).[53] In practice ϑ would be set near to, but less than, one, as opposed to near zero as under the original formulation. This change means that the algorithm can make finer changes when comparing collections of cells with low probabilities, but still remains stable for cells with larger probabilities, which would have estimates more likely to be volatile;

3. A mechanism is introduced whereby, if ρ is changed at the end of an interval of length ν, the values of θ are changed to reflect this. Specifically, if we define, for some j:

$$\Psi = \left\{k : \mathcal{X}_k^{(n\nu-1)} \cap \mathcal{X}_j^{(n\nu)} \neq \emptyset\right\},$$

for some $n \in \mathbb{N}$ then we will set, for all $1 \leq l \leq A$:

$$\theta_{jl}^{(n\nu)} = \frac{1}{|\Psi|} \sum_{k \in \Psi} \theta_{kl}^{(n\nu-1)} + \eta I_{\left\{s^{(n\nu-1)} \in \mathcal{X}_j^{(n\nu-1)}\right\}} I_{\{a^{(n\nu-1)} = a_l\}} d^{(n\nu-1)}. \tag{11}$$

By definition this only needs to be done at the end intervals of length ν (in practice ν is large). The second term (i.e. outside the summation) in equation (11) comes from equation (3) and just reflects the standard SARSA update. At most, an additional temporary copy of Ξ (to permit a comparison between the set of cells defined at $t = n\nu - 1$ and $t = n\nu$) and a temporary vector of real values of size X (to store copies of single columns from θ) are required to perform the necessary calculations. The nature of each cell $\mathcal{X}_j^{(n\nu)}$ is such that it will either (a) be equal to $\bigcup_{k \in \mathcal{K}} \mathcal{X}_k^{(n\nu-1)}$ for some non-empty subset of indices $\mathcal{K}$ or (b) a strict subset of a single cell $\mathcal{X}_k^{(n\nu-1)}$. Performing this operation tends to speed up learning, as there is less tendency to transfer previously learned weights to unrelated areas of the state space (which might occur as a result of updates to Ξ by PASA as it was originally defined);

53. Note that defining PASA in this new, alternative way poses challenges when seeking to prove convergence (see, for example, Proposition 4).

4. A further mechanism is introduced such that, whenever $t \bmod \nu = 0$ (i.e. for any iteration in which ρ and Ξ are updated), copies of $\bar{\Xi}$ and $\bar{u}$, which we can call $\bar{\Xi}'$ and $\bar{u}'$ respectively, are made immediately before $\bar{\Xi}$ is updated (and are discarded at the end of the update process). The copies are used as part of the process whereby $\bar{\Xi}$ and Ξ are updated.

Note that there is a natural bijection between the set of all possible elements of $\bar{\Xi}$ (not including the base cells) and the set of all possible pairs of cells generated by a cell split during the splitting process (equate each element of $\bar{\Xi}$ which is not a base cell with the "uppermost" cell created as a consequence of a cell split). At each stage of the splitting process, for $1 \leq k \leq X - X_0$, each element of the set of pairs of cells corresponding to an available split may or may not implicitly correspond to an element of $\bar{\Xi}'$. When making a decision to split a cell, the algorithm, instead of comparing the value of $u_{i_{\max}}$ to the value of u_{ρ_k} (as described in Section 2.4), will compare the value of $u_{i_{\max}}$ to the maximum value of u for all cells in Ξ_{k-1} for which splits implicitly exist already in $\bar{\Xi}'$ (we will denote this maximum as $u_{i_{\text{old}}}$). The threshold ϑ is applied as normal. Specifically, if $u_{i_{\max}}$ exceeds the value of $u_{i_{\text{old}}}$ by at least this threshold then ρ_k is set to equal $i_{\max}$. Otherwise, ρ_k is set to equal i_{old}. (As always we take the lowest index where more than one index equals the maximum.)

Furthermore, if the "uppermost" of the pair of new cells generated by the chosen cell split is an element of $\bar{\Xi}'$—we may suppose that the new "uppermost" cell corresponds to the element of $\bar{\Xi}'$ with index l—then the value of $\bar{u}_{X_0+k}$ will be made equal to $\bar{u}'_l$, rather than calculated in the manner detailed in Section 2.4. Otherwise, $\bar{u}_{X_0+k}$ will be made equal to $\bar{u}_{\rho_k}/2$. This is to avoid this value being unnecessarily distorted by the standard process PASA uses to estimate the visit frequency of newly split cells.[54]

The effect of introducing this change is that it can help encourage stability. In particular it can often prevent cell visit frequencies being re-estimated simply due to cells being split in a different order. The change can help to speed up convergence at virtually no computational cost;

5. Finally, the stochastic approximation algorithm in equation (5) is replaced such that $\bar{u}$ is approximated in the following way. We introduce a new variable u_{counter}. Each iteration we update the value as follows: $u_{\text{counter}} = u_{\text{counter}} + I_{\{s^{(t)} \in \bar{\mathcal{X}}_j\}}$. In any iteration such that $t \bmod \nu = 0$ instead of equation (5) we apply the following formula:

$$\bar{u}_j^{(t+1)} = \bar{u}_j^{(t)} + \varsigma \left(u_{\text{counter}}/\nu - \bar{u}_j^{(t)}\right)$$

and reset u_{counter} to zero. In moving to the approximation, the parameter ς needs to be re-weighted to reflect the change. The reason we replace equation (5) is for practical reasons relating to implementation.[55]

54. Alternative implementations of PASA which use a tree-based approach to govern cell splits—similar to that described in, for example, Nouri and Littman (2009) or Munos and Moore (2002)—can circumvent the need for some of these cumbersome technical details. (Such alternative implementations are beyond our present scope.)

55. On conventional computers, summation can be performed much more quickly than multiplication. By restricting the multiplication step in equation (5) to iterations at the end of each interval of size ν, we

References

David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2915–2923, 2016.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

Andrey Bernstein and Nahum Shimkin. Adaptive-resolution reinforcement learning with polynomial exploration in deterministic domains. *Machine Learning*, 81(3):359–397, 2010.

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996. ISBN 1886529108.

Andrea Bonarini, Alessandro Lazaric, and Marcello Restelli. Learning in complex environments through multiple adaptive partitions. In *Proceedings of the ECAI*, volume 6, pages 7–12, 2006.

Craig Boutilier, Richard Dearden, Moises Goldszmidt, et al. Exploiting structure in policy construction. In *IJCAI*, volume 14, pages 1104–1113, 1995.

Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement Learning and Dynamic Programming Using Function Approximators*, volume 39. CRC press, 2010.

Dotan Di Castro and Shie Mannor. Adaptive bases for reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 312–327. Springer, 2010.

Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University press, 2009.

Geoffrey J Gordon. Chattering in SARSA(λ). Technical report, CMU Learning Lab, 1996.

Geoffrey J Gordon. Reinforcement learning with function approximation converges to a region. In *Advances in Neural Information Processing Systems*, pages 1040–1046, 2001.

Marcus Hutter. Extreme state aggregation beyond Markov decision processes. *Theoretical Computer Science*, 650:73–91, 2016.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4868–4878, 2018.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Samuel Karlin and Yosef Rinott. Random replacement schemes and multivariate majorization. *Lecture Notes-Monograph Series*, pages 35–40, 1984.

can speed up the algorithm. Since ς is typically very small, the change has negligible effect on the value of $\bar{u}$.

Philipp W Keller, Shie Mannor, and Doina Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 449–456, 2006.

Harold Kushner and G George Yin. *Stochastic Approximation and Recursive Algorithms and Applications*, volume 35. Springer Science & Business Media, 2003.

Sridhar Mahadevan, Stephen Giguere, and Nicholas Jacek. Basis adaptation for sparse nonlinear reinforcement learning. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 654–660, 2013.

Albert W Marshall, Ingram Olkin, and Barry C Arnold. *Inequalities: Theory of Majorization and its Applications*, volume 143. Springer, second edition, 2011. doi: 10.1007/978-0-387-68276-1.

Francisco S Melo, Sean P Meyn, and M Isabel Ribeiro. An analysis of reinforcement learning with function approximation. In *Proceedings of the 25th International Conference on Machine Learning*, pages 664–671, 2008.

Ishai Menache, Shie Mannor, and Nahum Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 134(1):215–238, 2005.

Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Andrew W Moore and Christopher G Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning*, 21(3):199–233, 1995.

Rémi Munos and Andrew Moore. Variable resolution discretization in optimal control. *Machine Learning*, 49(2-3):291–323, 2002.

Ali Nouri and Michael L Littman. Multi-resolution exploration in continuous spaces. In *Advances in Neural Information Processing Systems*, pages 1209–1216, 2009.

Ronald Ortner. Adaptive aggregation for reinforcement learning in average reward Markov decision processes. *Annals of Operations Research*, 208(1):321–336, 2013.

Theodore J Perkins and Doina Precup. A convergent form of approximate policy iteration. In *Advances in Neural Information Processing Systems*, pages 1627–1634, 2003.

Warren B Powell. *Approximate Dynamic Programming: Solving The Curses of Dimensionality*. John Wiley & Sons, second edition, 2011.

Warren B Powell, Hugo P Simao, and Belgacem Bouzaiene-Ayari. Approximate dynamic programming in transportation and logistics: a unified framework. *EURO Journal on Transportation and Logistics*, 1(3):237–284, 2012.

Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Stuart I Reynolds. Adaptive resolution model-free reinforcement learning: Decision boundary partitioning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 783–790, 2000.

Gavin A Rummery and Mahesan Niranjan. On-line Q-learning using connectionist systems. Technical report, University of Cambridge, 1994.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. Reinforcement learning with soft state aggregation. In *Advances in Neural Information Processing Systems*, pages 361–368, 1995.

Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

Shimon Whiteson, Matthew E Taylor, and Peter Stone. Adaptive tile coding for value function approximation. Technical report, Computer Science Department, University of Texas at Austin, 2007.

Huizhen Yu and Dimitri P Bertsekas. Basis function adaptation methods for cost approximation in MDP. In *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 74–81. IEEE, 2009.

Journal of Machine Learning Research 20 (2019) 1-30 Submitted 6/18; Revised 7/19; Published 8/19

Time-to-Event Prediction with Neural Networks and Cox Regression

Håvard Kvamme HAAVAKVA@MATH.UIO.NO
Ørnulf Borgan BORGAN@MATH.UIO.NO
Ida Scheel IDASCH@MATH.UIO.NO
Department of Mathematics
University of Oslo
P.O. Box 1053 Blindern
0316 Oslo, Norway

Editor: Jon McAuliffe

Abstract

New methods for time-to-event prediction are proposed by extending the Cox proportional hazards model with neural networks. Building on methodology from nested case-control studies, we propose a loss function that scales well to large data sets and enables fitting of both proportional and non-proportional extensions of the Cox model. Through simulation studies, the proposed loss function is verified to be a good approximation for the Cox partial log-likelihood. The proposed methodology is compared to existing methodologies on real-world data sets and is found to be highly competitive, typically yielding the best performance in terms of Brier score and binomial log-likelihood. A python package for the proposed methods is available at `https://github.com/havakv/pycox`.

Keywords: Cox regression, customer churn, neural networks, non-proportional hazards, survival prediction

1. Introduction

In this paper, we consider methodology for time-to-event prediction, a part of survival analysis that reasons about when a future event will occur. Applications of time-to-event predictions can be found in a variety of settings such as survival prediction of cancer patients (e.g., Vigan et al., 2000), customer churn (e.g., Van den Poel and Lariviere, 2004), credit scoring (e.g., Dirick et al., 2017), and failure times of mechanical systems (e.g., Susto et al., 2015). Arguably, the field of survival analysis has predominantly focused on interpretability, potentially at some cost of predictive accuracy. This is perhaps the reason why binary classifiers from machine learning are commonly used in industrial applications where survival methodology is applicable. However, while the binary classifiers can provide predictions for *one* predetermined duration, one loses the interpretability and flexibility provided by modeling the event probabilities as a function of time. Furthermore, in time-to-event data, it is common that some individuals are not followed all the way to their event time, resulting in *censored* times rather than event times. While binary classifiers typically ignore these observations, one of the main objectives in survival analysis is to account for them. Hence,

©2019 Håvard Kvamme, Ørnulf Borgan, and Ida Scheel.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/18-424.html`.

in applications with a substantial amount of censoring, the use of survival models tends to be advantageous.

In our work, we propose an approach for combining machine learning methodology with survival models. We do this by extending the Cox proportional hazards model with neural networks, and further remove the proportionality constraint of the Cox model. Building on methodology from nested case-control studies (e.g., Langholz and Goldstein, 1996) we are able to do this in a scalable manner. The resulting methods have the flexibility of neural networks while modeling event times continuously. Building on the PyTorch framework (Paszke et al., 2017), we provide a python package for our methodology, along with all the simulations and data sets presented in this paper.[1]

The paper is organized as follows. Section 2 contains a summary of related work. In Section 3, we review some basic concepts from survival analysis and introduce the Cox proportional hazards model with our extensions. In Section 4 we discuss some evaluation criteria of methods for time-to-event prediction. In Section 5, we conduct a simulation study, verifying that the methods we propose behave as expected. In Section 6 we evaluate our methods on five real-world data sets and compare their performances with existing methodology. We conclude in Section 7.

2. Related Work

The extension of Cox regression with neural networks was first proposed by Faraggi and Simon (1995), who replaced the linear predictor of the Cox regression model, cf. formula (3) below, by a one hidden layer multilayer perceptron (MLP). It was, however, found that the model generally failed to outperform regular Cox models (Xiang et al., 2000; Sargent, 2001). Katzman et al. (2018) revisited these models in the framework of deep learning and showed that novel networks were able to outperform classical Cox models in terms of the C-index (Harrell Jr et al., 1982). Our work distinguishes itself from this in the following way: The method by Katzman et al. (2018), denoted DeepSurv, is constrained by the proportionality assumption of the Cox model while we propose an extension of the Cox model where proportionality is no longer a restriction. In this regard, we propose an alternative loss function that scales well for both the proportional and the non-proportional cases.

Similar works based on Cox regression include SurvivalNet (Yousefi et al., 2017), a framework for fitting proportional Cox models with neural networks and Bayesian optimization of the hyperparameters, and Zhu et al. (2016) and Zhu et al. (2017) which extended the Cox methodology to images. Both Zhu et al. (2016) and Zhu et al. (2017) replace the MLP of DeepSurv with a convolutional neural network and applied these methods to pathological images of lung cancer and to whole slide histopathological images.

An alternative approach to time-to-event prediction is to discretize the duration and compute the hazard or survival function on this predetermined time grid. Luck et al. (2017) proposed methods similar to DeepSurv, but with an additional set of discrete outputs for survival predictions and computed an isotonic regression loss over this time grid. Fotso (2018) parameterized a multi-task logistic regression with a neural net that directly computes the survival probabilities on the time grid. Lee et al. (2018) proposed a method,

1. Implementations of methods and the data sets are available at `https://github.com/havakv/pycox`.

denoted DeepHit, that estimates the probability mass function with a neural net and combine the log-likelihood with a ranking loss; see Appendix D for details. Furthermore, the method has the added benefit of being applicable for competing risks.

The majority of the papers mentioned benchmark their methods against the random survival forests (RSF) by Ishwaran et al. (2008). RSF computes a random forest using the log-rank test as the splitting criterion. It computes the cumulative hazards of the leaf nodes and averages them over the ensemble. Hence, RSF is a very flexible continuous-time method that is not constrained by the proportionality assumption.

3. Methodology

In the following, we give a brief review of some concepts in survival analysis. For a more in-depth introduction to the field, see, for example, Klein and Moeschberger (2003).

Our objective is to model the event distribution as a continuous function of time. So with $f(t)$ and $F(t)$ denoting the probability density function and the cumulative distribution function of an event time T^*, we want to model

$$\mathrm{P}(T^* \leq t) = \int_0^t f(s)\, ds = F(t).$$

As alternatives to $F(t)$, it is common to study the *survival function* $S(t)$ and the *hazard rate* $h(t)$. The survival function is defined as

$$S(t) = \mathrm{P}(T^* > t) = 1 - F(t),$$

and is commonly used for visualizing event probabilities over time. For specifying models, however, it is rather common to use the hazard rate

$$h(t) = \frac{f(t)}{S(t)} = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathrm{P}(t \leq T^* < t + \Delta t \mid T^* \geq t).$$

If we have the hazard rate, the survival function can be retrieved through the cumulative hazard, $H(t) = \int_0^t h(s) ds$, by

$$S(t) = \exp[-H(t)]. \tag{1}$$

The survival function and the hazard rate therefore provide contrasting views of the same quantities, and it may be useful to study both.

Working with real data, the true event times are typically not known for all individuals. This can occur when the follow-up time for an individual is not long enough for the event to happen, or an individual may leave the study before its termination. Instead of observing the true event time T^*, we then observe a possibly right-censored event time $T = \min\{T^*, C^*\}$, where C^* is the censoring time. In addition, we observe the indicator $D = \mathbb{1}\{T = T^*\}$ labeling the observed event time T as an event or a censored observation. Now, denoting individuals by i, with covariates $\mathbf{x}_i$ and observed duration T_i, the likelihood for censored survival times is given by

$$L = \prod_i f(T_i \mid \mathbf{x}_i)^{D_i} S(T_i \mid \mathbf{x}_i)^{1-D_i} = \prod_i h(T_i \mid \mathbf{x}_i)^{D_i} \exp[-H(T_i \mid \mathbf{x}_i)]. \tag{2}$$

We will later refer to this as the *full likelihood*.

3.1. Cox Regression

The Cox proportional hazards model (Cox, 1972) is one of the most used models in survival analysis. It provides a semi-parametric specification of the hazard rate

$$h(t \mid \mathbf{x}) = h_0(t) \exp[g(\mathbf{x})], \qquad g(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x}, \tag{3}$$

where $h_0(t)$ is a non-parametric *baseline hazard*, and $\exp[g(\mathbf{x})]$ is the *relative risk function*. Here, $\mathbf{x}$ is a covariate vector and $\boldsymbol{\beta}$ is a parameter vector. Note that the linear predictor $g(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x}$ does not contain an intercept term (bias weight). This is because the intercept would simply scale the baseline hazard, and therefore not contribute to the relative risk function,

$$h_0(t) \exp[g(\mathbf{x}) + b] = h_0(t) \exp[b] \exp[g(\mathbf{x})] = \tilde{h}_0(t) \exp[g(\mathbf{x})].$$

The Cox model in (3) is fitted in two steps. First, the parametric part is fitted by maximizing the *Cox partial likelihood*, which does not depend on the baseline hazard, then the non-parametric baseline hazard is estimated based on the parametric results. For individual i, let T_i denote the possibly censored event time and $\mathcal{R}_i$ denote the set of all individuals at risk at time T_i (not censored and have not experienced the event before time T_i). Note that $\mathcal{R}_i$ includes individuals with event times at T_i, so i is part of $\mathcal{R}_i$. The Cox partial likelihood, with Breslow's method for handling tied event times, is given by

$$L_{\text{cox}} = \prod_i \left(\frac{\exp[g(\mathbf{x}_i)]}{\sum_{j \in \mathcal{R}_i} \exp[g(\mathbf{x}_j)]} \right)^{D_i}, \tag{4}$$

and the negative partial log-likelihood can then be used as a loss function

$$\text{loss} = \sum_i D_i \log \left(\sum_{j \in \mathcal{R}_i} \exp[g(\mathbf{x}_j) - g(\mathbf{x}_i)] \right). \tag{5}$$

Let $\hat{\boldsymbol{\beta}}$ be the value of $\boldsymbol{\beta}$ that maximizes (4), or equivalently, minimizes (5). Then the cumulative baseline hazard function can be estimated by the *Breslow estimator*

$$\hat{H}_0(t) = \sum_{T_i \leq t} \Delta \hat{H}_0(T_i) \tag{6}$$

$$\Delta \hat{H}_0(T_i) = \frac{D_i}{\sum_{j \in \mathcal{R}_i} \exp[\hat{g}(\mathbf{x}_j)]},$$

where $\hat{g}(\mathbf{x}) = \hat{\boldsymbol{\beta}}^T \mathbf{x}$. If desired, the baseline hazard $h_0(t)$ can be estimated by smoothing the increments, $\Delta \hat{H}_0(T_i)$, of the Breslow estimate, but the cumulative baseline hazard typically provides the information we are interested in.

3.2. Cox with SGD

The Cox partial likelihood is usually minimized using Newton-Raphson's method. In our work, we instead want to fit the Cox model with mini-batch stochastic gradient descent

(SGD), to better scale to large data sets. As the loss in (5) sums over risk sets $\mathcal{R}_i$, which can be as large as the full data set, it cannot be computed in batches. Nevertheless, it is possible to do batched iterations by subsampling the data set (to a batch) and restrict the set $\mathcal{R}_i$ to only contain individuals in the current batch. This scales well for proportional methods such as DeepSurv (Katzman et al., 2018), but would be very computationally expensive for our non-proportional extension presented in Section 3.4. Hence, we propose an approximation of the loss that is easily batched.

Intuitively, we can approximate the risk set $\mathcal{R}_i$ with a sufficiently large subset $\tilde{\mathcal{R}}_i$, and weight the likelihood accordingly with weights w_i,

$$L = \prod_i \left(\frac{\exp[g(\mathbf{x}_i)]}{w_i \sum_{j \in \tilde{\mathcal{R}}_i} \exp[g(\mathbf{x}_j)]} \right)^{D_i}. \tag{7}$$

The weights should ensure that the weighted sum over the subset $\tilde{\mathcal{R}}_i$ in (7) is a reasonable approximation of the full sum over $\mathcal{R}_i$ in (4). By choosing a fixed sample size of the sampled risk sets $\tilde{\mathcal{R}}_i$, we can now optimize the objective by batched gradient descent. The individual i is always included in the sampled risk set $\tilde{\mathcal{R}}_i$ to ensure that each of the products in (7) is bounded above by 1. As the weights w_i do not contribute to the gradients of the logarithm of (7) (as can be seen by differentiating with respect to the model parameters), we can simply drop them from the loss function. Also, in practice we do not compute the loss for $D_i = 0$ as these entries do not contribute to (7). Finally, if we average the loss to make it independent of the data set size, we obtain

$$\text{loss} = \frac{1}{n} \sum_{i:D_i=1} \log \left(\sum_{j \in \tilde{\mathcal{R}}_i} \exp[g(\mathbf{x}_j) - g(\mathbf{x}_i)] \right), \tag{8}$$

where n denotes the number of events in the data set. In our experiments in Sections 5 and 6, we find that it is often sufficient to sample only *one* individual j from the risk set, which gives us the loss

$$\text{loss} = \frac{1}{n} \sum_{i:D_i=1} \log \left(1 + \exp[g(\mathbf{x}_j) - g(\mathbf{x}_i)]\right), \quad j \in \mathcal{R}_i \backslash \{i\}. \tag{9}$$

One benefit of (8) is that it is, in a sense, more interpretable than the negative partial log-likelihood in (5). Due to the sample dependence in the mean partial log-likelihood (MPLL), i.e., the expression in (5) divided by n, the magnitude of the MPLL is dependent on the size of the risk sets. Hence, for a change of batch size, the mean partial log-likelihood changes. This prohibits a comparison of losses across different batch sizes. Comparably, the loss in (8) is not affected by the choice of batch size, as the size of $\tilde{\mathcal{R}}_i$ is fixed. As a result, we can derive the range of values we expect the loss to be in. Using (9) as an example, we know that it is typically in the range (0, 0.693], as a trivial $g(\mathbf{x}) = \text{const}$, gives a loss $= \log(2) \approx 0.693$, and the minimum is obtained by letting $g(\mathbf{x}_i) \to \infty$, $g(\mathbf{x}_j) \to -\infty$, which results in a loss that tends towards 0.

Sampling of the risk sets in Cox's partial likelihood is commonly done in epidemiology and formalized through the nested case-control design, originally suggested by Thomas

(1977). In (8), *case* refers to the i's, while the *controls* are the j's sampled from $\mathcal{R}_i \backslash \{i\}$. Goldstein and Langholz (1992) show that for the Cox partial likelihood, the sampled risk sets produce consistent parameter estimators. While their results do not extend to non-linear models, it is still an indication that the loss function in (8) is reasonable.

Our sampling strategy deviates from that of the nested case-control literature in two ways. Firstly, we sample a new set of controls for every iteration, instead of keeping control samples fixed. Secondly, we sample controls with replacement, as this requires less computation than sampling without replacement. Note, however, that we typically sample a single control, in which case it does not matter if we sample with or without replacement.

3.3. Non-Linear Cox

Having established the simple loss function in (8), which can be computed with SGD, the generalization of the relative risk function $\exp[g(\mathbf{x})]$ is rather straightforward. In this paper, we replace the linear predictor $g(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x}$ by a $g(\mathbf{x})$ parameterized by a neural network. While our proposed loss function is not a requirement for the adaptation of a neural network (see, e.g., DeepSurv by Katzman et al., 2018), it really helps for the further extensions in Section 3.4. Also, it has been found that batched iterations can improve predictive performance (Keskar et al., 2016; Hoffer et al., 2017).

Our generalization of $g(\mathbf{x})$ leaves the presented theory in Sections 3.1 and 3.2 essentially unchanged, so we do not repeat the likelihoods and loss functions for this model.

We will later refer to the Cox proportional hazards model parameterized with a neural network as Cox-MLP. To differentiate between minimizing the negative partial log-likelihood in (5), as done by DeepSurv, and our case-control approximation in (8), we will denote the corresponding methods by Cox-MLP (DeepSurv) and Cox-MLP (CC), respectively.

For the non-linear Cox models, the loss does not necessarily have a unique minimizer for $g(\mathbf{x})$. Therefore, we add a penalty to the loss function to encourage $g(\mathbf{x})$ to not deviate too far from zero

$$\text{penalty} = \lambda \sum_{i:D_i=1} \sum_{j \in \tilde{\mathcal{R}}_i} |g(\mathbf{x}_j)|. \tag{10}$$

Here λ is a tuning parameter, and note that i is included in $\tilde{\mathcal{R}}_i$.

3.4. Non-Proportional Cox-Time

The proportionality assumption of the Cox model can be rather restrictive, and parameterizing the relative risk function with a neural net does not affect this constraint. Approaches for circumventing this restriction are typically based on grouping the data based on a categorical covariate and applying a stratified version of the Cox model (Klein and Moeschberger, 2003, chap. 9). We propose a parametric approach that does not require stratification. Continuing with the semi-parametric form of the Cox model, we now let the relative risk function depend on time,

$$h(t \,|\, \mathbf{x}) = h_0(t) \exp[g(t, \mathbf{x})]. \tag{11}$$

In practice, we let $g(t, \mathbf{x})$ handle the time as a regular covariate, which enables $g(t, \mathbf{x})$ to model interactions between time and the other covariates. This is similar to the approach taken in classical survival analysis, where the non-proportional effect of a covariate x may be modeled by including time-dependent covariates like $x \cdot t$ and $x \cdot \log t$.

The model (11) is no longer a proportional hazards model. However, it is still a relative risk model with the same partial likelihood as previously, only now with an additional covariate. Following the approach from Section 3.2, we have the loss function

$$\text{loss} = \frac{1}{n} \sum_{i:D_i=1} \log \left(\sum_{j \in \tilde{\mathcal{R}}_i} \exp[g(T_i, \mathbf{x}_j) - g(T_i, \mathbf{x}_i)] \right), \tag{12}$$

and we include the penalty in (10), with $g(T_i, \mathbf{x}_j)$ replacing $g(\mathbf{x}_j)$. We will later refer to models fitted by (12) as *Cox-Time*.

Note that the loss has the same T_i for both $\mathbf{x}_i$ and the $\mathbf{x}_j$'s. Consequently, if we had used the full risk set $\mathcal{R}_i$ instead of the subset $\tilde{\mathcal{R}}_i$, as is the case for the loss in (5), the loss would become very computationally expensive. In fact, for the full risk set, the time complexity of the loss would be $O(n \cdot |\mathcal{R}_i|) = O(n^2)$, where $|\mathcal{R}_i|$ denotes the size of the risk set. But for (12) we get $O(n \cdot |\tilde{\mathcal{R}}_i|) = O(n)$, as $|\tilde{\mathcal{R}}_i|$ is fixed and small. In the proportional case, to compute the loss in (5) one only needs to compute $g(\mathbf{x}_j)$ once (per iteration) for each j, and reuse that value in all other risk sets. This ensures the linear time complexity for the classical Cox models.

We can find the Breslow estimate for the cumulative baseline hazard $H_0(t)$ using (6) with $\hat{g}(\mathbf{x}_j)$ replaced by $\hat{g}(T_i, \mathbf{x}_j)$. Note that we still need the non-parametric baseline, as $g(t, \mathbf{x})$ is restricted to model interactions between the time and the covariates. To see this, consider $g(t, \mathbf{x}) = a(t, \mathbf{x}) + b(t)$, and observe that $b(t)$ cancels out in the loss.

3.5. Prediction

We can obtain predictions from the relative risk models by estimating the survival function in (1), $\hat{S}(t \,|\, \mathbf{x}) = \exp[-\hat{H}(t \,|\, \mathbf{x})]$. For the proportional hazards models, the relative risk function does not depend on time, enabling us to integrate only over the baseline hazard and compute the relative risk separately,

$$H(t \,|\, \mathbf{x}) = \int_0^t h_0(s) \exp[g(\mathbf{x})] \, ds = H_0(t) \exp[g(\mathbf{x})].$$

By first estimating $H_0(t)$ on the training data with (6), we only need to compute $\exp[g(\mathbf{x})]$ to obtain predictions. Computation of the estimate $\hat{H}_0(t)$ requires a single pass over the whole training set, in addition to sorting the training set by time.

In the case of models with non-proportional hazards, such as models fitted by Cox-Time in Section 3.4, predictions are much more computationally expensive. As the relative risk is time-dependent, we now need to integrate over both the baseline hazard and $g(t, \mathbf{x})$,

$$H(t \,|\, \mathbf{x}) = \int_0^t h_0(s) \exp[g(s, \mathbf{x})] \, ds.$$

In practice, we estimate the cumulative hazards by

$$\hat{H}(t \,|\, \mathbf{x}) = \sum_{T_i \leq t} \Delta \hat{H}_0(T_i) \exp[\hat{g}(T_i, \mathbf{x})],$$

where $\Delta \hat{H}_0(T_i)$ is an increment of the Breslow estimate and $\hat{g}(T_i, \mathbf{x})$ is the estimate of $g(T_i, \mathbf{x})$ obtained from the neural network. This is clearly rather computationally expensive as we need to compute $\hat{g}(T_i, \mathbf{x})$ for all distinct event times $T_i \leq t$. Furthermore, for continuous-time data, computation of the cumulative baseline hazard through the Breslow estimate,

$$\Delta \hat{H}_0(T_i) = \frac{D_i}{\sum_{j \in \mathcal{R}_i} \exp[\hat{g}(T_i, \mathbf{x}_j)]}, \tag{13}$$

scales quadratically.

To alleviate the computational cost, one can compute the cumulative hazards over a reduced number of distinct time points. Hence, Cox-Time is trained on continuous-time data but produces discrete-time predictions, with the benefit of the discretization happening after the network is fitted. In practice, we perform this discretization by computing the baseline on a random subset of the training data and subsequently control the resolution of the time grid through the sample size.

4. Evaluation Criteria

Metrics for evaluating the performance of methods for time-to-event prediction should account for the censored individuals. In the following, we describe the metrics used in the experimental sections of this paper.

4.1. Concordance Index

In survival analysis, the concordance index, or C-index (Harrell Jr et al., 1982), is arguably one of the most commonly applied discriminative evaluation metrics. This is likely a result of its interpretability, as it has a close relationship to classification accuracy (Ishwaran et al., 2008) and ROC AUC (Heagerty and Zheng, 2005). In short, the C-index estimates the probability that, for a random pair of individuals, the predicted survival times of the two individuals have the same ordering as their true survival times. See Ishwaran et al. (2008) for a detailed description.

As the C-index only depends on the ordering of the predictions, it is very useful for evaluating proportional hazards models. This is because the ordering of proportional hazards models does not change over time, which enables us to use the relative risk function instead of a metric for predicted survival time. It is, however, not obvious how the C-index should be applied for non-proportional hazards models (Gerds et al., 2012; Ishwaran et al., 2008). We will use a metric based on the time-dependent C-index by Antolini et al. (2005), which estimates the probability that observations i and j are concordant given that they are comparable,

$$C^{\text{td}} = \mathrm{P}\{\hat{S}(T_i \,|\, \mathbf{x}_i) < \hat{S}(T_i \,|\, \mathbf{x}_j) \mid T_i < T_j, \, D_i = 1\}. \tag{14}$$

However, to account for tied event times and survival estimates, we make the modifications listed by Ishwaran et al. (2008, Section 5.1, step 3). This is to ensure that predictions independent of $\mathbf{x}$, $\hat{S}(t\,|\,\mathbf{x}) = \hat{S}(t)$, yields $C^{\text{td}} = 0.5$ for unique event times. Note that for proportional hazards models, our metric is equivalent to the regular C-index.

4.2. Brier Score

The Brier score (BS) for binary classification is a metric of both discrimination and calibration of a model's estimates. In short, for N binary labels $y_i \in \{0, 1\}$ with probabilities p_i of $y_i = 1$, the BS is the mean squared error of the probability estimates $\hat{p}_i$, i.e., $\text{BS} = \frac{1}{N}\sum_i (y_i - \hat{p}_i)^2$. To get binary outcomes from time-to-event data, we choose a fixed time t and label data according to whether or not an individual's event time is shorter or longer than t. Graf et al. (1999) generalize the Brier score to account for censoring by weighting the scores by the inverse censoring distribution,

$$\text{BS}(t) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\hat{S}(t\,|\,\mathbf{x}_i)^2\,\mathbb{1}\{T_i \le t, D_i = 1\}}{\hat{G}(T_i)} + \frac{(1-\hat{S}(t\,|\,\mathbf{x}_i))^2\,\mathbb{1}\{T_i > t\}}{\hat{G}(t)}\right]. \tag{15}$$

Here N is the number of observations, $\hat{G}(t)$ is the Kaplan-Meier estimate of the censoring survival function, $\text{P}(C^* > t)$, and it is assumed that the censoring times and survival times are independent.

The BS can be extended from a single duration t to an interval by computing the integrated Brier score

$$\text{IBS} = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} \text{BS}(s)\,ds. \tag{16}$$

In practice, we approximate this integral by numerical integration, and we let the time span be the duration of the test set. In our experiments we found that using 100 grid points were sufficient to obtain stable scores.

4.3. Binomial Log-Likelihood

The mean binomial log-likelihood is a commonly used metric in binary classification that measures both discrimination and calibration of the estimates of a method. Using the same inverse censoring weighting as for the Brier score, we can apply this metric to censored duration time data,

$$\text{BLL}(t) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\log[1-\hat{S}(t\,|\,\mathbf{x}_i)]\,\mathbb{1}\{T_i \le t, D_i = 1\}}{\hat{G}(T_i)} + \frac{\log[\hat{S}(t\,|\,\mathbf{x}_i)]\,\mathbb{1}\{T_i > t\}}{\hat{G}(t)}\right]. \tag{17}$$

The binomial log-likelihood can also be integrated in the same manner as (16)

$$\text{IBLL} = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} \text{BLL}(s)\,ds.$$

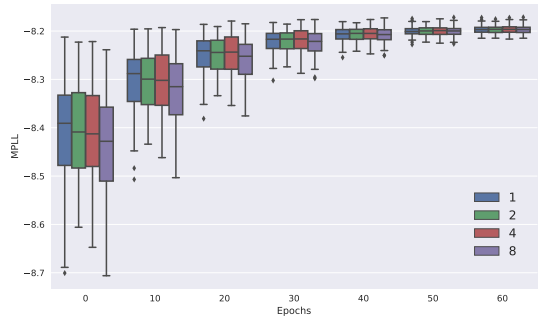


Figure 1: Box plots giving the mean partial log-likelihood (MPLL) of the test sets for different training epochs. The colors show how many controls were sampled during training (in addition to the case).

5. Simulations

To empirically investigate our proposed methodology, we conducted a series of simulations. These experiments are by no means exhaustive but are rather meant to verify that the methods behave as expected. In the following, we let *classical Cox* refer to a Cox regression with $g(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{x}$ obtained with the Lifelines python package (Davidson-Pilon et al., 2018). For experimental details exempt from the main article, we refer the reader to Appendix C.3.

We first investigate the behavior of our proposed loss (8). In particular, we want to examine the impact the number of sampled controls has on the fitted models, in addition to how well the results from using our loss agree with those from using the Cox partial log-likelihood, i.e., the loss (5). To this end, we simulated survival times from a proportional hazards model

$$\begin{aligned} h(t\,|\,\mathbf{x}) &= h_0(t)\exp[g(\mathbf{x})], \\ g(\mathbf{x}) &= \boldsymbol{\beta}^T\mathbf{x}, \end{aligned} \tag{18}$$

with constant baseline hazard $h_0(t) = 0.1$, and $\boldsymbol{\beta}^T = [0.44, 0.66, 0.88]$. The covariates were sampled uniformly from $[-1, 1]$. We drew censoring times independent of the covariates with constant hazard $c(t) = \frac{1}{30}$, and, in addition, we censored all individuals that were still under observation at time 30. This resulted in approximately 30 % censored individuals.

We sampled 10,000 individuals for training and 10,000 for testing, and fitted our Cox model by SGD as described in Section 3.2. This method will be referred to as *Cox-SGD*. Four models were fitted by sampling 1, 2, 4, and 8 controls (in addition to the case). The whole experiment was repeated 100 times, and the mean partial log-likelihood (MPLL) of the test sets are visualized in Figure 1. The figure indicates that the number of sampled controls does not affect the rate of convergence, but we note that the computational complexity increases with the number of sampled controls.

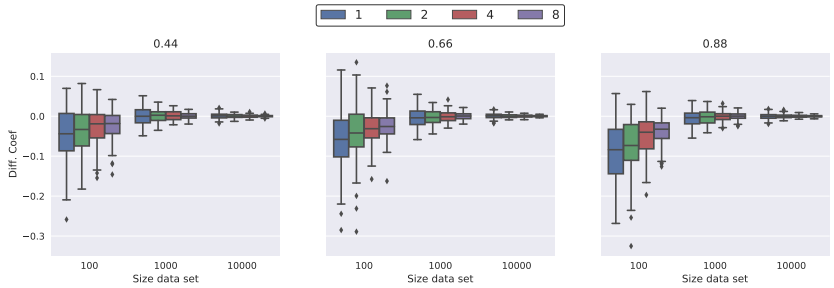


Figure 2: Differences between Cox-SGD and Classical Cox parameter estimates for different data set sizes. The numbers above the plots give the value of the true coefficient. Each box plot is based on 100 observations. The legend above the plots states the number of sampled controls for Cox-SGD.

The experiment was repeated with training sets of 1,000 individuals to verify that the results were not simply due to the size of the training data. The same patterns were found in this setting, so the figure is exempted from the paper.

Next, we compare the parameter estimates obtained from our proposed loss (Cox-SGD) with the estimates obtained with classical Cox regression. For data sets of sizes 100, 1,000, and 10,000, we fitted models to 100 sampled data sets. The differences between the Cox-SGD parameter estimates and the classical Cox estimates are displayed in Figure 2, where the legend above the plots gives the number of controls sampled for the Cox-SGD method. For the data sets of size 100, we observe that the Cox-SGD estimates seem to be slightly smaller than the Cox estimates, and this difference is larger for fewer sampled controls. However, as the data sets increase in size, the estimates for the two methods agree well.

Finally, we want to compare the likelihoods obtained by the two methods. As the mean partial log-likelihood depends on data set size, it is really not comparable across data sets. We will therefore instead use the full likelihood (2), which also depends on the baseline hazard, to compare the methods. Note that the partial likelihood may be interpreted as a profile likelihood, so the full likelihood and the partial likelihood are closely related (see, e.g., Klein and Moeschberger, 2003, p. 258). We obtain cumulative baseline hazard estimates by (6), and baseline hazard estimates by numerical differentiation of the cumulative baseline hazard estimates. We report the mean log-likelihood (MLL) of (2) to compare results for different data set sizes. In Figure 3, we show the difference in the MLL between the Cox-SGD method and the classical method for Cox regression (for each sampled data set). We used the training MLL as we are here only interested in the losses' abilities to optimize the objective, and not the generalization to a test set. From the figure, we observe that, for smaller data sets, more sampled controls in Cox-SGD seems to give likelihood estimates closer to those of a classical Cox regression. As the data sets increase in size, the MLL of the Cox-SGD seems to converge to that of the classical Cox, regardless of control sample

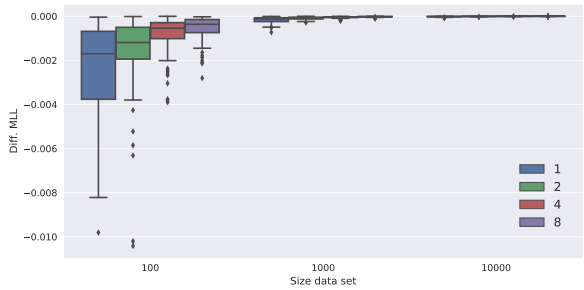


Figure 3: Differences in mean log-likelihood between Cox-SGD and classical Cox using (2). The figure gives results for different data set sizes, and the legend gives the number of sampled controls. Each box plot is based on 100 observations.

size. The MLL's of the classical Cox regression is approximately -2.2, meaning that even for the smallest data sets the differences in MLL for 1 sampled control is around 0.1 %. Hence, it seems that our loss (8) approximates the negative partial log-likelihood rather well. Furthermore, while a higher number of sampled controls can give lower training error, the effect of the number of sampled controls decreases with the size of the data sets.

In the further experiments, we will for simplicity only use one sampled control, as this was found sufficient. Furthermore, in Appendix C we have included simulations for the methods using neural networks for non-linear and non-proportional models, to verify that our proposed methods behave as expected. In summary, these simulations verify that for observations drawn from a proportional hazards model with a non-linear relative risk function, Cox models parameterized by neural networks provide better estimates of the partial log-likelihood than classical Cox models. Further, by drawing observations from a non-proportional relative risk model, the simulations verify that our Cox-Time method (Section 3.4) is able to obtain better estimates of the survival functions than methods which assume proportional hazards.

6. Experiments

In the following, we will evaluate our proposed methods on real data sets and compare their performance to existing methods from the literature. In total, we use five data sets. One large data set for a more in-depth analysis (see Section 6.2), and four smaller data sets commonly used in the survival analysis literature.

6.1. Four Common Survival Data Sets

We base this experimental section on the data sets provided by Katzman et al. (2018), as they are made available through the DeepSurv python package, and need no further prepro-

Data set	Size	Covariates	Unique Durations	Prop. Censored
SUPPORT	8,873	14	1,714	0.32
METABRIC	1,904	9	1,686	0.42
Rot. & GBSG	2,232	7	1,230	0.43
FLCHAIN	6,524	8	2,715	0.70

Table 1: Summary of the four data sets used in the experiments in Section 6.1.

cessing. The data sets include the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT), the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC), and the Rotterdam tumor bank and German Breast Cancer Study Group (Rot. & GBSG). Katzman et al. (2018) also provide the Worcester Heart Attack Study (WHAS) data set. However, their version of the data set is actually a case-control data set, meaning it contains multiple replications of some individuals, something the authors seem to have overlooked. We replace it with the Assay Of Serum Free Light Chain (FLCHAIN) made available in the survival packages of R (Therneau, 2015). For FLCHAIN, we remove individuals with missing values. Further, we remove the "chapter" covariate, which gives the cause of death. Table 1 provides a summary of the data sets. For a more detailed description, we refer to the original sources (Therneau, 2015; Katzman et al., 2018).

6.1.1. Methods and Hyperparameter Tuning

In Sections 3.3 and 3.4 we presented two new survival methods based on case-control sampling and neural networks: a proportional Cox method and a non-proportional Cox method, which we will refer to as *Cox-MLP (CC)* and *Cox-Time* respectively. We will compare our methods to a classical linear Cox regression referred to as *Classical Cox (Linear)*, DeepHit (Lee et al., 2018), and Random Survival Forests (RSF) (Ishwaran et al., 2008). We will also compare to a proportional Cox method similar to DeepSurv (Katzman et al., 2018), but our version performs batched SGD by computing the negative partial log-likelihood in (5) on a subset of the data set. Furthermore, we choose not to restrict the network structure and optimization scheme to that of Katzman et al. (2018). Hence, this method is identical to our proportional Cox method in Section 3.3, except that it computes the negative partial log-likelihood of a batch, while we use case-control sampling in the loss function. We will refer to these two methods as Cox-MLP (DeepSurv) and Cox-MLP (CC) respectively. We do not compare with Luck et al. (2017) as their method is another proportional hazard method and is therefore restricted in all the same ways as our other proportional methods. As we will show in Section 6.1.2, the proportional hazards assumption is very restrictive, and methods based on this assumption are therefore not able to compete with other methods such as DeepHit, RSF, and Cox-Time.

For the neural networks, we standardize the numerical covariates and encode the categorical covariates by entity embeddings (Guo and Berkhahn, 2016) half the size of the number of categories. For the Classical Cox (Linear) regression, we one-hot encoded the categorical covariates (dummy variables), and in RSF we simply passed the covariates without any transformations. The networks are standard multi-layer perceptrons with the same

Method	SUPPORT	METABRIC	Rot. & GBSG	FLCHAIN
Classical Cox (Linear)	0.598	0.628	0.666	0.790
Cox-MLP (DeepSurv)	0.611	0.636	0.674	0.790
Cox-MLP (CC)	0.613	0.643	0.669	**0.793**
Cox-Time	0.629	0.662	**0.677**	0.790
DeepHit	**0.642**	**0.675**	0.670	0.792
RSF (Mortality)	0.628	0.649	0.667	0.784
RSF (C^{td})	0.634	0.652	0.669	0.786

Table 2: Concordance (C^{td}) for the experiments in Section 6.1.

number of nodes in every layer, ReLU activations, and batch normalization between layers. We used dropout, normalized decoupled weight decay (Loshchilov and Hutter, 2019), and early stopping for regularization. SGD was performed by AdamWR (Loshchilov and Hutter, 2019) with an initial cycle length of one epoch, and we double the cycle length after each cycle. Learning rates were found using the methods proposed by Smith (2017).

As the four data sets are somewhat small, we scored our fitted models using 5-fold cross-validation, where the hyperparameter search was performed individually for each fold. For all the neural networks, we performed a random hyperparameter search over 300 parameter configurations and chose the model with the best score on a held-out validation set. We scored the proportional Cox methods by the partial log-likelihood, and for the Cox-Time method, we used the loss (12). Model hyperparameter tuning for DeepHit and RSF was performed with the time-dependent concordance index (Antolini et al., 2005). However, we also include an RSF tuned in the manner proposed by the authors, i.e., by computing the concordance of the mortality (Ishwaran et al., 2008, Section 4.1). The two versions of RSF are in the following denoted RSF (C^{td}) and RSF (Mortality), respectively. A list of the hyperparameter search spaces can be found in Appendix A.1.

6.1.2. Results

We compare the methods using the metrics presented in Section 4, i.e., the time-dependent concordance, the integrated Brier score, and the integrated binomial log-likelihood. While the concordance solely evaluates a method's discriminative performance, the Brier score and binomial log-likelihood also evaluate the calibration of the survival estimates.

Table 2 shows the time-dependent concordance, or C^{td}, averaged over the five cross-validation folds. As expected, the methods that assume proportional hazards, in general, perform worse than the less restrictive methods, and Cox-MLP (DeepSurv) and Cox-MLP (CC) are very close to each other. We see that RSF (C^{td}) has better concordance than RSF (Mortality), which is expected as RSF (C^{td}) uses the same metric for hyperparameter tuning as for evaluation. Cox-Time seems to do slightly better than the RSF methods, which is impressive as we have not used concordance for hyperparameter tuning. DeepHit seems to have the best discriminative performance overall, but, as we will see next, this comes at the cost of poorly calibrated survival estimates.

Method	SUPPORT	METABRIC	Rot. & GBSG	FLCHAIN
Classical Cox (Linear)	0.217	0.183	0.180	0.096
Cox-MLP (DeepSurv)	0.214	0.176	0.170	**0.092**
Cox-MLP (CC)	0.213	0.174	0.171	0.093
Cox-Time	**0.212**	**0.172**	**0.169**	0.102
DeepHit	0.223	0.184	0.178	0.120
RSF (Mortality)	0.215	0.175	0.171	0.093
RSF (C^{td})	**0.212**	0.176	0.171	0.093

Table 3: Integrated Brier score weighted by estimates of the censoring distribution for the experiments in Section 6.1.

Method	SUPPORT	METABRIC	Rot. & GBSG	FLCHAIN
Classical Cox (Linear)	-0.623	-0.538	-0.529	-0.322
Cox-MLP (DeepSurv)	-0.619	-0.532	-0.514	**-0.309**
Cox-MLP (CC)	-0.615	-0.515	-0.509	-0.314
Cox-Time	-0.613	**-0.511**	**-0.502**	-0.432
DeepHit	-0.637	-0.539	-0.524	-0.487
RSF (Mortality)	-0.619	-0.515	-0.507	-0.311
RSF (C^{td})	**-0.610**	-0.517	-0.507	-0.311

Table 4: Integrated binomial log-likelihood weighted by estimates of the censoring distribution for the experiments in Section 6.1.

Tables 3 and 4 show the integrated Brier score and integrated binomial log-likelihood, both weighted with Kaplan-Meier estimates of the censoring distribution. Here, for both metrics, closer to zero is better. We find that both metrics yield very similar results, as the orderings of the methods are almost identical. First, we see that Cox-Time seems to generally perform the best, but it struggles with the FLCHAIN data set. However, we note that, for FLCHAIN, Cox-MLP (DeepSurv) has the best integrated Brier score and binomial log-likelihood while Cox-MLP (CC) has the best concordance. This indicates that the proportionality assumption is quite reasonable for this data set.

Again, we find that there is very little difference between Cox-MLP (DeepSurv) and Cox-MLP (CC), which is as expected. The RSF methods generally perform well. Note that the two RSF methods perform equally well here, even though RSF (C^{td}) had the best concordance. Classical Cox (Linear) does rather poorly, which was expected as it has very restrictive model assumptions.

Even though DeepHit, in general, had the best concordance, it has the worst integrated Brier score and binomial log-likelihood in three out of the four data sets. The loss function in DeepHit, given by formula (D.2) in Appendix D, is a convex combination of the negative log-likelihood and a ranking loss, determined by a parameter α. For $\alpha = 1$ we obtain only the negative log-likelihood and for $\alpha = 0$ we obtain only the ranking loss. As we

Data set	Size	Churned	Censored	Prop. Censor	Unique users
Train	1,786,333	1,279,358	506,975	0.28	1,582,202
Test	661,748	473489	188,259	0.28	586,001
Validation	198,665	142,104	56,561	0.28	175,801

Table 5: Summary of the KKBox churn data set.

do hyperparameter tuning based on the concordance, we see that α tends towards smaller values, which results in excellent discriminative performance at the cost of poorly calibrated survival estimates.

6.2. KKBox Churn Case Study

Thus far we have compared competing survival methodologies on fairly small data sets. We now perform a case study on a much larger data set, as this is more interesting in the context of neural networks.

The WSDM KKBox's churn prediction challenge was proposed in preparation for the 11th ACM International Conference on Web Search and Data Mining.[2] The competition was hosted by Kaggle in 2017, with the goal of predicting customer churn on a data set donated by KKBox, the leading music streaming service in Asia. The available data provide us the opportunity to create a survival data set with event times and censoring indicators. We stick with the churn definition given by KKBox, were a customer is considered churned if he or she fails to resubscribe within 30 days after the previous subscription expired. Note, however, that our use of the data is not comparable to the Kaggle competition, as we work with survival times from the start of a subscription period, while they consider durations from a fixed calendar date.

KKBox provides multiple data sources, but as we are primarily interested in evaluating our methods, we spend less time on feature engineering and only use a subset of covariates with general customer information (e.g., city, age, price of subscription). Furthermore, a customer that has previously churned and later resubscribed, is treated as a new customer with some covariate information describing the previous subscription history. This gives us a total of 15 covariates. We split the data into a training, a testing, and a validation set, and some information about these subsets are listed in Table 5. A more in-depth description of the KKBox data set can be found in Appendix B.1.

6.2.1. Methods and Hyperparameter Tuning

We use the same methods as in Section 6.1. However, as this data set is very large, we replace the classical Cox regression with our Cox-SGD (Linear) method from Section 3.2.

We standardize and encode the covariates in the same manner as for the smaller data sets. The networks and training are also the same as earlier, but we multiply the learning rate by 0.8 at the start of every cycle, as we found this to give more stable training. The KKBox data set is quite large, so we are not able to explore as large a hyperparameter space as in the previous experiments. Hence, we do not include weight decay, and perform

2. `https://www.kaggle.com/c/kkbox-churn-prediction-challenge`

Method	Layers	Nodes	Dropout	α	σ
Cox-MLP (DeepSurv)	6	256	0.1	-	-
Cox-MLP (CC)	6	128	0	-	-
Cox-Time	8	256	0	-	-
DeepHit	6	512	0.1	0.001	0.5

Table 6: KKBox model architectures. α and σ only applies the DeepHit (see Appendix D).

Method	C^{td}	IBS	IBLL
Cox-SGD (Linear)	0.816	0.127	-0.406
Cox-MLP (DeepSurv)	0.841	0.111	-0.349
Cox-MLP (CC)	0.844	0.119	-0.379
Cox-Time	0.861	**0.107**	**-0.334**
DeepHit	**0.888**	0.147	-0.489
RSF (Mortality)	0.855	0.112	-0.352
RSF (C^{td})	0.870	0.111	-0.352

Table 7: Evaluation metrics for the KKBox data.

a grid search over a small number of suitable parameters. The hyperparameter search is described in detail in Appendix A.2.

The best configurations are given in Table 6. For RSF, the hyperparameter search based on C^{td} yielded 8 covariates sampled for each split and a minimum of 50 observations in each leaf node. With concordance of mortality as the validation criterion, the best fitted model used 2 covariates for splitting, and a minimum leaf node size of 10. Furthermore, we found that 500 trees were sufficient, as there was little improvement compared to 250 trees.

6.2.2. Results

For evaluation, we fitted each of the methods five times and computed the time-dependent concordance index (C^{td}), the integrated Brier score (IBS), and the integrated binomial log-likelihood (IBLL) of the fitted models. The median scores are presented in Table 7. We use the median because the two proportional Cox-MLP methods yielded rather unstable results, where some of the fitted models performed very badly.

From the table, we see that DeepHit continues to outperform the other methods in terms of concordance while having the worst performance in terms of IBS and IBLL. Furthermore, Cox-Time has the best IBS and IBLL, while still providing a decent concordance. RSF continues to do well across all metrics, while again, the tuning based on C^{td} seems to yield better results than tuning base on the concordance of the mortality. Cox-SGD (Linear) does rather poorly, as it is very restricted, and serves more as a baseline in this context. The Cox-MLP methods seem to again perform reasonably close to each other, at least when taking into account that we found both methods to yield rather unstable results. We are

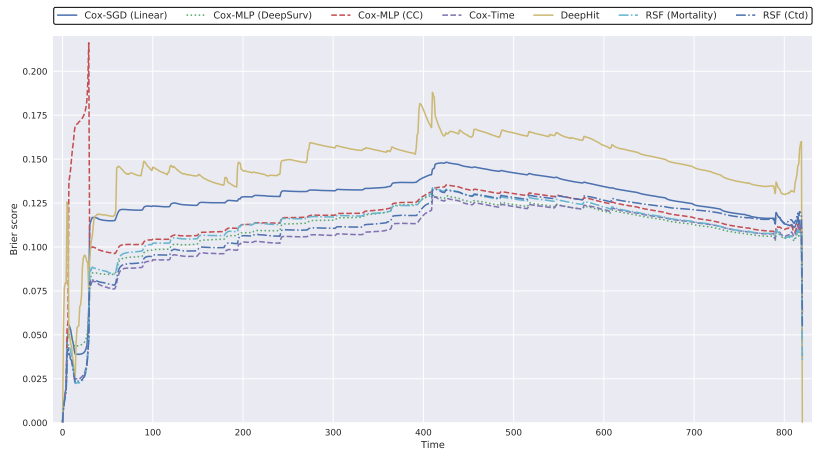


Figure 4: Brier score on KKBox data set. The methods are the same as in Table 7.

not sure why this was the case, but note that the combination of the flexible neural net and the proportionality constraint might be problematic for large data sets.

In Figure 4, we display the Brier scores used to obtain the IBS. Again, we see that DeepHit does poorly for all durations. The instability of the Cox-MLP (CC) is also very apparent for shorter durations. Cox-Time is clearly doing very well for all durations, but interestingly, we observe the Cox-MLP (DeepSurv) provides the best fitted model for larger durations. We could make a corresponding figure for the binomial log-likelihood, but as it is very similar to the Brier score plot, and provide us with no new insights, we have not included it.

6.2.3. Survival Curves

One of the benefits survival models have over binary classifiers is their ability to produce survival curves. In Figure B.1 in Appendix B, we show nine examples of estimated survival curves from Cox-Time on the test data. The curves nicely illustrate the extent of detail the method has learned.

To obtain a more general view of the predictions, we cluster the estimated survival curves of the test set. For an equidistant grid $0 = \tau_0 < \tau_1 < \cdots < \tau_m$, the survival curve of individual i is represented by a vector $[\hat{S}(\tau_0 \,|\, \mathbf{x}_i), \hat{S}(\tau_1 \,|\, \mathbf{x}_i), \ldots, \hat{S}(\tau_m \,|\, \mathbf{x}_i)]$, and by considering these as feature vectors, we apply K-means clustering to the test set with 10 clusters. In Figure 5, we display the cluster centers and the proportions of the test set assigned to each of the clusters. This is a reasonable approach for segmenting customers based on their churning behavior. A natural next step would be to further investigate the clusters, but as we consider this somewhat outside the scope of this paper, we only make

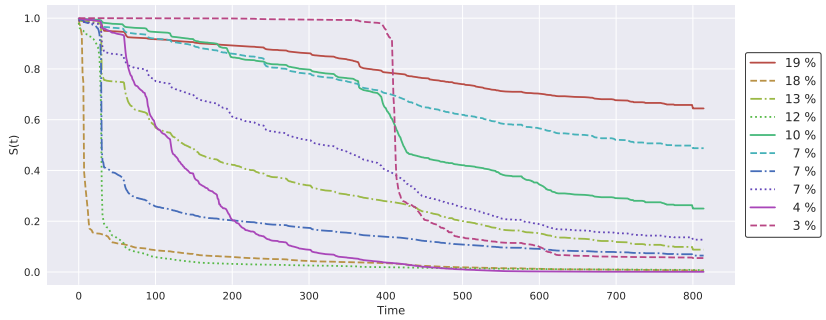


Figure 5: Cluster centers of survival curves from KKBox test data. The centers were generated by K-means clustering on survival curves from Cox-Time. The legend gives the proportion of test data assigned to each cluster (rounded to nearest integer).

a few observations. First, we see that 19 % of the customers are assigned to a cluster that does not provide much detailed information about their behavior, but instead provides a survival curve with a rather constant slope. In sharp contrast, the second largest group (18 %) is at high risk of churning immediately. Furthermore, we observe that many of the curves seem to have higher hazards at the end of each month (drops in the survival curves around 30, 60, 90 days), and we hypothesize that this is a result of customers paying for a full month at a time.

The smallest cluster, constituting only 3 % of the test data, has a sharp drop around day 400. Investigating the covariates of the assigned customers reveals that most of them had prepaid for a 411 days long subscription. However, the large drop after 400 days indicates that only around 25 % of them were interested in continuing their subscription.

Our choice of 10 clusters is mainly motivated by how many curves that can be visualized in a single plot, without being too crowded. Further increasing the number of clusters would likely reveal more detailed behavior.

7. Discussion

In this paper, we propose extensions of the Cox proportional hazards model. By parameterizing the relative risk function of a Cox model with neural networks, we can model rich relationships between the covariates and event times. Furthermore, we allow the networks to model interactions between the covariates and time, resulting in models that are no longer constrained by the proportionality assumption of the Cox model. Building on methods for nested case-control studies, we propose a loss function that can be computed in batches, enabling the models to scale to large data sets. We verify through simulation studies that fitting a Cox model using our proposed loss function gives results close to those obtained using the full Cox partial likelihood.

We compare our suggested methodology with classical Cox regression, random survival forests (RSF), DeepHit, and DeepSurv on 5 real-world data sets, and find that our proposed Cox-Time method performs very well, and has the best overall performance in terms of integrated Brier score (IBS) and integrated binomial log-likelihood (IBLL). DeepHit has, in general, the best discriminative performance, but this comes at the cost of poorly calibrated survival estimates.

Finally, we show how estimated survival curves (event probabilities as functions of time), can be used as a descriptive tool to better understand event-time data sets. This is illustrated by an example where we cluster the survival estimates of a customer churn data set, and show that this customer segmentation provides a useful view of the churning process.

Interesting expansions of our methodology include the extension to handle multiple competing events, time-dependent covariates, dynamic predictions, and recurrent events. Furthermore, it would be of interest to explore other data sources that require more advanced network structures such as convolutions and recurrent neural networks. Finally, less computationally expensive alternatives for creating survival estimates for the Cox-Time method should be explored.

Acknowledgments

This work was supported by The Norwegian Research Council 237718 through the Big Insight Center for research-driven innovation.

Appendix A. Details on Hyperparameter Tuning

In the following, we provide further details about the experiments conducted in Section 6. Here we list hyperparameter configurations and details about the model fitting procedures.

A.1. Four Common Survival Data Sets Tuning

Table A.1 gives the hyperparameter search space used for Rot. & GBSG, SUPPORT, METABRIC, and FLCHAIN. The square brackets describe continuous variables. In the experiments in Section 6.1, we sample 300 random parameter configurations for each method, for each fold of each data set. In the table, "α" and "σ" are given in (D.2) in Appendix D, "Num. durations" are the number of discrete durations (equidistant) used in DeepHit, "λ" is the penalty in (10), "Log duration" refers to a log-transform of the durations passed to Cox-Time, "Ridge" is a ridge penalty used in classical Cox regression, "Split covariates" and "Size leaf" are the number of covariates used for splitting, and the minimum node size of RSF.

A.2. KKBox Tuning

Hyperparameters in the KKBox study were found by a grid search over the relevant parameters in Table A.2. The table consists of three sections, where the top represents the networks, the bottom represents RSF, and the middle contains network parameters that were found on a smaller network with two layers and 128 nodes. "α" and "σ" controls the

Hyperparameter	Values
Layers	{1, 2, 4}
Nodes per layer	{64, 128, 256, 512}
Dropout	[0, 0.7]
Weigh decay	{0.4, 0.2, 0.1, 0.05, 0.02, 0.01, 0}
Batch size	{64, 128, 256, 512, 1024}
α (DeepHit)	[0, 1]
σ (DeepHit)	{0.1, 0.25, 0.5, 1, 2.5, 5, 10, 100}
Num. durations (DeepHit)	{50, 100, 200, 400}
λ (Cox-Time and Cox-MLP (CC))	{0.1, 0.01, 0.001, 0}
Log durations (Cox-Time)	{True, False}
Ridge (Cox (Linear))	{1000, 100, 10, 1, 0.1, 0.01, 10^{-3}, 10^{-4}, 10^{-5}}
Split covariates (RSF)	{2, 4, 6, 8}
Size leaf (RSF)	{2, 8, 32, 64, 128}

Table A.1: Hyperparameter search space for experiments on Rot. & GBSG, SUPPORT, METABRIC, and FLCHAIN.

Hyperparameter	Values
Layers	{4, 6, 8}
Nodes per layer	{128, 256, 512}
Dropout	{0, 0.1, 0.5}
α(*)	{0, 0.001, 0.1, 0.2, 0.5, 0.8, 0.9, 0.99, 0.999, 1}
σ(*)	{0.01, 0.1, 0.25, 0.5, 1, 10, 100}
Log durations(*)	{True, False}
Split covariates	{2, 4, 6, 8}
Size leaf	{8, 10, 20, 50}

Table A.2: KKBox hyperparameter configurations. (*) denotes parameters found with a two layer network with 128 nodes.

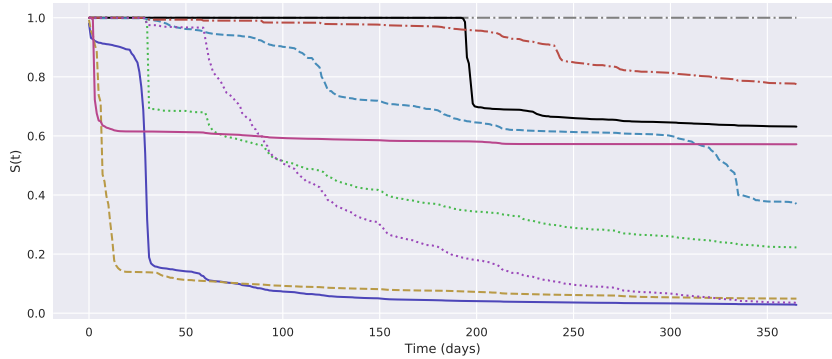


Figure B.1: Survival curves from the case study in Section 6.2, which models the times at which customers of a streaming service churn. Each curve gives the estimated probabilities for a customer not having churned (probabilities of still being a customer at any given time). The time axis shows the number of days since first subscription. The curves are generated by the Cox-Time method from Section 3.4.

loss function of DeepHit, and we assumed it should generalize well across network structures. The same goes for whether or not we should log-transform the durations passed to Cox-Time. Hence, to reduce the hyperparameter search, we found suitable values with a smaller network.

For the proposed Cox-MLP (CC) and Cox-Time, we used a fixed penalty $\lambda = 0.001$ in (10). All networks were trained with batch size of 1028, and the best performing architectures can be found in Table 6.

Appendix B. Details from KKBox Churn Case Study

In the following, we provide some details of the KKBox case study that were exempt from the main article.

In Figure B.1, we show an example of nine survival curves estimated by Cox-Time on the KKBox data set. Each line represents an individual from the test set. It is clear that the Cox-Time method has learned to represent a variety of survival curves.

B.1. KKBox Data Set

KKBox provides data consisting of general customer information (city, age, gender, initial registration), transaction logs listing how customers manage their subscription plans, and user logs containing some aggregate information of the customers' usage of the streaming

service. KKBox defines a customer as churned if he or she has failed to resubscribe to their service more than 30 days after their last subscription expired.

Through the transaction information, we can create a data set with survival times and censoring indicators. We keep KKBox's original churn definition of 30 days absence and calculate the survival time from the date of the first subscription or the earliest record of the customer in the transaction logs. Customers that have previously churned but resubscribed, are treated as new customers with some covariate information describing their previous subscription history.

All covariates are calculated at the time of subscription of each customer, and they do not change with time. This is a simplification, as the covariates of a customer are typically not stationary. However, as our objective is to evaluate the proposed methodology, we refrain from doing extensive feature engineering. In this regard, we further disregard the user logs, as we get a reasonable set of covariates from the customer and transaction information.

The data sets are summarized in Table 5. As some of the customers have churned multiple times, and are therefore included multiple times, the table also includes the number of unique customers in the respective data sets.

We have a total of 15 covariates, where 7 are numeric and 8 are categorical. The numerical covariates give the time between subscriptions (for customers that have previously churned), number of previous churns, time since first registration, number of days until the current subscription expires, listed price of the current subscription, the amount paid for the subscription, and age. All numerical covariates, except the number of previous churns, were log-transformed. The categorical covariates are gender (3 categories), city (22 categories), how the customer registered (6 categories), and 5 indicator variables specifying if an individual has previously churned, if the subscription is canceled, if the age is unrealistic, if the subscription is automatically renewed, and if we do not know when the customer first registered.

Appendix C. Additional Simulations

In the following, we provide some additional simulations used to further investigate the proposed methods, and in Appendix C.3 we explain how the simulated data were generated. We again stress that the aim of the simulations is only to verify the expected behavior of our methods, and should not be interpreted as a general evaluation of them.

C.1. Non-Linear Hazards

Continuing from the simulations in Section 5, we do a simple study of the increased flexibility provided by replacing the linear predictor in a Cox regression by a neural network. To evaluate this, we extend the simulations in Section 5 by replacing $g(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{x}$ with the non-linear function

$$g(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{x} + \frac{2}{3}(x_1^2 + x_3^2 + x_1x_2 + x_1x_3 + x_2x_3), \tag{C.1}$$

where x_i denotes element i of $\mathbf{x}$. The simulations are otherwise unchanged from the linear case in Section 5. We sample 10,000 training samples, 10,000 test samples, and 1,000 samples used for validation (validation data is used for early stopping), and fit the Cox-SGD and

Figure C.1: Partial log-likelihood estimates of Cox-SGD and Cox-MLP plotted against the true partial log-likelihoods, for 2,000 samples of the test set.

classical Cox regression from Section 5. Additionally, we fit the Cox model in Section 3.3, where $g(\mathbf{x})$ is parameterized by a one-hidden layer MLP (multilayer perceptron) with 64 hidden nodes and ReLU activations. This model will be referred to as *Cox-MLP* (we drop (CC) as we do not consider DeepSurv here).

In Figure C.1 we have, for 2,000 individuals of the test set, plotted the individual partial log-likelihood estimates of Cox-SGD and Cox-MLP against the true individual partial log-likelihoods. That is, for each i with $D_i = 1$, we plot

$$\hat{\ell}_i = -\log\left(\sum_{j\in\mathcal{R}_i} \exp[\hat{g}(\mathbf{x}_j) - \hat{g}(\mathbf{x}_i)]\right)$$

against the true ℓ_i. The closer a method estimates $g(\mathbf{x})$ to the true predictor in (C.1), the closer the scatter plot should be to the identity function (straight line with slope 1 through the origin). As expected, we see that Cox-MLP produces likelihood estimates very close to the true likelihoods, while Cox-SGD struggles to represent this non-linear function.

C.2. Non-Proportional Hazards

In our final simulations, we investigate the effect of removing the proportionality constraint in the Cox models. Building on the previous simulations, we add a time dependent term to the risk function. The hazard function is now given by $h(t\,|\,\mathbf{x}) = h_0 \exp[g(t, \mathbf{x})]$, with $h_0 = 0.02$ and

$$\begin{aligned} g(t, \mathbf{x}) &= a(\mathbf{x}) + b(\mathbf{x})\, t, \\ a(\mathbf{x}) &= g_{ph}(\mathbf{x}) + \text{sign}(x_3), \\ b(\mathbf{x}) &= |0.2\,(x_0 + x_1) + 0.5\, x_0 x_1|, \end{aligned}$$

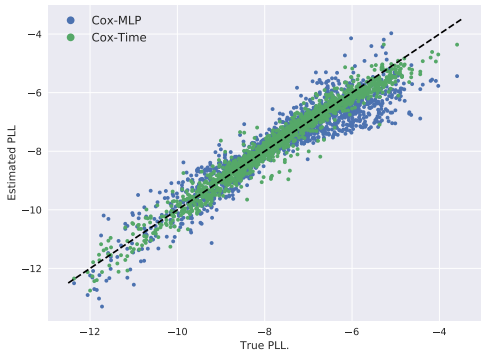


Figure C.2: Partial log-likelihood estimates of Cox-MLP and Cox-Time plotted against the true partial log-likelihoods, for 2,000 samples of the test set.

were $g_{ph}(\mathbf{x})$ is the function in (C.1). The simulations are otherwise unchanged from the previous experiments. We require the term $b(\mathbf{x})$ to be non-negative to ensure that the hazards increase with time. Furthermore, we have added $\text{sign}(x_3)$ to $a(x)$ as this is essentially equivalent to having two different baselines, $\tilde{h}_0 = h_0 \exp[\text{sign}(x_3)] \in \{0.0074, 0.054\}$. Both of these choices are motivated by our attempt to produce reasonable looking survival curves.

We sample 10,000 training samples, 10,000 test samples, and 2,000 samples for validation, and fit a Cox-MLP model and a Cox-Time model. Both Cox-MLP and Cox-Time parameterize g with a 4 hidden layer MLP with ReLU activations, 128 nodes in each layer, and dropout between layers with rate 0.1.

In Figure C.2, we have plotted the estimated partial log-likelihoods of Cox-MLP and Cox-Time against the true partial log-likelihoods for 2,000 samples of the test set. Though the difference between the methods is not very large, Cox-Time appears to capture the true values better than Cox-MLP.

To further illustrate some of the differences between Cox-MLP and Cox-Time, we have in Figure C.3 plotted nine survival curves $\hat{S}(t\,|\,\mathbf{x})$ from the test set. The figure shows both the true curves and the curves estimated by the two methods. It is seen that Cox-Time is able to estimate the true survival curves better than those produced by Cox-MLP.

C.3. Simulation Details

In the following, we explain in detail how we generated our simulated data sets in Section 5 and Appendices C.1 and C.2. We want to generate survival times T^* from relative risk models of the form

$$h(t\,|\,\mathbf{x}) = h_0(t)\exp[g(t,\mathbf{x})]. \tag{C.2}$$

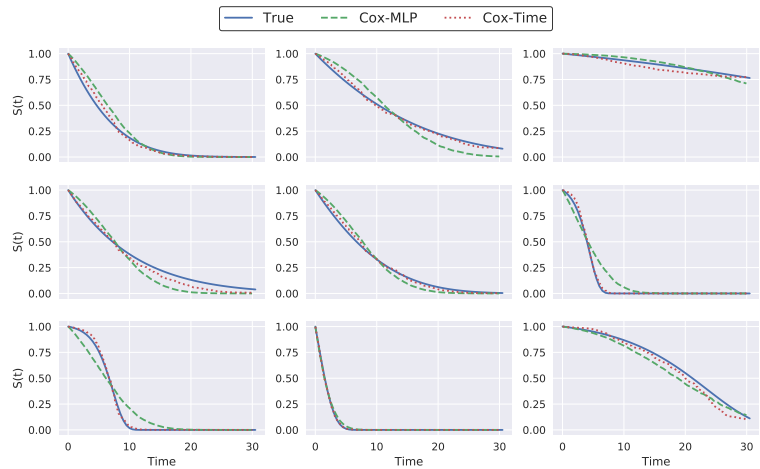


Figure C.3: Examples of survival curves $\hat{S}(t\,|\,\mathbf{x})$ from the test set in the non-proportional simulation study.

If the function g does not depend on t, we have a proportional hazards model, which is just a special case of the relative risk models. Let $H(t\,|\,\mathbf{x})$ denote the continuous (increasing) cumulative hazard $H(t\,|\,\mathbf{x}) = \int_0^t h(s\,|\,\mathbf{x})\,ds$, and let V be exponentially distributed with parameter 1. Then we can obtain survival times through the inverse cumulative hazard $T^* = H^{-1}(V\,|\,\mathbf{x})$. This can be shown to hold through the hazard's relationship to the survival function in (1),

$$S(t\,|\,\mathbf{x}) = \mathrm{P}(T^* > t\,|\,\mathbf{x}) = \mathrm{P}(H^{-1}(V\,|\,\mathbf{x}) > t) = \mathrm{P}(V > H(t\,|\,\mathbf{x})) = \exp[-H(t\,|\,\mathbf{x})],$$

as V is exponentially distributed with $\mathrm{P}(V > v) = \exp(-v)$. Hence, we can obtain survival times by transforming generated samples from an exponential distribution.

To obtain an analytical expression for the inverse cumulative hazard $H^{-1}(v\,|\,\mathbf{x})$, we restrict the models in (C.2) to have constant baseline h_0, in addition to a simple time dependence in $g(t,\mathbf{x})$. For the linear predictor in Section 5 and the non-linear predictor in Appendix C.1 we simulate with proportional hazards, meaning $g(t,\mathbf{x}) = g(\mathbf{x})$. Hence, we get cumulative hazards and inverse cumulative hazards of the form

$$H(t\,|\,\mathbf{x}) = t\,h_0\exp[g(\mathbf{x})],$$
$$H^{-1}(v\,|\,\mathbf{x}) = \frac{v}{h_0\exp[g(\mathbf{x})]}.$$

In the non-proportional simulations in Appendix C.2, we add a time dependent term $g(t, \mathbf{x}) = a(\mathbf{x}) + b(\mathbf{x})\, t$, which gives us

$$H(t \,|\, \mathbf{x}) = \frac{h_0 \exp[a(\mathbf{x})]\,(\exp[b(\mathbf{x})\, t] - 1)}{b(\mathbf{x})},$$
$$H^{-1}(v \,|\, \mathbf{x}) = \frac{1}{b(\mathbf{x})} \log\left(1 + \frac{v\, b(\mathbf{x})}{h_0\, \exp[a(\mathbf{x})]}\right).$$

Appendix D. DeepHit

DeepHit by Lee et al. (2018) is a discrete survival model for competing risks. However, as we only consider one type of event, we will express the method in terms of a single cause. DeepHit considers time to be discrete, so to fit it to the continuous-time data sets in Section 6, we discretize the event times with an equidistant grid between the smallest and largest duration in the training set. The number of discrete time-points is considered a hyperparameter, given by "Num. durations" in Table A.1.

Now, assume time is discrete with $0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_m$. Let $\mathbf{y}(\mathbf{x}) = [y_0(\mathbf{x}), y_1(\mathbf{x}), \ldots, y_m(\mathbf{x})]^T$ denote the output of a neural net with covariates $\mathbf{x}$ and a softmax output activation. Then, $\mathbf{y}(\mathbf{x})$ can be interpreted as the estimated probability mass function of the duration times $y_j(\mathbf{x}_i) = \hat{\mathrm{P}}(T_i^* = \tau_j \,|\, \mathbf{x}_i)$. The estimated survival function is then given by

$$\hat{S}(\tau_j \,|\, \mathbf{x}) = 1 - \sum_{k=1}^{j} y_k(\mathbf{x}),$$

and the discrete negative log-likelihood corresponding to the continuous version in (2), is

$$\mathrm{loss}_L = -\sum_{i=1}^{N} \left[D_i \log(y_{e_i}(\mathbf{x}_i)) + (1 - D_i) \log(\hat{S}[T_i \,|\, x_i]) \right].$$

Here, e_i denotes the index of the event time for observation i, i.e., $T_i = \tau_{e_i}$. Furthermore, DeepHit adds a second loss that attempts to improve its ranking capabilities,

$$\mathrm{loss}_{\mathrm{rank}} = \sum_{i,j} D_i\, \mathbb{1}\{T_i < T_j\} \exp\left(\frac{\hat{S}(T_i \,|\, \mathbf{x}_i) - \hat{S}(T_i \,|\, \mathbf{x}_j)}{\sigma}\right). \tag{D.1}$$

The loss of DeepHit is a combination of these two losses, where the ranking loss is scaled by a constant. We deviate slightly from the original implementation here and instead use a convex combination of the two,

$$\mathrm{loss} = \alpha\, \mathrm{loss}_L + (1 - \alpha)\, \mathrm{loss}_{\mathrm{rank}}, \tag{D.2}$$

where α and σ from (D.1) are considered hyperparameters.

References

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in Medicine*, 24(24):3927–3944, 2005.

David R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

Cameron Davidson-Pilon, Jonas Kalderstam, Ben Kuhn, Andrew Fiore-Gartland, Luis Moneda, Paul Zivich, Alex Parij, Kyle Stark, Steven Anton, Lilian Besson, et al. Camdavidsonpilon/lifelines: v0.14.1, 2018.

Lore Dirick, Gerda Claeskens, and Bart Baesens. Time to default in credit scoring using survival analysis: a benchmark study. *Journal of the Operational Research Society*, 68 (6):652–665, 2017.

David Faraggi and Richard Simon. A neural network model for survival data. *Statistics in Medicine*, 14(1):73–82, 1995.

Stephane Fotso. Deep neural networks for survival analysis based on a multi-task framework. *arXiv preprints arXiv:1801.05512*, 2018.

Thomas A. Gerds, Michael W. Kattan, Martin Schumacher, and Changhong Yu. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in Medicine*, 32(13):2173–2184, 2012.

Larry Goldstein and Bryan Langholz. Asymptotic theory for nested case-control sampling in the Cox regression model. *Annals of Statistics*, 20(4):1903–1928, 1992.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 18(17-18):2529–2545, 1999.

Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

Frank E. Harrell Jr, Robert M. Califf, David B. Pryor, Kerry L. Lee, and Robert A. Rosati. Evaluating the yield of medical tests. *Journal of the American Medical Association*, 247 (18):2543–2546, 1982.

Patrick J. Heagerty and Yingye Zheng. Survival model predictive accuracy and ROC curves. *Biometrics*, 61(1):92–105, 2005.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *arXiv preprints arXiv:1609.04836*, 2017.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *Annals of Applied Statistics*, 2(3):841–860, 2008.

Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1), 2018.

Nitish S. Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

John P. Klein and Melvin L. Moeschberger. *Survival Analysis: Techniques for Censored and Truncated Data*. Springer, New York, 2. edition, 2003.

Bryan Langholz and Larry Goldstein. Risk set sampling in epidemiologic cohort studies. *Statistical Science*, 11(1):35–53, 1996.

Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Margaux Luck, Tristan Sylvain, Héloïse Cardinal, Andrea Lodi, and Yoshua Bengio. Deep learning for patient-specific kidney graft survival analysis. *arXiv preprint arXiv:1705.10245*, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Daniel J. Sargent. Comparison of artificial neural networks with other statistical approaches. *Cancer*, 91(8):1636–1642, 2001.

L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017.

Gian A. Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2015.

Terry M. Therneau. A package for survival analysis in S. Version 2.38, 2015.

Duncan C. Thomas. Addendum to: Methods of cohort analysis: appraisal by application to asbestos mining, by F. D. K. Liddell, J, C. McDonald and D. C. Thomas. *Journal of the Royal Statistical Society: Series A (General)*, 140(4):469–491, 1977.

Dirk Van den Poel and Bart Lariviere. Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, 157(1): 196–217, 2004.

Antonio Vigan, Marlene Dorgan, Jeanette Buckingham, Eduardo Bruera, and Mari E. Suarez-Almazor. Survival prediction in terminal cancer patients: a systematic review of the medical literature. *Palliative Medicine*, 14(5):363–374, 2000.

Anny Xiang, Pablo Lapuerta, Alex Ryutov, Jonathan Buckley, and Stanley Azen. Comparison of the performance of neural network methods and Cox regression for censored survival data. *Computational Statistics & Data Analysis*, 34:243–257, 2000.

Safoora Yousefi, Fatemeh Amrollahi, Mohamed Amgad, Chengliang Dong, Joshua E. Lewis, Congzheng Song, David A. Gutman, Sameer H. Halani, Jose E. V. Vega, Daniel J. Brat, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific Reports*, 7(1):11707, 2017.

Xinliang Zhu, Jiawen Yao, and Junzhou Huang. Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 544–547, 2016.

Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang. WSISA: Making survival prediction from whole slide histopathological images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6855–6863, July 2017.

Journal of Machine Learning Research 20 (2019) 1-47 Submitted 8/18; Revised 6/19; Published 8/19

Logical Explanations for Deep Relational Machines Using Relevance Information

Ashwin Srinivasan ASHWIN.SRINIVASAN@WOLFSON.OXON.ORG
Department of Computer Sc. & Information Systems
BITS Pilani, K.K. Birla Goa Campus, Goa, India

Lovekesh Vig LOVEKESH.VIG@TCS.COM
TCS Research, New Delhi, India

Michael Bain M.BAIN@UNSW.EDU.AU
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia.

Editor: Luc De Raedt

Abstract

Our interest in this paper is in the construction of symbolic explanations for predictions made by a deep neural network. We will focus attention on deep relational machines (DRMs: a term introduced in Lodhi (2013)). A DRM is a deep network in which the input layer consists of Boolean-valued functions (features) that are defined in terms of relations provided as domain, or background, knowledge. Our DRMs differ from those in Lodhi (2013), which uses an Inductive Logic Programming (ILP) engine to first select features (we use random selections from a space of features that satisfies some approximate constraints on logical relevance and non-redundancy). But why do the DRMs predict what they do? One way of answering this was provided in recent work Ribeiro et al. (2016), by constructing readable proxies for a black-box predictor. The proxies are intended only to model the predictions of the black-box in local regions of the instance-space. But readability alone may not be enough: to be understandable, the local models must use relevant concepts in an meaningful manner. We investigate the use of a Bayes-like approach to identify logical proxies for local predictions of a DRM. As a preliminary step, we show that DRM's with our randomised propositionalization method achieve predictive performance that is comparable to the best reports in the ILP literature. Our principal results on logical explanations show: (a) Models in first-order logic can approximate the DRM's prediction closely in a small local region; and (b) Expert-provided relevance information can play the role of a prior to distinguish between logical explanations that perform equivalently on prediction alone.

Keywords: Explainable AI, Inductive Logic Programming, Deep Relational Machines

1. Introduction

In Stewart (1994) a contrast is presented between theories that predict everything, but explain nothing, and those that explain everything, but predict nothing. Both are seen as having limited value in the scientific enterprise, which requires models with both predictive

©2019 Ashwin Srinivasan and Lovekesh Vig and Michael Bain.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-517.html.

and explanatory power. Michie and Johnston (1984) add a further twist to this by suggesting that the limitations of the human brain may force a "window" on the complexity of explanations that can be feasibly understood, even if they are described in some readable form like a natural or a symbolic formal language.

Complexity limits notwithstanding, it is often assumed that that predictive and explanatory assessments refer to the *same* model. But this does not have to be so: early results in the Machine Learning (ML) literature on behavioural cloning point to areas where people perform predictive tasks using sub-symbolic models, for which entirely separate symbolic explanations were constructed by machine (Michie, 1986). It is not therefore inevitable that machine learning models be either purely sub-symbolic or purely symbolic.

There are, of course, problems for which one or the other is much better suited. Prominent recent examples of successful sub-symbolic learning abound with the use of deep neural networks in automatic caption learning ("woman in a white dress standing with tennis racquet": Karpathy and Li (2015)), speech-recognition (Siri, Google Now) and machine translation (Google Translate). It is equally evident that for learning recursive theories, program synthesis, and learning with complex domain-knowledge, symbolic techniques like Inductive Logic Programming (ILP) have proved to be remarkably effective. But there are also a number of problems that could benefit from an approach that requires not one or the other, but both forms of learning (in his book "Words and Rules", Steven Pinker conjectures language learning as one such task).

Our interest here therefore is in investigating the use of separate models for prediction and explanation. Specifically, we examine the use of a neural model for prediction, and a logical model for explanation. But in order to ensure that the models are about the same kinds of things, the models are not constructed independent of each other. The neural model is constructed in terms of features in (first-order) logic identified by a simple form of propositionalization methods developed in ILP. In turn, the predictions of the neural model are used to construct logical models in terms of the features.[1] Figure 1 shows how the various pieces we have just described fit together.

Having two separate models would appear to be an extravagance that could be ill-afforded if the neural model wasn't a very good predictor, or if the logical model wasn't a very good explainer.

In this paper, we are concerned with the analysis of relational data. That is, a data instance is described not just by values of some pre-defined attributes, but also by by pre-defined relations. A good example of such a data instance is a molecule known to target a malarial parasite. Such a molecule not only has attributes like its molecular weight, solubility, hydrophobicity and so on, but also has connected ring-structures, functional groups, and the like. For predictions on data like these, we use a specific form of neural models called *deep relational machines* or DRMs. The term, originally introduced in Lodhi (2013), can be broadly understood as a class of neural models in which the inputs include Boolean features defined using known relations. For example, a Boolean feature for molecular data may be $TRUE$ if the molecule has "a 7-member ring connected to a lactone ring with a peroxide bridge" and $FALSE$ otherwise. Here definitions of relations like a 7-member ring, lactone ring, and a peroxide bridge are known. In its original formulation, a DRM relied

1. We restrict ourselves in this paper to queries of the form: "What is the class of instance x?", but the ideas extend straightforwardly to tasks other than classification.

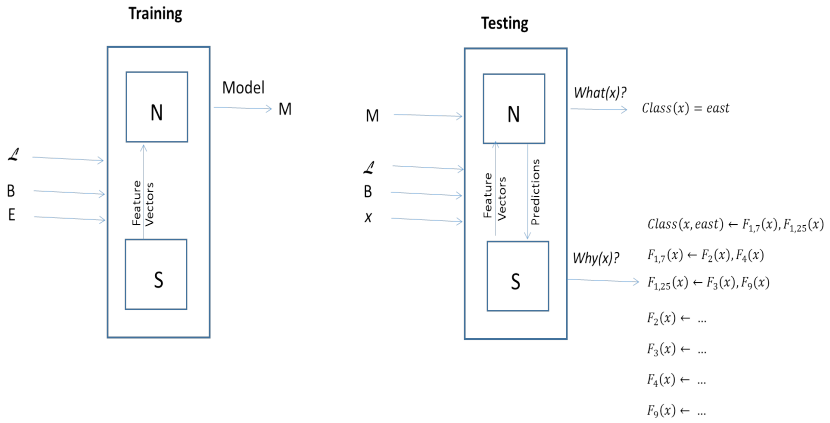


Figure 1: What this paper is about. N is a deep neural network, and S is a symbolic learner. B is domain-specific background knowledge; $\mathcal{L}$ language constraints; E a set of training examples. N is used to construct a predictive model M. Explanations of why M predicts the class of a test instance x are to be obtained as a logical model that uses features, that are pre-defined in B or "invented". Correctly, B and $\mathcal{L}$ on the right are augmentations of those on the left; and the S's on right and left are different instantiations of a first-order symbolic learner that is capable of producing definitions of features (left), or explanations (right).

on selection of Boolean features using Inductive Logic Programming (Muggleton and De Raedt, 1994). More recently, it has been shown in an extensive empirical study that DRMs with stochastic selection of relational features can achieve predictive accuracies that are often better than the state-of-the-art (Dash et al., 2018). We will therefore take this form of a DRM as the basis for prediction. Using the DRMs as predictors, our interest is in developing symbolic explanations for the predictions. Specifically:

- We present a method for extracting local symbolic explanations in terms of the input features of the DRM by employing a Bayes-like trade-off of predictive-fit and prior preference. The latter is obtained using domain-knowledge of relevance of predicates used in the explanations.

The combination of using prediction with relational features and relevance information extends the techniques proposed in Ribeiro et al. (2016) in the direction of making explanations both more expressive (since the features are definitions in first-order logic) and meaningful (by incorporating domain-specific preferences into explanations). We will demonstrate the use of combining predictive performance and relevance on real-life biochemical problems for which we have the necessary relevance information. However, the approach we propose can

prove useful to any relational learning task in which there is at least a partial-ordering over domain-predicates. These applications could be as diverse as recommender systems (for example, movies directed by certain directors may be more relevant to recommendations for a particular genre), classification (for example, when paper citations are to be classified as being the same, it may be equally relevant to check similarities in title, or the list of authors, but less relevant to examine the paper's keywords), or function-prediction (for example, when predicting gene-function using relations in the Gene Ontology, predicates with evidence-codes based on experimental evidence may be more relevant than those based on computational evidence). It may not be necessary to have prior knowledge for this — recent work suggests that it may be possible, for any relational learning task, to derive such a partial-ordering automatically from data, using a probabilistic definition of the generality of domain-predicates.[2] A decision-theoretic technique for determining relevant predicates for general relational learning tasks was proposed earlier in Srinivasan (2001).

The rest of the paper is organised as follows. In Section 2 we describe the DRMs we use for prediction. Section 3 describes the notion of explanations as used in this paper. Selection amongst several possible explanations is in Section 4, which introduces the use of a relevance-based prior in Section 4.1. Section 5 presents an empirical evaluation of the predictive and explanatory models, using some benchmark datasets. Appendix C contains details of the domain-specific relevance information used in the experiments.

2. A Deep Relational Machine for Prediction

One of the most remarkable recent advances in the area of Machine Learning is a resurgence of interest in neural networks, resulting from the automated construction of "deep networks" for prediction. Simplistically, Deep Learning is using neural networks with multiple hidden layers. Mathematically speaking, it is a composition of multiple simple non-linear functions trying to learn a hierarchy of intermediate features that most effectively aid the global learning task. Learning such intermediate features with neural networks has been made possible by three separate advances: (a) mathematical techniques that allow the training of neural networks with very large numbers of hidden layers; (b) the availability of very large amounts of data that allow the estimation of parameters (weights) in such complex networks; and (c) the advanced computational capabilities offered by modern day GPUs to train deep models. Despite successes across a wide set of problems, deep learning is unlikely to be sufficient for all kinds of data analysis problems. The principal difficulties appear to lie in the data and computational requirements to train such models. This is especially the case if many hidden layers are needed to encode complex concepts (features). For many industrial processes, acquiring data can incur significant costs, and simulators can be computationally very intensive.

Some of this difficulty may be alleviated if knowledge already available in the area of interest can be taken into account. Consider, for example, a problem in the area of drug-design. Much may be known already about the target of interest: small molecules that have proved to be effective, what can and cannot be synthesized cheaply, and so on. If these concepts are relevant to constructing a model for predicting good drugs, it seems both unnecessary and inefficient to require a deep network to re-discover them (the

2. See: `https://www.doc.ic.ac.uk/~shm/Papers/relevance18.pdf`

problem is actually worse — it may not even be possible to discover the concepts from first-principles using the data available). It is therefore of significant practical interest to explore ways in which prior domain-knowledge could be used in deep networks to reduce data and computational requirements.

Deep Relational Machines, or DRMs, proposed in Lodhi (2013), are deep neural networks with first-order Boolean functions at the input layer ("function F_1 is true if the instance x is a molecule containing a 7-membered ring connected to a lactone ring" — definitions of relations like 7-membered and lactone rings are expected to be present in the background knowledge).

We note that DRMs are one example of a much broader class of neural-symbolic models, which are surveyed in Besold et al. (2017). In ILP, pioneering work on the combination of neural-networks and symbolic features has been done by d'Avila Garcez and Zaverucha (1999) and extended in França et al. (2014, 2015). For the purposes of this paper, we will use the term "DRM" to mean a neural network with multiple hidden layers, and an input layer with values for first-order Boolean functions. We will further assume that the function definitions employ predicates that are defined as part of domain- or background-knowledge. In this sense, the neural networks in França et al. (2015) for example are DRMs, and the definitions and techniques for constructing explanations employed here apply without change to those models.

2.1. Input Features for a DRM

Ideally, we would like the inputs for a DRM to consist of all possible relational features, and let the network's training process decide on the features that are actually useful (in the usual manner: a feature that has 0 weights for all out-going edges is not useful). The difficulty with this is that the number of such features in first-order logic can be very large, often impractical to enumerate completely. In Lodhi (2013) the functions are learned using an Inductive Logic Programming (ILP) engine[3]. This follows a long line of research, sometimes called *propositionalization*, in which features constructed by ILP have been used by other learning methods like regression, decision-trees, SVMs, topic-models, and multiplicative weight-update linear threshold models, and neural-network models. In each of these, the final model is constructed in two steps: first, a set of features is selected, and then, the final model is constructed using these features, possibly in conjunction with other features already available. Usually the models show significant improvements in predictive performance when an existing feature set is enriched in this manner.

In Lodhi (2013), deep networks with ILP-features are shown to perform well, although the empirical evidence is limited. Recently, a simple approach for stochastic selection of relational features was introduced in Vig et al. (2018) in the context of evaluating embeddings. This form of randomised feature-selection has since been evaluated extensively in Dash et al. (2018) and found to give predictive performance better than the state-of-the-art for classification problems.

In this paper, we will use this form of stochastic selection of relational features as inputs for a DRM. For completeness, we clarify first what we mean by a relational instances and feature.

3. Terminology from Logic Programming and ILP is defined in Appendix A.

Definition 1 Relational Examples for Classification. *The set of examples provided can be defined as a binary relation Class which is a subset of the Cartesian product $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the set of relational instances (for simplicity, and without loss of generality, we will take each instance to be a ground first-order term), and $\mathcal{Y}$ denotes the finite set of class labels. Any single element of the set of relational examples will be denoted by $Class(a, c)$ where $a \in \mathcal{X}$ and $c \in \mathcal{Y}$.*

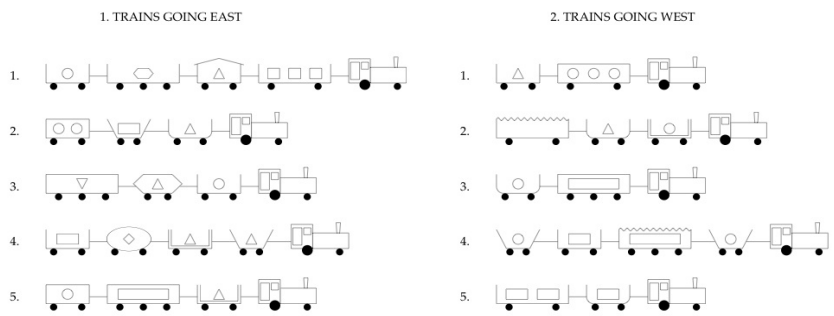
Definition 2 Relational Features. *A relational feature is a unary function $F_i : \mathcal{X} \mapsto \{TRUE, FALSE\}$ defined in terms of a conjunction of predicates $Cp_i(x)$. Each predicate in $Cp_i(x)$ is defined as part of domain- or background-knowledge B. That is, $F_i(x) \mapsto TRUE$ iff $Cp_i(x) \mapsto TRUE$ and FALSE otherwise. We will represent each relational feature as a single definite clause $\forall x\ (F_i(x) \leftarrow Cp_i(x))$ in a logic program, relying on the closed-world assumption for the complete definition of F_i. We will sometimes call the relational feature F_i simply the* feature F_i*, and the definite-clause definition for F_i the* feature-definition *for F_i. If the feature-definition of F_i is in B, we will sometimes say "feature F_i is in B." We will usually denote the set of features in B as $\mathcal{F}_B$ or simply $\mathcal{F}$.*

In terms of the classes of features identified in Saha et al. (2012), our definition corresponds to the least constrained class of features (F_d) in that paper. The relational features selected by an ILP engine in Lodhi (2013) are from this class, and those in França et al. (2014, 2015) are from the class of *independent* features (F_i) in Saha et al. (2012).

Definition 3 Classification clause. *A clause for classifying a relational example $Class(a, c)$ is a clause $\forall x(Class(x, c) \leftarrow Cp(x))$, where $Cp(x)$ is a conjunction of predicates. Each predicate in $Cp(x)$ is defined as part of domain- or background-knowledge B.*

It is evident from the definitions that some or all of a classification clause can be converted into relational features.

Example 1 The trains problem. *The figure below shows a well-known synthetic problem in the ILP literature, Michalski's "Trains" problem, originally posed in Michalski (1983):*



There are two sets of trains: Eastbound and Westbound. Descriptors for the trains include: number, shape and lengths of the car, shape of load, and so on. The task is to determine classification rules to distinguish Eastbound trains from Westbound ones.

For this problem each relational example is a pair, consisting of a ground first-order term and a class label. The ground term represents the train (for example $Train(Car(Long, Open, Rect, 3), Car(Short, Closed, Triangle, 1), \ldots))$ *and the label denotes whether or not it is Eastbound.*

We assume also that we have access to domain (or background) knowledge B *that allows us to access aspects of the relational instance such as* $Has_Car(Train(\ldots), Car(\ldots))$, $Closed(Car(\ldots))$ *and so on. Then a clause for classifying Eastbound trains is (we leave out the quantifiers for simplicity):*

$$Class(x, East) \leftarrow Has_Car(x, y), Short(y), Closed(y)$$

Here $Cp(x) = Has_Car(x, y) \wedge Short(y) \wedge Closed(y)$. *The following relational features can be obtained from this classification clause:*

$$F_1(x) \leftarrow Has_Car(x, y), Short(y)$$

and:

$$F_2(x) \leftarrow Has_Car(x, y), Closed(y)$$

The classification clause could now be written as:

$$Class(x, c) \leftarrow F_1(x), F_2(x)$$

Clauses like the last one in the example above will be called feature-clauses. Somewhat informally:

Definition 4 Feature-clause. *A feature-clause is a clause in which all negative literals are relational features.*

We note that "propositionalization" of relational data is simply a conversion of a relational instance into a vector-form, using relational features.

Definition 5 Feature Vector of a Relational Instance. *For a set of relational features* $\mathcal{F}$ *ordered in some canonical sequence* $(F_1, F_2, \ldots, F_d)$ *we obtain a Boolean-vector representation* $\mathbf{x}' \in \{0,1\}^d$ *of a relational instance* $x \in \mathcal{X}$ *using the function* $FV : \mathcal{X} \rightarrow \{0,1\}^d$, *where the* i^{th} *component* $FV_i(x) = 1$ *if* $F_i(x) \mapsto TRUE$, *and 0 otherwise. We will sometimes say* $\mathbf{a}'$ *is the* feature-space *representation of relational instance* a.

In this paper, the components for predicting the class-label of a relational instance a using a DRM M are shown in Fig. 2. This is a 2-step process: (1) The feature-vector representation $\mathbf{a}'$ of a is obtained using the feature-definitions found during the training stage; and (2) $\mathbf{a}'$ is provided as input to the model M found during training, which then computes the class-label (Fig. 3).
Although we are only concerned with DRMs in this paper, the logical explanations we construct apply more broadly to black-box models that use a feature-vector represent ion of relational instances in the sense defined in Defn. 5.

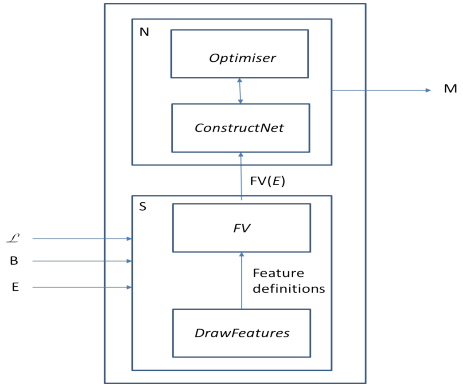


Figure 2: More details on the training component in Fig. 1. $\mathcal{L}$ denotes language constraints on the relational features, and $DrawFeatures$ is the stochastic feature construction procedure described in Vig et al. (2018). Features are used to represent ("propositionalize") each relational instance in the training data as a feature-vector. FV(E) denotes the feature-vectors of all relational instances E in the training data. Model M is a DRM. The optimiser finds the best structure and parameters for the DRM using FV(E).

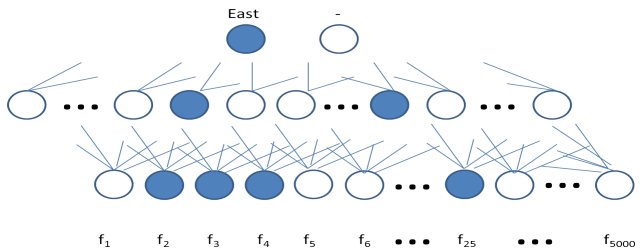


Figure 3: Prediction of the class of a relational instance a by a DRM. The inputs f_i are the component values of the feature-vector $FV_i(a)$. The shaded inputs denote a feature-value of 1. With a DRM, there is more to $f_i = 1$ than just an assignment of 1 to feature: it also means that some conjunction of background predicates $Cp_i(a)$ is $TRUE$ for the relational instance a. The shaded circles in hidden layers denote activated nodes. The shaded output node denotes the prediction of the relational instance is $East$.

3. Logical Explanations using Feature-Clauses

We introduce first a more general notion of a logical explanation.

Definition 6 Logical Explanation. *We assume a set of relational instances $\mathcal{X}$, and a finite set of classes $\mathcal{Y}$. Given a relational instance $a \in \mathcal{X}$ and a label $c \in \mathcal{Y}$, the statement $Class(a, c)$ will denote that the class of a is c. Let $e = Class(a, c)$. Then given background knowledge B, a logical explanation for e is a set of clauses H s.t. $B \cup H \models e$.*

An explanation using feature-clauses is a special case of a logical explanation.

Example 2 An explanation in the trains problem. *One logical explanation $H = \{C\}$ for the first train in the left column of in Example 1 is the feature-clause:*

$$C : Class(t1, East) \leftarrow F_1(t1), F_2(t1), F_3(t1)$$

where B contains:

$$\forall x(F_1(x) \leftarrow \exists y(Has_Car(x, y), Short(y)))$$
$$\forall x(F_2(x) \leftarrow \exists y(Has_Car(x, y), Closed(y)))$$
$$\forall x(F_3(x) \leftarrow \exists y(Has_Car(x, y), Long(y)))$$

Here $t1$ is used as short-form for a structured term, describing the train, and, as before, let us assume that appropriate definitions exist in B for predicates like Has_Car, $Long$, $Short$, and $Closed$ to succeed (or fail) on terms like $t1$.

We will now restrict attention to explanations consisting of feature-clauses. It is useful for this paper to consider such explanations in two categories: those consisting of a single feature-clause, and those consisting of multiple feature-clauses.

3.1. Single-Clause Explanations

Assume we have a set of d features $\mathcal{F}$ with corresponding feature-definitions in the background knowledge B. We will first consider the case where explanation H consists of a single feature-clause. That is, $H = \{C\}$, where C is a definite clause $\forall x\ (Class(x, c) \leftarrow Body)$. Here $Body$ is a conjunction of $F_i/1$ literals, each of which is defined in B. The ILP practitioner will recognise that finding a single feature-clause explanation for $Class(a, c)$ is an instance of the "Single example clause, single hypothesis clause" situation identified in Muggleton (1994), which forms the basis of Explanation-Based Learning (EBL – see Mitchell et al., 1986).

One explanation for $e = Class(a, c)$ can be obtained immediately as follows. Consider the set $F_a = \{F_i : F_i \in \mathcal{F}$ s.t. $FV_i(a) = 1\}$ (let us call these the set of *active* features for a). By definition, if $FV_i(a) = 1$ then $F_i(a) \mapsto TRUE$. For simplicity, let us suppose that $F_a = \{F_1, \ldots F_k\}$ is the set of active features for a, and let $C : Class(a, c) \leftarrow F_1(a), F_2(a), \ldots, F_k(a)$. Assuming the F_i are defined in B, $H = \{C\}$ is an explanation for e.[4] Algorithm 1 is a simple procedure to construct a single-clause explanation C' that relies on the result that if $C' \preceq_\theta C$, then $C' \models C$. Therefore, if $B \cup \{C\} \models e$, then evidently $B \cup \{C'\} \models e$. That is, $\{C'\}$ will also be an explanation for e. For reasons that will be apparent, we will call $\{C'\}$ an "unstructured" explanation.[5]

4. ILP practitioners will recognise C as being analogous to the most-specific clause in Muggleton (1995), and we will call it the most-specific feature-clause for e, given $\mathcal{F}$ and B.

Algorithm 1: A non-deterministic procedure for identifying a single-clause unstructured explanation.

1 **Algorithm** `ConstructUnstruct`$(e, B, \mathcal{F})$
  Input: A relational instance e; background knowledge B; and a set of features $\mathcal{F}$ with definitions in B.
  Output: A single feature-clause explanation H s.t. $B \cup H \models e$.
2 Let $e = Class(a, c)$
3 Let $a' = FV(a)$
4 Let $\mathcal{F}'$ be the set of features that map to $TRUE$ in a'
5 Let $\mathcal{H}$ be the subset-lattice of $\mathcal{F}'$
6 Let $\mathcal{F}''$ be any non-empty subset in $\mathcal{H}$
7 Let $C' = \forall x(Class(x, c) \leftarrow Body)$
8 where $Body$ is the conjunction of features $F_i(x)$ for $F_i \in \mathcal{F}''$,
9 **return** $\{C'\}$

Example 3 *ConstructUnstruct* **for the trains problem.** *Let the background knowledge B contain the relational features:*

$$\forall x(F_1(x) \leftarrow \exists y(Has_Car(x, y), Short(y)))$$
$$\forall x(F_2(x) \leftarrow \exists y(Has_Car(x, y), Closed(y)))$$
$$\forall x(F_3(x) \leftarrow \exists y(Has_Car(x, y), Long(y)))$$
$$\forall x(F_4 x) \leftarrow \exists y(Has_Car(x, y), Long(y), Closed(y)))$$

Then, for $ConstructUnstruct(Class(t1, East), B, \{F_1, F_2, F_3, F_4\})$ (where $t1$ denotes the first train on the left in Example 1):

1. *$a' = [1, 1, 1, 0]$;*
2. *$\mathcal{F}' = \{F_1, F_2, F_3\}$;*
3. *$\mathcal{F}''$ can be any non-empty subset of $\{F_1, F_2, F_3\}$ (for example, $\mathcal{F}'' = \{F_1, F_3\}$); and*
4. *C' is the clause $\forall x(Class(x, East) \leftarrow Body)$ where Body is the conjunction of literals $F_i(x)$ where $F_i \in \mathcal{F}''$ (for example, $C' = \forall x(Class(x, East) \leftarrow F_1(x), F_3(x))$)*

In practice, we will need some way of searching the space of subsets of $\mathcal{F}'$. A fidelity-measure introduced later allows us to compare unstructured explanations, and forms the basis for guiding a search strategy.

5. In practice, in Algorithm 1 the lattice in Step 5 would be represented by a graph, and finding an element of the lattice in Step 6 will involve some form of optimal graph-search to find an optimal (or near-optimal) solution. More on this later.

3.2. Multi-Clause Explanations

We now extend explanations to a restricted form of *multi-clause explanation*, able to use features not all of which are defined in background knowledge B. These "invented" features will be required to be (re-)expressible in terms of features already defined in B. In this paper we will require structured and unstructured explanations to be related by the logic-program transformation operations of *folding* and *unfolding*. That is, given an unstructured explanation H and a structured explanation H_1, H can be derived from H_1 using one or more unfolding transformations; and H_1 can be derived from H using one or more folding transformations. We describe the transformations, along with the conditions that ensure that the computable answers do not change when the transformation is applied.

Definition 7 (One-Step) Unfolding of a clause (Hogger, 1990). *Given a set of definite clauses P, w.l.o.g. let $C \in P$ s.t. $C = Head \leftarrow L_1, \ldots, L_i, \ldots, L_k$ $(k \geq 1)$. Let $C' \in P$ be a clause $L'_i \leftarrow Body'$ s.t. L_i and L'_i unify with m.g.u. θ. Then the (one-step) unfolding of C w.r.t. L_i using P is the clause $Unfold(C) : (Head \leftarrow L_1, \ldots, L_{i-1}, Body', L_{i+1}, \ldots, L_k)\theta$*

Definition 8 (One-Step) Folding of a clause (Hogger, 1990). *Given a set of definite clauses P, w.l.o.g. let $C \in P$ s.t. $C = Head \leftarrow L_1, \ldots, L_{i-1}, Body_i, L_{i+1}, \ldots, L_k$, where $Body_i$ is some literals $B_1, B_2, \ldots, B_j$ $(j, k \geq 1)$. Let C' be a clause $L'_i \leftarrow Body'$ s.t. there is a substitution θ that satisfies: (a) $Body_i = Body'_i\theta$; (b) Every existentially quantified variable y in C, $y\theta$ is a variable that occurs only in $Body_i$ and nowhere else; (c) For any pair of distinct existential variables y, z in C $y\theta \neq z\theta$; and (d) C' is the only clause in P whose positive literal unifies with $L'_i\theta$. Then the (one-step) folding of C w.r.t. $Body_i$ using P is the clause $Fold(C) : (Head \leftarrow L_1, \ldots, L_{i-1}, L'_i\theta, L_{i+1}, \ldots, L_k)$.*

Remark 1 Correctness of Transformations. *In Petterossi and Proietti (1998) the unfold and fold transformations defined above are shown to be correct as replacement rules w.r.t. the minimal-model (MM) semantics. That is, if P is a definite-clause program and $C \in P$, then $MM(P) = MM((P - \{C\}) \cup Trans(C))$, where $Trans(C)$ is $Fold(C)$ or $Unfold(C)$.*

We can use this to construct structured explanations that are correct, in the computational sense just identified. That is, a structured explanation can replace an unstructured explanation without altering the minimal model of the (program containing) the unstructured explanation. It will be convenient to define the notion of an "invented" feature.

Definition 9 Invented Feature. *Given background knowledge B, a feature F is said to be an invented feature if: (1) F is not defined in B; and (2) The definition of F is a feature-clause C whose body contains features in B only; or is a clause that unfolds to a feature-clause C containing features only in B. We will sometimes denote C as $UFC(F)$ (short for "unfolded feature-clause for F").*

That is, $UFC(F)$ unfolds the definition of F only to the extent that the result is a feature-clause (that is, all literals in the definition are features defined in the background knowledge).

Example 4 Invented features in the trains problem. *The features $F_{1,1}$ and $F_{1,2}$ below are invented features:*

$F_{1,1}(x) \leftarrow F_2(x), F_{1,2}(x)$
$F_{1,2}(x) \leftarrow F_4(x), F_9(x)$

where the F_i are features defined in background knowledge:

$F_2(x) \leftarrow Has_Car(x, y), Short(y)$
$F_4(x) \leftarrow Has_Car(x, y), Wheels(y, 3)$
$F_9(x) \leftarrow Has_Car(x, y), Load(y, triangle)$

Then $UFC(F_{1,1})$ is:

$F_{1,1,}(x) \leftarrow F_2(x), F_4(x), F_9(x)$

and $UFC(F_{1,2})$ is identical to the definition of $F_{1,2}$.

We distinguish between unstructured and structured definitions based on the presence or absence of invented features.

Definition 10 Unstructured and structured explanations. *Let $a \in \mathcal{X}$ and $\mathcal{Y}$ be a set of class labels, with $c \in \mathcal{Y}$. Let $N : \Re^d \rightarrow \mathcal{Y}$ be a predictive model such that $N(FV(a)) = c$. Given background knowledge B, let H be an explanation for $Class(a, c)$ containing the feature-clause $C : \forall x(Class(x, c) \leftarrow Body)$. Let $Body$ consist of features $\mathcal{F}_C$. Let $\mathcal{F}_C$ be partitioned into: (a) $\mathcal{F}_{C,old}$, consisting of features defined in B; and (b) $\mathcal{F}_{C,new}$, consisting of features defined in $H - B$.*

We will call H an unstructured explanation *iff: (1) $\mathcal{F}_C$ contains only old features (that is, $\mathcal{F}_{C,new} = \emptyset$); and (2) $H = \{C\}$.*

We will call H a structured explanation *iff: (1) $\mathcal{F}_C$ contains only invented features (that is, $\mathcal{F}_{C,old} = \emptyset$); and (2) $H = \{C\} \cup InvF$, where $InvF$ only contains clauses defining invented features in $\mathcal{F}_{C,new}$; (3) For each feature F in $\mathcal{F}_{C,new}$ there is a single clause definition in $InvF$ s.t. F unfolds to a unique feature-clause defined using features in B only; and (4) At least one $F \in \mathcal{F}_{C,new}$ unfolds to a feature-clause that contains at least 2 features from B (that is, there is at least one invented feature that is not a trivial rewrite of features in B).*[6]

That is, a structured explanation is a set of clauses containing a classification clause C along with definitions of invented features (and thus is a case of the "single example clause, multiple hypothesis clauses" situation in Muggleton (1994)).

Example 5 Unstructured and structured explanations for the trains problem. *Suppose we are given an instance $a \in \mathcal{X}$, and a set of features $\mathcal{F}$, each defined by feature-definitions in background B. Then we will call the following an unstructured explanation (x is universally quantified):*

$$C : Class(x, East) \leftarrow F_2(x), F_3(x), F_4(x), F_9(x)$$

6. If required, for a structured explanation containing a clause C with a $F \in \mathcal{F}_{C,new}$ defined by a feature-clause whose clause body contains exactly one feature $F' \in \mathcal{F}_{C,old}$, a more compact "semi-structured" explanation can be obtained by a one-step folding. See the example that follows.

The following is a structured explanation H_1:

$$Class(x, East) \leftarrow F_{1,1}(x), F_{1,2}(x)$$
$$F_{1,1}(x) \leftarrow F_2(x), F_3(x)$$
$$F_{1,2}(x) \leftarrow F_4(x), F_9(x)$$

where, as usual, the features $F_{2,3,4,9}$ are defined as background knowledge:

$$F_2(x) \leftarrow Has_Car(x, y), Short(y)$$
$$F_3(x) \leftarrow Has_Car(x, y), Closed(y)$$
$$F_4(x) \leftarrow Has_Car(x, y), Wheels(y, 3)$$
$$F_9(x) \leftarrow Has_Car(x, y), Load(y, triangle)$$

Note that H_1 unfolds to H. Another structured explanation that also unfolds to H is H_2 below:

$$Class(x, East) \leftarrow F_{1,1}(x), F_{1,2}(x)$$
$$F_{1,1}(x) \leftarrow F_1(x), F_4(x)$$
$$F_{1,2}(x) \leftarrow F_3(x), F_9(x)$$

We will assume that explanations of the form $Class(x, East) \leftarrow F_{1,1}(x), F_3(x), F_9(x)$ will be represented as H_2. Also, by definition, the following is not a structured explanation, since all new features are trivial rewrites of existing ones:

$$Class(x, East) \leftarrow F_{1,1}(x), F_{1,2}(x)$$
$$F_{1,1}(x) \leftarrow F_1(x)$$
$$F_{1,2}(x) \leftarrow F_3(x)$$

But this is a structured explanation:

$$Class(x, East) \leftarrow F_{1,1}(x), F_{1,2}(x)$$
$$F_{1,1}(x) \leftarrow F_1(x)$$
$$F_{1,2}(x) \leftarrow F_3(x), F_9(x)$$

It is obvious enough that every structured explanation allows the derivation by unfolding of a correct unstructured explanation.

Remark 2 Deriving an Unstructured Explanation from a Structured Explanation. *Let $H = \{C'\} \cup InvF$ be a structured explanation for a relational example $Class(a, c)$ given background knowledge B. W.l.o.g. from Defn. 10, C' is a clause $Class(x, c) \leftarrow \{F'_1(x), F'_2(x), \ldots, F'_k(x)\}$ s.t. each F'_i unfolds using $InvF$ to a unique feature-clause $F'_i(x) \leftarrow F_{i,1}(x), \ldots, F_{i,n_i}(x)$, where the $F_{i,\cdot}$ are defined in B. From Defn.7, unfolding C' w.r.t. the F'_i using $InvF$ results in the clause $C : Class(x, c) \leftarrow F_{1,1}, \ldots, F_{k,n_k}$. From Remark 1, the minimal model of $B \cup H$ will be unchanged by replacing C' with C. It follows that $H = \{C\}$ is an unstructured explanation for $Class(a, c)$ that is computationally equivalent to H'. Here "computationally equivalent" means that the structured explanation computes the same set of answers under the minimal model semantics as the unstructured explanation (see Hogger, 1990).*

One or more correct structured explanations follow from an unstructured explanation if the conditions in Defn. 8 hold. Constraints on the invented features can ensure this.

Remark 3 Deriving Structured Explanations from an Unstructured Explanation. *Let $H = \{C\}$ be an unstructured explanation for a relational example $Class(a, c)$ given background knowledge B. W.l.o.g. from Defn. 10, C is a clause $Class(x, c) \leftarrow \{F_1(x), F_2(x), \ldots, F_n(x)$ s.t. each F_i is defined in B. Let $InvF$ consist of k features $F'_1, \ldots, F'_k$ s.t. each F'_i is uniquely defined by features in a block of a k-partition of $F_1, \ldots, F_n$. Let C' be the clause $Class(x, c) \leftarrow F'_1(x), F'_2(x), \ldots, F'_k(x)$. Then $H' = \{C'\} \cup InvF$ is a structured explanation for $Class(a, c)$ that is computationally equivalent to $B \cup \{C\} \cup InvF$. This follows since the conditions (a)–(d) in Defn. 8 are trivially satisfied (condition (a) follows with θ being a simple renaming substitution; (b)–(c) follow since there are no existential variables in the definitions of the F'_i; and (d) since there is a single clause definition for each feature in $InvF$).*

Remark 3 suggests a straightforward non-deterministic procedure to construct a correct structured explanation (Algorithm 2).

Algorithm 2: A non-deterministic procedure for obtaining a structured explanation from an unstructured explanation, by inventing k features. The condition in Step 5 is not required by Remark 3, but prevents inventing features that are trivial rewrites of existing features (see Defn. 10).

1 **Algorithm** `ConstructStruct`(H, k)
  Input: An unstructured explanation H; and a number k (≥ 2).
  Output: A structured explanation H' that is *computationally equivalent* to H.
2 Let $H = \forall x(Class(x, c) \leftarrow Body)$.
3 Let F be the set of features in $Body$.
4 Let P_k be a set s.t. one of the following is true:
5 Either (a) P_k is a k-partition of F s.t. at least one block in P_k contains 2 or more elements;
6 Or (b) $P_k = \emptyset$ (if no such k-partition exists).
7 **if** $P_k = \emptyset$ **then**
8 **return** $\emptyset$
9 **end**
10 **else**
11 Let P_k consist of blocks $b_1, b_2, \ldots, b_k$.
12 **for** *each block* $b_i \in P_k$ **do**
13 Construct a new feature F'_i with definition $C_i = \forall x(F'_i(x) \leftarrow Body_i)$, where $Body_i$ is the conjunction of the features in b_i.
14 Let $C_0 = \forall x(Class(x, c) \leftarrow Body_0)$ where $Body_0$ is the conjunction of the $F'_1, F'_2, \ldots, F'_k$.
15 Let $H' = \bigcup_{i=0}^{k} C_k$.
16 **end**
17 **end**
18 **return** H'

Example 6 *ConstructStruct* **for the trains problem.** *Let H be the clause $\forall x(Class(x, East) \leftarrow F_2(x), F_3(x)F_4(x), F_9(x))$ where $F_{2,3,4,9}$ are defined in the background knowledge B. Then for $ConstructStruct(H, 2)$:*

1. *P_2 can be any 2-partition of $\{F_2, F_3, F_4, F_9\}$ s.t. at least one block of P_2 contains more than 1 element. (for example, $P_2 = \{[F_2, F_4], [F_3, F_9]\}$)*

2. *Feature-clauses C_1 and C_2 are are constructed using the set of features in each block (for example, $C_1 = F_1'(x) \leftarrow F_2(x), F_4(x)$ and $C_2 = F_2'(x) \leftarrow F_3(x), F_9(x)$)*

3. *A classification clause C_0 is constructed using the new feature-definitions (for example, $C_0 = \forall x(Class(x, East) \leftarrow F_1'(x), F_2'(x))$*

4. *A structured explanation $H' = \{C_0, C_1, C_2\}$ is returned*

In practice, we will need some way of searching the space of 2-partitions (in general, k-partitions). A partial-ordering introduced later allows us to compare structured explanations, and forms the basis for guiding a search strategy.

All explanations so far have only referred to a single relational instance. What is to be done when we seek explanations for more than one instance? It suffices to consider the case of 2 instances e_1 and e_2. Then, an explanation H for both instances is simply an explanation for $e_1 \wedge e_2$ (that is, $B \cup H \models (e_1 \wedge e_2)$). We use this to formulate a notion of "locally consistent" explanations.

3.3. Locally Consistent Explanations

Recent research called LIME ("Local Interpretable Model-agnostic Explanations" – see Ribeiro et al., 2016) proposes producing readable proxies "on-demand", for any kind of black-box predictor. The key feature is that a LIME-style explanatory model is constructed only when a prediction for an instance is sought, and the explanation is required to be *faithful* to the black-box's predictions for the instance and its near-neighbours. The principal intuition underlying LIME is this: a single comprehensible model may not be possible for the black-box's predictions over all instances but, nevertheless, it may be possible to construct multiple readable proxies, each of which is consistent with the predictions for groups of similar ("local") instances.

However, the requirement of having to be consistent with only local instances results in a different kind of problem. Since the set of near-neighbours of an instance may be quite small, there may be insufficient constraints, in data-theoretic terms, to narrow down a unique (or even a small number of) explanations. So, how then is an explanation to be selected? In LIME, this is left to the loss-function. In Bayesian terms, this means defining an appropriate prior to guide selection when the data are insufficient.

We draw on specific situations described in Muggleton (1994) to clarify what is meant by a local explanation, and its data-theoretic evaluation. Later we will introduce a domain-dependent prior to develop a Bayes-like selection of explanations.

The explanations we seek are for predictions of relational instances that are "close" to each other. Readers familiar with the ILP literature will recognise that the task of finding such an explanation corresponds to either the "Multiple examples, single hypothesis clause" or the "Multiple examples, multiple hypothesis clauses" situations in Muggleton (1994), depending on whether unstructured or structured explanations are constructed. We note first that if a logical explanation is extracted for the full DRM in the manner done

by França et al. (2015), then alternate explanations for an instance $e = Class(a, c)$ can be provided in terms of the rules that can be used to derive e. Each such explanation would be an unstructured explanation in the terminology of this paper. This does not alter the notion that follows of finding the best local explanation.

We have seen how to construct the most-specific feature-clause for $e = Class(a, c)$ from the set of active features for a. We now seek explanations that are not only consistent with a predictive model on an instance x, but also are consistent with predictions made by the predictive model in a *local neighbourhood* of x.

Definition 11 Neighbourhood. *Given relational instance $a \in \mathcal{X}$, and a set of d features $\mathcal{F}$, and given some $\epsilon \in \Re$, we denote the neighbourhood of a as $Nbd(a) = \{x : x \in \mathcal{X}$ and $Dist(FV(a), FV(x)) \leq \epsilon\}$. Here Dist is some appropriate distance measure defined over d-dimensional vectors.*

In practice the dimensionality d can be quite large, and since the standard Euclidean distance is known to be problematic in high-dimensional spaces (Aggarwal et al., 2001) we will use an alternative measure.

Definition 12 Locally Consistent Explanations. *Given relational instance $a \in \mathcal{X}$, and a predictive model N, let $N(FV(a)) = c$. We define the following subsets of $Nbd(a)$: $E^+(a) = \{b : b \in Nbd(a)$ and $N(FV(b)) = c)\}$ and $E^-(a) = \{b : b \in Nbd(a)$ and $N(FV(b)) \neq c\}$. Then an explanation H for $Class(a, c)$ given B is a locally consistent explanation if: (1) for each $a' \in E^+(a)$ H is an explanation for $Class(a', c)$ (that is, $B \cup H \models Class(a', c)$); and (2) for each $a' \in E^-(a)$ $B \wedge H \wedge \neg Class(a', c) \not\models \Box$.*

Example 7 Locally consistent explanation for the trains problem. *For simplicity, we consider the situation where $E^- = \emptyset$. Suppose we now know that the local neighbourhood of the first train (t1) in the left column of Example 1 only contains the second train (t2) in that column. Let us assume that $\mathcal{F}$ consists just of the functions defined in Example 2. With those definitions, and a predictive model N, let $N(FV(t1)) = N(FV(t2))$ (= East, say). Then $E^+(t1) = \{t1, t2\}$.*

The most-specific feature-clause for $Class(t1, East)$ given $\mathcal{F}$ and B is:

$$C1 : Class(t1, East) \leftarrow F_1(t1), F_2(t1), F_3(t1)$$

and for $Class(t2, East)$ is:

$$C2 : Class(t2, East) \leftarrow F_1(t2), F_2(t2)$$

where the function definitions are as before. Then a locally consistent explanation for $Class(t1, East)$ is the least-general-generalisation (or Lgg) of $C1$ and $C2$:

$$Lgg(C1, C2) : \forall x(Class(x, East) \leftarrow F_1(x), F_2(x))$$

In general, $E^- \neq \emptyset$, and results from ILP tell us that it may not be possible to find a single clause that is locally consistent. We next describe a simple, qualitative form of Bayes Rule that combines likelihood of the data, with a relevance-based prior preference over explanations.

It is useful before we proceed further to have a numerical measure of the extent to which an explanation is locally consistent.

Definition 13 Fidelity. *Let $a \in \mathcal{X}$. Given a predictive model N, let $N(FV(a)) = c$ and $E^+(a)$, $E^-(a)$ as before. Let $D = (E^+, E^-)$ and H be an explanation for $Class(a,c)$ given background knowledge B. Let: (1) $AgreePos(H) = \{b : b \in E^+(a)$ and $B \wedge H \models Class(b,c)\}$; and (2) $AgreeNeg(H) = \{b : b \in E^-(a)$ and $B \cup H \wedge \neg Class(b,c) \not\models \Box\}$. Then $Fidelity(H|D,B) = F_{H,a} = \frac{|AgreePos(H)|+|AgreeNeg(H)|}{|E^+(a)|+|E^-(a)|}$.*

We note that $AgreePos(H)$ and $AgreeNeg(H)$ are the same as true positives and true negatives in the classification literature, and so *fidelity* is a localized form of accuracy (in Ribeiro et al. (2016) the term "local fidelity" is used for a localized form of error). If the predictor allows a probabilistic prediction, then a finer-grained weighted fidelity is computable.

Remark 4 Structuring preserves fidelity. *Let $H' = \{C'\} \cup InvF$ be a structured explanation derived as in Remark 3 from an unstructured explanation $H = \{C\}$ for a relational example $Class(a,c)$, given definite clauses B and a predictor N s.t. $N(FV(a)) = c$. Let $E^+(a)$ and $E^-(a)$ denote relational examples obtained from the neighbourhood of a as before, and $D = (E^+(a), E^-(a))$. Then, $Fidelity(H'|D,B) = Fidelity(H|D,B)$.*

Let S be any finite set of ground $Class/2$ facts and $MM(P)$ be the minimal model of a definite clause program P. Since $InvF$ is a set of definite clauses containing only definitions of invented features, $S \cap MM(B \cup \{C\}) = S \cap MM(B \cup \{C\} \cup InvF)$. Now:

$$AgreePos(H) = \{Class(a',c) : Class(a',c) \in (E^+(a) \cap MM(B \cup H)\}$$
$$AgreeNeg(H) = \{Class(a',c) : Class(a',c) \in (E^-(a) - (E^-(a) \cap MM(B \cup H)\}.$$

Since $E^+(a)$ and $E^-(a)$ are finite sets of ground $Class/2$ facts, and $H = \{C\}$:

$$AgreePos(H) = \{Class(a',c) : Class(a',c) \in (E^+(a) \cap MM(B \cup \{C\} \cup InvF)\}$$
$$AgreeNeg(H) = \{Class(a',c) : Class(a',c) \in (E^-(a) - (E^-(a) \cap MM(B \cup \{C\} \cup InvF)\}.$$

From Remark 3, $MM(B \cup \{C\} \cup InvF) = MM(B \cup \{C'\} \cup InvF)$. Since $H' = \{C\} \cup InvF$, it follows immediately that $AgreePos(H) = AgreePos(H')$ and $AgreeNeg(H) = AgreeNeg(H')$ and therefore $Fidelity(H|D,B) = Fidelity(H'|D,B)$. $\Box$

Example 8 (Fidelity-preserving structuring for the trains problem). *The following is a structured explanation H_1:*

$$Class(x, East) \leftarrow F_{1,1}(x), F_{1,2}(x)$$
$$F_{1,1}(x) \leftarrow F_2(x), F_3(x)$$
$$F_{1,2}(x) \leftarrow F_4(x), F_9(x)$$

Since there are no new existentially quantified variables in feature-clauses, and there is a single clause defining each of $F_{1,1}$ and $F_{1,2}$, H_1 will unfold uniquely (see Gabbay et al. (1998)) to the unstructured explanation H:

$$Class(x, East) \leftarrow F_2(x), F_3(x), F_4(x), F_9(x)$$

That is, the minimal-models of H and H_1 will contain the same ground instances of $Class(x, East)$ (Hogger (1990)). Given the data D and background knowledge B it follows that the $AgreePos(H) = AgreePos(H_1)$ and $AgreeNeg(H) = AgreeNeg(H_1)$. Consequently, $Fidelity(H|D,B) = Fidelity(H_1|D,B)$.

From now on, we will take the black-box predictor to be a DRM, although what follows continues to apply to any black-box predictor that uses relational feature values as inputs. In the following section we address selecting a local explanation, but first we draw attention to some general issues for local fidelity.

Remark 5 Determination of local fidelity. *The fidelity measure can be seen, roughly, as a degree of equivalence between two theories, one predictive and the other explanatory, taken as an average of the local fidelities over a set of neighbourhoods of instances. However, this tells us nothing about a strategy to sample the* instances to be explained. *This must be based on the* objectives *for the explanation, which may vary considerably between applications. One strategy in LIME (Ribeiro et al., 2016), for example, is to select, or "pick" test instances that are diverse yet cover well the space of representable features. Since a human user is assumed to have only finite time to inspect explanations, a fixed number of instances is generated using a weighted coverage optimisation algorithm, maximising the use of important features (according to feature weights in the explanatory models) while minimising redundancy between instances. In Section 5.2 of this paper, however, a different strategy is adopted, due to differing assumptions regarding users.*

4. Selecting a Local Explanation

Given a relational instance $a \in \mathcal{X}$, let the prediction of a by a DRM N be c. But, for a definition of a neighbourhood, there may be several explanations for $Class(a,c)$ with the same maximal fidelity. How then should we select a single explanation? Provided we have some reasonable way of specifying prior-preferences, Bayes rule trades-off the fit to the data (likelihood) against prior preference (in Ribeiro et al. (2016), LIME's minimisation of the sum of a loss and a regularisation term can be seen as implementing a form of Bayesian selection).

A general approach for selecting amongst logical formulae is provided by labelled deductive systems (LDS – see Gabbay, 1996), in which logical formulae are extended with labels and an associated algebra. For our purposes, it is sufficient simply to consider labelled explanations.

Definition 14 Labelled Explanation. *Given a relational example $e = Class(a,c)$, and background knowledge B, $\alpha : H$ is a labelled explanation for e given B if: (a) H is an explanation for e given B; and (b) α is a element of a partially-ordered set of ground first-order terms Δ. Δ consists of annotations of all explanations for e given B.*

A comparison of labelled explanations follows simply from the partial ordering on the labels. That is, $\alpha : H_1 \preceq \beta : H_2$ iff $\alpha \preceq \beta$. Here we will take the label of an explanation H to be a pair $\langle L_H, P_H \rangle$, which allows several different kinds of comparisons, given background knowledge B and data D:

Quantitative. This is appropriate when both L_H and P_H are on an interval scale (that is, numeric values). Examples are: (a) the usual Bayesian comparison, using $L_H = P(D|H,B)$ and $P_H = P(H|B)$; and $\langle L_{H_1}, P_{H_1} \rangle : H_1 \preceq \langle L_{H_2}, P_{H_2} \rangle : H_2$ iff $\log L_{H_1} + \log P_{H_1} \leq \log L_{H_2} + \log P_{H_2}$; (b) Good's explicativity (Chapter 23 of Good (1983)),

which uses the same L_H, P_H, but uses the function $\log L_H + \gamma \log P_H$ with $0 < \gamma < 1$; and (c) Likelihood-based, using $L_H = P(D|H,B)$ and P_H is the uniform distribution.

Qualitative. Here both L_H and P_H are both on an ordinal scale (that is, only comparisons of values are possible). Examples are: (a) The qualitative Bayesian comparison in the manner proposed by Coletti and Scozzafava (1993). With some abuse of notation, $\langle L_{H_1}, P_{H_1}\rangle : H_1 \preceq \langle L_{H_2}, P_{H_2}\rangle : H_2$ iff $L_{H_1} \preceq L_{H_2}$ and $P_{H_1} \preceq P_{H_2}$. If it is not the case that $\langle L_{H_1}, P_{H_1}\rangle : H_1 \preceq \langle L_{H_2}, P_{H_2}\rangle : H_2$ or $\langle L_{H_2}, P_{H_2}\rangle : H_2 \preceq \langle L_{H_1}, P_{H_1}\rangle : H_1$, then the labelled explanations are not comparable; and (b) A dictionary-ordering, in which $\langle L_{H_1}, P_{H_1}\rangle : H_1 \preceq \langle L_{H_2}, P_{H_2}\rangle : H_2$ iff $L_{H_1} \prec L_{H_2}$, or $L_{H_1} = L_{H_2}$ and $P_{H_1} \preceq P_{H_2}$.[7]

Semi-Quantitative. Here, one of L_H or P_H is on an interval scale, and the other is on an ordinal scale. The qualitative comparisons above can be adapted to this, by replacing $\preceq$ and $\prec$ with $\leq$ and $<$ for the numeric quantity.

We will use the semi-quantitative dictionary ordering with $L_H = Fidelity(H|D,B)$ in this paper, and P_H is an ordinal-valued prior based on an assessment of relevance. Using the semi-quantitative setting and the dictionary ordering has some advantages:

(a) Quantitative selection based on a Bayesian score requires a definition of both $P(D|H,B)$ and $P(H|B)$. While the first can be obtained easily enough, it is not obvious how to specify a prior distribution over explanations. The usual approach of using a mathematically convenient function like $2^{-|H|}$, where $|H|$ is some measure of the size of H, may not be an appropriate translation of prior assessment of relevance of explanations; and

(b) A qualitative Bayesian approach as defined above usually ends up with many incomparable explanations. The dictionary ordering decomposes the task of identifying explanations into two parts: the first part that maximises fidelity, and the second part that maximises the prior amongst maximal fidelity explanations. Under some circumstances (see the Appendix), maximising fidelity is equivalent to maximising log likelihood. In those cases, the dictionary ordering uses the prior to select amongst maximum likelihood explanations. Usefully, there is also an implementation benefit that follows from the result in Remark 4: since structuring does not alter the fidelity, the first part can simply examine unstructured explanations.

We turn now to prior information P_H that captures some aspects of what constitutes a comprehensible explanation.

4.1. A Relevance-Based Prior

In Srinivasan et al. (2003) the authors investigate the utility of including an expert assessment of the relevance of relations included in the background knowledge.

7. This may not yield the same results as the qualitative Bayesian comparison above. Differences arise when $L_{H_1} \prec L_{H_2}$ but $P_{H_2} \preceq P_{H_1}$. Under the dictionary-ordering H_2 would be preferred, but the qualitative Bayesian approach would find H_1 and H_2 incomparable.

Example 9 Relevance information in the trains problem. *Suppose the background knowledge B for the trains problem contains definitions of predicates like $Has_Car/2$, $Short/1$, $Closed/1$, $Wheels/2$ and $Load/2$. For the problem of classifying trains as east-bound or west-bound, let us assume we are also given domain knowledge in the form of the relevance level of (sets) of predicates as follows: $r_1 : \{Wheels/2, Load/2\}, r_2 : \{Short/1, Closed/1\}$, and $r_1 \prec_r r_2$. That is, Wheels and Load are less relevant to the problem than Short and Closed.*

We note that this is different to the notion of logical relevance of features (defined in terms of the entailment relation, $\models$). Here, we are concerned with domain-relevance of predicates used to define those features. This latter form of domain-specific relevance information can also form the basis of a preference ordering over explanations. Here is the view of the domain expert involved in Srinivasan et al. (2003):[8]

> *I think it is reasonable to argue that [a hypothesis using] more relevant prior background knowledge is more probable. I think that what makes hypotheses more probable is also a function of whether the predicates used are related to each other. Often you see hypotheses that seem to mix apples and oranges together, which seems to make little sense. Though of course this mixing of predicates may be because the ML system is trying to express something not easily expressible with the given background predicates.*

This suggests that relevance information can constitute an important source of prior knowledge. One route by which this is exploited by ILP systems is in the form of search constraints ("hypotheses that do not contain oxygens connected to hetero-aromatic rings are irrelevant"), or, as in the case of Srinivasan et al. (2003), in the incremental construction of hypotheses. Our interest here is to extend this use of relevance to select amongst hypotheses, by devising a relevance-based partial ordering over hypotheses.[9]

Definition 15 Relevance-assignments and orderings. *We assume that for some set of predicates $\mathcal{P}$ in background knowledge B, we have domain-specific relevance labels drawn from a set $\mathcal{R}$. A relevance assignment is a function $R : \mathcal{P} \rightarrow \mathcal{R}$. We assume that there is domain-knowledge in the form of a total ordering $\prec_r$ over the elements of $\mathcal{R}$. Then, for $a, b \in \mathcal{R}$, $a \preceq_r b$ iff $a \prec_r b$ or $a = b$. We will call $\prec_r$ a relevance ordering.*

A relevance ordering naturally results in the concept of ordered intervals: $[a,b]$ is an *ordered relevance interval* (or simply a *relevance interval*) if $a, b \in \mathcal{R}$ and $a \preceq_r b$. The ordering $\prec_r$ gives rise in turn to an ordering over relevance intervals.

Definition 16 Ordering over relevance intervals. *Let $\mathcal{I}$ be the set of ordered relevance intervals. Let $[a,b]$ and $[c,d]$ be intervals in $\mathcal{I}$. Then $[a,b] \preceq_i [c,d]$ iff $a \preceq_r c$ and $b \preceq_r d$.*

Remark 6 ($\preceq_i$ is a partial ordering). *Let $\mathcal{I}$ be the set of ordered relevance intervals. Then: (a) Clearly for any interval $[a,b] \in \mathcal{I}$, $[a,b] \preceq_i [a,b]$; (b) For $[a,b] \in \mathcal{I}$ and $[c,d] \in \mathcal{I}$,*

8. R.D. King: personal communication
9. We will often reuse the generic symbols $\preceq$ and $\prec$ to denote partial- and total-orderings. The context will make it clear which sets these relations refer to.

let $[a,b] \preceq_i [c,d]$ and $[c,d] \preceq_i [a,b]$. Then $a \preceq_r c$ and $c \preceq_r a$ and $b \preceq_r d$ and $d \preceq_r b$. Since $\preceq_r$ is a partial order, it follows that $a = c$ and $b = d$; (c) For $[a,b] \in \mathcal{I}$, $[c,d] \in \mathcal{I}$ and $[e,f] \in \mathcal{I}$, let $[a,b] \preceq_i [c,d]$ and $[c,d] \preceq_i [e,f]$. From a similar argument to (b), it follows that $[a,b] \preceq_i [e,f]$. That is, $\preceq_i$ is reflexive, anti-symmetric and transitive.

We will use the following, slightly more general ordering, than $\preceq_i$:

Definition 17 (Ordering over sets of relevance-intervals) *Let S and S' be sets of relevance intervals. Then $S \preceq_s S'$ iff for every interval $[a,b]$ in S there exists at least one interval $[c,d]$ in S' s.t. $[a,b] \preceq_i [c,d]$. That is, $a \preceq_r c$ and $b \preceq_r d$.*

Remark 7 ($\preceq_s$ is a quasi-ordering). *Let $\mathcal{S}$ be the set whose elements are sets of relevance intervals and $\preceq_s$ as in Defn. 16. Then: (a) For $S \in \mathcal{S}$, clearly $S \preceq_s S$; and (b) Let $S_{1,2,3} \in \mathcal{S}$ s.t. $S_1 \preceq_s S_2$ and $S_2 \preceq_s S_3$. Then, from Defn. 16 for every interval $[a,b] \in S_1$ there is an interval $[c,d] \in S_2$ s.t. $[a,b] \preceq_i [c,d]$ and for every $[c,d] \in S_2$ there is an interval $[e,f] \in S_3$ s.t. $[c,d] \preceq_i [e,f]$. Since $\preceq_i$ is a partial order (Defn. 6), it follows that for every $[a,b] \in S_1$ there is an interval $[e,f] \in S_3$ s.t. $[a,b] \preceq_i [e,f]$. That is $S_1 \preceq_s S_3$. So, $\preceq_s$ is reflexive and transitive, and therefore a quasi-order. However, $\preceq_s$ is not anti-symmetric, as witnessed by $S_1 = \{[2,5],[3,6]\}$ and $S_2 = \{[3,6]\}$, with the usual ordering on numbers. Here $S_1 \preceq_s S_2$ and $S_2 \preceq_s S_1$, but $S_1 \neq S_2$.*

We now construct, in stages, the relevance of an explanation.

Definition 18 Relevance of features. *Given background knowledge B, let $\prec_r$ be a relevance ordering over a set of relevance labels $\mathcal{R}$, and let $R : \mathcal{P} \rightarrow \mathcal{R}$ be a relevance assignment for some subset $\mathcal{P}$ of B. Let $\forall x(F(x) \leftarrow Cp(x))$ be the feature-definition for $F/1$ in which $Cp(x)$ is a conjunction containing predicates from $\mathcal{P}$ only. Then the relevance of the feature F is $Relev(F) = [l,h]$, where l is the minimum relevance of predicates in $Cp(x)$ according to $\prec_r$ and R, and h is maximum relevance of predicates in $Cp(x)$ according to $\prec_r$ and R.*

Example 10 (Relevance of features for the trains problem). *Suppose we have the following feature-definitions (omitting quantifiers for simplicity):*

$F_2(x) \leftarrow Has_Car(x,y), Short(y)$
$F_3(x) \leftarrow Has_Car(x,y), Closed(y)$
$F_4(x) \leftarrow Has_Car(x,y), Wheels(y,3)$
$F_9(x) \leftarrow Has_Car(x,y), Load(y, triangle)$

Assume we are given a set of relevance labels $\mathcal{R} = \{r1, r2\}$, with $r1 \prec_r r2$. Let us further assume $\{(Wheels/1, r1), (Load/2, r1), (Short/1, r2), (Closed/1, r2)\}$ is our relevance-assignment. Then $Relev(F_2) = Relev(F_3) = [r2, r2]$, $Relev(F_4) = Relev(F_9) = [r1, r1]$.

Definition 19 Relevance of feature-clauses. *Let $\mathcal{F}$ be a set of features. Let C be a feature-clause. W.l.o.g. let the features in C be $\{F_1, F_2, \ldots, F_k\}$ where the $F_i \in \mathcal{F}$. Let $Relev(F_i) = [l_i, h_i]$. Then $Relev(C) = \{[l*, h*]\}$ where $l* = \min(l_1, l_2, \ldots, l_k)$ and $h* = \max(h_1, h_2, \ldots, h_k)$.*

Example 11 (Relevance of a feature-clause for the trains problem). *As before, let there be 4 relational features F_2, F_3, F_4, and F_9, with $Relev(F_2) = Relev(F_3) = [r2, r2]$, $Relev(F_4) = Relev(F_9) = [r1, r1]$. Then, the relevance of the feature-clause $C : F(x) \leftarrow F_2(x), F_3(x), F_4(x), F_9(x)$ is $R_C = \{[r1, r2]\}$.*

The relevance of explanations is constructed from the relevance of the feature-clauses in the explanation. We distinguish deliberately between unstructured and structured explanations.

Definition 20 Relevance of an explanation *Given background knowledge B, let H be an explanation for $Class(a,c)$ containing the clause $C : \forall x(Class(x,c) \leftarrow Body)$. If H is an unstructured explanation then $Relev(H) = Relev(C)$. Otherwise, if H is a structured explanation, then $Relev(H) = \bigcup_{F_i \in Body} Relev(UFC(F_i))$.*

From Defn. 20, the relevance of an explanation is a set of relevance intervals, and from Remark 7, it follows that set of explanations can be quasi-ordered, using the $\preceq_s$ relation over the relevance of explanations.

Example 12 (Relevance of explanations for the trains problem). *Suppose, as before, we have 4 relational features F_2, F_3, F_4, and F_9, with $Relev(F_2) = Relev(F_3) = [r2, r2]$, $Relev(F_4) = Relev(F_9) = [r1, r1]$. Suppose we have an unstructured explanation $H : Class(x, East) \leftarrow F_2(x), F_3(x), F_4(x), F_9(x)$. Then, from Example 11, $Relev_H = R_H = \{[r1, r2]\}$.*

On the other hand, suppose we have the structured explanation H_1:

$Class(x, East) \leftarrow F_{1,1}(x), F_{1,2}(x)$
$F_{1,1}(x) \leftarrow F_2(x), F_3(x)$
$F_{1,2}(x) \leftarrow F_4(x), F_9(x)$

$F_{1,1}, F_{1,2} \notin \mathcal{F}$, and are therefore invented features. Let $\mathcal{F}_2 = \{F_{1,1}, F_{1,2}\}$. Then, $Relev(H_1) = R_{H_1} = \bigcup_{F \in \mathcal{F}_2} Relev(UFC(F)) \cup \emptyset$. Now $Relev(UFC(F_{1,1})) = \{[r2, r2]\}$, $Relev(UFC(F_{1,2})) = \{[r1, r1]\}$ and $R_{H_1} = \{[r1, r1], [r2, r2]\}$.

It is interesting that although a pair of structured explanations may unfold to the same unstructured explanation (and therefore have the same fidelity), their relevance may not be the same. Intuitively, structuring any unstructured explanation will split the relevance-interval of the corresponding feature-clause into a set of intervals (see Defn. 20). In a "good structuring" each interval in this set will be narrower than the unstructured relevance-interval and will therefore be preferred under the relevance ordering (Defn. 17).

Remark 8 Structuring can increase relevance. *Let $H = \{C\}$ be an unstructured explanation for $Class(a,c)$, and let H' be a structured explanation containing a clause $C' : Class(x,c) \leftarrow Body$ that unfolds to C. Let $P_H = Relev(H)$ and $P_{H'} = Relev(H')$. Then $P_H \preceq_s P_{H'}$.*
Let C contain the features $\{F_1, \ldots, F_l\}$, where $Relev(F_i) = [l_i, h_i]$ Since H is an unstructured explanation, $Relev(H) = \{[l*, h*]\}$, where $l* = \min(l_1, \ldots, l_k)$ and $h* = \max(h_1, \ldots, h_k)$.

Let C' contain the invented features $\{F'_1, \ldots, F'_j\}$. Since C' unfolds to C, each F'_i unfolds to a clause containing some subset S'_i of $\{F_1, \ldots, F_k\}$, and $\bigcup_{i=1}^{j} S'_i = \{F_1, \ldots, F_k\}$. W.l.o.g. let $h = h_k$. By the constraint imposed on structured explanations, there must be at least one invented feature F'_m that unfolds to a clause containing F_k. Let $Relev(UFC(F'_m)) = [l'_m, h'_m]$. Clearly, $l* \preceq l'_m$ and $h* = h'_m$, and therefore $[l*, h*] \preceq [l'_m, h'_m]$. Since $[l'_m, h'_m] \in Relev(H')$, it follows from Defn. 17 that $Relev(H) \preceq_s Relev(H')$.*

Example 13 Comparing the relevance of explanations in the trains problem. *As before, suppose we have 4 relational features F_2, F_3, F_4, and F_9, with $Relev(F_2) = Relev(F_3) = [r2, r2]$, $Relev(F_4) = Relev(F_9) = [r1, r1]$. The unstructured explanation H: $Class(x, East) \leftarrow F_2(x), F_3(x), F_4(x), F_9(x)$ has relevance $R_H = \{[r1, r2]\}$.*

The following structured explanation H_1 unfolds to H:

$$Class(x, East) \leftarrow F_{1,1}(x), F_{1,2}(x)$$
$$F_{1,1}(x) \leftarrow F_2(x), F_3(x)$$
$$F_{1,2}(x) \leftarrow F_4(x), F_9(x)$$

As we saw in Example 12, $R_{H_1} = \{[r1, r1], [r2, r2]\}$. From Defn. 17, $R_H \preceq_s R_{H_1}$.

But structuring is not guaranteed to increase relevance. The following structured explanation H_2 also unfolds to H:

$$Class(x, East) \leftarrow F_{1,3}(x), F_{1,4}$$
$$F_{1,2}(x) \leftarrow F_3(x), F_4(x)$$
$$F_{1,4}(x) \leftarrow F_2(x), F_9(x)$$

Now $R_{H_2} = \{[r1, r2]\}$, and $R_H \not\preceq_s R_{H_2}$. Thus, although $H_{1,2}$ both unfold to H, a selection criterion that takes relevance into account would prefer H_1 over H_2 and H.

4.2. Other Relevance-Based Priors

Broadly speaking, priors can be categorised as theoretical (derived from domain knowledge), or empirical (derived from data). A relevance ordering over predicates represents just one form of a theoretical prior. Finer-grained theoretical priors based on relevance are possible. For example, some specific combination of predicates may be especially important (say, the presence of a lactone ring, connected to a 7-membered ring).

The machinery of comparing labelled explanations can be used without change if theoretically important features can be encoded as background predicates.

Example 14 (A theoretically important feature for the trains problem). *Let as assume we are given a set of relevance labels $\mathcal{R} = \{r1, r2, r*\}$, with $r1 \prec_r r2$ and $r2 \prec_r r*$. Assume as before, the following relevance-assignment: $\{(Wheels/1, r1), (Load/2, r1), (Short/1, r2), (Closed/1, r2)\}$. Suppose we know that the existence of short, closed cars is especially important. This can be encoded by the predicate in the background knowledge:*

$$Short_Closed(x) \leftarrow Has_Car(x, y), Short(y),\ Closed(y)$$

Augumenting the relevance-assignment with $(Short_Closed/1, r*)$, we are now able to distinguish between the features:

$$F_1(x) \leftarrow Has_Car(x, y), Short(y)$$
$$F_3(x) \leftarrow Has_Car(x, y), Closed(y)$$
$$F_{10}(x) \leftarrow Short_Closed(x)$$

Now, $Relev(F_1) = Relev(F_3) = [r2, r2]$, but $Relev(F_{10}) = [r*, r*]$. We would expect any explanation that uses F_{10} to be prefered over one that does not.

It may be possible to estimate empirically the relevance of features using data (for example, a feature that occurs frequently in explanations is highly relevant). Incorporating this kind of information will require changes to the definition calculating the relevance of features. The comparison of labelled expressions then proceeds as before: we do not pursue the use of empirical estimates of relevance in this paper.

4.3. Implementation

We finally have the pieces to define a label for an explanation, given data D and background B. Each explanation H will now have the label $\langle L_H, P_H \rangle$, where $L_H = Fidelity(H|D, B)$ and $P_H = Relev(H)$. Using a dictionary-ordering to compare labelled explanations allows us to decompose the task of identifying explanations into two parts: the first that maximises fidelity and the second that maximises the relevance. Further, as we have already seen (Remark 4), structuring cannot increase fidelity, but can increase relevance (Remark 8). Therefore, with a dictionary ordering on labels, it suffices to search first over the space of unstructured explanations, and then over the space of structured explanations that unfold to the unstructured explanations with maximal fidelity. Algorithm 3 extends the previous procedure of finding any unstructured explanation (Algorithm 1) to obtain a maximal-fidelity unstructured explanation.

Algorithm 3: Identifying an unstructured explanation with maximal fidelity. Here, $Clause(e, F)$ denotes the clause $\forall x(Class(x, c) \leftarrow Body)$, where $Body$ is a conjunction of literals $F_i(x)$ with $F_i \in F$. In practice, we will need to extend this procedure to return all unstructured explanations with maximal fidelity.

Algorithm `ConstructUnstructOpt`$(e, B, \mathcal{F}, E^+, E^-)$
- **Input:** A relational example e; background knowledge B; a set of features $\mathcal{F}$ with definitions in B; and E^+, E^- as defined in Defn. 12.
- **Output:** A maximal fidelity unstructured explanation H s.t. $B \cup H \models e$.

Let $e = Class(a, c)$.
Let $a' = FV(a)$.
Let $\mathcal{F}'$ be the set of features that map to $TRUE$ in a'.
Let $D = (E^+, E^-)$.
Let $\mathcal{H}$ be the subset-lattice of $\mathcal{F}'$.
Let $\mathcal{F}''$ be any element in $\mathcal{H}$ s.t.
  $Body$ is the conjunction of features in $\mathcal{F}''$,
  $C = \forall x(Class(x, c) \leftarrow Body)$,
  $L = Fidelity(\{C\}|D, B)$, and
  There is no other element $\mathcal{G}$ in $\mathcal{H}$ s.t. $Fidelity(\{Clause(e, \mathcal{G})\}|D, B) > L$.
return $\{C\}$

It is insufficient to simply get one maximal-fidelity explanation, and we will need to modify the procedure in Algorithm 3 return the set of unstructured explanations with maximal fidelity. For each element H in this set, Algorithm 4 extends *ConstructStruct* in Algorithm 2 to return a structured explanation with higher-relevance than H, if one exists. It is not hard to see that if $P_k = \emptyset$ then $P_{k+1} = \emptyset$. Therefore, it is only needed to call *ConstructExpl* with $k = 2, 3, \ldots$ until $P_k = \emptyset$. In experiments in this paper, we will adopt the even simpler strategy of only considering $k = 2$. That is, we will only consider 2-partitions of the set of features constituting the unstructured explanation H (in effect, seeking structured explanations with higher relevance than H, but using the minimum number of invented features). The structured explanations in Example 5 are examples of structures that can be obtained with $k = 2$.

Algorithm 4: A procedure for obtaining a structured explanation that is at least as relevant as an unstructured explanation H.

Algorithm `ConstructExpl`(H, B, k)
  Input: An unstructured explanation H; background knowledge B; and a number k (≥ 2).
  Output: An explanation H' s.t. $Relev(H|B) \preceq Relev(H'|B)$.
  Let $H = \forall x(Class(x, c) \leftarrow Body)$.
  Let $Relev(H) = R_H = \{[\alpha, \gamma]\}$.
  Let F be the set of features in $Body$.
  Let P_k be a set s.t. one of the following is true:
    Either (a) P_k is a k-partition of F that satisfies the following: at least one block in P_k contains 2 or more elements, and there is at least one block in P_k whose elements have a minimum relevance β and maximum relevance γ such that $\alpha \prec_r \beta \preceq_r \gamma$;
    Or (b) $P_k = \emptyset$ (if no such k-partition exists).
  if $P_k = \emptyset$ **then**
    return H
  end
  else
    Let P_k consist of blocks $b_1, b_2, \ldots, b_k$
    for *each block* $b_i \in P_k$ **do**
      Construct a new feature F'_i with definition $C_i = \forall x(F'_i(x) \leftarrow Body_i)$, where $Body_i$ is the conjunction of the features in b_i.
    end
    Let $C_0 = \forall x(Class(x, c) \leftarrow Body_0)$ where $Body_0$ is the conjunction of the $F'_1, F'_2, \ldots, F'_k$.
    Let $H' = \bigcup_{i=0}^{k} C_k$.
  end
  return $\{H'\}$

The structured explanation is obtained by inventing k features, the definition of at least one of which has a higher relevance than the unstructured explanation. Any process for obtaining such a k-partition is acceptable. With $k = 2$, we use a simple greedy strategy to identify a high-relevance block (one of the invented features). We start by selecting the feature with the highest relevance in the unstructured explanation. We continue to add

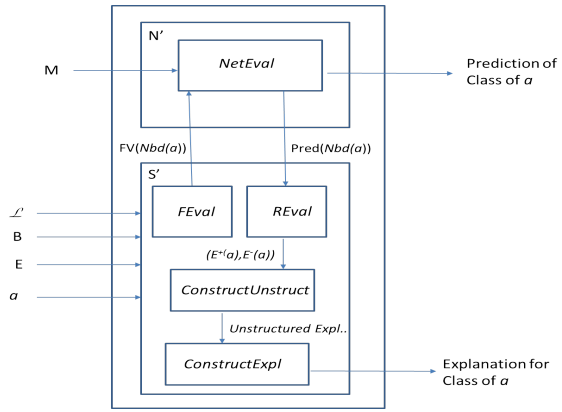


Figure 4: More details on the testing component in Fig. 1. E is the set of relational training instances, a is a test instance, and M is a DRM. FV($Nbd(a)$) denotes the feature-vectors for relational in E that are within the neighbourhood of a. Pred($Nbd(a)$) are the class predictions of instances in $Nbd(a)$ by the DRM. *NetEval* evaluates the model M on input data; *FEval* evaluates relational instances and returns their feature-vectors; and *REval* constructs relational examples. We assume that *FEval* has access to the feature-definitions found in the training stage.

features as long as the relevance of the corresponding invented feature is higher than that of the unstructured explanation. The second (low-relevance) block then contains all features from the unstructured explanation that are not in the high-relevance block.

Together, *ConstructUnstruct* and *ConstructExpl* are used to identify a local explanation for a relational instance e (see Fig. 4).

5. Empirical Evaluation

In this section, we investigate the following:

Explanation. We conduct the following experiments:

- **Expt. 1: Fidelity.** Can we construct a local symbolic explanations for an instance with high fidelity to local predictions made by the DRM?
- **Expt. 2: Relevance.** Does incorporating a prior preference based on relevance have any effect?

Some clarifications are necessary here: (a) By a local symbolic explanation in Expt. 1 we mean the use of a graph-search that returns the unstructured explanation with the highest fidelity, described in Section 4.3; and (b) By prior-preference in Expt. 2, we mean the

relevance-based ordering over structured or unstructured explanations. In Expt. 2, we confine ourselves to whether the use of the preference can change the explanation returned (either from a unstructured to a structured one, or from one unstructured explanation to another). We note that incorporation of a prior preference obtained from a human expert is still not sufficient to ensure comprehensibility of explanations by the expert. Evidence for this requires results in the form of cross-comparisons on the use of prior expert preference on explanations against expert comprehensibility of explanations. However, this is outside the scope of this paper.

It is useful, though not mandatory for the experiments here, that the models have good predictive performance. To this end, we report in an Appendix for the following:

Prediction. We conduct the following experiment:

> **Expt. 3: Accuracy.** Does the DRM constructed using randomly drawn features have good predictive performance?

By randomly drawn features in Expt. 3, we mean the rejection-sampling method described in Vig et al. (2018).

5.1. Materials

5.1.1. Data

We report results from experiments conducted using 7 well-studied real world problems from the ILP literature. These are: Mutagenesis (King et al., 1996a); Carcinogenesis (King and Srinivasan, 1996a); DssTox (Muggleton et al., 2008); and 4 datasets arising from the comparison of Alzheimer's drugs denoted here as *Amine*, *Choline*, *Scop* and *Toxic* (Srinivasan et al., 1996). Each of these have been shown to benefit from the use of a first-order representation and domain-knowledge, but there is still room for improvement in predictive accuracy. While good predictive performance is necessary, the principal motivation for the selection of the problems here is that for each dataset we also have access to domain-information about the relevance of predicates for the classification task considered.

Of these datasets, the first three (Mut188–DssTox) are predominantly relational in nature, with data in the form of the 2-d structure of the molecules (the atom and bond structure), which are diverse and can be of varying sizes. Some additional bulk properties of entire molecules obtained or estimated from this structure are also available. The Alzheimer datasets (Amine–Toxic) are best thought of as being quasi-relational. The molecules have a fixed template, but vary in number and kinds of substitutions made for positions on the template. A first-order representation has still been found to be useful, since it allows expressing concepts about the existence of one or more substitutions and their properties. The datasets range in size from a few hundred (relational) instances to a few thousands. This is extremely modest by the usual data requirements for deep learning. We refer the reader to the references cited for details of the domain-knowledge used for each problem.

5.1.2. Background Knowledge

For the relational datasets (Mut188–DssTox), background knowledge is in the form of general chemical knowledge of ring-structures and some functional groups. Background-

knowledge contains definitions used for concepts like: alcohols, aldehydes, halides, amides, amines, acids. esters, ethers, imines, ketones, nitro groups, hydrogen donors and acceptors, hydrophobic groups, positive- and negatively-charged groups, aromatic rings and non-aromatic rings, hetero-rings, 5- and 6-carbon rings and so on. These have been used in structure-activity applications of ILP before (King et al., 1996b; King and Srinivasan, 1996b). However, we note that none of these definitions are specifically designed for the tasks here. In addition, for Mut188 and Canc330, there are some bulk properties of the molecules that are available. For the Alzheimer problems (Amine–Toxic) domain knowledge consists of properties of the substituents in terms of some standard chemical measures like size, polarity, number of hydrogen donors and acceptors and so on. Predicates are also available to compare these values across substitutions. Again we refer the reader to the relevant ILP literature for more details.

In addition to the domain-predicates just described, we will also have information in the form of a relevance ordering as described in Srinivasan et al. (2003). That paper only refers to the Mut188 and Canc330 datasets. The same information is obtained from the domain-expert involved in that paper for the other problems in this paper. A complete description of the relevance assignment of predicates for each problem is in Appendix C.

5.1.3. Algorithms and Machines

Random features were constructed on an Intel Core i7 laptop computer, using VMware virtual machine running Fedora 13, with an allocation of 2GB for the virtual machine. The Prolog compiler used was Yap. Feature-construction uses the utilities provided by the Aleph ILP system (Srinivasan, 1999) for constructing most-specific clauses in a depth-bounded mode language, and for drawing clauses subsuming such most-specific clauses. No use is made of any of the search procedures within Aleph. The deep networks were constructed using the Keras library with Theano as the backend, and were trained using an NVIDIA K-40 GPU card.

5.2. Methods

The methods used for each of the experiments are straightforward

Explanation. Experiments 1 and 2 are concerned solely with the explanatory performance of the local symbolic models.

> For each dataset:
>
> 1. Construct a DRM using training data
> 2. For each test instance, obtain the local symbolic unstructured explanation(s) with the highest fidelity;
> 3. Estimate the overall fidelity of the symbolic explanations (Expt. 1)
> 4. Estimate the effect of using the relevance-based prior in hypothesis selection (Expt. 2)

Prediction. Experiment 3 (in the Appendix) is concerned solely with the predictive performance of the DRM on the datasets in the paper.[10]

For each dataset:

1. Obtain a set of random features $\mathcal{F}$;
2. Compute the Boolean-value for each $F \in \mathcal{F}$ for the data;
3. Construct a DRM N using training data and obtain its predictions on test instances;
4. Estimate the overall predictive performance of M (Expt. 1)

Some clarifications are necessary at this point:

- We use a straightforward Deep Neural Network (DNN) architecture. There are multiple, fully connected feedforward layers of rectified linear (ReLU) units followed by Dropout for regularization (see Goodfellow et al. (2016) for a description of these ideas). The model weights were initialized with a Gaussian distribution. The number of layers, number of units for each layer, the optimizers, and other training hyperparameters such as learning rate, were determined via a validation set, which is part of the training data. Since the data is limited for the datasets under consideration, after obtaining the model which yields the best validation score, the chosen model is then retrained on the complete training set (this includes the validation set) until the training loss exceeds the training loss obtained for the chosen model during validation.
- We use the Subtle algorithm (Blockeel and Valevich, 2016) to perform the subsumption-equivalence test used to determine redundant features.
- For all the datasets, 10-fold cross-validated estimates of the predictive performance using ILP methods are available in the ILP literature for comparison. We use the same approach. This requires constructing DRMs separately for each of the cross-validation training sets, and testing them on the corresponding test sets to obtain estimates of the predictive accuracy;
- We use the mode-language and depth constraints for the datasets that have been used previously in the ILP literature. For the construction of features, the rejection-sampler performs at most $10,000$ draws;
- We take *explanatory fidelity* to mean the probability that the prediction made by H on a randomly drawn instance agrees with the prediction made by the corresponding DRM. We use the same 10-fold cross-validation strategy for estimating this probability (for efficiency, we use the same splits as those used to estimate predictive accuracy).

10. A note of clarification is in order here. The purpose of the Experiment 3 is to show that a DRM can achieve predictive accuracies comparable to or better than the best reports in the literature for this data. This is necessary to support the main experiments on assessment of the use of relevance information, i.e., to control for the possibility of deficiencies with the DRM being used. We direct the reader elsewhere to an assessment of DRMs on large numbers of datasets, and against state-of-the-art prediction tools (Dash et al., 2018).

For a given train-test split Tr_i and Te_i, we proceed as follows. We obtain a DRM N_i using Tr_i. We start with a fidelity count of 0. For each instance x'_j in Te_i we obtain the class predicted by N_i for x'_j, and corresponding neighbourhood of x'_j in the training set Tr_i. This strategy ensures that all explanations are based only on data the model was actually trained on, an important consideration in the biological applications our datasets come from. The neighbourhood is partitioned into $\delta^+(x'_j)$ and $\delta^-(x'_j)$ using the predictions by N_i and high-fidelity unstructured explanation(s) H_{ij} are obtained. This is done using a beam-search over the lattice described in Algorithms 1 and 3. The size of the beam is 5 (that is, the top 5 unstructured explanations are returned).

- Assessments of fidelity require the definition of a neigbourhood. For each instance x'_j in Te_i we say any instance $x \in Tr_i$ is in the neighbourhood of x' iff $FV(x')$ and $FV(x)$ differ in no more than k features. This is just the Hamming distance between the pair of Boolean vectors. That is, the neighbourhood of a test instance x' consists of training instances x's whose feature-vector representation are within a k-bit Hamming distance of x'. We will consider $k = 5$ and $k = 10$ in the experiments.
- In all cases, estimates are obtained using the same 10-fold cross-validation splits reported in the ILP literature for the datasets used. This will allow a cross-comparison of the results from Expt.1 to those reports.

5.3. Results

Results of the empirical evaluation are tabulated in Fig. 5. Results on the predictive accuracy of the DRM are in Fig. 10 in Appendix D, with appropriate discussion. Some supplementary results are in Fig. 7. The principal observations that can be made from the main tabulations in Fig. 5 are these: (1) High-fidelity symbolic explanations can be obtained for local predictions made by the DRM; and (2) In 6 of the 7 problems, introducing a prior-preference based on relevance does affect the selection of explanations.
Together, the results provide evidence for the following:

(a) It is possible to extract symbolic explanations for the prediction made by the DRM for a randomly drawn (new) instance. The explanations are largely consistent with the predictions of the network for that instance and its near-neighbours in the training data examined before by the network; and

(b) It is possible to incorporate domain-knowledge in the form of expert assessment of relevance of background predicates into a preference ordering for selecting amongst explanations.

Note that if the inclusion of relevance does not make a difference to selecting an explanation, we would expect values of 0.0 in Fig. 5(b). It is evident that the observed values are clearly not 0, for all cases except $DssTox$. This exception is unsurprising, since all predicates used for this problem have the same relevance (see Appendix C).
Before we examine some critical factors affecting the selection of local explanations, we re-emphasise three important characteristics of explanations:

Problem	Fidelity
$Mut18$	0.99(0.01)
$Canc330$	0.99(0.01)
$DssTox$	0.87(0.03)
$Amine$	0.98(0.01)
$Choline$	0.89(0.01)
$Scop$	0.89(0.02)
$Toxic$	0.94(0.02)

(a)

Problem	$\langle L_H, R_H \rangle > \langle L_H, \emptyset \rangle$
$Mut188$	0.82(0.08)
$Canc330$	0.77(0.07)
$DssTox$	0.00(0.00)
$Amine$	0.38(0.15)
$Choline$	0.27(0.03)
$Scop$	0.27(0.06)
$Toxic$	0.24(0.14)

(b)

Figure 5: Experiments 1 and 2. (a) Mean fidelity of the explanatory (symbolic) model to the predictive (neural) model. The number tabulated is a 10-fold cross-validation estimate of the faithfullness of the symbolic model to a DRM's prediction assessed over the neighbourhood of a test instance. The entries are for the smallest neighbourhood (H_5: see the "Methods" section for how this is computed). The number in parentheses are estimates of standard deviations. (b) Relative frequency estimates of how often we can expect incorporation of a relevance-based prior to affect the selection of explanations. The tabulation is the proportion of explanations for which a Bayes label using both fidelity and relevance ($\langle L_H, R_H \rangle$) is better than one that uses fidelity only ($\langle L_H, \emptyset \rangle$). The explanations are for the H_5 neighbourhood. Again, the estimates are from the same 10-fold cross-validation splits used elsewhere.

	Network		
Problem	≈ #Inputs	#HL	#Units/HL
$Mut188$	7100	3	15
$Canc330$	5400	3	6
$DssTox$	200	4	6
$Amine$	2200	3	12
$Choline$	2500	3	6
$Scop$	2400	4	13
$Toxic$	2300	3	14

Figure 6: Characteristics of DRM models. The numbers are averages across a 10-fold cross-validation. For any fold, the number of input layers and the units in a layer are automatically obtained in an optimisation step.

1. Explanations are constructed to maintain fidelity to the DRM's predictions in a local region, and not to the actual class labels of the instances in the region. Fidelity is agnostic therefore to the predictive accuracy of the DRM, and it is possible to obtain high-fidelity explanations for a low predictive accuracy DRM. This occurs in the results here on the $Canc330$ dataset, which is known to be more noisy than the others. The DRM has low predictive accuracy on this dataset, but nevertheless we are able to construct high-fidelity explanations for it.

2. Local explanations are not constructed for the model's prediction on the entire instance space, and focus instead on just a few (similar) instances. Although the networks can often contain 1000's of input features (see Fig. 6), any single instance usually consists of only a few "active" features. This allows the construction of compact local explanations that only require the presence of a small number of features (more on compactness below).

3. Since (local) fidelity is measured against the DRM's predictions of instances in a region and not against their actual class labels, any subset of instances that are close to the test instance can be chosen. In our experiments, this region is selected from the training set. This ensures that all explanations are based only on data the DRM was actually trained on. An alternative could be by random generation, as is done in Ribeiro et al. (2016), although this may pose some difficulty in ensuring instances are drawn from an appropriate instance space (for biochemical tasks like the ones here, for example, they have to be real molecules).

Unsurprisingly, two key factors affecting the selection of local explanations are the neighbourhood size, and the relevance information provided. We consider each of these in turn.

5.3.1. The Role of the Neighbourhood

Neighbourhood size affects the local explanations constructed (see Fig. 7). In general, the fewer the instances in the local neighbourhood, the less the constraints imposed on the explanations. This leads to smaller explanations (fewer literals), and higher fidelity.

The high-fidelity values in Fig. 7 are not an artifact of instances being very similar within a neighbourhood (that is, class distribution will be predominantly of the same value). Figure 8 shows this is not the case. Although class-ratios are less skewed as the size of the neighbourhood increases (from H_5 to H_{10}), it is not the case that high-fidelity can simply be obtained by default. We believe this is because a 1-bit difference here can mean the presence or absence of a non-trivial relational structure. A change of 5 such structural features can therefore be a very substantial change.

Finally, structured explanations only appear to play role for larger neighbourhoods. We suggest this is not so much to do with the size of the neighbourhood, as to the corresponding increasing in the size of the explanations. In general, larger explanations (those with more features) are likely to benefit from structuring.

5.3.2. The Role of Relevance

Broadly speaking, the relevance-ordering will prefer labelled explanations with higher relevance. The tabulation in Fig. 5(b) tells us the problems for which we can expect this

Problem	Nbd. Size	
	H_5	H_{10}
*Mut*188	5(1)	20(3)
*Canc*330	3(1)	27(4)
DssTox	19(3)	83(10)
Amine	9(1)	23(20
Choline	14(1)	53(3)
Scop	8(1)	26(2)
Toxic	12(2)	42(2)

(a)

Problem	Fidelity	
	H_5	H_{10}
*Mut*188	0.99(0.01)	0.96(0.01)
*Canc*330	0.99(0.01)	0.92(0.02)
DssTox	0.87(0.03)	0.85(0.02)
Amine	0.98(0.01)	0.97(0.01)
Choline	0.89(0.01)	0.83(0.01)
Scop	0.89(0.02)	0.86(0.02)
Toxic	0.94(0.02)	0.90(0.03)

(b)

Problem	Expl. Size	
	H_5	H_{10}
*Mut*188	1(1)	2(1)
*Canc*330	1(1)	3(1)
DssTox	3(1)	4(1)
Amine	2(1)	2(1)
Choline	2(1)	3(1)
Scop	2(1)	2(1)
Toxic	2(1)	3(1)

(c)

Problem	Struc. Expl.	
	H_5	H_{10}
*Mut*188	0.00	0.04
*Canc*330	0.02	0.36
DssTox	0.00	0.00
Amine	0.00	0.00
Choline	0.00	0 00
Scop	0.00	0.00
Toxic	0.00	0.00

(d)

Figure 7: Effect of the neighbourhood. (a) Average numbers of neighbouring instances for local explanations (rounded up); (b) Average fidelity of local explanations; (c) Average number of literals in the maximal-fidelity explanations (rounded up); and (d) Average proportion of structured explanations. All numbers are estimates obtained from 10-fold cross-validation (the numbers in parentheses are estimates of the standard deviation).

Problem	Pos-to-Neg Ratio	
	H_5	H_{10}
*Mut*188	0.77	0.61
*Canc*330	0.63	0.46
DssTox	0.21	0.13
Amine	0.58	0.50
Choline	0.58	0.49
Scop	0.54	0.48
Toxic	0.43	0.33

Figure 8: Average class ratios in neighbourhoods.

information to make a difference, in terms of the proportion of explanations for which we can expect relevance to affect selection.

Since the relevance information provided is an ordering, it is possible, but perhaps unnecessary, to correlate numerically the differences observed in Fig. 5(b) against the ranking. Nevertheless, the specific result in the tabulation for DssTox points to an important issue about the relevance ranking. There are clear differences between the role of the relevance information amongst the datasets (significant effect in Mut188 and Canc330; minor effect in the Alzheimer datasets; and no effect in DssTox). We conjecture the following condition for relevance to have an effect on selection:

> *The larger the range of the relevance assignment, the more likely it is that relevance will play a role in selection of explanations.*

For the datasets here, the range of the relevance assignment has 1 value for DssTox; 2 values for the Alzheimer's datasets; and 4 values each for Mut188 and Canc330. The corresponding proportions of explanations where relevance plays a role in selection are: 0.0 (DssTox); 0.29 (Alzheimer datasets); and 0.80 (Mut188 and Canc330).

Finally, since explanations are generated "on-demand", it is impractical to show the explanations for all test instances. A snapshot is nevertheless useful, and one such is shown in Fig. 9. The example is from the *Canc*330 problem. The unstructured explanation shown has perfect fidelity. The alternate structured explanation shown in Fig. 9 is derived from this unstructured explanation has the same fidelity (as expected), but higher relevance. In effect, what the structuring achieves here is to group predicates with the *Has_property* relation, which has high-relevance; and separate out predicates that mix predicates with low- and high-relevance. Some measure of the explanatory convenience provided by the symbolic model is apparent if the reader keeps in mind that the corresponding prediction by the DRM is based on around 2000 input features, about 400 of which are equal to 1 for the test instance. It is interesting that the domain-expert could correctly identify the class of the example when shown the symbolic explanation.

6. Other Related Work

We have already noted the key reference to LIME and to reports in the ILP literature of immediate relevance to the work in this paper. Here we comment on other related work. The landmark work on structured induction of symbolic models is that of Shapiro (1987). In Shapiro's work structuring was top-down, with machine learning being used to learn sub-concepts identified by a domain-expert. The structuring is therefore hand-crafted, and with a sufficiently well-developed tool, a domain-expert can, in principle, invoke a machine learning procedure to construct sub-concepts using examples he or she provides. The technique was shown to yield more compact models than an unstructured approach on two large-scale chess problems, using decision-trees induced for sub-concepts.

Clearly the principal difficulty in the Shapiro-style of structured induction is the requirement for human intervention at the structuring step. The following notable efforts in ILP, or closely related areas, have been directed at learning structured theories automatically:

Unstructured explanation:

Label: $\langle L_H = 1.0, P_H = \{[1,4]\}\rangle$
Explanation H:
$Class(x,c) \leftarrow F_{537}(x), F_{1196}(x), F_{610}(x), F_{611}(x), F_{1657}(x)$

Structured explanation(s):

Label: $\langle\ L_{H_1} = 1.0, P_{H_1} = \{[1,4],[4,4]\}\rangle$
Explanation H_1:
$Class(x,c) \leftarrow F_{1,1}(x), F_{1,2}(x)$
$F_{1,1}(x) \leftarrow F_{1657}(x)$
$F_{1,2} \leftarrow F_{537}(x), F_{1196}(x), F_{610}(x), F_{611}(x)$

Feature-definitions:
$F_{537}(x) \leftarrow Atm(x,y,h,3,z), Gteq(z,0.115)$ (Relev = [1,1])
$F_{1196}(x) \leftarrow Atm(x,y,c,22,z), Gteq(z,-0.111), Atm(x,w,c,22,z)$ (Relev = [1,1])
$F_{610}(x) \leftarrow Non_ar_hetero_6_ring(x,u), Has_property(x,ames,p)$ (Relev = [2,4])
$F_{611}(x) \leftarrow Has_property(x,salmonella,n),$
$Has_property(x,mouse_lymph,p)$ (Relev = [4,4])
$F_{1657}(x) \leftarrow Has_property(x,cytogen_ca,n), Has_property(x,mouse_lymph,p),$
$Has_property(x,cytogen_sce,p)$ (Relev = [4,4])

Figure 9: Example explanations for a test instance from the $Canc330$ domain. The unstructured explanation H has perfect fidelity to the DRM's local predictions. The structured explanation H_1 is obtained from H and preserves fidelity, but predicates are grouped so that higher relevance is achieved.

- Inverse resolution, especially in the Duce system (Muggleton, 1987) was explicitly aimed at learning structured sets of rules in propositional logic. The sub-concepts are constructed bottom-up;

- Function decomposition, using HINT (Zupan et al., 2001), which learns hierarchies of concepts using automatic top-down function decomposition of propositional concepts;

- First-order theories with exceptions, using the GCWS approach (Bain, 1991), which automatically constructs hiearchical concepts. Structuring is restricted to learning exceptions to concepts learned at a higher level of the hierarchy;

- First-order tree-learning: an example is the TILDE system (Blockeel, 1999). In this, the tree-structure automatically imposes a structuring on the models. In addition, if each node in the tree is allowed a "lookahead"option, then nodes can contain conjunctions of first-order literals, each of which can be seen as defining a new feature. The model is thus a hierarchy of first-order features; and

- Meta-interpretive learning (Muggleton et al., 2015), which allows a very general form of predicate-invention, by allowing an abduction step when employing a meta-interpreter to use higher-order templates of rules that be used to construct proofs (in effect, explanations) for data. In principle, this would allow us not just to construct explanations on-demand, but also invent features on-demand. If the higher-order templates can be specialised to the domain, then it should be possible to control the feature-invention by relevance-information. Of course, this is unrelated to generating explanations for the predictions of a black-box classifier.

- Propositionalisation (relational feature construction) (Kramer, 2001) in principle enables the application of any propositional learner that can learn structured models, such as in (Zelezny and Lavrač, 2006), to the task of learning structured first-order models. More recently, propositionalization was applied in learning topic models, a class of two-level structured probabilistic models (Srinivasan et al., 2012). ILP propositionalization typically uses relational features constructed before learning commences (Srinivasan and King, 1996). For construction of relational explanations "on-demand", e.g., for local neighbourhoods of a test instance, there have been a number of approaches in ILP in which propositionalization occurs *during* the learning process. This appears to have been described first by (Alphonse and Rouveirol, 2000) where it was called "lazy propositionalization". Following this were the approaches of (Kramer, 2001), (Popescul and Ungar, 2004), and (Landwehr et al., 2006) in which this was termed "dynamic propositionalization".

Unsurprisingly, there has been a lot of research effort invested into extracting explanations for the predictions made by a neural network (see d'Avila Garcez et al. (2002), Chapter 3 for a full description of this line of research). Most of this effort has been in the direction of translating the network into a single logical model that guarantees correspondence to the neural model (for example, see Thrun, 1994). Alternatively, a single symbolic model can be learned that approximates the behaviour of the neural model over all inputs (for example, Craven and Shavlik, 1994; França et al., 2015)). Although distinct in their aims,

both approaches still result in two separate models, one neural and the other symbolic. In both cases the symbolic model is intended to be a readable proxy for the neural model, which can then form a basis for an explanation of "why" questions. But there are some inherent trade-offs:

- If we try to replicate exactly the behaviour of the neural network with a symbolic model (as is done, say, in Thrun, 1994), then the resulting model may not be any more comprehensible than the neural network; and
- If we try only to approximate the behaviour of the network using logical predicates (as is done, say, in Craven and Shavlik, 1994), then we run the risk of not being able to replicate the network's behaviour sufficiently accurately over all instances, because of inadequacies of the logical predicates available.

Both these issues are exacerbated for modern-day deep networks, with many hidden layers and large numbers of inputs. One way to side-step these difficulties is simply to drop the requirement—as is done here—of translating the *entire* network model into a single symbolic model.

An entirely different, and much more sophisticated kind of hybrid model combining connectionist and logical components has been proposed recently in the form of Lifted Relational Neural Networks (LRNNs: Sourek et al., 2015)). In LRNNs, the logical component is used to provide a template for ground neural network models, which are used to learn weights on the logical formulae. While we have largely stayed within the confines of classical ILP both for obtaining features and explanations, LRNNs are closely related to probabilistic models for ILP. An area of common interest arises though in the use of the network structure to invent new features (although in the LRNN case, this is not for local models as we proposed here).

7. Concluding Remarks

The recent successes of deep neural networks on predictive tasks have not, to any large extent, used either domain knowledge or representations significantly more expressive than simple relations (usually over sequences). The price for this has been a requirement for very large amounts of data, which provide the network with sufficient correlations necessary to identify predictively useful models. This works for problems where large amounts of data are being routinely generated automatically, and about which there may be little or no domain knowledge. The situation with scientific data is quite the opposite: data are sparse, but there is significant domain-knowledge, often built up over decades of painstaking experimental work. We would like powerful predictive models in such domains, but for this, the network would need a way of capturing what is known already, and be able to explain its predictions, in a language that is sufficiently expressive.

In this paper the predictive models are knowledge-rich deep networks (DRMs), which results suggest can achieve, and often exceed, the levels of predictivity reached by full first-order learners. However, our interests extend beyond prediction. We want to construct understandable models. For this we start with the approach taken in Ribeiro et al. (2016) and propose the use of a proxy for the DRM that acts as a readable explanation. The

proxy in this paper is in the form of a symbolic model for the predictions made by the DRM, constructed using techniques developed in Inductive Logic Programming (ILP). But there are at least three limitations we see arising from using the approach in Ribeiro et al. (2016). First, the goal is to generate a readable proxy for the *prediction* made by a black-box model. For us, the DRM is the black-box, but the approach here is sufficiently general to construct logical explanations for any predictor that uses relational features as input. We must emphasise that the logical explanation need not be the same as a readable proxy for the (true-)value of the instance. Second, readability does not guarantee comprehensibility: in Michie (1995) examples are shown of readable, but still incomprehensible, models. Thirdly, an important quality normally required of good explanations, causality, does not explicitly play a role. The first issue is inherent to the purpose of the model, and we have not attempted to change it here. The incorporation of a semantic prior based on relevance is a first attempt to address directly the second issue, and indirectly may address the third partially (non-causal explanations should have low relevance). To construct causal explanations correctly we will need more information than assigning relevance labels to predicates. We will also need the explanation-generator to pose counterfactual queries to the black-box, and the black-box to be able answer such queries with high accuracy. At this point, there is some evidence that symbolic learning could be adapted to suggest new experiments (see, for example, King et al., 2004), but it is not known how well DRMs will perform if the distribution of input values is very different to those that were used to train the network. So, at this point, we have restricted ourselves to constructing readable explanations for predictions that take into account prior preferences.

In this paper we have proposed the use of DRMs and the results here are consistent with previous work showing that such models are powerful predictors. However, it is important to understand also what the experimental evidence presented does not tell us. It does not tell us, for example, that a deep network without first-order features will not achieve the same performance as those tabulated here. However, it is not immediately apparent how this conjecture could be tested, since no more data are available for the problems. However it may be possible to transfer features constructed by a network trained on other problems with more data.

There are at least three separate directions in which we plan to extend the work here. First, interactions with the domain-expert suggests that the relevance information we have used here can be made much more fine-grained. It is possible, for example that certain combinations of predicates may be more relevant than others (or, importantly, certain combinations are definitely not relevant). None of this is accounted for in the current feature-generation process, and we intend to investigate relevance-guided sampling in place of the simple random sampling we use at present. Secondly, we would like to explore the construction of causal explanations for DRMs, by combining counterfactual reasoning with the use of a generative deep network capable of generating new instances. Thirdly, it is necessary at some point in the future, to establish the link between local symbolic explanations and human comprehensibility. For this, we would need to conduct a cross-comparison of structured and unstructured explanations and ratings of their comprehensibility by a domain-expert. Recently (Schmid et al., 2016) experiments have been reported in the ILP literature on assessing comprehensibility, when invention of predicates is allowed. Similar

experiments will help assess the human-comprehensibility of local symbolic explanations for black-box classifiers.

Acknowledgments

A.S. is a Visiting Professorial Fellow, School of CSE, UNSW Sydney. A.S. is supported by the SERB grant EMR/2016/002766.

Appendix A. Some terminology from Logic Programming and ILP

In this section we cover only terminology used in the paper. For additional background and further terminology see Chang and Lee (1973); Nilsson (1980); Lloyd (1987); Muggleton and De Raedt (1994).

A language of first order logic programs has a vocabulary of constants, variables, function symbols, predicate symbols, logical implication '$\leftarrow$', and punctuation symbols. A function or predicate can have a number of arguments known as *terms*. Terms are defined recursively. A constant symbol (or simply "constant") is a term. A variable symbol (or simply "variable") is a term. If f is an m-ary function symbol, and $t_1, \ldots, t_m$ are terms, then the function $f(t_1, \ldots, t_m)$ is a term. A term is said to be *ground* if it contains no variables. If p is an n-ary predicate symbol, and $t_1, \ldots, t_n$ are terms, then the predicate $p(t_1, \ldots, t_n)$ is an atom. Predicates with the same predicate symbol but different arities are distinguished by the notation p/n where p is a predicate of arity n. We follow the standard Prolog syntax conventions. Constant symbols are written as a lower-case letter followed by a string of lower- or upper-case letters, digits or underscores ('_'). Variables are written similarly, except that the first letter must be upper-case. A literal is either an atom or the negation of an atom. If a literal is an atom it is referred to as a positive literal, otherwise it is a negative literal. A clause is a disjunction of the form $A_1 \vee \ldots \vee A_i \vee \neg A_{i+1} \vee \ldots \vee \neg A_k$, where each A_j is an atom. Alternatively, such a clause may be represented as an implication (or "rule") $A_1, \ldots, A_i \leftarrow A_{i+1}, \ldots, A_k$. A definite clause $A_1 \leftarrow A_2, \ldots, A_k$ has exactly one positive literal, called the *head* of the clause, with the literals $A_2, \ldots, A_k$ known as the *body* of the clause. A definite clause with a single positive literal is called a *unit* clause, and a clause with at most one positive literal is called a Horn clause. A Horn clause with no positive literal is called a *goal* clause. A set of Horn clauses is referred to as a logic program. It is often useful to represent a clause as a set of literals.

A *substitution* θ is a finite set $\{v_1/t_1, \ldots, v_n/t_n\}$ mapping a set of n distinct variables v_i, $1 \leq i \leq n$, to terms t_j, $1 \leq j \leq n$ such that no term is identical to any of the variables. A substitution containing only ground terms is a *ground* substitution. For substitution θ and clause C the expression $C\theta$ denotes the clause where every occurrence in C of a variable from θ is replaced by the corresponding term from θ. If θ is a ground substitution then $C\theta$ is called a ground clause. Since a clause is a set, for two clauses C, D, the set inclusion $C\theta \subseteq D$ is a partial order called *subsumption*, usually written C θ-subsumes D and denoted by $C \preceq D$. For a set of clauses S and the subsumption ordering $\preceq$, we have that for every pair of clauses $C, D \in S$, there is a least upper bound and greatest lower bound, called, respectively, the least general generalisation (lgg) and most general unifier (mgu) of C and

D, which are unique up to variable renaming. The subsumption partial ordering on clauses enables the definition of a lattice, called the *subsumption lattice*.

Appendix B. Fidelity and Likelihood

The model for noisy data in McCreath and Sharma (1998) can be adapted to the construction of local explanations that are not completely consistent with local predictions (that is, fidelity < 1).

Remark 9 (Bayesian Posterior McCreath and Sharma (1998)) *Given a relational instance $a \in \mathcal{X}$, let N be predictive model s.t. $N(FV(a)) = c$ for $c \in \mathcal{Y}$. Let sets E^+ and E^- denote the local neighbourhood of a and $D = (E^+, E^-)$. Let H be a local explanation for a (not necessarily consistent), using background knowledge B. Then, if $\theta(H)$ is the proportion of all instances in $\mathcal{X} \times \mathcal{Y}$ covered by H, and ϵ is an estimate of inconsistency (noise) allowed $\epsilon \in [0, 1]$), the log posterior is given by McCreath and Sharma (1998):*

$$\log P(H|D, B) \;=\; \log P(D|H, B) + \log P(H|B) - \log(D|B)$$

where $P(H|B)$ denotes the prior probability, and:

$$\log P(D|H, B) \;=\; |TP(H)|\log\left(\frac{1-\epsilon}{\theta(H)} + \epsilon\right) + |TN(H)|\log\left(\frac{1-\epsilon}{1-\theta(H)} + \epsilon\right) + |FPN(H)|\log(\epsilon)$$

is the log-likelihood. Here $TP(H)$ is the set $\{e : e \in E^+ \text{ and } B \wedge H \models e\}$; $TN(H)$ is the set $\{e : e \in E^- \text{ and } B \wedge H \wedge \neg e \not\models \Box\}$; and $FPN(H) = D - (TP(H) \cup TN(H))$.

It is not hard to see that TP and TN correspond to $AgreePos$ and $AgreeNeg$ in Defn. 13. In some cases, it is in fact sufficient to maximise fidelity, to maximise the log-likelihood.

Remark 10 (Fidelity and Log-Likelihood) *Let $H_{1,2}$ be local explanations for a relational example $Class(a, c)$, given D, B. If $\theta(H_1) = \theta(H_2)$, $|TP(H_2)| \geq |TP(H_1)|$ and $|TN(H_2)| \geq |TN(H_1)|$ then: (a) $Fidelity(H_2|D, B) \geq Fidelity(H_1|D, B)$; and (b) $\log P(D|H_2, B) \geq \log P(D|H_1, B)$.*

For given D, B, from Defn. 9 the log-likelihood is $\log P(D|H, B)$. Then, from Defn. 13, $Fidelity(H_1|D, B) = \frac{|TP(H_1)|+|TN(H_1)|}{|D|}$, and $Fidelity(H_2|D, B) = \frac{|TP(H_2)|+|TN(H_2)|}{|D|}$. Since $|TP(H_2) \geq |TP(H_1)|$, and $|TN(H_2)| \geq |TN(H_1)|$, trivially we obtain $Fidelity(H_2|D, B) \geq Fidelity(H_1|D, B)$.

For any explanation H, s.t. $0 < \theta)(H) < 1$ and for a fixed ϵ s.t. $0 \leq \epsilon \leq 1$, the log-multipliers $k_{1,2}(\theta(H))$ of $|TP(H)|$ and $|TN(H)|$ in the expression for $\log P(D|H, B)$ are both positive; and the log-multiplier k_3 of $|FPN(H)|$ is at most 0. Therefore $\log P(D|H)$ is $k_1(\theta(H))|TP| + k_2(\theta(H))|TN| - k_3(|D|) + k_3(|TP| + |TN|)$. If $\theta(H_1) = \theta(H_2)$, then $k_1(\theta(H_1)) = k_1(\theta(H_2)) = k_1$, say, and $k_2(\theta(H_1) = k_2(\theta(H_2)) = k_2$, say. Then $\log P(D|H_1)$ is $k_1|TP(H_1)| + k_2|TN(H_1)| - k_3(|D|) + k_3(|TP(H_1)| + |TN(H_1)|)$ and $\log P(D|H_2)$ is $k_1|TP(H_2)| + k_2|TN(H_2)| - k_3(|D|) + k_3(|TP(H_2)| + |TN(H_2)|)$, Since $|TP(H_2) \geq |TP(H_1)|$ and $|TN(H_2)| \geq |TN(H_1)|$, and $k_{1,2} > 0$ and $k_3 \geq 0$ it follows trivially that $\log P(D|H_2, B) \geq \log P(D|H_1, B)$.

Problem	Accuracy			
	OptILP Srinivasan and Ramakrishnan (2011)	*Stat* Saha et al. (2012)	*DRM* Lodhi (2013)	*DRM* (here)
*Mut*188	0.88(0.02)	0.85(0.05)	0.90(0.06)	0.91(0.06)
*Canc*330	0.58(0.03)	0.60(0.02)	–	0.68(0.03)
DssTox	0.73(0.02)	0.72(0.01)	0.66(0.02)	0.70(06)
Amine	0.80(0.02)	0.81(0.00)	–	0.89(0.04)
Choline	0.77(0.01)	0.74(0.00)	–	0.81(0.03)
Scop	0.67(0.02)	0.72(0.02)	–	0.82(0.06)
Toxic	0.87(0.01)	0.84(0.01)	–	0.93(0.03)

Figure 10: Experiment 3. Estimated predictive accuracies of DRMs against some of the best reported performances in the ILP literature. All estimates are from the same 10-fold cross-validation splits in the reports cited.

Appendix C. Relevance Information

The following problem-specific relevance assignments for predicates in the background knowledge were obtained from R.D. King, University of Manchester.

Problem	Relevance	Predicates
Mut188	1	Atoms and bonds
	2	3-dimensional distance
	3	Functional groups and rings
	4	LUMO, hydrophobicity
	5	Expert-identified indicator variables
Canc330	1	Atoms and bonds
	2	Functional groups and rings
	3	Carcinogenic alerts
	4	Outcome of genetic tests
DssTox	1	Atoms and bonds
Alzh. Datasets	1	Substitutions at templates
	2	Hansch-type predicates (size, polarity *etc.*)

Appendix D. Predictive Accuracy of the DRM

The principal observations that can be made from the main tabulations in Fig. 10 are that the predictive accuracy of the DRMs clearly compare favourably to the best reports in the literature. Therefore the results provide evidence that a deep relational machine (DRM) equipped with domain knowledge and randomly drawn first-order features can construct good predictive models using (by deep-learning standards) very few data instances.

Quantitative assessments of the results are also possible, with the usual cautions associated with small numbers and multiple comparisons. The appropriate test for comparing the predictive accuracy of the DRM is the Wilcoxon signed-rank test, with the null hypothesis that the DRM's accuracy is the same as the method being compared. This yields P-values of < 0.05 for the comparisons against *OptILP* and *Stat* (we omit a comparison against the DRM in Lodhi (2013), due to lack of data).

Appendix E. Example Explanations

The following explanations are for a test-instance in the *Canc*330 problem (specifically, test-instance 2 on the 3rd cross-validation split). This example was chosen since it illustrates a number of interesting aspects: all explanations have perfect fidelity; structuring increases relevance; and there are several structured explanations possible. The DRM has 2196 features, of which 397 are active for this instance. The DRM correctly predicts the instance as belonging to the "positive" class. All the symbolic explanations below predict the same class-values as the DRM for the test-instance and its neighbours.

Unstructured explanation:

Label: $\langle L_H = 1.0, P_H = \{[1,4]\}\rangle$
Explanation H:
$Class(x,c) \leftarrow F_{537}(x), F_{1196}(x), F_{610}(x), F_{611}(x), F_{1657}(x)$

Structured explanation(s):

Label: $\langle\ L_{H_1} = 1.0, P_{H_1} = \{[1,4],[4,4]\}\rangle$
Explanation H_1:
$Class(x,c) \leftarrow F_{1,1}(x), F_{1,2}(x)$
$F_{1,1}(x) \leftarrow F_{1657}(x)$
$F_{1,2} \leftarrow F_{537}(x), F_{1196}(x), F_{610}(x), F_{611}(x)$

Label: $\langle\ L_{H_2} = 1.0, P_{H_2} = \{[1,4],[4,4]\}\rangle$
Explanation H_2:
$Class(x,c) \leftarrow F_{1,1}(x), F_{1,2}(x)$
$F_{1,1}(x) \leftarrow F_{611}(x)$
$F_{1,2} \leftarrow F_{537}(x), F_{1196}(x), F_{610}(x), F_{1657}(x)$

Label: $\langle\ L_{H_3} = 1.0, P_{H_3} = \{[1,4],[4,4]\}\rangle$
Explanation H_3:
$Class(x,c) \leftarrow F_{1,1}(x), F_{1,2}(x)$
$F_{1,1}(x) \leftarrow F_{1657}(x), F_{611}(x)$
$F_{1,2} \leftarrow F_{537}(x), F_{1196}(x), F_{610}(x)$

Feature-definitions:
$F_{537}(x) \leftarrow Atm(x,y,h,3,z), Gteq(z,0.115)$ (Relev = [1,1])
$F_{1196}(x) \leftarrow Atm(x,y,c,22,z), Gteq(z,-0.111), Atm(x,w,c,22,z)$ (Relev = [1,1])
$F_{610}(x) \leftarrow Non_ar_hetero_6_ring(x,u), Has_property(x,ames,p)$ (Relev = [2,4])
$F_{611}(x) \leftarrow Has_property(x,salmonella,n),$
$Has_property(x,mouse_lymph,p)$ (Relev = [4,4])
$F_{1657}(x) \leftarrow Has_property(x,cytogen_ca,n), Has_property(x,mouse_lymph,p),$
$Has_property(x,cytogen_sce,p)$ (Relev = [4,4])

References

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings.*, pages 420–434, 2001.

E. Alphonse and C. Rouveirol. Lazy propositionalisation for Relational Learning. In W. Horn, editor, *ECAI-2000: Proc. 14th European Conf. on Artificial Intelligence*, pages 256–260, 2000.

Michael Bain. Experiments in non-monotonic learning. In *Proceedings of the Eighth International Workshop (ML91), Northwestern University, Evanston, Illinois, USA*, pages 380–384, 1991.

Tarek R. Besold, Artur S. d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation. *CoRR*, abs/1711.03902, 2017.

H. Blockeel and S. Valevich. Subtle. Available at: https://dtai.cs.kuleuven.be/software/subtle/, 2016.

Hendrik Blockeel. Top-down induction of first order logical decision trees. *AI Commun.*, 12(1-2):119–120, 1999.

C-L. Chang and R. C-T. Lee. *Symbolic Logic and Mechanical Theorem Proving.* Academic Press, London, 1973.

G. Coletti and R. Scozzafava. A coherent qualitative Bayes' theorem and its application in artificial intelligence. In *Proc. 2nd Intl. Symp. Uncertainty Modeling and Analysis*, pages 40–44, 1993.

Mark Craven and Jude W. Shavlik. Using sampling and queries to extract rules from trained neural networks. In *Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994*, pages 37–45, 1994.

Tirtharaj Dash, Ashwin Srinivasan, Lovekesh Vig, Oghenejokpeme Orhobor, and Ross King. Large scale assessment of deep relational machines. In E. Bellodi F. Riguzzi and R. Zese, editors, *Proceedings of the 28th International Conference on Inductive Logic Programming*, volume 11105 of *Lecture Notes in Computer Science*, pages 22–37. Springer, 2018.

Artur S. d'Avila Garcez and Gerson Zaverucha. The connectionist inductive learning and logic programming system. *Appl. Intell.*, 11(1):59–77, 1999.

Artur S. d'Avila Garcez, Krysia B. Broda, and Dov M. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications.* Perspectives in Neural Computing. Springer, 2002.

Manoel V. M. França, Gerson Zaverucha, and Artur S. d'Avila Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine Learning*, 94(1):81–104, 2014.

Manoel Vitor Macedo França, Artur S. d'Avila Garcez, and Gerson Zaverucha. Relational knowledge extraction from neural networks. In *Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches co-located with the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015), Montreal, Canada, December 11-12, 2015.*, 2015.

D. Gabbay. *Labelled Deductive Systems*, volume 1. Clarendon Press, 1996.

Dov M. Gabbay, C.J. Hogger, and J.A. Robinson. *Logic Programming.* Vol. 5 of Handbook of Logic in Artificial Intelligence and Logic Programming. Clarendon Press, Oxford, 1998.

I.J. Good. *Good Thinking: The Foundations of Probability and its Applications.* University of Minnesota, Minneapolis, 1983.

Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning.* Adaptive computation and machine learning. MIT Press, 2016.

C.J. Hogger. *Essentials of Logic Programming.* Clarendon Press, Oxford, 1990.

Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137, 2015.

R. D. King and A. Srinivasan. Prediction of rodent carcinogenicity bioassays from molecular structure using inductive logic programming. *Environmental Health Perspectives*, 104:pp. 1031–1040, Oct. 1996a.

R. D. King, S. H. Muggleton, A. Srinivasan, and M J Sternberg. Structure-activity relationships derived by machine learning: the use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 93(1):438–42, January 1996a.

R.D. King and A. Srinivasan. Prediction of rodent carcinogenicity bioassays from molecular structure using inductive logic programming. *Environmental Health Perspectives*, 104(5): 1031–1040, 1996b.

R.D. King, S.H. Muggleton, A. Srinivasan, and M.J.E. Sternberg. Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. of the National Academy of Sciences*, 93:438–442, 1996b.

Ross D. King, Kenneth E. Whelan, Ffion M. Jones, Philip G. K. Reiser, Christopher H. Bryant, Stephen H. Muggleton, Douglas B. Kell, and Stephen G. Oliver. Functional genomic hypothesis generation and experimentation by robot scientist. *Nature*, 427:247–52, 2004.

S. Kramer. Demand-Driven Construction of Structural Features in ILP. In C. Rouveirol and M. Sebag, editors, *ILP 2001: Proc. 11th Intl. Conference on Inductive Logic Programming*, number 2157 in LNAI, Berlin, 2001. Springer.

N. Landwehr, A. Passerini, L. De Raedt, and P. Frasconi. K-Foil: Learning Simple Relational Kernels. In Y. Gil and R. Mooney, editors, *AAAI-2006: Proc. 21st National Conference on Artificial Intelligence*, pages 389–394, 2006.

J. W. Lloyd. *Logic Programming, 2nd Edition*. Springer-Verlag, Berlin, 1987.

Huma Lodhi. Deep relational machines. In *Neural Information Processing - 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part II*, pages 212–219, 2013.

Eric McCreath and Arun Sharma. LIME: A system for learning relations. In *Algorithmic Learning Theory, 9th International Conference, ALT '98, Otzenhausen, Germany, October 8-10, 1998, Proceedings*, pages 336–374, 1998.

R.S. Michalski. A theory and methodology of inductive learning. In R. Michalski, J. Carbonnel, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 83–134. Tioga, Palo Alto, CA, 1983.

D. Michie. The superarticulacy phenomenon in the context of software manufacture. *Proc. R. Soc. Lond. A*, 405:185–212, 1986.

D. Michie and R. Johnston. *The Creative Computer: Machine Intelligence and Human Knowledge*. Viking Press, 1984.

Donald Michie. Consciousness as an engineering issue, part 2. *Journal of Consciousness Studies*, 2(1):52–66, 1995.

T.M. Mitchell, R.M. Keller, and S.T. Kedar-Cabelli. Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1(1):47–80, 1986.

S. Muggleton. Inductive Logic Programming: derivations, successes and shortcomings. *SIGART Bulletin*, 5(1):5–11, 1994.

S. Muggleton. Inverse Entailment and Progol. *New Gen. Comput.*, 13:245–286, 1995.

S. Muggleton and L. De Raedt. Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, 19(20):629–679, 1994.

S.H. Muggleton. Duce, an oracle based approach to constructive induction. In *IJCAI-87*, pages 287–292. Kaufmann, 1987.

Stephen Muggleton, José Carlos Almeida Santos, and Alireza Tamaddoni-Nezhad. Toplog: ILP using a logic program declarative bias. In *Logic Programming, 24th International Conference, ICLP 2008, Udine, Italy, December 9-13 2008, Proceedings*, pages 687–692, 2008.

Stephen H. Muggleton, Dianhuan Lin, and Alireza Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited. *Machine Learning*, 100(1):49–73, 2015.

N. J. Nilsson. *Principles of Artificial Intelligence*. Springer-Verlag, Berlin, 1980.

A. Petterossi and M. Proietti. Transformation of Logic Programs. In D. Gabbay. Hogger and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 5 - Logic Programming*, pages 697–787. Clarendon Press, Oxford, 1998.

A. Popescul and L. Ungar. Dynamic Feature Generation for Relational Learning. In *3rd International Workshop on Multi-Relational Data Mining*, 2004.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

Amrita Saha, Ashwin Srinivasan, and Ganesh Ramakrishnan. What kinds of relational features are useful for statistical learning? In *ILP*, 2012.

Ute Schmid, Christina Zeller, Tarek R. Besold, Alireza Tamaddoni-Nezhad, and Stephen Muggleton. How does predicate invention affect human comprehensibility? In *Inductive Logic Programming - 26th International Conference, ILP 2016, London, UK, September 4-6, 2016, Revised Selected Papers*, pages 52–67, 2016.

A.D. Shapiro. *Structured Induction in Expert Systems*. Addison-Wesley, Wokingham, 1987.

Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezný, and Ondrej Kuzelka. Lifted relational neural networks. In *Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches co-located with the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015), Montreal, Canada, December 11-12, 2015.*, 2015.

A. Srinivasan. The Aleph Manual. Available at http://www.comlab.ox.ac.uk/oucl/ research/areas/machlearn/Aleph/, 1999.

A. Srinivasan and R. King. Feature construction with inductive logic programming: a study of quantitative predictions of biological activity aided by structural attributes. In S. Muggleton, editor, *ILP'96: Proc. 6th Inductive Logic Programming Workshop*, volume LNAI 1314, pages 89–104, 1996.

A. Srinivasan, S. H. Muggleton, M. J. E. Sternberg, and R. D. King. Theories for mutagenicity: A study in first-order and feature-based induction. *Artif. Intell.*, 85(1-2): 277–299, 1996.

A. Srinivasan, R.D. King, and M.E. Bain. An empirical study of the use of relevance information in Inductive Logic Programming. *Machine Learning Research*, 4(Jul):369–383, 2003.

Ashwin Srinivasan. Extracting context-sensitive models in inductive logic programming. *Machine Learning*, 44(3):301–324, 2001.

Ashwin Srinivasan and Ganesh Ramakrishnan. Parameter screening and optimisation for ILP using designed experiments. *Journal of Machine Learning Research*, 12:627–662, 2011. URL `http://portal.acm.org/citation.cfm?id=1953067`.

Ashwin Srinivasan, Tanveer Faruquie, Indrajit Bhattacharya, and Ross King. Topic models with relational features for drug design. In *ILP*, 2012.

Ian Stewart. The Ultimate in Anty-Particles. *Scientific American*, July, 1994.

Sebastian Thrun. Extracting rules from artifical neural networks with distributed representations. In *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pages 505–512, 1994.

L. Vig, A. Srinivasan, M. Bain, and A. Verma. An investigation into the role of domain-knowledge on the use of embeddings. In N. Lachiche and C. Vrain, editors, *Inductive Logic Programming. ILP 2017.*, volume 10759 of *Lecture Notes in Computer Science*, Cham, 2018. Springer.

F. Zelezny and N. Lavrač. Propositionalization-based relational subgroup discovery with RSD. *Machine Learning*, 62:33–63, 2006.

Blaz Zupan, Ivan Bratko, Marko Bohanec, and Janez Demsar. Function decomposition in machine learning. In *Machine Learning and Its Applications, Advanced Lectures*, pages 71–101, 2001.

Journal of Machine Learning Research 20 (2019) 1-27 Submitted 8/18; Published 8/19

Decoupling Sparsity and Smoothness in the Dirichlet Variational Autoencoder Topic Model

Sophie Burkhardt burkhardt@informatik.uni-mainz.de
Stefan Kramer kramer@informatik.uni-mainz.de
University of Mainz
Department of Computer Science
Staudingerweg 9
55128 Mainz, Germany

Editor: Mohammad Emtiyaz Khan

Abstract

Recent work on variational autoencoders (VAEs) has enabled the development of generative topic models using neural networks. Topic models based on latent Dirichlet allocation (LDA) successfully use the Dirichlet distribution as a prior for the topic and word distributions to enforce sparseness. However, there is a trade-off between sparsity and smoothness in Dirichlet distributions. Sparsity is important for a low reconstruction error during training of the autoencoder, whereas smoothness enables generalization and leads to a better log-likelihood of the test data. Both of these properties are encoded in the Dirichlet parameter vector. By rewriting this parameter vector into a product of a sparse binary vector and a smoothness vector, we decouple the two properties, leading to a model that features both a competitive topic coherence and a high log-likelihood. Efficient training is enabled using rejection sampling variational inference for the reparameterization of the Dirichlet distribution. Our experiments show that our method is competitive with other recent VAE topic models.

Keywords: variational autoencoders, topic models, Dirichlet distribution, reparameterization, generative models

1. Introduction

Generative models have been successfully employed for the modeling of text data since they allow to learn a model of the data and thus are highly interpretable. The arguably most successful model of this kind has been latent Dirichlet allocation (LDA, Blei et al., 2003), which represents each document as a mixture of different topics. This model is typically trained using either Markov chain Monte Carlo (Griffiths and Steyvers, 2004) or variational Bayes (Blei et al., 2003). Recently, several approaches were put forward to train LDA using variational autoencoder neural networks (Miao et al., 2016; Srivastava and Sutton, 2017), thus making them more easily adaptable and extendable by using advances from deep learning research.

Using a Dirichlet distribution as a prior for the latent variables has advantages and disadvantages. The biggest advantage is the sparsity that it encourages in the latent variables. However, we find that the desired sparsity is at odds with the generalization performance of the model. This was previously noticed by Wang and Blei (2009) who

©2019 Sophie Burkhardt and Stefan Kramer.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-569.html.

tried to decouple sparsity and smoothness in the discrete hierarchical Dirichlet process. In the case of variational autoencoders (VAEs), the problem can be analyzed in terms of the KL-divergence between two Dirichlet distributions, which diverges if the prior parameter is close to zero. Therefore, we are faced with a trade-off: Either the model is sparse and achieves a good reconstruction error, but has a high KL-divergence term and thus a low log-likelihood of the model, or, it avoids low settings of the prior and thereby achieves a better log-likelihood at the cost of a higher reconstruction error. A similar observation was made by Srivastava and Sutton (2018) who observe that a slow minimization of the KL-divergence leads to better topics. In this paper we will discuss how this trade-off can be avoided to achieve both sparsity and smoothness in a variational autoencoder Dirichlet topic model.

One remaining problem with the training of generative models using variational autoencoders is the need for a differentiable non-centered reparameterization function (Kingma and Welling, 2014a) for the distribution of the latent variables. Unfortunately such a function is not available for the most widely used probability distribution in topic modeling, the Dirichlet distribution. A model that does not directly reparameterize the Dirichlet distribution is ProdLDA (Srivastava and Sutton, 2017) which employs a Laplace approximation for the Dirichlet distribution, thus enabling the training of a Dirichlet variational autoencoder. Other recent solutions include implicit reparameterization gradients (Figurnov et al., 2018), using an approximation for the inverse CDF (Joo et al., 2019) and using the Weibull distribution as a replacement (Zhang et al., 2018). In this work we will instead apply rejection sampling variational inference (Naesseth et al., 2017) to autoencoders.

The contributions of our paper are as follows:

1. We introduce a Dirichlet variational autoencoder based on RSVI (DVAE, see Figure 1a), which is as powerful and efficient as the Gaussian variational autoencoders except it uses the Dirichlet distribution as a prior on the latent variables.

2. We identify the trade-off between sparsity and smoothness as the main factor that prevents successful training of DVAE topic models and identify the KL-divergence between two Dirichlet distributions as the crucial factor influencing model performance.

3. We provide an adapted neural network architecture to decouple sparsity and smoothness.

4. We show that our Dirichlet variational autoencoder has an improved topic coherence, whereas the adapted sparse Dirichlet variational autoencoder has a competitive perplexity.

5. We propose to use the new topic redundancy measure to obtain further information on topic quality when topic coherence scores are high.

6. We perform an extensive experimental comparison with state-of-the-art VAE topic models to show that our model achieves the highest topic coherence.

2. Related Work

Variational autoencoders (VAEs) were first introduced by Kingma and Welling (2014b) and Rezende et al. (2014). These type of deep neural networks combine ideas from approximate Bayesian inference and deep neural networks, yielding generative models that can be trained by neural networks. Models are trained by stochastic backpropagation using the reparameterization trick. This trick allows the gradient to be backpropagated through the stochastic latent variables efficiently if a reparameterization is available as is the case for the Gaussian distribution.

Topic models based on latent Dirichlet allocation (LDA) (Blei et al., 2003) are directed generative models, which makes it possible to train such topic models using variational autoencoders. The first such neural variational document model (NVDM) was developed by Miao et al. (2016). This model successfully uses the idea of variational autoencoders to train a topic model with a Gaussian prior for the latent variables. This work was extended by Miao et al. (2017), who introduce Gaussian stick breaking priors in a model which is able to let the number of topics grow dynamically.

Another model using Dirichlet stick-breaking priors is SB-VAE (Nalisnick and Smyth, 2017), however, it requires Taylor expansion to compute the KL-divergence and a Kumaraswamy distribution to approximate the posterior. Since it uses a nonparametric stick-breaking prior instead of a parametric Dirichlet it is not directly comparable to our model.

LDA topic models use a Dirichlet distribution as a prior for the latent variables since it promotes sparsity and leads to more interpretable topics. Therefore, Srivastava and Sutton (2017) propose a VAE topic model called ProdLDA that employs a Laplace approximation for modeling a Dirichlet prior of the latent variables. As their results show, while the perplexity is not as good as that of previous models, the topic coherence is significantly improved over Gaussian VAEs. They also show that the Dirichlet prior leads to more sparseness in the document-topic distributions. This work is extended by Srivastava and Sutton (2018) who introduce a hierarchical VAE topic model. Their analysis shows that batch normalization leads to a slower minimization of the KL-divergence which improves the topic quality.

The reason that VAEs cannot be directly trained using a Dirichlet prior is that there is no simple variable transformation for the Dirichlet distribution. Whereas a Gaussian variable $z \sim N(\mu, \sigma^2)$ can be reparameterized as $z = \mu + \epsilon\sigma, \epsilon \sim N(0, 1)$ thus allowing the gradient to be backpropagated through the latent variable z, there is no such reparameterization for the Dirichlet distribution. However, there is a method to solve this problem which is based on rejection samplers (Naesseth et al., 2017), called RSVI. The Dirichlet distribution can be represented by gamma distributions for which there exists an efficient rejection sampler. The proposal function of this rejection sampler can be used for reparameterization with one minor difference: A correction term has to be added to the gradient that accounts for the difference between the proposal function and the true gamma distribution. In this paper we show how this elegant method can be used for efficient and effective training of Dirichlet variational autoencoders.

A different approach was taken by Zhang et al. (2018), who used the Weibull distribution as an approximation to the gamma distribution. This takes advantage of the fact that the Weibull and Gamma distributions have similar probability density functions and the Weibull

distribution is reparameterizable. Zhang et al. (2018) also compare their model with a model trained using RSVI. Overall, their model structure is different however, with a hierarchical generative distribution and a hybrid training algorithm.

Other recent solutions include implicit reparameterization gradients by Figurnov et al. (2018) who introduce a trick to avoid inverting the CDF by differentiating it instead. Furthermore, Joo et al. (2019) use an approximation for the inverse CDF of the gamma distribution. We compare to both of these methods (see Section 5.3 for details).

Our work draws significant inspiration from Wang and Blei (2009). This work is concerned with the HDP-LDA topic model which corresponds to a hierarchical Dirichlet process (HDP) applied to text, resulting in the nonparametric version of latent Dirichlet allocation (Blei et al., 2003). The model represents topics by multinomial distributions over words $\beta \sim Dir(\gamma)$, where β is the parameter vector of the multinomial distribution which is drawn from a Dirichlet distribution with parameter γ. The decoupling of sparsity and smoothness is applied to these probability distributions β, such that first the words are identified that contribute to the topic and then the Dirichlet distribution is only over the sparse word vector, i.e. a reduced probability simplex. The sparsity is modeled by assuming the topics are generated as $\beta \sim Dir(b \cdot \gamma)$, where b is a vector of Bernoulli variables with a Beta prior. In contrast, we apply the decoupling to the document-topic distributions instead of the word-topic distributions.

3. Background

In this section, we first introduce LDA topic models. We then explain how topic models are trained with variational autoencoders and how the variational autoencoders can be adapted to use the Dirichlet distribution as a prior for the latent variables.

3.1. LDA

Please note that the notation in this subsection follows the established conventions in topic modeling work. The rest of the paper may use the same variable names in different contexts.

Topic models based on latent Dirichlet allocation (LDA) describe each document in a collection of text documents as a mixture of topics. This means that each document d is associated with a document-topic distribution $\theta_d \sim Dir(\alpha)$ with a Dirichlet prior. For each word w_i, a topic z_i is drawn from θ_d and the word w_i is drawn from a topic-word distribution ϕ_{z_i} associated with the topic z_i. The joint distribution of the latent variables z, θ and ϕ is then optimized to maximize the marginal log-likelihood of the data D under the given model,

$$L(D|\theta, \phi, z) = \sum_{d \in D} \log p(d|z, \theta, \phi). \tag{1}$$

3.2. Variational Autoencoders

The notation for variational autoencoders (VAEs) unfortunately does not correspond well to the common notation in the context of LDA. In VAEs the document-topic distribution is generated by the encoder network parameterized by ϕ, whereas the topic-word distribution is given through the decoder network parameterized by θ. Since this paper mainly focuses

on VAEs, we adopt the notation common to VAEs here, although it is exactly reversed compared to the topic model notation.

Following Kingma and Welling (2014b), we define a generative model with a latent variable z and an observed input x that are generated from a generative distribution $p_\theta(z)p_\theta(x|z)$ with parameters θ. This generative model corresponds to the reconstruction/decoder part of the VAE, whereas the encoder corresponds to the variational approximation $q_\phi(z|x)$ parameterized by ϕ.

VAEs are generative models similar to LDA topic models. Accordingly, the input data x is assumed to be generated from a conditional distribution $p_\theta(x|z)$. The latent topic vectors z are generated from a prior distribution $p_\theta(z)$. Generally, it is assumed that the integral of the marginal likelihood $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$ is intractable as well as the posterior and the integrals required for mean-field variational Bayes. In the case of LDA, it is possible to train the model using mean-field variational Bayes, however, this restricts the model and does not easily allow for certain modifications. For example, using VAEs, it is easily possible to lift the restriction of the mixture model by removing the constraint on the parameters θ, which makes the model more expressive (Srivastava and Sutton, 2017).

The marginal log-likelihood of the model is given by

$$\log p(x) = \sum_{i=1}^{N} \log p(x_i) = \sum_{i=1}^{N} \left(KL[q(z_i|x_i) \,\|\, p(z_i|x_i)] + \mathcal{L}(\theta, \phi; x_i) \right),$$

where N is the number of documents in the data set.

Since the KL-divergence is always positive, the second term is the variational lower bound $\mathcal{L}$, which can be written as

$$\begin{aligned}\log p(x_i) \geq \mathcal{L}(\theta, \phi; x_i) = \mathbb{E}_{q(z|x_i)}[-\log q(z|x_i)] + \mathbb{E}_{q(z|x_i)}[\log p(x_i, z)] = \\ - KL[q(z_i|x_i) \,\|\, p(z_i)] + \mathbb{E}_{q(z_i|x_i)}[\log p(x_i|z_i)],\end{aligned}$$

where the first term in the second line corresponds to the entropy and the second term in the last line corresponds to the negative reconstruction error, whereas the KL-divergence acts as a regularizer that keeps the variational distribution close to the prior. To form an estimate of the ELBO, we need to sample z from q_ϕ and calculate the ELBO by accumulating the contributions of the samples

$$\tilde{\mathcal{L}}(\theta, \phi; x_i) = \frac{1}{L} \sum_{l=1}^{L} (\log p(x_i, z_{i,l}) - \log q(z_{i,l}|x_i)),$$

where L denotes the number of samples.

3.3. Rejection Sampling Variational Inference

The idea of rejection sampling variational inference (RSVI) is to use the proposal function of a rejection sampler as the reparameterization function. This makes it possible to use probability distributions that do not have non-central differentiable reparameterizations such as the Dirichlet distribution. It also utilizes statistical research into efficient rejection samplers.

The Dirichlet distribution can be simulated using gamma random variables since if $z_i \sim \Gamma(\alpha_i)$ then $\tilde{z}_{1:K} = \frac{z_{1:K}}{\sum_i z_i} \sim Dirichlet(\alpha_{1:K})$. This is relevant because for the gamma distribution there exists an efficient rejection sampler

$$z = h_\Gamma(\epsilon, \alpha) := (\alpha - \frac{1}{3})(1 + \frac{\epsilon}{\sqrt{9\alpha - 3}})^3, \epsilon \sim N(0, 1). \tag{2}$$

However, using this proposal function is not equivalent to using a gamma distribution because some samples are rejected. Therefore we need the distribution of an accepted sample $\epsilon \sim s(\epsilon)$, which is obtained by marginalizing over the uniform variable u of the rejection sampler,

$$\pi(\epsilon; \theta) = \int \pi(\epsilon, u; \theta) du = s(\epsilon) \frac{q(h(\epsilon, \theta))}{M_\theta r(h(\epsilon, \theta))},$$

where r is the proposal function for the rejection sampler, $z = h(\epsilon, \theta), \epsilon \sim s(\epsilon)$ is the reparameterization of the proposal distribution and M_θ is a constant used in the rejection sampler. Using this, the ELBO can be rewritten as

$$\begin{aligned}\mathbb{E}_{q(z|x_i)}[-\log q(z|x_i)] + \mathbb{E}_{q(z|x_i)}[\log p(x_i, z)] = \\ \mathbb{E}_{q(z|x_i)}[-\log q(z|x_i)] + \mathbb{E}_{\pi(\epsilon;\theta)}[\log p(x_i, h(\epsilon, \theta))]\end{aligned}$$

As Naesseth et al. (2017) show, the gradient can then be decomposed into three parts

$$\nabla_\theta \mathcal{L}(\theta) = g_{\text{rep}} + g_{\text{cor}} + \nabla_\theta \mathbb{E}_{q(z|x_i)}[-\log q(z|x_i)], \tag{3}$$

where g_{rep} and g_{cor} for the case of a one-sample Monte Carlo estimator are given as

$$g_{\text{rep}} = \nabla_z \log p(x_i, z) \nabla_\theta h(\epsilon, \theta),$$

$$g_{\text{cor}} = \log p(x_i, z) \nabla_\theta \log \frac{q(h(\epsilon, \theta))}{r(h(\epsilon, \theta))},$$

and the entropy can be calculated analytically. g_{rep} corresponds to the gradient assuming that the proposal is exact and always accepted and g_{cor} corresponds to a correction part of the gradient that accounts for not using an exact proposal.

3.4. Shape Augmentation for Gamma Distribution

The rejection sampler for the gamma distribution has higher acceptance rates for higher values of the parameter α. Naesseth et al. (2017) use this fact to augment the shape of the gamma distribution and thereby achieve a lower variance gradient. As they show, a $\Gamma(\alpha, 1)$ distributed variable z can be expressed as $z = \tilde{z} \prod_{i=1}^{B} u_i^{\frac{1}{\alpha+i-1}}$ for a positive integer B and i.i.d. uniform random variables u, where $\tilde{z} \sim \Gamma(\alpha + B, 1)$. This makes it possible to use the rejection sampler for $\alpha < 1$ and also decreases the correction term for increasing $\alpha + B$.

4. Method

In this section, the proposed method is described. We first discuss the trade-off between sparsity and smoothness, then put forward the adapted VAE topic model, explain how the KL-divergence is is minimized and cover the neural network architecture in detail. Finally, the issue of component collapsing is discussed.

4.1. Sparsity and Smoothness

The trade-off between sparsity and smoothness is a known issue in the topic modeling literature (Wang and Blei, 2009). Aspects of it were also discussed in recent literature by Srivastava and Sutton (2017, 2018). According to them, batch normalization and dropout slow down the minimization of the KL-divergence in the beginning of training. This has the effect of increasing the smoothness and thereby increases the topic coherence because the model generalizes better. A fast minimization puts a greater emphasis on sparseness which ensures a lower perplexity, but is prone to overfitting. A minimization that is too fast leads to maximal smoothness which means the Dirichlet parameter is equal to the prior. In this case no meaningful topics can be learned since the latent variables are only sampling noise. This problem is known as component collapsing.

But influencing the training process through e.g. batch normalization or dropout is only one way to control this trade-off. By decoupling both aspects explicitly we can allow both, a good generalization performance and a good fit to the data at hand. To do this, first, the topics are selected that represent the current document. Second, a distribution is sampled for the selected topics. The details are described in the next section.

4.2. DVAE Sparse

A schematic view of our methods is depicted in Figures 1a and 1c. We will now describe the sparse Dirichlet variational autoencoder (DVAE Sparse) depicted in Figure 1c. The DVAE model works analogously except that the vector b is set to a vector of only ones. The model equation is given by $p(x, z, \theta) = p(x|z, \theta) \cdot p(z|\theta) \cdot p(\theta)$. The input x is transformed to the Dirichlet parameter α and a sparsity parameter b by a neural network, the latent variables z are sampled from $Dir(b \cdot \alpha)$ and the decoder network reconstructs the original input. This is in contrast to the Gaussian autoencoder shown in Figure 1b, where the latent variables are drawn from a Gaussian distribution. To be able to backpropagate the gradients through the latent variables z, we use rejection sampling variational inference.

The parameter b is inspired by Wang and Blei (2009), who use a Bernoulli distributed variable. In contrast, we simply use a sigmoid function and subsequently round the result to zero or one. The distribution $Dir(b \cdot \alpha)$ is a degenerate Dirichlet distribution over the sub-simplex specified by

$$b = l_2(\lambda) = \text{round}(\frac{1}{1 + e^{-\lambda}}).$$

If the vector b only consists of ones, this reduces to the normal Dirichlet distribution $Dir(\alpha)$.

We optimize α in an unconstrained domain by letting

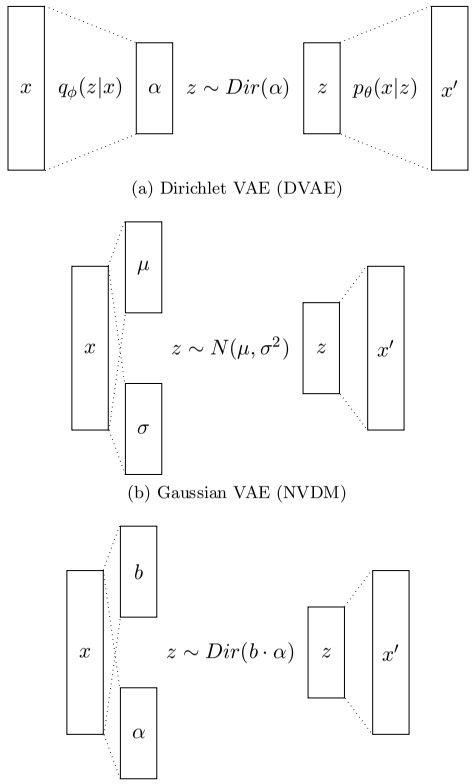
$$\alpha = \log(1 + \exp(a)), \tag{4}$$

where $a = l_1(\lambda)$ is a linear transformation of λ and λ is the transformation of the input through the neural network $\lambda = f_{\text{MLP}}(x)$.

The ELBO is given by

$$\mathcal{L}(\theta, \phi; x_i) = -KL[q(z|x_i) \,\|\, p(z)] + \mathbb{E}_{q(z|x_i)}[\log p(x_i|z)] = \tag{5}$$

$$- KL[q(z|x_i) \,\|\, p(z)] + \mathbb{E}_{\pi(\epsilon;\phi)}[\log p(x_i|h_\Gamma(\epsilon, \phi))]. \tag{6}$$



(a) Dirichlet VAE (DVAE)

(b) Gaussian VAE (NVDM)

(c) Sparse Dirichlet VAE (DVAE Sparse)

Figure 1: Schematic figure of Dirichlet VAE, Gaussian VAE and the proposed Sparse Dirichlet VAE. For the Dirichlet VAE the input is transformed by a neural network to yield parameter α. The latent variables z are subsequently sampled from $Dir(\alpha)$. Finally the input is reconstructed as x'. The Gaussian VAE has parameters μ and σ instead and uses a Gaussian distribution.

Here, the expectation representing the negative reconstruction error is rephrased with respect to the distribution of the accepted sample π and the latent variables z are reparameterized by $h_\Gamma(\epsilon, \phi)$ given in Equation 2. $\pi(\epsilon; \phi)$, the distribution of the accepted sample ϵ, is obtained by marginalizing over the uniform variable u,

$$\pi(\epsilon; \phi) = \int \pi(\epsilon, u; \phi) du = s(\epsilon) \frac{q(h_\Gamma(\epsilon, \phi))}{r(h_\Gamma(\epsilon, \phi))}.$$

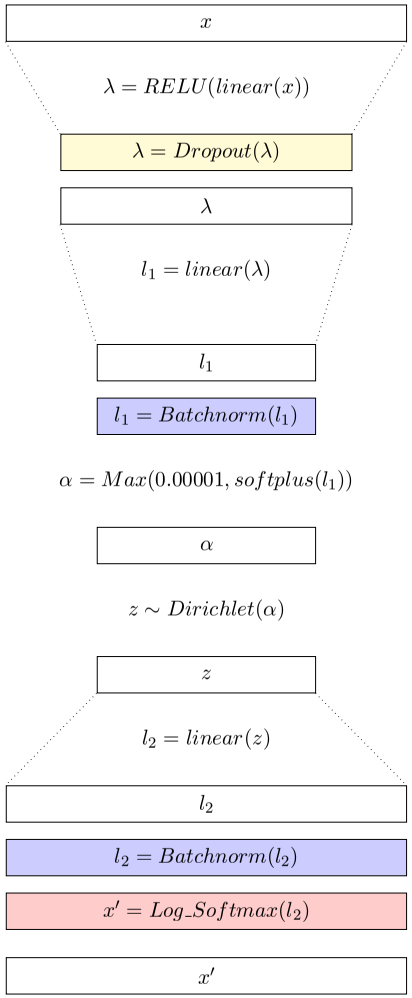


Figure 2: Illustration of the neural network architecture used for DVAE, ProdLDA, the implicit reparameterization gradient method, the Weibull VAE method and the inverse CDF gradient method. The architecture was proposed by Srivastava and Sutton (2017) because it allows to train topic models with a high learning rate, thus avoiding local minima and converging quickly.

Here, we use the parameter ϕ since our encoder network is parameterized by ϕ.

The gradient of the ELBO for the generative/decoder network is given as follows:

$$\nabla_\theta \mathcal{L}(\theta, \phi; x_i) = \nabla_\theta \mathbb{E}_{q(z|x_i)} \left[\log p(x_i|z)\right].$$

This corresponds to the logarithm of the reconstruction probability. Gradient backpropagation for this term is unproblematic since the samples used to estimate L depend on q which is parameterized by ϕ. Therefore we do not need to take care of reparameterization for this term. The gradients of the ELBO for the the variational/encoder network are as follows:

$$\nabla_\phi \mathcal{L}(\theta, \phi; x_i) = \nabla_\phi - KL[q(z|x_i) \,\|\, p(z)] + \nabla_\phi \mathbb{E}_{\pi(\epsilon;\theta)} \left[\log p(x_i|h_\Gamma(\epsilon, \phi))\right] \tag{7}$$

Here, we need to reparameterize the latent variables z. The gradient of the KL-divergence will be discussed in Section 4.3 and the rest of the gradient is given analogously to Equation 3. Note that we now use the expectation of the conditional distribution $p(x_i|z)$ instead of the joint distribution $p(x_i, z)$, since we write the ELBO in a different way which is more common for VAEs. The gradient based on a one-sample Monte Carlo estimator is given as

$$\nabla_\phi \mathcal{L}(\theta, \phi; x_i) = \nabla_\phi - KL[q(z|x_i) \,\|\, p(z)] + g^\phi_{\text{rep}} + g^\phi_{\text{cor}}. \tag{8}$$

$$g^\phi_{\text{rep}} = \nabla_z \log p(x_i|z) \nabla_\phi h(\epsilon, \phi)$$

$$g^\phi_{\text{cor}} = \log p(x_i|z) \nabla_\phi \log \frac{q(h(\epsilon, \phi))}{r(h(\epsilon, \phi))}$$

To understand the step from Equation 7 to Equation 8, we refer to the derivation by Naesseth et al. (2017),

$$\begin{aligned}
&\mathbb{E}_{\pi(\epsilon;\theta)}[f(h(\epsilon, \phi))] = \int s(\epsilon) \nabla_\phi \left(f(h(\epsilon, \phi)) \frac{q(h(\epsilon, \phi); \phi)}{r(h(\epsilon, \phi); \phi)} \right) d\epsilon = \\
&\int s(\epsilon) \frac{q(h(\epsilon, \phi); \phi)}{r(h(\epsilon, \phi); \phi)} \nabla_\phi f(h(\epsilon, \phi)) d\epsilon + \int s(\epsilon) f(h(\epsilon, \phi)) \nabla_\phi \left(\frac{q(h(\epsilon, \phi); \phi)}{r(h(\epsilon, \phi); \phi)} \right) d\epsilon = \\
&\mathbb{E}_{\pi(\epsilon;\theta)} \left[\nabla_\phi f(h(\epsilon, \phi))\right] + \mathbb{E}_{\pi(\epsilon;\theta)} \left[f(h(\epsilon, \phi)) \nabla_\phi \log \frac{q(h(\epsilon, \phi); \phi)}{r(h(\epsilon, \phi); \phi)} \right],
\end{aligned} \tag{9}$$

where in the last step the log-derivative trick was used and $f(h(\epsilon, \phi))$ in our case corresponds to $\log p(x_i|h(\epsilon, \phi))$.

4.3. The KL-Divergence

The KL-divergence can be calculated analytically, meaning that we do not have to use any samples and thus do not need to take care of reparameterization in the gradient back propagation for this term. $p(z)$ is a Dirichlet with prior α^0, whereas $q(z|x)$ is the Dirichlet parameterized by α as given by the variational network parameterized by ϕ.

$$\begin{aligned}
KL[q(z|x) \,\|\, p(z)] = &\log(\Gamma(\sum_k \alpha_k)) - \log(\Gamma(\sum_k \alpha^0_k)) + \\
&\sum_k \log \Gamma(\alpha^0_k) - \sum_k \log \Gamma(\alpha_k) + \sum_k (\alpha_k - \alpha^0_k)(\Psi(\alpha) - \Psi(\sum_k \alpha_k)),
\end{aligned}$$

where Ψ is the digamma function.

In our experiments we found that the analytical calculation of the KL-divergence, even though it is theoretically more appealing, does not lead to optimal results. We therefore resort to using a simple sampling approximation of the KL-divergence by calculating the KL-divergence as $KL[q(z|x) \,\|\, p(z)] = q(z|x) \log \frac{q(z|x)}{p(z)}$. Note that test results are calculated using the analytical KL-divergence to enable a fair comparison. Sampling the KL-divergence leads to more stable gradients, however, it introduces a second correction term for the gradient of the encoder network, since we are now using the rejection sampler again to produce z.

We therefore rewrite the ELBO as

$$\mathcal{L}(\theta, \phi; x_i) = -KL[q(z|x_i) \,\|\, p(z)] + \mathbb{E}_{q(z|x_i)}[\log p(x_i|z)] = \mathbb{E}_{\pi(\epsilon;\phi)}\left[\log \frac{p(h_\Gamma(\epsilon,\phi))}{q(h_\Gamma(\epsilon,\phi)|x)}\right] + \mathbb{E}_{\pi(\epsilon;\phi)}[\log p(x_i|h_\Gamma(\epsilon,\phi))].$$

The gradient of the KL-divergence is now given as

$$\nabla_\phi KL[q(h_\Gamma(\epsilon,\phi))|x) \,\|\, p(h_\Gamma(\epsilon,\phi))] = \nabla_\phi \log \frac{p(h_\Gamma(\epsilon,\phi))}{q(h_\Gamma(\epsilon,\phi)|x)} + g^\phi_{\text{kl-cor}}, \tag{10}$$

where $g^\phi_{\text{kl-cor}}$ is defined as

$$g^\phi_{\text{kl-cor}} = \log \frac{p(h_\Gamma(\epsilon,\phi))}{q(h_\Gamma(\epsilon,\phi)|x)} \nabla_\phi \log \frac{q(h_\Gamma(\epsilon,\phi))}{r(h_\Gamma(\epsilon,\phi))}.$$

The log fraction in this correction term is the same as the one for g^ϕ_{cor}, however, it is weighted by the KL-divergence instead of the reconstruction probability. The derivation is again done using Equation 9 except now $f(h(\epsilon,\phi))$ corresponds to $\log \frac{p(z)}{q(h(\epsilon,\phi)|x)}$. Note, that the two terms in Equation 10 correspond to a reparameterization part and a score function part. If the proposal is equal to the distribution and always accepted this reduces to the reparameterization gradient, whereas if the proposal is always rejected, this reduces to the score function gradient, also known as REINFORCE.[1]

4.4. Neural Network Architecture

The architecture of the neural network for the DVAE variational autoencoder is illustrated in Figure 2.

1. Encoder: The input is transformed using a RELU-layer with dropout. The result is linearly transformed and batch normalization is applied. Batch normalization is crucial for the convergence of the model. VAEs are prone to so-called component collapsing, a well-known problem (Bowman et al., 2016). Using batch normalization in combination with a relatively high learning rate (10^{-2}–10^{-4}), helps to avoid component collapsing and leads to a fast convergence (Srivastava and Sutton, 2017). A softplus transformation is applied because the Dirichlet parameter needs to be positive. Furthermore, a minimum value of 0.00001 is used for the Dirichlet parameter α which is necessary for stability of the gradients. The latent variables z are sampled from the Dirichlet distribution.

1. https://casmls.github.io/general/2017/04/25/rsvi.html

2. Decoder: The decoder is structured as follows: The latent variables z are transformed linearly. Again, batch normalization is applied. The final result is obtained through a log-softmax transformation. Note that the decoder weights are not restricted to be on the probability simplex as would be necessary if a Dirichlet prior was put on the topic-word distributions. This lifts the restriction of the mixture model as proposed by Srivastava and Sutton (2017) (The same was done in the NVDM model by Miao et al. (2016), although it was not explicitly discussed.). This is why in our experiments we compare to the ProdLDA model of Srivastava and Sutton (2017) and not to AVITM, which puts the simplex constraint on the decoder weights. Nevertheless, we can use the decoder weights to infer our topics since it is sufficient to rank the words according to the respective weights for each topic. The weights do not have be given a probabilistic interpretation as in LDA models. The words with the largest weights for each latent topic, are usually representative for that topic. It was shown by Srivastava and Sutton (2017) and is confirmed in our experiments that the non-mixture version yields lower perplexity but higher topic coherence. The reason for this is suspected to be the lifting of the mixture model contraint which makes the model more expressive.

4.5. Component Collapsing

As mentioned in the previous section, VAEs often suffer from the well-known issue of component collapsing (Bowman et al., 2016). This is the problem of a local minimum that is usually encountered early on in the training process and which the model cannot escape as the training continues. It occurs because it is often easier for the model to minimize the KL-divergence than to minimize the reconstruction error. This leads to parameters that are equal to the prior and thus, many or all topics are the same. This means that there are a lot of redundant topics, but the perplexity is relatively low. To prevent this from happening, a common method is annealing of the KL-divergence (Bowman et al., 2016) to slowly introduce it into the loss function. Other methods include batch normalization and dropout (Srivastava and Sutton, 2017, 2018). In our experiments, we found that batch normalization had the largest effect. None of the implemented Dirichlet autoencoders learn meaningful topics without batch normalization. NVDM does not employ batch normalization, but has to use a lower learning rate to prevent component collapsing.

5. Experiments

Here, we first describe the experimental settings, the data sets, baseline methods and the evaluation measures. Finally we describe the experimental results on the different performance measures.

5.1. Experimental Settings

The number of neurons was set to 100 for all hidden layers in the neural network models. We used a single sample for all methods. A batch size of 200 was used for the training of all

models. Training was monitored on a validation set with early stopping and a look-ahead of 30 iterations. We used the Adam optimizer for training. For NVDM each training iteration spans ten iterations for the decoder and ten iterations for the encoder. DVAE and ProdLDA optimize decoder and encoder jointly. Since we noticed that DVAE has no difference in performance between using the correction part of the gradient when shape augmentation is used, we omitted that part in the experiments to save unnecessary computation. Similar to Srivastava and Sutton (2017), we employ batch normalization to improve the convergence of our model. The learning rate is optimized for all neural network methods using Bayesian optimization.

5.2. Data Sets

- `20news`: We used the same version of this data set as Srivastava and Sutton (2017). It is a widely used text data set consisting of news articles that are grouped into 20 different categories. It has 11,000 training instances and a 2,000 word vocabulary.
- `NIPS`: The NIPS data set was retrieved in a preprocessed format from the UCI Machine Learning Repository (Perrone et al., 2017). It consist of 5,812 documents and has 11,463 features. It contains text data from NIPS conference papers published between 1987 and 2015. Compared to the other data sets, individual documents are large. 1,000 documents were separated as a test set.
- `KOS`: The 3,430 blog entries of this data set were originally extracted from `http://www.dailykos.com/`, the data set is available in the UCI Machine Learning Repository `https://archive.ics.uci.edu/ml/datasets/Bag+of+Words`. The number of features is 6,906.
- `Rcv1`:[2] This corpus contains 810,000 documents of Reuters news stories. We separated 10,000 documents as a testset and pruned the vocabulary to 10,000 words after stopword removal following Miao et al. (2016).

5.3. Methods

5.3.1. Dirichlet Autoencoders

We compare to four different autoencoders that all have exactly the same architecture and hyperparameters as DVAE. They are only different with respect to the modeling of the Dirichlet prior on the latent variables.

1. ProdLDA: This method uses a Laplace approximation to approximate the Dirichlet prior. We use the implementation provided by the authors.[3]
2. Inverse CDF (Joo et al., 2019): This method builds on previous work by Knowles (2015). If $X \sim \Gamma(\alpha, \beta)$ and $F(x; \alpha, \beta)$ is a CDF of the random variable X, the inverse CDF can be approximated by $F^{-1}(u; \alpha, \beta) \approx \beta^{-1}(u\alpha\Gamma(\alpha))^{1/\alpha}$, where $u \sim \text{Uniform}(0,1)$. The approximated inverse CDF can then be used as a reparameterization function.

2. `http://trec.nist.gov/data/reuters/reuters.html`
3. `https://github.com/akashgit/autoencoding_vi_for_topic_models`

3. Implicit reparameterization gradients (Figurnov et al., 2018): This method introduces a trick to avoid inverting the CDF by differentiating it instead. It is integrated into the tensorflow library and can be used out-of-the-box.
4. Weibull autoencoder (Zhang et al., 2018): This method uses the Weibull distribution instead of the Gamma distribution because it has a similar PDF and the Weibull distribution can be reparameterized easily. The Weibull PDF is given as

$$P(x|k, \lambda_{\text{weibull}}) = \frac{k}{\lambda_{\text{weibull}}^k} x^{k-1} \exp\left(x/\lambda_{\text{weibull}}\right)^k.$$

The reparameterization for $x \sim Weibull(k, \lambda_{\text{weibull}})$ is

$$x = \lambda(-\ln(1-\epsilon))^{1/k}, \epsilon \sim \text{Uniform}(0,1),$$

and the KL-divergence from the gamma distribution is analytically computed as

$$KL[Weibull(k, \lambda_{\text{weibull}}) \,\|\, Gamma(\alpha, \beta)] =$$
$$\alpha \ln \lambda_{\text{weibull}} - \frac{\gamma\alpha}{k} - \ln k - \beta\lambda_{\text{weibull}}\Gamma\left(1 + \frac{1}{k}\right) + \gamma + 1 + \alpha \ln \beta - \ln \Gamma(\alpha).$$

The parameters λ_{weibull} and k are computed through the neural network as

$$\lambda_{\text{weibull}} = \log(1 + \exp(b)), k = \log(1 + \exp(c)),$$

where $b = l_2(\lambda)$ and $c = l_3(\lambda)$, where l_2 and l_3 are linear transformations of λ which is the same as in Equation 4.

Additionally, k is restricted to be greater or equal to 0.1 because for values of k that are lower than that the method becomes unstable.

5.3.2. Other Comparison Methods

Additionally, we compare to two other algorithms:

1. NVDM: This is the Gaussian VAE topic model. We used the code provided by the authors.[4]
2. SCVB: As a representative of the original LDA models, we compare to SCVB (Foulds et al., 2013), a stochastic variant of collapsed variational Bayes. This method is trained online using minibatches in the same way as the neural networks. Therefore the method is applicable to large data sets. We used our own Java reimplementation.

4. `https://github.com/ysmiao/nvdm`

5.4. Evaluation

All models are evaluated on three measures, perplexity, topic redundancy and topic coherence. The per-word perplexity is calculated from the ELBO as $\exp(-\frac{1}{N_w}\sum_d^D \log p(d))$, where N_w is the number of words in the corpus. We use the analytical KL-divergence for all models to calculate the perplexity to allow a fair comparison.

The topic coherence is calculated following Srivastava and Sutton (2017) using the normalized pointwise mutual information (NPMI), averaged over all pairs of words of all topics, where the NPMI is set to zero for the case that a word pair does not occur together. We calculate the topic coherence on the test set using the 10 first words of each topic, i.e. the 10 words with the highest probability for that topic. The NPMI for topic t is given as follows:

$$\mathrm{NPMI}(t) = \sum_{j=2}^{N}\sum_{i=1}^{j-1} \frac{\log \frac{P(w_j,w_i)}{P(w_i)P(w_j)}}{-\log P(w_i,w_j)},$$

where N is the number of words in topic t, w_i is the ith word of topic t and $P(w_i, w_j)$ is the probability of words w_i and w_j occurring together in the test set, which is approximated by counting the number of documents where both words appear together[5] and dividing the result by the total number of documents.

The topic redundancy corresponds to the average probability of each word to occur in one of the other topics of the same model. The redundancy for topic k is thus given as

$$R(k) = \frac{1}{K-1}\sum_{i=1}^{N}\sum_{j\neq k} P(w_{ik}, j),$$

where $P(w_{ik}, j)$ is one if the ith word of topic k, w_{ik}, occurs in topic j and otherwise zero, and $K-1$ is the number of topics excluding the current topic.

5.5. Experimental Results

5.5.1. Perplexity

We compare our models, DVAE and DVAE Sparse, to the Laplace approximation *ProdLDA*, the Gaussian VAE *NVDM* and the online LDA model *SCVB* in Tables 1 and 2. Additional results are in the Appendix in Tables 4 and 5. The overall best perplexity is achieved by NVDM as can be seen from the average ranks in each table, whereas the Weibull VAE has the worst perplexity, closely followed by DVAE. Implicit gradients VAE and inverse CDF VAE are worse than NVDM and DVAE Sparse, but a little bit better than DVAE and Weibull VAE. The reason for the high perplexity of DVAE is the KL-divergence part of the ELBO (Equation 6). The KL-divergence between two Dirichlet distributions can have extreme values as shown in Figure 3. If the Dirichlet parameter is close to zero, the KL-divergence and therefore also the perplexity is high. As Tables 1 and 2 show, the DVAE Sparse model successfully circumvents this problem by decoupling sparsity and smoothness. The perplexity of DVAE Sparse, while not as good as that of NVDM, is mostly better or at least comparable to that of the ProdLDA model.

5. We do not use a window size but always consider the whole document as a unit.

data set	DVAE	DVAE Sp.	Impl.	Inv.	Weibull	ProdLDA	NVDM	SCVB
	Perplexity							
				50 topics				
KOS	7872	2408	2894	2598	7795	2811	**2097**	2365
NIPS	2779	2197	2327	2141	3.55E+52	2303	**1933**	3973
20news	5009	1066	1842	1533	5209	1128	**813**	817
Rcv1	5885	1464	1635	1547	2344	2265	**907**	2034
				200 topics				
KOS	2.59E+05	2397	3238	3233	3.11E+04	3161	**2216**	2222
NIPS	3110	2152	2250	2135	8.50E+167	2532	**1957**	5787
20news	4.93E+05	1075	3021	3033	1.82E+08	1271	**875**	951
Rcv1	8.66E+04	1614	1471	1.07E+10	4426	2559	**891**	2779
av. rank	7.125	2.875	4.750	4.375	7.375	4.500	**1.000**	4.000
	Topic Coherence							
				50 topics				
KOS	0.194	0.050	0.131	0.130	0.067	**0.218**	0.070	0.145
NIPS	**0.313**	0.295	0.276	0.224	0.167	0.293	0.070	0.116
20news	**0.324**	0.248	0.322	0.304	0.271	0.223	0.140	0.244
Rcv1	**0.381**	0.293	0.340	0.162	0.216	0.140	0.110	0.239
				200 topics				
KOS	**0.212**	0.072	0.118	0.076	0.086	0.071	0.060	0.107
NIPS	**0.280**	0.234	0.227	0.208	0.176	0.277	0.060	0.105
20news	**0.307**	0.174	0.298	0.281	0.258	0.150	0.140	0.204
Rcv1	**0.373**	0.115	0.237	0.138	0.069	0.370	0.051	0.263
av. rank	1.125	4.875	3.000	4.625	5.375	4.500	7.750	4.750
	Topic Redundancy							
				50 topics				
KOS	0.017	0.014	0.024	**0.011**	0.020	0.040	0.012	0.110
NIPS	0.016	0.013	0.008	**0.003**	0.123	0.019	0.003	0.088
20news	0.026	0.025	0.022	0.018	0.039	0.020	**0.007**	0.037
Rcv1	0.009	0.269	0.002	**0.001**	0.095	0.017	0.002	0.039
				200 topics				
KOS	0.026	0.013	0.016	0.010	0.020	**0.006**	0.010	0.088
NIPS	0.021	0.010	0.012	0.008	0.022	0.018	**0.003**	0.058
20news	0.030	0.016	0.030	0.027	0.044	0.013	**0.011**	0.046
Rcv1	0.007	0.017	0.104	0.817	0.668	0.253	**0.004**	0.024
av. rank	4.875	4.125	4.375	2.750	6.875	4.375	1.625	7.000

Table 1: Perplexity, topic coherence and topic redundancy for **Dirichlet prior 0.02**.

data set	DVAE	DVAE Sp.	Impl.	Inv.	Weibull	ProdLDA	NVDM	SCVB
	Perplexity							
				50 topics				
KOS	4190	2314	2399	2242	4274	2555	**2097**	2579
NIPS	2686	2185	2263	2064	1.83E+141	2250	**1933**	4290
20news	2158	1080	2505	2443	1546	1041	**813**	903
Rcv1	3449	1646	1506	1355	1815	2165	**907**	1986
				200 topics				
KOS	2.06E+04	2758	2713	3807	1.21E+12	3127	**2216**	2500
NIPS	2744	2072	2076	**1936**	1.70E+249	2422	1957	5911
20news	1.71E+04	1100	2257	1678	1.13E+09	1294	875	**874**
Rcv1	1.01E+04	1196	1356	1675	2463	2146	**891**	3317
av. rank	6.875	3.250	4.500	3.625	7.000	4.750	1.250	4.750
	Topic Coherence							
				50 topics				
KOS	0.185	0.117	0.134	0.127	0.051	**0.220**	0.070	0.137
NIPS	**0.313**	0.284	0.284	0.233	0.306	0.312	0.070	0.111
20news	**0.337**	0.248	0.318	0.258	0.243	0.286	0.140	0.234
Rcv1	**0.389**	0.345	0.325	0.083	0.311	0.194	0.110	0.249
				200 topics				
KOS	**0.204**	0.104	0.113	0.137	0.133	0.140	0.060	0.110
NIPS	**0.281**	0.243	0.240	0.208	0.186	0.224	0.060	0.101
20news	**0.303**	0.222	0.302	0.270	0.264	0.216	0.140	0.188
Rcv1	**0.364**	0.121	0.321	0.187	0.147	0.200	0.051	0.257
av. rank	1.125	4.875	3.125	4.875	5.125	3.500	7.750	5.625
	Topic Redundancy							
				50 topics				
KOS	0.021	0.024	0.010	**0.007**	0.016	0.055	0.012	0.111
NIPS	0.014	0.014	0.009	**0.002**	0.075	0.018	0.003	0.086
20news	0.027	0.035	0.033	0.063	0.032	0.027	**0.007**	0.035
Rcv1	0.007	0.011	0.005	0.004	0.012	0.023	**0.002**	0.036
				200 topics				
KOS	0.030	0.033	0.019	0.031	0.032	0.014	**0.010**	0.084
NIPS	0.018	0.010	0.009	0.003	0.033	0.014	**0.003**	0.062
20news	0.024	0.033	0.039	0.034	0.050	0.022	**0.011**	0.065
Rcv1	0.008	0.023	0.017	0.023	0.262	0.019	**0.004**	0.021
av. rank	3.750	5.500	3.500	3.750	6.125	4.500	1.375	7.500

Table 2: Perplexity, topic coherence and topic redundancy for **Dirichlet prior 0.1**.

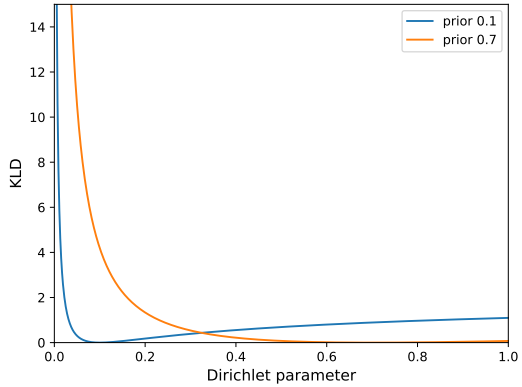
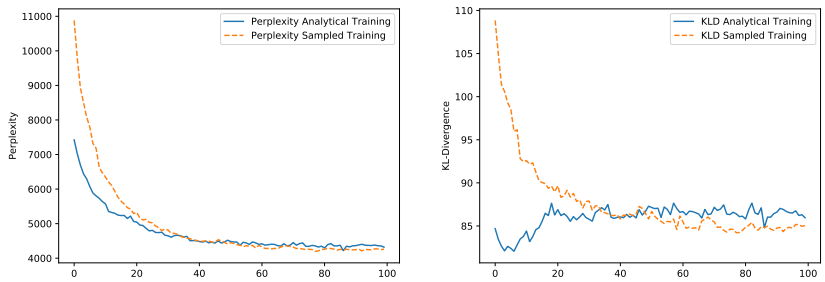


Figure 3: This plot shows the KL-divergence between two Dirichlet distributions with identical parameters, where only one element of one parameter vector is varied between zero and one. All other elements are fixed to 0.1 (blue line) or 0.7 (orange line). We can see that the KL-divergence diverges for small parameter values.

5.5.2. Topic Coherence

In terms of topic coherence (see Tables 1 and 2, middle section, as well as additional results in the Appendix), DVAE has consistently high values as compared to all other methods with the best average rank for all settings of the Dirichlet parameter. A slight decrease in the average rank (1.125,1.125,1.250,1.750) can be observed for increasing values of the Dirichlet parameter which indicates that DVAE is especially good at modeling the sparseness implied by small parameter settings. DVAE Sparse is slightly worse than ProdLDA, but generally still better than NVDM and SCVB in terms of the average rank in topic coherence. This shows a trade-off between perplexity and topic coherence which can be influenced through the sparseness of the model. While DVAE Sparse has a far better perplexity than DVAE, its topic coherence is generally worse. While increased sparseness may thus lead to a lower topic coherence, this is not the only indicator for the topic quality (see Section 5.5.3).

Taking a closer look at the topic outputs, it is not always the case that a higher topic coherence corresponds to better topics. In Table 3 we listed all the topics that ProdLDA learned on the KOS data set that have the words “iraq” and “war” in the most frequent words. This concerns ten out of fifty topics. ProdLDA often learns topics of relatively high quality that are redundant. This way, it is able to achieve a high topic coherence, but the diversity of the topics is not accounted for by this measure. DVAE in contrast learned only one topic containing the words “iraq” and “war” (see also Table 3), whereas DVAE Sparse only learned five topics containing these two terms. While NVDM only learned a



(a) Perplexity (b) KL-divergence

Figure 4: This figure compares the performance of DVAE trained with the analytical KL-divergence with the performance of DVAE trained with the sampled KL-divergence on the 20news data set using a Dirichlet prior of 0.02. Reported is the perplexity using the analytical KL-divergence as well as the analytical KL-divergence. We can see that the final perplexity and KL-divergence for the sampled version are lower even though the convergence is slower in the beginning.

single topic on the Iraq war, the quality of this topic is subjectively bad as none of the other words are closely related to the war. Nevertheless, ProdLDA has a topic coherence of 0.22 and DVAE and DVAE Sparse only reach 0.19 and 0.12, respectively. This shows that, while topic coherence is considered to be closely related to human judgement about topic quality (Lau et al., 2014), it is not a good indicator of topic quality in the case where topic redundancy is a problem.

Comparing the implicit gradients VAE, inverse CDF VAE and Weibull VAE in term of topic coherence, Tables 1 and 2 show that the implicit gradients method achieves the second best topic coherence overall and is only worse than DVAE. The inverse CDF and Weibull methods however, show a topic coherence that is in general comparable to the DVAE Sparse method. Judging from the topic coherence results, the approximations that are made by the Weibull and inverse CDF autoencoders are not as good as the implicit gradients and RSVI methods. We can therefore conclude that the RSVI and implicit gradients methods are preferable for Dirichlet parameter settings < 1 as are typical in topic modeling. The difference between the DVAE and the implicit gradients VAE method is minimal according to Figurnov et al. (2018), however, the used Dirichlet parameter is not reported since it was optimized during training. We therefore hypothesize that our improved performance is due to the sampled KL-divergence as well as a better approximation for low settings of the Dirichlet parameter.

5.5.3. Topic Redundancy

To quantify the redundancy of the topics we also reported the average probability of each word to occur in one of the other topics of the same model. The results reported in Tables 1

	DVAE
1	iraq hussein war destruction weapons saddam bush bin mass ladens
	DVAE Sparse
1	iraq war iraqi soldiers military sunni pleasure forces insurgency kurdish
2	war iraq baghdad juan cole soldiers helicopter iraqi killed troops
3	occupation license war iraq reid freeway iraqi ministries forces idiot
4	iraq bush war president administration billion house fiscal bushs cost
5	iraq weapons administration intelligence war bush saddam united clarke destruction
	ProdLDA
1	war iraq iraqi military baghdad juan administration bush mortar gulf
2	war iraq baghdad juan iraqi cole moqtada insurgency sunni soldiers
3	war fatalities authorities cow uniform barnes iraq administration ashamed mps
4	war iraq jury emotional assertions ice melanie film rationing studies
5	war iraq baghdad iraqi uniform prisoners military occupation saddam noble
6	iraq war iraqi moqtada kurdish baghdad shiite sunni kurds shia
7	bush patriot war aclu counterterrorism oversight iraq provisions allday sunset
8	iraq war interrogation pentagon apples ghraib abu guantanamo intelligence weapons
9	iraq war iraqi juan sistani baghdad kufa duration forces cole
10	bush gwb contracts ownership london defence iraq amazingly war infamous
	NVDM
1	senate chemical war races voted iraq barbero return qaqaa materiel

Table 3: 10 out of 50 topics for ProdLDA are about the Iraq war on the KOS data set with a Dirichlet prior of 0.1. The topic coherence is 0.22 for this run, since topic coherence does not account for the diversity of topics. The topic coherence for DVAE Sparse is only 0.12, however, the topic quality is higher as can be seen from the topic redundancy score and qualitatively from this example. For DVAE, only one topic is about the Iraq war and the topic redundancy score is lower than that of ProdLDA and DVAE Sparse while the topic coherence is still high.

and 2 show that the lowest redundancy scores were achieved by NVDM, however, NVDM has a far worse topic coherence. In our experience the overall topic quality of all other models was mostly better than that of NVDM. If the topic coherence is low, however, the redundancy score is not that meaningful, since a low redundancy can also be expected from a model that assigns words to topics randomly. If a model achieves a high topic coherence as can be seen in Table 5 for the implicit gradients method with the Rcv1 data set and 200 topics, this can still be associated with a very hight topic redundancy (0.45 in this case) which points to component collapsing and devalues the high topic coherence score. DVAE almost always has a low topic redundancy except for a prior of 0.6 and 200 topics on the KOS data set. The topic coherence is also low in this case. We can therefore conclude that topic redundancy is an important measure if the topic coherence is also high. This is because it helps to differentiate models with a diverse set of high quality topics and models where component collapsing led to a single topic being repeated several times.

6. Discussion

DVAE, DVAE Sparse, implicit gradients VAE, inverse CDF VAE and Weibull VAE are most closely related to the ProdLDA model. All of these models can be trained with relatively high learning rates (10^{-3}–10^{-2}) as compared to NVDM, which is trained with much lower learning rates (around 10^{-5}). This speeds up the training process considerably. The efficiency of ProdLDA, DVAE and DVAE Sparse and the other Dirichlet autoencoders is comparable as long as the correction term is disregarded for the DVAE models. This can safely be done as long as the shape augmentation parameter is chosen high enough (≥ 5). The training efficiency might also be affected by other architectural decisions that warrant further investigation in the future. NVDM trains the encoder and decoder networks separately, whereas DVAE and ProdLDA update both networks at once. DVAE and ProdLDA use dropout and batch normalization, which enable the high learning rates. All these details need to be taken into consideration and their potential influence will be investigated in future work.

All models with a Dirichlet latent parameter vector are characterized by a relatively high topic coherence as compared to the Gaussian model NVDM. This shows that it is worth considering to use the Dirichlet distribution in cases where interpretable topics are important. Nevertheless, topic coherence alone is not a good indicator of topic quality and should be viewed in conjunction with our newly introduced measure of topic redundancy. This is especially important in VAE topic models due to the problem of component collapsing, which is more prevalent in VAEs than in other LDA topic models.

Comparing our two models DVAE and DVAE Sparse, DVAE clearly leads to topics of higher quality. The perplexity, however, is bad in comparison. According to our analysis this is due to the KL-divergence term in the ELBO that leads to high values when the parameter is predicted to be close to zero. By modeling the sparseness in a separate vector, this problem is alleviated, leading to lower perplexities. DVAE Sparse therefore serves as a test to show that our original hypothesis is correct, namely that the coupling of sparsity and smoothness is responsible for the high perplexity scores. Furthermore, this shows that perplexity scores are not always meaningful when comparing different topic models, as in our experiments the topic quality has nothing to do with the perplexity a model achieves.

The KL-divergence can be sampled or calculated analytically. In our experiments we chose to sample the KL-divergence during training. The comparison methods all use analytical KL-divergence terms. For the KL-divergence of the Dirichlet distribution it can make a big difference whether the KL-divergence is sampled or not. In the case where the parameter is close to zero, the analytical KL-divergence takes on very large values (see Figure 3), whereas in this case the sampled KL-divergence is often negative in fact. This can lead to much lower perplexity scores when the sampled KL-divergence is used. Therefore, we chose to calculate the testing perplexity using the analytical KL-divergence. (In the other case, when the parameter is larger, the analytical KL-divergence is often lower than the sampled KL-divergence, although the difference is not as big as in the first case.) This stability issue has a large influence on the perplexity and our sparse method avoids it to a certain extent, but not completely. A related method avoids this by preconditioning the gradient in a Hamiltonian MCMC method with the inverse of the Fisher information matrix (Cong et al., 2017; Patterson and Teh, 2013). According to Cong et al. (2017), this makes

the stability problem completely disappear. How such a method could be applied to the variational autoencoder however, remains open.

Our experiments show that DVAE is superior to other state-of-the-art topic models in terms of topic coherence. This further underlines the argument by Srivastava and Sutton (2017) that the sparsity induced by the Dirichlet indeed leads to higher topic coherence. The comparison with other Dirichlet autoencoders shows that while our version of the RSVI autoencoder works best, the implicit gradients VAE is the most competitive method, whereas the inverse CDF and Weibull autoencoders are even worse than the Laplace approximation of ProdLDA in terms of topic coherence.

We discussed the trade-off between sparsity and smoothness in Section 4.1. This trade-off can be influenced using the DVAE Sparse model with the additional Bernoulli variable. Furthermore, the sampling of the KL-divergence is an important factor for increasing the topic coherence. As shown in Figure 4, while the convergence of perplexity and KL-divergence seems to be slower in the beginning, this may lead to a lower perplexity and KL-divergence in the final stages of training. For the shown comparison the final topic coherence score was 0.33 for the training method with the analytical KL-divergence, whereas training with the sampled KL-divergence led to a topic coherence of 0.35 after 100 iterations of training. This effect is similar to the effect of batch normalization as reported by Srivastava and Sutton (2018) who did a similar experiment. They show that batch normalization slows down the convergence of the KL-divergence in the beginning of training. This reduces the component collapsing and enables the learning of more meaningful topics.

When talking about a trade-off, this implies that there is a continuum of models between extreme sparseness and extreme smoothness. This can in principle be explored using our DVAE Sparse model. While at the moment, the parameter for our Bernoulli variables is $p = 0.5$, this parameter could also be set to other values and thereby increase or decrease the sparseness of the model. On the one hand, a value of $p = 0$ would mean that the vector b will always consist of only ones. This would reduce the model to the original DVAE model, which is a smooth model. On the other hand, for $p = 1$ the vector b would consist of only zeros, which entails maximal sparseness and only topics with zero values, preventing any topics from being learned. Intermediary values can be used to adjust the degree of sparseness and interpolate between the sparse and the smooth model.

7. Conclusion

We have introduced a variational autoencoder with a Dirichlet prior called DVAE. A trade-off between sparsity and smoothness was identified, leading to high perplexity as well as topic coherence scores. To test our hypothesis about the reason for the high perplexity scores, a new model called DVAE Sparse was introduced. DVAE Sparse decouples sparsity and smoothness and thus enables lower perplexity scores on the one hand while on the other hand the topic coherence is still competitive. The topic quality was also measured in terms of topic redundancy which is a good indicator for component collapsing as opposed to topic coherence. Overall, this shows that, despite the bad perplexity scores, DVAE is an effective model. An extensive comparison with all known training methods for Dirichlet VAEs shows that our DVAE using the RSVI method achieves the highest topic coherence scores.

To conclude, we think that this work is an important step towards making VAEs with Dirichlet priors more widely used. In the future we would like to extend our model to work with hierarchical Dirichlet priors.

Acknowledgements

We thank the anonymous reviewers for their helpful comments.

Appendix A. Additional Results

data set	DVAE	DVAE Sp.	Impl.	Inv.	Weibull	ProdLDA	NVDM	SCVB
	Perplexity							
				50 topics				
KOS	2858	2322	2232	**2056**	2923	2537	2097	2501
NIPS	2666	2142	2223	4858	1.46E+26	2239	**1933**	8137
20news	1345	1030	1678	1484	1212	1076	**813**	961
Rcv1	2542	1234	1458	1480	1397	1788	**907**	1990
				200 topics				
KOS	4503	2443	2611	2395	4839	3739	**2216**	2404
NIPS	2551	2043	1983	2155	inf	2313	**1957**	6627
20news	2869	1123	1734	2120	4.66E+04	1496	**875**	925
Rcv1	2913	1052	1337	1519	1613	2206	**891**	4573
av. rank	6.625	2.875	4.125	4.375	6.625	5.125	1.125	5.125
	Topic Coherence							
				50 topics				
KOS	0.202	0.151	0.132	0.135	0.146	**0.232**	0.070	0.147
NIPS	**0.330**	0.263	0.284	0.144	0.120	0.281	0.070	0.093
20news	**0.342**	0.265	0.280	0.276	0.243	0.291	0.140	0.218
Rcv1	**0.402**	0.121	0.326	0.150	0.308	0.094	0.110	0.246
				200 topics				
KOS	**0.218**	0.070	0.143	0.098	0.152	0.124	0.060	0.114
NIPS	**0.295**	0.213	0.236	0.244	0.088	0.260	0.060	0.074
20news	**0.334**	0.197	0.300	0.260	0.215	0.297	0.140	0.183
Rcv1	0.286	0.113	**0.386**	0.138	0.256	0.078	0.051	0.227
av. rank	1.250	5.250	3.000	4.750	4.500	3.750	7.875	5.625
	Topic Redundancy							
				50 topics				
KOS	0.024	0.015	0.015	**0.007**	0.032	0.051	0.012	0.094
NIPS	0.018	0.012	0.010	0.030	0.068	0.014	**0.003**	0.095
20news	0.029	0.028	0.073	0.048	0.033	0.023	**0.007**	0.041
Rcv1	0.007	**0.001**	0.005	0.009	0.002	0.033	0.002	0.034
				200 topics				
KOS	0.029	**0.006**	0.051	0.047	0.013	0.081	0.010	0.168
NIPS	0.017	0.010	0.005	0.015	0.029	0.016	**0.003**	0.096
20news	0.042	0.025	0.042	0.036	0.034	0.043	**0.011**	0.151
Rcv1	0.010	**0.002**	0.098	0.015	0.025	0.024	0.004	0.025
av. rank	4.625	2.125	5.000	4.625	4.875	5.500	1.625	7.625

Table 4: Perplexity, topic coherence and topic redundancy for **Dirichlet prior 0.3**.

data set	DVAE	DVAE Sp.	Impl.	Inv.	Weibull	ProdLDA	NVDM	SCVB
				Perplexity				
				50 topics				
KOS	2362	2259	2206	**1988**	2744	2623	2097	2515
NIPS	2626	2115	2219	2005	3.60E+47	2247	**1933**	4806
20news	1053	954	1531	1182	1369	1139	**813**	853
Rcv1	2154	1191	1460	7.09E+201	1294	1765	**907**	2525
				200 topics				
KOS	2943	**2116**	2378	2688	4821	4263	2216	2171
NIPS	2450	1981	1971	**1848**	7.73E+112	2261	1957	1.13E+04
20news	1035	1065	2073	1357	4997	2104	**875**	943
Rcv1	2681	971	1942	7.11E+201	1433	2393	**891**	3295
av. rank	5.250	2.875	4.500	4.500	6.625	5.750	1.500	5.000
				Topic Coherence				
				50 topics				
KOS	**0.205**	0.192	0.144	0.128	0.057	0.160	0.070	0.139
NIPS	**0.313**	0.254	0.294	0.261	0.287	0.312	0.070	0.093
20news	**0.371**	0.250	0.303	0.245	0.206	0.324	0.140	0.214
Rcv1	**0.425**	0.285	0.327	0.130	0.294	0.260	0.110	0.230
				200 topics				
KOS	0.110	0.081	0.153	**0.155**	0.141	0.137	0.060	0.123
NIPS	**0.296**	0.232	0.255	0.234	0.139	0.263	0.060	0.061
20news	**0.319**	0.235	0.285	0.261	0.192	0.313	0.140	0.180
Rcv1	0.291	0.186	**0.333**	0.165	0.274	0.077	0.051	0.196
av. rank	1.750	4.750	2.625	4.750	5.000	3.375	7.875	5.875
				Topic Redundancy				
				50 topics				
KOS	0.025	0.030	0.012	**0.006**	0.018	0.054	0.012	0.092
NIPS	0.015	0.012	0.010	0.004	0.081	0.020	**0.003**	0.105
20news	0.033	0.027	0.080	0.029	0.029	0.033	**0.007**	0.055
Rcv1	0.007	0.004	0.007	0.142	**0.002**	0.027	0.002	0.032
				200 topics				
KOS	0.216	0.012	0.019	0.042	0.023	0.070	**0.010**	0.253
NIPS	0.014	0.013	0.009	0.008	0.079	0.014	**0.003**	0.103
20news	0.039	0.025	0.044	0.044	0.026	0.035	**0.011**	0.231
Rcv1	0.046	**0.002**	0.453	0.223	0.009	0.013	0.004	0.056
av. rank	5.500	3.000	4.750	4.250	4.125	5.375	1.500	7.500

Table 5: Perplexity, topic coherence and topic redundancy for **Dirichlet prior 0.6**.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944937`.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.

Yulai Cong, Bo Chen, Hongwei Liu, and Mingyuan Zhou. Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 864–873, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 439–450, 2018.

James Foulds, Levi Boyles, Christopher DuBois, Padhraic Smyth, and Max Welling. Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 446–454, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7.

Thomas L Griffiths and Mark Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 101, pages 5228–5235. National Acad Sciences, 2004.

Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. Dirichlet variational autoencoder. *CoRR*, abs/1901.02739, 2019. URL `http://arxiv.org/abs/1901.02739`.

Diederik Kingma and Max Welling. Efficient gradient-based inference through transformations between Bayes nets and neural nets. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1782–1790, Bejing, China, 22–24 Jun 2014a. PMLR.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014b.

David A Knowles. Stochastic gradient variational bayes for gamma approximating distributions. *arXiv*, page 1509.01631, 2015. URL `https://arxiv.org/abs/1509.01631`.

Jey Han Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL*, pages 530–539. The Association for Computer Linguistics, 2014.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1727–1736, New York, New York, USA, 20–22 Jun 2016. PMLR.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering discrete latent topics with neural variational inference. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2410–2419, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Christian Naesseth, Francisco Ruiz, Scott Linderman, and David Blei. Reparameterization gradients through acceptance-rejection sampling algorithms. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 489–498, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

Eric Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Sam Patterson and Yee Whye Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3102–3110. Curran Associates, Inc., 2013.

Valerio Perrone, Paul A. Jenkins, Dario Spanò, and Yee Whye Teh. Poisson random fields for dynamic feature models. *Journal of Machine Learning Research*, 18(1):4626–4670, January 2017. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=3122009.3208008`.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Akash Srivastava and Charles A. Sutton. Variational inference in pachinko allocation machines. *CoRR*, abs/1804.07944, 2018. URL `http://arxiv.org/abs/1804.07944`.

Chong Wang and David M. Blei. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1982–1989. Curran Associates, Inc., 2009.

Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In *International Conference on Learning Representations*, 2018.

Journal of Machine Learning Research 20 (2019) 1-59 Submitted 9/18; Revised 7/19; Published 8/19

More Efficient Estimation for Logistic Regression with Optimal Subsamples

HaiYing Wang HAIYING.WANG@UCONN.EDU
Department of Statistics
University of Connecticut
Storrs, CT 06269, USA

Editor: Tong Zhang

Abstract

In this paper, we propose improved estimation method for logistic regression based on subsamples taken according the optimal subsampling probabilities developed in Wang et al. (2018). Both asymptotic results and numerical results show that the new estimator has a higher estimation efficiency. We also develop a new algorithm based on Poisson subsampling, which does not require to approximate the optimal subsampling probabilities all at once. This is computationally advantageous when available random-access memory is not enough to hold the full data. Interestingly, asymptotic distributions also show that Poisson subsampling produces a more efficient estimator if the sampling ratio, the ratio of the subsample size to the full data sample size, does not converge to zero. We also obtain the unconditional asymptotic distribution for the estimator based on Poisson subsampling. Pilot estimators are required to calculate subsampling probabilities and to correct biases in un-weighted estimators; interestingly, even if pilot estimators are inconsistent, the proposed method still produce consistent and asymptotically normal estimators.

Keywords: Asymptotic Distribution, Logistic Regression, Massive Data, Optimal Subsampling, Poisson Sampling.

1. Introduction

Extraordinary amounts of data that are collected offer unparalleled opportunities for advancing complicated scientific problems. However, the incredible sizes of big data bring new challenges for data analysis. A major challenge of big data analysis lies with the thirst for computing resources. Faced with this, subsampling has been widely used to reduce the computational burden, in which intended calculations are carried out on a subsample that is drawn from the full data, see Drineas et al. (2006a,b,c); Mahoney and Drineas (2009); Drineas et al. (2011); Mahoney (2011); Halko et al. (2011); Clarkson and Woodruff (2013); Kleiner et al. (2014); McWilliams et al. (2014); Yang et al. (2017), among others.

A key to success of a subsampling method is to specify nonuniform sampling probabilities so that more informative data points are sampled with higher probabilities. For this purpose, normalized statistical leverage scores or its variants are often used as subsampling probabilities in the context of linear regression, and this approach is termed *algorithmic leveraging* (Ma et al., 2015). It has demonstrated remarkable performance in better using of a fixed amount of computing power (Avron et al., 2010; Meng et al., 2014). Statistical leverage scores only contain information in the covariates and do not take into account the information contained in the observed responses. Wang et al. (2018) derived optimal subsampling probabilities that minimize the asymptotic mean squared error (MSE) of the subsampling-based estimator in the context of logistic regression. The optimal subsampling probabilities directly depend on both the covariates and the responses to take more informative subsamples. Wang et al. (2018) used a inverse probability weighted estimator based on the optimal subsample, where more informative data points are assigned smaller weights in the objective function. Thus, we can improve the estimation efficiency based on the optimal subsample by using a better weighting scheme.

In this paper, we propose more efficient estimators based on subsamples taken randomly according to the optimal subsampling probabilities. We will derive asymptotic distributions to show that asymptotic variance-covariance matrices of the new estimators are smaller, in Loewner ordering, than that of the weighted estimator in Wang et al. (2018). We also consider to use Poisson subsampling. Asymptotic distributions show that Poisson subsampling is more efficient in parameter estimation when the subsample size is proportional to the full data sample size. It is also computationally beneficial to use Poisson subsampling because there is no need to calculate and use subsampling probabilities for all data points simultaneously.

Before presenting the framework of the paper, we give a brief review of the emerging field of subsampling-based methods. For linear regression, Drineas et al. (2006d) developed a subsampling method and focused on finding influential data units for the least squares (LS) estimates. Drineas et al. (2011) developed an algorithm by processing the data with randomized Hadamard transform and then using uniform subsampling to approximate LS estimates. Drineas et al. (2012) developed an algorithm to approximate statistical leverage scores that are used for algorithmic leveraging. Yang et al. (2015) showed that using normalized square roots of statistical leverage scores as subsampling probabilities yields better approximation than using original statistical leverage scores, if they are very nonuniform. The aforementioned studies focused on developing algorithms for fast approximation of LS estimates. Ma et al. (2015) considered the statistical properties of algorithmic leveraging. They derived biases and variances of leverage-based subsampling estimators in linear regression and proposed a shrinkage algorithmic leveraging method to improve the performance. Raskutti and Mahoney (2016) considered both the algorithmic and statistical aspects of solving large-scale LS problems using random sketching. Wang et al. (2019) and Wang (2019) developed an information-based optimal subdata selection method to select subsample deterministically for ordinary LS in linear regression. The aforesaid results were obtained exclusively within the context of linear models. Fithian and Hastie (2014) proposed a computationally efficient local case-control subsampling method for logistic regression with large imbalanced data. Han et al. (2019) developed a local uncertainty sampling approach for multi-class logistic regression. Recently, Wang et al. (2018) developed an Optimal Subsampling Method under the A-optimality Criterion (OSMAC) for logistic regression; Yao and Wang (2019) and Ai et al. (2019) extended this method to include multi-class logistic regression and generalized linear regression models, respectively. Although they derived optimal subsampling probabilities, they did not investigate whether a better weighting scheme can further improve the estimation efficiency.

This paper focuses on logistic regression models, which are widely used for statistical inference in many disciplines, such as business, computer science, education, and genetics,

©2019 HaiYing Wang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-596.html.

among others (Hosmer Jr et al., 2013). Based on optimal subsamples taken according to OSMAC developed in Wang et al. (2018), more efficient methods, in terms of both parameter estimation and numerical computation, will be proposed. The reminder of the paper is organized as follows. Model setups and notations are introduced in Section 2. The OSMAC will also be briefly reviewed in this section. Section 3 presents the more efficient estimator and its asymptotic properties. Section 4 considers Poisson subsampling. Section 5 discusses issues related to practical implementation and summaries the methods from Sections 3 and 4 into two practical algorithms. Section 6 gives unconditional asymptotic distributions for the estimator from Poisson subsampling. Section 7 discusses asymptotic distributions with pilot and model misspecifications. Section 8 evaluates the practical performance of the proposed methods using numerical experiments. Section 9 concludes, and the appendix contains proofs and technical details.

2. Model setup and optimal subsampling

Let $y \in \{0, 1\}$ be a binary response variable and $\mathbf{x}$ be a d dimensional covariate. A logistic regression model describes the conditional probability of $y = 1$ given $\mathbf{x}$, and it has the following form,

$$\mathbb{P}(y = 1|\mathbf{x}) = p(\mathbf{x}, \boldsymbol{\beta}) = \frac{e^{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}}}{1 + e^{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}}}, \tag{1}$$

where $\boldsymbol{\beta}$ is a $d \times 1$ vector of unknown regression coefficients belonging to a compact subset of $\mathbb{R}^d$.

With independent full data of size N from Model (1), say, $\mathcal{D}_N = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$, the unknown parameter $\boldsymbol{\beta}$ is often estimated by the maximum likelihood estimator (MLE), denoted as $\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}$. It is the maximizer of the log-likelihood function, namely,

$$\hat{\boldsymbol{\beta}}_{\mathrm{MLE}} = \arg\max_{\boldsymbol{\beta}} \ \ell_f(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left\{ y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} - \log\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i}\right) \right\}.$$

Since there is no general closed-form solution to the MLE, Newton's method or iteratively reweighted least squares method (McCullagh and Nelder, 1989) is often adopted to find it numerically. This typically takes $O(\zeta N d^2)$ time, where ζ is the number of iterations in the optimization procedure. For super-large data set, the computing time $O(\zeta N d^2)$ may be too long to afford, and iterative computation is infeasible if the data volume is larger than the available random-access memory (RAM). To overcome this computational bottleneck for the application of logistic regression to massive data, Wang et al. (2018) developed the OSMAC under the subsampling framework.

Let π_1, ..., π_N be subsampling probabilities such that $\sum_{i=1}^{N} \pi_i = 1$. Using subsampling with replacement, draw a random subsample of size n according to the probabilities $\{\pi_i\}_{i=1}^{N}$ from the full data. We use * to indicate quantities for a subsample, namely, denote the covariates, responses, and subsampling probabilities in a subsample as $\mathbf{x}_i^*$, y_i^*, and π_i^*, respectively, for $i = 1, ..., n$. Wang et al. (2018) define the weighted subsample estimator $\hat{\boldsymbol{\beta}}_w^{\pi}$ to be

$$\hat{\boldsymbol{\beta}}_w^{\pi} = \arg\max_{\boldsymbol{\beta}} \ell_w^*(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \frac{y_i^* \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^* - \log\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^*}\right)}{\pi_i^*}.$$

The key to success here is how to specify the values for π_i's so that more informative data points are sampled with higher probabilities. Wang et al. (2018) derived optimal subsampling probabilities that minimize the asymptotic MSE of $\hat{\boldsymbol{\beta}}_w^{\pi}$. They first showed that $\hat{\boldsymbol{\beta}}_w^{\pi}$ is asymptotically normal. Specifically, for large n and N, the conditional distribution of $\sqrt{n}(\hat{\boldsymbol{\beta}}_w^{\pi} - \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})$ given the full data $\mathcal{D}_N$ can be approximated by a normal distribution with mean $\mathbf{0}$ and variance-covariance matrix $\mathbf{V}_N = \mathbf{M}_N^{-1}\mathbf{V}_{Nc}\mathbf{M}_N^{-1}$, in which

$$\mathbf{M}_N = \frac{1}{N}\sum_{i=1}^{N} \phi_i(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}, \quad \mathbf{V}_{Nc} = \frac{1}{N}\sum_{i=1}^{N} \frac{|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|^2 \mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{N\pi_i},$$

and $\phi_i(\boldsymbol{\beta}) = p(\mathbf{x}_i, \boldsymbol{\beta})\{1 - p(\mathbf{x}_i, \boldsymbol{\beta})\}$ with $p(\mathbf{x}_i, \boldsymbol{\beta}) = e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}}/(1 + e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}})$. Based on this asymptotic distribution, they derive the following two optimal subsampling probabilities

$$\pi_i^{\mathrm{Aopt}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \frac{|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|\|\mathbf{M}_N^{-1}\mathbf{x}_i\|}{\sum_{j=1}^{N} |y_j - p(\mathbf{x}_j, \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|\|\mathbf{M}_N^{-1}\mathbf{x}_j\|}, \quad i = 1, ..., N; \tag{2}$$

$$\pi_i^{\mathrm{Lopt}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \frac{|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|\|\mathbf{x}_i\|}{\sum_{j=1}^{N} |y_j - p(\mathbf{x}_j, \hat{\boldsymbol{\beta}}_{\mathrm{MLE}})|\|\mathbf{x}_j\|}, \quad i = 1, ..., N. \tag{3}$$

Here, $\{\pi_i^{\mathrm{Aopt}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\}_{i=1}^{N}$ minimize $\mathrm{tr}(\mathbf{V}_N)$, the trace of $\mathbf{V}_N$, and this is the A-optimality criterion in optimum experimental designs (Atkinson et al., 2007); $\{\pi_i^{\mathrm{Lopt}}(\hat{\boldsymbol{\beta}}_{\mathrm{MLE}})\}_{i=1}^{N}$ minimize $\mathrm{tr}(\mathbf{V}_{Nc})$, and this is a choice of the L-optimality criterion. These subsampling probabilities have a lot of nice properties and meaningful interpretations. More details can be found in Section 3 of Wang et al. (2018).

For ease of presentation, use the following general notation to denote subsampling probabilities

$$\pi_i^{\mathrm{OS}}(\boldsymbol{\beta}) = \frac{|y_i - p(\mathbf{x}_i, \boldsymbol{\beta})|h(\mathbf{x}_i)}{\sum_{j=1}^{N} |y_j - p(\mathbf{x}_j, \boldsymbol{\beta})|h(\mathbf{x}_j)}, \quad i = 1, ..., N, \tag{4}$$

where $h(\mathbf{x})$ is a univariate function of $\mathbf{x}$. We provide some intuitions on choosing $h(\mathbf{x})$. Let L be a matrix with d columns. Choosing $h(\mathbf{x}) = \|L\mathbf{M}_N^{-1}\mathbf{x}\|$ minimizes the trace of $L\mathbf{V}_N L^{\mathrm{T}}$, which is the conditional asymptotic variance-covariance matrix of $L\hat{\boldsymbol{\beta}}_w^{\pi}$ (scaled by n) given the full data $\mathcal{D}_N$. Two special choices of $h(\mathbf{x})$ correspond to $L = \mathbf{I}$ (the identity matrix) and $L = \mathbf{M}_N$. If $L = \mathbf{I}$, then $h(\mathbf{x}) = \|\mathbf{M}_N^{-1}\mathbf{x}\|$ and $\pi_i^{\mathrm{OS}}(\boldsymbol{\beta})$ becomes $\pi_i^{\mathrm{Aopt}}(\boldsymbol{\beta})$; if $L = \mathbf{M}_N$, then $h(\mathbf{x}) = \|\mathbf{x}\|$ and $\pi_i^{\mathrm{OS}}(\boldsymbol{\beta})$ becomes $\pi_i^{\mathrm{Lopt}}(\boldsymbol{\beta})$. If one is interested in a specific component of $\boldsymbol{\beta}$, say β_j, then L can be chosen as a row vector with the j-th element being one and all other elements being zero. With this choice, $h(\mathbf{x}) = \|\mathbf{M}_{N,\cdot j}^{-1}\mathbf{x}\|$ where $\mathbf{M}_{N,\cdot j}^{-1}$ means the j-th row of $\mathbf{M}_N^{-1}$, and the asymptotic variance of $\hat{\boldsymbol{\beta}}_{w,j}^{\pi}$ is minimized. If $h(\mathbf{x}) = 1$, then $\pi_i^{\mathrm{OS}}(\boldsymbol{\beta})$'s are proportional to the local case-control subsampling probabilities (Fithian and Hastie, 2014).

Note that $\{\pi_i^{\mathrm{OS}}(\boldsymbol{\beta})\}_{i=1}^{N}$ depend on the unknown $\boldsymbol{\beta}$, so a pilot estimate of $\boldsymbol{\beta}$ is required to approximate them. Let $\hat{\boldsymbol{\beta}}_0$ be a pilot estimator from a pilot subsample taken from the full data, for which we will provide more details in Section 5. The original weighted OSMAC estimator is

$$\hat{\boldsymbol{\beta}}_w = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \frac{y_i^* \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^* - \log\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^*}\right)}{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)^*}. \tag{5}$$

In Wang et al. (2018), $\hat{\boldsymbol{\beta}}_u$ has exceptional performance because $\{\pi_i^{\rm OS}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ are able to include more informative data points in the subsample. However, we can improve the weighting scheme adopted in (5). Intuitively, a larger $\pi_i^{\rm OS}(\hat{\boldsymbol{\beta}}_0)$ means that the data point $(\mathbf{x}_i, y_i)$ contains more information about $\boldsymbol{\beta}$, but it has a smaller weight in the objective function in (5). This reduces contributions of more informative data points to the objective function for parameter estimation.

The weighted estimator in (5) is used because $\{\pi_i^{\rm OS}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ depend on the responses y_i's and an un-weighted estimator is biased. If the bias can be corrected, then the resultant estimator can be more efficient in parameter estimation, because an un-weighted estimator often has a smaller variance-covariance matrix compared with an inverse probability weighted estimator. Intuitively, if some data points with very small values of $\pi_i^{\rm OS}(\hat{\boldsymbol{\beta}}_0)$ are selected in the subsample, then the target function in (5) would be dominated by these data points. As a result, the variance-covariance matrix of the weighted estimator would be inflated by small values of $\pi_i^{\rm OS}(\hat{\boldsymbol{\beta}}_0)$. Note that π_i's appear in the denominator of $\mathbf{V}_{Nc}$ in the asymptotic variance-covariance matrix of the weighted estimator. A major goal of this paper is to develop un-weighted estimation procedures. Interestingly, for the subsampling probabilities in (4), the bright idea proposed in Fithian and Hastie (2014) can be used to correct the bias of the un-weighted estimator.

3. More efficient estimator

Let $\{(\mathbf{x}_1^*, y_1^*), ..., (\mathbf{x}_n^*, y_n^*)\}$ be a random subsample of size n taken from the full data using sampling with replacement according to the probabilities $\{\pi_i^{\rm OS}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ defined in (4). Using this subsample, we present a more efficient estimation procedure based on un-weighted estimator with bias correction. Remember that a pilot estimate is required, and we use $\hat{\boldsymbol{\beta}}_0$ to denote it. Here, we focus the discussion on the new estimation procedure and assume that $\hat{\boldsymbol{\beta}}_0$ is obtained based on a pilot subsample of size n_0 and it is consistent. More details about this pilot estimator will be provided in Section 5, and the scenario that $\hat{\boldsymbol{\beta}}_0$ is inconsistent will be investigated in Section 7.1. The following procedure describes how to obtain the un-weighted estimator with bias correction, denoted as $\hat{\boldsymbol{\beta}}_{uw}$.

Calculate the naive un-weighted estimator

$$\tilde{\boldsymbol{\beta}}_{uw} = \arg\max_{\boldsymbol{\beta}} \ell_{uw}^*(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^n \left\{\boldsymbol{\beta}^{\rm T}\mathbf{x}_i^* y_i^* - \log\left(1 + e^{\boldsymbol{\beta}^{\rm T}\mathbf{x}_i^*}\right)\right\}, \tag{6}$$

and then let

$$\hat{\boldsymbol{\beta}}_{uw} = \tilde{\boldsymbol{\beta}}_{uw} + \hat{\boldsymbol{\beta}}_0. \tag{7}$$

The naive un-weighted estimator $\tilde{\boldsymbol{\beta}}_{uw}$ in (6) is biased, and the bias is corrected in (7) using $\hat{\boldsymbol{\beta}}_0$. We will show in the following that $\hat{\boldsymbol{\beta}}_{uw}$ is asymptotically unbiased. This, together with the fact that $\hat{\boldsymbol{\beta}}_0$ is consistent, shows the interesting fact that $\tilde{\boldsymbol{\beta}}_{uw}$ converges to $\mathbf{0}$ in probability as n_0, n, and N go to infinity.

To investigate the asymptotic properties, we use $\boldsymbol{\beta}_t$ to denote the true value of $\boldsymbol{\beta}$, and summarize some regularity conditions in the following.

Assumption 1 *The matrix* $\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\rm T}\}$ *is finite and positive-definite.*

Assumption 2 *The covariate* $\mathbf{x}$ *and function* $h(\cdot)$ *satisfy that* $\mathbb{E}\{\|\mathbf{x}\|^2 h^2(\mathbf{x})\} < \infty$, *and* $\mathbb{E}\{\|\mathbf{x}\|^2 h(\mathbf{x})\} < \infty$.

Assumption 3 *As* $n \to \infty$, $n\mathbb{E}\{h(\mathbf{x})I(\|\mathbf{x}\|^2 > n)\} \to 0$, *where* $I()$ *is the indicator function.*

Assumption 1 is required to establish the asymptotic normality. This is a commonly used assumption, e.g., in Fithian and Hastie (2014); Wang et al. (2018), among others. Assumptions 2 and 3 impose moment conditions on the covariate distribution and the function $h(\mathbf{x})$. When $h(\mathbf{x}) = 1$, if $\mathbb{E}\|\mathbf{x}\|^2 < \infty$, then both the two conditions in Assumption 2 and the condition in Assumption 3 hold. Thus, the assumptions required in this paper are not stronger than those required by Fithian and Hastie (2014). When $h(\mathbf{x}) = \|\mathbf{x}\|$, by Hölder's inequality,

$$\begin{aligned} n\mathbb{E}\{h(\mathbf{x})I(\|\mathbf{x}\|^2 > n)\} &\le n(\mathbb{E}\|\mathbf{x}\|^3)^{1/3}\{\mathbb{E}I(\|\mathbf{x}\|^2 > n)\}^{2/3} \\ &= (\mathbb{E}\|\mathbf{x}\|^3)^{1/3}\{n^{3/2}\mathbb{P}(\|\mathbf{x}\|^3 > n^{3/2})\}^{2/3}. \end{aligned}$$

Note that $n^{3/2}I(\|\mathbf{x}\|^3 > n^{3/2}) \le \|\mathbf{x}\|^3$ and $I(\|\mathbf{x}\|^3 > n^{3/2}) \to 0$ in probability. Thus, if $\mathbb{E}(\|\mathbf{x}\|^3) < \infty$, then $n^{3/2}\mathbb{P}(\|\mathbf{x}\|^3 > n^{3/2}) = \mathbb{E}\{n^{3/2}I(\|\mathbf{x}\|^3 > n^{3/2})\} \to 0$ (see Theorem 1.3.6 of Serfling, 1980). Therefore, if $\mathbb{E}(\|\mathbf{x}\|^3) < \infty$, Assumption 3 holds. This shows that $\mathbb{E}\|\mathbf{x}\|^4 < \infty$ implies all the three conditions required in Assumptions 2 and 3. Note that Wang et al. (2018) requires that $\mathbb{E}(e^{\mathbf{v}^{\rm T}\mathbf{x}}) < \infty$ for any $\mathbf{v} \in \mathbb{R}^d$ in order to establish the asymptotic properties when a pilot estimate is used to approximate optimal subsampling probabilities. Thus, the required conditions in this paper are weaker than those required in Wang et al. (2018). Assumptions 1 and 2 are required in all the theorems in this paper while Assumption 3 is only required in Theorems 1, 18, and 24.

Theorem 1 *Under Assumptions 1-3, conditional on* $\mathcal{D}_N$, *if* $\hat{\boldsymbol{\beta}}_0$ *is consistent, then as* n_0, n, *and* N *go to infinity,*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\rm wMLE}) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\big), \tag{8}$$

in distribution; furthermore, if $n/N \to 0$, *then*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\big) \tag{9}$$

in distribution, where

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \left[\frac{\mathbb{E}\{\phi(\boldsymbol{\beta})h(\mathbf{x})\mathbf{x}\mathbf{x}^{\rm T}\}}{4\Phi(\boldsymbol{\beta})}\right]^{-1}, \quad \Phi(\boldsymbol{\beta}) = \mathbb{E}\{\phi(\boldsymbol{\beta})h(\mathbf{x})\}, \quad \phi(\boldsymbol{\beta}) = p(\mathbf{x}, \boldsymbol{\beta})\{1 - p(\mathbf{x}, \boldsymbol{\beta})\},$$

and $\hat{\boldsymbol{\beta}}_{\rm wMLE}$ *is a weighted MLE based on the full data defined as*

$$\hat{\boldsymbol{\beta}}_{\rm wMLE} = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^N |y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)| h(\mathbf{x}_i)\big[y_i\mathbf{x}_i^{\rm T}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0) - \log\{1 + e^{\mathbf{x}_i^{\rm T}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)}\}\big]. \tag{10}$$

Here $\hat{\boldsymbol{\beta}}_{\rm wMLE}$ *satisfies that*

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\rm wMLE} - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Sigma}_{\rm wMLE}\big), \tag{11}$$

in distribution if $\hat{\boldsymbol{\beta}}_0$ is obtained from a uniform pilot subsample of size n_0 such that $n_0/\sqrt{N} = o(1)$ or if $\hat{\boldsymbol{\beta}}_0$ is independent of $\mathcal{D}_N$, where

$$\boldsymbol{\Sigma}_{\text{wMLE}} = [\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}]^{-1}\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h^2(\mathbf{x})\mathbf{x}\mathbf{x}\}[\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}]^{-1}.$$

Remark 2 *Theorem 1 shows that the un-weighted estimator $\hat{\boldsymbol{\beta}}_{uw}$ is $\sqrt{n}$-consistent to $\hat{\boldsymbol{\beta}}_{\text{wMLE}}$, a weighted MLE based on the full data in conditional probability, while Theorem 5 of Wang et al. (2018) shows that the weighted estimator $\hat{\boldsymbol{\beta}}_w$ is $\sqrt{n}$-consistent to $\hat{\boldsymbol{\beta}}_{\text{MLE}}$, the un-weighted MLE based on the full data in conditional probability. Specifically,* (8) *implies that given $\mathcal{D}_N$ in probability,*

$$\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\text{wMLE}} = O_{P|\mathcal{D}_N}(n^{-1/2}). \tag{12}$$

The $O_{P|\mathcal{D}_N}(n^{-1/2})$ expression in (12) *means that for any $\epsilon > 0$, there exist a δ_ϵ such that as $n, N \to \infty$,*

$$\mathbb{P}\Big\{\sup_n \mathbb{P}(\|\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\text{wMLE}}\| > n^{-1/2}\delta_\epsilon|\mathcal{D}_N) \le \epsilon\Big\} \to 1.$$

Note that if a sequence is bounded in conditional probability, then it is bounded in unconditional probability, i.e., if $a_n = O_{P|\mathcal{D}_N}(1)$, then $a_n = O_P(1)$ (Xiong and Li, 2008; Cheng and Huang, 2010). Therefore, (12) *implies that $\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\text{wMLE}} = O_P(n^{-1/2})$. Similarly,* (11) *implies that $\hat{\boldsymbol{\beta}}_{\text{wMLE}} - \boldsymbol{\beta}_t = O_P(N^{-1/2})$. Thus, $\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_t = O_P(n^{-1/2} + N^{-1/2}) = O_P(n^{-1/2})$, showing the $\sqrt{n}$-consistency of $\hat{\boldsymbol{\beta}}_{uw}$ to the true parameter under the unconditional distribution.*

Remark 3 *For $\hat{\boldsymbol{\beta}}_{\text{wMLE}}$, if $\hat{\boldsymbol{\beta}}_0$ is fixed, say $\hat{\boldsymbol{\beta}}_0 = \boldsymbol{\beta}_0$, then the population log-likelihood for the objective function in* (10) *is*

$$\mathbb{E}\Big(a(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\beta}_0)\Big[p(\mathbf{x}, \boldsymbol{\beta} - \boldsymbol{\beta}_0)\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0) - \log\{1 + e^{\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)}\}\Big]\Big),$$

where $a(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\beta}_0) = \big[p(\mathbf{x}, \boldsymbol{\beta})\{1 - p(\mathbf{x}, \boldsymbol{\beta}_0)\} + \{1 - p(\mathbf{x}, \boldsymbol{\beta})\}p(\mathbf{x}, \boldsymbol{\beta}_0)\big]h(\mathbf{x})$. If $h(\mathbf{x}) = 1$, then this population log-likelihood is identical to that for the local case-control subsampling estimator. For general $h(\mathbf{x})$, since it does not rely on the response variable, we expect that $\hat{\boldsymbol{\beta}}_{\text{wMLE}}$ inherits the main properties of the the local case-control subsampling estimator, including those under model misspecification. Indeed this is the case, and more details for the scenarios of misspecifications will be presented in Section 7.

Theorem 1 shows that, asymptotically, the distribution of $\hat{\boldsymbol{\beta}}_{uw}$ given $\mathcal{D}_N$ is centered around $\hat{\boldsymbol{\beta}}_{\text{wMLE}}$ with variance-covariance matrix $n^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$, and the distribution of $\hat{\boldsymbol{\beta}}_{\text{wMLE}}$ is centered around $\boldsymbol{\beta}_t$ with variance-covariance matrix $N^{-1}\boldsymbol{\Sigma}_{\text{wMLE}}$. Thus, both $n^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ and $N^{-1}\boldsymbol{\Sigma}_{\text{wMLE}}$ should be considered in accessing the quality of $\hat{\boldsymbol{\beta}}_{uw}$ for estimating the true parameter $\boldsymbol{\beta}_t$. However, in subsampling setting, it is expected that $n \ll N$; otherwise, the computational benefit is minimum. Thus, $n^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ is the dominating term in quantifying the variation of $\hat{\boldsymbol{\beta}}_{uw}$. If $n/N \to 0$, then the variation of $\hat{\boldsymbol{\beta}}_{\text{wMLE}}$ can be ignored as stated in (9).

Now we compare the estimation efficiency of $\hat{\boldsymbol{\beta}}_{uw}$ with that of the weighted estimator $\hat{\boldsymbol{\beta}}_w$. With the optimal subsampling probabilities $\{\pi_i^{\text{OS}}(\hat{\boldsymbol{\beta}}_{\text{MLE}})\}_{i=1}^N$, the asymptotic variance-covariance matrix (scaled by n), $\mathbf{V}_N$, for the weighted estimator $\hat{\boldsymbol{\beta}}_w$ has a form of $\mathbf{V}_N^{\text{OS}} = \mathbf{M}_N^{-1}\mathbf{V}_{Nc}^{\text{OS}}\mathbf{M}_N^{-1}$, where

$$\mathbf{V}_{Nc}^{\text{OS}} = \left\{\frac{1}{N}\sum_{i=1}^N |y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_{\text{MLE}})|h(\mathbf{x}_i)\right\}\left\{\frac{1}{N}\sum_{i=1}^N \frac{|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_{\text{MLE}})|\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{h(\mathbf{x}_i)}\right\}.$$

Note that the full data MLE $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ is consistent under Assumptions 1-2. If $\mathbb{E}\{\|\mathbf{x}\|^2/h(\mathbf{x})\} < \infty$, then from Lemma 28 in the appendix and the law of large numbers, $\mathbf{V}_N^{\text{OS}}$ converges in probability to $\mathbf{V}^{\text{OS}} = \mathbf{M}^{-1}\mathbf{V}_c^{\text{OS}}\mathbf{M}^{-1}$, where

$$\mathbf{M} = \mathbb{E}\{\phi(\boldsymbol{\beta}_t)\mathbf{x}\mathbf{x}^{\mathrm{T}}\} \quad \text{and} \quad \mathbf{V}_c^{\text{OS}} = 4\Phi(\boldsymbol{\beta}_t)\mathbb{E}\left\{\frac{\phi(\boldsymbol{\beta}_t)\mathbf{x}\mathbf{x}^{\mathrm{T}}}{h(\mathbf{x})}\right\}.$$

Note that the asymptotic distribution of $\hat{\boldsymbol{\beta}}_w$ given $\mathcal{D}_N$ is centered around $\hat{\boldsymbol{\beta}}_{\text{MLE}}$. It can be shown that under Assumptions 1-2,

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\text{MLE}} - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}\big(\mathbf{0}, \mathbf{M}^{-1}\big),$$

in distribution. Thus, both $n^{-1}\mathbf{V}^{\text{OS}}$ and $N^{-1}\mathbf{M}^{-1}$ should be considered in accessing the quality of $\hat{\boldsymbol{\beta}}_w$ for estimating the true parameter $\boldsymbol{\beta}_t$. However, similar to the case for $\hat{\boldsymbol{\beta}}_{uw}$, $N^{-1}\mathbf{M}^{-1}$ is small compared with $n^{-1}\mathbf{V}^{\text{OS}}$ if $n \ll N$, and it is negligible if $n/N \to 0$. Therefore, the relative performance between $\hat{\boldsymbol{\beta}}_{uw}$ and $\hat{\boldsymbol{\beta}}_w$ are mainly determined by the relative magnitude between $\mathbf{V}^{\text{OS}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$. We have the following result comparing $\mathbf{V}^{\text{OS}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$.

Proposition 4 *If $\mathbf{M}$, $\mathbf{V}_c^{\text{OS}}$, and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ are finite and positive definite matrices, then*

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} \le \mathbf{V}^{\text{OS}}. \tag{13}$$

Here, the inequality is in the Loewner ordering, i.e., for positive semi-definite matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \ge \mathbf{B}$ if and only if $\mathbf{A} - \mathbf{B}$ is positive semi-definite. If $h(\mathbf{x}) = 1$, then the equality in (13) *holds. Furthermore, note that the asymptotic variance-covariance matrix (scaled by n) for uniform subsampling estimator is $\mathbf{M}^{-1}$. If $\boldsymbol{\beta}_t \ne \mathbf{0}$ and $h(\mathbf{x}) = \|L\mathbf{M}^{-1}\mathbf{x}\|$ for some matrix L, then*

$$\mathrm{tr}(L\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}L^{\mathrm{T}}) \le \mathrm{tr}(L\mathbf{V}^{\text{OS}}L^{\mathrm{T}}) \le \mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}\mathrm{tr}(L\mathbf{M}^{-1}L^{\mathrm{T}}) < \mathrm{tr}(L\mathbf{M}^{-1}L^{\mathrm{T}}). \tag{14}$$

Remark 5 *This proposition shows that $\hat{\boldsymbol{\beta}}_{uw}$ is typically more efficient than $\hat{\boldsymbol{\beta}}_w$ in estimating $\boldsymbol{\beta}_t$. The numerical results in Section 8 also confirm this. Assume that $n/N \to \rho$. For the un-weighted estimator, the variation of $\sqrt{N}(\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\text{wMLE}})$ is measured by $\rho^{-1}\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ and the variation of $\sqrt{N}(\hat{\boldsymbol{\beta}}_{\text{wMLE}} - \boldsymbol{\beta}_t)$ is measured by $\boldsymbol{\Sigma}_{\text{wMLE}}$, while for the weighted estimator the variation of $\sqrt{N}(\hat{\boldsymbol{\beta}}_w - \hat{\boldsymbol{\beta}}_{\text{MLE}})$ is measured by $\rho^{-1}\mathbf{V}^{\text{OS}}$ and the variation of $\sqrt{N}(\hat{\boldsymbol{\beta}}_{\text{MLE}} - \boldsymbol{\beta}_t)$ is measured by $\mathbf{M}^{-1}$. Note that $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$, $\boldsymbol{\Sigma}_{\text{wMLE}}$, $\mathbf{V}^{\text{OS}}$, and $\mathbf{M}^{-1}$ are all fixed constant matrices that do not depend on ρ, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} \le \mathbf{V}^{\text{OS}}$, and $\boldsymbol{\Sigma}_{\text{wMLE}} = \boldsymbol{\Sigma}_{\text{MLE}}$ if $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} = \mathbf{V}^{\text{OS}}$. Thus, if ρ is small enough, $\hat{\boldsymbol{\beta}}_{uw}$ is more efficient than $\hat{\boldsymbol{\beta}}_w$ in estimating $\boldsymbol{\beta}_t$, and we do not need to require that $n/N \to 0$.*

Since the equality in (13) holds if $h(\mathbf{x}) = 1$, this indicates that for subsample obtained from local case-control subsampling with replacement, the weighted and un-weighted estimators have the same conditional asymptotic distribution.

4. Poisson subsampling

For the more efficient estimator $\hat{\boldsymbol{\beta}}_{uw}$ in Section 3 as well as the weighted estimator $\hat{\boldsymbol{\beta}}_{w}$, the subsampling procedure used is sampling with replacement, which is faster to compute than sampling without replacement for a fixed sample size. In addition, the resultant subsample are independent and identically distributed (i.i.d.) conditional on the full data. However, to implement sampling with replacement, subsampling probabilities $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ need to be used all at once, and a large amount of random numbers need to be generated all at once. This may reduce the computational efficiency, and it may require a large RAM to implement the method. Furthermore, since a data point may be included for multiple times in the subsample, the resultant estimator may not be the most efficient.

To enhance the computation and estimation efficiency of the subsample estimator, we consider Poisson subsampling, which is also fast to compute and the resultant subsample can be independent without conditioning on the full data. Note that for subsampling with replacement, a resultant subsample is generally not independent, although it is i.i.d conditional on the full data. As another advantage with Poisson subsampling, there is no need to calculate subsampling probabilities all at once, nor to generate a large amount of random numbers all at once. Furthermore, a data point cannot be included in the subsample for more than one time. A limitation of Poisson subsampling is that the subsample size is always random. Due to this, we use n^* to denote the actual subsample size, and abuse the notation in this section to use n to denote the expected subsample size, i.e., $\mathbb{E}(n^*) = n$.

Note that $\{\pi_i^{\mathrm{OS}}(\boldsymbol{\beta})\}_{i=1}^N$ depend on the full data through the term in the denominator, $\sum_{i=1}^N |y_i - p(\mathbf{x}_i, \boldsymbol{\beta})|h(\mathbf{x}_i)$. Write $\Psi_N(\boldsymbol{\beta}) = N^{-1}\sum_{i=1}^N |y_i - p(\mathbf{x}_i, \boldsymbol{\beta})|h(\mathbf{x}_i)$, and denote its limit as $\Psi(\boldsymbol{\beta}) = \mathbb{E}\{|y - p(\mathbf{x}, \boldsymbol{\beta})|h(\mathbf{x})\}$. Note that $\Psi(\boldsymbol{\beta}_t) = 2\Phi(\boldsymbol{\beta}_t)$. The pilot subsample can be used to obtain an estimator of $\Psi(\boldsymbol{\beta}_t)$ to approximate $\Psi_N(\boldsymbol{\beta})$. Let $\hat{\Psi}_0$ be a pilot estimator of $\Psi(\boldsymbol{\beta}_t)$. Here, we focus on the Poisson subsampling procedure and assume that such $\hat{\Psi}_0$ is available and consistent. We will provide more details on $\hat{\Psi}_0$ in Section 5 and Section 7.

With $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ available, the Poisson subsampling procedure is described as the following. For $i = 1, ..., N$, calculate $\pi_i^p = |y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)/(N\hat{\Psi}_0)$, generate $u_i \sim U(0, 1)$, and include $(\mathbf{x}_i, y_i, \pi_i^p)$ in the subsample if $u_i \le n\pi_i^p$. For the obtained subsample, say $\{(\mathbf{x}_1^*, y_1^*, \pi_1^{p*}), ..., (\mathbf{x}_{n^*}^*, y_{n^*}^*, \pi_{n^*}^{p*})\}$, calculate

$$\tilde{\boldsymbol{\beta}}_p = \arg\max_{\boldsymbol{\beta}} \ell_p^*(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n^*} (n\pi_i^{p*} \vee 1)\big\{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^* y_i^* + \log(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^*})\big\}, \tag{15}$$

and let $\hat{\boldsymbol{\beta}}_p = \tilde{\boldsymbol{\beta}}_p + \hat{\boldsymbol{\beta}}_0$. Note that here the actual subsample size n^* is random.

Poisson subsampling does not require to calculate π_i^p's all at once; each π_i^p is calculated for each individual data point when scanning through the full data. Thus, one pass through the data finishes the sampling. For the estimation step, if π_i^p is large so that $n\pi_i^p > 1$, then this more informative data point will be given a larger weight, $n\pi_i^p$, in the objective function in (15). The following theorem describes asymptotic properties of $\hat{\boldsymbol{\beta}}_p$.

Theorem 6 *Under Assumptions 1-2 and assume that $\hat{\boldsymbol{\beta}}_0$ is consistent, conditional on $\mathcal{D}_N$, as n_0, n, and N go to infinity, if $n/N \to 0$, then*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}(0,\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}),$$

in distribution; if $n/N \to \rho \in (0, 1)$, then

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}) \longrightarrow \mathbb{N}(0,\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_\rho\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}), \tag{16}$$

in distribution, where

$$\boldsymbol{\Lambda}_\rho = \frac{\mathbb{E}\big[|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\{\Psi(\boldsymbol{\beta}_t) - \rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\}_+\mathbf{x}\mathbf{x}^{\mathrm{T}}\big]}{4\Psi^2(\boldsymbol{\beta}_t)}$$

with $\psi(\boldsymbol{\beta}) = y - p(\mathbf{x}, \boldsymbol{\beta})$ and $\Psi(\boldsymbol{\beta}) = \mathbb{E}\{|y - p(\mathbf{x}, \boldsymbol{\beta})|h(\mathbf{x})\}$, and $()_+$ means the positive part of the quantity, i.e., $a_+ = aI(a > 0)$.

Remark 7 *Similar to the case of Theorem 1,* (16) *implies that given $\mathcal{D}_N$ in probability, $\hat{\boldsymbol{\beta}}_p - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} = O_{P|\mathcal{D}_N}(n^{-1/2})$, which implies that $\hat{\boldsymbol{\beta}}_p - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} = O_P(n^{-1/2})$ unconditionally because if a sequence is stochastically bounded in conditional probability, then it is also stochastically bounded in unconditional probability (Xiong and Li, 2008; Cheng and Huang, 2010). Since $\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_t = O_P(N^{-1/2})$, we have $\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t = O_P(n^{-1/2} + N^{-1/2}) = O_P(n^{-1/2})$, showing that $\hat{\boldsymbol{\beta}}_p$ is $\sqrt{n}$-consistent to $\boldsymbol{\beta}_t$ unconditionally on the full data.*

Theorem 6 shows that with Poisson subsampling, the asymptotic variance-covariance matrices may differ for different sampling ratios n/N. In addition, comparing Theorems 1 and 6, we know that $\hat{\boldsymbol{\beta}}_{uw}$ and $\hat{\boldsymbol{\beta}}_p$ have the same asymptotic distribution if $n/N \to 0$. This is intuitive because if the sampling ratio n/N is small, sampling with replacement has close performance to sampling without replacement. However, if the sampling ratio n/N does not converge to zero, then $\hat{\boldsymbol{\beta}}_{uw}$ and $\hat{\boldsymbol{\beta}}_p$ have the same asymptotic mean but different asymptotic variance-covariance matrices. The following result compares the two asymptotic variance-covariance matrices.

Proposition 8 *If $\rho > 0$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ is a finite and positive definite matrix, then*

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_\rho\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} < \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t},$$

under the Loewner ordering.

This proposition shows that Poisson subsampling is more efficient than sampling with replacement.

5. Pilot estimate and practical implementation

Since $\{\pi_i^{\mathrm{OS}}(\boldsymbol{\beta})\}_{i=1}^N$ depend on the unknown $\boldsymbol{\beta}$, a pilot estimate of $\boldsymbol{\beta}$ is required to approximate them. The pilot estimate can be obtained by taking a pilot subsample using uniform subsampling or case-control subsampling. For uniform subsampling, all subsampling probabilities are equal, while for case-control subsampling, the subsampling probability for the

cases ($y_i = 1$) is different from that for the controls ($y_i = 0$). Let the subsampling probabilities used to take the pilot subsample be

$$\pi_{0i} = \frac{c_0(1-y_i) + c_1 y_i}{N}, \tag{17}$$

where c_0 and c_1 are two constants that can be used to balance the numbers of 0's and 1's in the responses for the pilot subsample. If $c_0 = c_1 = 1$, then $\pi_{0i} = N^{-1}$ corresponds to the uniform subsampling. This choice is recommended due to its simplicity if the proportion of 1's is close to 0.5 (Wang et al., 2018). If $c_0 \neq c_1$, then π_{0i}'s are the case-control subsampling probabilities. This choice is recommended for imbalanced full data. Often, some prior information about the marginal probability $\mathbb{P}(y = 1)$ is available. If p_{pr} is the prior marginal probability, we can choose $c_0 = \{2(1-p_{pr})\}^{-1}$ and $c_1 = (2p_{pr})^{-1}$. The pilot estimate $\hat{\boldsymbol{\beta}}_0$ can be obtained using the pilot subsample. For uniform subsampling, weighted and un-weighted estimators are the same. For case-control subsampling, we use un-weighted estimators with bias correction for both sampling with replacement and Poisson subsampling.

To obtain a final estimator, Wang et al. (2018) pooled the pilot subsample with the second stage subsample taken using approximated optimal subsampling probabilities. While this does not make a difference asymptotically since n_0 is typically a small term compared with n, i.e., $n_0 = o(n)$, using the pilot subsample helps to improve the finite sample performance in practical applications. However, pooling the raw samples may not be the most computationally efficient way of utilizing the pilot subsample. Since $\hat{\boldsymbol{\beta}}_0$ is already calculated, we can use it directly to improve the second stage estimator using the aggregation procedure in the divide-and-conquer method (Lin and Xie, 2011; Schifano et al., 2016). This avoids iterative calculations on the pilot subsample for the second time.

For subsampling with replacement, when the full data cannot be loaded into available RAM, special considerations have to be given in practical implementation. If the full data is larger than available RAM while subsampling probabilities $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ can still be loaded in available RAM, one can calculate $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ by reading the data from hard drive line-by-line or block-by-block, generate row indexes for a subsample, and then scan the data line-by-line or block-by-block to take the subsample. A detailed procedure is provided in Section A of the appendix.

For Poisson subsampling, the pilot subsample can also be used to construct $\hat{\Psi}_0$ to approximate $\Psi_N(\boldsymbol{\beta})$. We use the following expression to obtain $\hat{\Psi}_0$.

$$\hat{\Psi}_0 = \frac{1}{N}\sum_{i=1}^{n_0^*} \frac{|y_i^{*0} - p(\mathbf{x}_i^{*0}, \hat{\boldsymbol{\beta}}_0)| h(\mathbf{x}_i^{*0})}{(n_0 \pi_{0i}^*) \wedge 1}, \tag{18}$$

where $(\mathbf{x}_i^{*0}, y_i^{*0})$'s are observations in the pilot subsample. If $h(\mathbf{x}) = \|L\mathbf{M}_N^{-1}\mathbf{x}\|$ for some L, then the pilot subsample is used to approximate $\mathbf{M}_N$ through

$$\hat{\mathbf{M}}_0 = \frac{1}{N}\sum_{i=1}^{n_0^*} \frac{\phi_i^{*0}(\hat{\boldsymbol{\beta}}_0)\mathbf{x}_i^{*0}(\mathbf{x}_i^{*0})^{\mathrm{T}}}{(n_0 \pi_{0i}^*) \wedge 1},$$

where $\phi_i^{*0}(\boldsymbol{\beta}) = p(\mathbf{x}_i^{*0}, \boldsymbol{\beta})\{1 - p(\mathbf{x}_i^{*0}, \boldsymbol{\beta})\}$. It can be verified that $\hat{\Psi}_0$ and $\hat{\mathbf{M}}_0$ converge in probability to $\Psi(\boldsymbol{\beta}_t)$ and $\mathbf{M}$, respectively.

Taking into account all aforementioned issues in this section, including how to obtain pilot estimates, how to combine them with the second stage estimates, as well as how to process data file line-by-line, we summarize practical implementation procedures in Algorithm 1 for sampling with replacement and in Algorithm 2 for Poisson subsampling.

Algorithm 1 More efficient estimation based on subsampling with replacement

Step 1: obtain the pilot $\hat{\boldsymbol{\beta}}_0$

(1) Take pilot subsample $(\mathbf{x}_i^{*0}, y_i^{*0})$, $i = 1, ..., n_0$ using sampling with replacement according to subsampling probabilities $\{\pi_{0i}\}_{i=1}^N$ in (17).

(2) Calculate

$$\tilde{\boldsymbol{\beta}}_0 = \arg\max_{\boldsymbol{\beta}} \ell_{uw}^{*0}(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n_0} \left\{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^{*0} y_i^{*0} - \log\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^{*0}}\right)\right\},$$

and let $\hat{\boldsymbol{\beta}}_0 = \tilde{\boldsymbol{\beta}}_0 + \mathbf{b}$, where $\mathbf{b} = \{\log(c_0/c_1), 0, ..., 0\}^{\mathrm{T}}$.

Step 2: obtain the more efficient estimator $\hat{\boldsymbol{\beta}}_{uw}$

(1) Calculate $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ defined in equation (4); take subsample $(\mathbf{x}_i^*, y_i^*)$, $i = 1, ..., n$ according to sampling probabilities $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ using sampling with replacement.

(2) Calculate

$$\tilde{\boldsymbol{\beta}}_{uw} = \arg\max_{\boldsymbol{\beta}} \ell_{uw}^*(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^* y_i^* - \log\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i}\right)\right\},$$

and let $\hat{\boldsymbol{\beta}}_{uw} = \tilde{\boldsymbol{\beta}}_{uw} + \hat{\boldsymbol{\beta}}_0$.

Step 3: combine the two estimators $\hat{\boldsymbol{\beta}}_0$ and $\hat{\boldsymbol{\beta}}_{uw}$

Calculate

$$\check{\boldsymbol{\beta}}_{uw} = \{\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw})\}^{-1}\{\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0)\hat{\boldsymbol{\beta}}_0 + \ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw})\hat{\boldsymbol{\beta}}_{uw}\},$$

where $\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0) = \sum_{i=1}^{n_0} \phi_i^{*0}(\tilde{\boldsymbol{\beta}}_0)\mathbf{x}_i^{*0}(\mathbf{x}_i^{*0})^{\mathrm{T}}$, $\ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw}) = \sum_{i=1}^{n} \phi_i^*(\tilde{\boldsymbol{\beta}}_{uw})\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}}$, and $\phi_i^*(\boldsymbol{\beta}) = p(\mathbf{x}_i^*, \boldsymbol{\beta})\{1 - p(\mathbf{x}_i^*, \boldsymbol{\beta})\}$. .

The variance-covariance matrix of $\check{\boldsymbol{\beta}}_{uw}$ can be estimated by

$$\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{uw}) = \{\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw})\}^{-1}\left[\sum_{i=1}^{n_0}\{\psi_i^{*0}(\tilde{\boldsymbol{\beta}}_0)\}^2\mathbf{x}_i^{*0}(\mathbf{x}_i^{*0})^{\mathrm{T}} + \sum_{i=1}^{n}\{\psi_i^*(\tilde{\boldsymbol{\beta}}_{uw})\}^2\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}}\right]\{\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw})\}^{-1}, \tag{19}$$

where $\psi_i^*(\boldsymbol{\beta}) = y_i^* - p(\mathbf{x}_i^*, \boldsymbol{\beta})$.

Algorithm 2 More efficient estimation based on Poisson subsampling

Step 1: obtain the pilots $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$

(1) For $i = 1, ..., N$, calculate $\pi_{0i} = \frac{c_0(1-y_i)+c_1 y_i}{N}$, generate $u_{0i} \sim U(0,1)$, and add $(\mathbf{x}_i, y_i, \pi_{i1})$ in the subsample if $u_{0i} \le n_0 \pi_{0i}$.

(2) For the obtained subsample, say $(\mathbf{x}_i^{*0}, y_i^{*0}, \pi_{0i}^{*0})$, $i = 1, ..., n_0^*$, calculate

$$\tilde{\boldsymbol{\beta}}_0 = \arg\max_{\boldsymbol{\beta}} \ell_p^{*0}(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n_0^*} (n\pi_{0i}^{*0} \vee 1)\{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^{*0} y_i^{*0} + \log\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^{*0}}\right)\},$$

let $\hat{\boldsymbol{\beta}}_0 = \tilde{\boldsymbol{\beta}}_0 + \mathbf{b}$, and then calculate $\hat{\Psi}_0$ in equation (18).

Step 2: obtain the more efficient estimator $\hat{\boldsymbol{\beta}}_p$

(1) For $i = 1, ..., N$, calculate $\pi_i^p = \frac{|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)| h(\mathbf{x}_i)}{N\hat{\Psi}_0}$, generate $u_i \sim U(0,1)$, and if $u_i \le n\pi_i^p$ add $(\mathbf{x}_i, y_i, \pi_i^p)$ in the subsample.

(2) For the obtained subsample, say $\{(\mathbf{x}_1^*, y_1^*, \pi_1^{p*}), ..., (\mathbf{x}_{n^*}^*, y_{n^*}^*, \pi_{n^*}^{p*})\}$, calculate

$$\tilde{\boldsymbol{\beta}}_p = \arg\max_{\boldsymbol{\beta}} \ell_p^*(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n^*} (n\pi_i^{p*} \vee 1)\{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^* y_i^* + \log(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i^*})\},$$

and let $\hat{\boldsymbol{\beta}}_p = \tilde{\boldsymbol{\beta}}_p + \hat{\boldsymbol{\beta}}_0$.

Step 3: combine the two estimators $\hat{\boldsymbol{\beta}}_0$ and $\hat{\boldsymbol{\beta}}_p$

Calculate

$$\check{\boldsymbol{\beta}}_p = \{\ddot{\ell}_p^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_p^*(\tilde{\boldsymbol{\beta}}_p)\}^{-1}\{\ddot{\ell}_p^{*0}(\tilde{\boldsymbol{\beta}}_0)\hat{\boldsymbol{\beta}}_0 + \ddot{\ell}_p^*(\tilde{\boldsymbol{\beta}}_p)\hat{\boldsymbol{\beta}}_p\},$$

where $\ddot{\ell}_p^{*0}(\tilde{\boldsymbol{\beta}}_0) = \sum_{i=1}^{n_0^*} \phi_i^{*0}(\tilde{\boldsymbol{\beta}}_0)\mathbf{x}_i^{*0}(\mathbf{x}_i^{*0})^{\mathrm{T}}$ and $\ddot{\ell}_p^*(\tilde{\boldsymbol{\beta}}_p) = \sum_{i=1}^{n^*} \phi_i^*(\tilde{\boldsymbol{\beta}}_p)\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}}$.
The variance-covariance matrix of $\check{\boldsymbol{\beta}}_p$ can be estimated by

$$\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_p) = \{\ddot{\ell}_p^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_p^*(\tilde{\boldsymbol{\beta}}_p)\}^{-1}\Bigg[\sum_{i=1}^{n_0^*}\{\psi_i^{*0}(\tilde{\boldsymbol{\beta}}_0)\}^2\mathbf{x}_i^{*0}(\mathbf{x}_i^{*0})^{\mathrm{T}} + \sum_{i=1}^{n^*}\{\psi_i^*(\tilde{\boldsymbol{\beta}}_p)\}^2\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}}\Bigg]\{\ddot{\ell}_p^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_p^*(\tilde{\boldsymbol{\beta}}_p)\}^{-1}. \quad (20)$$

Remark 9 *In Algorithm 1 and Algorithm 2, if $n_0 = o(n)$, then the results for $\hat{\boldsymbol{\beta}}_{uw}$ in Theorem 1 hold for $\check{\boldsymbol{\beta}}_{uw}$ and the results for $\hat{\boldsymbol{\beta}}_p$ in Theorem 6 hold for $\check{\boldsymbol{\beta}}_p$ as well. This is because $\{\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw})\}^{-1}\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0)\sqrt{n}(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_t) = O_p(\sqrt{n_0}/\sqrt{n}) = o_P(1)$ and $\{\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw})\}^{-1}\ddot{\ell}_{uw}(\tilde{\boldsymbol{\beta}}_{uw}) \to 1$ in probability. The reason for $\check{\boldsymbol{\beta}}_p$ is similar.*

Remark 10 *In Algorithm 1 and Algorithm 2, to combine the two stage estimates using the second derivative of the objective functions, the inconsistent estimators $\tilde{\boldsymbol{\beta}}_0$, and $\tilde{\boldsymbol{\beta}}_{uw}$ or $\tilde{\boldsymbol{\beta}}_p$ should be used, because their limits correspond to the terms in the asymptotic variance-covariance matrices of the more efficient estimators. This is an advantage of the proposed estimators for implementation using existing software that fit logistic regression. One can use the inverse of the estimated variance-covariance matrix from the software output to replace the second derivative of the objective function.*

Remark 11 *The variance-covariance estimators $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{uw})$ in (19) and $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_p)$ in (20) can be replaced by the following simplified estimators,*

$$\hat{\mathbb{V}}_s(\check{\boldsymbol{\beta}}_{uw}) = \{\ddot{\ell}_{uw}^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_{uw}^*(\tilde{\boldsymbol{\beta}}_{uw})\}^{-1} \qquad \text{and} \qquad \hat{\mathbb{V}}_s(\check{\boldsymbol{\beta}}_p) = \{\ddot{\ell}_p^{*0}(\tilde{\boldsymbol{\beta}}_0) + \ddot{\ell}_p^*(\tilde{\boldsymbol{\beta}}_p)\}^{-1},$$

respectively. If the subsampling ratio n/N is much smaller than one, then $\hat{\mathbb{V}}_s(\check{\boldsymbol{\beta}}_{uw})$ and $\hat{\mathbb{V}}_s(\check{\boldsymbol{\beta}}_p)$ perform very similarly to $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{uw})$ and $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_p)$, respectively.

Remark 12 *The time complexity of Algorithm 1 is the same as that of Algorithm 2 in Wang et al. (2018). The major computing time is to calculate $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ in Step 2, but it does not require iterative calculations on the full data. Once $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ are available, the calculations of $\hat{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_{uw}$ are fast because they are done on the subsamples only. To calculate $\{\pi_i^{\mathrm{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$, the required time varies. For π_i^{Lopt}, the required time is $O(Nd)$; for π_i^{Aopt}, the required time is $O(Nd^2)$. Thus, the time complexity of Algorithm 1 with π_i^{Lopt} is $O(Nd)$ and the time complexity with π_i^{Aopt} is $O(Nd^2)$, if the sampling ratio n/N is much smaller than one.*

6. Unconditional distribution

Asymptotic distributional results in Sections 3 and 4, as well as in Wang et al. (2018), are about conditional distributions, i.e., they are about conditional distributions of subsample-based estimators given the full data. We investigate the unconditional distribution of $\hat{\boldsymbol{\beta}}_p$ in this section.

Theorem 13 *Under Assumptions 1 and 2, if the pilot estimators are obtained from a uniform subsample of sample size $n_0 = o(\sqrt{N})$ and $\mathbb{E}\{h^3(\mathbf{x})\|\mathbf{x}\|^3\}$, $\mathbb{E}\{h^3(\mathbf{x})\|\mathbf{x}\|^2\}$, $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^3\}$, and $\mathbb{E}\{h^2(\mathbf{x})\}$ are finite, or if $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ are independent of the data $\mathcal{D}_N$, then as n_0, n, and N go to infinity such that $n/N \to \rho \in [0,1)$, we have*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}(0,\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_u\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}), \quad (21)$$

in distribution, where

$$\boldsymbol{\Lambda}_u = \frac{\mathbb{E}[|\psi(\boldsymbol{\beta}_t)|\{\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x}) \vee \Psi(\boldsymbol{\beta}_t)\}h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]}{4\Psi^2(\boldsymbol{\beta}_t)}.$$

Remark 14 *If the pilot estimators $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ are obtained through the full data $\mathcal{D}_N$, stronger moment conditions are required. Note that $h(\mathbf{x})$ is often a function of the norm of $\mathbf{x}$, such as in π_i^{Lopt}, π_i^{Aopt}, and the local case-control subsampling. In general, if $h(\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|^a$ for some matrix $\mathbf{A}$ and constant $a \geq 0$, then the four additional moment conditions reduce to one requirement of $\mathbb{E}\{h^3(\mathbf{x})\|\mathbf{x}\|^3\} < \infty$.*

Remark 15 *If $\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x}) \leq \Psi(\boldsymbol{\beta}_t)$ almost surely, then $\boldsymbol{\Lambda}_u$ reduced to $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}$ and as a result $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_u\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ reduces to $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$. Furthermore, if the subsampling probabilities are propositional to the local case-control subsampling probabilities, i.e., $h(\mathbf{x}) = 1$, then $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_u\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ reduces to $4\mathbb{E}\{\phi(\boldsymbol{\beta})\}\mathbf{M}^{-1}$. For the uniform Poisson subsampling estimator, the unconditional asymptotic variance-covariance matrix (scaled by n) is $\mathbf{M}^{-1}$. From (14), with the same expected subsample size, the proposed method has a higher estimation efficiency than subsampling proportional to the local case-control subsampling probabilities, which is more efficient than the uniform Poisson subsampling approach.*

Remark 16 *Fithian and Hastie (2014)'s investigation corresponds to the case of $h(\mathbf{x}) = 1$ and $\rho = 2\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}$. For this scenario in Theorem 13, the asymptotic variance-covariance matrix of $\hat{\boldsymbol{\beta}}_p$ reduces to $2N^{-1}\mathbf{M}^{-1}$, which is the same as obtained in Fithian and Hastie (2014). This result is particularly neat in the fact that this asymptotic variance-covariance matrix is proportional to that from the full data MLE with a multiplier of 2. The result in Theorem 13 is more general. It shows that if $h(\mathbf{x}) = 1$, then as long as $\rho|\psi(\boldsymbol{\beta}_t)| \leq 2\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}$ (which is satisfied if $\rho = 2\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}$), the asymptotic variance-covariance matrix of $\hat{\boldsymbol{\beta}}_p$ can be written as*

$$\frac{4\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}}{\rho N}\mathbf{M}^{-1},$$

which is proportional to that of the full data MLE with a multiplier of $4\rho^{-1}\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}$. We need to emphasize that this simple representation holds only when $\rho|\psi(\boldsymbol{\beta}_t)| \leq 2\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}$ almost surely. If the subsampling ratio ρ gets closer to one, the asymptotic variance-covariance matrix in (21) may not be simplified.

From Theorems 6 and 13, the conditional asymptotic distribution and unconditional asymptotic distribution of $\hat{\boldsymbol{\beta}}_p$ are the same if $n/N \to 0$. This is intuitive, because if the sampling ratio n/N is small, the variation of $\hat{\boldsymbol{\beta}}_p$ due to the variation of the full data is small compared with the variation due to the variation of the subsampling. However, if the sampling ratio n/N does not converge to zero, then the conditional asymptotic distribution and unconditional asymptotic distribution of $\hat{\boldsymbol{\beta}}_p$ are quite different. First, we notice that under the unconditional distribution, $\hat{\boldsymbol{\beta}}_p$ is asymptotically unbiased to $\boldsymbol{\beta}_t$, while under the conditional distribution, $\hat{\boldsymbol{\beta}}_p$ is asymptotically biased with the bias being $\hat{\boldsymbol{\beta}}_{\text{wMLE}} - \boldsymbol{\beta}_t = O_P(N^{-1/2})$. Second, since the variation of $\hat{\boldsymbol{\beta}}_p$ due to the variation of the full data is not negligible, we expect that the asymptotic variance-covariance matrix for the unconditional distribution to be larger than that for the conditional distribution. Indeed this is true, and we present it in the following proposition.

Proposition 17 *If $\rho > 0$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ is a finite and positive definite matrix, then*

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_u\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} \geq \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} > \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_\rho\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}, \tag{22}$$

under the Loewner ordering. Furthermore, if $\mathbb{P}\{\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x}) > \Psi(\boldsymbol{\beta}_t)\} > 0$, then the "$\geq$" sign in (22) can be replaced by "$>$", the strict great sign.

Fithian and Hastie (2014) obtained unconditional distribution of local case-control estimator by assuming that the pilot estimate is independent of the data. Our Theorem 13 includes this scenario, and the required assumptions are the same as those required in Fithian and Hastie (2014). In practice, a consistent pilot estimator that is independent of the data may not be available and a pilot subsample from the full data is required to construct it. For this scenario, a pilot estimator is dependent on the data, and we need a stronger moment condition to establish the asymptotic normality. For local case-control subsampling, $h(\mathbf{x}) = 1$, and the additional moment requirement is that $\mathbb{E}(\|\mathbf{x}\|^3) < \infty$.

7. Misspecifications

In this section, we discuss the effect when the pilot estimates are misspecified or when the model is misspecified. Pilot estimates misspecification often occurs when they are from other data sources or when they are calculated based on convenient subsamples, e.g., using the first n_0 observations in the full data to calculate them. In these cases, it is reasonable to assume that the pilot estimates are independent of $\mathcal{D}_N$ and we use this assumption in this section.

7.1. Pilot estimates misspecification

Here, we assume that the model is correctly specified but the pilot estimates $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ converge to limits that are different from the true parameters for the current data. Interestingly, in this case, the proposed estimators are still consistent and no specific convergence rate is required for $\hat{\boldsymbol{\beta}}_0$ or $\hat{\Psi}_0$.

The following theorem describes the asymptotic distribution of $\hat{\boldsymbol{\beta}}_{uw}$, the estimator based on subsampling with replacement. Note that $\hat{\Psi}_0$ is not required by $\hat{\boldsymbol{\beta}}_{uw}$.

Theorem 18 *When the logistic regression model in (1) is correctly specified and the pilot estimator $\hat{\boldsymbol{\beta}}_0$ that is independent of $\mathcal{D}_N$ is inconsistent, i.e., $\hat{\boldsymbol{\beta}}_0 \to \boldsymbol{\beta}_0$ in probability for some $\boldsymbol{\beta}_0$ that is different from $\boldsymbol{\beta}_t$, then under Assumptions 1-3, conditional on $\mathcal{D}_N$, as n, and N go to infinity,*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\text{wMLE}}) \longrightarrow \mathbb{N}\big\{\mathbf{0},\ \Psi(\boldsymbol{\beta}_0)\boldsymbol{\varsigma}_a^{-1}\big\},$$

in distribution; furthermore, if $n/N \to 0$, then

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}\big\{\mathbf{0},\ \Psi(\boldsymbol{\beta}_0)\boldsymbol{\varsigma}_a^{-1}\big\},$$

in distribution, where $\Psi(\boldsymbol{\beta}_0) = \mathbb{E}\{|\psi(\boldsymbol{\beta}_0)|h(\mathbf{x})\}$ and

$$\boldsymbol{\varsigma}_a = \mathbb{E}[\{1 - p(\mathbf{x}, \boldsymbol{\beta}_t)\}p(\mathbf{x}, \boldsymbol{\beta}_0)p(\mathbf{x}, \boldsymbol{\beta}_t - \boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}].$$

Here $\hat{\boldsymbol{\beta}}_{\text{wMLE}}$ satisfies that,

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\text{wMLE}} - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\varsigma}_a^{-1}\boldsymbol{\varsigma}_b\boldsymbol{\varsigma}_a^{-1}\big),$$

in distribution, where

$$\varsigma_b = \mathbb{E}\{\phi(\boldsymbol{\beta}_0)\phi(\boldsymbol{\beta}_t - \boldsymbol{\beta}_0)h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}.$$

Remark 19 *If $\boldsymbol{\beta}_0 = \boldsymbol{\beta}_t$, then direct calculations show that $\Psi(\boldsymbol{\beta}_0)\varsigma_a^{-1} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$ and $\varsigma_a^{-1}\varsigma_b\varsigma_a^{-1} = \boldsymbol{\Sigma}_{\mathrm{wMLE}}$, that is, the results in Theorem 18 reduce to the same results in Theorem 1.*

Remark 20 *If the pilot estimator $\hat{\boldsymbol{\beta}}_0$ is very wrong such that $\boldsymbol{\beta}_t^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 < 0$, i.e., $p(\mathbf{x}, \boldsymbol{\beta}_t) > 0.5 > p(\mathbf{x}, \boldsymbol{\beta}_0)$ or $p(\mathbf{x}, \boldsymbol{\beta}_t) < 0.5 < p(\mathbf{x}, \boldsymbol{\beta}_0)$, then it can be shown that*

$$\Psi(\boldsymbol{\beta}_0)\varsigma_a^{-1} > \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}.$$

Detailed proof for this result is presented in Section B.4.1 of the appendix.

The following theorem describes the asymptotic distribution of $\hat{\boldsymbol{\beta}}_p$, the estimator based on Poisson subsampling. Note that $\hat{\boldsymbol{\beta}}_p$ requires both $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$.

Theorem 21 *Assume that the logistic regression model is correctly specified, and the pilot estimators $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ are independent of $\mathcal{D}_N$ and they are inconsistent, i.e., $\hat{\boldsymbol{\beta}}_0 \to \boldsymbol{\beta}_0$ and $\hat{\Psi}_0 \to \Psi_0$ in probability for some $\boldsymbol{\beta}_0$ and Ψ_0, respectively. Under Assumptions 1-2, conditional on $\mathcal{D}_N$, as n and N go to infinity, if $n/N \to 0$, then*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \Psi_0\varsigma_a^{-1}\big),$$

in distribution; if $n/N \to \rho$, then

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \Psi_0\varsigma_a^{-1}\varsigma_c\varsigma_a^{-1}\big),$$

in distribution, where

$$\varsigma_c = \mathbb{E}\Big[|\psi(\boldsymbol{\beta}_0)|\big\{1 - \rho\Psi_0^{-1}|\psi(\boldsymbol{\beta}_0)|h(\mathbf{x})\big\}_+\psi^2(\boldsymbol{\beta}_t - \boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\Big].$$

Remark 22 *If $\boldsymbol{\beta}_0 = \boldsymbol{\beta}_t$ and $\Psi_0 = \Psi(\boldsymbol{\beta}_t)$, then direct calculations show that $\Psi_0^{-1}\varsigma_c = \boldsymbol{\Lambda}_\rho$, and thus the results in Theorem 21 reduce to the same results in Theorem 6.*

Remark 23 *We have a result similar to that in Proposition 8. By direct calculation, we know that*

$$\varsigma_c < \mathbb{E}\Big[|\psi(\boldsymbol{\beta}_0)|\psi^2(\boldsymbol{\beta}_t - \boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\Big] = \varsigma_a,$$

under the Loewner ordering, which indicates that

$$\Psi_0\varsigma_a^{-1}\varsigma_c\varsigma_a^{-1} < \Psi_0\varsigma_a^{-1}.$$

Thus, when the pilot estimators are misspecified, Poisson subsampling still has a higher estimation efficiency compared with subsampling with replacement if $\Psi_0 = \mathbb{E}\{|\psi(\boldsymbol{\beta}_0)|h(\mathbf{x})\}$, which is the case if $\hat{\Psi}_0$ is constructed from a pilot subsample.

7.2. Model misspecification

In this section, we consider the case when the logistic regression model is misspecified, namely, the model in (1) is not correct. Instead, we assume that the true probability of $y = 1$ given $\mathbf{x}$ is

$$\mathbb{P}(y = 1|\mathbf{x}) = p_t(\mathbf{x}),$$

for some unknown function $p_t(\mathbf{x})$. When the logistic regression model is misspecified, we need to define the meaning of consistency because there is no true $\boldsymbol{\beta}$ any more. In this case, consistency often means that the estimator converges to a limit that minimizes expected loss with respect to a specified loss function. Here, if we denote the limit as $\boldsymbol{\beta}_l$ and define it to be the minimizer of

$$\mathbb{E}\Big\{-p_t(\mathbf{x})h(\mathbf{x})\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} + h(\mathbf{x})\log\big(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}}\big)\Big\},$$

then $\boldsymbol{\beta}_l$ satisfies

$$\mathbb{E}\big[\{p_t(\mathbf{x}) - p(\mathbf{x}, \boldsymbol{\beta}_l)\}h(\mathbf{x})\mathbf{x}\big] = \mathbf{0},$$

where $p(\mathbf{x}, \boldsymbol{\beta}) = e^{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}}(1 + e^{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}})^{-1}$.

Now we investigate the asymptotic properties of the proposed estimators under model misspecification. In this case, we need to assume that the pilot estimators are consistent which is also required in the local case-control subsampling method. In addition, to investigate the asymptotic normality, we also need an additional assumption on the convergence rate of the pilot estimator $\hat{\boldsymbol{\beta}}_0$.

The following theorem describes the asymptotic behavior of the estimator $\hat{\boldsymbol{\beta}}_{uw}$ based on subsampling with replacement.

Theorem 24 *Assume that the pilot sample is independent of $\mathcal{D}_N$ and the pilot estimator $\hat{\boldsymbol{\beta}}_0$ satisfies that $\sqrt{n_0}(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_l) \to N(\mathbf{0}, \boldsymbol{\Sigma}_0)$ in distribution. Under Assumptions 1-3, if $n_0/N \to \rho_0$ and $n/N \to \rho$ with $\rho_0, \rho \in (0, 1)$, then conditional on $\mathcal{D}_N$, as n_0, n, and N go to infinity,*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \omega\boldsymbol{\kappa}_a^{-1}\big) \tag{23}$$

in distribution, where

$$\boldsymbol{\kappa}_a = \frac{1}{4}\mathbb{E}\big[\{p_t(\mathbf{x}) - 2p_t(\mathbf{x})p(\mathbf{x}, \boldsymbol{\beta}_l) + p(\mathbf{x}, \boldsymbol{\beta}_l)\}h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big],$$
$$\omega = \mathbb{E}\big[\{p_t(\mathbf{x}) - 2p_t(\mathbf{x})p(\mathbf{x}, \boldsymbol{\beta}_l) + p(\mathbf{x}, \boldsymbol{\beta}_l)\}h(\mathbf{x})\big],$$

and $\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}$ satisfies that

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_l) \longrightarrow \mathbb{N}\Big\{\mathbf{0},\ \boldsymbol{\kappa}_a^{-1}\big(\boldsymbol{\kappa}_b + \rho_0^{-1}\boldsymbol{\kappa}_c\boldsymbol{\Sigma}_0\boldsymbol{\kappa}_c\big)\boldsymbol{\kappa}_a^{-1}\Big\}, \tag{24}$$

in distribution, with

$$\boldsymbol{\kappa}_b = \frac{1}{4}\mathbb{E}\big[\{p_t(\mathbf{x}) - 2p_t(\mathbf{x})p(\mathbf{x}, \boldsymbol{\beta}_l) + p^2(\mathbf{x}, \boldsymbol{\beta}_l)\}h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big],\ \textit{and}$$
$$\boldsymbol{\kappa}_c = \frac{1}{4}\mathbb{E}\big[\{1 - 2p(\mathbf{x}, \boldsymbol{\beta}_l)\}\{p_t(\mathbf{x}) - p(\mathbf{x}, \boldsymbol{\beta}_l)\}h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big].$$

Remark 25 *If the model is correctly specified, i.e.,* $p_t(\mathbf{x}) = p(\mathbf{x}, \boldsymbol{\beta}_t)$*, then* $\omega\boldsymbol{\kappa}_a^{-1} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$*,* $\boldsymbol{\kappa}_b = \frac{1}{4}\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$ *and* $\boldsymbol{\kappa}_c = \mathbf{0}$*, and therefore the results in Theorem 24 reduce to the same expressions as those in Theorem 1.*

From Theorem 24, with model misspecification, it is critical to have a good pilot estimator $\hat{\boldsymbol{\beta}}_0$. Note that the pilot sample size is typically much smaller than the full data sample size, so ρ_0 can be close to zero. From (24), we see that the asymptotic variance-covariance matrix of $\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}$ can be inflated by a small pilot sample size.

The following theorem presents asymptotic results for the estimator based on Poisson subsampling.

Theorem 26 *Assume that the pilot sample is independent of* $\mathcal{D}_N$ *and the pilot estimators satisfy that* $\sqrt{n_0}(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_l) \to N(\mathbf{0}, \boldsymbol{\Sigma}_0)$ *in distribution and* $\hat{\Psi}_0 \to \omega$ *in probability. Under Assumptions 1-2, if* $n_0/N \to \rho_0$ *and* $n/N \to \rho$ *with* $\rho_0, \rho \in (0,1)$*, then conditional on* $\mathcal{D}_N$*, as* n_0*,* n*, and* N *go to infinity,*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}) \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\kappa}_a^{-1}\boldsymbol{\kappa}_d\boldsymbol{\kappa}_a^{-1}\big).$$

in distribution, where $\boldsymbol{\kappa}_a$ *and* $\boldsymbol{\kappa}_b$ *are defined in Theorem 24, and*

$$\boldsymbol{\kappa}_d = \frac{1}{4}\mathbb{E}\Big[|\psi(\boldsymbol{\beta}_l)|\{1 - \rho\omega^{-1}|\psi(\boldsymbol{\beta}_l)|h(\mathbf{x})\}_+ h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\Big].$$

Remark 27 *Similarly to Proposition 8, we have that*

$$\boldsymbol{\kappa}_a^{-1}\boldsymbol{\kappa}_d\boldsymbol{\kappa}_a^{-1} < \boldsymbol{\kappa}_a^{-1}$$

under the Loewner ordering, indicating that the estimator based on Poisson subsampling has a smaller conditional variance-covariance matrix.

For Poisson subsampling, compared with $\hat{\boldsymbol{\beta}}_0$, we require a much weaker assumption on $\hat{\Psi}_0$; we only need it to converge without specifying certain convergence rate. The reason is that the effect of $\hat{\Psi}_0$ on all the subsampling probabilities are the same and it mainly controls the expected subsample size, while $\hat{\boldsymbol{\beta}}_0$ affects individual subsampling probabilities differently corresponding to different values of $\mathbf{x}_i$ and y_i.

8. Numerical evaluations

We evaluate the performance of the more efficient estimators in terms of both estimation efficiency and computational efficiency in this section.

8.1. Estimation efficiency

In this section, we use numerical experiments based on simulated and real data sets to evaluate the estimators proposed in this paper. For simulation, to compare with the original OSMAC estimator, we use exactly the same setup used in Section 5.1 of Wang et al. (2018). Specifically, the full data sample size $N = 10,000$ and the true value of $\boldsymbol{\beta}$, $\boldsymbol{\beta}_t$, is a 7×1 vector of 0.5. The following 6 distributions of $\mathbf{x}$ are considered: multivariate

normal distribution with mean zero (mzNormal), multivariate normal distribution with nonzero mean (nzNormal), multivariate normal distribution with mean zero and unequal variances (ueNormal), mixture of two multivariate normal distributions with different means (mixNormal), multivariate t distribution with degrees of freedom 3 (T_3), and exponential distribution (EXP). Detailed explanations of these distributions can be found in Section 5.1 of Wang et al. (2018).

To evaluate the estimation performance of the new estimators compared with the original weighted OSMAC estimator, we define the estimation efficiency of $\check{\boldsymbol{\beta}}_{\mathrm{new}}$ relative to $\check{\boldsymbol{\beta}}_w$ as

$$\text{Relative Efficiency} = \frac{\mathrm{MSE}(\check{\boldsymbol{\beta}}_w)}{\mathrm{MSE}(\check{\boldsymbol{\beta}}_{\mathrm{new}})},$$

where $\check{\boldsymbol{\beta}}_{\mathrm{new}} = \check{\boldsymbol{\beta}}_{uw}$ for the subsampling with replacement estimator described in Algorithm 1 and $\check{\boldsymbol{\beta}}_{\mathrm{new}} = \check{\boldsymbol{\beta}}_p$ for Poisson subsampling estimator described in Algorithm 2. We calculate empirical MSEs from $S = 1000$ subsamples using

$$\mathrm{MSE}(\check{\boldsymbol{\beta}}) = \frac{1}{S}\sum_{s=1}^{S}\|\check{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}_t\|^2, \tag{25}$$

where $\check{\boldsymbol{\beta}}^{(s)}$ is the estimate from the s-th subsample. We fixed the first step sample size $n_0 = 200$ and choose n to be 100, 200, 400, 600, 800, and 1000. This is the same setup used in Wang et al. (2018).

Figure 1 presents the relative efficiency of $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_p$ based on two different choices of π_i^{OS}: π_i^{Aopt} and π_i^{Lopt}. It is seen that in general $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_p$ are more efficient than $\check{\boldsymbol{\beta}}_w$. Among the six cases, the only case that $\check{\boldsymbol{\beta}}_w$ can be more efficient is when $\mathbf{x}$ has a T_3 distribution and π^{Lopt} is used, but the difference is not very significant. For all other cases, $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_p$ are more efficient. For example, when $\mathbf{x}$ has the nzNormal distribution, $\check{\boldsymbol{\beta}}_p$ can be 250% as efficient as $\check{\boldsymbol{\beta}}_w$ if π^{Aopt} is used. Between $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_p$, $\check{\boldsymbol{\beta}}_p$ is more efficient than $\check{\boldsymbol{\beta}}_{uw}$ for all cases. We also calculate the empirical unconditional MSE by generating the full data in each repetition of the simulation. The results are similar and thus are omitted.

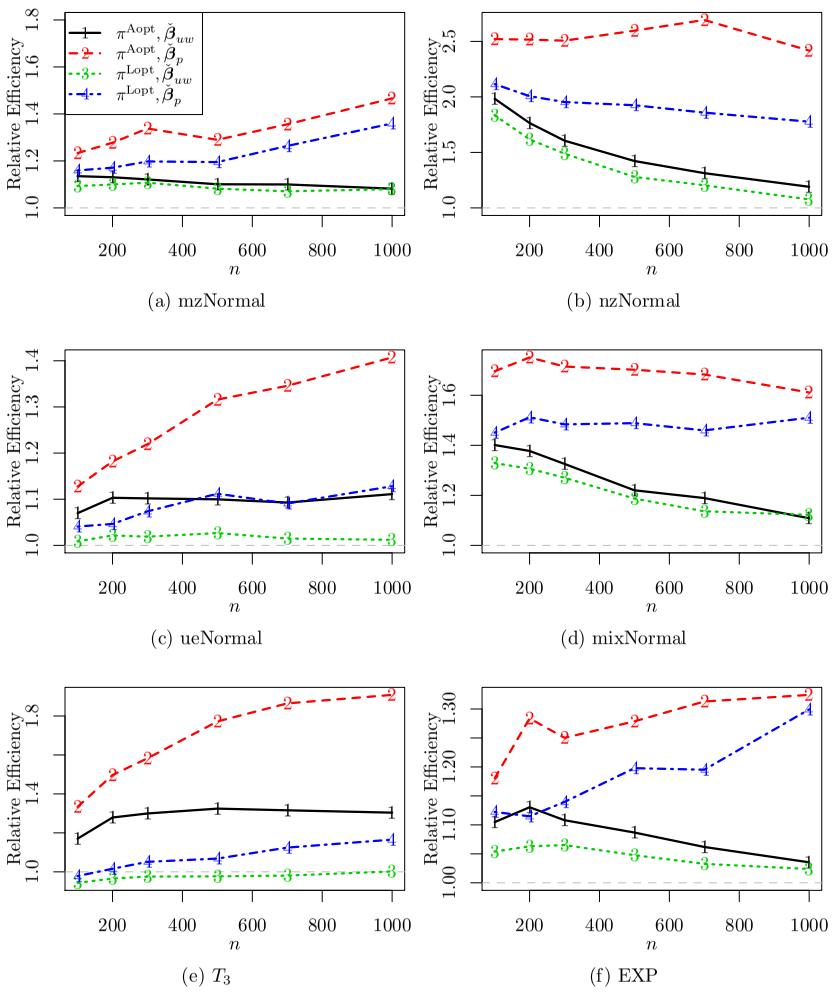
To evaluate the performance of the proposed method with different choices of the subsampling probabilities for subsampling with replacement and Poisson subsampling, Figure 2 plots empirical MSEs of using π^{Aopt}, π^{Lopt}, π^{lcc} (local case-control), and the uniform subsampling probability. In general, π^{Aopt} with Poisson subsampling has the smallest empirical MSEs while uniform subsampling with replacement has the worst estimation efficiency. This agrees with our theoretical results: 1) π^{Aopt} minimizes the asymptotic MSE of the parameter estimator which corresponds to the empirical MSE defined in (25) for the experiments, while π^{Lopt} minimizes the asymptotic MSE of a transformed parameter estimator, and 2) Poisson subsampling has a higher estimation efficiency compared with subsampling with replacement.

To assess the performance of $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{uw})$ in (19) and $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_p)$ in (20), we use $\mathrm{tr}\{\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{uw})\}$ and $\mathrm{tr}\{\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_p)\}$ to estimate the MSEs of $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_p$, and compare the average estimated MSEs with the unconditional empirical MSEs. We focus on the unconditional MSE because conditional inference may not be appropriate if n/N is not small enough. Figure 3 presents the results for using π^{Lopt}. Results for using π^{Aopt} are similar and thus are omitted. Note

that our purpose here is to evaluate the quality of $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{uw})$ in (19) and $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{p})$ in (20), so in this figure we plot the original empirical and estimated MSEs without scaling then using the MSEs of $\check{\boldsymbol{\beta}}_{w}$. Here, a closer value between the estimated MSE and the empirical MSE indicates a better performance of $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{uw})$ or $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{p})$. From Figure 3, the estimated MSEs are very close to the empirical MSEs, except for the case of nzNormal covariate for subsampling with replacement. In this case, the responses are imbalanced with about 95% being 1's. For this scenario, the variance-covariance estimator for $\check{\boldsymbol{\beta}}_{w}$ proposed in Wang et al. (2018) also has a similar problem of underestimation. For Poisson subsampling, the problem of underestimation from $\hat{\mathbb{V}}(\check{\boldsymbol{\beta}}_{p})$ is not significant.

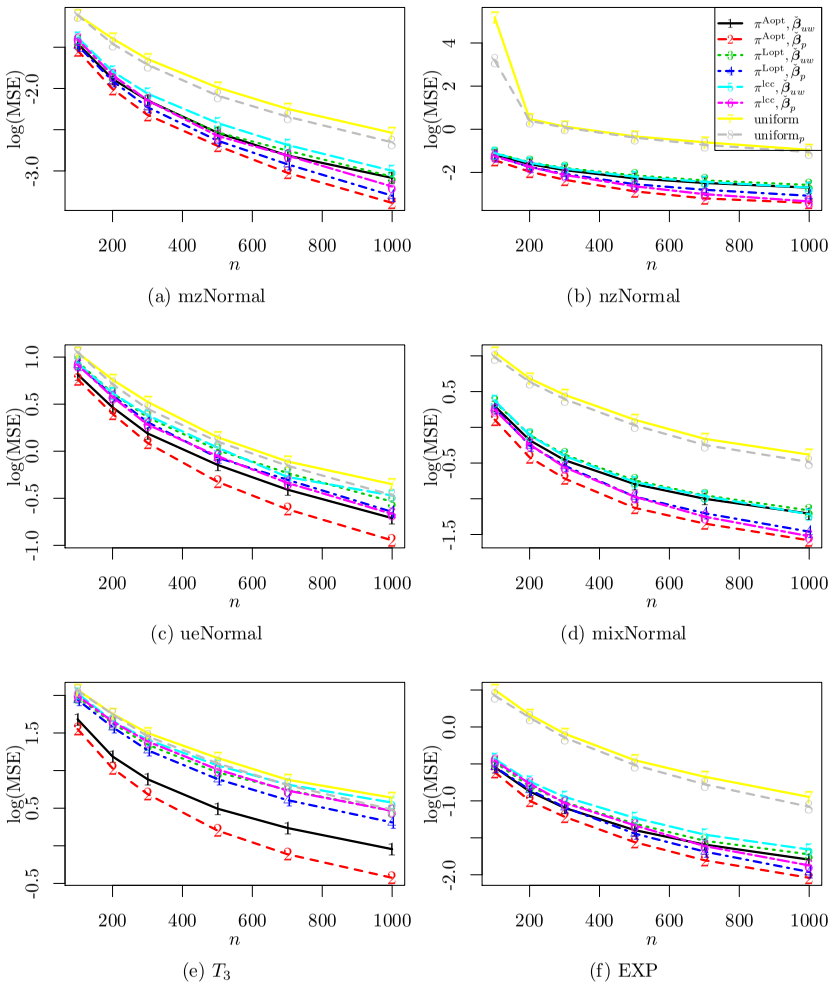
We also apply the more efficient estimation methods to a supersymmetric (SUSY) benchmark data set (Baldi et al., 2014) available from the Machine Learning Repository (Dua and Karra Taniskidou, 2017). The data set contains a binary response variable indicating whether a process produces new supersymmetric particles or not and 18 covariates that are kinematic features about the process. The full sample size is $N = 5,000,000$ and the data file is about 2.4 gigabytes. About 54.24% of the responses in the full data are from the background process. We use the more efficient estimation methods with subsample size n to estimate parameters in logistic regression.

Figures 4 gives the relative efficiency of $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_{p}$ to $\check{\boldsymbol{\beta}}_{w}$ for both π_i^{Lopt} and π_i^{Aopt}. It is seen that when π_i^{Aopt} are used, $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_{p}$ always outperform $\check{\boldsymbol{\beta}}_{w}$. When π_i^{Lopt} are used, $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_{p}$ may not be as efficient as $\check{\boldsymbol{\beta}}_{w}$, but they become more efficient when the second stage sample size n gets larger. It is also seen that $\check{\boldsymbol{\beta}}_{p}$ dominates $\check{\boldsymbol{\beta}}_{uw}$ and π^{Aopt} dominates π^{Lopt} in estimation efficiency.



(a) mzNormal (b) nzNormal (c) ueNormal (d) mixNormal (e) T_3 (f) EXP

Figure 1: Relative efficiency for different second step subsample size n with the first step subsample size being fixed at $n_0 = 200$. A relative efficiency larger than one means the associate method is more efficient than the original OSMAC estimator.

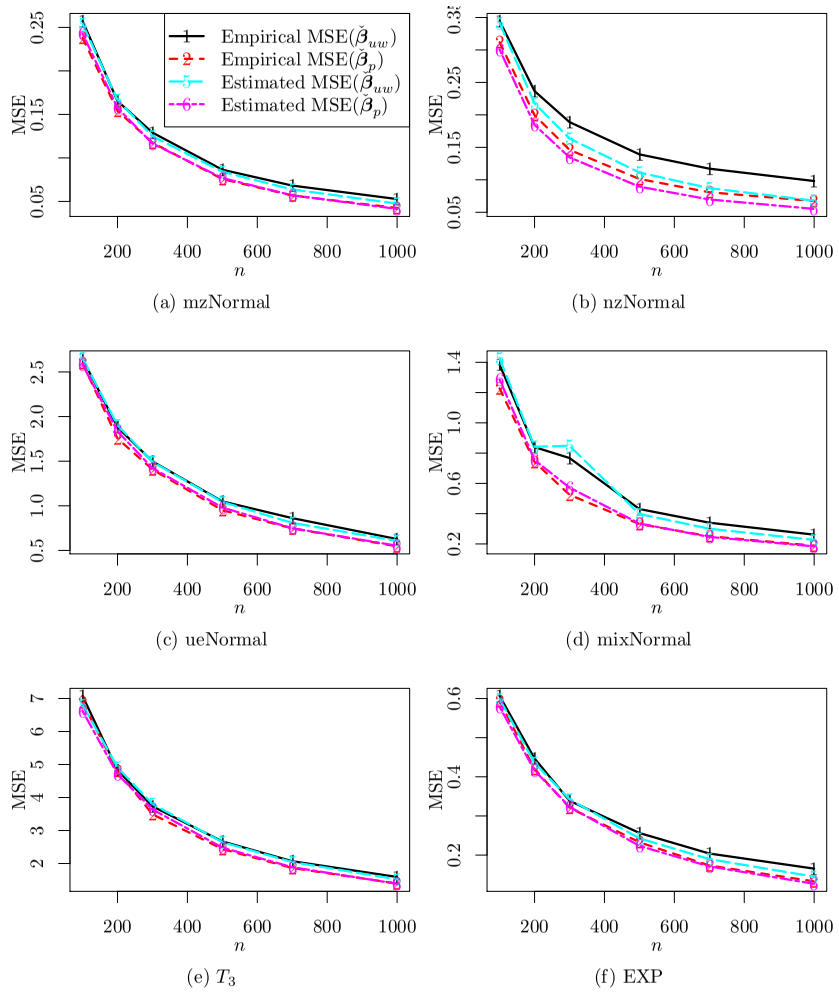


(a) mzNormal (b) nzNormal

(c) ueNormal (d) mixNormal

(e) T_3 (f) EXP

Figure 2: MSE for different subsampling probabilities with second step subsample size n and a fixed first step subsample size $n_0 = 200$. Logarithm is taken on MSEs for better presentation.



(a) mzNormal (b) nzNormal

(c) ueNormal (d) mixNormal

(e) T_3 (f) EXP

Figure 3: Empirical and estimated MSEs for different second step subsample size n based on π^{Lopt} with the first step subsample size being fixed at $n_0 = 200$.

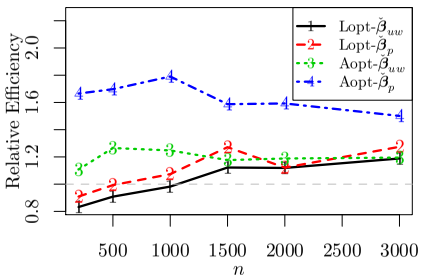


Figure 4: Relative efficiency for the SUSY data set with $n_0 = 200$ and different second step subsample size n. The gray horizontal dashed line is the reference line when relative efficiency is one.

8.2. Computational efficiency

We consider the computational efficiency of the more efficient estimation methods in this section. Note that they have the same order of computational time complexity, so they should have similar computational efficiency as the weighted estimator. For Poisson subsampling, there is no need to calculate $\{\pi_i^p\}_{i=1}^N$ all at once and random numbers can be generated on the go, so it requires less RAM and may require less CPU times as well. To confirm this, we record the computing time of implementing each of them for the case when $\mathbf{x}$ is mzNormal. All methods are implemented in the R programming language (R Core Team, 2017), and computations are carried out on a desktop running Ubuntu Linux 16.04 with an Intel I7 processor and 16GB RAM. Only one logical CPU is used for the calculation. We set the value of d to $d = 50$, the values of N to be $N = 10^4, 10^5, 10^6$ and 10^7, and the subsample sizes to be $n_0 = 200$ and $n = 1000$.

Table 1 gives the required CPU times (in seconds) to obtain $\check{\boldsymbol{\beta}}_w$, $\check{\boldsymbol{\beta}}_{uw}$, and $\check{\boldsymbol{\beta}}_p$, using π_i^{Lopt} and π_i^{Aopt}. The computing times for using the full data (Full) are also given for comparisons. It is seen that $\check{\boldsymbol{\beta}}_{uw}$ and $\check{\boldsymbol{\beta}}_p$ are a little faster than $\check{\boldsymbol{\beta}}_w$ but the advantages are not very significant. The reason is that the original OSMAC estimator $\check{\boldsymbol{\beta}}_{uw}$ pools the pilot subsample with the second stage subsample and performs iterative calculations on the pilot subsample twice, while the proposed method combines the pilot estimator with the second stage estimator which only requires iterative calculations on the pilot subsample once. Since the pilot subsample size is small, the difference is not significant. Note that these times are obtained when all the calculations are done in the RAM, and only the CPU times for implementing each method are counted while the time to generate the data is not counted.

Table 1: CPU seconds when the full data are generated and kept in the RAM. Here $n_0 = 200$, $n = 1000$, and the full data size N varies; the covariates are from a $d = 50$ dimensional multivariate normal distribution.

Method	N			
	10^4	10^5	10^6	10^7
π_i^{Lopt}, $\check{\boldsymbol{\beta}}_w$	0.14	0.13	0.45	5.24
π_i^{Lopt}, $\check{\boldsymbol{\beta}}_{uw}$	0.08	0.11	0.41	3.71
π_i^{Lopt}, $\check{\boldsymbol{\beta}}_p$	0.08	0.11	0.43	3.88
π_i^{Aopt}, $\check{\boldsymbol{\beta}}_w$	0.13	0.32	3.31	35.15
π_i^{Aopt}, $\check{\boldsymbol{\beta}}_{uw}$	0.12	0.31	3.29	34.98
π_i^{Aopt}, $\check{\boldsymbol{\beta}}_p$	0.12	0.31	3.29	35.06
Full	0.15	1.62	15.05	247.89

For big data problem, it is common that the full data are larger than the size of the available RAM, and full data can not be loaded into the RAM. For this scenario, one has to load the data into RAM line-by-line or block-by-block. Note that communication between CPU and hard drive is much slower than communication between CPU and RAM. Thus, this will dramatically increase the computing time. To mimic this situation, we store the full data on hard drive and use `readlines()` function to process data 1000 rows each time. We also use a smaller computer with 8GB RAM to implement the method. For the case when $N = 10^7$, the full data is about 9.1GB which is larger than the available RAM.

The computing times when data are scanned from hard drive are reported Table 2. Here the computing times can be over thousand times longer than those when data are loaded into RAM. Note that these computing times can be reduced dramatically if we use some other programming language like C++ (Stroustrup, 1986) or Julia (Bezanson et al., 2017). However, for fair comparisons, we use the same programming language R here. Furthermore, our main purpose here is to demonstrate the computational advantage of subsampling so the real focus is on the relative performance among different methods. From Table 2, it is seen that using π^{Aopt} does not cost much more time than using π^{Lopt}. The reason for this observation is that the major computing time is spent in data processing and the computing times used in calculating the subsampling probabilities are short. We also notice that Poisson subsampling is more computational efficient than subsampling with replacement since it calculates subsampling probabilities and generates random numbers on the go and requires one time less to scan the full data. Poisson subsampling only used about 2% of the time required by implementing the full data approach.

Table 2: CPU seconds when the full data are scanned from hard drive. Here $n_0 = 200$, $n = 1000$, and the full data size N varies; the covariates are from a $d = 50$ dimensional multivariate normal distribution.

Method	N			
	10^4	10^5	10^6	10^7
π_i^{Lopt}, $\check{\boldsymbol{\beta}}_w$	4.26	41.60	441.46	4374.94
π_i^{Lopt}, $\check{\boldsymbol{\beta}}_{uw}$	4.13	41.42	413.09	4384.99
π_i^{Lopt}, $\check{\boldsymbol{\beta}}_p$	2.77	27.58	272.32	2699.13
π_i^{Aopt}, $\check{\boldsymbol{\beta}}_w$	4.43	41.75	434.96	4393.38
π_i^{Aopt}, $\check{\boldsymbol{\beta}}_{uw}$	4.10	41.83	417.55	4369.04
π_i^{Aopt}, $\check{\boldsymbol{\beta}}_p$	2.88	27.93	273.24	2719.51
Full	139.46	1411.78	14829.63	138134.69

9. Summary

In this paper, we have proposed a new un-weighted estimator for logistic regression based on an OSMAC subsample. We have derived conditional asymptotic distribution of the new estimator which has a smaller variance-covariance matrix compared with the weighted estimator.

We have also investigated the asymptotic properties if Poisson subsampling is used, and showed that the resultant estimator has the same conditional asymptotic distribution if the subsampling ratio converges to zero. However, if the subsampling ratio converges to a positive constant, the estimator based on Poisson subsampling has a smaller variance-covariance matrix.

In addition, we have derive the unconditional asymptotic distribution for the proposed estimator based on Poisson subsampling. Interestingly, if the subsampling ratio converges to zero, the unconditional asymptotic distribution is the same as the conditional asymptotic distribution, indicating that the variation of the full data can be ignored. If the subsampling ratio does not converge to zero, the unconditional asymptotic distribution has a larger variance-covariance matrix. Our results also include the local case-control sampling method. With a stronger moment condition that the third moment of the covariate is finite, we do not require the pilot estimate to be independent of the data.

Furthermore, we have proved consistency and asymptotic normality for the proposed estimators under two types of misspecifications: one is that pilot estimators are inconsistent, and the other is that the logistic regression model is misspecified.

Acknowledgments

The author would like to thank the reviewers for their insightful comments and suggestions, which greatly helped improve the paper and strengthen the proposed methods. This work was partially supported by the NSF grant DMS-1812013 and a UConn REP grant.

Appendix A. Subsampling with replacement from hard drive

If the full data can be loaded into available RAM, subsampling probabilities can be calculated in RAM and subsampling with replacement can be implemented directly. Otherwise, special considerations have to be given in practical implementation. If the full data is larger than available RAM while subsampling probabilities $\{\pi_i^{\text{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ can still be loaded in available RAM, one can calculate $\{\pi_i^{\text{OS}}(\hat{\boldsymbol{\beta}}_0)\}_{i=1}^N$ by scanning the data from hard drive line-by-line or block-by-block, generate row indexes for a subsample, and then scan the data line-by-line or block-by-block to take the subsample. To be specific, one can draw a subsample, say $\{idx_1, ..., idx_n\}$, from $\{1, ..., N\}$, sort the indexes to have $\{idx_{(1)}, ..., idx_{(n)}\}$, and then use the Algorithm 3 to scan the data line-by-line or block-by-block in order to obtain the subsample.

Algorithm 3 Obtain the subsample with the given indexes by scanning through the full data

```
Input: data file, subsample indexes {idx_(1), ..., idx_(n)}.
i ← 1
j ← 1
while i ≤ N and j ≤ n do
  readline(data file)
  if i == idx_(j) then
    include the i-th data point into the subsample
    while i == idx_(j) do
      j ← j + 1
    end while
  end if
  i ← i + 1
end while
```

Clearly, Algorithm 3 takes no more than linear time to run. Here, we assume that a generic function `readline()` reads a single line (or multiple lines) from the data file and stop at the beginning of the next line (or next block) in the data file. No calculation is performed on a data line if it is not included in the subsample. Such functionality is provided by most programming languages. For example, Julia (Bezanson et al., 2017) and Python (van Rossum, 1995) has a function `readline()` that read a file line-by-line; R (R Core Team, 2017) has a function `readLines()` that read one or multiple lines; C (Kernighan and Ritchie, 1988) and C++ (Stroustrup, 1986) has a function `getline()` to read one line at a time.

Appendix B. Proofs and technical details

In this appendix, we provide proofs for the results in the paper. Technical details related to sampling with replacement in Section 3 are presented in Section B.1; technical details related to Poisson subsampling in Section 4 are presented in Section B.2; technical details related to unconditional results in Section 6 are presented in Section B.3; and technical details related to model misspecification in Section 7 are presented in Section B.4.1.

B.1. Proofs for subsampling with replacement

In this section we prove the results in Section 3. For ease of presentation, we use notation λ to denote the log-likelihood shifted by $\hat{\boldsymbol{\beta}}_0$. For the subsample, $\lambda^*_{uw}(\boldsymbol{\beta}) = \ell^*_{uw}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)$. Denote the first and second derivatives of $\lambda^*_{uw}(\boldsymbol{\beta})$ as $\dot{\lambda}^*_{uw}(\boldsymbol{\beta}) = \partial\lambda^*_{uw}(\boldsymbol{\beta})/\partial\boldsymbol{\beta}$ and $\ddot{\lambda}^*_{uw}(\boldsymbol{\beta}) = \partial^2\lambda^*_{uw}(\boldsymbol{\beta})/(\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\mathrm{T}})$.

Note that from Xiong and Li (2008); Cheng and Huang (2010), the fact that a sequence converges to 0 in conditional probability is equivalent to the fact that it converges to 0 in unconditional probability. This can also be proved directly by using the fact the probability measure is bounded by 1. Thus, in the following, we will use $o_P(1)$ to denote a sequence converging to 0 in probability without stating whether the underlying probability measure is conditional or unconditional.

We first present some lemmas that will be used to prove Theorem 1, and provide their proofs in Sections B.1.2 - B.1.5.

Lemma 28 *Let $\mathbf{v}_1, ..., \mathbf{v}_N$ be i.i.d. random vectors with the same distribution of $\mathbf{v}$. Let g_{1N} be a bounded function and g_2 be a fixed function that does not depend on N. If $g_{1N}(\mathbf{v}) = o_P(1)$ and $\mathbb{E}|g_2(\mathbf{v})| < \infty$, then*

$$\frac{1}{N}\sum_{i=1}^{N} g_{1N}(\mathbf{v}_i)g_2(\mathbf{v}_i) = o_P(1).$$

Lemma 29 *Let $\boldsymbol{\eta}_i = |\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i$, where $\psi_i(\boldsymbol{\beta}) = y_i - p(\mathbf{x}_i, \boldsymbol{\beta})$. Under Assumptions 1 and 2, conditional on the consistent $\hat{\boldsymbol{\beta}}_0$, if $n_0/\sqrt{N} \to 0$, then*

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_t) = \frac{\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}}{2\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\}}\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\boldsymbol{\eta}_i + o_P(1),$$

which converges in distribution to a normal distribution with mean $\mathbf{0}$ and variance-covariance matrix $[\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}]^{-1}\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h^2(\mathbf{x})\mathbf{x}\mathbf{x}\}[\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}]^{-1}$, as n_0 and N go to infinity.

Lemma 30 *Let*

$$\dot{\lambda}^*_{uw}(\boldsymbol{\beta}_t) = \sum_{i=1}^{n}\{y_i^* - p_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i^*.$$

Under Assumptions 1 and 2, conditional on $\mathcal{D}_N$ and the consistent $\hat{\boldsymbol{\beta}}_0$, as n_0, n, and N go to infinity,

$$\frac{\dot{\lambda}^*_{uw}(\boldsymbol{\beta}_t)}{\sqrt{n}} - \frac{\sqrt{n}\sum_{i=1}^{N}\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)} \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}\big),$$

in distribution.

Lemma 31 *Under Assumptions 1-3, as n_0, n, and N go to infinity, for any $\mathbf{s}_n \to 0$ in probability,*

$$\frac{1}{n}\sum_{i=1}^{n}\phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n)\|\mathbf{x}_i^*\|^2 - \sum_{i=1}^{N}\pi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\|\mathbf{x}_i\|^2 = o_P(1).$$

Proof of Theorem 1. The estimator $\hat{\boldsymbol{\beta}}_{uw}$ is the maximizer of

$$\lambda^*_{uw}(\boldsymbol{\beta}) = \sum_{i=1}^{n}\Big[(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)^{\mathrm{T}}\mathbf{x}_i^* y_i^* - \log\big\{1 + e^{(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)^{\mathrm{T}}\mathbf{x}_i^*}\big\}\Big],$$

so $\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_t)$ is the maximizer of $\gamma(\mathbf{s}) = \lambda^*_{uw}(\boldsymbol{\beta}_t + \mathbf{s}/\sqrt{n}) - \lambda^*_{uw}(\boldsymbol{\beta}_t)$. By Taylor's expansion,

$$\gamma(\mathbf{s}) = \frac{1}{\sqrt{n}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}^*_{uw}(\boldsymbol{\beta}_t) + \frac{1}{2n}\sum_{i=1}^{n}\phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i^*)^2,$$

where $\phi_i^*(\boldsymbol{\beta}) = p_i^*(\boldsymbol{\beta})\{1 - p_i^*(\boldsymbol{\beta})\}$, and $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$.

From Lemma 31,

$$\frac{1}{n}\sum_{i=1}^{n}\phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}} - \sum_{i=1}^{N}\pi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = o_P(1).$$

From Lemma 28 and the law of large numbers,

$$\sum_{i=1}^{N}\pi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \frac{\frac{1}{N}\sum_{i=1}^{N}|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\Psi_N(\hat{\boldsymbol{\beta}}_0)} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1} + o_P(1).$$

Combining the above two equations, we have that $n^{-1}\sum_{i=1}^{n}\phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}}$ converges in probability to $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}$, a positive definite matrix. In addition, from Lemma 29 and Lemma 30, $\dot{\lambda}^*_{uw}(\boldsymbol{\beta}_t)/\sqrt{n}$ is stochastically bounded. Thus, from the Basic Corollary in page 2 of Hjort and Pollard (2011), the maximizer of $\gamma(\mathbf{s})$, $\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_t)$, satisfies

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_t) = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\frac{1}{\sqrt{n}}\dot{\lambda}^*_{uw}(\boldsymbol{\beta}_t) + o_P(1)$$

given $\mathcal{D}_N$ and $\hat{\boldsymbol{\beta}}_0$. Thus,

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}) = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\left\{\frac{1}{\sqrt{n}}\dot{\lambda}^*_{uw}(\boldsymbol{\beta}_t) - \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}\sqrt{n}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_t)\right\} + o_P(1). \tag{26}$$

From Lemma 29,

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}\sqrt{n}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_t) = \frac{\sqrt{n}\sum_{i=1}^{N}\boldsymbol{\eta}_i}{2N\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\}} = \frac{\sqrt{n}\sum_{i=1}^{N}\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)} + o_P(1), \tag{27}$$

Combining equations (26) and (27), Lemma 30, Slutsky's theorem, and the fact that a conditional probability is bounded by one, Theorem 1 follows. ■

B.1.1. Proof of Proposition 4

Proof of Proposition 4. To prove that $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} \leq \mathbf{V}^{\mathrm{OS}} = \mathbf{M}^{-1}\mathbf{V}_c^{\mathrm{OS}}\mathbf{M}^{-1}$, we just need to show that

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}_{\mathrm{MLE}}}^{-1} \geq \mathbf{M}(\mathbf{V}_c^{\mathrm{OS}})^{-1}\mathbf{M}.$$

From the strong law of large numbers,

$$\mathbf{M} = \frac{1}{N}\sum_{i=1}^{N}\phi_i(\boldsymbol{\beta}_t)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} + o(1),$$

$$\mathbf{V}_c^{\mathrm{OS}} = 4\Phi(\boldsymbol{\beta}_t)\frac{1}{N}\sum_{i=1}^{N}\frac{\phi_i(\boldsymbol{\beta}_t)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{h(\mathbf{x}_i)} + o(1),$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} = 4\Phi(\boldsymbol{\beta}_t)\left\{\frac{1}{N}\sum_{i=1}^{N}\phi_i(\boldsymbol{\beta}_t)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\right\}^{-1} + o(1),$$

almost surely. Thus, we only need to verify that

$$\sum_{i=1}^{N}\phi_i(\boldsymbol{\beta}_t)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} \geq \left\{\sum_{i=1}^{N}\phi_i(\boldsymbol{\beta}_t)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\right\}\left\{\sum_{i=1}^{N}\frac{\phi_i(\boldsymbol{\beta}_t)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{h(\mathbf{x}_i)}\right\}^{-1}\left\{\sum_{i=1}^{N}\phi_i(\boldsymbol{\beta}_t)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\right\}.$$

Denote $\mathbf{Z} = \left\{\sqrt{\phi_1(\boldsymbol{\beta}_t)}\mathbf{x}_1, ..., \sqrt{\phi_N(\boldsymbol{\beta}_t)}\mathbf{x}_N\right\}^{\mathrm{T}}$, and $\mathbf{H} = \mathrm{diag}\{h(\mathbf{x}_1), ..., h(\mathbf{x}_N)\}$. The above inequality can be written as

$$\mathbf{Z}^{\mathrm{T}}\mathbf{H}\mathbf{Z} \geq \mathbf{Z}^{\mathrm{T}}\mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{Z},$$

which is true if

$$\mathbf{H} \geq \mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}} \tag{28}$$

Note that $(\mathbf{H}^{-1/2}\mathbf{Z})\{(\mathbf{H}^{-1/2}\mathbf{Z})(\mathbf{H}^{-1/2}\mathbf{Z})^{\mathrm{T}}\}^{-1}(\mathbf{H}^{-1/2}\mathbf{Z})^{\mathrm{T}}$ is the projection matrix of $\mathbf{H}^{-1/2}\mathbf{Z}$, so it is, under the Loewner ordering, smaller than or equal to the identity matrix $\mathbf{I}_N$, namely,

$$\mathbf{I}_N \geq \mathbf{H}^{-1/2}\mathbf{Z}(\mathbf{Z}\mathbf{H}^{-1}\mathbf{Z}^{\mathrm{T}})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{H}^{-1/2},$$

which implies (28). If $h(\mathbf{x}) = 1$, the equality can be verified directly.

The first inequality in (14) can be verified directly using the result in (13). For the second inequality in (14), noting that $h(\mathbf{x}) = \|L\mathbf{M}^{-1}\mathbf{x}\|$, by the CauchySchwarz inequality, we have

$$\begin{aligned}
\mathrm{tr}(L\mathbf{V}^{\mathrm{OS}}L^{\mathrm{T}}) &= \mathrm{tr}(L\mathbf{M}^{-1}\mathbf{V}_c^{\mathrm{OS}}\mathbf{M}^{-1}L^{\mathrm{T}}) \\
&= 4\Phi(\boldsymbol{\beta}_t)\mathrm{tr}\big[L\mathbf{M}^{-1}\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)h^{-1}(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big\}\mathbf{M}^{-1}L^{\mathrm{T}}\big] \\
&= 4\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\}\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)h^{-1}(\mathbf{x})\|L\mathbf{M}^{-1}\mathbf{x}\|^2\big\} \\
&= 4\big[\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)\|L\mathbf{M}^{-1}\mathbf{x}\|\big\}\big]^2 \\
&\leq 4\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)\|L\mathbf{M}^{-1}\mathbf{x}\|^2\big\} \\
&= 4\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}\mathrm{tr}\big[L\mathbf{M}^{-1}\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\mathbf{x}\mathbf{x}^{\mathrm{T}}\}\mathbf{M}^{-1}L^{\mathrm{T}}\big] \\
&= 4\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\}\mathrm{tr}(L\mathbf{M}^{-1}L^{\mathrm{T}}) \\
&< \mathrm{tr}(L\mathbf{M}^{-1}L^{\mathrm{T}}),
\end{aligned}$$

which finishes the proof. ■

B.1.2. Proof of Lemma 28

Proof of Lemma 28. Let B be a bound for g_{1N} i.e., $|g_{1N}| \leq B$. For any $\epsilon > 0$, by Markov's inequality,

$$\begin{aligned}
&\mathbb{P}\left\{\left|\frac{1}{N}\sum_{i=1}^{N}g_{1N}(\mathbf{v}_i)g_2(\mathbf{v}_i)\right| > \epsilon\right\} \leq \frac{\mathbb{E}|g_{1N}(\mathbf{v})g_2(\mathbf{v})|}{\epsilon} \\
&= \frac{\mathbb{E}[|g_{1N}(\mathbf{v})||g_2(\mathbf{v})|I\{|g_2(\mathbf{v})| \leq K\}]}{\epsilon} + \frac{\mathbb{E}[|g_{1N}(\mathbf{v})||g_2(\mathbf{v})|I\{|g_2(\mathbf{v})| > K\}]}{\epsilon} \\
&\leq \frac{K}{\epsilon}\mathbb{E}|g_{1N}(\mathbf{v})| + \frac{B}{\epsilon}\mathbb{E}\{|g_2(\mathbf{v})|I(|g_2(\mathbf{v})| > K)\}.
\end{aligned}$$

For any $\zeta > 0$, we can choose a K large enough such that $\mathbb{E}\{|g_2(\mathbf{v})|I(|g_2(\mathbf{v})| \leq K)\} < \zeta\epsilon/(2B)$, since $\mathbb{E}|g_2(\mathbf{v})| < \infty$. The facts that $g_{1N}(\mathbf{v}_i) \leq B$ and $g_{1N}(\mathbf{v}_i) = o_P(1)$ imply that $\mathbb{E}|g_{1N}(\mathbf{v})| = o(1)$. Thus, there is a N_ζ such that $\mathbb{E}|g_{1N}(\mathbf{v})| < \zeta\epsilon/(2K)$ when $N > N_\zeta$. Therefore, for any $\zeta > 0$, $\mathbb{P}\{|N^{-1}\sum_{i=1}^{N}g_{1N}(\mathbf{v}_i)g_2(\mathbf{v}_i)| > \epsilon\} < \zeta$ for sufficiently large N. This finishes the proof. ■

B.1.3. Proof of Lemma 29

Proof of Lemma 29. Since $\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}$ is the maximizer of

$$\lambda_{\mathrm{wMLE}}(\boldsymbol{\beta}) = \sum_{i=1}^{N}|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\big[y_i\mathbf{x}_i^{\mathrm{T}}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0) - \log\{1 + e^{\mathbf{x}_i^{\mathrm{T}}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)}\}\big],$$

$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_t)$ is the maximizer of $\gamma_{\mathrm{wMLE}}(\mathbf{s}) = \lambda_{\mathrm{wMLE}}(\boldsymbol{\beta}_t + \mathbf{s}/\sqrt{N}) - \lambda_{\mathrm{wMLE}}(\boldsymbol{\beta}_t)$. By Taylor's expansion,

$$\gamma_{\mathrm{wMLE}}(\mathbf{s}) = \frac{1}{\sqrt{N}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}_{\mathrm{wMLE}}(\boldsymbol{\beta}_t) - \frac{1}{2N}\sum_{i=1}^{N}|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{N})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i)^2$$

where

$$\dot{\lambda}_{\mathrm{wMLE}}(\boldsymbol{\beta}_t) = \sum_{i=1}^{N}\boldsymbol{\eta}_i = \sum_{i=1}^{N}|\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i, \tag{29}$$

and $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$.

Since that $n_0/\sqrt{N} \to 0$ and $\|\boldsymbol{\eta}_i\|$ is bounded by $\|h(\mathbf{x}_i)\mathbf{x}_i\|$, we can ignore the data points that are used in obtaining the pilot $\hat{\boldsymbol{\beta}}_0$. The following discussion focus on $\boldsymbol{\eta}_i$'s for which $(\mathbf{x}_i, y_i)$'s are not included in the pilot subsample. Note that for such $\boldsymbol{\eta}_i$'s, $\mathbb{E}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0) = \mathbf{0}$

because

$$\begin{aligned}\mathbb{E}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0,\mathbf{x}_i) =&\mathbb{E}\Big[|\psi_i(\hat{\boldsymbol{\beta}}_0)|\{y_i-p(\mathbf{x}_i,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\}h(\mathbf{x}_i)\mathbf{x}_i\Big|\hat{\boldsymbol{\beta}}_0,\mathbf{x}_i\Big]\\ =&-p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)p(\mathbf{x}_i,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\{1-p(\mathbf{x}_i,\boldsymbol{\beta}_t)\}h(\mathbf{x}_i)\mathbf{x}_i\\ &+\{1-p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)\}\{1-p(\mathbf{x}_i,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\}p(\mathbf{x}_i,\boldsymbol{\beta}_t)h(\mathbf{x}_i)\mathbf{x}_i\\ =&-\frac{e^{\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_0}}{1+e^{\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_0}}\frac{e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_t-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_0}}{1+e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_t-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_0}}\frac{1}{1+e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_t}}h(\mathbf{x}_i)\mathbf{x}_i\\ &+\frac{1}{1+e^{\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_0}}\frac{1}{1+e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_t-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_0}}\frac{e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_t}}{1+e^{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_t}}h(\mathbf{x}_i)\mathbf{x}_i=0.\end{aligned}\tag{30}$$

This also gives that

$$\mathbb{V}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0)=\mathbb{E}\big\{\mathbb{V}(\boldsymbol{\eta}_i|\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)\big|\hat{\boldsymbol{\beta}}_0\big\}+\mathbb{V}\big\{\mathbb{E}(\boldsymbol{\eta}_i|\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)\big|\hat{\boldsymbol{\beta}}_0\big\}=\mathbb{E}\big\{\mathbb{V}(\boldsymbol{\eta}_i|\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)\big|\hat{\boldsymbol{\beta}}_0\big\}.$$

Now, since

$$\begin{aligned}\mathbb{V}(\boldsymbol{\eta}_i|\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)=&\mathbb{E}\Big[|y_i-p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)|^2\big\{y_i-p(\mathbf{x}_i,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\big\}^2h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\Big|\hat{\boldsymbol{\beta}}_0,\mathbf{x}_i\Big]\\ =&p(\mathbf{x}_i,\boldsymbol{\beta}_t)\{1-p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)\}^2\big\{1-p(\mathbf{x}_i,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\big\}^2h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\ &+\{1-p(\mathbf{x}_i,\boldsymbol{\beta}_t)\}\{p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)\}^2\big\{p(\mathbf{x}_i,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\big\}^2h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\ =&\phi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\hat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_t)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}},\end{aligned}$$

we have

$$\mathbb{V}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0)=\mathbb{E}\Big\{\phi(\hat{\boldsymbol{\beta}}_0)\phi(\hat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_t)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\Big|\hat{\boldsymbol{\beta}}_0\Big\}.$$

Let $\|\|$ denote the Frobenius norm if applied on a martix, i.e., for a matrix A, $\|A\|^2=\mathrm{tr}(AA^{\mathrm{T}})$, and denote $\mathbb{V}(\boldsymbol{\eta}_i|\boldsymbol{\beta}_t)=\mathbb{E}\{\phi(\boldsymbol{\beta}_t)\phi(\boldsymbol{\beta}_t-\boldsymbol{\beta}_t)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\}=0.25\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\}$. Notice that $\big|\phi(\hat{\boldsymbol{\beta}}_0)\phi(\hat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_t)-0.25\phi_i(\boldsymbol{\beta}_t)\big|h^2(\mathbf{x}_i)\|\mathbf{x}_i\|^2$ converges to 0 in probability and it is bounded by $h^2(\mathbf{x}_i)\|\mathbf{x}_i\|^2$, an integrable random variable under Assumption 2. Thus,

$$\mathbb{E}\big\|\mathbb{V}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0)-\mathbb{V}(\boldsymbol{\eta}_i|\boldsymbol{\beta}_t)\big\|\le\mathbb{E}\big\{\big|\{\phi(\hat{\boldsymbol{\beta}}_0)\phi(\hat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}_t)-0.25\phi_i(\boldsymbol{\beta}_t)\big|h^2(\mathbf{x}_i)\|\mathbf{x}_i\|^2\big\}=o(1).$$

This implies that

$$\mathbb{V}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0)=\mathbb{V}(\boldsymbol{\eta}_i|\boldsymbol{\beta}_t)+o_P(1)=0.25\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i\big\}+o_P(1).$$

For $\boldsymbol{\eta}_i$'s that $(\mathbf{x}_i,y_i)$'s are not included in the pilot subsample, conditional on $\hat{\boldsymbol{\beta}}_0$, $\boldsymbol{\eta}_i$'s are i.i.d. with mean $\mathbf{0}$ and variance $\mathbb{V}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0)$. Since for any $\epsilon>0$,

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^N\mathbb{E}\Big\{\|\boldsymbol{\eta}_i\|^2I(\|\boldsymbol{\eta}_i\|>\sqrt{N}\epsilon)\Big|\hat{\boldsymbol{\beta}}_0\Big\}\le&\frac{1}{N}\sum_{i=1}^N\mathbb{E}\Big\{\|h(\mathbf{x}_i)\mathbf{x}_i\|^2I(\|h(\mathbf{x}_i)\mathbf{x}_i\|>\sqrt{N}\epsilon)\Big|\hat{\boldsymbol{\beta}}_0\Big\}\\ =&\mathbb{E}\{\|h(\mathbf{x})\mathbf{x}\|^2I(\|h(\mathbf{x})\mathbf{x}\|>\sqrt{N}\epsilon)\}\to0,\end{aligned}$$

the Lindeberg-Feller central limit theorem (Section *2.8 of van der Vaart, 1998) applies conditional on $\hat{\boldsymbol{\beta}}_0$. Thus, we have, conditional on $\hat{\boldsymbol{\beta}}_0$,

$$\frac{\dot{\lambda}_{\mathrm{wMLE}}(\boldsymbol{\beta}_t)}{\sqrt{N}}\longrightarrow\mathbb{N}\Big[\mathbf{0},\ \frac{\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h^2(\mathbf{x})\mathbf{x}\mathbf{x}\}}{4}\Big],$$

in distribution. From Lemma 28, conditional on $\hat{\boldsymbol{\beta}}_0$,

$$\begin{aligned}&\frac{1}{N}\sum_{i=1}^N|y_i-p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0+\acute{\mathbf{s}}/\sqrt{N})\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\ &=\frac{1}{4}\mathbb{E}\{|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}+o_{P|\hat{\boldsymbol{\beta}}_0}(1)=\frac{1}{2}\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}+o_P(1).\end{aligned}$$

Thus, from the Basic Corollary in page 2 of Hjort and Pollard (2011), the maximizer of $\gamma_{\mathrm{wMLE}}(\mathbf{s})$, $\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}-\boldsymbol{\beta}_t)$, satisfies

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}-\boldsymbol{\beta}_t)=2[\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}]^{-1}\frac{1}{\sqrt{N}}\dot{\lambda}_{\mathrm{wMLE}}(\boldsymbol{\beta}_t)+o_P(1).\tag{31}$$

Note that

$$[\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}]^{-1}=\frac{\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}}{4\Phi(\boldsymbol{\beta}_t)}.\tag{32}$$

Combining equations (29), (31), and (32), we have

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}-\boldsymbol{\beta}_t)=\frac{\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}}{2\Phi(\boldsymbol{\beta}_t)}\frac{1}{\sqrt{N}}\dot{\lambda}_{\mathrm{wMLE}}(\boldsymbol{\beta}_t)+o_P(1).$$

An application of Slutsky's theorem yields the result for the asymptotic normality. ■

B.1.4. Proof of Lemma 30

Proof of Lemma 30. Note that given $\mathcal{D}_N$ and $\hat{\boldsymbol{\beta}}_0$, $\{y_i^*-p_i^*(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i^*$ are i.i.d. random vectors. We now exam their mean and variance, and check the Lindeberg-Feller condition (Section *2.8 of van der Vaart, 1998) under the conditional distribution given $\mathcal{D}_N$ and $\hat{\boldsymbol{\beta}}_0$. For the expectation, we have,

$$\mathbb{E}\big[\{y^*-p(\mathbf{x}_i^*,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^*\big|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0\big]=\sum_{i=1}^N\pi_i(\hat{\boldsymbol{\beta}}_0)\psi_i(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\mathbf{x}_i=\frac{\sum_{i=1}^N\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)},$$

where $\Psi_N(\boldsymbol{\beta})=N^{-1}\sum_{i=1}^N|y_i-p(\mathbf{x}_i,\boldsymbol{\beta})|h(\mathbf{x}_i)$. From Lemma 29 and its proof, $\sum_{i=1}^N\boldsymbol{\eta}_i=O_P(\sqrt{N})$ conditional on $\hat{\boldsymbol{\beta}}_0$ in probability, i.e., for any $\epsilon>0$, there exits a K such that $\mathbb{P}\{\mathbb{P}(\sum_{i=1}^N\boldsymbol{\eta}_i/\sqrt{N}>K\big|\hat{\boldsymbol{\beta}}_0)<\epsilon\}\to1$ as $n_0,N\to\infty$. From Xiong and Li (2008), we know that $\sum_{i=1}^N\boldsymbol{\eta}_i=O_P(\sqrt{N})$ unconditionally. Thus, for the expectation, we have

$$\Delta=\mathbb{E}\big[\{y^*-p(\mathbf{x}_i^*,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^*\big|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0\big]=O_P(1/\sqrt{N}).$$

For the variance,

$$
\begin{aligned}
&\mathbb{V}\big[\{y^* - p(\mathbf{x}_i^*, \boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^* | \mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big] \\
=&\sum_{i=1}^N \pi_i(\hat{\boldsymbol{\beta}}_0)\{y_i - p(\mathbf{x}_i, \boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}^2 \mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} - \Delta^2 \\
=&\frac{\frac{1}{N}\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\Psi_N(\hat{\boldsymbol{\beta}}_0)} - O_P(1/N) \\
=&\frac{\frac{1}{N}\sum_{i=1}^N |\psi_i(\boldsymbol{\beta}_t)|(y_i - 0.5)^2 h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\Psi_N(\boldsymbol{\beta}_t)} + o_P(1) \\
=&\frac{1}{4}\frac{\mathbb{E}\{|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}}{\Psi(\boldsymbol{\beta}_t)} + o_P(1) = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1} + o_P(1)
\end{aligned}
$$

where the third equality is from Lemma 28 and the fact that $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$, and the forth equality is from the law of large numbers.

Now we check the Lindeberg-Feller condition (Section *2.8 of van der Vaart, 1998) under the conditional distribution. Denote $\dot{\lambda}_{ri}^* = \{y_i^* - p_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i^*$.

$$
\begin{aligned}
&\mathbb{E}\frac{1}{n}\sum_{i=1}^n \{\|\dot{\lambda}_{ri}^*\|^2 I(\|\dot{\lambda}_{ri}^*\| > \sqrt{n}\epsilon)\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\} \\
\le&\mathbb{E}\big\{\|\mathbf{x}^*\|^2 I(\|\mathbf{x}\| > \sqrt{n}\epsilon)\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big\} \\
=&\sum_{i=1}^N \pi_i(\hat{\boldsymbol{\beta}}_0)\big\{\|\mathbf{x}_i\|^2 I(\|\mathbf{x}_i\| > \sqrt{n}\epsilon)\big\} \\
\le&\frac{\frac{1}{N}\sum_{i=1}^N \big\{h(\mathbf{x}_i)\|\mathbf{x}_i\|^2 I(\|\mathbf{x}_i\| > \sqrt{n}\epsilon)\big\}}{\Psi_N(\hat{\boldsymbol{\beta}}_0)} \\
\le&\frac{\frac{1}{N}\sum_{i=1}^N \big\{h(\mathbf{x}_i)\|\mathbf{x}_i\|^2 I(\|\mathbf{x}_i\| > \sqrt{n}\epsilon)\big\}}{\Psi_N(\hat{\boldsymbol{\beta}}_0)} = o_P(1),
\end{aligned}
$$

by Lemma 28 and the fact that $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$. Thus, applying the Lindeberg-Feller central limit theorem (Section *2.8 of van der Vaart, 1998) finishes the proof. ■

B.1.5. Proof of Lemma 31

Proof of Lemma 31. We begin with the following partition,

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n \phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n)\|\mathbf{x}_i^*\|^2 \\
=&\frac{1}{n}\sum_{i=1}^n \phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n)\|\mathbf{x}_i^*\|^2 I(\|\mathbf{x}_i^*\|^2 \le n) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n)\|\mathbf{x}_i^*\|^2 I(\|\mathbf{x}_i^*\|^2 > n) \\
\equiv&\Delta_1 + \Delta_2.
\end{aligned}
$$

The second term Δ_2 is $o_P(1)$ because it is non-negative and

$$
\begin{aligned}
\mathbb{E}(\Delta_2|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) =& \sum_{i=1}^N \pi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n)\|\mathbf{x}_i\|^2 I(\|\mathbf{x}_i\|^2 > n) \\
\le& \frac{\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\|\mathbf{x}_i\|^2 I(\|\mathbf{x}_i\|^2 > n)}{\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)} \\
\le& \frac{\frac{1}{N}\sum_{i=1}^N h(\mathbf{x}_i)\|\mathbf{x}_i\|^2 I(\|\mathbf{x}_i\|^2 > n)}{\Psi_N(\hat{\boldsymbol{\beta}}_0)} = o_P(1)
\end{aligned}
$$

as $n, N \to \infty$, where the last step is from Lemma 28.

Similarly, we can show that

$$
\mathbb{E}(\Delta_1|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) - \sum_{i=1}^N \pi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\|\mathbf{x}_i\|^2 = o_P(1).
$$

Thus, we only need to show that $\Delta_1 - \mathbb{E}(\Delta_1|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) = o_P(1)$. For this, we show that the conditional variance of Δ_1 goes to 0 in probability. Notice that

$$
\begin{aligned}
&\mathbb{V}(\Delta_1|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) \\
=&\frac{1}{n}\mathbb{V}\{\phi^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\|\mathbf{x}^*\|^2 I(\|\mathbf{x}^*\|^2 \le n)\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\} \\
\le&\frac{1}{16n}\mathbb{E}\big\{\|\mathbf{x}^*\|^4 I(\|\mathbf{x}^*\|^2 \le n)\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big\} \\
=&\frac{1}{16n}\sum_{i=1}^n \mathbb{E}\big\{\|\mathbf{x}^*\|^4 I(i-1 < \|\mathbf{x}^*\|^2 \le i)\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big\} \\
\le&\frac{1}{16n}\sum_{i=1}^n i^2\mathbb{E}\big\{I(i-1 < \|\mathbf{x}^*\|^2 \le i)\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big\} \\
\le&\frac{1}{16n}\sum_{i=1}^n i^2\big\{\mathbb{P}(\|\mathbf{x}^*\|^2 > i-1\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) - \mathbb{P}(\|\mathbf{x}^*\|^2 > i\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0)\big\} \\
=&\frac{1}{16n}\Big\{\mathbb{P}(\|\mathbf{x}^*\|^2 > 0\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) - n^2\mathbb{P}(\|\mathbf{x}^*\|^2 > n\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) + \sum_{i=1}^{n-1}(2i+1)\mathbb{P}(\|\mathbf{x}^*\|^2 > i\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0)\Big\} \\
\le&\frac{1}{16n}\Big\{1 + \sum_{i=1}^{n-1} 3i\mathbb{P}(\|\mathbf{x}^*\|^2 > i\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0)\Big\}
\end{aligned}
$$

This is $o_P(1)$ because

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} i\mathbb{P}(\|\mathbf{x}^*\|^2 > i|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0) = \frac{1}{n}\sum_{i=1}^{n} i\sum_{j=1}^{n}\pi_j(\hat{\boldsymbol{\beta}}_0)I(\|\mathbf{x}_j\|^2 > i)\\
&= \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{n} i\pi_j(\hat{\boldsymbol{\beta}}_0)I(\|\mathbf{x}_j\|^2 > i)\\
&= \frac{\frac{1}{N}\sum_{j=1}^{n}\frac{1}{n}\sum_{i=1}^{n} i|\psi_j(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_j)I(\|\mathbf{x}_j\|^2 > i)}{\Psi_N(\hat{\boldsymbol{\beta}}_0)}\\
&\le \frac{\frac{1}{N}\sum_{j=1}^{n}\frac{1}{n}\sum_{i=1}^{n} ih(\mathbf{x}_j)I(\|\mathbf{x}_j\|^2 > i)}{\Psi_N(\hat{\boldsymbol{\beta}}_0)},
\end{aligned}$$

and the numerator is non-negative and has an expectation

$$\frac{1}{N}\sum_{j=1}^{n}\frac{1}{n}\sum_{i=1}^{n} i\mathbb{E}\{h(\mathbf{x})I(\|\mathbf{x}\|^2 > i)\}$$

which is $o(1)$ since $i\mathbb{E}\{h(\mathbf{x})I(\|\mathbf{x}\|^2 > i)\} = o(1)$ as $i \to \infty$. ■

B.2. Proofs for Poisson subsampling

In this section we prove the results in Section 4 about Poisson subsampling.

Define $\delta_i^{\hat{\boldsymbol{\beta}}_0} = I\{u_i \le n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\}$, and use notation λ_p to denote the log-likelihood shifted by $\hat{\boldsymbol{\beta}}_0$, i.e., $\lambda_p(\boldsymbol{\beta}) = \ell_p^*(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)$. Using these notations, the estimator $\hat{\boldsymbol{\beta}}_p$ is the maximizer of

$$\lambda_p(\boldsymbol{\beta}) = \sum_{i=1}^{N}\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\big[(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}_0)^{\mathrm{T}}\mathbf{x}_i y_i - \log\{1 + e^{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}_0)^{\mathrm{T}}\mathbf{x}_i}\}\big], \tag{33}$$

Denote the first and second derivatives of $\lambda_p(\boldsymbol{\beta})$ as $\dot{\lambda}_p(\boldsymbol{\beta}) = \partial\lambda_p(\boldsymbol{\beta})/\partial\boldsymbol{\beta}$ and $\ddot{\lambda}_p(\boldsymbol{\beta}) = \partial^2\lambda_p(\boldsymbol{\beta})/(\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\mathrm{T}})$. Two lemmas similar to Lemmas 30 and 31 are derived below which will be used to prove Theorem 6. We will prove these two lemmas in Sections B.2.2 and B.2.3.

Lemma 32 *Let*

$$\dot{\lambda}_p(\boldsymbol{\beta}_t) = \sum_{i=1}^{N}\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\{y_i - p(\mathbf{x}_i, \boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i.$$

Under Assumptions 1 and 2, conditional on $\mathcal{D}_N$, the consistent estimator $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$, if $n = o(N)$, then

$$\frac{\dot{\lambda}_p(\boldsymbol{\beta}_t)}{\sqrt{n}} - \frac{\sqrt{n}\sum_{i=1}^{N}\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)} \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\big),$$

in distribution; if $n/N \to \rho \in (0,1)$, then

$$\frac{\dot{\lambda}_p(\boldsymbol{\beta}_t)}{\sqrt{n}} - \frac{\sqrt{n}\sum_{i=1}^{N}\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)} \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Lambda}_\rho\big),$$

in distribution.

Lemma 33 *Under Assumptions 1 and 2, as n_0, n, and N go to infinity, for any $\mathbf{s}_n \to 0$ in probability,*

$$\frac{1}{n}\sum_{i=1}^{N}\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n)\|\mathbf{x}_i\|^2 - \sum_{i=1}^{N}\pi_i^p(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\|\mathbf{x}_i\|^2 = o_P(1).$$

Proof of Theorem 6. The estimator $\hat{\boldsymbol{\beta}}_p$ is the maximizer of (33), so $\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t)$ is the maximizer of $\gamma_p(\mathbf{s}) = \lambda_p(\boldsymbol{\beta}_t + \mathbf{s}/\sqrt{n}) - \lambda_p(\boldsymbol{\beta}_t)$. By Taylor's expansion,

$$\gamma_p(\mathbf{s}) = \frac{1}{\sqrt{n}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}_p(\boldsymbol{\beta}_t) + \frac{1}{2n}\sum_{i=1}^{N}\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i)^2$$

where $\phi_i(\boldsymbol{\beta}) = p(\mathbf{x}_i, \boldsymbol{\beta})\{1 - p(\mathbf{x}_i, \boldsymbol{\beta})\}$, and $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$.

From Lemmas 32 and 33, conditional on $\mathcal{D}_N$, and $\hat{\boldsymbol{\beta}}_0$,

$$\frac{1}{n}\sum_{i=1}^{N}\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1} + o_P(1).$$

In addition, from Lemma 32, conditional on $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$, $\dot{\lambda}_p(\boldsymbol{\beta}_t)/\sqrt{n}$ converges in distribution to a normal limit. Thus, from the Basic Corollary in page 2 of Hjort and Pollard (2011), the maximizer of $\gamma_p(\mathbf{s})$, $\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t)$, satisfies

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t) = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\frac{1}{\sqrt{n}}\dot{\lambda}_p(\boldsymbol{\beta}_t) + o_P(1)$$

given $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$. Combining this with Lemma 32, Slutsky's theorem, and the fact that a conditional probability is bounded, Theorem 6 follows. ■

B.2.1. Proof of Proposition 8

Proof of Proposition 8. To prove that $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\boldsymbol{\Lambda}_\rho\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} < \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}$, we just need to show that $\boldsymbol{\Lambda}_\rho < \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}$. This is true because

$$\begin{aligned}
\boldsymbol{\Lambda}_\rho &= \frac{\mathbb{E}\big[|\psi(\boldsymbol{\beta}_t)|\{\Psi(\boldsymbol{\beta}_t) - \rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\}_+ h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big]}{4\Psi^2(\boldsymbol{\beta}_t)}\\
&< \frac{\mathbb{E}\big\{|\psi(\boldsymbol{\beta}_t)|\Psi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big\}}{4\Psi^2(\boldsymbol{\beta}_t)} = \frac{\mathbb{E}\big\{|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big\}}{4\Psi(\boldsymbol{\beta}_t)} = \frac{\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big\}}{4\Phi(\boldsymbol{\beta}_t)} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}.
\end{aligned}$$

■

B.2.2. Proof of Lemma 32

Proof of Lemma 32. Note that, $\delta_i^{\hat{\boldsymbol{\beta}}_0} = I\{u_i \le n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\}$, where u_i are i.i.d. with the standard uniform distribution. Thus, given $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$, $\dot{\lambda}_p(\boldsymbol{\beta}_t)$ is a sum of N independent random vectors. We now exam the mean and variance of $\dot{\lambda}_p(\boldsymbol{\beta}_t)$. Recall that $\boldsymbol{\eta}_i = |\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i$, and $\psi_i(\boldsymbol{\beta}) = y_i - p(\mathbf{x}_i, \boldsymbol{\beta})$. For the mean, we have,

$$
\begin{aligned}
&\frac{1}{\sqrt{n}}\mathbb{E}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\}\\
&= \frac{1}{\sqrt{n}}\sum_{i=1}^N \{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\wedge 1\}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\\
&= \frac{1}{\sqrt{n}}\sum_{i=1}^N n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i = \frac{\sqrt{n}}{\sqrt{N}}\frac{\sum_{i=1}^N \boldsymbol{\eta}_i}{\hat{\Psi}_0\sqrt{N}} = O_P(\sqrt{n/N}),
\end{aligned}
$$

where the last equality is from Lemma 29.

For the variance,

$$
\begin{aligned}
&\frac{1}{n}\mathbb{V}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\}\\
=&\frac{1}{n}\sum_{i=1}^N [\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\wedge 1\} - \{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\wedge 1\}^2]\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}^2\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\
=&\sum_{i=1}^N \pi_i^p(\hat{\boldsymbol{\beta}}_0)\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} - n\sum_{i=1}^N \{\pi_i^p(\hat{\boldsymbol{\beta}}_0)\}^2\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\
=&\frac{\frac{1}{N}\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\hat{\Psi}_0}\\
&-\frac{n}{N}\frac{\frac{1}{N}\sum_{i=1}^N \psi_i^2(\hat{\boldsymbol{\beta}}_0)\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\hat{\Psi}_0^2}\\
\equiv&\Delta_3 - \Delta_4
\end{aligned}
\tag{34}
$$

Note that $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$, $\mathbb{E}\{h^2(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$, and $|\psi_i(\cdot)|$ are bounded. Thus, from Lemma 28, if $n/N \to \rho$,

$$
\Delta_4 \to \rho\frac{\mathbb{E}\{\psi^2(\boldsymbol{\beta}_t)h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}}{4\Psi^2(\boldsymbol{\beta}_t)},
\tag{35}
$$

in probability.

For the term Δ_3 in (34), it is equal to

$$
\begin{aligned}
\Delta_3 =& \frac{1}{\hat{\Psi}_0^2}\frac{1}{N}\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|\Big\{\frac{n|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)}{N}\vee\hat{\Psi}_0\Big\}\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\
=& \frac{1}{\hat{\Psi}_0^2}\frac{n}{N^2}\sum_{i=1}^N \psi_i^2(\hat{\boldsymbol{\beta}}_0)\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} I\Big\{\frac{n|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)}{N} > \hat{\Psi}_0\Big\}\\
&+ \frac{1}{\hat{\Psi}_0}\frac{1}{N}\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} I\Big\{\frac{n|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)}{N} \le \hat{\Psi}_0\Big\}.
\end{aligned}
$$

Since $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$, $\mathbb{E}\{h^2(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$, and $|\psi_i(\cdot)|$ are bounded, from Lemma 28, if $n/N \to \rho$, as n_0, n, and N go to infinity,

$$
\begin{aligned}
\Delta_3 \to& \frac{\rho\mathbb{E}\big[\psi^2(\boldsymbol{\beta}_t)h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}} I\big\{\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x}) \ge \Psi(\boldsymbol{\beta}_t)\big\}\big]}{4\Psi^2(\boldsymbol{\beta}_t)}\\
&+ \frac{\mathbb{E}\big[|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}} I\big\{\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x}) \le \Psi(\boldsymbol{\beta}_t)\big\}\big]}{4\Psi(\boldsymbol{\beta}_t)}\\
=& \frac{\mathbb{E}\Big(|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big[\{\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\}\vee\Psi(\boldsymbol{\beta}_t)\}\big]\Big)}{4\Psi^2(\boldsymbol{\beta}_t)},
\end{aligned}
\tag{36}
$$

in probability. From, (34), (35), and (36), if $n/N \to \rho$,

$$
\frac{1}{n}\mathbb{V}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\} = \frac{\mathbb{E}\big[|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\{\Psi(\boldsymbol{\beta}_t) - \rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\}_+\big]}{4\Psi^2(\boldsymbol{\beta}_t)} + o_P(1).
$$

Specifically, when $\rho = 0$,

$$
\frac{1}{n}\mathbb{V}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} + o_P(1).
$$

Now we check the Lindeberg-Feller condition (Section *2.8 of van der Vaart, 1998) under the condition distribution. Denote $\dot{\lambda}_{pi} = \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i$. For any $\epsilon > 0$

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^N \mathbb{E}\Big\{\|\dot{\lambda}_{pi}\|^2 I(\|\dot{\lambda}_{pi}\| > \sqrt{n}\epsilon)\Big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big\}\\
&\le \frac{1}{n}\sum_{i=1}^N \mathbb{E}\Big[\big\|\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\mathbf{x}_i\big\|^2 I(\big\|\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\mathbf{x}_i\big\| > \sqrt{n}\epsilon)\Big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big]\\
&= \sum_{i=1}^N \pi_i^p(\hat{\boldsymbol{\beta}}_0)\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\|\mathbf{x}_i\|^2 I(\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\|\mathbf{x}_i\| > \sqrt{n}\epsilon)\\
&\le \frac{|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\{n/N|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i) + \hat{\Psi}_0\}\|\mathbf{x}_i\|^2 I(\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)+1\}\|\mathbf{x}_i\| > \sqrt{n}\epsilon)}{\hat{\Psi}_0^2}\\
&\le \frac{\frac{1}{N}\sum_{i=1}^N h^2(\mathbf{x}_i)\|\mathbf{x}_i\|^2 I(\{h(\mathbf{x}_i)/\hat{\Psi}_0 + 1\}\|\mathbf{x}_i\| > \sqrt{n}\epsilon)}{\hat{\Psi}_0^2}\\
&+ \frac{\frac{1}{N}\sum_{i=1}^N h(\mathbf{x}_i)\|\mathbf{x}_i\|^2 I(\{h(\mathbf{x}_i)/\hat{\Psi}_0 + 1\}\|\mathbf{x}_i\| > \sqrt{n}\epsilon)}{\hat{\Psi}_0} = o_P(1),
\end{aligned}
$$

where the last equality is from Lemma 28. Thus, applying the Lindeberg-Feller central limit theorem (Section *2.8 of van der Vaart, 1998) finishes the proof. ■

B.2.3. Proof of Lemma 33

Proof of Lemma 33. Note that, from Lemma 28,

$$
\sum_{i=1}^N \pi_i^p(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \frac{1}{\hat{\Psi}_0 N}\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} + o_P(1);
$$

and from the strong law of large numbers

$$
\frac{1}{n}\sum_{i=1}^N \delta_i^{\boldsymbol{\beta}_t}\{n\pi_i^p(\boldsymbol{\beta}_t)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \boldsymbol{\beta}_t)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t} + o_P(1),
$$

where $\delta_i^{\boldsymbol{\beta}_t} = I\{u_i \le n\pi_i^p(\boldsymbol{\beta}_t)\}$. Thus, if we show that

$$
\Delta_5 \equiv \frac{1}{n}\sum_{i=1}^N \Big|\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n) - \delta_i^{\boldsymbol{\beta}_t}\{n\pi_i^p(\boldsymbol{\beta}_t)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \boldsymbol{\beta}_t)\Big|\|\mathbf{x}_i\|^2 = o_P(1),
$$

then the result in Lemma 33 follows. Noting that Δ_5 is nonnegative, we prove $\Delta_5 == o_P(1)$ by showing that $\mathbb{E}(\Delta_5|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0) = o_P(1)$. Note that given $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$, the only

random terms in Δ_5 are $\delta_i^{\hat{\boldsymbol{\beta}}_0} = I\{u_i \le n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\}$ and $\delta_i^{\boldsymbol{\beta}_t} = I\{u_i \le n\pi_i^p(\boldsymbol{\beta}_t)\}$. We have that

$$
\begin{aligned}
&\mathbb{E}(\Delta_5|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0)\\
\le&\frac{1}{n}\sum_{i=1}^N \{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\wedge n\pi_i^p(\boldsymbol{\beta}_t)\wedge 1\}\\
&\qquad\times\Big|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n) - \{n\pi_i^p(\boldsymbol{\beta}_t)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \boldsymbol{\beta}_t)\Big|\|\mathbf{x}_i\|^2\\
&+\frac{1}{n}\sum_{i=1}^N |n\pi_i^p(\hat{\boldsymbol{\beta}}_0) - n\pi_i^p(\boldsymbol{\beta}_t)|\Big|n\pi_i^p(\hat{\boldsymbol{\beta}}_0) + n\pi_i^p(\boldsymbol{\beta}_t) + 2\Big|\|\mathbf{x}_i\|^2\\
\equiv&\Delta_6 + \Delta_7.
\end{aligned}
$$

Note that $n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\wedge n\pi_i^p(\boldsymbol{\beta}_t)\wedge 1 \le n\pi_i^p(\hat{\boldsymbol{\beta}}_0)$. Thus Δ_6 is bounded by

$$
\frac{1}{\hat{\Psi}_0}\frac{1}{N}\sum_{i=1}^N \Big|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \mathbf{s}_n) - \{n\pi_i^p(\boldsymbol{\beta}_t)\vee 1\}\phi_i(\boldsymbol{\beta}_t - \boldsymbol{\beta}_t)\Big|h(\mathbf{x}_i)\|\mathbf{x}_i\|^2,
$$

which is $o_P(1)$ by Lemma 28 if $|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\} - \{n\pi_i^p(\boldsymbol{\beta}_t)\vee 1\}| = o_P(1)$. This is true because

$$
\begin{aligned}
|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\} - \{n\pi_i^p(\boldsymbol{\beta}_t)\vee 1\}| \le& n|\pi_i^p(\hat{\boldsymbol{\beta}}_0) - \pi_i^p(\boldsymbol{\beta}_t)|\\
\le&\frac{nh(\mathbf{x}_i)}{N}\left|\frac{|\psi_i(\hat{\boldsymbol{\beta}}_0)|}{\hat{\Psi}_0} - \frac{|\psi_i(\boldsymbol{\beta}_t)|}{\Psi_N(\boldsymbol{\beta}_t)}\right| = o_P(1).
\end{aligned}
$$

The term Δ_7 is bounded by

$$
\frac{1}{N}\sum_{i=1}^N \left|\frac{|\psi_i(\hat{\boldsymbol{\beta}}_0)|}{\hat{\Psi}_0} - \frac{|\psi_i(\boldsymbol{\beta}_t)|}{\Psi_N(\boldsymbol{\beta}_t)}\right|\left|\frac{|\psi_i(\hat{\boldsymbol{\beta}}_0)|}{\hat{\Psi}_0} + \frac{|\psi_i(\boldsymbol{\beta}_t)|}{\Psi_N(\boldsymbol{\beta}_t)} + \frac{2}{h(\mathbf{x}_i)}\right|h^2(\mathbf{x}_i)\|\mathbf{x}_i\|^2 = o_P(1),
$$

where the last equality is from Lemma 28 and the fact that $\mathbb{E}\{h^2(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$. ■

B.3. Proofs for unconditional distribution

In this section we prove Theorem 13 in Section 6. A lemma similar to Lemma 32 is presented below and will be proved later in this section. Lemma 33 can be used in the proof of Theorem 13 because for the problem considered in this paper, convergence to zero in probability is equivalent to convergence to zero in probability under the conditional probability measure (Xiong and Li, 2008).

For the pilot subsample taken according to the subsampling probabilities π_{0i} in (17), we define $\delta_i^{(1)} = I\{u_{0i} \le \frac{c_0(1-y_i)+c_1 y_i}{N}\}$, where u_{0i} are i.i.d. standard uniform random variables. With this notation, the estimator $\hat{\Psi}_0$ defined in (18) can be written as

$$
\hat{\Psi}_0 = \frac{1}{N}\sum_{i=1}^N \frac{\delta_i^{(1)}|y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)}{n\pi_{0i}\wedge 1}. \tag{37}
$$

Lemma 34 *Let $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ be constructed according to Step 1 of Algorithm 2, respectively. For*

$$\dot{\lambda}_p(\boldsymbol{\beta}_t) = \sum_{i=1}^{N} \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\{y_i - p(\mathbf{x}_i, \boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i,$$

under the same assumptions of Theorem 13, if $n = o(N)$, then

$$\frac{\dot{\lambda}_p(\boldsymbol{\beta}_t)}{\sqrt{n}} \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\big),$$

in distribution; if $n/N \to \rho \in (0,1)$, then

$$\frac{\dot{\lambda}_p(\boldsymbol{\beta}_t)}{\sqrt{n}} \longrightarrow \mathbb{N}\big(\mathbf{0},\ \boldsymbol{\Lambda}_u\big),$$

in distribution.

Proof of Theorem 13. The proof of this theorem is similar to that of Theorem 6. The key difference is that Lemma 34 is about asymptotic distribution unconditionally.

The estimator $\hat{\boldsymbol{\beta}}_p$ is the maximizer of

$$\lambda_p(\boldsymbol{\beta}) = \sum_{i=1}^{N} \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\big[(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)^{\mathrm{T}}\mathbf{x}_i y_i - \log\{1 + e^{(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_0)^{\mathrm{T}}\mathbf{x}_i}\}\big],$$

so $\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t)$ is the maximizer of $\gamma_p(\mathbf{s}) = \lambda_p(\boldsymbol{\beta}_t + \mathbf{s}/\sqrt{n}) - \lambda_p(\boldsymbol{\beta}_t)$. By Taylor's expansion,

$$\gamma_p(\mathbf{s}) = \frac{1}{\sqrt{n}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}_p(\boldsymbol{\beta}_t) + \frac{1}{2n}\sum_{i=1}^{N} \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i)^2$$

where $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$.

From Lemma 33,

$$\frac{1}{n}\sum_{i=1}^{N} \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\phi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})\mathbf{x}_i(\mathbf{x}_i)^{\mathrm{T}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1} + o_P(1).$$

In addition, from Lemma 34, $\dot{\lambda}_p(\boldsymbol{\beta}_t)/\sqrt{n}$ converges in distribution to a normal limit. Thus, from the Basic Corollary in page 2 of Hjort and Pollard (2011), the maximizer of $\gamma_p(\mathbf{s})$, $\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t)$, satisfies

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_t) = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}\frac{1}{\sqrt{n}}\dot{\lambda}_p(\boldsymbol{\beta}_t) + o_P(1).$$

Combining this with Lemma 34 and Slutsky's theorem, Theorem 13 follows. ■

B.3.1. Proof of Proposition 17

Proof of Proposition 17 To prove (22), we just need to show that $\boldsymbol{\Lambda}_u \geq \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1} > \boldsymbol{\Lambda}_\rho$. From Proposition 8, we know that $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1} > \boldsymbol{\Lambda}_\rho$. To show that $\boldsymbol{\Lambda}_u \geq \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1}$, we notice that

$$\begin{aligned}\boldsymbol{\Lambda}_u &= \frac{\mathbb{E}[\phi(\boldsymbol{\beta}_t)\{\rho\phi(\boldsymbol{\beta}_t)h(\mathbf{x}) \vee \Phi(\boldsymbol{\beta}_t)\}h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]}{4\Phi^2(\boldsymbol{\beta}_t)}. \\ &\geq \frac{\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\Phi(\boldsymbol{\beta}_t)\mathbf{x}\mathbf{x}^{\mathrm{T}}\big\}}{4\Phi^2(\boldsymbol{\beta}_t)} = \frac{\mathbb{E}\big\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big\}}{4\Phi(\boldsymbol{\beta}_t)} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1},\end{aligned}$$

where the strict inequality holds if $\rho\phi(\boldsymbol{\beta}_t)h(\mathbf{x}) \vee \Phi(\boldsymbol{\beta}_t) \neq \rho\phi(\boldsymbol{\beta}_t)h(\mathbf{x})$ with positive probability, i.e., $\mathbb{P}\{\rho\phi(\boldsymbol{\beta}_t)h(\mathbf{x}) > \Phi(\boldsymbol{\beta}_t)\} > 0$. ■

B.3.2. Proof of Lemma 34

Proof of Lemma 34.

We first proof the case when the pilot estimates $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ depend on the data. For any $l \in \mathbb{R}^d$, denote $\tau_{Ni} = \sqrt{N/n}\hat{\Psi}_0\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i^{\mathrm{T}}l$, $i = 1, ..., N$, where $\delta_i^{\hat{\boldsymbol{\beta}}_0} = I\{u_i \leq n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\}$, and u_i are i.i.d. standard uniform random variables. Note that τ_{Ni}'s have the same distribution but they are not independent. Again, since $n_0 = o(\sqrt{N})$, we can focus on τ_{Ni}'s that $(\mathbf{x}_i, y_i)$'s are not included in the pilot subsample. We now exam the mean and variance of these τ_{Ni}'s. For the mean, based on calculation similar to that in (30), we have,

$$\mathbb{E}\big(\tau_{Ni}\big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big) = \sqrt{nN}\hat{\Psi}_0\mathbb{E}\big\{\pi_i^p(\hat{\boldsymbol{\beta}}_0)\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i^{\mathrm{T}}l\big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\} = \frac{\sqrt{n}\mathbb{E}(\boldsymbol{\eta}_i\big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0)}{\sqrt{N}} = 0,$$

which implies that

$$\mathbb{E}\tau_{Ni} = 0.$$

For the variance, $\mathbb{V}(\tau_{Ni}) = \mathbb{E}(\tau_{Ni}^2)$, we start with the condition expectation,

$$\begin{aligned}\mathbb{E}\big(\tau_{Ni}^2\big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big) =& N\hat{\Psi}_0^2\mathbb{E}\Big[\pi_i^p(\hat{\boldsymbol{\beta}}_0)\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)(\mathbf{x}_i^{\mathrm{T}}l)^2\Big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big] \\ =& \mathbb{E}\Big[|\psi_i(\hat{\boldsymbol{\beta}}_0)|\Big\{\frac{n}{N}|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i) \vee \hat{\Psi}_0\Big\}\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}}l)^2\Big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big].\end{aligned}$$

If we let

$$\Upsilon_{Ni} = |\psi_i(\hat{\boldsymbol{\beta}}_0)|\Big\{\frac{n}{N}|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i) \vee \hat{\Psi}_0\Big\}\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}}l)^2,$$

then $\mathbb{V}(\tau_{Ni}) = \mathbb{E}(\Upsilon_{Ni})$. Note that

$$\Upsilon_{Ni} \to \Upsilon_i = 0.25|\psi_i(\boldsymbol{\beta}_t)|\{\rho|\psi_i(\boldsymbol{\beta}_t)|h(\mathbf{x}_i) \vee \Psi(\boldsymbol{\beta}_t)\}h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}}l)^2,$$

in probability. We now show that

$$\mathbb{E}(\Upsilon_{Ni}) \to \mathbb{E}(\Upsilon_i) = 0.25\mathbb{E}[|\psi(\boldsymbol{\beta}_t)|\{\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x}) \vee \Psi(\boldsymbol{\beta}_t)\}h(\mathbf{x})(\mathbf{x}^{\mathrm{T}}l)^2].$$

Let $\Xi_i = |\Upsilon_{Ni} - \Upsilon_i|$. For any ϵ,

$$\begin{aligned}|\mathbb{E}(\Upsilon_{Ni}) - \mathbb{E}(\Upsilon_i)| &\le \mathbb{E}\{\Xi_i I(\Xi_i > \epsilon)\} + \mathbb{E}\{\Xi_i I(\Xi_i \le \epsilon)\} \\ &\le \mathbb{E}\big[\{h^2(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2 + \Upsilon_i + \hat{\Psi}_0 h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2\} I(\Xi_i > \epsilon)\big] + \epsilon\end{aligned}$$

We know that $\mathbb{E}\big[\{h^2(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2 + \Upsilon_i\} I(\Xi_i > \epsilon)\big] \to 0$ since $\mathbb{E}\{h^2(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2 + \Upsilon_i\} < \infty$ for any $l \in \mathbb{R}^d$, and $I(\Xi_i > \epsilon)$ is bounded and is $o_P(1)$. Similarly, $\mathbb{E}\{\hat{\Psi}_0 h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2 I(\Xi_i > \epsilon)\} \le \mathbb{E}\{h(\mathbf{x})\}\mathbb{E}\{h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2 I(\Xi_i > \epsilon)\} \to 0$. Thus, $\mathbb{E}(\Upsilon_{Ni}) - \mathbb{E}(\Upsilon_i) \to 0$, and we have finished proving that

$$\mathbb{V}(\tau_{Ni}) \to \mathbb{E}(\Upsilon_i). \tag{38}$$

In the following, we exam the third moment of τ_{Ni} and prove that

$$\mathbb{E}|\tau_{Ni}|^3 = o(\sqrt{N}). \tag{39}$$

For the conditional expectation,

$$\begin{aligned}&\mathbb{E}\big(|\tau_{Ni}|^3 \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big) \\ &= N\sqrt{N/n}\hat{\Psi}_0^3 \mathbb{E}\Big[\pi_i^p(\hat{\boldsymbol{\beta}}_0)\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}^2 \psi_i^3(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)(\mathbf{x}_i^{\mathrm{T}} l)^3 \Big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big] \\ &= \sqrt{N/n}\mathbb{E}\Big[|\psi_i(\hat{\boldsymbol{\beta}}_0)| \Big\{\frac{n}{N}|\psi_i(\hat{\boldsymbol{\beta}}_0)| h(\mathbf{x}_i) \vee \hat{\Psi}_0\Big\}^2 \psi_i^3(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0) h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^3 \Big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big] \\ &\le 2\|l\|^3 \sqrt{N/n}\mathbb{E}\big[\{h^2(\mathbf{x}_i) + \hat{\Psi}_0^2\} h(\mathbf{x}_i)\|\mathbf{x}_i\|^3 \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big] \\ &\le 2\|l\|^3 \sqrt{N/n}\big[\mathbb{E}\{h^3(\mathbf{x}_i)\|\mathbf{x}_i\|^3\} + \mathbb{E}\{\hat{\Psi}_0^2 | \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\}\mathbb{E}\{h(\mathbf{x}_i)\|\mathbf{x}_i\|^3\}\big].\end{aligned}$$

Since $\mathbb{E}\{h^3(\mathbf{x}_i)\|\mathbf{x}_i\|^3\} < \infty$ and $\mathbb{E}\{h(\mathbf{x}_i)\|\mathbf{x}_i\|^3\} < \infty$, (39) follows if $\mathbb{E}\{\hat{\Psi}_0^2\} = O(1)$. This is true because

$$\begin{aligned}\mathbb{E}\{\hat{\Psi}_0^2\} &= \mathbb{E}\Big\{\frac{1}{N}\sum_{k_1=1}^N \frac{\delta_{k_1}^{(1)}|y_{k_1} - p(\mathbf{x}_{k_1}, \hat{\boldsymbol{\beta}}_0)| h(\mathbf{x}_{k_1})}{n_0/N} \frac{1}{N}\sum_{k_2=1}^N \frac{\delta_{k_2}^{(1)}|y_{k_2} - p(\mathbf{x}_{k_2}, \hat{\boldsymbol{\beta}}_0)| h(\mathbf{x}_{k_2})}{n_0/N}\Big\} \\ &\le \frac{1}{N^2}\sum_{k_1 \ne k_2}^N \mathbb{E}\Big\{\frac{\delta_{k_1}^{(1)} h(\mathbf{x}_{k_1})}{n_0/N}\frac{\delta_{k_2}^{(1)} h(\mathbf{x}_{k_2})}{n_0/N}\Big\} + \frac{1}{N^2}\sum_{k=1}^N \mathbb{E}\Big\{\frac{\delta_k^{(1)} h^2(\mathbf{x}_k)}{\{n_0/N\}^2}\Big\} \\ &= \frac{1}{N^2}\sum_{k_1 \ne k_2}^N \mathbb{E}\{h(\mathbf{x}_{k_1})h(\mathbf{x}_{k_2})\} + \frac{1}{N^2}\sum_{j=1}^N \mathbb{E}\Big\{\frac{1}{n_0/N}h^2(\mathbf{x}_k)\Big\} \to \mathbb{E}\{h^2(\mathbf{x})\}.\end{aligned}$$

Denote $\nu_{Ni} = \tau_{Ni}\{\mathbb{V}(\tau_{Ni})\}^{-1/2}$. We know that ν_{Ni}'s, for which $(\mathbf{x}_i, y_i)$'s are not included in the pilot subsample, are i.i.d. conditional on $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$. Thus, from Theorem 7.3.2 of Chow and Teicher (2003), they are interchangeable. The fact that $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ are consistent estimators implies that they are a sequence of two estimators, and for each $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$, τ_{Ni} are interchangeable and can be . For this setup, the central limit theorem in Theorem 2 of Blum et al. (1958) can be applied to prove the asymptotic normality.

It is evident that ν_{Ni} have mean 0 and variance 1. It is also easy to verify that, for $i \ne j$,

$$\mathbb{E}(\nu_{Ni}\nu_{Nj}) = \mathbb{E}\{\mathbb{E}(\nu_{Ni}\nu_{Nj} | \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0)\} = 0, \tag{40}$$

and

$$\frac{1}{\sqrt{N}}\mathbb{E}\{|\nu_{Ni}|^3\} = \mathbb{E}|\tau_{Ni}|^3\{\mathbb{V}(\tau_{Ni})\}^{-3/2} \to 0 \tag{41}$$

which follows from (39). We now show that for $i \ne j$,

$$\mathbb{E}\{\nu_{Ni}^2 \nu_{Nj}^2\} \to 1. \tag{42}$$

Since $\nu_{Ni} = \tau_{Ni}\{\mathbb{V}(\tau_{Ni})\}^{-1/2}$, from (38), to prove (42), we only need to show that $\mathbb{E}(\tau_{Ni}^2\tau_{Nj}^2) \to \mathbb{E}(\Upsilon_i)\mathbb{E}(\Upsilon_j) = \mathbb{E}(\Upsilon_i\Upsilon_j)$, where the equality is because Υ_i and Υ_j are independent. Noting that τ_{Ni}^2 and τ_{Nj}^2 are conditionally independent, we have $\mathbb{E}\big(\tau_{Ni}^2\tau_{Nj}^2 \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big) = \mathbb{E}\big(\tau_{Ni}^2 \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big)\mathbb{E}\big(\tau_{Ni}^2 \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big) = \mathbb{E}\big(\Upsilon_{Ni} \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big)\mathbb{E}\big(\Upsilon_{Nj} \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big) = \mathbb{E}\big(\Upsilon_{Ni}\Upsilon_{Nj} \big| \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big)$, so we know that $\mathbb{E}\big(\tau_{Ni}^2\tau_{Nj}^2\big) = \mathbb{E}\big(\Upsilon_{Ni}\Upsilon_{Nj}\big)$.

Now we prove that $\mathbb{E}\big(\Upsilon_{Ni}\Upsilon_{Nj}\big) \to \mathbb{E}\big(\Upsilon_i\Upsilon_j\big)$. Let $\Xi_{ij} = |\Upsilon_{Ni}\Upsilon_{Nj} - \Upsilon_i\Upsilon_j|$. For any $\epsilon > 0$,

$$\begin{aligned}&|\mathbb{E}(\Upsilon_{Ni}\Upsilon_{nj}) - \mathbb{E}(\Upsilon_i\Upsilon_j)| \\ &\le \mathbb{E}\{\Xi_{ij} I(\Xi_{ij} > \epsilon)\} + \mathbb{E}\{\Xi_{ij} I(\Xi_{ij} \le \epsilon)\} \\ &\le \mathbb{E}\big[\{h^2(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2 h^2(\mathbf{x}_j)(\mathbf{x}_j^{\mathrm{T}} l)^2 + \Upsilon_i\Upsilon_j\} I(\Xi_{ij} > \epsilon)\big] \\ &\quad + \mathbb{E}\big[\hat{\Psi}_0^2 h(\mathbf{x}_i)\|\mathbf{x}_i\|^2 h(\mathbf{x}_j)\|\mathbf{x}_j\|^2 I(\Xi_{ij} > \epsilon)\big] \\ &\quad + \mathbb{E}(\hat{\Psi}_0)\mathbb{E}\big[\{h(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2 h^2(\mathbf{x}_j)(\mathbf{x}_j^{\mathrm{T}} l)^2 + h(\mathbf{x}_j)(\mathbf{x}_j^{\mathrm{T}} l)^2 h^2(\mathbf{x}_i)(\mathbf{x}_i^{\mathrm{T}} l)^2\} I(\Xi_{ij} > \epsilon)\big] + \epsilon\end{aligned}$$

Since $i \ne j$, $\mathbb{E}\{h^2(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$, $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^2\} < \infty$, $\mathbb{E}(\hat{\Psi}_0) < \infty$, and $I(\Xi_{ij} > \epsilon) = o_P(1)$ is bounded, the above results indicates that (42) holds.

Since (40), (41), (42) are satisfied, the central limit theorem in Theorem 2 of Blum et al. (1958) holds for ν_{Ni}, which gives that

$$\frac{1}{\sqrt{N}}\sum_{i=1}^N \nu_{Ni} \to \mathbb{N}(0, 1),$$

in distribution. Note that

$$\begin{aligned}\frac{1}{\sqrt{N}}\sum_{i=1}^N \nu_i &= \frac{\hat{\Psi}_0}{\sqrt{n}\{\mathbb{V}(\tau_{Ni})\}^{1/2}}\sum_{i=1}^N \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i^{\mathrm{T}} l \\ &= \frac{\hat{\Psi}_0}{\sqrt{n}\{\mathbb{V}(\tau_{Ni})\}^{1/2}} l^{\mathrm{T}}\dot{\lambda}_p(\boldsymbol{\beta}_t) = \frac{\Psi}{\sqrt{n}\{\mathbb{V}(\tau_{Ni})\}^{1/2}} l^{\mathrm{T}}\dot{\lambda}_p(\boldsymbol{\beta}_t) + o_P(1).\end{aligned}$$

Thus, from Slutsky's theorem, for any $l \in \mathbb{R}^d$,

$$\frac{1}{\sqrt{n}} l^{\mathrm{T}}\dot{\lambda}_p(\boldsymbol{\beta}_t) \to \mathbb{N}(0, l^{\mathrm{T}}\boldsymbol{\Lambda}_u l) \tag{43}$$

in distribution, where

$$\begin{aligned}\boldsymbol{\Lambda}_u = \frac{\mathbb{V}(\tau_{Ni})}{\Psi^2(\boldsymbol{\beta}_t)} &= \frac{\mathbb{E}[|\psi(\boldsymbol{\beta}_t)|\{\rho|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x}) \vee \Psi(\boldsymbol{\beta}_t)\}h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]}{4\Psi^2(\boldsymbol{\beta}_t)} \\ &\ge \frac{\mathbb{E}[|\psi(\boldsymbol{\beta}_t)|h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]}{4\Psi(\boldsymbol{\beta}_t)} = \frac{\mathbb{E}[\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]}{4\Phi(\boldsymbol{\beta}_t)} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}_t}^{-1},\end{aligned}$$

and the equality holds if $\rho = 0$, i.e., $n/N \to 0$. Based on (43), from the Cramér-Wold theorem, we have that

$$\frac{1}{\sqrt{n}}\dot{\lambda}_p(\boldsymbol{\beta}_t) \to \mathbb{N}(0, \boldsymbol{\Lambda}_u)$$

in distribution.

When the pilot estimates $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$ are independent of the data, if we can prove the results in Lemma 34 under the conditional distribution given $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$, then the result follows unconditionally. We provide the proof under the conditional distribution in the following. The proof is similar to the proof of Lemma 32 and thus we provide only the outline. The difference is we do not conditional on the full data $\mathcal{D}_N$ here.

Note that, given $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$, $\dot{\lambda}_p(\boldsymbol{\beta}_t)$ is a sum of N independent random vectors. We now exam the mean and variance of $\dot{\lambda}_p(\boldsymbol{\beta}_t)$ given $\hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$. For the mean,

$$\frac{1}{\sqrt{n}}\mathbb{E}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\} = \mathbf{0}.$$

For the variance,

$$\frac{1}{n}\mathbb{V}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\} = \frac{\mathbb{E}\Big[|\psi_i(\hat{\boldsymbol{\beta}}_0)|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\Big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big]}{\hat{\Psi}_0},$$

which, under Assumptions 1 and 2, converges in probability to $\boldsymbol{\Lambda}_u$.

To check the Lindeberg-Feller condition (Section *2.8 of van der Vaart, 1998) under the condition distribution, we note that for any $\epsilon > 0$,

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^N \mathbb{E}\Big\{\big\|\dot{\lambda}_{pi}\big\|^2 I(\big\|\dot{\lambda}_{pi}\big\| > \sqrt{n}\epsilon)\Big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big\} \\
&\le \frac{\mathbb{E}\Big[\{h(\mathbf{x})\|\mathbf{x}\|^2 + \hat{\Psi}_0\}h(\mathbf{x})I(\{h(\mathbf{x})/\hat{\Psi}_0 + 1\}\|\mathbf{x}\| > \sqrt{n}\epsilon)\Big|\hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\Big]}{\hat{\Psi}_0^2} = o_P(1).
\end{aligned}$$

Thus, applying the Lindeberg-Feller central limit theorem (Section *2.8 of van der Vaart, 1998) finishes the proof.

■

B.4. Proofs for cases of misspecifications

B.4.1. Proofs with pilot misspecification

Proof of Theorem 18. By similar arguments used in the proof of Theorem 1, we know that $\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_t)$ is the maximizer of

$$\gamma(\mathbf{s}) = \frac{1}{\sqrt{n}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_t) + \frac{1}{2n}\sum_{i=1}^n \phi_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i^*)^2,$$

where $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$, and

$$\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_t) = \sum_{i=1}^n \{y_i^* - p_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i^*.$$

Given $\mathcal{D}_N$ and $\hat{\boldsymbol{\beta}}_0$, $\{y_i^* - p_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i^*$ are i.i.d. random vectors. We exam their mean and variance, and check the Lindeberg-Feller condition under the conditional distribution. For the expectation, direct calculations give

$$\mathbb{E}\big[\{y^* - p(\mathbf{x}_i^*, \boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^*\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big] = \sum_{i=1}^N \pi_i(\hat{\boldsymbol{\beta}}_0)\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i = \frac{\sum_{i=1}^N \boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)}, \tag{44}$$

where $\boldsymbol{\eta}_i = |\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i$. Conditional on $\hat{\boldsymbol{\beta}}_0$, $\boldsymbol{\eta}_i$'s are i.i.d., and we still have $\mathbb{E}(\boldsymbol{\eta}_i|\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0) = \mathbf{0}$ and thus $\mathbb{E}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0) = \mathbf{0}$ due to (30). Thus

$$\begin{aligned}
\mathbb{V}(\boldsymbol{\eta}_i|\hat{\boldsymbol{\beta}}_0) = \mathbb{E}(\boldsymbol{\eta}_i\boldsymbol{\eta}_i^{\mathrm{T}}|\hat{\boldsymbol{\beta}}_0) &= \mathbb{E}\big\{\psi^2(\hat{\boldsymbol{\beta}}_0)\psi^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big|\hat{\boldsymbol{\beta}}_0\big\} \\
&= \mathbb{E}\big\{\psi^2(\boldsymbol{\beta}_0)\psi^2(\boldsymbol{\beta}_t - \boldsymbol{\beta}_0)h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\big\} + o_P(1) = \boldsymbol{\varsigma}_b + o_P(1),
\end{aligned}$$

where the third equality is from Lemma 28 and the facts that $\psi^2(\cdot) \le 1$ and $\mathbb{E}\{h^2(\mathbf{x})\|\mathbf{x}|^2\} < \infty$.

Similar to the proof of Lemma 29, the Lindeberg-Feller central limit theorem applies conditional on $\hat{\boldsymbol{\beta}}_0$. Thus, we have that, conditional on $\hat{\boldsymbol{\beta}}_0$,

$$\frac{\sum_{i=1}^N \boldsymbol{\eta}_i}{\sqrt{N}} \longrightarrow \mathbb{N}(\mathbf{0},\ \boldsymbol{\varsigma}_b), \tag{45}$$

in distribution.

From (44) and (45), we have

$$\Delta = \mathbb{E}\big[\{y^* - p(\mathbf{x}_i^*, \boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^*\big|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big] = O_P(1/\sqrt{N}). \tag{46}$$

For the conditional variance of $\{y_i^* - p_i^*(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i^*$, using similar approach to the proof of Lemma 30, we have

$$\begin{aligned}
&\mathbb{V}\big[\{y^* - p(\mathbf{x}_i^*, \boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^*|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0\big] \\
=&\sum_{i=1}^N \pi_i(\hat{\boldsymbol{\beta}}_0)\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} - \Delta^2 \\
=&\frac{\frac{1}{N}\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i^2(\boldsymbol{\beta}_t - \hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\Psi_N(\hat{\boldsymbol{\beta}}_0)} - O_P(1/N) \\
=&\frac{\mathbb{E}\{|\psi(\boldsymbol{\beta}_0)|\psi^2(\boldsymbol{\beta}_t - \boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}}{\Psi(\boldsymbol{\beta}_0)} + o_P(1) = \frac{\boldsymbol{\varsigma}_a}{\Psi(\boldsymbol{\beta}_0)} + o_P(1)
\end{aligned} \tag{47}$$

The Lindeberg-Feller condition under the conditional distribution can be verified similarly to the proof of Lemma 30. Thus, we have

$$\frac{\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_t)}{\sqrt{n}} - \frac{\sqrt{n}\sum_{i=1}^N \boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)} \longrightarrow \mathbb{N}\Big\{\mathbf{0},\ \frac{\boldsymbol{\varsigma}_a}{\Psi(\boldsymbol{\beta}_0)}\Big\}, \tag{48}$$

in conditional distribution.

From Lemmas 28 and 31, and the law of large numbers, we have

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\phi_i^*(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0+\acute{\mathbf{s}}/\sqrt{n})\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}}\\
&=\sum_{i=1}^{N}\pi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}+o_P(1)\\
&=\frac{\frac{1}{N}\sum_{i=1}^{N}|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\Psi_N(\hat{\boldsymbol{\beta}}_0)}+o_P(1)\\
&=\frac{\mathbb{E}\{|\psi(\boldsymbol{\beta}_0)|\phi(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}}{\Psi(\boldsymbol{\beta}_0)}+o_P(1)=\frac{\boldsymbol{\varsigma}_a}{\Psi(\boldsymbol{\beta}_0)}+o_P(1).
\end{aligned}$$

Since $\boldsymbol{\varsigma}_a$ is a positive definite matrix, and combining (44), (45), (46) and (47) we know that $\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_t)/\sqrt{n}$ is stochastically bounded, from the Basic Corollary in page 2 of Hjort and Pollard (2011), we have that

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw}-\boldsymbol{\beta}_t)=\Psi(\boldsymbol{\beta}_0)\boldsymbol{\varsigma}_a^{-1}\frac{1}{\sqrt{n}}\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_t)+o_P(1) \tag{49}$$

given $\mathcal{D}_N$ and $\hat{\boldsymbol{\beta}}_0$.

Now note that $\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}-\boldsymbol{\beta}_t)$ is the maximizer of

$$\frac{1}{\sqrt{N}}\mathbf{s}^{\mathrm{T}}\sum_{i=1}^{N}\boldsymbol{\eta}_i-\frac{1}{2N}\sum_{i=1}^{N}|y_i-p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0+\acute{\mathbf{s}}/\sqrt{N})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i)^2$$

with $\acute{\mathbf{s}}$ between $\mathbf{0}$ and $\mathbf{s}$. From Lemma 28, conditional on $\hat{\boldsymbol{\beta}}_0$,

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}|y_i-p(\mathbf{x}_i,\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0+\acute{\mathbf{s}}/\sqrt{N})\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\
&=\frac{1}{N}\sum_{i=1}^{N}|y_i-p(\mathbf{x}_i,\boldsymbol{\beta}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}+o_P(1)\\
&=\mathbb{E}\{|\psi(\boldsymbol{\beta}_0)|\phi_i(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\}+o_P(1)=\boldsymbol{\varsigma}_a+o_P(1)
\end{aligned}$$

Thus, from the Basic Corollary in page 2 of Hjort and Pollard (2011), we have

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}-\boldsymbol{\beta}_t)=\boldsymbol{\varsigma}_a^{-1}\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\boldsymbol{\eta}_i+o_P(1). \tag{50}$$

Combining this with (49), we have

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw}-\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}})=\Psi(\boldsymbol{\beta}_0)\boldsymbol{\varsigma}_a^{-1}\Big\{\frac{1}{\sqrt{n}}\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_t)-\frac{\sqrt{n}}{N\Psi(\boldsymbol{\beta}_0)}\sum_{i=1}^{N}\boldsymbol{\eta}_i\Big\}+o_P(1).$$

Combining the above two equations with (48), Slutsky's theorem, and the fact that a conditional probability is bounded, Theorem 18 follows. ■

Proof of Remark 20 We first observe the following equations by direct calculations.

$$\begin{aligned}
\Psi(\boldsymbol{\beta}_0)&=\mathbb{E}\big[\{p(\mathbf{x},\boldsymbol{\beta}_t)+p(\mathbf{x},\boldsymbol{\beta}_0)-2p(\mathbf{x},\boldsymbol{\beta}_0)p(\mathbf{x},\boldsymbol{\beta}_t)\}h(\mathbf{x})\big],\ \text{and}\\
p(\mathbf{x},\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)&=\frac{p(\mathbf{x},\boldsymbol{\beta}_t)\{1-p(\mathbf{x},\boldsymbol{\beta}_0)\}}{p(\mathbf{x},\boldsymbol{\beta}_t)\{1-p(\mathbf{x},\boldsymbol{\beta}_0)\}+p(\mathbf{x},\boldsymbol{\beta}_0)\{1-p(\mathbf{x},\boldsymbol{\beta}_t)\}}.
\end{aligned} \tag{51}$$

We need to verify that

$$\frac{\mathbb{E}[\{1-p(\mathbf{x},\boldsymbol{\beta}_t)\}p(\mathbf{x},\boldsymbol{\beta}_0)p(\mathbf{x},\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]}{\mathbb{E}\big[\{p(\mathbf{x},\boldsymbol{\beta}_t)+p(\mathbf{x},\boldsymbol{\beta}_0)-2p(\mathbf{x},\boldsymbol{\beta}_0)p(\mathbf{x},\boldsymbol{\beta}_t)\}h(\mathbf{x})\big]}<\frac{\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}}{4\mathbb{E}\{\phi(\boldsymbol{\beta}_t)h(\mathbf{x})\}}$$

Note that from (51)

$$\begin{aligned}
&2\{1-p(\mathbf{x},\boldsymbol{\beta}_t)\}p(\mathbf{x},\boldsymbol{\beta}_0)p(\mathbf{x},\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)<\phi(\boldsymbol{\beta}_t)\\
\Leftrightarrow\ &\{p(\mathbf{x},\boldsymbol{\beta}_t)-p(\mathbf{x},\boldsymbol{\beta}_0)\}\{1-2p(\mathbf{x},\boldsymbol{\beta}_0)\}>0,
\end{aligned}$$

and

$$\begin{aligned}
&p(\mathbf{x},\boldsymbol{\beta}_t)+p(\mathbf{x},\boldsymbol{\beta}_0)-2p(\mathbf{x},\boldsymbol{\beta}_0)p(\mathbf{x},\boldsymbol{\beta}_t)>2\phi(\boldsymbol{\beta}_t)\\
\Leftrightarrow\ &\{p(\mathbf{x},\boldsymbol{\beta}_t)-p(\mathbf{x},\boldsymbol{\beta}_0)\}\{2p(\mathbf{x},\boldsymbol{\beta}_t)-1\}>0.
\end{aligned}$$

Thus the inequality holds if

$$\begin{aligned}
&p(\mathbf{x},\boldsymbol{\beta}_t)>0.5>p(\mathbf{x},\boldsymbol{\beta}_0)\quad\text{or}\quad p(\mathbf{x},\boldsymbol{\beta}_t)<0.5<p(\mathbf{x},\boldsymbol{\beta}_0)\\
\Leftrightarrow\ &\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_t>0>\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\quad\text{or}\quad\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_t<0<\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0\\
\Leftrightarrow\ &\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_t\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0<0.
\end{aligned}$$

This finishes the proof. ■

Proof of Theorem 21. Similarly to the proof of Theorem 6, $\sqrt{n}(\hat{\boldsymbol{\beta}}_p-\boldsymbol{\beta}_t)$ is the maximizer of

$$\frac{1}{\sqrt{n}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}_p(\boldsymbol{\beta}_t)+\frac{1}{2n}\sum_{i=1}^{N}\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0+\acute{\mathbf{s}}/\sqrt{n})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i)^2,$$

where $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$, and

$$\dot{\lambda}_p(\boldsymbol{\beta}_t)=\sum_{i=1}^{N}\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\{y_i-p(\mathbf{x}_i,\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i.$$

Similarly to the proof of Lemma 32, we first notice that given $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$, $\dot{\lambda}_p(\boldsymbol{\beta}_t)$ is a sum of N independent random vectors. We now exam the mean and variance of $\dot{\lambda}_p(\boldsymbol{\beta}_t)$. For the mean, we have,

$$\frac{1}{\sqrt{n}}\mathbb{E}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0,\hat{\Psi}_0\big\}=\frac{\sqrt{n}}{\sqrt{N}}\frac{\sum_{i=1}^{N}\boldsymbol{\eta}_i}{\hat{\Psi}_0\sqrt{N}}=O_P(\sqrt{n/N}),$$

where the last equality is due to (45).

For the variance,

$$\begin{aligned}\frac{1}{n}\mathbb{V}\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0,\hat{\Psi}_0\} =&\frac{\frac{1}{N}\sum_{i=1}^N|\psi_i(\hat{\boldsymbol{\beta}}_0)|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\psi_i^2(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\hat{\Psi}_0}\\ &-\frac{n}{N}\frac{\frac{1}{N}\sum_{i=1}^N\psi_i^2(\hat{\boldsymbol{\beta}}_0)\psi_i^2(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\hat{\Psi}_0^2}\\ \equiv&\Delta_8-\Delta_9\end{aligned}$$

From Lemma 28, if $n/N\to\rho$, using a similar approach used in the proof of Lemma 32, we have

$$\begin{aligned}\Delta_8=&\frac{1}{\hat{\Psi}_0^2}\frac{1}{N}\sum_{i=1}^N|\psi_i(\hat{\boldsymbol{\beta}}_0)|\Big\{\frac{n|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)}{N}\vee\hat{\Psi}_0\Big\}\psi_i^2(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\ =&\frac{1}{\Psi_0^2}\mathbb{E}\Big[|\psi(\boldsymbol{\beta}_0)|\{\rho|\psi(\boldsymbol{\beta}_0)|h(\mathbf{x})\vee\Psi(\boldsymbol{\beta}_0)\}\psi^2(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\Big]+o_P(1)\end{aligned}$$

and

$$\Delta_9=\rho\frac{\mathbb{E}\{\psi^2(\boldsymbol{\beta}_0)\psi^2(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)h^2(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\}}{\Psi_0^2}=\rho\frac{\boldsymbol{\varsigma}_b}{\Psi_0^2}+o_P(1).$$

Thus,

$$\begin{aligned}&\frac{1}{n}\mathbb{V}\{\dot{\lambda}_p(\boldsymbol{\beta}_t)|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0,\hat{\Psi}_0\}\\ &=\frac{\mathbb{E}\Big[|\psi(\boldsymbol{\beta}_0)|\{1-\Psi_0^{-1}\rho|\psi(\boldsymbol{\beta}_0)|h(\mathbf{x})\}_+\psi^2(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\Big]}{\Psi_0}+o_P(1)=\frac{\boldsymbol{\varsigma}_c}{\Psi_0}+o_P(1).\end{aligned}$$

Applying the Lindeberg-Feller central limit theorem, we have

$$\frac{\dot{\lambda}_p(\boldsymbol{\beta}_t)}{\sqrt{n}}-\frac{\sqrt{n}\sum_{i=1}^N\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)}\longrightarrow\mathbb{N}\Big(\mathbf{0},\ \frac{\boldsymbol{\varsigma}_c}{\Psi_0^2}\Big),\tag{52}$$

in conditional distribution.

Using a similar approach used to prove Lemma 33, we have

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^N\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_t-\hat{\boldsymbol{\beta}}_0+\mathbf{s}_n)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\ &=\sum_{i=1}^N\pi_i^p(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}+o_P(1)\\ &=\frac{\frac{1}{N}\sum_{i=1}^N|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_t-\boldsymbol{\beta}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\hat{\Psi}_0}+o_P(1)=\frac{\boldsymbol{\varsigma}_a}{\Psi_0}+o_P(1).\end{aligned}$$

From (45) and (52), $\dot{\lambda}_p(\boldsymbol{\beta}_t)/\sqrt{n}$ is stochastically bounded. In addition, $\boldsymbol{\varsigma}_a$ is finite and positive-definite. Thus, from the Basic Corollary in page 2 of Hjort and Pollard (2011), $\sqrt{n}(\hat{\boldsymbol{\beta}}_p-\boldsymbol{\beta}_t)$ satisfies

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p-\boldsymbol{\beta}_t)=\Psi_0\boldsymbol{\varsigma}_a^{-1}\frac{1}{\sqrt{n}}\dot{\lambda}_p(\boldsymbol{\beta}_t)+o_P(1),$$

given $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$. Combining this with (50), (52), Slutsky's theorem, and the fact that a conditional probability is bounded by one, Theorem 21 follows. ■

B.4.2. Proofs with model misspecification

Proof of Theorem 24. By similar arguments used in the proof of Theorem 1, we know that $\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw}-\boldsymbol{\beta}_l)$ is the maximizer of

$$\frac{1}{\sqrt{n}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_l)+\frac{1}{2n}\sum_{i=1}^n\phi_i^*(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0+\acute{\mathbf{s}}/\sqrt{n})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i^*)^2,$$

where $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$, and

$$\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_l)=\sum_{i=1}^n\{y_i^*-p(\mathbf{x}_i^*,\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i^*.$$

We abuse the notation and redefine $\boldsymbol{\eta}_i=|\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i$ in this proof. By similar arguments used in the proof of Lemma 30, we have that

$$\mathbb{E}\big[\{y^*-p(\mathbf{x}^*,\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^*\big|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0\big]=\sum_{i=1}^N\pi_i(\hat{\boldsymbol{\beta}}_0)\psi_i(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)\mathbf{x}_i=\frac{\sum_{i=1}^N\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)},$$

and

$$\begin{aligned}&\mathbb{V}\big[\{y^*-p(\mathbf{x}^*,\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)\}\mathbf{x}^*\big|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0\big]\\ &=\sum_{i=1}^N\pi_i(\hat{\boldsymbol{\beta}}_0)\{y_i-p(\mathbf{x}_i,\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)\}^2\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}-\Delta^2\\ &=\frac{\frac{1}{N}\sum_{i=1}^N|\psi_i(\boldsymbol{\beta}_l)|(y_i-0.5)^2h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\Psi_N(\boldsymbol{\beta}_l)}+o_P(1)=\frac{\boldsymbol{\kappa}_a}{\omega}+o_P(1).\end{aligned}$$

The Lindeberg-Feller condition under the conditional distribution can be verified using a similar approached used in the proof of Lemma 30. Thus, conditional on $\mathcal{D}_N$ and $\hat{\boldsymbol{\beta}}_0$, as n_0, n, and N go to infinity,

$$\frac{\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_t)}{\sqrt{n}}-\frac{\sqrt{n}\sum_{i=1}^N\boldsymbol{\eta}_i}{N\Psi_N(\hat{\boldsymbol{\beta}}_0)}\longrightarrow\mathbb{N}\Big(\mathbf{0},\ \frac{\boldsymbol{\kappa}_a}{\omega}\Big),\tag{53}$$

in conditional distribution. Now we exam $\boldsymbol{\eta}_i$. For the j-th element of $\boldsymbol{\eta}_i$,

$$\begin{aligned}\eta_{ij} =&|\psi_i(\hat{\boldsymbol{\beta}}_0)|\{y_i - p_i(\mathbf{x}_i, \boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0)\}h(\mathbf{x}_i)x_{ij}\\ =&(2y_i-1)\{y_i - p_i(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)\}\{y_i - p_i(\mathbf{x}_i, \boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0)\}h(\mathbf{x}_i)x_{ij}\\ =&0.5(2y_i-1)^2\{y_i - p_i(\mathbf{x}_i, \boldsymbol{\beta}_l)\}h(\mathbf{x}_i)x_{ij} + \acute{\eta}_{ij}h(\mathbf{x}_i)x_{ij}\mathbf{x}_i^{\mathrm{T}}(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_l),\end{aligned} \tag{54}$$

where

$$\begin{aligned}\acute{\eta}_{ij} =&(2y_i-1)\Big[\{y_i - p_i(\mathbf{x}_i, \acute{\boldsymbol{\beta}})\}p_i(\mathbf{x}_i, \acute{\mathbf{s}})\{1 - p_i(\mathbf{x}_i, \acute{\mathbf{s}})\}\\ &- p_i(\mathbf{x}_i, \acute{\boldsymbol{\beta}})\{1 - p_i(\mathbf{x}_i, \acute{\boldsymbol{\beta}})\}\{y_i - p_i(\mathbf{x}_i, \acute{\mathbf{s}})\}\Big],\end{aligned}$$

with $\acute{\mathbf{s}} = \boldsymbol{\beta}_l - \acute{\boldsymbol{\beta}}$, and $\acute{\boldsymbol{\beta}}$ being between $\boldsymbol{\beta}_l$ and $\hat{\boldsymbol{\beta}}_0$. Note that $|\dot{\eta}_{ij}| \le 2$ and

$$\acute{\eta}_{ij} \to \frac{(2y_i-1)}{4}\big[\{y_i - p_i(\mathbf{x}_i, \boldsymbol{\beta}_l)\} - 2(2y_i-1)p_i(\mathbf{x}_i, \boldsymbol{\beta}_l)\{1 - p_i(\mathbf{x}_i, \boldsymbol{\beta}_l)\}\big],$$

in probability. Thus, from Lemma 28 and direct calculations, we have that

$$\frac{1}{N}\sum_{i=1}^N h(\mathbf{x}_i)(\acute{\boldsymbol{\eta}}_i \circ \mathbf{x}_i)\mathbf{x}_i^{\mathrm{T}} = \boldsymbol{\kappa}_c + o_P(1), \tag{55}$$

where $\acute{\boldsymbol{\eta}}_i = (\acute{\eta}_{i1}, ..., \acute{\eta}_{id})^{\mathrm{T}}$ and $\circ$ is the Hadamard product. From (54) and (55), we have that

$$\sum_{i=1}^N \boldsymbol{\eta}_i = \frac{1}{2}\sum_{i=1}^N (2y_i-1)^2\{y_i - p_i(\mathbf{x}_i, \boldsymbol{\beta}_l)\}h(\mathbf{x}_i)\mathbf{x}_i + \boldsymbol{\kappa}_c N(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_l) + o_P\{N(\hat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}_l)\}.$$

Thus, from the central limit theorem and Slutsky's theorem, and the fact that $\hat{\boldsymbol{\beta}}_0$ is independent of $\mathcal{D}_N$,

$$\frac{\sum_{i=1}^N \boldsymbol{\eta}_i}{\sqrt{N}} \longrightarrow \mathbb{N}\Big(\mathbf{0},\ \boldsymbol{\kappa}_b + \frac{\boldsymbol{\kappa}_c\boldsymbol{\Sigma}_0\boldsymbol{\kappa}_c}{\rho_0}\Big). \tag{56}$$

From Lemma 28 and using a similar approach to prove Lemma 31, we have

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^n \phi_i^*(\boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})\mathbf{x}_i^*(\mathbf{x}_i^*)^{\mathrm{T}}\\ &= \sum_{i=1}^N \pi_i(\hat{\boldsymbol{\beta}}_0)\phi_i(\boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} + o_P(1)\\ &= \frac{\frac{1}{N}\sum_{i=1}^N |\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\frac{1}{N}\sum_{i=1}^N |y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)} + o_P(1) = \frac{\boldsymbol{\kappa}_a}{\omega} + o_P(1).\end{aligned}$$

Since $\boldsymbol{\kappa}_a$ is a positive definite matrix, and $\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_l)/\sqrt{n}$ is stochastically bounded due to (53) and (56), from the Basic Corollary in page 2 of Hjort and Pollard (2011), $\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_l)$ satisfies

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \boldsymbol{\beta}_l) = \omega\boldsymbol{\kappa}_a^{-1}\frac{1}{\sqrt{n}}\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_l) + o_P(1) \tag{57}$$

given $\mathcal{D}_N$ and $\hat{\boldsymbol{\beta}}_0$.

Using similar arguments used in the proof of Lemma 29, we know that $\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_l)$ is the maximizer of

$$\frac{1}{\sqrt{N}}\mathbf{s}^{\mathrm{T}}\sum_{i=1}^N \boldsymbol{\eta}_i - \frac{1}{2N}\sum_{i=1}^N |y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{N})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i)^2,$$

where $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$. From Lemma 28,

$$\frac{1}{N}\sum_{i=1}^N |y_i - p(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)\phi_i(\boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{N})\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \boldsymbol{\kappa}_a + o_P(1).$$

Thus, from (56) and the Basic Corollary in page 2 of Hjort and Pollard (2011), we know that $\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_l)$ satisfies

$$\sqrt{N}(\hat{\boldsymbol{\beta}}_{\mathrm{wMLE}} - \boldsymbol{\beta}_l) = \boldsymbol{\kappa}_a^{-1}\frac{1}{\sqrt{N}}\sum_{i=1}^N \boldsymbol{\eta}_i + o_P(1). \tag{58}$$

From (56), Slutsky's theorem, and the fact that a conditional probability is bounded by one, the result in (24) follows.

From (57) and (58), we have

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{uw} - \hat{\boldsymbol{\beta}}_{\mathrm{wMLE}}) = \omega\boldsymbol{\kappa}_a^{-1}\Big\{\frac{1}{\sqrt{n}}\dot{\lambda}_{uw}^*(\boldsymbol{\beta}_l) - \frac{\sqrt{n}\sum_{i=1}^N \boldsymbol{\eta}_i}{\omega N}\Big\} + o_P(1).$$

Thus, from (53), Slutsky's theorem, and the fact that a conditional probability is bounded by one, (23) of Theorem 24 follows. ■

Proof of Theorem 26. Using similar arguments used in the proof Theorem 6, we know that $\sqrt{n}(\hat{\boldsymbol{\beta}}_p - \boldsymbol{\beta}_l)$ is the maximizer of

$$\frac{1}{\sqrt{n}}\mathbf{s}^{\mathrm{T}}\dot{\lambda}_p(\boldsymbol{\beta}_l) + \frac{1}{2n}\sum_{i=1}^N \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\phi_i(\boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0 + \acute{\mathbf{s}}/\sqrt{n})(\mathbf{s}^{\mathrm{T}}\mathbf{x}_i)^2,$$

where $\acute{\mathbf{s}}$ lies between $\mathbf{0}$ and $\mathbf{s}$, and

$$\dot{\lambda}_p(\boldsymbol{\beta}_l) = \sum_{i=1}^N \delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0) \vee 1\}\{y_i - p(\mathbf{x}_i, \boldsymbol{\beta}_l - \hat{\boldsymbol{\beta}}_0)\}\mathbf{x}_i.$$

Given $\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0$ and $\hat{\Psi}_0$, $\dot{\lambda}_p(\boldsymbol{\beta}_l)$ is a sum of independent variables, and the Lindeberg-Feller condition under the condition distribution can be verified similarly to the proof of Lemma 32. Now we exam the conditional mean and variance of $\dot{\lambda}_p(\boldsymbol{\beta}_l)$. For the mean, from (56) we have,

$$\frac{1}{\sqrt{n}}\mathbb{E}\big\{\dot{\lambda}_p(\boldsymbol{\beta}_l)|\mathcal{D}_N, \hat{\boldsymbol{\beta}}_0, \hat{\Psi}_0\big\} = \frac{\sqrt{n}}{\sqrt{N}}\frac{\sum_{i=1}^N \boldsymbol{\eta}_i}{\hat{\Psi}_0\sqrt{N}} = O_P(\sqrt{n/N}).$$

For the variance,

$$\begin{aligned}
&\frac{1}{n}\mathbb{V}\{\dot{\lambda}_p(\boldsymbol{\beta}_l)|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0,\hat{\Psi}_0\}\\
=&\frac{\frac{1}{N}\sum_{i=1}^N|\psi_i(\hat{\boldsymbol{\beta}}_0)|\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\psi_i^2(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\hat{\Psi}_0}\\
&-\frac{n}{N}\frac{\frac{1}{N}\sum_{i=1}^N\psi_i^2(\hat{\boldsymbol{\beta}}_0)\psi_i^2(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}}{\hat{\Psi}_0^2}\equiv\Delta_{10}+\Delta_{11}. \qquad (59)
\end{aligned}$$

Note that $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^2\}<\infty$, $\mathbb{E}\{h^2(\mathbf{x})\|\mathbf{x}\|^2\}<\infty$, and $|\psi_i(\cdot)|$ are bounded. Thus, from Lemma 28, if $n/N\to\rho$,

$$\Delta_{11}\to\rho\frac{\boldsymbol{\kappa}_b}{\omega^2}, \qquad (60)$$

in probability. For the term Δ_{10} in (59), since $\mathbb{E}\{h(\mathbf{x})\|\mathbf{x}\|^2\}<\infty$, $\mathbb{E}\{h^2(\mathbf{x})\|\mathbf{x}\|^2\}<\infty$, and $|\psi_i(\cdot)|$ are bounded, from Lemma 28, if $n/N\to\rho$, as n_0, n, and N go to infinity, by a similar approach used in the proof of Lemma 32, we have

$$\begin{aligned}
\Delta_{10}=&\frac{1}{\hat{\Psi}_0^2}\frac{n}{N^2}\sum_{i=1}^N\psi_i^2(\hat{\boldsymbol{\beta}}_0)\psi_i^2(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)h^2(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}I\Big\{\frac{n|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)}{N}>\hat{\Psi}_0\Big\}\\
&+\frac{1}{\hat{\Psi}_0}\frac{1}{N}\sum_{i=1}^N|\psi_i(\hat{\boldsymbol{\beta}}_0)|\psi_i^2(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0)h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}I\Big\{\frac{n|\psi_i(\hat{\boldsymbol{\beta}}_0)|h(\mathbf{x}_i)}{N}\le\hat{\Psi}_0\Big\}\\
=&\frac{\mathbb{E}\Big(|\psi(\boldsymbol{\beta}_l)|[\{\rho|\psi(\boldsymbol{\beta}_l)|h(\mathbf{x})\}\vee\omega]h(\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}\Big)}{4\omega^2}+o_P(1). \qquad (61)
\end{aligned}$$

From, (59), (60), and (61), if $n/N\to\rho$,

$$\frac{1}{n}\mathbb{V}\{\dot{\lambda}_p(\boldsymbol{\beta}_l)|\mathcal{D}_N,\hat{\boldsymbol{\beta}}_0,\hat{\Psi}_0\}=\frac{\boldsymbol{\kappa}_d}{\omega}+o_P(1).$$

From the above results, conditional on $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$, we know that

$$\frac{\dot{\lambda}_p(\boldsymbol{\beta}_l)}{\sqrt{n}}-\frac{\sqrt{n}\sum_{i=1}^N\boldsymbol{\eta}_i}{N\hat{\Psi}_0}\longrightarrow\mathbb{N}\Big(\mathbf{0},\ \frac{\boldsymbol{\kappa}_d}{\omega^2}\Big), \qquad (62)$$

in distribution.

In addition, from Lemma 28, using an approach similar to the proof of Lemma 33, we have

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^N\delta_i^{\hat{\boldsymbol{\beta}}_0}\{n\pi_i^p(\hat{\boldsymbol{\beta}}_0)\vee 1\}\phi_i(\boldsymbol{\beta}_l-\hat{\boldsymbol{\beta}}_0+\acute{\mathbf{s}}/\sqrt{n})\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\\
&=\frac{1}{4n}\sum_{i=1}^N\delta_i^{\boldsymbol{\beta}_l}\{n\pi_i^p(\boldsymbol{\beta}_l)\vee 1\}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}+o_P(1)\\
&=\frac{1}{4N\hat{\Psi}_0}\sum_{i=1}^N|\psi_i(\boldsymbol{\beta}_l)|h(\mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}+o_P(1)=\frac{\boldsymbol{\kappa}_a}{\omega}+o_P(1). \qquad (63)
\end{aligned}$$

Thus, based on (56), (62), and (63), from the Basic Corollary in page 2 of Hjort and Pollard (2011), $\sqrt{n}(\hat{\boldsymbol{\beta}}_p-\boldsymbol{\beta}_l)$, satisfies

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_p-\boldsymbol{\beta}_l)=\omega\boldsymbol{\kappa}_a\frac{1}{\sqrt{n}}\dot{\lambda}_p(\boldsymbol{\beta}_l)+o_P(1),$$

given $\mathcal{D}_N$, $\hat{\boldsymbol{\beta}}_0$, and $\hat{\Psi}_0$. Combining this with (58), (62), Slutsky's theorem, and the fact that a conditional probability is bounded by one, Theorem 26 follows. ■

References

Mingyao Ai, Jun Yu, Huiming Zhang, and HaiYing Wang. Optimal subsampling algorithms for big data generalized linear models. *Statistica Sinica*, 2019. doi: 10.5705/ss.202018.0439.

Anthony Atkinson, Alexander Donev, and Randall Tobias. *Optimum experimental designs, with SAS*, volume 34. Oxford University Press, 2007.

H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, 32:1217–1236, 2010.

P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5(4308): http://dx.doi.org/10.1038/ncomms5308, 2014.

Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi: 10.1137/141000671.

JR Blum, H Chernoff, M Rosenblatt, and H Teicher. Central limit theorems for interchangeable processes. *Canad. J. Math*, 10:222–229, 1958.

Guang Cheng and Jianhua Huang. Bootstrap consistency for general semiparametric m-estimation. *The Annals of Statistics*, 38(5):2884–2915, 2010.

Yuan Shih Chow and Henry Teicher. *Probability Theory: Independence, Interchangeability, Martingales*. Springer, New York, 2003.

Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 81–90. ACM, 2013.

P. Drineas, M.W. Mahoney, S. Muthukrishnan, and T. Sarlos. Faster least squares approximation. *Numerische Mathematik*, 117:219–249, 2011.

P. Drineas, M. Magdon-Ismail, M.W. Mahoney, and D.P. Woodruff. Faster approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13: 3475–3506, 2012.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1): 132–157, 2006a.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006b.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006c.

Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Sampling algorithms for l_2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1127–1136. Society for Industrial and Applied Mathematics, 2006d.

Dheeru Dua and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

William Fithian and Trevor Hastie. Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics*, 42(5):1693, 2014.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Lei Han, Kean Ming Tan, Ting Yang, and Tong Zhang. Local uncertainty sampling for large-scale multi-class logistic regression. *Annals of Statistics*, 2019. URL `https://www.e-publications.org/ims/submission/AOS/user/submissionFile/37810?confirm=045881a9`.

Nils Lid Hjort and David Pollard. Asymptotics for minimisers of convex processes. *arXiv preprint arXiv:1107.3806*, 2011.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall Professional Technical Reference, 2nd edition, 1988. ISBN 0131103709.

Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):795–816, 2014.

Nan Lin and Ruibin Xie. Aggregated estimating equation estimation. *Statistics and Its Interface*, 4:73–83, 2011.

P. Ma, M.W. Mahoney, and B. Yu. A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research*, 16:861–911, 2015.

Michael W Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

Michael W Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

Peter McCullagh and James A Nelder. *Generalized Linear Models, no. 37 in Monograph on Statistics and Applied Probability*. Chapman & Hall,, 1989.

Brian McWilliams, Gabriel Krummenacher, Mario Lucic, and Joachim M Buhmann. Fast and robust least squares estimation in corrupted linear models. In *Advances in Neural Information Processing Systems*, pages 415–423, 2014.

X. Meng, M.A. Saunders, and M.W. Mahoney. LSRN: A parallel iterative solver for strongly over- or under- determined systems. *SIAM Journal on Scientific Computing*, 36:C95–C118, 2014.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

Garvesh Raskutti and Michael Mahoney. A statistical perspective on randomized sketching for ordinary least-squares. *Journal of Machine Learning Research*, 17:1–31, 2016.

Elizabeth D. Schifano, Jing Wu, Chun Wang, Jun Yan, and Ming-Hui Chen. Online updating of statistical inference in the big data setting. *Technometrics*, 58(3):393–403, 2016.

R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, New York, 1980.

Bjarne Stroustrup. *The C++ programming language*. Pearson Education India, 1986.

A.W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, London, 1998.

G. van Rossum. Python tutorial, technical report cs-r9526. Technical report, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1995.

HaiYing Wang. Divide-and-conquer information-based optimal subdata selection algorithm. *Journal of Statistical Theory and Practice*, 2019. doi: 10.1007/s42519-019-0048-5.

HaiYing Wang, Rong Zhu, and Ping Ma. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844, 2018.

HaiYing Wang, Min Yang, and John Stufken. Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association*, 114(525): 393–405, 2019.

Shifeng Xiong and Guoying Li. Some results on the convergence of conditional distributions. *Statistics & Probability Letters*, 78(18):3249–3253, 2008.

Tianbao Yang, Lijun Zhang, Rong Jin, and Shenghuo Zhu. An explicit sampling dependent spectral error bound for column subset selection. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 135–143, 2015.

Yun Yang, Mert Pilanci, Martin J Wainwright, et al. Randomized sketches for kernels: Fast and optimal nonparametric regression. *The Annals of Statistics*, 45(3):991–1023, 2017.

Yaqiong Yao and HaiYing Wang. Optimal subsampling for softmax regression. *Statistical Papers*, 60(2):235–249, 2019.

Journal of Machine Learning Research 20 (2019) 1-34 Submitted 9/18; Revised 7/19; Published 8/19

Spurious Valleys in One-hidden-layer Neural Network Optimization Landscapes

Luca Venturi VENTURI@CIMS.NYU.EDU
Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012

Afonso S. Bandeira BANDEIRA@CIMS.NYU.EDU
Joan Bruna BRUNA@CIMS.NYU.EDU
Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012
Center for Data Science, 60 5th Avenue, New York, NY 10011

Editor: Animashree Anandkumar

Abstract

Neural networks provide a rich class of high-dimensional, non-convex optimization problems. Despite their non-convexity, gradient-descent methods often successfully optimize these models. This has motivated a recent spur in research attempting to characterize properties of their loss surface that may explain such success.

In this paper, we address this phenomenon by studying a key topological property of the loss: the presence or absence of *spurious valleys*, defined as connected components of sub-level sets that do not include a global minimum. Focusing on a class of one-hidden-layer neural networks defined by smooth (but generally non-linear) activation functions, we identify a notion of intrinsic dimension and show that it provides necessary and sufficient conditions for the absence of spurious valleys. More concretely, finite intrinsic dimension guarantees that for sufficiently overparametrised models no spurious valleys exist, independently of the data distribution. Conversely, infinite intrinsic dimension implies that spurious valleys do exist for certain data distributions, independently of model overparametrisation. Besides these positive and negative results, we show that, although spurious valleys may exist in general, they are confined to low risk levels and avoided with high probability on overparametrised models.

1. Introduction

Modern machine learning applications involve datasets of increasing dimensionality, complexity and size, which in turn motivate the use of high-dimensional, non-linear models, as illustrated in many deep learning algorithms across computer vision, speech and natural language understanding. The prevalent strategy for learning is to rely on Stochastic Gradient Descent (SGD) methods, that typically operate on non-convex objectives. In this context, an outstanding goal is to provide a theoretical framework that explains under what conditions – relating input data distribution, choice of architecture and choice of optimization scheme – this setup will be successful.

More precisely, let $\mathbf{\Phi}_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{R}^m$ denote a model class parametrized by $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^P$, which in the case of Neural Networks (NNs) contains the aggregated weights across all layers. In a supervised learning setting, this model is deployed on some data $(\mathbf{X}, \mathbf{Y})$ random variable

©2019 Luca Venturi, Afonso S. Bandeira and Joan Bruna.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-674.html.

taking values in $\mathbb{R}^n \times \mathbb{R}^m$, to predict targets $\mathbf{Y}$ given input $\mathbf{X}$, and its risk for a given $\boldsymbol{\theta}$ is

$$L(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim P}[\ell(\mathbf{\Phi}_{\boldsymbol{\theta}}(\mathbf{X}), \mathbf{Y})] \tag{1}$$

where ℓ is a convex loss, such as a square loss or a logistic regression loss. In the following we refer to (1) as the risk, the energy or the loss interchangeably. The aim is to find $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{\theta})$ and this is attempted in practice by running SGD iteration. Under some technical conditions, the expected gradient is known to converge to zero (Bottou et al., 2016). Understanding the nature of such stationary points - and therefore the landscape of the loss function - is a task of fundamental importance to understand performance of SGD.

Whereas there is a growing literature in analyzing the behavior of SGD on non-convex objectives (Soudry et al., 2017; Ji and Telgarsky, 2018; Gunasekar et al., 2018; Wilson et al., 2017), we focus here on properties of the optimization problem above that are algorithm independent. A common factor shared in the above cited works (and in common practice) is that *overparametrisation* of the model class (i.e. $P \gg 1$) often leads to improved performance, despite the potential increase in generalization error.

Our analysis focuses mostly on the class of one-hidden-layer neural networks, with a hidden layer of size p, and covers both empirical and population risk landscapes. More specifically, we look at presence (or absence) of *spurious valleys*, defined as connected components of the sub-level sets that do not contain a global minima. We define two quantities depending on the functional space spanned by neural networks of different widths: the upper intrinsic dimension, defined as the dimension of this linear space, and the lower intrinsic dimension, defined as the minimum number of hidden units to describe any element of the functional space. Upper and lower intrinsic dimensions define only two scenarios: either (i) they are both finite, enabling positive results; or (ii) they are both infinite, implying the negative results.

Summary of contributions More specifically, we show that:

- For Empirical Risk Minimization or polynomial activations, spurious valleys do not occur as long as the network is sufficiently over-parametrised. For the case of linear and quadratic activations, our results are (up to a constant factor) tight.
- For non-polynomial non-negative activations, for any hidden width, we construct data distributions which yield spurious valleys with positive measure, whose value is arbitrarily far from the one of the global.
- Finally, drawing on connections with random features expansions, we show that, even if spurious valleys may appear in general, their measure decreases as the width increases. This holds up to a low energy threshold, which approaches the global minimum at a rate inversely proportional to the hidden layer size (up to log factors).

Related works A considerable amount of literature has attempted to characterize the landscape of the loss function (1) by studying its critical points. Global optimality results have been obtained for NN architectures with linear activations (Hardt and Ma, 2016; Kawaguchi, 2016; Yun et al., 2018), quadratic activations (Soltanolkotabi et al., 2017; Du and Lee, 2018) and some more general non-linear activations, under appropriate regularity assumptions (Soudry and Carmon, 2016; Nguyen and Hein, 2017; Feizi et al., 2017). Some other insights have been obtained by leveraging tools for complexity analysis of spin glasses

(Choromanska et al., 2015) and random matrix theory (Pennington and Bahri, 2017). Other analysis involved studying goodness of the initialization of the parameter values $\boldsymbol{\theta}_0$ (Daniely et al., 2016; Safran and Shamir, 2016; Du et al., 2017) or other topological properties of the loss (1), such as connectivity of sub-level sets (Draxler et al., 2018; Freeman and Bruna, 2017).

Several other type of analysis of the convergence of NNs gradient-based optimization algorithms have been considered in the literature. For example, (Ge et al., 2017b) proved convergence of GD on a modified loss; (Shamir, 2018) compared optimization properties of residual networks with respect to linear models; in (Dauphin et al., 2014) it is argued that the issues arising in the optimization of NN architectures are due to the presence of saddle points in the loss function rather than spurious local minima. Optimization landscapes have also been studied in other contexts than from NNs training, such as non-convex low rank problems (Ge et al., 2017a), matrix completion (Ge et al., 2016), problems arising in semidefinite programming (Boumal et al., 2016; Bandeira et al., 2016) and implicit generative modeling (Bottou et al., 2017).

Structure of the paper The rest of the paper is structured as follows. Section 2 formally introduces the notion of spurious valleys and explains why this is a relevant concept from the optimization point of view. It also defines the intrinsic dimensions of a network (Section 2.2). In Section 3 we state our main positive results (Theorem 8) and we discuss two settings where they bear fruit: polynomial activation functions and empirical risk minimization. Section 4 is dedicated to constructions of worst case scenarios for activation with infinite lower intrinsic dimension. We then show, in Section 5, that, even if spurious valleys may exist, they tend to be confined to regimes of low risk. Some conclusive discussion is reported in Section 6.

1.1. Notation

We introduce notation we use throughout the rest of the paper. For any integers $n \leq m$ we denote $[n, m] = \{n, n+1, \dots, m\}$ and, if $n > 0$, $[n] = [1, n]$. We denote scalar valued variables as lowercase non-bold; vector valued variables as lowercase bold; matrix and tensor valued variables and multivariate random variables (r.v.'s) as uppercase bold. Given a vector $\mathbf{v} \in \mathbb{R}^n$, we denote its components as v_i; given a matrix $\mathbf{W} \in \mathbb{R}^{n\times m}$, we denote its rows as $\mathbf{w}_i$; given a tensor $\mathbf{T} \in \mathbb{R}^{n_1\times\cdots\times n_k}$, we denote its components as $T_{i_1\cdots i_k}$. Given some vectors $\mathbf{v}_i \in \mathbb{R}^{n_i}$, $i \in [k]$, the tensor product $\mathbf{v}_1 \otimes \cdots \otimes \mathbf{v}_k$ denotes the $n_1 \times \cdots \times n_k$ dimensional tensor $\mathbf{T}$ whose components are given by $T_{i_1\cdots i_k} = v_{i_1} \cdots v_{i_k}$; given a vector $\mathbf{v}$, we denote $\mathbf{v}^{\otimes k} = \otimes_{i=1}^k \mathbf{v}$. We denote by $\mathrm{S}^k(\mathbb{R}^n)$ the space of order k symmetric tensors on $\mathbb{R}^n$. For any $\mathbf{T} \in \mathrm{S}^k(\mathbb{R}^n)$, we define the symmetric rank (Comon et al., 2008) as $\mathrm{rk}_{\mathrm{S}}(\mathbf{T}) = \min\Big\{p \geq 1 : \mathbf{T} = \sum_{i=1}^p u_i \mathbf{w}_i^{\otimes k} \text{ for some } \mathbf{u} \in \mathbb{R}^p, \mathbf{w}_1, \dots, \mathbf{w}_p \in \mathbb{R}^n\Big\}$. We define $\mathrm{rk}_{\mathrm{S}}(k, n) = \max\{\mathrm{rk}_{\mathrm{S}}(\mathbf{T}) : \mathbf{T} \in \mathrm{S}^k(\mathbb{R}^n)\}$. Finally, $\mathbb{S}^{n-1} \subset \mathbb{R}^n$ denotes the $(n-1)$-dimensional sphere $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1\}$.

2. Preliminaries

2.1. Problem setup

Let $(\mathbf{X}, \mathbf{Y})$ be two r.v.'s. These r.v.'s take values in $\mathbb{R}^n$ and $\mathbb{R}^m$ and represent the *input* and *output* data, respectively. We consider oracle square loss functions $L : \Theta \to \mathbb{R}$ of the form

$$L(\boldsymbol{\theta}) \doteq \mathbb{E}[\ell(\boldsymbol{\Phi}(\mathbf{X}; \boldsymbol{\theta}), \mathbf{Y})] \tag{2}$$

where $\ell : \mathbb{R}^m \times \mathbb{R}^m \to [0, \infty)$ is convex in its first argument. For every $\boldsymbol{\theta} \in \Theta$, the function $\boldsymbol{\Phi}(\cdot; \boldsymbol{\theta}) : \mathbb{R}^n \to \mathbb{R}^m$ models the dependence of the output on the input as $\mathbf{Y} \simeq \boldsymbol{\Phi}(\mathbf{X}; \boldsymbol{\theta})$. We focus on one-hidden-layer NN functions $\boldsymbol{\Phi}$, i.e. $\boldsymbol{\Phi}$ of the form

$$\boldsymbol{\Phi}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x}) \tag{3}$$

where $\boldsymbol{\theta} = (\mathbf{U}, \mathbf{W}) \in \Theta \doteq \mathbb{R}^{m\times p} \times \mathbb{R}^{p\times n}$. Here p represents the width of the hidden layer and $\sigma : \mathbb{R} \to \mathbb{R}$ is a continuous element-wise *activation* function.

The loss function $L(\boldsymbol{\theta})$ is (in general) a non-convex object; it may present spurious (i.e. non global) local minima. In this work, we characterize $L(\boldsymbol{\theta})$ by determining absence or presence of spurious valleys, as defined below.

Definition 1 *For all $c \in \mathbb{R}$ we define the sub-level set of L as $\Omega_L(c) = \{\boldsymbol{\theta} \in \Theta : L(\boldsymbol{\theta}) \leq c\}$. We define a* spurious valley *as a path-connected component of a sub-level set $\Omega_L(c)$ which does not contain a global minimum of the loss $L(\boldsymbol{\theta})$.*

Since, in practice, the loss (2) is minimized with a gradient descent based algorithm, then absence of spurious valleys is a desirable property, if we wish the algorithm to converge to an optimal parameter. It is easy to see that $L(\boldsymbol{\theta})$ not having spurious valleys is implied by the following property:

P.1 Given any *initial* parameter $\tilde{\boldsymbol{\theta}} \in \Theta$, there exists a continuous path $\boldsymbol{\theta} : t \in [0, 1] \mapsto \boldsymbol{\theta}_t \in \Theta$ such that:

(a) $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}$

(b) $\boldsymbol{\theta}_1 \in \arg\min_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{\theta})$

(c) The function $t \in [0, 1] \mapsto L(\boldsymbol{\theta}_t)$ is non-increasing

As pointed out in (Freeman and Bruna, 2017), this implies that L has no strict spurious (i.e. non global) local minima. The absence of generic (i.e. non-strict) spurious local minima is guaranteed if the path $\boldsymbol{\theta}_t$ is such that the function $L(\boldsymbol{\theta}_t)$ is strictly decreasing. For sake of clarity, we review these properties in the following lemma (the proof is reported in the Appendix E).

Lemma 2 *Be $\boldsymbol{\theta} \mapsto L(\boldsymbol{\theta})$ a continuous function. Then, property **P.1** implies absence of spurious valleys. In particular, this implies absence of strict spurious minima, and of (generally non-strict) spurious minima if property **P.1** holds with strictly decreasing paths $t \mapsto L(\boldsymbol{\theta}_t)$. Conversely, presence of spurious valleys implies existence of spurious minima.*

In the following, we prove absence of spurious valleys by proving that property **P.1** holds. Intuitively, we should think about spurious valleys as regions of the parameter space from which it is impossible to 'escape' without 'up-climbing' the loss value.

Notice that for many activation functions used in practice (such as the ReLU $\sigma(z) = z_+$), the parameter $\boldsymbol{\theta}$ determining the function $\boldsymbol{\Phi}(\cdot;\boldsymbol{\theta})$ is determined up to the action of a symmetry group (e.g., in the case of the ReLU, σ is a positive homogeneous function). This already prevents strict minima: for any value of the parameter $\boldsymbol{\theta} \in \Theta$ there exists a (often large) *manifold* $\mathcal{U}_{\boldsymbol{\theta}} \subset \Theta$ intersecting $\boldsymbol{\theta}$ along which the loss function is constant.

ERM vs population loss In the following, we consider the loss (2) defined for a generic distribution $(\mathbf{X}, \mathbf{Y})$. In case of a distribution with a finite number of atoms, this corresponds to empirical risk minimization (ERM), which is (usually) the regime where machine learning algorithms perform optimization. On the other hand, for a generic data distribution, this loss is what is called *population* loss, and corresponds to the actual objective that machine learning algorithms aim to minimize. In our work we are interested in analyzing not only the ERM case, but more general population losses. While we in fact focus on highly over-parametrised neural networks, we aim to provide results which apply to the regime where number of data points goes to infinity before the number of parameters.

2.2. Intrinsic dimension of a network

The main result of this work is to exploit that the property of absence of spurious valleys is related to the complexity of the functional space $V_\sigma = \{f = \Phi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ defined by the network architecture. We therefore define two measures of such complexity which we will use to show, respectively, positive and negative results in this regard.

To simplify the discussion, we introduce some notation which we will use throughout the rest of the paper. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a continuous activation function. For every $\mathbf{v} \in \mathbb{R}^n$ we denote $\psi_{\sigma,\mathbf{v}}$ to be the function $\psi_{\sigma,\mathbf{v}} : \mathbf{x} \in \mathbb{R}^n \mapsto \sigma(\langle \mathbf{v}, \mathbf{x}\rangle) \in \mathbb{R}$. We refer to each $\psi_{\sigma,\mathbf{v}}$ as a *filter* function. If $\mathbf{X}$ is a r.v. taking values in $\mathbb{R}^n$, we denote by $L^2_{\mathbf{X}}$ the space of square integrable function on $\mathbb{R}^n$ w.r.t. the probability measure induced by the r.v. $\mathbf{X}$. We then define the two following functional spaces:

$$V_{\sigma,p} = \left\{ f = \Phi(\cdot;\boldsymbol{\theta}) : \boldsymbol{\theta} = (\mathbf{u}^T, \mathbf{W}) \in \Theta = \mathbb{R}^p \times \mathbb{R}^{p\times n} \right\}$$
$$\mathcal{R}_2(\sigma, n) = \left\{ \mathbf{X} \text{ r.v. taking values in } \mathbb{R}^n : \psi_{\sigma,\mathbf{v}} \in L^2_{\mathbf{X}} \text{ for every } \mathbf{v} \in \mathbb{R}^n \right\}$$

$V_{\sigma,p}$ represents the space of (one-dimensional output) functions modeled by the network architecture and $\mathcal{R}_2(\sigma, n)$ to be the space of (n-dimensional) input data distributions for which the filter functions have finite second moment. We finally define

$$V_\sigma = \operatorname{span}(\{f : f \in V_{\sigma,1}\}) = \bigcup_{p=1}^{\infty} V_{\sigma,p}$$

as the linear space spanned by the functions $\psi_{\mathbf{v},\sigma}$ for $\mathbf{v} \in \mathbb{R}^n$.

Definition 3 *Let σ be a continuous activation function and $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$ a r.v. We define*[1]

$$\dim^*(\sigma, \mathbf{X}) = \dim_{L^2_{\mathbf{X}}}(V_\sigma)$$

1. For any linear subspace $V \subseteq L^2_{\mathbf{X}}$, $\dim_{L^2_{\mathbf{X}}}(V)$ denotes the dimension of V as a subspace of $L^2_{\mathbf{X}}$.

as the upper intrinsic dimension of the pair $(\sigma, \mathbf{X})$. We define the level n upper intrinsic dimension of σ as $\dim^*(\sigma, n) = \dim(V_\sigma) = \sup\{\dim^*(\sigma, \mathbf{X}) : \mathbf{X} \in \mathcal{R}_2(\sigma, n)\}$.

The upper intrinsic dimension $\dim^*(\sigma, \mathbf{X})$ defined above is therefore the dimension of the functional space spanned by the filter functions $\psi_{\sigma,\mathbf{v}} \in L^2_{\mathbf{X}}$ or, equivalently, of the image of the map $\Phi : \boldsymbol{\theta} \in \Theta \mapsto \Phi(\cdot;\boldsymbol{\theta}) \in L^2_{\mathbf{X}}$. Notice that $\dim^*(\sigma, \mathbf{X}) \leq \dim(L^2_{\mathbf{X}})$. In particular, if the distribution $\mathbf{X}$ is discrete, i.e. it is concentrated on a finite number of points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^n$, then $\dim^*(\sigma, \mathbf{X}) \leq \dim(L^2_{\mathbf{X}}) \leq N$. Otherwise, if the distribution $\mathbf{X}$ is not discrete, then $\dim(L^2_{\mathbf{X}}) = \infty$.

The n level upper intrinsic dimension $\dim^*(\sigma, n)$ is defined as the dimension of the functional linear space V_σ. We note that if $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$ is a r.v. with almost surely (a.s.) positive density w.r.t. the Lebesgue measure dx, then $\dim^*(\sigma, n) = \dim^*(\sigma, \mathbf{X})$.

The following lemma exhausts all the cases when the upper intrinsic dimension is not infinite.

Lemma 4 *Let σ be a continuous activation function and $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$ such that $\dim\big(L^2_{\mathbf{X}}\big) = \infty$. If $\sigma(z) = \sum_{k=0}^{d} a_k z^k$ is a polynomial, then*

$$\dim^*(\sigma, \mathbf{X}) \leq \sum_{i=1}^{d} \binom{n+i-1}{i} \mathbf{1}_{\{a_i \neq 0\}} = O(n^d)$$

Otherwise (i.e. if σ is not a polynomial) it holds $\dim^*(\sigma, \mathbf{X}) = \infty$.

The proof of the above lemma is based on the universal approximation theorem (Leshno et al., 1993). We then define the lower intrinsic dimension, which corresponds to the concept of 'how many hidden neurons are needed to represent a generic function of V_σ'.

Definition 5 *Let σ be a continuous activation function and $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$ a r.v. We define*[2]

$$\dim_*(\sigma, \mathbf{X}) = \max\Big\{ p \geq 1 \ : \ V_{\sigma,p-1} \subsetneq_{L^2_{\mathbf{X}}} V_{\sigma,p} \Big\}$$

as the lower dimension of the pair $(\sigma, \mathbf{X})$. We define the level n lower dimension of σ as $\dim_*(\sigma, n) = \max\{p \geq 1 \ : \ V_{\sigma,p-1} \subsetneq V_{\sigma,p}\} = \sup\{\dim_*(\sigma, \mathbf{X}) : \mathbf{X} \in \mathcal{R}_2(\sigma, n)\}$.

If $\dim_*(\sigma, \mathbf{X})$ is finite, then it corresponds to the minimum number of hidden neurons which are needed to represent any function of V_σ with the NN architecture (3). Clearly, this implies that

$$\dim_*(\sigma, \mathbf{X}) \leq \dim^*(\sigma, \mathbf{X})$$

for every continuous activation function σ and any $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$. As with the upper instrinsic dimension, we note that if $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$ is a r.v. with a.s. positive density w.r.t. the Lebesgue measure dx, then $\dim_*(\sigma, n) = \dim_*(\sigma, \mathbf{X})$.

In the case of homogeneus polynomial activations $\sigma(z) = z^k$ with $k \geq 1$ integer, the level n lower dimension of σ coincides with the notion of (maximal) symmetric tensor rank.

2. For any subsets $V, W \subseteq L^2_{\mathbf{X}}$, we say that $V \subsetneq_{L^2_{\mathbf{X}}} W$ if $V \subsetneq W$ as subsets of $L^2_{\mathbf{X}}$ (and similar with other inclusions or equalities).

Lemma 6 *Let $\sigma(z) = z^k$, with k positive integer. Then*

$$\dim_*(\sigma, n) = \mathrm{rk}_{\mathrm{S}}(k, n)$$

Finally, the next lemma implies that for most non-polynomial activation functions practical interest, the lower intrinsic dimension $\dim_*(\sigma, n)$ is infinite.

Lemma 7 *Let σ be a continuous activation function such that $\sigma \in L^2(\mathbb{R}, e^{-x^2/2}\,dx)$ and $n > 1$. Then $\dim_*(\sigma, n) = \infty$ if and only if σ is not a polynomial.*

The proof of the above Lemma is based on Hermite decomposition and on the correspondence between one-hidden-layer nets and symmetric tensors (Mondelli and Montanari, 2018).

3. Finite intrinsic dimension and absence of spurious valleys

In this section we provide our positive results. Essentially they state that if the width of the network matches the dimension of the functional space V_σ spanned by its filter functions, then no spurious valleys exist. We first provide the main result (Theorem 8) in a general form, which allows a straight-forward derivation of two cases of interest: empirical risk minimization (Corollary 9) and polynomial activations (Corollary 10).

Theorem 8 *For any continuous activation function σ and r.v. $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$ with finite upper intrinsic dimension $\dim^*(\sigma, \mathbf{X}) < \infty$, the loss function*

$$L(\boldsymbol{\theta}) = \mathbb{E}[\ell(\boldsymbol{\Phi}(\mathbf{X}; \boldsymbol{\theta}), \mathbf{Y})]$$

for one-hidden-layer NNs $\boldsymbol{\Phi}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x})$ admits no spurious valleys in the over-parametrised regime $p \geq \dim^*(\sigma, \mathbf{X})$.

Sketch of the proof The proof consists of showing that we can construct a descent path verifying property **P.1** starting from any parameters $\boldsymbol{\theta}$. The construction can be articulated in two main parts. First, we show that we can map the starting parameter $\boldsymbol{\theta}_0 = (\mathbf{U}_0, \mathbf{W}_0)$ to another parameter $\boldsymbol{\theta}_{1/2} = (\mathbf{U}_{1/2}, \mathbf{W}_{1/2})$ such that the functions $\left\{\mathbf{x} \mapsto \sigma(\langle \mathbf{w}_{1/2,i}, \mathbf{x}\rangle)\right\}_{i\in[p]}$ form a basis of V_σ. It follows that there exists a minimal function $\mathbf{f} \in V_\sigma^m \doteq \{(f_1, \dots, f_m) : f_i \in V_\sigma\}$, i.e.

$$\mathbf{f} \in \arg\min_{\mathbf{g}\in V_\sigma^m} \mathbb{E}[\ell(\mathbf{g}(\mathbf{X}), \mathbf{Y})]$$

which can be represented as $\mathbf{f} = \boldsymbol{\Phi}(\cdot; \boldsymbol{\theta}_1 = (\mathbf{U}_1, \mathbf{W}_{1/2}))$ for some $\mathbf{U}_1$. The second part of the path can be thus taken as $t \mapsto (1-t)\mathbf{U}_{1/2} + t\mathbf{U}_1$: as the loss function is convex, this is descent path.

The above result can be interpreted as follows: if the network is such that any of its output units Φ_i can be chosen from the whole linear space spanned by its filter functions V_σ, then the associated optimization problem is such that there always exists a descent path to an optimal solution, for any initialization of the parameters.

Applying the observations in Section 2.2 describing the cases of finite intrinsic dimension, we immediately get the following corollaries.

Corollary 9 (ERM) *Consider N data points $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \subset \mathbb{R}^n \times \mathbb{R}^m$. For one-hidden-layer NNs $\boldsymbol{\Phi}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x})$, where σ is any continuous activation function, the empirical loss function*

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N \ell(\boldsymbol{\Phi}(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i)$$

admits no spurious valleys in the over-parametrized regime $p \geq N$.

Comparison with existing results This results was already shown in (Livni et al., 2014). The only difference with our result is that we allow for rank degeneracy in the matrix $\sigma(\mathbf{W}[\mathbf{x}_1|\cdots|\mathbf{x}_N])$. However, its proof illustrates the danger of studying empirical risk minimization landscapes in over-parametrised regimes, since it bypasses all the geometric and algebraic properties needed in the population risk setting - which may be more relevant to understand the generalization properties of the model.

Other works considered the landscape of empirical risk minimization for deep networks. For ReLu-like activations, multi-layer networks and square losses, (Soudry and Carmon, 2016) showed that (almost surely) there exists no differentiable spurious minima if one of the layer weights $\mathbf{W}_i \in \mathbb{R}^{p_i \times p_{i-1}}$ satisfy $p_i p_{i-1} \geq N$. (Nguyen and Hein, 2017) showed that no spurious minima occur for multi-layer NNs for a class of losses and activations, if one of the layers inner width exceeds the number of data points and the critical points verify certain non-degeneracy conditions.

Corollary 10 (Polynomial activations) *For one-hidden-layer NNs $\boldsymbol{\Phi}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x})$ with polynomial activation function $\sigma(z) = a_0 + a_1 z + \cdots + a_d z^d$, the loss function $L(\boldsymbol{\theta}) = \mathbb{E}[\ell(\boldsymbol{\Phi}(\mathbf{X}; \boldsymbol{\theta}), \mathbf{Y})]$ admits no spurious valleys in the over-parametrized regime*

$$p \geq \sum_{i=1}^d \binom{n+i-1}{i} \mathbf{1}_{\{a_i \neq 0\}} = O(n^d)$$

Under the hypothesis of Corollary 10 with $p = O(n^d)$, a generic function of V_σ, $\boldsymbol{\Phi}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{u}^T\sigma(\mathbf{W}\mathbf{x})$, can be also represented, for some $\boldsymbol{\gamma} = \boldsymbol{\gamma}(\boldsymbol{\theta})$, in the generalized linear form

$$\boldsymbol{\Phi}(\mathbf{x}; \boldsymbol{\theta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\varphi}(\mathbf{x})\rangle$$

with $\boldsymbol{\varphi}(\mathbf{x}) = (x_{k_1} \cdots x_{k_j})_{\{1\leq k_1 \leq \cdots \leq k_j \leq n, j\in[d]\}}$. The parameters $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ differ for their dimensions:

$$\dim(\boldsymbol{\gamma}) = O(n^d) < \dim(\boldsymbol{\theta}) = (n+1)\cdot O(n^d) = O(n^{d+1})$$

One would therefore like Corollary 10 to hold also (at least) for $p \geq O(n^{d-1})$. In the next section we address this problem for the linear activation $\sigma(z) = z$ and the quadratic activation $\sigma(z) = z^2$.

3.1. Improved over-parametrization bounds for homogeneous polynomial activations

The over-parametrization bounds obtained in Corollary 10 are quite non-desiderable in practical applications. We show that they can indeed be improved, for the case of linear and quadratic networks.

3.1.1. Linear networks case

Linear networks have been considered as a first order approximation of feed-forward multi-layers networks (Kawaguchi, 2016). It was shown, in several works (Kawaguchi, 2016; Freeman and Bruna, 2017; Yun et al., 2018), that, for linear networks of any depth

$$\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{W}_{K+1}\cdots\mathbf{W}_1\mathbf{x} \tag{4}$$

with $\boldsymbol{\theta} = (\mathbf{W}_{K+1}, \mathbf{W}_K, \ldots, \mathbf{W}_2, \mathbf{W}_1) \in \mathbb{R}^{m\times p_K} \times \mathbb{R}^{p_K\times p_{K-1}} \times \cdots \mathbb{R}^{p_2\times p_1} \times \mathbb{R}^{p_1\times n}$, the loss function (2) has no spurious local minima, if $\min_{i\in[K]} p_i \geq \min\{n, m\}$. This corresponds exactly with over-parametrization regime in Corollary 10, for the case of one-hidden-layer networks. The following theorem improves on Corollary 10 for the case of multi-layer linear networks, showing that no over-parametrisation is required in this case to avoid spurious valleys, for square loss functions.

Theorem 11 (Linear networks) *For linear NNs (4) of any depth $K \geq 1$ and of any layer widths $p_k \geq 1$, $k \in [K]$, and any input-output dimensions $n, m \geq 1$, the square loss function $L(\boldsymbol{\theta}) = \mathbb{E}\|\mathbf{\Phi}(\mathbf{X};\boldsymbol{\theta}) - \mathbf{Y}\|^2$ admits no spurious valleys.*

3.1.2. Quadratic networks case

Quadratic activations $\sigma(z) = z^2$ have been considered in the literature (Livni et al., 2014; Du and Lee, 2018; Soltanolkotabi et al., 2017) as second order approximation of general non-linear activations. Corollary 10 says that, if $p \geq n(n+1)/2$, the loss function (2) admits no spurious valleys. In the following theorem we relax the over-parametrisation requirement and show that $p > 2n$ is sufficient for the statement to hold, in the case of square loss functions and one dimensional output ($m = 1$).

Theorem 12 (Quadratic networks) *For one-hidden-layer NNs $\Phi(\mathbf{x};\boldsymbol{\theta}) = \mathbf{u}^T\sigma(\mathbf{W}\mathbf{x})$ with quadratic activation function $\sigma(z) = z^2$ and one-dimensional output ($m = 1$), the square loss function $L(\boldsymbol{\theta}) = \mathbb{E}|\Phi(\mathbf{X};\boldsymbol{\theta}) - Y|^2$ admits no spurious valleys in the over-parametrised regime $p \geq 2n + 1 = O(n)$.*

Sketch of the proof The proof (reported in Section A) consists in constructing a path satisfying property **P.1** and improves upon the proof of Theorem 8 by leveraging the special linearized structure of the network for quadratic activation. For every parameter $\boldsymbol{\theta} = (\mathbf{u}, \mathbf{W}) \in \mathbb{R}^p \times \mathbb{R}^{p\times n}$, we can write

$$\Phi(\mathbf{x};\boldsymbol{\theta}) = \sum_{i=1}^{p} u_i(\langle \mathbf{w}_i, \mathbf{x}\rangle)^2 = \Big\langle \sum_{i=1}^{p} u_i\mathbf{w}_i\mathbf{w}_i^T, \mathbf{x}\mathbf{x}^T\Big\rangle_F$$

We notice that $\Phi(\cdot;\boldsymbol{\theta})$ can also be represented by a NN $\Phi(\cdot;\hat{\boldsymbol{\theta}})$ with n hidden units; indeed, if $\sum_{i=1}^n \sigma_i\mathbf{v}_i\mathbf{v}_i^T$ is the SVD of $\sum_{i=1}^p u_i\mathbf{w}_i\mathbf{w}_i^T$, then $\Phi(\mathbf{x};\boldsymbol{\theta}) = \langle \sum_{i=1}^n \sigma_i\mathbf{v}_i\mathbf{v}_i^T, \mathbf{x}\mathbf{x}^T\rangle_F$. Therefore $p \geq n$ is sufficient to describe any element in V_σ. A path to the symmetric matrix defining the optimal network is then constructed by mapping the above decomposition defined by the standard form of the network.

The factor 2 in the statement is due to some technicalities in the proof, but a more involved proof should be able to extend the result to the regime $p \geq n$. The extension of such mechanism for higher order tensors (appearing as a result of multiple layers or high-order polynomial activations) using tensor decomposition also seems possible and is left for future work.

Comparison with previous works The same optimization landscape has been considered in the works (Soltanolkotabi et al., 2017) and (Du and Lee, 2018). In the first work, the authors show absence of spurious minima for the case of $p \geq 2n$ and of ERM (loss evaluated on N data points), but for fixed output layer weights; under some assumption on the output layer weights, the result is shown to still hold for $p \geq n$, if $n \leq N \leq O(n^2)$. This last condition can be removed by considering the regularized loss with non-zero weight decay, as shown in (Du and Lee, 2018); in the same work, the authors also proved absence of spurious minima in the case $p < n$ and $p(p+1) \geq 2N$ for a randomly regularized loss (with high probability).

By relaxing the statement to absence of spurious valleys, we showed that this holds for the square loss (both in population and ERM setting) and the optimisation problem over both layer weights if $p > 2n$.

3.1.3. Lower to upper intrinsic dimension gap

As observed in Lemma 6 $\dim_*(\sigma(z) = z, n) = 1$ and $\dim_*(\sigma(z) = z^2, n) = n$ for all integer $n \geq 1$. Therefore, Theorem 11 and Theorem 12 say that, for $\sigma(z) = z^k$, $k \in [2]$, and $m = 1$, the square loss function $L(\boldsymbol{\theta}) = \mathbb{E}|\Phi(\mathbf{X};\boldsymbol{\theta}) - Y|^2$ admits no spurious valleys in the over-parametrized regime $p \geq O(\dim_*(\sigma, n))$. We conjecture that this hold for any (sufficiently regular) activation function with finite intrinsic lower dimension.

4. Infinite intrinsic dimension and presence of spurious valleys

This section is devoted to the construction of worst-case scenarios for non-over parametrised networks. The main result (Theorem 13) essentially states that, for networks with width smaller than the lower intrinsic dimension defined above, spurious valleys can be created by choosing adversarial data distributions. We then show how this implies negative results for under-parametrized polynomial architectures and a large variety of architectures used in practice.

Theorem 13 *Consider the square loss function $L(\boldsymbol{\theta}) = \mathbb{E}\|\mathbf{\Phi}(\mathbf{X};\boldsymbol{\theta}) - \mathbf{Y}\|^2$ for one-hidden-layer NNs $\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x})$ with non-negative activation function $\sigma \geq 0$ such that $\sigma \in L^2(\mathbb{R}, e^{-x^2}\,dx)$. If $p \leq \frac{1}{2}\dim_*(\sigma, n-1)$, then there exists a r.v. $(\mathbf{X}, \mathbf{Y})$ such that the square loss function L admits spurious valleys. In particular, for any given $M > 0$, the r.v. $\mathbf{Y}$ can be chosen in such a way that there exists a (non-empty) open set $\Omega \subset \Theta$ such that*

$$M/2 + \min_{\boldsymbol{\theta}\in\Omega} L(\boldsymbol{\theta}) \geq \sup_{\boldsymbol{\theta}\in\Omega} L(\boldsymbol{\theta}) \geq \min_{\boldsymbol{\theta}\in\Omega} L(\boldsymbol{\theta}) \geq M + \min_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{\theta}) \tag{5}$$

and any path $\boldsymbol{\theta} : [0,1] \to \Theta$ such that $\boldsymbol{\theta}_0 \in \Omega$ and $\boldsymbol{\theta}_1$ is a global minima verifies

$$\max_{t\in[0,1]} L(\boldsymbol{\theta}_t) \geq \min_{\boldsymbol{\theta}\in\Omega} L(\boldsymbol{\theta}) + M \tag{6}$$

Equation (5) in Theorem 13 says that any local descent algorithm, if initialized in $\boldsymbol{\theta}_0 \in \Omega$, at its best it will only be able to produce a final parameter value which is at least M far from optimality. Equation (6) implies that any path starting from parameter belonging to Ω must 'up-climb' at least $M/2$ in the loss value. In the following we refer to such property, as stated in Theorem 13, by saying that *the loss function has arbitrarily bad spurious valleys*. Note that this result ensures that spurious valleys have positive Lebesgue measure, so there is a positive probability that gradient descent methods initialized with a measure that is absolutely continuous with respect to Lebesgue will get stuck in a bad local minima.

Applying the observations describing the values of the lower intrinsic dimension for different activation functions, we get the following corollaries.

Corollary 14 (Homogeneous even degree polynomial activations) *Consider the case of activation $\sigma(z) = z^{2k}$ with $k \geq 1$ integer. For one-hidden-layer NNs $\boldsymbol{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x})$, if $n \geq 2$ and the hidden layer width satisfies*

$$p \leq \begin{cases} n-1 & \text{if } k = 1 \\ \frac{1}{2}\,\mathrm{rk}_{\mathrm{S}}(2k, n-1) & \text{if } k > 1 \end{cases}$$

then there exists a r.v. $(\mathbf{X},\mathbf{Y})$ such that the square loss function $L(\boldsymbol{\theta}) = \mathbb{E}\|\boldsymbol{\Phi}(\mathbf{X};\boldsymbol{\theta}) - \mathbf{Y}\|^2$ has arbitrarily bad spurious valleys.

This follows by Theorem 13 and Lemma 6, since $\dim_*(\sigma(z) = z^{2k}, n) = \mathrm{rk}_{\mathrm{S}}(2k, n)$. For the well known case $k = 1$ (symmetric matrices) it holds $\mathrm{rk}_{\mathrm{S}}(2, n) = n$; therefore Corollary 14 implies that the bound provided in Corollary 10 is almost (up to a factor 2) tight. Notice that our result is indeed in line with the results discussed in Section 3.1.2.

Corollary 15 (Spurious valleys exist in generic architectures) *If $n \geq 2$, for one-hidden-layer NNs $\boldsymbol{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x})$ with any hidden layer width $p \geq 1$ and continuous non-negative non-polynomial activation function $\sigma \in L^2(\mathbb{R}, e^{-x^2/2})$, then there exists a r.v. $(\mathbf{X},\mathbf{Y})$ such that the square loss function $L(\boldsymbol{\theta}) = \mathbb{E}\|\boldsymbol{\Phi}(\mathbf{X};\boldsymbol{\theta}) - \mathbf{Y}\|^2$ has arbitrarily bad spurious valleys. This setting includes the following activation functions:*

- *The ReLU activation function $\sigma(z) = z_+$ and some relaxations of it, such as softplus activation functions $\sigma(z) = \beta^{-1}\log\big(1 + e^{\beta z}\big)$, with $\beta > 0$;*
- *The sigmoid activation function $\sigma(z) = (1 + e^{-z})^{-1}$ and the approximating erf function $\sigma(z) = 2/\pi \int_0^z e^{-u}\,du$, which represents an approximation to the sigmoid function.*

This follows by Theorem 13 by observing that $\dim_*(\sigma, n) = \infty$ if σ is one of the above activation functions.

Discussion and comparison with previous works Several works showed existence of spurious minima: (Safran and Shamir, 2017) showed counterexamples under Gaussian input distributions, for $p = n - 1 \in \{8, \ldots, 19\}$, using a computer-assisted proof; (Swirszcz et al., 2016) and (Zhou and Liang, 2017) provided a few numerical examples; (Yun et al., 2018) showed existence of spurious minima for ReLU-like activations under non-realizability, and provided counterexamples for smooth activations. For any number of hidden neurons p, we

give a (constructive) proof of existence of a data distribution which creates spurious valleys, under the only assumption of non-negative continuous activation function. We also remark that while in the above works the authors proved existence of spurious local minima, we prove that, in fact, arbitrarily bad spurious valleys can exist, which is a stronger negative characterization.

The results of this section can be interpreted as worst-case scenarios for the problem of optimizing (2). We showed that, even for simple one-hidden-layer neural network architectures with non-linear activation functions used in practice (such as ReLU), global optimality results can not hold, unless we make some assumptions on the data distributions.

5. Typical spurious valleys and low-energy barriers

In the previous section it was shown that whenever the number of hidden units p is below the lower intrinsic dimension, then one can show worst-case data distributions that yield a landscape with arbitrarily bad spurious valleys. A natural follow-up question is thus to consider the complexity of the energy landscape in a *typical* scenario, defined in terms of both parameter initialisation (how likely are descent algorithms to fall into a spurious valley?) and energy value (how deep are typical spurious valleys?).

In this section, we study the energy landscape under generic data distributions in case of homogeneous activation, and show that, although spurious valleys may appear, they tend do so below a certain energy level, controlled by the decay of the spectral decomposition of the kernel defined by the activation function and by the amount of parametrisation p. This offers a first glimpse at the empirical success of local descent algorithms in conditions where p is indeed below the intrinsic dimension.

We consider oracle square loss functions of the form

$$L(\boldsymbol{\theta}) = \mathbb{E}|\Phi(\mathbf{X};\boldsymbol{\theta}) - Y|^2 \tag{7}$$

for one-dimensional output one-hidden-layer NNs $\Phi(\mathbf{x};\boldsymbol{\theta}) = \mathbf{u}^T\sigma(\mathbf{W}\mathbf{x})$, with $\boldsymbol{\theta} = (\mathbf{u},\mathbf{W}) \in \mathbb{R}^p \times \mathbb{R}^{p\times n}$, σ a positively homogeneous function, and $\mathbf{X}, Y$ square integrable r.v. Notice that we can write

$$L(\boldsymbol{\theta}) = \mathbb{E}|\Phi(\mathbf{X};\boldsymbol{\theta}) - f^*(\mathbf{X})|^2 + \mathbb{E}|Y - f^*(\mathbf{X})|^2$$

for some measurable $f^* : \mathbb{R}^n \to \mathbb{R}$ such that $f^*(\mathbf{X}) = \mathbb{E}[Y \,|\, \mathbf{X}]$. In particular this implies that

$$\min_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{\theta}) \geq \mathcal{R}(\mathbf{X}, Y) \doteq \mathbb{E}|Y - f^*(\mathbf{X})|^2$$

If f^* can be written as a one-hidden-layer neural network with an arbitrary number of hidden units, that is

$$f^*(\mathbf{x}) = \int_{\mathbb{R}^n} \sigma(\langle \mathbf{x}, \mathbf{w}\rangle)\rho(\mathbf{w})\,d\mu(\mathbf{w})$$

for some measure μ and weight function ρ, then a possible approach to find a proper approximation of f^* is through random features sampling (Rahimi and Recht, 2008). Applying some recent results (Bach, 2017b) relating random features expansions with kernel quadrature rules, we show that this implies the following statement: *as the network width increases, spurious valleys tend to be confined to decreasingly low loss value.* In this regime, large loss barriers are therefore avoided with high probability over initialization of the parameters. The statement is made more rigorous in the following:

Theorem 16 *Let $d\tau$ be the uniform distribution over the unit sphere $\mathbb{S}^n$ and consider an initial parameter $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{u}}, \tilde{\mathbf{W}})$ with $\tilde{\mathbf{w}}_i \sim d\tau$ sampled i.i.d. Then the following hold:*

1. *There exists a path $t \in [0,1] \mapsto \boldsymbol{\theta}_t$ such that $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}$, the function $t \in [0,1] \mapsto L(\boldsymbol{\theta}_t)$ is non-increasing, and*

$$L(\boldsymbol{\theta}_1) \leq \mathcal{R}(\mathbf{X}, Y) + \lambda \quad \text{if} \quad p \geq O\big(-\lambda^{-1}\log(\lambda\delta)\big)$$

with probability greater or equal then $1-\delta$, for every $\lambda, \delta \in (0,1)$.

2. *If f^* is sufficiently regular[3], there exists a path $t \in [0,1] \mapsto \boldsymbol{\theta}_t$ such that $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}$, the function $t \in [0,1] \mapsto L(\boldsymbol{\theta}_t)$ is non-increasing, and*

$$L(\boldsymbol{\theta}_1) \leq \mathcal{R}(\mathbf{X}, Y) + O(p^{-1+\delta})$$

with probability greater or equal then $1 - e^{-O(p^\delta)}$ for every $\delta \in (0,1)$.

Sketch of the proof Assume that f^* admits the representation

$$f^*(\mathbf{x}) = \int_\Theta \rho(\mathbf{w})\sigma(\langle \mathbf{x}, \mathbf{w}\rangle)\, d\tau(\mathbf{w})$$

for some density ρ. If $\mathbf{w}_i \sim d\tau$, $i \in [p]$, are drawn i.i.d., we have

$$\mathbb{E}\left(\frac{1}{p}\sum_{i=1}^p \rho(\mathbf{w}_i)\sigma(\langle \mathbf{w}_i, \mathbf{x}\rangle) - f^*(\mathbf{x})\right)^2 = O\left(\frac{1}{p}\right)$$

Notice that by only moving the second layer, we can construct a (linear) descent path from $(\tilde{\mathbf{u}}, \tilde{\mathbf{W}})$ to $(\mathbf{u}, \tilde{\mathbf{W}})$, where $u_i = \rho(\mathbf{w}_i)$. The proof is then concluded by applying an Hoeffding's-type inequality to get property 2. if it holds $\rho \in L^\infty(\mathbb{S}^n\, d\tau)$ or by applying Proposition 1 in (Bach, 2017b) to obtain property 1.

Related works Many recent works leveraged arguments based on random features to explain the empirical success of local descent algorithms to train neural networks (see e.g. (Jacot et al., 2018; Allen-Zhu et al., 2018; Oymak and Soltanolkotabi, 2019; Yehudai and Shamir, 2019; Ma et al., 2019; Du et al., 2018)). In Theorem 16, we used this type of technique to show properties of the optimization landscape. The main limitation shared by our and the cited results is the gap between the regimes in which the apply (high over-parametrized NNs) and the regimes attained in practice. A current important direction is to understand the dynamics of neural networks training over kernel approximation and to extend such results to *moderatly* over-parametrized architectures.

Remark 17 *Notice that in the previous description of the problem, we dropped bias terms from the neural network architectures for sake of simplicity, as we can immediately generalize to the biases case by stacking the bias in the weights and input random variables. As a bias term is needed in order to use universal approximation results, with abuse of notation, in the above theorem we wrote $\langle \mathbf{w}, \mathbf{x}\rangle \doteq \langle \mathbf{w}^{(n)}, \mathbf{x}\rangle + w_{n+1}$ for $\mathbf{w} \in \mathbb{S}^n, \mathbf{x} \in \mathbb{R}^n$, where $\mathbf{w}^{(n)} = (w_1, \ldots, w_n)$ represents a neuron weight and w_{n+1} a bias term; again, note that this can be done by simply considering the r.v. $\tilde{\mathbf{X}} \doteq (\mathbf{X}, 1)$ in place of $\mathbf{X}$.*

3. More precisely, if the function f^* can be written as $f^*(\mathbf{x}) = \int_{\mathcal{W}} g^*(\mathbf{w})\psi_{\mathbf{w}}(\mathbf{x})\, d\tau(\mathbf{w})$ for some $g^* \in L^\infty_{d\tau}$.

6. Future directions

We considered the problem of characterizing the loss surface of neural networks from the perspective of optimization, with the goal of deriving weak certificates that enable - or prevent - the existence of descent paths towards global minima.

The topological properties studied in this paper, however, do not yet capture fundamental aspects that are necessary to explain the empirical success of deep learning methods. We identify a number of different directions that deserve further attention.

The positive results presented above rely on being able to reduce the network to the case when (convex) optimization over the output layer is sufficient to reach optimal weight values. A better understanding of first layer dynamics needs to be carried out. Moreover, in such positive results we only proved non-existence of (high) energy barriers. While this is an interesting property from the optimization point of view, it is also not sufficient to guarantee convergence of local descent algorithms. Another informative property of the loss function that should be addressed in future works is the existence of local descents in non optimal points: for every $\boldsymbol{\theta}_0 \in \Theta$ non optimal and any neighborhood $\mathcal{U} \subseteq \Theta$ of $\boldsymbol{\theta}_0$, there exists $\boldsymbol{\theta} \in \mathcal{U}$ such that $L(\boldsymbol{\theta}) < L(\boldsymbol{\theta}_0)$. More generally, our present work is not informative on the performance of gradient descent in the regimes with no spurious valley.

The other very important point to be addressed in future is how to extend the above results to architectures of more practical interest. Depth and the specific linear structure of Convolutional Neural Networks, critical to explain the excellent empirical performance of deep learning in computer vision, text or speech, need to be exploited, as well as specific design choices such as Residual connections and several normalization strategies – as done recently in (Shamir, 2018) and (Santurkar et al., 2018) respectively. This also requires making specific assumptions on the data distribution, and is left for future work.

Acknowledgements We would like to thank Gérard Ben Arous and Léon Bottou for fruitful discussions, and Jean Ponce for valuable comments and corrections of the original version of this manuscript. LV would also like to thank Jumageldi Charyyev for fruitful discussions on the proofs of several propositions and Andrea Ottolini for valuable comments on a previous version of this manuscript. LV was partially supported by NSF grant DMS-1719545. ASB was partially supported by NSF grants DMS-1712730 and DMS-1719545, and by a grant from the Sloan Foundation. JB acknowledges the partial support by the Alfred P. Sloan Foundation, NSF RI-1816753, NSF CAREER CIF 1845360, and Samsung Electronics.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017a.

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017b.

Afonso S Bandeira, Nicolas Boumal, and Vladislav Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In *Conference on Learning Theory*, pages 361–382, 2016.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Leon Bottou, Martin Arjovsky, David Lopez-Paz, and Maxime Oquab. Geometrical insights for implicit generative modeling. *arXiv preprint arXiv:1712.07822*, 2017.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1254–1279, 2008.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A Hamprecht. Essentially no barriers in neural network energy landscape. *arXiv preprint arXiv:1803.00885*, 2018.

Simon S Du and Jason D Lee. On the power of over-parametrization in neural networks with quadratic activation. *arXiv preprint arXiv:1803.01206*, 2018.

Simon S Du, Jason D Lee, Yuandong Tian, Barnabas Poczos, and Aarti Singh. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

Soheil Feizi, Hamid Javadi, Jesse Zhang, and David Tse. Porcupine neural networks:(almost) all local optima are global. *arXiv preprint arXiv:1710.02196*, 2017.

Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *ICLR 2017*, 2017.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *arXiv preprint arXiv:1704.00708*, 2017a.

Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017b.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018.

Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pages 586–594, 2016.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pages 855–863, 2014.

Chao Ma, Lei Wu, et al. A comparative analysis of the optimization and generalization property of two-layer neural network and random feature models under gradient descent dynamics. *arXiv preprint arXiv:1904.04326*, 2019.

Marco Mondelli and Andrea Montanari. On the connection between learning two-layers neural networks and tensor decomposition. *arXiv preprint arXiv:1802.07301*, 2018.

Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. *arXiv preprint arXiv:1704.08045*, 2017.

Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *arXiv preprint arXiv:1902.04674*, 2019.

Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *International Conference on Machine Learning*, pages 2798–2806, 2017.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

Itay Safran and Ohad Shamir. On the quality of the initial basin in overspecified neural networks. In *International Conference on Machine Learning*, pages 774–782, 2016.

Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks. *arXiv preprint arXiv:1712.08968*, 2017.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?(no, it is not about internal covariate shift). *arXiv preprint arXiv:1805.11604*, 2018.

Ohad Shamir. Are resnets provably better than linear predictors? *arXiv preprint arXiv:1804.06739*, 2018.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926*, 2017.

Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

Daniel Soudry, Elad Hoffer, and Nathan Srebro. The implicit bias of gradient descent on separable data. *arXiv preprint arXiv:1710.10345*, 2017.

Grzegorz Swirszcz, Wojciech Marian Czarnecki, and Razvan Pascanu. Local minima in training of neural networks. *arXiv preprint arXiv:1611.06310*, 2016.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. *arXiv preprint arXiv:1904.00687*, 2019.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small nonlinearities in activation functions create bad local minima in neural networks. *arXiv preprint arXiv:1802.03487*, 2018.

Yi Zhou and Yingbin Liang. Critical points of neural networks: Analytical forms and landscape properties. *arXiv preprint arXiv:1710.11205*, 2017.

Appendix A. Proofs of Section 3

Notations For any r.v.'s $\mathbf{X}$ and $\mathbf{Y}$ with values in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively, we denote $\mathbf{\Sigma_X} = \mathbb{E}\big[\mathbf{X}\mathbf{X}^T\big]$ and $\mathbf{\Sigma_{XY}} = \mathbb{E}\big[\mathbf{X}\mathbf{Y}^T\big]$. For every integer $n \geq 1$, we denote by $GL(n)$, $O(n)$ and $SO(n)$, respectively, the general linear group, the orthogonal group and the special orthogonal group of real $n \times n$ matrices. $\mathbf{I}$ denotes the identity matrix and $\mathbf{e}_1, \dots, \mathbf{e}_n$ the standard basis in $\mathbb{R}^n$.

A.1. Proof of Theorem 8

We note that, under the assumptions of Theorem 8, the same optimal NN functions $\Phi_i(\cdot;\boldsymbol{\theta})$ could also be obtained using a generalized linear model, where the representation function has the linear form $\Phi_i(\mathbf{x};\boldsymbol{\theta}) = \langle \boldsymbol{\theta}_i, \boldsymbol{\varphi}(\mathbf{x})\rangle$, for some parameter independent function $\boldsymbol{\varphi} : \mathbb{R}^n \to \mathbb{R}^{\dim^*(\sigma,\mathbf{X})}$. The main difference between the two models is that the former requires the choice of a non-linear activation function σ, while the latter implies the choice of a kernel functions. This is the content of the following lemma.

Lemma 18 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a continuous function and $\mathbf{X} \in \mathcal{R}_2(\sigma, n)$ a r.v. Assume that the linear space*

$$V_{\sigma,\mathbf{X}} \doteq \operatorname{span}(\{f : f \in V_{\sigma,1}\}) \subseteq L^2_{\mathbf{X}}$$

is finite dimensional. Then there exists a scalar product $\langle\cdot,\cdot\rangle$ on $V_{\sigma,\mathbf{X}}$ and a map $\mathbf{x} \in \mathbb{R}^n \mapsto \boldsymbol{\varphi}(\mathbf{x}) \in V_{\sigma,\mathbf{X}}$ such that

$$\langle \boldsymbol{\psi}_{\sigma,\mathbf{w}}, \boldsymbol{\varphi}(\mathbf{x})\rangle = \psi_{\sigma,\mathbf{w}}(\mathbf{x}) = \sigma(\langle \mathbf{w}, \mathbf{x}\rangle) \tag{8}$$

for all $\mathbf{w} \in \mathbb{R}^n$. Moreover, the function $\mathbf{w} \in \mathbb{R}^n \mapsto \boldsymbol{\psi}_{\sigma,\mathbf{w}} \in V_{\sigma,\mathbf{X}}$ is continuous.

Proof For sake of simplicity, in the following we write $\psi_{\mathbf{w}}$ for $\psi_{\sigma,\mathbf{w}}$ and V for $V_{\sigma,\mathbf{X}}$. Let $\boldsymbol{\psi}_{\mathbf{w}_1}, \dots, \boldsymbol{\psi}_{\mathbf{w}_q}$ be a basis of V. If $\boldsymbol{\psi}_{\mathbf{w}} = \sum_{i=1}^q \alpha_i \boldsymbol{\psi}_{\mathbf{w}_i}$ and $\psi_{\mathbf{v}} = \sum_{j=1}^q \beta_j \boldsymbol{\psi}_{\mathbf{w}_j}$, then we can define a scalar product on V as

$$\langle \boldsymbol{\psi}_{\mathbf{w}}, \boldsymbol{\psi}_{\mathbf{v}}\rangle \doteq \sum_{i=1}^q \alpha_i \beta_i$$

If we define the map $\mathbf{x} \in \mathbb{R}^n \mapsto \boldsymbol{\varphi}(\mathbf{x}) \in V$ as

$$\boldsymbol{\varphi}(\mathbf{x}) = \sum_{i=1}^q \psi_{\mathbf{w}_i}(\mathbf{x}) \boldsymbol{\psi}_{\mathbf{w}_i}$$

then property (8) follows directly by the definition of the function $\psi_{\mathbf{w}}$. Moreover, we can choose $\mathbf{x}_1, \dots, \mathbf{x}_q$ such that $\boldsymbol{\varphi}(\mathbf{x}_1), \dots, \boldsymbol{\varphi}(\mathbf{x}_q)$ is a basis of V. Now we need to show that, for $i \in [q]$, the map $\mathbf{w} \mapsto \langle \boldsymbol{\psi}_{\mathbf{w}}, \boldsymbol{\psi}_{\mathbf{w}_i}\rangle$ is continuous. Let $\mathbf{M}$ be the matrix $\mathbf{M} \doteq (\psi_{\mathbf{w}_j}(\mathbf{x}_i))_{i,j} \in \mathbb{R}^{q\times q}$ and $\mathbf{z}(\mathbf{w})$ be the vector $\mathbf{z}(\mathbf{w}) \doteq (\psi_{\mathbf{w}}(\mathbf{x}_i))_i \in \mathbb{R}^q$. Then $\langle \boldsymbol{\psi}_{\mathbf{w}}, \boldsymbol{\psi}_{\mathbf{w}_i}\rangle = (\mathbf{M}^{-1}\mathbf{z}(\mathbf{w}))_i$, which is continuous in $\mathbf{w}$. This shows that the map $\mathbf{w} \in \mathbb{R}^n \mapsto \boldsymbol{\psi}_{\mathbf{w}} \in V$ is continuous. ■

The non-trivial fact captured by Theorem 8 is the following: when the capacity of network is large enough to match a generalized linear model, but still finite, then the problem of optimizing the loss function (2), which is in general a highly non-convex object, satisfies an

interesting optimization property in view of the local descent algorithms which are used in practice to solve it.

Proof [Proof of Theorem 8] Thanks to Lemma 18, there exist two continuous maps $\boldsymbol{\varphi}, \boldsymbol{\psi} : \mathbb{R}^n \to \mathbb{R}^q \simeq V_{\sigma,\mathbf{X}}$, with $q = \dim^*(\sigma, \mathbf{X})$, such that $\sigma(\langle \mathbf{w}, \mathbf{x}\rangle) = \langle \boldsymbol{\psi}(\mathbf{w}), \boldsymbol{\varphi}(\mathbf{x})\rangle$ for every $\mathbf{w}, \mathbf{x} \in \mathbb{R}^n$. Therefore, every one-hidden-layer NN $\boldsymbol{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\sigma(\mathbf{W}\mathbf{x})$ can be written as $\boldsymbol{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\boldsymbol{\psi}(\mathbf{W})\boldsymbol{\varphi}(\mathbf{x})$, where, if $\mathbf{W} \in \mathbb{R}^{p\times n}$, then $\boldsymbol{\psi}(\mathbf{W}) \in \mathbb{R}^{p\times q}$ (that is $\boldsymbol{\psi}$ is applied row-wise).

The proof of the Theorem consists in exploiting the above *linearized* representation of $\boldsymbol{\Phi}$ to show that property **P.1** holds (remind that this is equivalent to saying that the loss function has no spurious valleys). Given an initial parameter $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{U}}, \tilde{\mathbf{W}})$, we want to construct a continuous path $t \in [0,1] \mapsto \boldsymbol{\theta}_t = (\mathbf{U}_t, \mathbf{W}_t)$, such that the function $t \in [0,1] \mapsto L(\boldsymbol{\theta}_t)$ is non-increasing and such that $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}$, $\boldsymbol{\theta}_1 \in \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$, where $L(\boldsymbol{\theta}) = \mathbb{E}[\ell(\boldsymbol{\Phi}(\mathbf{X};\boldsymbol{\theta}), \mathbf{Y})]$. The construction of such a path can be articulated in two main steps:

Step 1. The first part of the path consist showing that we can assume that $\mathrm{rk}(\boldsymbol{\psi}(\tilde{\mathbf{W}})) = q$ w.l.o.g. Let $\mathbf{w}_1^T, \dots, \mathbf{w}_p^T \in \mathbb{R}^n$ be the rows of $\tilde{\mathbf{W}}$; suppose that $\mathrm{rk}(\boldsymbol{\psi}(\tilde{\mathbf{W}})) = r < q$ (otherwise there is nothing to show) and that $\boldsymbol{\psi}(\mathbf{w}_{i_1}), \dots, \boldsymbol{\psi}(\mathbf{w}_{i_r})$ are linearly independent. Denote $I = \{i_1, \dots, i_r\}$, $J = [1,p] \setminus I = \{j_1, \dots, j_{p-r}\}$ and $\mathbf{u}_1, \dots, \mathbf{u}_p$ the columns of $\tilde{\mathbf{U}}$. For $j \in J$, we can write

$$\boldsymbol{\psi}(\mathbf{w}_j) = \sum_{k=1}^{r} a_j^k \, \boldsymbol{\psi}(\mathbf{w}_{i_k}) \quad \text{for some } a_j^k \in \mathbb{R} \tag{9}$$

If we define $\mathbf{U}_1$ such that (denoting $\mathbf{u}_{1,i}$ the i-th row of $\mathbf{U}_1$)

$$\mathbf{u}_{1,i} = \mathbf{u}_i + \sum_{k=1}^{n-r} a_k^i \, \mathbf{u}_{j_k} \quad \text{for } i \in I, \quad \mathbf{u}_{1,j} = 0 \quad \text{for } j \in J$$

then $\mathbf{U}_1\tilde{\mathbf{W}} = \tilde{\mathbf{U}}\tilde{\mathbf{W}}$. The path $t \in [0,1/2] \mapsto \boldsymbol{\theta}_t = (2t\,\mathbf{U}_1 + (1-2t)\tilde{\mathbf{U}}, \tilde{\mathbf{W}})$ leaves the network unchanged, i.e. $\boldsymbol{\Phi}(\cdot;\tilde{\boldsymbol{\theta}}) = \boldsymbol{\Phi}(\cdot;\boldsymbol{\theta}_t)$ for $t \in [0,1/2]$. At this point, we can select $\mathbf{w}_{1,j_1}, \dots, \mathbf{w}_{1,j_{p-r}} \in \mathbb{R}^n$ such that the matrix $\mathbf{W}_1$ with rows $\mathbf{w}_{1,i} = \mathbf{w}_i$ for $i \in I$ and $\mathbf{w}_{1,j}$ for $j \in J$, verifies $\mathrm{rk}(\boldsymbol{\psi}(\mathbf{W}_1)) = q$. Notice that the existence of such vectors $\mathbf{w}_{1,j_k}$, $k \in [p-r]$, is guaranteed by the definition of $q = \dim^*(\sigma, \mathbf{X})$. The path $t \in [1/2, 1] \mapsto \boldsymbol{\theta}_t = (\mathbf{U}_1, (2t-1)\mathbf{W}_1 + (2-2t)\tilde{\mathbf{W}})$ leaves the network unchanged, i.e. $\boldsymbol{\Phi}(\cdot;\boldsymbol{\theta}_0) = \boldsymbol{\Phi}(\cdot;\boldsymbol{\theta}_t)$ for $t \in [0,1]$. The new parameter value $\boldsymbol{\theta}_1 = (\mathbf{U}_1, \mathbf{W}_1)$ satisfies $\mathrm{rk}(\boldsymbol{\psi}(\mathbf{W}_1)) = q$.

Step 2. By step 1, we can assume that $\mathrm{rk}(\tilde{\mathbf{W}}) = q$. Since the network has the form $\boldsymbol{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\boldsymbol{\psi}(\mathbf{W})\boldsymbol{\varphi}(\mathbf{x})$ and since the function ℓ is convex, there exists $\mathbf{U}^* \in \mathbb{R}^{m\times p}$ such that $\boldsymbol{\theta} = (\mathbf{U}^*, \tilde{\mathbf{W}}) \in \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$. The proof is therefore concluded by selecting the path $t \in [0,1] \mapsto \boldsymbol{\theta}_t = (t\mathbf{U}^* + (1-t)\tilde{\mathbf{U}}, \tilde{\mathbf{W}})$.

This shows that property **P.1** holds and therefore it proves the theorem. ■

A.2. Proof of Theorem 11

The first step for proving Theorem 11 consists in extending the result of Theorem 8 to the case of one-hidden-layer linear NNs $\boldsymbol{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\mathbf{W}\mathbf{x}$ with $\mathbf{U} \in \mathbb{R}^{m\times p}$, $\mathbf{W} \in \mathbb{R}^{p\times n}$ with $p < n$

and square loss functions $L(\boldsymbol{\theta}) = \mathbb{E}\|\boldsymbol{\Phi}(\mathbf{X};\boldsymbol{\theta}) - \mathbf{Y}\|^2$. We start by pointing out a symmetry property of this type of networks: for every $\mathbf{G} \in GL(p)$ it holds that

$$\boldsymbol{\Phi}(\mathbf{x};(\mathbf{U},\mathbf{W})) = \mathbf{U}\mathbf{W}\mathbf{x} = (\mathbf{U}\mathbf{G}^{-1})(\mathbf{G}\mathbf{W})\mathbf{x} = \boldsymbol{\Phi}(\mathbf{x};(\mathbf{U}\mathbf{G}^{-1}, \mathbf{G}\mathbf{W})) \tag{10}$$

This means that the map $\boldsymbol{\theta} \mapsto \boldsymbol{\Phi}(\cdot;\boldsymbol{\theta})$ is defined up to an action of the group $GL(p)$ over the parameter space $\Theta = \mathbb{R}^{m\times p} \times \mathbb{R}^{p\times n}$; the same remark holds for the loss function $L(\boldsymbol{\theta})$. We can therefore think about the loss function as defined over the topological quotient $\Theta/GL(p)$. We denote the orbit of an element $\boldsymbol{\theta} = (\mathbf{U},\mathbf{W}) \in \Theta$ as

$$[\boldsymbol{\theta}] = [\mathbf{U},\mathbf{W}] = \{\mathbf{G}\cdot\boldsymbol{\theta} = (\mathbf{U}\mathbf{G}^{-1}, \mathbf{G}\mathbf{W}) : \mathbf{G} \in GL(p)\}$$

If g is a real-valued function defined on Θ such that $g(\mathbf{G}\cdot\boldsymbol{\theta}) = g(\boldsymbol{\theta})$ for all $\mathbf{G} \in GL(p)$ and $\boldsymbol{\theta} \in \Theta$, then one can equivalently consider g as defined on $\Theta/GL(p)$ as $g([\boldsymbol{\theta}]) = g(\boldsymbol{\theta})$; for simplicity we denote $g[\boldsymbol{\theta}] = g([\boldsymbol{\theta}])$. This is exactly the case for the loss function $L(\boldsymbol{\theta})$. In the proof of Theorem 8, we describe how to construct a path from an initial parameter value $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{U}}, \tilde{\mathbf{W}})$ to a parameter value $\boldsymbol{\theta}_1 = (\mathbf{q}(\mathbf{W}_1), \mathbf{W}_1)$, with $\mathrm{rk}(\mathbf{W}_1) = p$ and $\mathbf{q} : \mathbb{R}^{p\times n} \to \mathbb{R}^{m\times p}$ the function defined by

$$\mathbf{q}(\mathbf{W}) = \boldsymbol{\Sigma}_{\mathbf{YX}}\mathbf{W}^T(\mathbf{W}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{W}^T)^\dagger \in \arg\min_{\mathbf{U}} L(\boldsymbol{\theta})|_{\boldsymbol{\theta}=(\mathbf{U},\mathbf{W})}$$

(see Lemma 28). Therefore, let $\tilde{\boldsymbol{\theta}} = (\mathbf{q}(\tilde{\mathbf{W}}), \tilde{\mathbf{W}})$ with $\mathrm{rk}(\tilde{\mathbf{W}}) = p$, be an initial parameter. Since an optimal parameter is given by $\boldsymbol{\theta} = (\mathbf{q}(\mathbf{W}), \mathbf{W})$ for some $\mathbf{W}$, we seek for a path in the form $\boldsymbol{\theta}_t = (\mathbf{q}(\mathbf{W}_t), \mathbf{W}_t)$ with $\mathrm{rk}(\mathbf{W}_t) = p$ for all $t \in [0,1]$. This path must be such that $t \mapsto L(\boldsymbol{\theta}_t)$ is non-increasing. If we assume that $\boldsymbol{\Sigma}_{\mathbf{X}} = \mathbf{I}$, it holds

$$L(\boldsymbol{\theta}_t) = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{Y}}) - \mathrm{tr}(\mathbf{M}\mathbf{P}_{\mathbf{W}_t})$$

where $\mathbf{M}$ is a PSD matrix and, for every matrix $\mathbf{W}$, $\mathbf{P}_{\mathbf{W}}$ denotes the orthogonal projection on the rows of $\mathbf{W}$, that is $\mathbf{P}_{\mathbf{W}} = \mathbf{W}^\dagger\mathbf{W}$ (see Lemma 28). Therefore it is equivalent for the path $\boldsymbol{\theta}_t = (\mathbf{q}(\mathbf{W}_t), \mathbf{W}_t)$ to be such that the function

$$t \in [0,1] \mapsto f(\mathbf{W}_t) \doteq \mathrm{tr}(\mathbf{M}\mathbf{P}_{\mathbf{W}_t})$$

is non-decreasing. In particular, the function f is defined up to the action of the group $GL(p)$ on Θ. Since we look for $\mathbf{W}_t$ of rank p, we can consider f as defined on $G(p,n)$, the Grassmanian of p dimensional linear subspaces of $\mathbb{R}^n$. The proof below for the linear one-hidden-layer case is articulated as follows. We first construct a path $[\mathbf{W}_t] \in G(p,n)$ such that $[\mathbf{W}_0] = [\tilde{\mathbf{W}}]$, $[\mathbf{W}_1]$ maximizes f and such that the function $t \in [0,1] \mapsto f[\mathbf{W}_t]$ is non-decreasing (Lemma 19). We then show that such a path can be *lifted* to a corresponding path $\mathbf{W}_t \in \mathbb{R}^{p\times n}$ (Lemma 20). Finally, we show that we can drop the assumption $\boldsymbol{\Sigma}_{\mathbf{X}} = \mathbf{I}$ and the result still holds (Lemma 21).

Lemma 19 *Let $[\tilde{\mathbf{W}}] \in G(p,n)$ and assume $\boldsymbol{\Sigma}_{\mathbf{X}} = \mathbf{I}$. Then there exists a continuous path $t \in [0,1] \mapsto [\mathbf{W}_t] \in G(p,n)$ such that $[\mathbf{W}_0] = [\tilde{\mathbf{W}}]$, $[\mathbf{W}_1]$ maximizes f and such that the function $t \in [0,1] \mapsto f[\mathbf{W}_t]$ is non-decreasing.*

Proof While it is geometrically intuitive that the results should hold, we derive a constructive proof. We start by noticing that if $[\mathbf{W}] \in G(p,n)$ and $\mathbf{w}_1, \dots, \mathbf{w}_p$ is an orthonormal basis of $[\mathbf{W}]$, then

$$f[\mathbf{W}] = \sum_{i=1}^{p} \mathbf{w}_i^T \mathbf{M} \mathbf{w}_i \tag{11}$$

Moreover, if $\mathbf{M} = \sum_{j=1}^{n} \sigma_i \mathbf{v}_j \mathbf{v}_j^T$ is the SVD of $\mathbf{M}$, where $\sigma_1 \geq \dots \geq \sigma_n \geq 0$, then (11) can be written as

$$f[\mathbf{W}] = \sum_{j=1}^{n} \sigma_j \sum_{i=1}^{p} \langle \mathbf{v}_j, \mathbf{w}_i \rangle^2$$

In particular the maximum of f is obtained for $[\mathbf{W}] = [\mathbf{V}] \doteq [\mathbf{v}_1, \dots, \mathbf{v}_p]$ (with some abuse of notation, we identify a subspace with one of its basis). To prove the result is therefore sufficient to show a path $[\mathbf{W}_t]$ from any $[\mathbf{W}_0] = [\tilde{\mathbf{W}}]$ to $[\mathbf{W}_1] = [\mathbf{V}]$, such that the function $t \in [0,1] \mapsto f[\mathbf{W}_t]$ is non-decreasing. To do this we construct a finite sequence of paths

$$[\mathbf{W}_t^i] \quad \text{such that} \quad [\mathbf{W}_0^i] = [\mathbf{W}^{i-1}] \quad \text{and} \quad [\mathbf{W}_1^i] = [\mathbf{W}^i]$$

for $i \in [p]$, with $[\mathbf{W}^0] = [\tilde{\mathbf{W}}]$, $[\mathbf{W}^p] = [\mathbf{V}]$ and

$$\mathbf{W}^i = [\mathbf{v}_1, \dots, \mathbf{v}_i, \mathbf{w}_{i+1}^{i-1}, \dots, \mathbf{w}_p^{i-1}] \quad \text{for } i \in [p]$$

where $\mathbf{w}_1^j = \mathbf{v}_1, \dots, \mathbf{w}_j^j = \mathbf{v}_j, \mathbf{w}_{j+1}^j, \dots, \mathbf{w}_p^j$ is an orthonormal basis of $[\mathbf{W}^j]$, for $j \in [0,p]$. Moreover, the paths $[\mathbf{W}_t^i]$ are such that the functions $t \in [0,1] \mapsto f[\mathbf{W}_t^i]$ are non-decreasing. Such paths are defined as follows. Let $i \in [0, p-1]$ and consider

$$[\mathbf{W}^i] = [\mathbf{w}_1^i = \mathbf{v}_1, \dots, \mathbf{w}_i^i = \mathbf{v}_i, \mathbf{w}_{i+1}^i, \dots, \mathbf{w}_p^i]$$

We define

$$\mathbf{u}_{i+1}^i = \begin{cases} \frac{\mathbf{P}_{\mathbf{W}^i} \mathbf{v}_{i+1}}{\|\mathbf{P}_{\mathbf{W}^i} \mathbf{v}_{i+1}\|} & \text{if } \mathbf{P}_{\mathbf{W}^i} \mathbf{v}_{i+1} \neq \mathbf{0} \\ \mathbf{0} & \text{o.w.} \end{cases}$$

Then we complete $\mathbf{v}_1, \dots, \mathbf{v}_i, \mathbf{u}_{i+1}^i$ to an orthonormal basis of $[\mathbf{W}^i]$:

$$\mathbf{v}_1, \dots, \mathbf{v}_i, \mathbf{u}_{i+1}^i, \dots, \mathbf{u}_p^i$$

We call $\mathbf{w}_j^{i+1} = \mathbf{u}_j^i$ for $j \in [i+2, p]$ and we define

$$[\mathbf{W}^{i+1}] = [\mathbf{v}_1, \dots, \mathbf{v}_i, \mathbf{w}_{i+1}^{i+1} = \mathbf{v}_{i+1}, \mathbf{w}_{i+2}^{i+1}, \dots, \mathbf{w}_p^{i+1}]$$

The path $[\mathbf{W}_t^i]$ is then obtained by moving $\mathbf{u}_{i+1}^i$ to $\mathbf{v}_{i+1}$ on a geodesic on the unit sphere $S^{n-1} \subset \mathbb{R}^n$, i.e.

$$[\mathbf{W}_t^{i+1}] = [\mathbf{v}_1, \dots, \mathbf{v}_i, \mathbf{u}_{i+1}^i(t), \mathbf{u}_{i+2}^i, \dots, \mathbf{u}_p^i]$$

where we defined

$$\mathbf{u}_{i+1}^i(t) \;=\; (1 - (1-\mu_{i+1})t)\mathbf{u}_{i+1}^i \;+\; \sqrt{1 - (1-(1-\mu_{i+1})t)^2} \cdot \frac{\mathbf{v}_{i+1} - \mu_{i+1}\mathbf{u}_{i+1}^i}{\sqrt{1-\mu_{i+1}^2}}$$

for $\mu_{i+1} = \langle \mathbf{u}_{i+1}^i, \mathbf{v}_{i+1} \rangle$. The fact that the function $t \in [0,1] \mapsto f[\mathbf{W}_t^{i+1}]$ is non-decreasing can be proved by noticing that

$$f[\mathbf{W}_t^{i+1}] - f[\mathbf{W}^i] = \sum_{j=i+1}^{n} \sigma_j \langle \mathbf{u}_{i+1}^i(t), \mathbf{v}_j \rangle^2$$

and by showing that the derivative of the RHS is greater or equal than 0. This concludes the proof of the lemma. ■

Lemma 20 *Let $\tilde{\mathbf{W}} \in \mathbb{R}^{p \times n}$ and assume $\mathbf{\Sigma}_X = \mathbf{I}$. Then there exists a continuous path $t \in [0,1] \mapsto \mathbf{W}_t \in \mathbb{R}^{p \times n}$ such that $\mathbf{W}_0 = \tilde{\mathbf{W}}$, $\mathbf{W}_1$ maximizes f and such that the function $t \in [0,1] \mapsto f(\mathbf{W}_t)$ is non-decreasing.*

Proof The only thing we need to prove in this case is that we can *lift* the paths $[\mathbf{W}_t^i] \in G(p,n)$ from the proof of Lemma 19 to continuous paths $\mathbf{W}_t^i \in \mathbb{R}^{n \times p}$. We first notice that if the basis $\{\mathbf{w}_1^i, \dots, \mathbf{w}_p^i\}$ and $\{\mathbf{w}_1^i, \dots, \mathbf{w}_i^i, \mathbf{u}_{i+1}^i, \dots, \mathbf{u}_p^i\}$ are defined as above, then we can assume (up to changing some signs) that they have all the same orientation, for all $i \in [0,p]$. Therefore we can define the matrices $\mathbf{W}^i \in \mathbb{R}^{p \times n}$ with rows $\mathbf{w}_1^i, \dots, \mathbf{w}_p^i$ and the matrices $U^i \in \mathbb{R}^{p \times n}$ with rows $\mathbf{w}_1^i, \dots, \mathbf{w}_i^i, \mathbf{u}_{i+1}^i, \dots, \mathbf{u}_p^i$, for $i \in [0,p]$. The paths $\mathbf{W}_t^{i+1}$ are defined in the same way as in the proof of Lemma 19. Notice that such paths go from $\mathbf{W}_0^{i+1} = \mathbf{U}^i$ to $\mathbf{W}_1^{i+1} = \mathbf{W}^{i+1}$. It remains to construct paths from $\mathbf{W}^i$ to $\mathbf{U}^i$. Consider the matrix

$$\mathbf{O}^i = \mathbf{W}_i^T \mathbf{U}_i \in SO(n)$$

Notice that $\mathbf{W}^i \mathbf{O}^i = \mathbf{U}^i$. In particular there exist $\mathbf{A}^i$ real skew-symmetric such that $\mathbf{O}^i = e^{\mathbf{A}^i}$. Therefore the paths $t \in [0,1] \mapsto \mathbf{U}_t^i = \mathbf{W}^i e^{t\mathbf{A}^i}$ go from $\mathbf{U}_0^i = \mathbf{W}^i$ to $\mathbf{U}_1^i = \mathbf{U}^i$. Moreover $f(\mathbf{U}_t^i)$ is constant in t (since the underlying linear subspace does not change). The only thing that remains to prove is that, given the matrix $\tilde{\mathbf{W}} \in \mathbb{R}^{n \times p}$ with columns $\mathbf{w}_1, \dots, \mathbf{w}_p$, there is a path from $\tilde{\mathbf{W}}$ to $\mathbf{W}^0$. Now, $\mathbf{W}^0$ was chosen as a matrix with orthonormal columns such that $[\tilde{\mathbf{W}}] = [\mathbf{W}^0]$. Therefore if $\tilde{\mathbf{W}} = \mathbf{O}\mathbf{\Lambda}\mathbf{U}$ is the SVD of $\tilde{\mathbf{W}}$ with $\mathbf{U} = \mathbf{W}^0$, $\mathbf{\Lambda} = \mathrm{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{p \times p}$ (with $\sigma_i > 0$, $i \in [p]$) and $\mathbf{O} \in SO(p)$, there exists $\mathbf{A}$ real skew-symmetric such that $\mathbf{O} = e^{\mathbf{A}}$. Thus the path $t \in [0,1] \mapsto \mathbf{W}_t = e^{(1-t)\mathbf{A}} \mathbf{\Lambda}^{1-t} \mathbf{W}^0$ is a path between $\mathbf{W}_0 = \tilde{\mathbf{W}}$ and $\mathbf{W}_1 = \mathbf{W}^0$. This concludes the proof of the lemma. ■

Lemma 21 *Lemma 20 holds even if we drop the assumption $\mathbf{\Sigma}_\mathbf{X} = \mathbf{I}$.*

Proof For sake of simplicity we distinguish two cases.
Case 1: $\mathrm{rk}(\mathbf{\Sigma}_\mathbf{X}) = n$. Let $\mathbf{K} = (\mathbf{\Sigma}_\mathbf{X})^{1/2}$. Then $\tilde{\mathbf{X}} = \mathbf{K}^{-1}\mathbf{X}$ is such that $\mathbf{\Sigma}_{\tilde{\mathbf{X}}} = \mathbf{I}$. Therefore, if $t \in [0,1] \mapsto \boldsymbol{\theta}_t = (\mathbf{U}_t, \mathbf{W}_t)$ is the path given by Lemma 20 for the case $\mathbf{X} = \tilde{\mathbf{X}}$, the sought path (for $\mathbf{X} = \mathbf{X}$) is given by $t \in [0,1] \mapsto (\mathbf{U}_t, \mathbf{W}_t \mathbf{K}^{-1})$.
Case 2: $\mathrm{rk}(\mathbf{\Sigma}_\mathbf{X}) < n$. In this case, if $r = \mathrm{rk}(\mathbf{\Sigma}_\mathbf{X})$, $\mathbf{X}$ belongs to a r-dimensional subspace of $\mathbb{R}^n$ (a.s.), call it V. If $\mathbf{O} \in \mathbb{R}^{n \times r}$ is a matrix with an orthonormal basis of V as columns, then $\mathbf{O}\mathbf{O}^T\mathbf{X} = \mathbf{X}$ (a.s.), and, if $\tilde{\mathbf{X}} = \mathbf{O}^T\mathbf{X}$ then $\tilde{\mathbf{X}} \in \mathbb{R}^r$ and $\mathrm{rk}(\mathbf{\Sigma}_{\tilde{\mathbf{X}}}) = r$. Therefore, if $t \in [0,1] \mapsto \boldsymbol{\theta}_t = (\mathbf{U}_t, \mathbf{W}_t)$ is the path given by case 1 for $\mathbf{X} = \tilde{\mathbf{X}}$, the sought path (for $\mathbf{X} = \mathbf{X}$) is given by $t \in [0,1] \mapsto (\mathbf{U}_t, \mathbf{W}_t \mathbf{O}^T)$. ■

This concludes the proof of non-existence of spurious valleys for the square loss function of linear one-hidden-layer NNs $\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{U}\mathbf{W}\mathbf{x}$. The fact that such proof does not require any assumptions on the dimensions of the layers n, p, m neither on the rank of the initial layers, allows us to prove non-existence of spurious valleys for the square loss function of linear NNs of any depth $K \geq 1$:

$$\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{W}_{K+1}\cdots\mathbf{W}_1\mathbf{x} \tag{12}$$

We start by proving a simple lemma.

Lemma 22 *Let $\tilde{\mathbf{U}} = \tilde{\mathbf{M}}^1\cdots\tilde{\mathbf{M}}^n$, where $\tilde{\mathbf{U}} \in \mathbb{R}^{r_0\times r_n}$ and $\tilde{\mathbf{M}}^i \in \mathbb{R}^{r_{i-1}\times r_i}$. Suppose that $t \in [0,1] \mapsto \mathbf{U}_t$ is a given continuous path between $\mathbf{U}_0 = \tilde{\mathbf{U}}$ and another matrix $\mathbf{U}_1 \in \mathbb{R}^{r_0\times r_n}$. If $r_i \geq \min\{r_0, r_n\}$ for all i, then there exist continuous paths $\mathbf{M}_t^i$ such that $\mathbf{M}_0^i = \tilde{\mathbf{M}}^i$ and such that $\mathbf{U}_t = \mathbf{M}_t^1\ldots\mathbf{M}_t^n$.*

Proof The statement can be proved by induction. If $n = 1$ there is nothing to prove. Assume now (by induction) that it holds for all decompositions of $\mathbf{U}_0$ with size less than n. Let $r = r_h = \min_{i\in[n-1]} r_i$ and assume (w.l.o.g.) that $r_n = \min\{r_0, r_n\}$. We want to describe two paths $t \in [0,1] \mapsto \mathbf{V}_t \in \mathbb{R}^{r_0\times r}$, $t \in [0,1] \mapsto \mathbf{W}_t \in \mathbb{R}^{r\times r_n}$ such that $\mathbf{U}_t = \mathbf{V}_t\mathbf{W}_t$ and $\mathbf{V}_0 = \tilde{\mathbf{M}}^1\cdots\tilde{\mathbf{M}}^h$, $\mathbf{W}_0 = \tilde{\mathbf{M}}^{h+1}\cdots\tilde{\mathbf{M}}^n$. By operating as in step 1 in the proof of Theorem 8, we can assume $\mathrm{rk}(\mathbf{W}_0) = r_n$. Moreover (up to adding a linear path in $\mathbf{V}_t$) we can assume that $\mathbf{V}_0 = \mathbf{U}_0\mathbf{W}_0^\dagger$. We can then define $\mathbf{V}_t = \mathbf{U}_t\mathbf{W}_0^\dagger$ and $\mathbf{W}_t = \mathbf{W}_0$ for $t \in (0,1]$. We thus factorized $\mathbf{U}_t$ as $\mathbf{U}_t = \mathbf{V}_t\mathbf{W}_t$. By induction, we can assume that we can factorize $\mathbf{V}_t = \mathbf{M}_t^1\cdots\mathbf{M}_t^h$ and $\mathbf{W}_t = \mathbf{M}_t^{h+1}\cdots\mathbf{M}_t^n$. This concludes the proof. ■

We can now conclude the proof of Theorem 11.

Proof [Proof of Theorem 11] Consider a linear network $\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta})$ as in (12), where

$$\mathbf{W}_k \in \mathbb{R}^{p_k\times p_{k-1}} \quad \text{for } k \in [K+1]$$

We select $p_s = \min_{i\in[K]} p_k$. Then the network can be written as

$$\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \hat{\mathbf{W}}^2\hat{\mathbf{W}}^1\mathbf{x} \quad \text{where} \quad \hat{\mathbf{W}}^2 = \mathbf{W}^{K+1}\cdots\mathbf{W}^{s+1}, \quad \hat{\mathbf{W}}^1 = \mathbf{W}^s\cdots\mathbf{W}^1 \tag{13}$$

Now we want to prove property that given an initial parameter $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{W}}^{K+1},\ldots,\tilde{\mathbf{W}}^1)$, there exists a continuous path $\boldsymbol{\theta}_t = (\mathbf{W}_t^{K+1},\ldots,\mathbf{W}_t^1)$ such that $L(\boldsymbol{\theta}_t)$ is non-increasing and such that $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}$ and $L(\boldsymbol{\theta}_1) = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$. If we call $\hat{\tilde{\mathbf{W}}}^i$, $i = 1,2$, the matrices defined in (13) for $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$, then by Lemma 21 there exists a path $(\hat{\mathbf{W}}_t^2, \hat{\mathbf{W}}_t^1)$ satisfying the above. Thanks to Lemma 22, we can decompose

$$\hat{\mathbf{W}}_t^2 = \mathbf{W}_t^{K+1}\cdots\mathbf{W}_t^{s+1}, \quad \hat{\mathbf{W}}_t^1 = \mathbf{W}_t^s\cdots\mathbf{W}_t^1 \tag{14}$$

in a continuous way. Since p_s was to chosen as the minimum, it also holds that

$$\min_{\boldsymbol{\theta}=(\hat{\tilde{\mathbf{W}}}^2,\hat{\tilde{\mathbf{W}}}^1)} L(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}=(\mathbf{W}^{K+1},\ldots,\mathbf{W}^1)} L(\boldsymbol{\theta})$$

Therefore this is a suitable path and this concludes the proof of the theorem. ■

A.3. Proof of Theorem 12

Proof [Proof of Theorem 12] Let $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{u}}, \tilde{\mathbf{W}})$ be a starting parameter value. We aim to construct a continuous path $t \in [0,1] \mapsto \boldsymbol{\theta}_t \in \Theta$ starting in $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}$ and such that $L(\boldsymbol{\theta}_1) = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ and such that the function $t \in [0,1] \mapsto L(\boldsymbol{\theta}_t)$ is non-increasing. Such a path can be constructed in two steps.

Step 1. Let $\mathbf{A} = \sum_{k=1}^p \tilde{u}_k\tilde{\mathbf{w}}_k\tilde{\mathbf{w}}_k^T$ and $\sum_{k=1}^n \mathbf{u}_k^*\mathbf{w}_k^*(\mathbf{w}_k^*)^T$ be the SVD of $\mathbf{A}$. We define the parameters value $\boldsymbol{\theta}^* = (\mathbf{u}^*, \mathbf{W}^*)$ where $\mathbf{u}^* = (u_1^*,\ldots,u_n^*,0,\ldots,0)$ and $\mathbf{W}^*$ is the $p\times n$ matrix with rows $\mathbf{w}_i^*$ for $i \in [n]$ and $\mathbf{0}$ for $i \in [n+1,p]$. The first step consists in continuously mapping $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{u}}, \tilde{\mathbf{W}})$ to $\boldsymbol{\theta}^* = (\mathbf{u}^*, \mathbf{W}^*)$ with a path $\boldsymbol{\theta}_t$ such that $L(\boldsymbol{\theta}_t)$ is constant; the construction of such a path is detailed in Lemma 23.

Step 2. As noticed above, the network can be written as $\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{u}^T\sigma(\mathbf{W}\mathbf{x}) = \langle\mathbf{A},\mathbf{M}\rangle_F$, where $\mathbf{A} = \sum_{k=1}^p u_k\mathbf{w}_k\mathbf{w}_k^T$ and $\mathbf{M} = \mathbf{x}\mathbf{x}^T$. The square loss $L(\boldsymbol{\theta})$ is convex in the parameter $\mathbf{A}$. Be $\bar{\mathbf{A}}$ a minima of L as function of $\mathbf{A}$ and $\sum_{i=1}^n \bar{u}_k\bar{\mathbf{w}}_k\bar{\mathbf{w}}_k^T$ be the SVD of $\bar{\mathbf{A}}$; also let $\bar{\mathbf{u}} = (0,\ldots,0,\bar{u}_1,\ldots,\bar{u}_n)$ and $\bar{\mathbf{W}}$ be the $p\times n$ matrix with rows $\mathbf{0}$ for $i \in [p-n]$ and $\bar{\mathbf{w}}_i$ for $i \in [p-n+1,p]$. By the previous step we can assume that the initial parameter $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{u}}, \tilde{\mathbf{W}})$ is such that $\tilde{u}_i = 0$ and $\tilde{\mathbf{w}}_i = \mathbf{0}$ for $i \in [n+1,p]$. Then the path $\boldsymbol{\theta}_t = (1-t)(\mathbf{u},\mathbf{W}) + t(\bar{\mathbf{u}},\bar{\mathbf{W}})$ verifies property **P.1**. This indeed follows from the fact that $\mathbf{\Phi}(\mathbf{x};\boldsymbol{\theta}_t) = (1-t)\langle\mathbf{A},\mathbf{M}\rangle_F + t\langle\bar{\mathbf{A}},\mathbf{M}\rangle_F$ and from the convexity of the loss L as function of $\mathbf{A}$.

This shows that property **P.1** holds and so it concludes the proof of Theorem 12. ■

To conclude the proof we just need to prove the following lemmas.

Lemma 23 *Let $\boldsymbol{\theta} = (\mathbf{u},\mathbf{W})$ be an initial parameter and $\boldsymbol{\theta}^* = (\mathbf{u}^*,\mathbf{W}^*)$ be as in step 1 of the proof of Theorem 12. Then there exists a continuous path $\boldsymbol{\theta}_t$ from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^*$ such that the loss $L(\boldsymbol{\theta}_t)$ is constant (as a function of t).*

Proof Notice that we can assume $\mathbf{u} \in \{-1,0,1\}^p$. This can be done simply scaling (continuously) each row $\mathbf{w}_k$ of $\mathbf{W}$ by $\sqrt{|u_k|}$. Assume first that $\mathbf{u} \in \{\pm 1\}^p$. The general case ($u_k = 0$ for some k) is addressed in Remark 26. The sought path $\boldsymbol{\theta}_t$ can be constructed by iterating two steps (a finite amount of times). First we select a row $\mathbf{w}_k$ and construct a continuous path that maps this row to one of the $\mathbf{w}_i^*$; then we orthogonalize (w.r.t. such $\mathbf{w}_i^*$) the rest of rows $\mathbf{w}_j$, $j \neq k$. These two steps are constructed so that $\mathbf{A}$ never changes and therefore the loss is constant. The first step is described in Lemma 24, while the second is detailed in Lemma 25. At this point the parameter $\boldsymbol{\theta} = (\mathbf{u},\mathbf{W})$ verifies $u_i = u_i^*$, $\mathbf{w}_i = \mathbf{w}_i^*$ and $\mathbf{w}_j \in \langle\{\mathbf{w}_i^*\}\rangle^\perp$ for $j \neq k$. In particular it holds

$$\sum_{\substack{j=1\\ j\neq i}}^n u_j^*\mathbf{w}_j^*(\mathbf{w}_j^*)^T = \sum_{\substack{j=1\\ j\neq k}}^p u_k\mathbf{w}_k\mathbf{w}_k^T$$

Therefore, an induction step applied on the reduced parameter values

$$\mathbf{u}_{-k} = (u_1,\ldots,\widehat{u_k},\ldots,u_p)$$

and $\mathbf{W}_{-k} = [\mathbf{w}_1,\ldots,\widehat{\mathbf{w}_k},\ldots,\mathbf{w}_p]^T P$, where $\mathbf{P} = \sum_{j=1,j\neq i}^n \mathbf{w}_j^*\mathbf{e}_j^T \in \mathbb{R}^{n\times(n-1)}$, concludes the proof. The fact that the non-zero components of $\mathbf{u}$ and $\mathbf{W}$ coincide with the first n is not

necessary, but we can clearly assume it to hold w.l.o.g. ■

Lemma 24 *The first step described in the Proof of Lemma 23 can be performed when $p > 2n$.*

Proof Let $E_+ = \{k \in [p] : u_k = 1\}$, $E_- = \{k \in [p] : u_k = -1\}$ and $p_+ = |E_+|$, $p_- = |E_-|$. Accordingly we define

$$\mathbf{W}_+ = ([\mathbf{w}_k]_{k\in E_+})^T \in \mathbb{R}^{p_+\times n} \quad \text{and} \quad \mathbf{W}_- = ([\mathbf{w}_k]_{k\in E_-})^T \in \mathbb{R}^{p_-\times n}$$

Notice that then we can write

$$\mathbf{A} = \mathbf{W}_+^T\mathbf{W}_+ - \mathbf{W}_-^T\mathbf{W}_-$$

The main step of the proof is to observe that $\mathbf{A}$ (and therefore the loss) is invariant to the action of orthogonal matrices $\mathbf{Q}_+ \in SO(p_+)$ and $\mathbf{Q}_- \in SO(p_-)$. So, if $\mathbf{Q}_+(t)$ (resp. $\mathbf{Q}_-(t)$) is a continuous paths in $SO(p_+)$ (resp. in $SO(p_-)$) starting at the identity, acting on $\mathbf{W}$ as

$$\mathbf{W}_+(t) \doteq \mathbf{Q}_+(t)\mathbf{W}_+, \quad \mathbf{W}_-(t) \doteq \mathbf{Q}_-(t)\mathbf{W}_-$$

we have that

$$\mathbf{A} = \mathbf{W}_+(t)^T\mathbf{W}_+(t) - \mathbf{W}_-(t)^T\mathbf{W}_-(t)$$

is constant for all t. Now, since $p = p_+ + p_- > 2n$, it follows that either $p_+ > n$ or $p_- > n$. Assume w.l.o.g. that $p_+ > n$. Since $p_+ > n$, we can rotate the subspace generated by the columns of $\mathbf{W}_+$ so that its first row is $\mathbf{0}$. That is, there exist $\mathbf{h} \in \mathbb{R}^{p_+}$ non-zero such that $\mathbf{h}^T\mathbf{W}_+ = 0$ and $\|\mathbf{h}\| = 1$. It suffices to choose a path $\mathbf{Q}(t)$ in $SO(p_+)$ whose first row equals $\mathbf{h}$ at $t = 1$. It follows that $\mathbf{Q}(1)\mathbf{W}_+$ has a first row equal to $\mathbf{0}$. We then set the corresponding $u_1 = 0$, which does not change the loss, and finally set $\mathbf{w}_1$ to the desired eigenvector $\mathbf{w}_1^*$. ■

Lemma 25 *Assume that after the step in Lemma 24, the first row of $\mathbf{W}_+$ (resp. $\mathbf{W}_-$) is given by $\mathbf{w}_i^*$. Then we can map all the other rows of $\mathbf{W}$ to be orthogonal to $\mathbf{w}_i^*$, while keeping $\mathbf{A}$ constant.*

Proof To simplify the notation we assume (w.l.o.g.) that $\mathbf{w}_i^* = \mathbf{w}_1^*$ and that

$$\mathbf{W} = [\mathbf{w}_1^*, \mathbf{w}_2, \cdots, \mathbf{w}_p]^T$$

Now we want to construct a path

$$\mathbf{u}_t = (u_{1,t}, u_2, \ldots, u_p)$$
$$\mathbf{W}_t = [\mathbf{w}_1^*, \mathbf{w}_{2,t}, \cdots, \mathbf{w}_{p,t}]^T$$

such that $\mathbf{w}_{2,1}, \ldots, \mathbf{w}_{p,1} \in \langle\{\mathbf{w}_1^*\}\rangle^\perp$. To do this we simply take

$$\mathbf{w}_{k,t} \doteq \mathbf{w}_k - t\langle\mathbf{w}_1^*, \mathbf{w}_k\rangle\mathbf{w}_1^*$$

If $\mathbf{A}_t = \sum_{k=1}^p u_{k,t}\mathbf{w}_{k,t}\mathbf{w}_{k,t}^T$, we can show that there exists a choice of $u_{1,t}$ such that $\mathbf{A}_t = \mathbf{A}$ for all $t \in [0,1]$. It holds that

$$\begin{aligned}\mathbf{A}_t &= u_{1,t}\,\mathbf{w}_1^*(\mathbf{w}_1^*)^T \\ &\quad + \sum_{k=2}^p u_k\big[(1-t)^2(w_k^1)^2\mathbf{w}_1^*(\mathbf{w}_1^*)^T + (1-t)w_k^1\big(\tilde{\mathbf{w}}_k(\mathbf{w}_1^*)^T + \mathbf{w}_1^*\tilde{\mathbf{w}}_k^T\big) + \tilde{\mathbf{w}}_k\tilde{\mathbf{w}}_k^T\big]\end{aligned}$$

where $w_k^1 \doteq \langle\mathbf{w}_k, \mathbf{w}_1^*\rangle$ and $\tilde{\mathbf{w}}_k = \mathbf{w}_k - w_k^1\mathbf{w}_1^*$. In particular

$$\mathbf{A}_t = \mathbf{V}^*\left[\begin{array}{c|c} a_t & \mathbf{b}_t^T \\ \hline \mathbf{b}_t & \mathbf{A}_{2:n,2:n}\end{array}\right](\mathbf{V}^*)^T$$

where $\mathbf{V}^* = [\mathbf{w}_1^*, \cdots, \mathbf{w}_n^*] \in O(n)$. Since $\sum_{k=2}^p u_k w_k^1\,\tilde{\mathbf{w}}_k = 0$, it follows

$$\mathbf{b}_t = (1-t)\sum_{k=2}^p u_k w_k^1\,\tilde{\mathbf{w}}_k = 0 \quad \text{for all } t \in [0,1]$$

If we take

$$u_{1,t} = \lambda_1 - (1-t)^2\sum_{k=2}^p u_k(w_k^1)^2$$

it holds that

$$a_t = u_{1,t} + (1-t)^2\sum_{k=2}^p u_k(w_k^1)^2 = \lambda_1 \quad \text{for all } t \in [0,1]$$

Therefore, $\mathbf{A}_t = \mathbf{A}$ constant. This concludes the proof of the lemma. ■

Remark 26 *In the proof of Lemma 23, we assumed that (after rescaling) $\mathbf{u} \in \{\pm 1\}^p$. In general, it could be that $u_k = 0$ for some k. In this case we can first map the corresponding vectors $\mathbf{w}_k$ to $\mathbf{0}$ and the map such u_k to 1, without affecting the loss.*

Appendix B. Proofs of Section 4

Proof [Proof of Theorem 13] We consider here the case $m = 1$, but the same proof can be extended to the case $m > 1$. We start by properly choosing a r.v. $(\mathbf{X}, \mathbf{Y})$. Be $\bar{\mathbf{X}} \in \mathcal{R}_2(\sigma, n-1)$ a $(n-1)$ dimensional r.v. and $\bar{X}_n \in \mathcal{R}_2(\sigma, 1)$ a one dimensional r.v. We consider $\tilde{\mathbf{X}} = Z\bar{\mathbf{X}}$, $X_n = (1-Z)\bar{X}_n$ and $\mathbf{X} = (\tilde{\mathbf{X}}, X_n)$, where $Z \sim \mathrm{Ber}(1/2)$ and $\bar{\mathbf{X}}, \bar{X}_n, Z$ are independent. By hypothesis, $p \le 2^{-1}\underline{\dim}_*(\sigma, \tilde{\mathbf{X}})$. The proof is based on the fact that (for a proper choice of $\tilde{\mathbf{X}}$) this implies that $\overline{V_{\sigma,p-1}^+} \neq V_{\sigma,p}^+$, where we defined

$$V_{\sigma,p}^+ = \{\Phi(\cdot;\boldsymbol{\theta}) : \boldsymbol{\theta} \in [0,\infty)^p \times \mathbb{R}^{p\times n}\} \subseteq L^2_{\mathbf{X}}$$

(see the remark at the end of the proof). The r.v. Y is taken to be $Y = g_1(\mathbf{X}) - g_2(\mathbf{X})$, where $g_2 = \beta\psi_{\sigma,\mathbf{v}} \in V_{\sigma,1}^+$, $\beta > 0$, $\mathbf{v} = \mathbf{e}_n$, and $g_1 = \sum_{i=1}^p \alpha_i\psi_{\sigma,\mathbf{v}_i} \in V_{\sigma,p}^+$, $\boldsymbol{\alpha} \in (0,\infty)^p$, $\mathbf{v}_i \in \langle\{\mathbf{e}_n\}\rangle^\perp$, $i \in [p]$, is such that

$$\inf_{f\in V_{\sigma,p-1}^+} \mathbb{E}|f(\mathbf{X}) - g_1(\mathbf{X})|^2 = \epsilon > 0$$

We define

$$V_{\sigma,(p-1,1)} = \left\{ f = f_1 - f_2 : f_1 \in V^+_{\sigma,p-1}, f_2 \in V^+_{\sigma,1} \right\}$$

Notice that, for every path $\boldsymbol{\theta} : t \in [0,1] \mapsto \boldsymbol{\theta}_t \in \Theta$ such that $\Phi(\cdot;\boldsymbol{\theta}_0) \in V^+_{\sigma,p}$ and $\Phi(\cdot;\boldsymbol{\theta}_1) \in V_{\sigma,(p-1,1)}$, there exists $t_0 \in (0,1)$ such that $\Phi(\cdot;\boldsymbol{\theta}_{t_0}) \in V^+_{\sigma,p-1}$. Consider the *lifted* square loss function $L : V_{\sigma,p} \to [0,\infty)$ defined as

$$L(f) = \mathbb{E}|f(\mathbf{X}) - g(\mathbf{X})|^2 \qquad \text{for } f \in V_{\sigma,p}$$

We want to show that

$$L_{(p-1,0)} \doteq \min_{f \in V^+_{\sigma,p-1}} L(f) > L_{(p,0)} \doteq \min_{f \in V^+_{\sigma,p}} L(f) > L_{(p-1,1)} \doteq \min_{f \in V_{\sigma,(p-1,1)}} L(f)$$

It holds that

$$\begin{aligned} L_{(p-1,0)} = & \min_{f \in V^+_{\sigma,p-1}} \left\{ \mathbb{E}|f(\mathbf{X}) - g_1(\mathbf{X})|^2 \right\} + 2 \min_{f \in V^+_{\sigma,p-1}} \{\mathbb{E}[f(\mathbf{X}) g_2(\mathbf{X})]\} \\ & + \mathbb{E}|g_2(\mathbf{X})|^2 - C\sigma(0) \\ & \geq \epsilon + L_{(p,0)} - C\sigma(0) \end{aligned}$$

where $C = \mathbb{E}[g_1(\tilde{\mathbf{X}})] + \mathbb{E}[g_2(X_n)]$,and that

$$\begin{aligned} L_{(p,0)} = & \min_{f \in V^+_{\sigma,p}} \left\{ \mathbb{E}|f(\mathbf{X}) - g_1(\mathbf{X})|^2 \right\} + 2 \min_{f \in V^+_{\sigma,p}} \{\mathbb{E}[f(\mathbf{X}) g_2(\mathbf{X})]\} \\ & + \mathbb{E}|g_2(\mathbf{X})|^2 - C\sigma(0) \\ & \geq \beta^2 \, \mathbb{E}|\psi_{\sigma,\mathbf{v}}(X_n)|^2 - C\sigma(0) \end{aligned}$$

Finally, it holds that

$$L_{(p-1,1)} \leq \min_{i \in [1,p]} \alpha_i^2 \, \mathbb{E}|\psi_{\sigma,\mathbf{v}_i}(\mathbf{X})|^2$$

Given $M > 0$, up to multiply g_1 by a positive constant, it holds that

$$\begin{aligned} \epsilon & \geq M + C\sigma(0) \\ \beta^2 & \geq \frac{M + C\sigma(0) + \min_{i \in [1,p]} \alpha_i^2 \, \mathbb{E}|\psi_{\sigma,\mathbf{v}_i}(\mathbf{X})|^2}{\mathbb{E}|\psi_{\sigma,\mathbf{v}}(X_n)|^2} \end{aligned}$$

To finish the proof, consider $\mathcal{U} = \{\boldsymbol{\theta} = (\mathbf{u}, \mathbf{W}) \in \Theta : \mathbf{u} \in (0,\infty)^p\}$ and $\boldsymbol{\theta}^* \in \mathcal{U}$ such that

$$L(\boldsymbol{\theta}^*) = \min_{\boldsymbol{\theta} \in \mathcal{U}} L(\boldsymbol{\theta})$$

Then, (by continuity of L) there exists a neighborhood $\boldsymbol{\theta}^* \in \Omega \subset \mathcal{U}$ such that $\sup_{\boldsymbol{\theta} \in \Omega} L(\boldsymbol{\theta}) \leq L(\boldsymbol{\theta}^*) + M/2$. The set Ω then verifies the statement of the theorem.

■

Remark 27 *In the proof of Theorem 13 we used the fact that, if $p \geq 1$ verifies $p \leq \frac{1}{2}\dim_*(\sigma,n)$, then there exist $\mathbf{X} \in \mathcal{R}_2(\sigma,n)$ such that $\overline{V^+_{\sigma,p-1}} \neq V^+_{\sigma,p}$ (in the $L^2_{\mathbf{X}}$ metric). Assume $\mathbf{X}$ is a n-dimensional standard Gaussian variable. Consider first the case $\sigma(z) = z^k$. If $k = 2$, then*

$$V^+_{\sigma,p} \simeq \{\mathbf{M} \in \mathrm{S}^2(\mathbb{R}^n) : \mathbf{M} \textit{ is PSD}\}$$

In particular, this implies that $V^+_{\sigma,p-1}$ is not dense in $V^+_{\sigma,p}$ if $p \leq n = \dim_*(\sigma,n)$ (which justifies the statement of Corollary 14). If $k > 2$, let $p \leq \frac{1}{2}\dim_*(\sigma,n)$ and assume that $\overline{V^+_{\sigma,p-1}} = V^+_{\sigma,p}$. This implies that every tensor $\mathbf{T} = \sum_{i=1}^p \mathbf{v}_i^{\otimes k}$ can be approximated up to any accuracy by $\mathbf{T} = \sum_{i=1}^{p-1} \tilde{\mathbf{v}}_i^{\otimes k}$ for some $\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_{p-1} \in \mathbb{R}^n$. But this also implies that every tensor $\mathbf{T} \in \mathbb{S}^k(\mathbb{R}^n)$ has border rank less or equal than $2\big(\frac{1}{2}\dim_*(\sigma,n) - 1\big) = \mathrm{rk}_{\mathrm{S}}(k,n) - 2$, which contradicts the definition of $\mathrm{rk}_{\mathrm{S}}(k,n)$. For non-polynomial σ, we can get the same result, by using the decomposition (18) and proceeding as above.

Appendix C. Proof of Theorem 16

Proof If we denote by $d\mu$ the probability distribution of $\mathbf{X}$, the continuous function

$$\psi : (\mathbf{w}, \mathbf{x}) \in \mathbb{S}^n \times \mathbb{R}^n \mapsto \psi_{\mathbf{w}}(\mathbf{x}) = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$$

belongs to $L^2(\mathbb{S}^n \times \mathbb{R}^n, d\tau \otimes d\mu)$. We consider the kernel associated with the neural network architecture

$$k(\mathbf{x}, \mathbf{y}) = \int_{\mathcal{W}} \psi_{\mathbf{w}}(\mathbf{x}) \psi_{\mathbf{w}}(\mathbf{y}) \, d\tau(\mathbf{w}) \tag{15}$$

The above defines a continuous symmetric, positive semi-definite kernel k, along with $\mathbb{H}$, the RKHS associated, and the integral operator $\Sigma : L^2(\mathbb{R}^n, d\mu) \to \mathbb{H} \subseteq L^2(\mathbb{R}^n, d\mu)$ defined as

$$f \mapsto \left(\Sigma f : \mathbf{x} \mapsto \int_{\mathbb{R}^n} f(\mathbf{y}) k(\mathbf{x}, \mathbf{y}) \, d\mu(\mathbf{y}) \right)$$

The operator Σ admits a spectral decomposition in $L^2(\mathbb{R}^n, d\mu)$: $\Sigma e_k = \lambda_k e_k$ for an orthonormal basis $\{e_k\}_{k \geq 1}$ of $L^2(\mathbb{R}^n, d\mu)$ and non-increasing sequence of non-negative eigenvalues $\{\lambda_k\}_{k \geq 1}$. Moreover the RKHS $\mathbb{H}$ is dense in $L^2(\mathbb{R}^n, d\mu)$ (see Lemma 30), which is equivalent to have $\lambda_k > 0$ for all $k \geq 1$. The expectation in (15) provides a singular value decomposition for Σ in terms of functions in $L^2(\mathbb{S}^n, d\tau)$. Indeed, given $g \in L^2(\mathbb{S}^n, d\tau)$, the linear operator $T : L^2(\mathbb{S}^n, d\tau) \to L^2(\mathbb{R}^n, d\mu)$ defined as

$$g \mapsto \left(Tg : \mathbf{x} \mapsto \int_{\mathbb{S}^n} g(\mathbf{w}) \psi_{\mathbf{w}}(\mathbf{x}) \, d\tau(\mathbf{w}) \right)$$

satisfies $\Sigma = TT^*$. It follows that there exists an orthonormal basis of $L^2(\mathbb{S}^n, d\tau)$, $\{f_k\}_{k \geq 1}$ such that $Tf_k = \lambda_k^{1/2} e_k$ and therefore $\psi_{\mathbf{w}} = \sum_{k=1}^\infty \lambda_k^{1/2} f_k(\mathbf{w}) e_k$. Finally, it can be shown (Bach, 2017a) that in fact $\mathbb{H} = \mathrm{Im}(T)$, and thus $\mathbb{H}$ consists of functions f that can be written, for some $g \in L^2(\mathbb{S}^n, d\tau)$ as

$$f(\mathbf{x}) = \int_{\mathbb{S}^n} g(\mathbf{w}) \psi_{\mathbf{w}}(\mathbf{x}) d\tau(\mathbf{w}) = \langle g, \psi(\cdot, \mathbf{x}) \rangle_{L^2(\mathbb{S}^n, d\tau)} \quad \text{for } \mathbf{x} \in \mathbb{R}^n$$

For an account of these properties, we refer to Bach (Bach, 2017b). Thanks to the density of $\mathbb{H}$ in $L^2(\mathbb{R}^n, d\mu)$, we can assume, without loss of generality, that

$$f^*(\mathbf{x}) = \int_{\mathbb{S}^n} g^*(\mathbf{w})\psi_{\mathbf{w}}(\mathbf{x})d\tau(\mathbf{w})$$

for some $g^* \in L^2(\mathbb{S}^n, d\tau)$. Now, given an initial set of first layer weights $\mathbf{w}_1, \dots, \mathbf{w}_p \in \mathbb{S}^n$ sampled i.i.d. from $d\tau$, and $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_p]^T$, we define the empirical kernel

$$k_{\mathbf{W}}(\mathbf{x}, \mathbf{y}) = \frac{1}{p}\sum_{i=1}^{p} \sigma(\langle \mathbf{x}, \mathbf{w}_i\rangle)\sigma(\langle \mathbf{y}, \mathbf{w}_i\rangle)$$

which in turn defines an empirical RKHS $\mathbb{H}_{\mathbf{W}}$. Keeping the first layer weights fixed and optimizing the output layer weights thus gives us the ability to find a function $f^*_{\mathbf{W}} \in \mathbb{H}_{\mathbf{W}}$ that best approximates f^*:

$$\|f^*_{\mathbf{W}} - f^*\|_{L^2(\mathbb{R}^n, d\mu)} = \min_{f \in \mathbb{H}_{\mathbf{W}}} \|f - f^*\|_{L^2(\mathbb{R}^n, d\mu)} \doteq R(\mathbf{W})$$

Given an initial parameter parameter value $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{u}}, \tilde{\mathbf{W}})$ (here we incorporated $\tilde{\mathbf{b}}$ in $\tilde{\mathbf{W}}$) as in the statement, consider the path

$$\boldsymbol{\theta}_t = (t\mathbf{q}(\tilde{\mathbf{W}}) + (1-t)\tilde{\mathbf{u}}, \tilde{\mathbf{W}}) \quad \text{where} \quad \mathbf{q}(\tilde{\mathbf{W}}) = \arg\min_{\mathbf{u} \in \mathbb{R}^p} L(\boldsymbol{\theta})|_{\boldsymbol{\theta} = (\mathbf{u}, \tilde{\mathbf{W}})}$$

By convexity of L, the function $t \in [0,1] \mapsto L(\boldsymbol{\theta}_t)$ is non-increasing and it holds that

$$L(\boldsymbol{\theta}_1) \leq \mathcal{R}(\mathbf{X}, Y) + R(\tilde{\mathbf{W}})$$

Applying Proposition 1 from Bach (Bach, 2017b), it holds that

$$R(\mathbf{W}) \leq 4\lambda \quad \text{if} \quad p \geq 5d(\lambda)\log(16d(\lambda)/\delta)$$

with probability greater or equal than $1-\delta$, where

$$\begin{aligned} d(\lambda) &= \max_{\mathbf{w} \in \mathbb{S}^n} \mathbb{E}\big[\varphi_{\mathbf{w}}(\mathbf{X})((\Sigma + \lambda I)^{-1}\psi_{\mathbf{w}})(\mathbf{X})\big] \\ &= \max_{\mathbf{w} \in \mathbb{S}^n} \sum_{k=1}^{\infty} \frac{\lambda_k}{\lambda_k + \lambda} f_k(\mathbf{w})^2 \leq \lambda^{-1} \max_{\mathbf{w} \in \mathbb{S}^n} \sum_{k=1}^{\infty} \lambda_k f_k(\mathbf{w})^2 = \lambda^{-1} \max_{\mathbf{w} \in \mathbb{S}^n} \|\psi_{\mathbf{w}}\|^2_{L^2(\mathbb{R}^n, d\mu)} \end{aligned}$$

This shows part 1 of the statement. To prove part 2, notice that $f^*_{\mathbf{W}} = \Phi(\cdot; \boldsymbol{\theta})$ with $\boldsymbol{\theta} = (\mathbf{u}^*_{\mathbf{W}}, \mathbf{W})$ for some $\mathbf{u}^*_{\mathbf{W}} \in \mathbb{R}^p$. By taking $u_k = \frac{1}{p} g^*(\mathbf{w}_k)$ for $k \in [p]$ and denoting $Z(\mathbf{w}) := g^*(\mathbf{w})\psi_{\mathbf{w}}$ and by Z the r.v. $Z = Z(\mathbf{v})$, for $\mathbf{v} \sim d\tau$, with values in $L^2(\mathbb{R}^n, d\mu)$, it holds

$$R(\mathbf{W}) \leq \Big\| \frac{1}{p}\sum_{k=1}^{p} Z(\mathbf{w}_k) - \mathbb{E}_\tau[Z] \Big\|_{L^2(\mathbb{R}^n, d\mu)} \tag{16}$$

Note that $C \doteq \sup_{\mathbf{w} \in \mathbb{S}^n} \|Z(\mathbf{w})\|_{L^2(\mathbb{R}^n, d\mu)} \leq \|g^*\|_{L^\infty(\mathbb{S}^n, d\tau)} \max_{\mathbf{w} \in \mathbb{S}^n} \|\psi_{\mathbf{w}}\|_{L^2(\mathbb{R}^n, d\mu)} < \infty$ if $\|g^*\|_{L^\infty(\mathbb{S}^n, d\tau)} < \infty$. Then, applying Lemma 29 to the bound (16), we get that

$$\mathbb{P}_\tau\{R(\mathbf{W}) \leq \varepsilon\} \geq 1 - \exp\Big\{ -\Big(\varepsilon - \sqrt{v(p)}\Big)^2 / (2v(p)) \Big\}$$

for every $\varepsilon \geq \sqrt{v(p)}$, with $v(p) = C^2/p$. The result follows by taking $\varepsilon = v(p)^{1/2} + p^{\delta/2 - 1/2}$. ■

Appendix D. Proofs of Section 2.2

Proof [Proof of Lemma 4] If σ is a polynomial of any degree d, then it holds that $\dim^*(\sigma, n) < \infty$. Indeed, let $\sigma(z) = a_0 + a_1 z + \cdots + a_d z^d$, for some $a_k \in \mathbb{R}$. If $I = \{k \in [0,d] : a_k \neq 0\}$, then

$$V_\sigma \subseteq \mathbb{R}_I[\mathbf{x}] \doteq \{\mathbf{x} \mapsto \sum_{k \in I} \sum_{|\boldsymbol{\beta}| = k} \alpha_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}} : \alpha_{\boldsymbol{\beta}} \in \mathbb{R}\}$$

It follows that

$$\dim^*(\sigma, n) = \dim(V_\sigma) \leq \dim(\mathbb{R}_I[\mathbf{x}]) = \sum_{k=0}^{d} \binom{n+k-1}{k} \mathbf{1}_{\{a_k \neq 0\}} = O(n^d)$$

This proves one implication. We prove the other one by contradiction. Assume now that σ is not a polynomial and that $\dim(V_\sigma) = q < \infty$. Thanks to Theorem 1 in Leshno et al. (Leshno et al., 1993), for every continuous function $g : \mathbb{R}^n \to \mathbb{R}$, any compact set $K \subset \mathbb{R}^n$, and any $\varepsilon > 0$ there exist $h \in V_\sigma$ such that

$$\sup_{\mathbf{x} \in K} |h(\mathbf{x}) - g(\mathbf{x})| < \varepsilon \tag{17}$$

Now, let $g : \mathbb{R}^n \to \mathbb{R}$ be a continuous function supported on a compact set $C \subset \mathbb{R}^n$. We call $C_c(\mathbb{R}^n)$ the set of the real-valued continuous functions from $\mathbb{R}^n$ with compact support. Thanks to (17), we can find a sequence of compact sets $\{K_m\}_{m \geq 1}$ of $\mathbb{R}^n$ such that

$$C \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K_m \subseteq \cdots \subseteq \cup_{m=1}^{\infty} K_m = \mathbb{R}^n$$

and a sequence of functions $\{h_m\}_{m \geq 1} \subset V_\sigma$ such that

$$\|g - h_m \mathbb{1}_{K_m}\|_{L^2_{\mathbf{X}}} = \|(g - h_m)\mathbb{1}_{K_m}\|_{L^2_{\mathbf{X}}} < 2^{-m}$$

In particular this implies that

$$\|h_n \mathbb{1}_{K_n} - h_m \mathbb{1}_{K_m}\|_{L^2_{\mathbf{X}}} < 2^{1 - \min\{n,m\}} \to 0$$

as $n, m \to \infty$, i.e. $\{h_m \mathbb{1}_{K_m}\}_{m \geq 1}$ is a Cauchy sequence in $L^2_{\mathbf{X}}$ and therefore it admits a limit $\lim_{n \to \infty} h_m \mathbb{1}_{K_m} = g \in L^2_{\mathbf{X}}$. Since $\dim(V_\sigma) = q < \infty$, there exists $\mathbf{w}_1, \dots, \mathbf{w}_q \in \mathbb{R}^n$ such that every $h \in V_\sigma$ can be written as

$$h(\mathbf{x}) = \langle \mathbf{u}, \boldsymbol{\gamma}(\mathbf{x})\rangle$$

for some $\mathbf{u} \in \mathbb{R}^q$, where $\boldsymbol{\gamma}(\mathbf{x}) = (\sigma(\langle \mathbf{w}_1, \mathbf{x}\rangle), \dots, \sigma(\langle \mathbf{w}_q, \mathbf{x}\rangle))$. Let $\{\mathbf{u}_m\}_{m \geq 1} \subset \mathbb{R}^q$ such that $h_m(\mathbf{x}) = \langle \mathbf{u}_m, \boldsymbol{\gamma}(\mathbf{x})\rangle$. Thanks to the above calculations, we know that the sequence $\{\|h_m \mathbb{1}_K\|_{L^2_{\mathbf{X}}}\}_{m \geq 1}$ is bounded for any arbitrary compact set $K \subseteq \mathbb{R}^n$. Since

$$\|h_m \mathbb{1}_K\|^2_{L^2_{\mathbf{X}}} = \mathbf{u}_m^T M \mathbf{u}_m$$

where $M = \mathbb{E}\big[\boldsymbol{\gamma}(\mathbf{X})\boldsymbol{\gamma}(\mathbf{X})^T \mathbb{1}_{\{\mathbf{X} \in K\}}\big] \in \mathbb{R}^{q \times q}$, this implies that the sequence $\{\mathbf{u}_m\}_{m \geq 1}$ is bounded (unless $g = 0$). Therefore (up to extracting a sub-sequence) we can assume that it has a limit $\mathbf{u} \in \mathbb{R}^q$. If we call $h \in V_\sigma$ the function defined as $h(\mathbf{x}) = \langle \mathbf{u}, \boldsymbol{\gamma}(\mathbf{x})\rangle$, it is easy to check (from the above calculations) that $h = g$ in $L^2_{\mathbf{X}}$. This shows that $C_c(\mathbb{R}^n) \subseteq V_\sigma$, which

in turn implies that V_σ is dense in $L^2_{\mathbf{X}}$ (since $C_c(\mathbb{R}^n)$ is dense in $L^2_{\mathbf{X}}$). But this is impossible, since $\dim(V_\sigma) = q < \infty = \dim(L^2_{\mathbf{X}})$. Therefore, it must hold $\dim(V_\sigma) = \infty$. ■

Proof [Proof of Lemma 6] Let $\boldsymbol{\theta} = (\mathbf{u}, \mathbf{W}) \in [0,\infty)^p \times \mathbb{R}^{p\times n}$. For every $\mathbf{x} \in \mathbb{R}^n$ it holds

$$\Phi(\mathbf{x};\boldsymbol{\theta}) = \sum_{i=1}^p u_i(\langle \mathbf{w}_i, \mathbf{x}\rangle)^k = \sum_{i=1}^p u_i \langle \mathbf{w}_i^{\otimes k}, \mathbf{x}^{\otimes k}\rangle_F = \Big\langle \sum_{i=1}^p u_i \mathbf{w}_i^{\otimes k}, \mathbf{x}^{\otimes k}\Big\rangle_F$$

For any $p \geq 1$ and $(\mathbf{u}, \mathbf{W}) \in [0,\infty)^p \times \mathbb{R}^{p\times n}$, $\sum_{i=1}^p u_i \mathbf{w}_i^{\otimes k} \in S^k(\mathbb{R}^n)$. By definition of $\mathrm{rk}_{\mathrm{S}}(k,n)$, it follows that there exists $q \leq \mathrm{rk}_{\mathrm{S}}(k,n)$ and $\tilde{\boldsymbol{\theta}} = (\tilde{\mathbf{u}}, \tilde{\mathbf{W}}) \in [0,\infty)^q \times \mathbb{R}^{q\times n}$ such that

$$\sum_{i=1}^p u_i \mathbf{w}_i^{\otimes k} = \sum_{i=1}^q \tilde{u}_i \tilde{\mathbf{w}}_i^{\otimes k} \quad \Rightarrow \quad \Phi(\cdot;\boldsymbol{\theta}) = \Phi(\cdot;\tilde{\boldsymbol{\theta}})$$

By definition of $\dim_*(\sigma,n)$, this implies that $\dim_*(\sigma,n) \leq \mathrm{rk}_{\mathrm{S}}(k,n)$. The equality follows by choosing $(\mathbf{u},\mathbf{W}) \in [0,\infty)^p \times \mathbb{R}^{p\times p}$ such that $\mathrm{rk}_{\mathrm{S}}(\sum_{i=1}^p u_i \mathbf{w}_i^{\otimes k}) = \mathrm{rk}_{\mathrm{S}}(k,n)$. ■

Proof [Proof of Lemma 7] If σ is polynomial, then one implication follows by Lemma 4. Now, assume that $\sigma \in L^2(\mathbb{R}, e^{-x^2/2}\,dx)$ is a continuous non-polynomial activation and let $\mathbf{X} \sim N(\mathbf{0}, \mathbf{I})$ be a r.v. in $\mathcal{R}_2(\sigma,n)$. Then, we can write $\sigma(z) = \sum_{k=0}^\infty \hat{\sigma}_k h_k(z)$, where h_k is the k-th Hermite polynomial. It follows that, for $\boldsymbol{\theta} = (\mathbf{u},\mathbf{W})$,

$$\mathbb{E}|\Phi(\mathbf{X};\boldsymbol{\theta})|^2 = \sum_{k=1}^\infty \hat{\sigma}_k^2 \Big\| \sum_{i=1}^p u_i \mathbf{w}_i^{\otimes k}\Big\|_F^2 \tag{18}$$

(see Lemma 1 from (Mondelli and Montanari, 2018)). Since σ is not polynomial and $n > 1$, $V_{\sigma,p} \neq V_{\sigma,p+1}$, where

$$V_{\sigma,p} \doteq \Big\{ \mathbf{x} \mapsto \sum_{k=1}^p u_k \sigma(\langle \mathbf{w}_k, \mathbf{x}\rangle) : (\mathbf{u},\mathbf{W}) \in \mathbb{R}^p \times \mathbb{R}^{p\times n}\Big\}$$

Indeed, if $V_{\sigma,p} = V_{\sigma,p+1}$, then $V_{\sigma,p} = V_{\sigma,q}$ for every $q > p$. Let k be a positive integer such that $\hat{\sigma}_k \neq 0$ and such that $\mathrm{rk}_{\mathrm{S}}(k,n) > p$. Let $\mathbf{G} = \sum_{i=1}^q \alpha_i \mathbf{v}_i^{\otimes k}$ a symmetric tensor with $\mathrm{rk}_{\mathrm{S}}(\mathbf{G}) = q = \mathrm{rk}_{\mathrm{S}}(k,n)$ and $g = \sum_{i=1}^q \alpha_i \psi_{\sigma,\mathbf{v}_i}$. If $g \in V_{\sigma,p}$, then there exists $\boldsymbol{\theta} = (\mathbf{u},\mathbf{W}) \in \Theta = \mathbb{R}^p \times \mathbb{R}^{p\times n}$ such that

$$0 = \mathbb{E}|\Phi(\mathbf{X};\boldsymbol{\theta}) - g(\mathbf{X})|^2 = \sum_{k=1}^\infty \hat{\sigma}_k^2 \Big\| \sum_{i=1}^p u_i \mathbf{w}_i^{\otimes k} - \sum_{i=1}^q \alpha_i \mathbf{v}_i^{\otimes k}\Big\|_F^2$$

But this would imply that $\mathrm{rk}_{\mathrm{S}}(\mathbf{G}) \leq p$, which is a contradiction. This concludes the proof. ■

Appendix E. Proofs of Additional Lemmas

Lemma 2 *Be $\boldsymbol{\theta} \mapsto L(\boldsymbol{\theta})$ a continuous function. Then, property **P.1** implies absence of spurious valleys. In particular, this implies absence of strict spurious minima, and of (generally non-strict) spurious minima if property **P.1** holds with strictly decreasing paths $t \mapsto L(\boldsymbol{\theta}_t)$. Conversely, presence of spurious valleys implies existence of spurious minima.*

Proof Assume that property **P.1** holds. Consider any value $c > 0$ such that $\Omega_L(c)$ is non-empty and let $\mathcal{U}$ be a connected component of $\Omega_L(c)$. Given a point $\boldsymbol{\theta} \in \mathcal{U}$ there exists a path from $\boldsymbol{\theta}$ satisfying property **P.1**. This means that $\mathcal{U}$ contains a global minima, and therefore it can not be a spurious valley. Similarly, assume that property **P.1** holds with strictly decreasing paths and that the function L admits a strict local minima. This means that there exists a point $\boldsymbol{\theta}_0$ such that $\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) < L(\boldsymbol{\theta}_0) < L(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$ in $B_\epsilon(\boldsymbol{\theta})$, for some $\epsilon > 0$. But this implies that for any path $t \in [0,1] \mapsto \boldsymbol{\theta}_t$ if holds $L(\boldsymbol{\theta}_t) > L(\boldsymbol{\theta}_0)$ for some $t > 0$ sufficiently small, a contradiction. To see the last point, assume that there exist spurious valleys and consider $\mathcal{U}$ a connected component of $\Omega_L(c)$ for some $c > 0$. Then $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ is a spurious minima. ■

Lemma 28 *Consider the optimization problem*

$$\underset{\mathbf{W}\in\mathbb{R}^{m\times n}}{\arg\min}\ \ell(\mathbf{W}) \quad \textit{where} \quad \ell(\mathbf{W}) = \mathbb{E}\|\mathbf{W}\mathbf{X} - \mathbf{Y}\|^2 \tag{19}$$

for two square integrable r.v.'s $\mathbf{X}$ and $\mathbf{Y}$ with values in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively. Then one solution to (19) is given by

$$\mathbf{W} = \boldsymbol{\Sigma}_{\mathbf{YX}}\boldsymbol{\Sigma}_{\mathbf{X}}^\dagger \tag{20}$$

Similarly, one solution to the optimization problem

$$\underset{\mathbf{U}\in\mathbb{R}^{m\times p}}{\arg\min}\ \ell(\mathbf{U};\mathbf{W}) \quad \textit{where} \quad \ell(\mathbf{U};\mathbf{W}) = \mathbb{E}\|\mathbf{U}\mathbf{W}\mathbf{X} - \mathbf{Y}\|^2$$

for any $\mathbf{W} \in \mathbb{R}^{p\times n}$ is given by

$$\mathbf{U} = \mathbf{q}(\mathbf{W}) \doteq \boldsymbol{\Sigma}_{\mathbf{YX}}\mathbf{W}^T(\mathbf{W}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{W}^T)^\dagger \tag{21}$$

Assuming $\boldsymbol{\Sigma}_{\mathbf{X}}$ invertible, the minimal value obtained by $\ell(\mathbf{U};\mathbf{W})$ is given by

$$\ell(\mathbf{q}(\mathbf{W});\mathbf{W}) = \mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{Y}}) - \mathrm{tr}((\mathbf{W}\mathbf{K})^\dagger(\mathbf{W}\mathbf{K})\mathbf{M}) \tag{22}$$

where $\mathbf{K} = (\boldsymbol{\Sigma}_{\mathbf{X}})^{1/2}$ and $\mathbf{M} = \mathbf{K}^{-1}\boldsymbol{\Sigma}_{\mathbf{XY}}\boldsymbol{\Sigma}_{\mathbf{YX}}\mathbf{K}^{-1}$. If $\mathbf{M} = \sum_{i=1}^n \lambda_i \mathbf{v}_i\mathbf{v}_i^T$ is the SVD of $\mathbf{M}$, the quantity (22) is minimized over $\mathbf{W}$ for $(\mathbf{W}\mathbf{K})^\dagger(\mathbf{W}\mathbf{K}) = \sum_{i=1}^{p\wedge n} \mathbf{v}_i\mathbf{v}_i^T$.

Proof The first part of the lemma can be shown by writing problem (19) as

$$\underset{\mathbf{W}\in\mathbb{R}^{m\times n}}{\arg\min}\ \ell(\mathbf{W}) \quad \text{where} \quad \ell(\mathbf{W}) = \mathrm{tr}(\mathbf{W}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{W}^T) - 2\,\mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{YX}}\mathbf{W}^T) \tag{23}$$

and by taking $\mathbf{W}$ as a stationary point of the above $\ell(\mathbf{W})$. Using this fact, one minima of the function $\ell(\mathbf{U};\mathbf{W})$ is given by

$$\mathbf{U} = \boldsymbol{\Sigma}_{\mathbf{YXW}}(\boldsymbol{\Sigma}_{\mathbf{W}}\mathbf{X})^\dagger = \boldsymbol{\Sigma}_{\mathbf{YX}}\mathbf{W}^T(\mathbf{W}\boldsymbol{\Sigma}_{\mathbf{X}}\mathbf{W}^T)^\dagger$$

Now assume that $\mathbf{\Sigma_X}$ is invertible; let $\mathbf{K} = (\mathbf{\Sigma_X})^{1/2}$ and $\mathbf{M} = \mathbf{K}^{-1}\mathbf{\Sigma_{XY}}\mathbf{\Sigma_{YX}}\mathbf{K}^{-1}$. Then it holds

$$\begin{aligned}\ell(\mathbf{q}(\mathbf{W});\mathbf{W}) &= \operatorname{tr}(\mathbf{q}(\mathbf{W})\mathbf{W}\mathbf{\Sigma_X}\mathbf{W}^T\mathbf{q}(\mathbf{W})^T) - 2\operatorname{tr}(\mathbf{\Sigma_{YX}}\mathbf{W}^T\mathbf{q}(\mathbf{W})^T) + \operatorname{tr}(\mathbf{\Sigma_Y})\\ &= \operatorname{tr}(\mathbf{\Sigma_{YX}}\mathbf{W}^T(\mathbf{W}\mathbf{\Sigma_X}\mathbf{W}^T)^\dagger\mathbf{W}\mathbf{\Sigma_X}\mathbf{W}^T(\mathbf{W}\mathbf{\Sigma_X}\mathbf{W}^T)^\dagger\mathbf{W}\mathbf{\Sigma_{XY}})\\ &\quad - 2\operatorname{tr}(\mathbf{\Sigma_{YX}}\mathbf{W}^T(\mathbf{W}\mathbf{\Sigma_X}\mathbf{W}^T)^\dagger\mathbf{W}\mathbf{\Sigma_{XY}}) + \operatorname{tr}(\mathbf{\Sigma_Y})\\ &= -\operatorname{tr}(\mathbf{\Sigma_{YX}}\mathbf{W}^T(\mathbf{W}\mathbf{\Sigma_X}\mathbf{W}^T)^\dagger\mathbf{W}\mathbf{\Sigma_{XY}}) + \operatorname{tr}(\mathbf{\Sigma_Y})\\ &= -\operatorname{tr}(\mathbf{M}(\mathbf{WK})^T((\mathbf{WK})(\mathbf{WK})^T)^\dagger(\mathbf{WK})) + \operatorname{tr}(\mathbf{\Sigma_Y})\\ &= \operatorname{tr}(\mathbf{\Sigma_Y}) - \operatorname{tr}((\mathbf{WK})^\dagger(\mathbf{WK})\mathbf{M})\end{aligned}$$

Finally, we notice that the matrix $(\mathbf{WK})^\dagger(\mathbf{WK})$ is the orthogonal projection on the space spanned by the rows of $\mathbf{WK}$, which we denote by $\mathbf{P_{WK}}$. In particular $\mathbf{P_{WK}}$ has the form $\mathbf{P_{WK}} = \sum_{i=1}^r \mathbf{w}_i\mathbf{w}_i^T$ for some $\{\mathbf{w}_1,\dots,\mathbf{w}_r\} \subset \mathbb{R}^n$ orthonormal vectors and $r \le p \wedge n$. Therefore, minimize $\ell(\mathbf{q}(\mathbf{W});\mathbf{W})$ over $\mathbf{W}$ it is equivalent to maximize the quantity

$$\sum_{i=1}^r \mathbf{w}_i^T\mathbf{M}\mathbf{w}_i$$

over the sets of $\mathbf{w}_1,\dots,\mathbf{w}_r$ orthonormal vectors of $\mathbb{R}^n$, $r \le p \wedge n$. Clearly, this is for $\mathbf{w}_1 = \mathbf{v}_1,\dots,\mathbf{w}_{p\wedge n} = \mathbf{v}_{p\wedge n}$. This concludes the proof of the lemma. ■

Lemma 29 *Let $X_1,\dots,X_n$ be independent zero-mean r.v.'s taking values in a separable Hilbert space such that $\|X_i\| \le c_i$ with probability one and denote $v = \sum_{i=1}^n c_i^2$. Then, for all $t \ge v$, it holds*

$$\mathbb{P}\left\{\left\|\sum_{i=1}^n X_i\right\| > t\right\} \le e^{-(t-\sqrt{v})^2/(2v)}$$

Proof The proof can be found in (Boucheron et al., 2013), Example 6.3. ■

Lemma 30 *Be $\mathbb{H} \subset L^2_{\mathbf{X}}$ the RKHS defined in the proof of Theorem 16. Then $\mathbb{H}$ is dense in $L^2_{\mathbf{X}}$.*

Proof First, note that the function $\mathbf{x} \in \mathbb{R}^n \mapsto k(\mathbf{x},\mathbf{x})$ is in $L^1(\mathbb{R}^n, d\mu)$. Indeed

$$\begin{aligned}\int_{\mathbb{R}^n}\int_{\mathbb{S}^n} \psi_{\mathbf{w}}(\mathbf{x})^2\, d\tau(\mathbf{w})\, d\mu(\mathbf{x}) &= \int_{\mathbb{R}^n}(1+\|\mathbf{x}\|^2)\int_{\mathbb{S}^n}\psi_{\mathbf{w}}(\mathbf{x}/\|(\mathbf{x},1)\|)^2\, d\tau(\mathbf{w})\, d\mu(\mathbf{x})\\ &\le (1+\mathbb{E}\|\mathbf{X}\|^2)\max_{\mathbf{w},\mathbf{y}\in\mathbb{S}^n}\psi_{\mathbf{w}}(\mathbf{y})^2\end{aligned}$$

This implies that $\mathbb{H} \subseteq L^2(\mathbb{R}^n, d\mu)$. Now, we would like to show that V_σ is dense in $\overline{\mathbb{H}}$, where

$$V_\sigma = \left\{\sum_{i=1}^k u_i\psi_{\mathbf{w}_i} : \mathbf{u} \in \mathbb{R}^k, \mathbf{w}_1,\dots,\mathbf{w}_k \in \mathbb{S}^n, k \ge 1\right\}$$

It suffices to show that, for every $\mathbf{w} \in \mathbb{S}^{n-1}$, there exists a sequence $\{f_n\}_{n\ge1} \subset \mathbb{H}$ such that $f_n \to \psi_{\mathbf{w}}$ in $L^2_{\mathbf{X}}$. Choose $g_k \in L^2(\mathbb{S}^n, d\tau)$ such that $\operatorname{supp}(g_k) \subseteq B_{1/k}(\mathbf{w}) \doteq \{\mathbf{v} \in \mathbb{S}^n : \|\mathbf{v}-\mathbf{w}\| \le 1/k\}$, $\int_{\mathbb{S}^n} g(\mathbf{v})\, d\tau(\mathbf{v}) = 1$ and $g_k \ge 0$, and define $f_k \in \mathbb{H}$ as $f_k(x) = \int_{\mathbb{S}^n} g_k(\mathbf{v})\psi_{\mathbf{v}}(x)\, d\tau(\mathbf{x})$. Then

$$\begin{aligned}\|f_k - \psi_{\mathbf{w}}\|^2_{L^2(\mathbb{R}^n,d\mu)} &= \int_{\mathbb{R}^n}\left(\int_{\mathbb{S}^n} g_k(\mathbf{v})(\psi_{\mathbf{v}}(\mathbf{x}) - \psi_{\mathbf{w}}(\mathbf{x}))\, d\tau(\mathbf{v})\right)^2 d\mu(\mathbf{x})\\ &\le (1+\mathbb{E}\|X\|^2)\max_{\substack{\mathbf{v}\in B_{1/k}(\mathbf{w})\\ \mathbf{y}\in\mathbb{S}^n}}(\psi_{\mathbf{v}}(\mathbf{y}) - \psi_{\mathbf{w}}(\mathbf{y}))^2 \to 0\end{aligned}$$

as $k \to \infty$. This shows that V_σ is dense in $\overline{\mathbb{H}}$. Thanks to Theorem 1 in (Hornik, 1991), it holds that V_σ is dense in $L^2(\mathbb{R}^n, d\mu)$. This implies the statement of the lemma. ■

Journal of Machine Learning Research 20 (2019) 1-47 Submitted 1/19; Revised 7/19; Published 8/19

Stochastic Variance-Reduced Cubic Regularization Methods

Dongruo Zhou DRZHOU@CS.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Pan Xu PANXU@CS.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Quanquan Gu QGU@CS.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Editor: Zhihua Zhang

Abstract

We propose a stochastic variance-reduced cubic regularized Newton method (SVRC) for non-convex optimization. At the core of SVRC is a novel semi-stochastic gradient along with a semi-stochastic Hessian, which are specifically designed for cubic regularization method. For a nonconvex function with n component functions, we show that our algorithm is guaranteed to converge to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum within $\widetilde{O}(n^{4/5}/\epsilon^{3/2})$[1] second-order oracle calls, which outperforms the state-of-the-art cubic regularization algorithms including subsampled cubic regularization. To further reduce the sample complexity of Hessian matrix computation in cubic regularization based methods, we also propose a sample efficient stochastic variance-reduced cubic regularization (Lite-SVRC) algorithm for finding the local minimum more efficiently. Lite-SVRC converges to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum within $\widetilde{O}(n + n^{2/3}/\epsilon^{3/2})$ Hessian sample complexity, which is faster than all existing cubic regularization based methods. Numerical experiments with different nonconvex optimization problems conducted on real datasets validate our theoretical results for both SVRC and Lite-SVRC.

Keywords: Cubic Regularization, Nonconvex Optimization, Variance Reduction, Hessian Sample Complexity, Local Minimum

1. Introduction

We study the following unconstrained finite-sum nonconvex optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

1. Here $\widetilde{O}$ hides poly-logarithmic factors.

©2019 Dongruo Zhou, Pan Xu and Quanquan Gu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-055.html.

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is a general nonconvex function. Such nonconvex optimization problems are ubiquitous in machine learning, including training deep neural network (LeCun et al., 2015), robust linear regression (Yu and Yao, 2017) and nonconvex regularized logistic regression (Reddi et al., 2016b). In principle, finding the global minimum of (1) is generally a NP-hard problem (Hillar and Lim, 2013) due to the lack of convexity.

Instead of finding the global minimum, various algorithms have been developed in the literature (Nesterov and Polyak, 2006; Cartis et al., 2011a; Carmon and Duchi, 2016; Agarwal et al., 2017; Xu et al., 2018b; Allen-Zhu and Li, 2018) to find an approximate local minimum of (1). In particular, a point $\mathbf{x}$ is said to be an (ϵ_g, ϵ_H)-approximate local minimum of F if

$$\|\nabla F(\mathbf{x})\|_2 \le \epsilon_g, \quad \lambda_{\min}(\nabla^2 F(\mathbf{x})) \ge -\epsilon_H, \tag{2}$$

where $\epsilon_g, \epsilon_H > 0$ are predefined precision parameters. It has been shown that such approximate local minima can be as good as global minima in some problems. For instance, Ge et al. (2016) proved that any local minimum is actually a global minimum in matrix completion problems. Therefore, to develop an algorithm to find an approximate local minimum is of great interest both in theory and practice.

A very important and popular method to find the approximate local minimum is cubic-regularized (CR) Newton method, which was originally introduced by Nesterov and Polyak (2006). Generally speaking, in the k-th iteration, CR solves a sub-problem which minimizes a cubic-regularized second-order Taylor expansion at current iterate $\mathbf{x}_k$. The update rule can be written as follows:

$$\mathbf{h}_k = \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \nabla F(\mathbf{x}_k), \mathbf{h}\rangle + \frac{1}{2}\langle \nabla^2 F(\mathbf{x}_k)\mathbf{h}, \mathbf{h}\rangle + \frac{M}{6}\|\mathbf{h}\|_2^3, \tag{3}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k, \tag{4}$$

where $M > 0$ is a penalty parameter used in CR. Nesterov and Polyak (2006) proved that to find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum of a nonconvex function F, CR requires at most $O(\epsilon^{-3/2})$ iterations. However, a major drawback for CR is that it needs to sample n individual gradients $\nabla f_i(\mathbf{x}_k)$ and Hessian matrices $\nabla^2 f_i(\mathbf{x}_k)$ in (3) at each iteration, which leads to a total $O(n\epsilon^{-3/2})$ Hessian sample complexity, i.e., number of queries to the stochastic Hessian $\nabla^2 f_i(\mathbf{x})$ for some i and $\mathbf{x}$. Such computational cost will be extremely expensive when n is large as in many large scale machine learning problems.

To overcome the computational burden of CR based methods, some recent studies have proposed to use sub-sampled Hessian instead of the full Hessian (Kohler and Lucchi, 2017; Xu et al., 2017a) to reduce the Hessian complexity. In detail, Kohler and Lucchi (2017) proposed a sub-sampled cubic-regularized Newton method (SCR), which uses a subsampled Hessian instead of full Hessian to reduce the per iteration sample complexity of Hessian evaluations. Xu et al. (2017a) proposed a refined convergence analysis of SCR, as well as a subsampled Trust Region algorithm (Conn et al., 2000). Nevertheless, SCR bears a much slower convergence rate than the original CR method, and the total Hessian sample complexity for SCR to achieve an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum is $\widetilde{O}(\epsilon^{-5/2})$. This suggests that the computational cost of SCR could be even worse than CR when $\epsilon \lesssim n^{-1}$.

In this paper, we propose a novel cubic regularization algorithm named Stochastic Variance-Reduced Cubic regularization (SVRC), which incorporates the variance reduction techniques (Johnson and Zhang, 2013; Xiao and Zhang, 2014; Allen-Zhu and Hazan,

2016; Reddi et al., 2016a) into the cubic-regularized Newton method. The key component in our algorithm is a novel semi-stochastic gradient, together with a semi-stochastic Hessian, that are specifically designed for cubic regularization. Furthermore, we prove that, for L_2-Hessian Lipschitz functions, to attain an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum, our proposed algorithm requires $O(n+n^{4/5}/\epsilon^{3/2})$ Second-order Oracle (SO) calls and $O(1/\epsilon^{3/2})$ Cubic Subproblem Oracle (CSO) calls. Here an SO oracle represents an evaluation of triple $(f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x}))$, and a CSO oracle denotes an evaluation of the exact solution (or inexact solution) of the cubic subproblem (3). Compared with the original cubic regularization algorithm (Nesterov and Polyak, 2006), which requires $O(n/\epsilon^{3/2})$ SO calls and $O(1/\epsilon^{3/2})$ CSO calls, our proposed SVRC algorithm reduces the SO calls by a factor of $\Omega(n^{1/5})$.

The second-order oracle complexity is dominated by the maximum number of queries to one of the elements in the triplet $(f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x}))$, and therefore is not always desirable in reflecting the computational complexity of multifarious applications. Therefore, we need to focus more on the Hessian sample complexity of cubic regularization methods for relatively high dimensional problems. Based on the SVRC algorithm, in order to further reduce the Hessian sample complexity, we also develop a sample efficient stochastic variance-reduced cubic-regularized Newton method called Lite-SVRC, which significantly reduces the sample complexity of Hessian matrix evaluations in stochastic CR methods. Under mild conditions, we prove that Lite-SVRC achieves a lower Hessian sample complexity than existing cubic regularization based methods. We prove that Lite-SVRC converges to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum of a nonconvex function within $\widetilde{O}(n + n^{2/3}\epsilon^{-3/2})$ Hessian sample complexity.

We summarize the major contributions of this paper as follows:

- We present a novel cubic regularization method (SVRC) with improved oracle complexity. To the best of our knowledge, this is the first algorithm that outperforms cubic regularization without any loss in convergence rate. In sharp contrast, existing subsampled cubic regularization methods (Kohler and Lucchi, 2017; Xu et al., 2017a) suffer from worse convergence rates than cubic regularization.
- We also extend SVRC to the case with inexact solution to the cubic regularization subproblem. Similar to previous work (Cartis et al., 2011a; Xu et al., 2017a), we layout a set of sufficient conditions, under which the output of the inexact algorithm is still guaranteed to have the same convergence rate and oracle complexity as the exact algorithm. This further sheds light on the practical implementation of our algorithm.
- As far as we know, our work is the first to rigorously demonstrate the advantage of variance reduction for second-order optimization algorithms. Although there exist a few studies (Lucchi et al., 2015; Moritz et al., 2016; Rodomanov and Kropotov, 2016) using variance reduction to accelerate Newton method, none of them can deliver faster rates of convergence than standard Newton method.
- We also propose a lite version of SVRC, namely, the Lite-SVRC algorithm, which only requires a constant batch size of Hessian evaluations at each iteration. The proposed Lite-SVRC further improves the Hessian sample complexity of SVRC and outperforms the state-of-the-art result by achieving $\widetilde{O}(n + n^{2/3}\epsilon^{-3/2})$ Hessian sample complexity.

- We conduct extensive numerical experiments with different types of nonconvex optimization problems on various real datasets to validate our theoretical results for both SVRC and Lite-SVRC.

When the short version of this paper was submitted to ICML, there was a concurrent work by Wang et al. (2018a), which applies the idea of stochastic variance reduction to cubic regularization as well. Their algorithms have a worse Hessian sample complexity than Lite-SVRC. Since the short version of this paper was published in ICML, there have been two followup works by Wang et al. (2018b) and Zhang et al. (2018), which both proposed similar algorithms to our Lite-SVRC algorithm, and achieved the same Hessian sample complexity. However, Wang et al. (2018b) and Zhang et al. (2018)'s results rely on the adaptive choice of batch size for stochastic Hessian. Furthermore, Zhang et al. (2018)'s result relies on a stronger notion of Hessian Lipschitz condition. We will discuss the key difference between our Lite-SVRC algorithm and the algorithms in Wang et al. (2018a,b); Zhang et al. (2018) in detail in Section 7.

Notation: We use $a(x) = O(b(x))$ if $a(x) \leq Cb(x)$, where C is a constant independent of any parameters in our algorithm. We use $\widetilde{O}(\cdot)$ to hide polynomial logarithm terms. We use $\|\mathbf{v}\|_2$ to denote the 2-norm of vector $\mathbf{v} \in \mathbb{R}^d$. For symmetric matrix $\mathbf{H} \in \mathbb{R}^{d\times d}$, we use $\|\mathbf{H}\|_2$ and $\|\mathbf{H}\|_{S_r}$ to denote the spectral norm and Schatten r- norm of $\mathbf{H}$. We denote the smallest eigenvalue of $\mathbf{H}$ to be $\lambda_{\min}(\mathbf{H})$.

2. Related Work

Cubic Regularization and Trust-region Newton Method Traditional Newton method in convex setting has been widely studied in past decades (Bennett, 1916; Bertsekas, 1999). The most related work to ours is the nonconvex cubic regularized Newton method, which was originally proposed in Nesterov and Polyak (2006). Cartis et al. (2011a) presented an adaptive framework of cubic regularization, which uses an adaptive estimation of the local Lipschitz constant and approximate solution to the cubic subproblem. To connect cubic regularization with traditional trust region method (Conn et al., 2000; Cartis et al., 2009, 2012, 2013), Blanchet et al. (2016); Curtis et al. (2017); Martínez and Raydan (2017) showed that the trust-region Newton method can achieve the same iteration complexity as the cubic regularization method. To overcome the computational burden of gradient and Hessian matrix evaluations, Kohler and Lucchi (2017); Xu et al. (2017a,b) proposed to use subsampled gradient and Hessian in cubic regularization. On the other hand, in order to solve the cubic subproblem (3) more efficiently, Carmon and Duchi (2016) proposed to use gradient descent, while Agarwal et al. (2017) proposed a sophisticated algorithm based on approximate matrix inverse and approximate PCA. Tripuraneni et al. (2018) proposed a refined stochastic cubic regularization algorithm based on above subproblem solver. However, none of the aforementioned variants of cubic regularization outperforms the original cubic regularization method in terms of oracle complexity.

Finding Approximate Local Minima There is another line of work for finding approximate local minima which focuses on escaping from nondegenerated saddle points using the negative curvature. Ge et al. (2015); Jin et al. (2017) showed that simple (stochastic) gradient descent with an injected uniform noise over a small ball is able to converge to approximate local minima. Carmon et al. (2018); Royer and Wright (2018); Allen-Zhu (2018)

showed that by calculating the negative curvature using Hessian information or Hessian vector product, one can find approximate local minima faster than first-order methods. Xu et al. (2018b); Allen-Zhu and Li (2018); Jin et al. (2018) further proved that gradient methods with additive noise are also able to find approximate local minima faster than the first-order methods. Yu et al. (2017) proposed the GOSE algorithm to save negative curvature computation and Yu et al. (2018) improved the gradient complexity by exploring the third-order smoothness of objective functions. Raginsky et al. (2017); Zhang et al. (2017); Xu et al. (2018a) proved that a family of algorithms based on discretizations of Langevin dynamics can find a neighborhood of the global minimum of nonconvex objective functions.

Variance Reduction Variance-reduced techniques play an important role in our proposed algorithm, which have been extensively studied for large-scale finite-sum optimization problems. Variance reduction was first proposed in convex finite-sum optimization (Roux et al., 2012; Johnson and Zhang, 2013; Xiao and Zhang, 2014; Defazio et al., 2014), which uses semi-stochastic gradient to reduce the variance of the stochastic gradient and improves the gradient complexity of both stochastic gradient descent (SGD) and gradient descent (GD). Representative algorithms include Stochastic Average Gradient (SAG) (Roux et al., 2012), Stochastic Variance Reduced Gradient (SVRG) (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014), to mention a few. Garber and Hazan (2015); Shalev-Shwartz (2016) studied non-convex finite-sum problems where each individual function may be non-convex, but their sum is still convex. Reddi et al. (2016a) and Allen-Zhu and Hazan (2016) extended SVRG to the general non-convex finite-sum optimization, and proved that SVRG is able to converge to a first-order stationary point with the same convergence rate as gradient descent, yet with an $\Omega(n^{1/3})$ improvement in gradient complexity. Recently Zhou et al. (2018b) and Fang et al. (2018) further improved the gradient complexity of SVRG type of algorithms to converge to a first-order stationary point in nonconvex optimization to an optimal rate. However, to the best of our knowledge, it is still an open problem whether variance reduction can improve the oracle complexity of second-order optimization algorithms.

The remainder of this paper is organized as follows: we present the stochastic variance-reduced cubic regularization (SVRC) algorithm in Section 3. We present our theoretical analysis of the proposed SVRC algorithm in Section 4 and discuss on SVRC with inexact cubic subproblem oracles in Section 5. In Section 6, we propose a modified algorithm, Lite-SVRC, to further reduce Hessian sample complexity and present its theoretical analysis in Section 7. We conduct thorough numerical experiments on different nonconvex optimization problems and on different real world datasets to validate our theory in Section 8. We conclude our work in Section 9.

3. Stochastic Variance-Reduced Cubic Regularization

In this section, we present a novel algorithm, which utilizes stochastic variance reduction techniques to improve cubic regularization method.

As is discussed in the introduction, to reduce the computation burden of gradient and Hessian matrix evaluations in the cubic regularization updates in (3), subsampled gradient and Hessian matrix have been used in subsampled cubic regularization (Kohler and Lucchi, 2017; Xu et al., 2017b) and stochastic cubic regularization (Tripuraneni et al., 2018).

Algorithm 1 Stochastic Variance Reduction Cubic Regularization (**SVRC**)

Input: batch size parameters b_g, b_h, cubic penalty parameters $\{M_{s,t}\}$, epoch number S, epoch length T and starting point $\mathbf{x}_0$.
Initialization $\widehat{\mathbf{x}}^1 = \mathbf{x}_0$
for $s = 1, \ldots, S$ **do**
  $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$
  $\mathbf{g}^s = \nabla F(\widehat{\mathbf{x}}^s) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\widehat{\mathbf{x}}^s), \mathbf{H}^s = \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(\widehat{\mathbf{x}}^s)$
  for $t = 0, \ldots, T-1$ **do**
    Sample index set $I_g, I_h, |I_g| = b_g, |I_h| = b_h$;
    $\mathbf{v}_t^s = \frac{1}{b_g}\sum_{i_t \in I_g}\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s - \big(\frac{1}{b_g}\sum_{i_t \in I_g}\nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s) - \mathbf{H}^s\big)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)$
    $\mathbf{U}_t^s = \frac{1}{b_h}\sum_{j_t \in I_h}\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{H}^s$
    $\mathbf{h}_t^s = \operatorname{argmin}\langle \mathbf{v}_t^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{U}_t^s\mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}\|_2^3$,
    $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$
  end for
  $\widehat{\mathbf{x}}^{s+1} = \mathbf{x}_T^s$
end for
Output: $\mathbf{x}_{\text{out}} = \mathbf{x}_t^s$, where s, t are uniformly random chosen from $s \in [S]$ and $t \in [T]$.

Nevertheless, the stochastic gradient and Hessian matrix have large variances, which undermine the convergence performance. Inspired by SVRG (Johnson and Zhang, 2013), we propose to use a semi-stochastic version of gradient and Hessian matrix, which can control the variances automatically. Specifically, our algorithm has two loops. At the beginning of the s-th iteration of the outer loop, we denote $\widehat{\mathbf{x}}^s = \mathbf{x}_0^s$. We first calculate the full gradient $\mathbf{g}^s = \nabla F(\widehat{\mathbf{x}}^s)$ and Hessian matrix $\mathbf{H}^s = \nabla^2 F(\widehat{\mathbf{x}}^s)$, which are stored for further references in the inner loop. At the t-th iteration of the inner loop, we calculate the following semi-stochastic gradient and Hessian matrix:

$$\mathbf{v}_t^s = \frac{1}{b_g}\sum_{i_t \in I_g}\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s - \frac{1}{b_g}\sum_{i_t \in I_g}\big(\nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s) - \mathbf{H}^s\big)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s), \tag{5}$$

$$\mathbf{U}_t^s = \frac{1}{b_h}\sum_{j_t \in I_h}\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{H}^s, \tag{6}$$

where I_g and I_h are batch index sets, and the batch sizes $b_g = |I_g|, b_h = |I_h|$ will be decided later. In each inner iteration, we solve the following cubic regularization subproblem:

$$\begin{aligned} \mathbf{h}_t^s &= \operatorname{argmin} m_t^s(\mathbf{h}), \\ m_t^s(\mathbf{h}) &= \langle \mathbf{v}_t^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{U}_t^s\mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}\|_2^3, \end{aligned} \tag{7}$$

where $\{M_{s,t}\}$ are cubic regularization parameters, which may depend on s and t. Then we perform the update $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$ in the t-th iteration of the inner loop. The proposed algorithm is displayed in Algorithm 1.

There are two notable features of our "estimator" of the full gradient and Hessian in each inner loop, compared with that used in SVRG (Johnson and Zhang, 2013). The first

is that our gradient and Hessian estimators consist of mini-batches of stochastic gradient and Hessian. The second one is that we use second-order information when we construct the gradient estimator $\mathbf{v}_t^s$, while classical SVRG only uses first-order information to build it. Intuitively speaking, both features are used to make a more accurate estimation of the true gradient and Hessian with affordable oracle calls. Note that similar approximations of the gradient and Hessian matrix have been staged in recent work by Gower et al. (2018) and Wai et al. (2017), where they used this new kind of estimator for traditional SVRG in the convex setting, which radically differs from our setting.

4. Theoretical Analysis of SVRC

In this section, we prove the convergence rate of SVRC (Algorithm 1) to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum. We first lay out the following Hessian Lipschitz assumption, which is necessary for our analysis and is widely used in the literature (Nesterov and Polyak, 2006; Xu et al., 2016; Kohler and Lucchi, 2017).

Assumption 1 (Hessian Lipschitz) *There exists a constant $L_2 > 0$, such that for all $\mathbf{x}, \mathbf{y}$ and $i \in [n]$*

$$\left\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\mathbf{y})\right\|_2 \le L_2 \|\mathbf{x} - \mathbf{y}\|_2.$$

The Hessian Lipschitz assumption plays a central role in controlling the changing speed of second order information. In fact, this is the only assumption we need to prove our theoretical results for SVRC. We then define the following optimal function gap between initial point $\mathbf{x}_0$ and the global minimum of F.

Definition 2 (Optimal Gap) *For function $F(\cdot)$ and the initial point $\mathbf{x}_0$, let Δ_F be*

$$\Delta_F = \inf\{\Delta \in \mathbb{R} : F(\mathbf{x}_0) - F^* \le \Delta\},$$

where $F^* = \inf_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})$.

W.L.O.G., we assume $\Delta_F < +\infty$ throughout this paper. Before we present nonasymptotic convergence results of Algorithm 1, we define the following useful notation

$$\mu(\mathbf{x}) = \max\left\{\|\nabla F(\mathbf{x})\|_2^{3/2}, -\frac{\lambda_{\min}^3\big(\nabla^2 F(\mathbf{x})\big)}{L_2^{3/2}}\right\}. \tag{8}$$

A similar definition also appears in Nesterov and Polyak (2006) with a slightly different form, which is used to describe how much a point is similar to a true local minimum. In particular, according to the definition in (8), $\mu(\mathbf{x}) \le \epsilon^{3/2}$ holds if and only if

$$\|\nabla F(\mathbf{x})\|_2 \le \epsilon, \quad \lambda_{\min}\big(\nabla^2 F(\mathbf{x})\big) > -\sqrt{L_2\epsilon}. \tag{9}$$

Therefore, in order to find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum of the nonconvex function F, it suffices to find $\mathbf{x}$ which satisfies $\mu(\mathbf{x}) < \epsilon^{3/2}$. Next we formally define our oracles:

Definition 3 (Second-order Oracle) *Given an index i and a point $\mathbf{x}$, one second-order oracle (SO) call returns such a triple:*

$$[f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x})]. \tag{10}$$

Definition 4 (Cubic Subproblem Oracle) *Given a vector $\mathbf{g} \in \mathbb{R}^d$, a Hessian matrix $\mathbf{H}$ and a positive constant θ, one Cubic Subproblem Oracle (CSO) call returns $\mathbf{h}_{sol}$, where $\mathbf{h}_{sol}$ can be solved exactly as follows*

$$\mathbf{h}_{sol} = \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \mathbf{g}, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{h}, \mathbf{H}\mathbf{h}\rangle + \frac{\theta}{6}\|\mathbf{h}\|_2^3.$$

Remark 5 *The second-order oracle is a special form of* Information Oracle *firstly introduced by Nemirovsky and Yudin (1983), which returns gradient, Hessian and all high order derivatives of the objective function $F(\mathbf{x})$. Here, our second-order oracle will only returns first and second order information at some point of single objective f_i instead of F. We argue that it is a reasonable adaption because in this paper we focus on finite-sum objective function. The Cubic Subproblem Oracle will return an exact or inexact solution of (7), which plays an important role in both theory and practice.*

Now we are ready to give a general convergence result of Algorithm 1:

Theorem 6 *Under Assumption 1, suppose that the cubic regularization parameter $M_{s,t}$ of Algorithm 1 satisfies that $M_{s,t} = C_M L_2$, where L_2 is the Hessian Lipschitz parameter and $C_M \ge 100$ is a constant. The batch sizes b_g and b_h satisfy that*

$$b_g \ge 5T^4, \ b_h \ge 100T^2 \log d, \tag{11}$$

where $T \ge 2$ is the length of the inner loop of Algorithm 1 and d is the dimension of the problem. Then the output of Algorithm 1 satisfies

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \le \frac{240 C_M^2 L_2^{1/2} \Delta_F}{ST}. \tag{12}$$

Remark 7 *According to (8), to ensure that $\mathbf{x}_{out}$ is an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum, we can set the right hand side of (12) to be less than $\epsilon^{3/2}$. This immediately implies that the total iteration complexity of Algorithm 1 is $ST = O(\Delta_F L_2^{1/2}\epsilon^{-3/2})$, which matches the iteration complexity of cubic regularization (Nesterov and Polyak, 2006).*

Remark 8 *Note that there is a $\log d$ term in the expression of the parameter, and it is only related to Hessian batch size b_h. The $\log d$ term comes from matrix concentration inequalities, which is believed to be unavoidable (Tropp et al., 2015). In other words, the batch size of Hessian matrix b_h has an inevitable relation to dimension d, unlike the batch size of gradient b_g.*

The result in Theorem 6 depends on a series of parameters. In the following corollary, we will show how to choose these parameters in practice to achieve a better oracle complexity.

Corollary 9 *Under Assumption 1, let the cubic regularization parameter $M_{s,t} = M = C_M L_2$, where $C_M \geq 100$ is a constant. Let the epoch length $T = n^{1/5}$, batch sizes $b_g = 5n^{4/5}$, $b_h = 100n^{2/5}\log d$, and the number of epochs $S = \max\{1, 240C_M^2 L_2^{1/2}\Delta_F n^{-1/5}\epsilon^{-3/2}\}$. Then Algorithm 1 will find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum $\mathbf{x}_{out}$ within*

$$O\Big(n + \frac{\Delta_F\sqrt{L_2}n^{4/5}}{\epsilon^{3/2}}\Big) \text{ SO calls} \tag{13}$$

and

$$O\Big(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\Big) \text{ CSO calls.} \tag{14}$$

Remark 10 *Corollary 9 states that we can reduce the SO calls by setting the batch size b_g, b_h related to n. In contrast, in order to achieve an $(\epsilon, \sqrt{L_2\epsilon})$ local minimum, original cubic regularization method in Nesterov and Polyak (2006) needs $O(n/\epsilon^{3/2})$ second-order oracle calls, which is by a factor of $n^{1/5}$ worse than ours. And subsampled cubic regularization (Kohler and Lucchi, 2017; Xu et al., 2017b) requires $\widetilde{O}(n/\epsilon^{3/2} + 1/\epsilon^{5/2})$ SO calls, which is also worse than our algorithm.*

In Table 1, we summarize the comparison of our SVRC algorithm with the most related algorithms in terms of SO and CSO oracle complexities. It can be seen from Table 1 that our algorithm (SVRC) achieves the lowest (SO and CSO) oracle complexity compared with the original cubic regularization method (Nesterov and Polyak, 2006) which employs full gradient and Hessian evaluations and the subsampled cubic method (Kohler and Lucchi, 2017; Xu et al., 2017b). In particular, our algorithm reduces the SO oracle complexity of cubic regularization by a factor of $n^{1/5}$ for finding an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum.

Algorithm	SO calls	CSO calls	Gradient Lipschitz	Hessian Lipschitz
CR	$O\Big(\frac{n}{\epsilon^{3/2}}\Big)$	$O\Big(\frac{1}{\epsilon^{3/2}}\Big)$	no	yes
SCR	$\widetilde{O}\Big(\frac{n}{\epsilon^{3/2}} + \frac{1}{\epsilon^{5/2}}\Big)$[2]	$O\Big(\frac{1}{\epsilon^{3/2}}\Big)$	yes	yes
SVRC (Algorithm 1)	$\widetilde{O}\Big(n + \frac{n^{4/5}}{\epsilon^{3/2}}\Big)$	$O\Big(\frac{1}{\epsilon^{3/2}}\Big)$	no	yes

Table 1: Comparisons between different methods to find $(\epsilon, \sqrt{L_2\epsilon})$-local minimum on the second-order oracle (SO) complexity and the cubic sub-problem oracle (CSO) complexity. The compared methods include (1) CR: Cubic regularization (Nesterov and Polyak, 2006) and (2) SCR: Subsampled cubic regularization (Kohler and Lucchi, 2017; Xu et al., 2017b).

2. It is the refined rate proved by Xu et al. (2017b) for the subsampled cubic regularization algorithm proposed in Kohler and Lucchi (2017).

5. SVRC with Inexact Oracles

In practice, the exact solution to the cubic subproblem (7) cannot be obtained. Instead, one can only get an approximate solution by some inexact solver. Thus we replace the CSO oracle in (4) with the following inexact CSO oracle

$$\widetilde{\mathbf{h}}_{\text{sol}} \approx \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \mathbf{g}, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{h}, \mathbf{H}\mathbf{h}\rangle + \frac{\theta}{6}\|\mathbf{h}\|_2^3.$$

To analyze the performance of SVRC with inexact cubic subproblem solver, we relax the exact solver $\mathbf{h}_t^s$ in Line 10 of Algorithm 1 with

$$\widetilde{\mathbf{h}}_t^s \approx \operatorname{argmin} m_t^s(\mathbf{h}). \tag{15}$$

The ultimate goal of this section is to prove that the theoretical results of SVRC still hold with inexact subproblem solvers. To this end, we present the following sufficient condition, under which inexact solution can ensure the same oracle complexity as the exact solution:

Condition 11 (Inexact Condition) *For each s, t and a given $\delta > 0$, $\widetilde{\mathbf{h}}_t^s$ satisfies δ-inexact condition if $\widetilde{\mathbf{h}}_t^s$ satisfies*

$$m_t^s(\widetilde{\mathbf{h}}_t^s) \leq -\frac{M_{s,t}}{12}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \delta,$$
$$\|\nabla m_t^s(\widetilde{\mathbf{h}}_t^s)\|_2 \leq M_{s,t}^{1/3}\delta^{2/3},$$
$$\big|\|\widetilde{\mathbf{h}}_t^s\|_2 - \|\mathbf{h}_t^s\|_2\big| \leq M_{s,t}^{-1/3}\delta^{1/3}.$$

Remark 12 *Similar inexact conditions have been studied in the literature of cubic regularization. For instance, Nesterov and Polyak (2006) presented a practical way to solve the cubic subproblem without termination condition. Cartis et al. (2011a); Kohler and Lucchi (2017) presented termination criteria for approximate solution to cubic subproblem, which is slightly different from Condition 11.*

Now we present the convergence result of SVRC with inexact CSO oracles:

Theorem 13 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$, which satisfies Condition 11. Under the same conditions of Theorem 6, the output of Algorithm 1 satisfies*

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \leq \frac{240C_M^2 L_2^{1/2}\Delta_F}{ST} + 480C_M^2 L_2^{1/2}\delta. \tag{16}$$

Remark 14 *By the definition of $\mu(\mathbf{x})$, in order to attain an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum, we require $\mathbb{E}[\mu(\mathbf{x}_{out})] \leq \epsilon^{3/2}$ and thus $480C_M^2 L_2^{1/2}\delta < \epsilon^{3/2}$, which implies that δ in Condition 11 should satisfy $\delta < (480C_M^2 L_2^{1/2})^{-1}\epsilon^{3/2}$. Thus the total iteration complexity of Algorithm 1 with inexact oracle is still $O(\Delta_F L_2^{1/2}\epsilon^{-3/2})$.*

By the same choice of parameters, Algorithm 1 with inexact oracle can achieve a reduction in SO calls.

Corollary 15 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$, which satisfies Condition 11 with $\delta = (960C_M^2)^{-1}L_2^{-1/2}\epsilon^{3/2}$. Under Assumption 1, let the cubic regularization parameter $M_{s,t} = M = C_M L_2$, where $C_M \geq 100$ is a constant. Let the epoch length $T = n^{1/5}$, batch sizes $b_g = 5n^{4/5}$ and $b_h = 100n^{2/5}\log d$, and the number of epochs $S = \max\{1, 480C_M^2 L_2^{1/2}\Delta_F n^{-1/5}\epsilon^{-3/2}\}$. Then Algorithm 1 will find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum within*

$$O\Big(n + \frac{\Delta_F\sqrt{L_2}n^{4/5}}{\epsilon^{3/2}}\Big) \text{ SO calls} \tag{17}$$

and

$$O\Big(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\Big) \text{ CSO calls.} \tag{18}$$

Remark 16 *It is worth noting that even with the inexact CSO oracle satisfying Condition 11, the SO and CSO complexities of SVRC remain the same as that of SVRC with exact CSO oracle. Furthermore, this result always holds with any inexact cubic sub-problem solver.*

6. Lite-SVRC for Efficient Hessian Sample Complexity

As we discussed in the introduction section, when the problem dimension d is relatively high, we may want to focus more on the Hessian sample complexity of cubic regularization methods than the second-order oracle complexity. In this section, we present a new algorithm Lite-SVRC based on SVRC, which trades the second-order oracle complexity for a more affordable Hessian sample complexity. As is displayed in Algorithm 2, our Lite-SVRC algorithm has similar structure as Algorithm 1 with S epochs and T iterations within each epoch. At the t-th iteration of the s-th epoch, we also use a semi-stochastic gradient $\widetilde{\mathbf{v}}_t^s$ and Hessian $\mathbf{U}_t^s$ to replace the full gradient and full Hessian in CR subproblem (3) as follows

$$\widetilde{\mathbf{v}}_t^s = \frac{1}{B_{g;s,t}}\sum_{i_t\in I_g}\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s, \tag{19}$$

$$\mathbf{U}_t^s = \frac{1}{B_h}\sum_{j_t\in I_h}\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{H}^s, \tag{20}$$

where $\widehat{\mathbf{x}}^s$ is the reference point at which $\mathbf{g}^s$ and $\mathbf{H}^s$ are computed, I_g and I_h are sampling index sets (with replacement), $B_{g;s,t}$ and B_h are sizes of I_g and I_h.

Compared with SVRC (Algorithm 1), Lite-SVRC uses a lite version of semi-stochastic gradient $\widetilde{\mathbf{v}}_t^s$. Note that the additional Hessian information in the semi-stochastic gradient in (5) actually increases the Hessian sample complexity. Therefore, with the goal of reducing the Hessian sample complexity, the standard semi-stochastic gradient (Johnson and Zhang, 2013; Xiao and Zhang, 2014) is used in this section. Note that similar semi-stochastic gradient and Hessian have been proposed in Johnson and Zhang (2013); Xiao and Zhang (2014) and Gower et al. (2018); Wai et al. (2017); Zhou et al. (2018a); Wang et al. (2018a,b); Zhang et al. (2018) respectively. In Algorithm 2, we choose fixed batch size of stochastic Hessian as $B_h = |I_h|$. However, the batch size of stochastic gradient is chosen adaptively at each iteration:

$$B_{g;s,t} = D_g/\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2, \tag{21}$$

where D_g is a constant only depending on n and d.

Algorithm 2 Sample efficient stochastic variance-reduced cubic regularization method (Lite-SVRC)

Input: batch size parameters D_g, B_h, cubic penalty parameter $\{M_{s,t}\}$, epoch number S, epoch length T and starting point $\mathbf{x}_0$.
Initialization $\widehat{\mathbf{x}}^1 = \mathbf{x}_0$
for $s = 1, \ldots, S$ **do**
  $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$
  $\mathbf{g}^s = \nabla F(\widehat{\mathbf{x}}^s) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\widehat{\mathbf{x}}^s), \mathbf{H}^s = \nabla^2 F(\widehat{\mathbf{x}}^s) = \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(\widehat{\mathbf{x}}^s)$
  $\mathbf{h}_0^s = \mathrm{argmin}_{\mathbf{h}\in\mathbb{R}^d}\, m_0^s(\mathbf{h}) = \langle \mathbf{g}^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{H}^s\mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,0}}{6}\|\mathbf{h}\|_2^3$
  $\mathbf{x}_1^s = \mathbf{x}_0^s + \mathbf{h}_0^s$
  for $t = 1, \ldots, T-1$ **do**
    $B_{g;s,t} = D_g/\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2, t > 0$
    Sample index set $I_g, I_h \subseteq [n]$, $|I_g| = B_{g;s,t}, |I_h| = B_h$
    $\widetilde{\mathbf{v}}_t^s = \frac{1}{B_{g;s,t}}\big(\sum_{i_t\in I_g}\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big) + \mathbf{g}^s$
    $\mathbf{U}_t^s = \frac{1}{B_h}\big(\sum_{j_t\in I_h}\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big) + \mathbf{H}^s$
    $\mathbf{h}_t^s = \mathrm{argmin}_{\mathbf{h}\in\mathbb{R}^d}\, m_t^s(\mathbf{h}) = \langle \widetilde{\mathbf{v}}_t^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{U}_t^s\mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}\|_2^3$
    $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$
  end for
  $\widehat{\mathbf{x}}^{s+1} = \mathbf{x}_T^s$
end for
Output: $\mathbf{x}_{\mathrm{out}} = \mathbf{x}_t^s$, where s, t are uniformly random chosen from $s \in [S]$ and $t \in [T]$.

In addition, the major difference between our algorithm and the SVRC algorithms proposed in Wang et al. (2018a); Zhang et al. (2018); Wang et al. (2018b) is that our algorithm uses a constant Hessian minibatch size instead of an adaptive one in each iteration, and thus the parameter tuning of our algorithm is much easier. In sharp contrast, the minibatch sizes of the stochastic Hessian in the algorithm proposed by Wang et al. (2018a); Zhang et al. (2018); Wang et al. (2018b) are dependent on both accuracy parameter ϵ and the current update $\mathbf{h}_t^s$, which make the update an implicit one and it is hard to tune such hyperparameters in practice.

7. Theoretical Analysis of Lite-SVRC

In this section, we present our theoretical results on the Hessian sample complexity of Lite-SVRC (Algorithm 2). Different from the analysis of SVRC in Section 4 which only requires the Hessian Lipschitz condition (Assumption 1), we will need additionally the following smoothness assumption for the analysis of Lite-SVRC:

Assumption 17 (Gradient Lipschitz) *There exists a constant $L_1 > 0$, such that for all $\mathbf{x}, \mathbf{y}$ and $i \in \{1, ..., n\}$*

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \leq L_1\|\mathbf{x} - \mathbf{y}\|_2.$$

Assumptions 1 and 17 are mild and widely used in the line of research for finding approximate global minima (Carmon and Duchi, 2016; Carmon et al., 2018; Agarwal et al., 2017; Wang et al., 2018a; Yu et al., 2018).

Recall the definition in (8), we need to upper bound $\mu(\mathbf{x}_{\text{out}})$ in order to find the approximate local minimum. The following theorem spells out the upper bound of $\mu(\mathbf{x}_{\text{out}})$.

Theorem 18 *Under Assumptions 1 and 17, suppose that $n > 10, M_{s,t} = C_M L_2$, $D_g \geq C_1 L_1^2/L_2^2 \cdot n^{4/3}/C_M$ and $B_h > 144(C_1 C_h)^{2/3} n^{2/3}/C_M^2$, where $C_h = 1200(\log d)$ and C_M, C_1 are absolute constants. Then the output $\mathbf{x}_{out}$ of Algorithm 2 satisfies*

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \leq \frac{216 C_M^2 L_2^{1/2} \Delta_F}{ST}. \tag{22}$$

Remark 19 *Theorem 18 suggests that with a fixed number of inner loops T, if we run Algorithm 2 for sufficiently large S epochs, then we have a point sequence $\mathbf{x}_i$ where $\mathbb{E}[\mu(\mathbf{x}_i)] \to 0$. That being said, $\mathbf{x}_i$ will converge to a local minimum, which is consistent with the convergence analysis in existing related work (Nesterov and Polyak, 2006; Kohler and Lucchi, 2017; Wang et al., 2018a).*

Now we provide a specific choice of parameters used in Theorem 18 to derive the total Hessian sample complexity of Algorithm 2.

Corollary 20 *Under the same assumptions as in Theorem 18, let batch size parameters satisfy $D_g = 4L_1^2/L_2^2 \cdot n^{4/3}$ and $B_h = \log d \cdot (C_h \cdot n)^{2/3}$. Set the inner loop parameter $T = n^{1/3}$ and cubic regularization parameter $M_{s,t} = C_M L_2$, where C_M is an absolute constant. Set the epoch number $S = O(\max\{L_2^{1/2}\Delta_F/(\epsilon^{3/2} n^{1/3}), 1\})$. Then the output $\mathbf{x}_{out}$ from Algorithm 2 is an $(\epsilon, \sqrt{L_2 \epsilon})$-approximate local minimum after*

$$\widetilde{O}\left(n + \frac{\Delta_F \sqrt{L_2}}{\epsilon^{3/2}} \cdot n^{2/3}\right) \text{ stochastic Hessian evaluations.} \tag{23}$$

Moreover, the total number of CSO calls of Algorithm 2 is

$$O\left(\frac{\Delta_F \sqrt{L_2}}{\epsilon^{3/2}}\right).$$

Remark 21 *Note that the CSO oracle complexity of Lite-SVRC is the same as SVRC. In what follows, we present a comprehensive comparison on Hessian sample complexity between our Lite-SVRC and other related algorithms in Table 2. The algorithm proposed in Wang et al. (2018a) has two versions: sample with replacement and sample without replacement. For the completeness, we present both versions in Wang et al. (2018a). From Table 2 we can see that Lite-SVRC strictly outperforms CR by a factor of $n^{1/3}$ and outperforms SVRC by a factor of $n^{2/15}$ in terms of Hessian sample complexity. Lite-SVRC also outperforms SCR when $\epsilon = O(n^{-2/3})$, which suggests that the variance reduction scheme makes Lite-SVRC perform better in the high accuracy regime. More importantly, our proposed Lite-SVRC does not rely on the assumption that the function F is Lipschitz continuous, which is required by the algorithm proposed in Wang et al. (2018a). In terms of Hessian sample complexity, our algorithm directly improves that of Wang et al. (2018a) by a factor of $n^{2/33}$. The Hessian sample complexity of Lite-SVRC is the same as that of the algorithms recently proposed in Wang et al. (2018b) and Zhang et al. (2018). Nevertheless, Lite-SVRC uses a constant Hessian sample batch size in contrast to the adaptive batch size as used in Wang et al. (2018b); Zhang et al. (2018), which makes the use of Lite-SVRC algorithm much simpler and more practical.*

Algorithm	Per-iteration	Total	Function Lipschitz	Gradient Lipschitz	Hessian Lipschitz
CR	$O(n)$	$O\left(\frac{n}{\epsilon^{3/2}}\right)$	No	No	Yes
SCR	$\widetilde{O}\left(\frac{1}{\epsilon}\right)$	$\widetilde{O}\left(\frac{1}{\epsilon^{5/2}}\right)$	No[3]	Yes	Yes
$\text{SVRC}_{\text{with}}$[4]	$\widetilde{O}\left(\frac{\lVert\mathbf{x}_t^s - \widetilde{\mathbf{x}}^s\rVert_2^2}{\lVert\mathbf{h}_t^s\rVert_2^2}\right)$	$\widetilde{O}\left(n + \frac{n^{3/4}}{\epsilon^{3/2}}\right)$	Yes	Yes	Yes
$\text{SVRC}_{\text{without}}$[4]	$\widetilde{O}\left(\left(\frac{1}{n} + \frac{\lVert\mathbf{h}_t^s\rVert_2^2}{\lVert\mathbf{x}_t^s - \widetilde{\mathbf{x}}^s\rVert_2^2}\right)^{-1}\right)$	$\widetilde{O}\left(n + \frac{n^{8/11}}{\epsilon^{3/2}}\right)$	Yes	Yes	Yes
$\text{SVRC}_{\text{Wang}}$	$\widetilde{O}\left(\frac{\lVert\mathbf{x}_t^s - \widetilde{\mathbf{x}}^s\rVert_2^2}{\max\{\epsilon, \lVert\mathbf{h}_t^s\rVert_2^2\}}\right)$	$\widetilde{O}\left(n + \frac{n^{2/3}}{\epsilon^{3/2}}\right)$	No	Yes	Yes
$\text{SVRC}_{\text{Zhang}}$	$\widetilde{O}\left(\frac{\lVert\mathbf{x}_t^s - \widetilde{\mathbf{x}}^s\rVert_2^2}{\epsilon}\right)$	$\widetilde{O}\left(n + \frac{n^{2/3}}{\epsilon^{3/2}}\right)$	No	Yes	Yes
SVRC[5] (Algorithm 1)	$\widetilde{O}(n^{4/5})$	$\widetilde{O}\left(n + \frac{n^{4/5}}{\epsilon^{3/2}}\right)$	No	No	Yes
Lite-SVRC (Algorithm 2)	$\widetilde{O}(n^{2/3})$	$\widetilde{O}\left(n + \frac{n^{2/3}}{\epsilon^{3/2}}\right)$	No	Yes	Yes

Table 2: Comparisons of per-iteration and total sample complexities of Hessian evaluations for different algorithms: CR (Nesterov and Polyak, 2006), SCR (Kohler and Lucchi, 2017; Xu et al., 2017a), $\text{SVRC}_{\text{with}}$ (Wang et al., 2018a), $\text{SVRC}_{\text{without}}$ (Wang et al., 2018a), $\text{SVRC}_{\text{Wang}}$ (Wang et al., 2018b), $\text{SVRC}_{\text{Zhang}}$ (Zhang et al., 2018), SVRC (Algorithm 1) and Lite-SVRC (Algorithm 2). Similar to Table 1, the CSO oracle complexities of all the methods being compared are the same, i.e. $O(1/\epsilon^{3/2})$. Therefore, we omit it for simplicity.

Recall the inexact cubic subproblem solver defined in Section 5. The same inexact CSO oracles can also be used in Algorithm 2. In what follows, we present the convergence result of Lite-SVRC with inexact CSO oracles.

3. Although the refined SCR in Xu et al. (2017b) does not need function Lipschitz, the original SCR in Kohler and Lucchi (2017) needs it.
4. In Wang et al. (2018a), both algorithms need to calculate $\lambda_{\min}(\nabla^2 F(\mathbf{x}_t^s))$ at each iteration to decide whether the algorithm should continue, which adds additional $O(n)$ Hessian sample complexity. We choose not to include this into the results in the table.
5. For SVRC (Algorithm 1), we present its second-order oracle calls derived in Section 4 as the Hessian sample complexity.

Theorem 22 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$ satisfying Condition 11. Then under the same conditions of Theorem 18, the output of Algorithm 2 satisfies*

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \leq \frac{216C_M^2 L_2^{1/2}\Delta_F}{ST} + 432C_M^2 L_2^{1/2}\delta. \tag{24}$$

In addition, Algorithm 2 with inexact oracle can also reduce the Hessian sample complexity, which is summarized in the following corollary.

Corollary 23 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$ satisfying Condition 11 with $\delta = (864C_M^2 L_2^{1/2})^{-1}\epsilon^{3/2}$. Then with the same choice of parameters in Corollary 20, Algorithm 2 will find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum within*

$$\widetilde{O}\bigg(n + \frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}} \cdot n^{2/3}\bigg) \text{ stochastic Hessian evaluations,}$$

and

$$O\bigg(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\bigg) \text{ CSO calls.}$$

8. Experiments

In this section, we conduct experiments on real world datasets to support our theoretical analysis of the proposed SVRC and Lite-SVRC algorithms. We investigate two nonconvex problems on three different datasets, *a9a*, *ijcnn1* and *covtype*, which are all common datasets used in machine learning and the sizes are summarized in Table 3.

8.1. Baseline Algorithms

To validate the superior performance of the proposed SVRC (Algorithm 1) in terms of second-order oracles, we compare it with the following baseline algorithms: (1) trust-region Newton methods (TR) (Conn et al., 2000); (2) Adaptive Cubic regularization (Cartis et al., 2011a,b); (3) Subsampled Cubic regularization (Kohler and Lucchi, 2017); (4) Gradient Cubic regularization (Carmon and Duchi, 2016) and (5) Stochastic Cubic regularization (Tripuraneni et al., 2018). To demonstrate the improvement of Lite-SVRC (Algorithm 2) on Hessian sample complexity, we further conduct experiments to compare Lite-SVRC with all the baselines above including SVRC. In addition, we also compare Lite-SVRC with (6) SVRC-without (Wang et al., 2018a), which focuses on reducing the Hessian sample complexity as well. In addition, there are two versions of SVRC in Wang et al. (2018a), but the one based on sampling without replacement performs better in both theory and experiments. We therefore only compare with this one. Note that the SVRC algorithms in Wang et al. (2018b); Zhang et al. (2018) are essentially the same as our Lite-SVRC algorithm, except in the choice of batch size for stochastic Hessian. Thus we do not compare our Lite-SVRC with these algorithms (Wang et al., 2018b; Zhang et al., 2018).

Dataset	sample size n	dimension d
a9a	32,561	123
covtype	581,012	54
ijcnn1	35,000	22

Table 3: Datasets used in experiments.

8.2. Implementation Details

For Subsampled Cubic and SVRC-without, the sample size B_k is dependent on $\|\mathbf{h}_k\|_2$ (Kohler and Lucchi, 2017) and B_h is dependent on $\|\mathbf{h}_t^s\|_2$ (Wang et al., 2018a), which make these two algorithms implicit algorithms. To address this issue, we follow the suggestion in Kohler and Lucchi (2017); Wang et al. (2018a) and use $\|\mathbf{h}_{k-1}\|_2$ and $\|\mathbf{h}_{t-1}^s\|_2$ instead of $\|\mathbf{h}_k\|_2$ and $\|\mathbf{h}_t^s\|_2$. Furthermore, we choose the penalty parameter $M_{s,t}$ for SVRC, SVRC-without and Lite-SVRC as constants which are suggested by the original papers of these algorithms. Finally, to solve the CR sub-problem in each iteration, we choose to solve the sub-problem approximately in the Krylov subspace spanned by Hessian related vectors, as used by Kohler and Lucchi (2017).

8.3. Nonconvex Optimization Problems

In this subsection, we formulate the nonconvex optimization problems that will be studied in our experiments. In particular, we choose two nonconvex regression problem as our objectives with the following nonconvex regularizer

$$g(\lambda, \gamma, \mathbf{x}) = \lambda \cdot \sum_{i=1}^{d} \frac{(\gamma x_i)^2}{1 + (\gamma x_i)^2}, \tag{25}$$

where λ, γ are the control parameters and x_i is the i-th coordinate of $\mathbf{x}$. λ and γ are set differently for each dataset. This regularizer has been widely used in nonconvex regression problem, which can be regarded as a special example of robust nonlinear regression (Reddi et al., 2016b; Kohler and Lucchi, 2017; Wang et al., 2018a).

8.3.1. Logistic Regression with Nonconvex Regularizer

The first problem is a binary logistic regression problem with a nonconvex regularizer g (Reddi et al., 2016b). Given training data $\mathbf{x}_i \in \mathbb{R}^d$ and label $y_i \in \{0, 1\}$, $1 \leq i \leq n$, our goal is to solve the following optimization problem:

$$\min_{\mathbf{s}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \big[y_i \cdot \log\phi(\mathbf{s}^\top\mathbf{x}_i) + (1 - y_i) \cdot \log[1 - \phi(\mathbf{s}^\top\mathbf{x}_i)]\big] + g(\lambda, \gamma, \mathbf{s}), \tag{26}$$

where $\phi(x) = 1/(1 + \exp(-x))$ is the sigmoid function and g is defined in (25).

8.3.2. Nonlinear Least Square with Nonconvex Regularizer

Another problem is the nonlinear least square problem with a nonconvex regularizer $g(\lambda, \gamma, \mathbf{x})$ defined in (25). The nonlinear least square problem is also studied in Xu et al. (2017b).

Given training data $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, $1 \le i \le n$, our goal is to minimize the following problem

$$\min_{\mathbf{s}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \left[y_i - \phi(\mathbf{s}^\top \mathbf{x}_i)\right]^2 + g(\lambda, \gamma, \mathbf{s}). \tag{27}$$

Here $\phi(x) = 1/(1 + \exp(-x))$ is again the sigmoid function and g is defined in (25).

8.4. Experimental Results for SVRC

In this subsection, we present the experimental results for SVRC compared with baseline algorithms (1)-(5) listed in Section 8.1. Here, we fix $\lambda = 10$ and $\gamma = 1$ of the nonconvex regularizer g in (25) for both the logistic regression and the nonlinear least square problems.

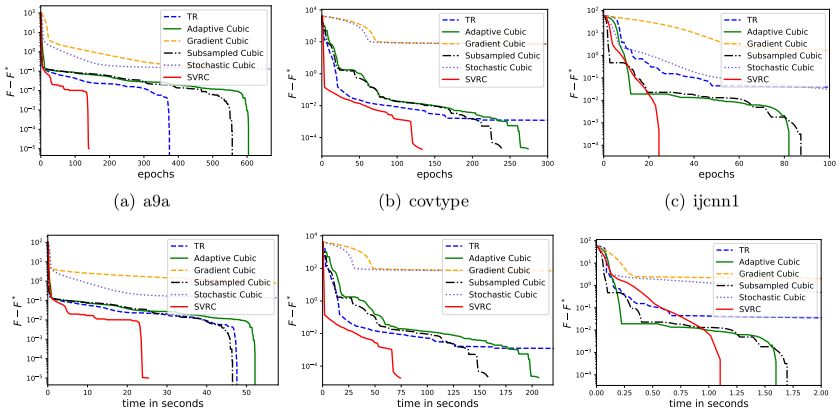
Calculation for SO calls: For *Subsampled Cubic*, each loop takes $(B_g + B_h)$ SO calls, where B_g and B_h are the subsampling sizes of gradient and Hessian. For *Stochastic Cubic*, each loop costs $(n_g + n_h)$ SO calls, where n_g and n_h denote the subsampling sizes of gradient and Hessian-vector operator. *Gradient Cubic*, *Adaptive Cubic* and *TR* cost n SO calls in each loop. We define the amount of epochs to be the amount of SO calls divided by n.

Parameters: For each algorithm and each dataset, we choose different b_g, b_h, T for the best performance. Meanwhile, we choose the cubic regularization parameter as $M_{s,t} = \alpha/(1+\beta)^{(s+t/T)}$, α, $\beta > 0$ for each iteration. When $\beta = 0$, it has been proved to enjoy good convergence performance. This choice of parameter is similar to the choice of penalty parameter in *Subsampled Cubic* and *Adaptive Cubic*, which sometimes makes some algorithms behave better in our experiments.

Subproblem Solver: With regard to the cubic subproblem solver for solving (7), we choose the Lanczos-type method used in Cartis et al. (2011a), which finds the global minimizer $\mathbf{h}_t^s$ of $m_t^s(\mathbf{h})$ in a Krylov subspace $\mathcal{K}_l = \text{span}\{\mathbf{v}_t^s, \mathbf{U}_t^s\mathbf{v}_t^s, (\mathbf{U}_t^s)^2\mathbf{v}_t^s, \ldots, (\mathbf{U}_t^s)^{l-1}\mathbf{v}_t^s\}$, where $l \ll d$ is the dimension of $\mathcal{K}_l$ and can be selected manually or adaptively (Cartis et al., 2011a; Kohler and Lucchi, 2017). The computational complexity of Lanczos-type method consists of two parts according to Carmon and Duchi (2018). First, $(l-1)$ matrix-vector products are performed to calculate the basis of $\mathcal{K}_l$, whose computational complexity is $O(d^2 l)$. Second, the minimizer of $m_t^s(\mathbf{h})$ is computed in subspace $\mathcal{K}_l$, whose computational complexity is $O(l \log l)$. Thus, the total computational complexity of Lanczos-type method is $O(d^2 l)$.

At each iteration, SVRC needs to compute the semi-stochastic gradient $\mathbf{v}_t^s$ and Hessian $\mathbf{U}_t^s$, which costs $O(db_g + d^2 b_h)$ computational complexity for both nonconvex regularized logistic regression and nonlinear least square problems, where b_g and b_h are the mini-batch sizes of stochastic gradient and Hessian respectively. Putting these pieces together, the per-iteration complexity of SVRC is $O(db_g + d^2 b_h + d^2 l)$, and the total computational complexity of SVRC is $O(ST(db_g + d^2 b_h + d^3))$, where S is the number of epochs and T is the length of epoch.

For the binary logistic regression problem in (26), the parameters of $M_{s,t} = \alpha/(1+\beta)^{(s+t/T)}, \alpha, \beta > 0$ are set as follows: $\alpha = 0.05, \beta = 0$ for *a9a* and *ijcnn1* datasets and $\alpha = 5e3, \beta = 0.15$ for *covtype*. The experimental results are shown in Figure 1. For the non-linear least squares problem in (27), we set $\alpha = 0.05, 1e8, 0.003$ and $\beta = 0, 1, 0.5$ for *a9a*, *covtype* and *ijcnn1* datasets respectively. The experimental results are shown in Figure 2. From both Figures 1 and 2, we can see that SVRC outperforms all the other baseline algorithms on all the datasets. The only exception happens in the non-linear least square problem on the *covtype* dataset, where our algorithm behaves a little worse than Adaptive Cubic at the high accuracy regime in terms of epoch counts. However, under this setting, our algorithm still outperforms the other baselines in terms of the CPU time.



(a) a9a (b) covtype (c) ijcnn1

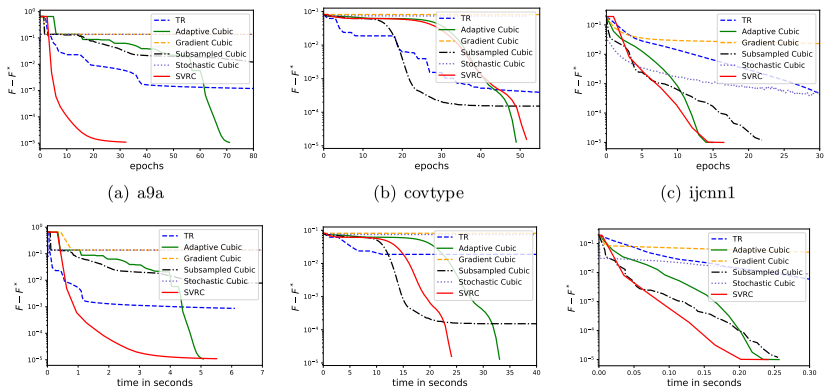
(d) a9a (e) covtype (f) ijcnn1

Figure 1: Logarithmic function value gap for nonconvex regularized logistic regression on different datasets. (a), (b) and (c) present the oracle complexity comparison; (d), (e) and (f) present the runtime comparison.

8.5. Experimental Results for Lite-SVRC

In this subsection, we present the experimental results for Lite-SVRC compared with all the baselines listed in Section 8.1. For Lite-SVRC, we use the same cubic subproblem solver used for SVRC in the previous subsection.

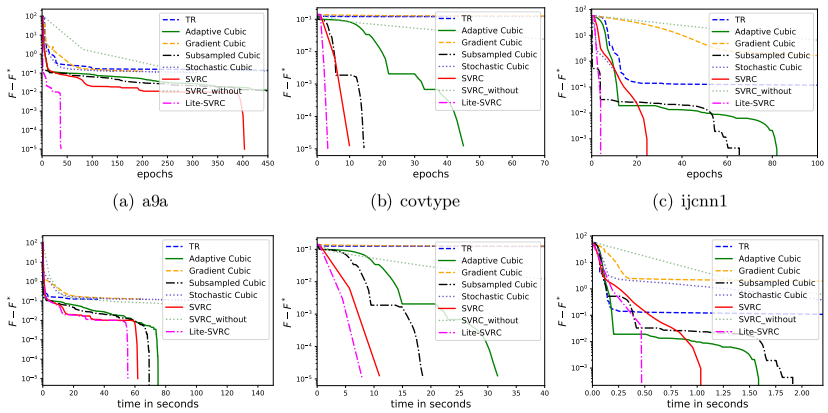
In the binary logistic regression problem in (26), for the nonconvex regularizer g in (25), we set $\lambda = 10^{-3}$ for all three datasets, and set $\gamma = 10, 50, 100$ for *a9a*, *ijcnn1* and *covtype* datasets respectively. The experimental results are displayed in Figure 3. The first row of the figure shows the plots of function value gap v.s. Hessian sample complexity of all the compared algorithms, and the second row presents the plots of function value gap v.s. CPU runtime (in seconds) of all the algorithms. It can be seen from Figure 3 that Lite-SVRC performs the best among all algorithms regarding both sample complexity of Hessian and runtime on all three datasets, which is consistent with our theoretical analysis. We remark that SVRC performs the second best in most settings in terms of both Hessian sample complexity and runtime. It should also be noted that although SVRC-without is also a variance-reduced method similar to Lite-SVRC and SVRC, it indeed performs much worse



(a) a9a (b) covtype (c) ijcnn1

(d) a9a (e) covtype (f) ijcnn1

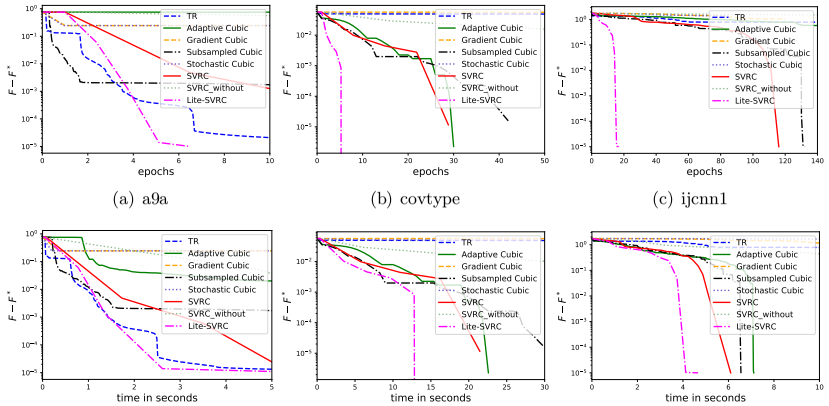
Figure 2: Logarithmic function value gap for nonlinear least square on different datasets. (a), (b) and (c) present the oracle complexity comparison; (d), (e) and (f) present the runtime comparison.



(a) a9a (b) covtype (c) ijcnn1

(d) a9a (e) covtype (f) ijcnn1

Figure 3: Function value gap of different algorithms for nonconvex regularized logistic regression problems on different datasets. (a)-(c) are plotted w.r.t. Hessian sample complexity. (d)-(e) are plotted w.r.t. CPU runtime.



(a) a9a (b) covtype (c) ijcnn1

(d) a9a (e) covtype (f) ijcnn1

Figure 4: Function value gap of different algorithms for nonlinear least square problems on different datasets. (a)-(c) are plotted w.r.t. Hessian sample complexity. (d)-(e) are plotted w.r.t. CPU runtime.

than other methods, because as we pointed out in the introduction, it needs to compute the minimum eigenvalue of the Hessian in each iteration, which actually makes the Hessian sample complexity even worse than Subsampled Cubic, let alone the runtime complexity.

For the least square problem in (27), the parameters λ and γ in the nonconvex regularizer for different datasets are set as follows: $\lambda = 5 \times 10^{-3}$ for all three datasets, and $\gamma = 10, 20, 50$ for *a9a*, *ijcnn1* and *covtype* datasets respectively. The experimental results are summarized in Figure 4, where the first row shows the plots of function value gap v.s. Hessian sample complexity and the second row presents the plots of function value gap v.s. CPU runtime (in seconds). It can be seen that Lite-SVRC again achieves the best performance among all the algorithms regarding to both sample complexity of Hessian and runtime when the required precision is high, which supports our theoretical analysis.

9. Conclusions

In this paper, we propose two novel second-order algorithms for non-convex optimization: SVRC and Lite-SVRC. Our proposed algorithm SVRC is the first algorithm which improves the oracle complexity of cubic regularization and its subsampled variants under certain regime using variance reduction techniques. We also show that similar oracle complexity also holds with inexact oracles. Under both exact and inexact oracle settings our algorithm outperforms the state-of-the-art methods. Furthermore, our proposed algorithm Lite-SVRC achieves a lower sample complexity of Hessian compared with SVRC and existing variance reduction based cubic regularization algorithms. Extensive experiments on various nonconvex optimization problems and datasets validate our theory.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments. This research was sponsored in part by the National Science Foundation IIS-1904183 and IIS-1906169. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

Appendix A. Proof of Main Theoretical Results for SVRC

In this section, we present the proofs of our main theoretical results for SVRC. Let us first recall the notations used in Algorithm 1. $\mathbf{v}_t^s$ and $\mathbf{U}_t^s$ are the semi-stochastic gradient and Hessian defined in (5) and (6) respectively. $\mathbf{x}_t^s$'s are the iterates and $\widehat{\mathbf{x}}^s$'s are the reference points used in Algorithm 1. b_g and b_h are the batch sizes of semi-stochastic gradient and Hessian. S and T are the number of epochs and epoch length of Algorithm 1. We set $M_{s,t} := M = C_M L_2$ as suggested by Theorems 6 and 13, where $C_M > 0$ is a constant. $\mathbf{h}_t^s$ is the exact minimizer of $m_t^s(\mathbf{h})$, where $m_t^s(\mathbf{h})$ is defined in (7). $\widetilde{\mathbf{h}}_t^s$ is the inexact minimizer defined in (15).

In order to prove Theorems 6 and 13, we lay down the following useful technical lemmas. The first lemma is standard in the analysis cubic regularization methods.

Lemma 24 *Suppose F is L_2-Hessian Lipschitz for some constant $L_2 > 0$. For the semi-stochastic gradient and Hessian defined in* (5) *and* (6)*, we have the following results:*

$$\mathbf{v}_t^s + \mathbf{U}_t^s \mathbf{h}_t^s + \frac{M}{2}\|\mathbf{h}_t^s\|_2 \mathbf{h}_t^s = 0, \tag{28}$$

$$\mathbf{U}_t^s + \frac{M}{2}\|\mathbf{h}_t^s\|_2 \mathbf{I} \succeq 0, \tag{29}$$

$$\langle \mathbf{v}_t^s, \mathbf{h}_t^s\rangle + \frac{1}{2}\langle \mathbf{U}_t^s \mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{M}{6}\|\mathbf{h}_t^s\|_2^3 \le -\frac{M}{12}\|\mathbf{h}_t^s\|_2^3. \tag{30}$$

The next two important lemmas control the variances of $\mathbf{v}_t^s$ and $\mathbf{U}_t^s$.

Lemma 25 *For the semi-stochastic gradient $\mathbf{v}_t^s$ defined in* (5)*, we have*

$$\mathbb{E}_{i_t}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \le \frac{L_2^{3/2}}{b_g^{3/4}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where $\mathbb{E}_{i_t}$ is the expectation over all $i_t \in I_g$.

Lemma 26 *Let $\mathbf{U}_t^s$ be the semi-stochastic Hessian defined in* (6)*. If the batch size satisfy $b_h \ge 400\log d$, then we have*

$$\mathbb{E}_{j_t}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 \le 1200 L_2^3 \left(\frac{\log d}{b_h}\right)^{3/2}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where $\mathbb{E}_{j_t}$ is the expectation over all $j_t \in I_h$.

Lemma 27 *For the semi-stochastic gradient and Hessian defined in* (5) *and* (6) *and $\mathbf{h} \in \mathbb{R}^d$, we have*

$$\langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \mathbf{h}\rangle \le \frac{M}{27}\|\mathbf{h}\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}},$$
$$\big\langle \big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}, \mathbf{h}\big\rangle \le \frac{2M}{27}\|\mathbf{h}\|_2^3 + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3.$$

Lemma 28 *Let $\mathbf{h} \in \mathbb{R}^d$ and $C_M \ge 100$. For the semi-stochastic gradient and Hessian defined in* (5) *and* (6)*, we have*

$$\mu(\mathbf{x}_t^s + \mathbf{h}) \le 9C_M^{3/2}\Big[M^{3/2}\|\mathbf{h}\|_2^3 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 + \|\nabla m_t^s(\mathbf{h})\|_2^{3/2} + M^{3/2}\big|\|\mathbf{h}\|_2 - \|\mathbf{h}_t^s\|_2\big|^3\Big].$$

Lemma 29 *Let $\mathbf{h} \in \mathbb{R}^d$ and $C \ge 3/2$, For the semi-stochastic gradient and Hessian defined in* (5) *and* (6)*, we have*

$$\|\mathbf{x}_t^s + \mathbf{h} - \widehat{\mathbf{x}}^s\|_2^3 \le 2C^2\|\mathbf{h}\|_2^3 + (1 + 3/C)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \tag{31}$$

Lemma 30 *We define constant series c_t for $0 \le t \le T$ as follows: $c_T = 0$ and $c_t = c_{t+1}(1 + 3/T) + M(500T^3)^{-1}$ for $0 \le t \le T-1$. Then we have for any $1 \le t \le T$,*

$$M/24 - 2c_t T^2 \ge 0. \tag{32}$$

A.1. Proof of Theorem 6

Proof [Proof of Theorem 6] We first upper bound $F(\mathbf{x}_{t+1}^s)$ as follows

$$\begin{aligned}
F(\mathbf{x}_{t+1}^s) &\le F(\mathbf{x}_t^s) + \langle \nabla F(\mathbf{x}_t^s), \mathbf{h}_t^s\rangle + \frac{1}{2}\big\langle \nabla^2 F(\mathbf{x}_t^s)\mathbf{h}_t^s, \mathbf{h}_t^s\big\rangle + \frac{L_2}{6}\|\mathbf{h}_t^s\|_2^3 \\
&= F(\mathbf{x}_t^s) + \langle \mathbf{v}_t^s, \mathbf{h}_t^s\rangle + \frac{1}{2}\langle \mathbf{U}_t^s\mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{M}{6}\|\mathbf{h}_t^s\|_2^3 + \langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \mathbf{h}_t^s\rangle \\
&\quad + \frac{1}{2}\big\langle \big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}_t^s, \mathbf{h}_t^s\big\rangle - \frac{M - L_2}{6}\|\mathbf{h}_t^s\|_2^3 \\
&\le F(\mathbf{x}_t^s) - \frac{M}{12}\|\mathbf{h}_t^s\|_2^3 + \left(\frac{M}{27}\|\mathbf{h}_t^s\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}}\right) \\
&\quad + \frac{1}{2}\left(\frac{2M}{27}\|\mathbf{h}_t^s\|_2^3 + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\right) - \frac{M - L_2}{6}\|\mathbf{h}_t^s\|_2^3 \\
&\le F(\mathbf{x}_t^s) - \frac{M}{12}\|\mathbf{h}_t^s\|_2^3 + \frac{2}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3,
\end{aligned} \tag{33}$$

where the first inequality follows from Lemma 40 and the second inequality holds due to Lemmas 24 and 27. We define

$$R_t^s = \mathbb{E}\big[F(\mathbf{x}_t^s) + c_t\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big], \tag{34}$$

where c_t is defined in Lemma 30. Then by Lemma 29, for $T \geq 3/2$ we have

$$c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 = c_{t+1}\|\mathbf{h}_t^s + \mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \leq 2c_{t+1}T^2\|\mathbf{h}_t^s\|_2^3 + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \quad (35)$$

Applying Lemma 28 with $\mathbf{h} = \mathbf{h}_t^s$, we have

$$\begin{aligned}\big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s) &\leq \frac{M}{24}\|\mathbf{h}_t^s\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3}{24M^2} \\ &\quad + \frac{\|\nabla m_t^s(\mathbf{h}_t^s)\|_2^{3/2}}{24M^{1/2}} + \frac{M}{24}\big|\|\mathbf{h}_t^s\|_2 - \|\mathbf{h}_t^s\|_2\big|^3 \\ &= \frac{M}{24}\|\mathbf{h}_t^s\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3}{24M^2}, \quad (36)\end{aligned}$$

where the equality is due to Lemma 24. Adding (33) with (35) and (36) and taking total expectation, we have

$$\begin{aligned}&R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \\ &= \mathbb{E}\Big[F(\mathbf{x}_{t+1}^s) + c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s)\Big] \\ &\leq \mathbb{E}\big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 - \|\mathbf{h}_t^s\|_2^3(M/24 - 2c_{t+1}T^2)\big] \\ &\quad + \mathbb{E}\Big[3M^{-1/2}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + 28M^{-2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\Big] \\ &\leq \mathbb{E}\big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] + \mathbb{E}\Big[3M^{-1/2}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \\ &\quad + 28M^{-2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\Big], \quad (37)\end{aligned}$$

where the third inequality holds due to Lemma 30. To further bound (37), we have

$$\frac{3}{M^{1/2}}\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \leq \frac{3L_2^{3/2}}{M^{1/2}b_g^{3/4}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \leq \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \quad (38)$$

where the first inequality holds due to Lemma 25, the second inequality holds due to $M \geq 100L_2$ and $b_g \geq 5T^4$ from the condition of Theorem 6. We also have

$$\frac{28}{M^2}\mathbb{E}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 \leq \frac{28 \times 15000L_2^3}{M^2(b_h/\log d)^{3/2}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \leq \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \quad (39)$$

where the first inequality holds due to Lemma 26, where we have $b_h \geq 100T^2\log d \geq 400\log d$, and the second inequality holds due to $M \geq 100L_2$ and $b_h \geq 100T^2\log d$ from the assumption of Theorem 6. Thus, submitting (38) and (39) into (37), we have

$$\begin{aligned}R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] &\leq \mathbb{E}\bigg[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\bigg(c_{t+1}(1+3/T) + \frac{M}{500T^3}\bigg)\bigg] \\ &= \mathbb{E}\big[F(\mathbf{x}_t^s) + c_t\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] \\ &= R_t^s, \quad (40)\end{aligned}$$

where the first equality holds due to the definition of c_t in Lemma 30. Telescoping (40) from $t = 0$ to $T-1$, we have

$$R_0^s - R_T^s \geq \sum_{t=1}^T \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_t^s)].$$

By the definition of c_T in Lemma 30, we have $c_T = 0$, then $R_T^s = \mathbb{E}\big[F(\mathbf{x}_T^s) + c_T\|\mathbf{x}_T^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}F(\widehat{\mathbf{x}}^{s+1})$; meanwhile by the definition of $\mathbf{x}_0^s$, we have $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$. Thus we have $R_0^s = \mathbb{E}\big[F(\mathbf{x}_0^s) + c_0\|\mathbf{x}_0^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}F(\widehat{\mathbf{x}}^s)$, which implies

$$\mathbb{E}F(\widehat{\mathbf{x}}^s) - \mathbb{E}F(\widehat{\mathbf{x}}^{s+1}) = R_0^s - R_T^s \geq \big(240C_M^2L_2^{1/2}\big)^{-1}\sum_{t=1}^T\mathbb{E}[\mu(\mathbf{x}_t^s)]. \quad (41)$$

Finally, telescoping (41) from $s = 1$ to S yields

$$\Delta_F \geq \sum_{s=1}^S \mathbb{E}F(\widehat{\mathbf{x}}^s) - \mathbb{E}F(\widehat{\mathbf{x}}^{s+1}) \geq \big(240C_M^2L_2^{1/2}\big)^{-1}\sum_{s=1}^S\sum_{t=1}^T\mathbb{E}[\mu(\mathbf{x}_t^s)].$$

By the definition about choice of $\mathbf{x}_{\text{out}}$, we complete the proof. ■

A.2. Proof of Corollary 9

Proof We can verify that the parameter setting in Corollary 9 satisfies the requirement of Theorem 6. Thus, submitting the choice of parameters into Theorem 6, the output of Algorithm 1 $\mathbf{x}_{\text{out}}$ satisfies that

$$\mathbb{E}[\mu(\mathbf{x}_{\text{out}})] \leq \frac{240C_M^2L_2^{1/2}\Delta_F}{ST} \leq \epsilon^{3/2}, \quad (42)$$

which indeed implies that $\mathbf{x}_{\text{out}}$ is an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum. Next we calculate how many SO calls and CSO calls are needed. Algorithm 1 needs to calculate full gradient $\mathbf{g}_s$ and full Hessian $\mathbf{H}_s$ at the beginning of each epoch, with n SO calls. In each epoch, Algorithm 1 needs to calculate $\mathbf{v}_t^s$ and $\mathbf{U}_t^s$ with $b_g + b_h$ SO calls at each iteration. Thus, the total amount of SO calls is

$$\begin{aligned}Sn + (ST)(b_g + b_h) &\leq n + C_1\Delta_F L_2^{1/2}n^{4/5}\epsilon^{-3/2} + C_1\Delta_F L_2^{1/2}\epsilon^{-3/2}(5n^{4/5} + 1000n^{2/5}\log d) \\ &= \widetilde{O}\bigg(n + \frac{\Delta_F\sqrt{L_2}n^{4/5}}{\epsilon^{3/2}}\bigg),\end{aligned}$$

where $C_1 = 240C_M^2$. For the CSO calls, Algorithm 1 needs to solve cubic subproblem at each single iteration. Thus, the total amount of CSO calls is

$$ST \leq C_1\Delta_F L_2^{1/2}\epsilon^{-3/2} = O\bigg(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\bigg).$$

■

A.3. Proof of Theorem 13

Proof [Proof of Theorem 13] Similar to (33) in the proof of Theorem 6, we have

$$\begin{aligned}
F(\mathbf{x}_{t+1}^s) &\le F(\mathbf{x}_t^s) + \langle \nabla F(\mathbf{x}_t^s), \widetilde{\mathbf{h}}_t^s\rangle + \frac{1}{2}\langle \nabla^2 F(\mathbf{x}_t^s)\widetilde{\mathbf{h}}_t^s, \widetilde{\mathbf{h}}_t^s\rangle + \frac{L_2}{6}\|\widetilde{\mathbf{h}}_t^s\|_2^3 \\
&= F(\mathbf{x}_t^s) + \langle \mathbf{v}_t^s, \widetilde{\mathbf{h}}_t^s\rangle + \frac{1}{2}\langle \mathbf{U}_t^s\widetilde{\mathbf{h}}_t^s, \widetilde{\mathbf{h}}_t^s\rangle + \frac{M}{6}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \widetilde{\mathbf{h}}_t^s\rangle \\
&\quad + \frac{1}{2}\langle (\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s)\widetilde{\mathbf{h}}_t^s, \widetilde{\mathbf{h}}_t^s\rangle - \frac{M-L_2}{6}\|\widetilde{\mathbf{h}}_t^s\|_2^3 \\
&\le F(\mathbf{x}_t^s) - \frac{M}{12}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \delta + \left(\frac{M}{27}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}}\right) \\
&\quad + \frac{1}{2}\left(\frac{2M}{27}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \frac{27}{M^2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\right) - \frac{M-L_2}{6}\|\widetilde{\mathbf{h}}_t^s\|_2^3 \\
&\le F(\mathbf{x}_t^s) - \frac{M}{12}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \frac{2}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{27}{M^2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3 + \delta,
\end{aligned} \tag{43}$$

where the second inequality holds because $\widetilde{\mathbf{h}}_t^s$ is an inexact solver satisfying Condition 11. By Lemma 29 with $\mathbf{h} = \widetilde{\mathbf{h}}_t^s$, we have

$$c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 = c_{t+1}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s + \widetilde{\mathbf{h}}_t^s\|_2^3 \le 2c_{t+1}T^2\|\widetilde{\mathbf{h}}_t^s\|_2^3 + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \tag{44}$$

By Lemma 28, we also have

$$\begin{aligned}
&\big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s) \\
&= \big(240C_M^2L_2^{1/2}\big)^{-1}\mu\big(\mathbf{x}_t^s + \widetilde{\mathbf{h}}_t^s\big) \\
&\le \frac{M}{24}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3}{24M^2} + \frac{\|\nabla m_t^s(\widetilde{\mathbf{h}}_t^s)\|_2^{3/2}}{24M^{1/2}} + \frac{M\big|\|\widetilde{\mathbf{h}}_t^s\|_2 - \|\mathbf{h}_t^s\|_2\big|^3}{24},
\end{aligned} \tag{45}$$

Since $\widetilde{\mathbf{h}}_t^s$ is an inexact solver satisfying Condition 11, we have

$$\frac{\|\nabla m_t^s(\widetilde{\mathbf{h}}_t^s)\|_2^{3/2}}{24M^{1/2}} + \frac{M\big|\|\widetilde{\mathbf{h}}_t^s\|_2 - \|\mathbf{h}_t^s\|_2\big|^3}{24} \le \frac{\delta}{24} + \frac{\delta}{24} < \delta. \tag{46}$$

Submitting (46) into (45), we have

$$\big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s) \le \frac{M}{24}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3}{24M^2} + \delta. \tag{47}$$

Then adding (43), (44) and (47) up, we have

$$\begin{aligned}
&R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \\
&= \mathbb{E}\Big[F(\mathbf{x}_{t+1}^s) + c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s)\Big] \\
&\le \mathbb{E}\Big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 - \|\widetilde{\mathbf{h}}_t^s\|_2^3\big(M/24 - 2c_{t+1}T^2\big)\Big] \\
&\quad + \mathbb{E}\left[\frac{3}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{28}{M^2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\right] + 2\delta \\
&\le \mathbb{E}\big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] \\
&\quad + \mathbb{E}\left[\frac{3}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{28}{M^2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\right] + 2\delta.
\end{aligned} \tag{48}$$

Since the parameter setting is the same as Theorem 6, by (38) and (39), we have

$$\frac{3}{M^{1/2}}\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \le \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \tag{49}$$

$$\frac{28}{M^2}\mathbb{E}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3 \le \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \tag{50}$$

Submitting (49) and (50) into (48) yields

$$\begin{aligned}
R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] &\le \mathbb{E}\left[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\left(c_{t+1}(1+3/T) + \frac{M}{500T^3}\right)\right] + 2\delta \\
&= \mathbb{E}\big[F(\mathbf{x}_t^s) + c_t\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] + 2\delta \\
&= R_t^s + 2\delta,
\end{aligned} \tag{51}$$

where the first equality holds due to the definition of c_t in Lemma 30. Telescoping (40) from $t = 0$ to $T-1$, we have

$$R_0^s - R_T^s \ge \sum_{t=1}^T \Big(\big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_t^s)] - 2\delta\Big).$$

By the definition of c_T in Lemma 30, we have $c_T = 0$, then $R_T^s = \mathbb{E}\big[F(\mathbf{x}_T^s) + c_T\|\mathbf{x}_T^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}[F(\widehat{\mathbf{x}}^{s+1})]$; meanwhile by the definition of $\mathbf{x}_0^s$, we have $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$. Thus we have $R_0^s = \mathbb{E}\big[F(\mathbf{x}_0^s) + c_0\|\mathbf{x}_0^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}[F(\widehat{\mathbf{x}}^s)]$, which further implies

$$\mathbb{E}[F(\widehat{\mathbf{x}}^s)] - \mathbb{E}[F(\widehat{\mathbf{x}}^{s+1})] = R_0^s - R_T^s \ge \sum_{t=1}^T \Big(\big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_t^s)] - 2\delta\Big). \tag{52}$$

Finally, telescoping (52) from $s = 1$ to S, we obtain

$$\Delta_F \ge \sum_{s=1}^S \mathbb{E}[F(\widehat{\mathbf{x}}^s)] - \mathbb{E}[F(\widehat{\mathbf{x}}^{s+1})] \ge \sum_{s=1}^S\sum_{t=1}^T \Big[\big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_t^s)] - 2\delta\Big].$$

By the definition about choice of $\mathbf{x}_{\text{out}}$, we finish the proof. ■

A.4. Proof of Corollary 15

Proof [Proof of Corollary 15] Under the parameter choice in Corollary 15, it holds that

$$\mathbb{E}[\mu(\mathbf{x}_{\text{out}})] \le \frac{240C_M^2 L_2^{1/2}\Delta_F}{ST} + 480C_M^2 L_2^{1/2}\delta \le \epsilon^{3/2}/2 + \epsilon^{3/2}/2 = \epsilon^{3/2}. \tag{53}$$

Thus, $\mathbf{x}_{\text{out}}$ is an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum. By the proof of Corollary 9, the total amount of SO calls is

$$\begin{aligned} Sn + (ST)(b_g + b_h) &\le n + C_1\Delta_F L_2^{1/2} n^{4/5}\epsilon^{-3/2} + C_1\Delta_F L_2^{1/2}\epsilon^{-3/2}(5n^{4/5} + 1000n^{2/5}\log d) \\ &= \widetilde{O}\bigg(n + \frac{\Delta_F\sqrt{L_2}n^{4/5}}{\epsilon^{3/2}}\bigg), \end{aligned}$$

where $C_1 = 480C_M^2$. For the CSO calls, Algorithm 1 needs to solve cubic subproblem at each single iteration. Thus, the total amount of CSO calls is

$$ST \le C_1\Delta_F L_2^{1/2}\epsilon^{-3/2} = O\bigg(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\bigg).$$

■

Appendix B. Proof of Technical Lemmas in Appendix A

In this section, we prove the technical lemmas used in Appendix A.

B.1. Proof of Lemma 24

The result of Lemma 24 is typical in the literature of cubic regularization (Nesterov and Polyak, 2006; Cartis et al., 2011a,b), but no exactly the same result has been shown in any formal way. Thus we present the proof here for self-containedness.

Proof [Proof of Lemma 24] For simplicity, we let $\mathbf{g} = \mathbf{v}_t^s, \mathbf{H} = \mathbf{U}_t^s, \theta = M_t$ and $\mathbf{h}_{\text{opt}} = \mathbf{h}_t^s$. Then we need to prove

$$\mathbf{g} + \mathbf{H}\mathbf{h}_{\text{opt}} + \frac{\theta}{2}\|\mathbf{h}_{\text{opt}}\|_2\mathbf{h}_{\text{opt}} = \mathbf{0}, \tag{54}$$

$$\mathbf{H} + \frac{\theta}{2}\|\mathbf{h}_{\text{opt}}\|_2\mathbf{I} \succeq \mathbf{0}, \tag{55}$$

$$\langle\mathbf{g}, \mathbf{h}_{\text{opt}}\rangle + \frac{1}{2}\langle\mathbf{H}\mathbf{h}_{\text{opt}}, \mathbf{h}_{\text{opt}}\rangle + \frac{\theta}{6}\|\mathbf{h}_{\text{opt}}\|_2^3 \le -\frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3. \tag{56}$$

Let $\lambda = \theta\|\mathbf{h}_{\text{opt}}\|_2/2$. Note that $\mathbf{h}_{\text{opt}} = \operatorname{argmin} m(\mathbf{h})$, then the necessary condition $\nabla m(\mathbf{h}_{\text{opt}}) = \mathbf{0}$ and $\nabla^2 m(\mathbf{h}_{\text{opt}}) \succeq \mathbf{0}$ can be written as

$$\nabla m(\mathbf{h}_{\text{opt}}) = \mathbf{g} + \mathbf{H}\mathbf{h}_{\text{opt}} + \lambda\mathbf{h}_{\text{opt}} = \mathbf{0}, \tag{57}$$

$$\mathbf{w}^\top\nabla^2 m(\mathbf{h}_{\text{opt}})\mathbf{w} = \mathbf{w}^\top\bigg(\mathbf{H} + \lambda\mathbf{I} + \lambda\bigg(\frac{\mathbf{h}_{\text{opt}}}{\|\mathbf{h}_{\text{opt}}\|_2}\bigg)\bigg(\frac{\mathbf{h}_{\text{opt}}}{\|\mathbf{h}_{\text{opt}}\|_2}\bigg)^\top\bigg)\mathbf{w} \ge 0, \forall\mathbf{w} \in \mathbb{R}^d. \tag{58}$$

Apparently, (57) directly implies (54). To prove (55), we adapt the proof of Lemma 5.1 in Agarwal et al. (2017). Note that if $\langle\mathbf{w}, \mathbf{h}_{\text{opt}}\rangle = 0$, then (58) directly implies (55). So we only need to focus on the case that $\langle\mathbf{w}, \mathbf{h}_{\text{opt}}\rangle \ne 0$.

Since $\langle\mathbf{w}, \mathbf{h}_{\text{opt}}\rangle \ne 0$, there exists $\eta \ne 0$ such that $\|\mathbf{h}_{\text{opt}} + \eta\mathbf{w}\|_2 = \|\mathbf{h}_{\text{opt}}\|_2$. (In fact, we can find $\eta = -2\langle\mathbf{w}, \mathbf{h}_{\text{opt}}\rangle/\|\mathbf{w}\|_2^2$ satisfies the requirement). Next we will take a close look at the difference $m(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - m(\mathbf{h}_{\text{opt}})$. On one hand, we have

$$\begin{aligned} &m(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - m(\mathbf{h}_{\text{opt}}) \\ &= \mathbf{g}^\top[(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - \mathbf{h}_{\text{opt}}] + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top\mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} - \frac{\mathbf{h}_{\text{opt}}^\top\mathbf{H}\mathbf{h}_{\text{opt}}}{2} \\ &= -[(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - \mathbf{h}_{\text{opt}}]^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{h}_{\text{opt}} + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top\mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} - \frac{\mathbf{h}_{\text{opt}}^\top\mathbf{H}\mathbf{h}_{\text{opt}}}{2} && (59) \\ &= \frac{\lambda\eta^2}{2}\|\mathbf{w}\|_2^2 + [\mathbf{h}_{\text{opt}} - (\mathbf{h}_{\text{opt}} + \eta\mathbf{w})]^\top\mathbf{H}\mathbf{h}_{\text{opt}} + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top\mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} - \frac{\mathbf{h}_{\text{opt}}^\top\mathbf{H}\mathbf{h}_{\text{opt}}}{2} && (60) \\ &= \frac{\lambda\eta^2}{2}\|\mathbf{w}\|_2^2 + \frac{\mathbf{h}_{\text{opt}}^\top\mathbf{H}\mathbf{h}_{\text{opt}}}{2} - (\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top\mathbf{H}\mathbf{h}_{\text{opt}} + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top\mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} \\ &= \frac{\lambda\eta^2}{2}\|\mathbf{w}\|_2^2 + \frac{\eta^2}{2}\mathbf{w}^\top\mathbf{H}\mathbf{w} = \frac{\eta^2}{2}\mathbf{w}^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{w}, \end{aligned}$$

where (59) holds due to (57) and (60) holds due to the definition of η. On the other hand, by the definition of $\mathbf{h}_{\text{opt}}$, $m(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - m(\mathbf{h}_{\text{opt}}) \ge 0$. Thus, we have proved (55). Finally, we prove (56) by showing that

$$\begin{aligned} &\langle\mathbf{g}, \mathbf{h}_{\text{opt}}\rangle + \frac{1}{2}\langle\mathbf{H}\mathbf{h}_{\text{opt}}, \mathbf{h}_{\text{opt}}\rangle + \frac{\theta}{6}\|\mathbf{h}_{\text{opt}}\|_2^3 \\ &= \bigg\langle\mathbf{g} + \mathbf{H}\mathbf{h}_{\text{opt}} + \frac{\theta}{2}\|\mathbf{h}_{\text{opt}}\|_2\mathbf{h}_{\text{opt}}, \mathbf{h}_{\text{opt}}\bigg\rangle - \frac{1}{2}\mathbf{h}_{\text{opt}}^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{h}_{\text{opt}} - \frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3 \\ &= -\frac{1}{2}\mathbf{h}_{\text{opt}}^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{h}_{\text{opt}} - \frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3 && (61) \\ &\le -\frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3, && (62) \end{aligned}$$

where (61) holds due to (54) and (62) holds due to (55). ■

B.2. Proof of Lemma 25

In order to prove Lemma 25, we need the following useful lemma.

Lemma 31 *Suppose* $\mathbf{a}_1, \ldots, \mathbf{a}_N$ *are i.i.d. and* $\mathbb{E}\mathbf{a}_i = 0$, *then*

$$\mathbb{E}\bigg\|\frac{1}{N}\sum_{i=1}^N \mathbf{a}_i\bigg\|_2^{3/2} \le \frac{1}{N^{3/4}}\big(\mathbb{E}\|\mathbf{a}_i\|_2^2\big)^{3/4}.$$

Proof [Proof of Lemma 25] For simplification, we use $\mathbb{E}$ to replace $\mathbb{E}_{\mathbf{v}_{i_t}}$. We have

$\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}$

$$= \mathbb{E}\Big\|\frac{1}{b_g}\sum\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s - \Big[\frac{1}{b_g}\sum \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s) - \mathbf{H}^s\Big](\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s)\Big\|_2^{3/2}$$
$$= \mathbb{E}\Big\|\frac{1}{b_g}\sum\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s)$$
$$+ \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big]\Big\|_2^{3/2}.$$

Now we set the parameters in Lemma 31 as $N = b_g$ and

$$\mathbf{a}_{i_t} = \nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^{s+1}) + \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s).$$

We can check that $\mathbf{a}_{i_t}$ satisfy the assumption of Lemma 31. Thus, by Lemma 31, we have

$$\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \le \frac{1}{b_g^{3/4}}\Big(\mathbb{E}\big\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)$$
$$- \nabla F(\mathbf{x}_t^s) + \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2^2\Big)^{3/4}. \tag{63}$$

By Assumption 1 and Lemma 40, we have

$$\begin{aligned}
&\big\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s) + \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2 \\
&\le \big\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2 + \big\|\nabla F(\mathbf{x}_t^s) - \nabla F(\widehat{\mathbf{x}}^s) - \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2 \\
&\le \frac{L_2}{2}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 + \frac{L_2}{2}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 \\
&= L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2.
\end{aligned} \tag{64}$$

Plugging (64) into (63) yields

$$\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \le \frac{1}{b_g^{3/4}}\big(L_2^2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^4\big)^{3/4} = \frac{L_2^{3/2}}{b_g^{3/4}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3.$$

■

B.3. Proof of Lemma 26

In order to prove Lemma 26, we need the following supporting lemma.

Lemma 32 *Suppose that $q \ge 2, p \ge 2$, and fix $r \ge \max\{q, 2\log p\}$. Consider i.i.d. random self-adjoint matrices $\mathbf{Y}_1, ..., \mathbf{Y}_N$ with dimension $p \times p$, $\mathbb{E}\mathbf{Y}_i = \mathbf{0}$. It holds that*

$$\Big[\mathbb{E}\Big\|\sum_{i=1}^N \mathbf{Y}_i\Big\|_2^q\Big]^{1/q} \le 2\sqrt{er}\Big\|\Big(\sum_{i=1}^N \mathbb{E}\mathbf{Y}_i^2\Big)^{1/2}\Big\|_2 + 4er\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^q\big)^{1/q}.$$

Proof [Proof of Lemma 26] For simplicity, we use $\mathbb{E}$ to denote $\mathbb{E}_{j_t}$. We have

$$\mathbb{E}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 = \mathbb{E}\Big\|\nabla^2 F(\mathbf{x}_t^s) - \frac{1}{b_h}\Big(\sum\big(\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s\big)\Big)\Big\|_2^3$$

$$= \mathbb{E}\Big\|\frac{1}{b_h}\Big[\sum\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)\big]\Big]\Big\|_2^3. \tag{65}$$

We apply Lemma 32 with parameters

$$q = 3, p = d, r = 2\log p, \mathbf{Y}_{j_t} = \nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s), N = b_h.$$

It can be easily checked that these parameters satisfy the assumption of Lemma 32. Meanwhile, by Assumption 1, we have the following upper bound for $\mathbf{Y}_{j_t}$:

$$\begin{aligned}
\big\|\mathbf{Y}_{j_t}\big\|_2 &= \big\|\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)\big\|_2 \\
&\le \big\|\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big\|_2 + \big\|\mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)\big\|_2 \\
&\le L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 + L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \\
&= 2L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2.
\end{aligned} \tag{66}$$

By Lemma 32, we have

$$\Big[\mathbb{E}\Big\|\sum \mathbf{Y}_{j_t}\Big\|_2^3\Big]^{1/3} \le 2\sqrt{er}\Big\|\Big(\sum \mathbb{E}\mathbf{Y}_{j_t}^2\Big)^{1/2}\Big\|_2 + 4er\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^3\big)^{1/3}. \tag{67}$$

The first term in RHS of (67) can be bounded as

$$\begin{aligned}
2\sqrt{er}\Big\|\Big(\sum \mathbb{E}\mathbf{Y}_{j_t}^2\Big)^{1/2}\Big\|_2 &= 2\sqrt{er}\Big\|\sum \mathbb{E}\mathbf{Y}_{j_t}^2\Big\|_2^{1/2} \\
&= 2\sqrt{Ner}\Big\|\mathbb{E}\mathbf{Y}_{j_t}^2\Big\|_2^{1/2} \\
&\le 2\sqrt{Ner}\Big(\mathbb{E}\Big\|\mathbf{Y}_{j_t}^2\Big\|_2\Big)^{1/2} \\
&= 2\sqrt{Ner}\Big(\mathbb{E}\|\mathbf{Y}_{j_t}\|_2^2\Big)^{1/2} \\
&\le 4L_2\sqrt{Ner}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2,
\end{aligned} \tag{68}$$

where the first inequality holds due to Jensen's inequality, the third equality holds because $\big\|\mathbf{Y}_{j_t}^2\big\|_2 = \|\mathbf{Y}_{j_t}\|_2^2$ and the last inequality holds due to (66). The second term in RHS of (67) can be bounded as

$$4er\big(\mathbb{E}\max_i \big\|\mathbf{Y}_i\big\|_2^3\big)^{1/3} \le 4er[(2L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2)^3]^{1/3} = 8L_2er\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2. \tag{69}$$

Submitting (68), (69) into (67), we have

$$\Big[\mathbb{E}\Big\|\sum \mathbf{Y}_{j_t}\Big\|_2^3\Big]^{1/3} \le 4L_2\sqrt{Ner}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 + 8L_2er\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2,$$

which immediately implies

$$\mathbb{E}\Big\|\frac{1}{N}\sum \mathbf{Y}_{j_t}\Big\|_2^3 \le 64L_2^3\Big(\sqrt{\frac{er}{N}} + \frac{2er}{N}\Big)^3\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \tag{70}$$

Submitting (70) into (65) with $\mathbf{Y}_{j_t} = \nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)$, $r = 2\log d$, $N = b_h$, we have

$$\begin{aligned}\mathbb{E}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 &\le 64L_2^3\bigg(\sqrt{\frac{2e\log d}{b_h}} + \frac{4e\log d}{b_h}\bigg)^3 \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \\ &\le 1200L_2^3\bigg(\frac{\log d}{b_h}\bigg)^{3/2}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,\end{aligned}$$

where the last inequality holds due to $b_h \ge 400\log d$. ■

B.4. Proof of Lemma 27

Proof we have

$$\begin{aligned}\langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \mathbf{h}\rangle &\le \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 \cdot \|\mathbf{h}\|_2 \\ &= \bigg(\frac{M^{1/3}}{9^{1/3}}\|\mathbf{h}\|_2\bigg)\cdot\bigg(\frac{9^{1/3}}{M^{1/3}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2\bigg) \\ &\le \frac{1}{3}\bigg(\frac{M^{1/3}}{9^{1/3}}\|\mathbf{h}\|_2\bigg)^3 + \frac{2}{3}\bigg(\frac{9^{1/3}}{M^{1/3}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2\bigg)^{3/2} \\ &= \frac{M}{27}\|\mathbf{h}\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}},\end{aligned}$$

where the second inequality holds due to Young's inequality. Meanwhile, we have

$$\begin{aligned}\big\langle\big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big)\mathbf{h}, \mathbf{h}\big\rangle &\le \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big\|_2 \cdot \|\mathbf{h}\|_2^2 \\ &= \bigg(\frac{M^{2/3}}{9^{2/3}}\|\mathbf{h}\|_2^2\bigg)\cdot\bigg(\frac{9^{2/3}}{M^{2/3}}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big\|_2\bigg) \\ &\le \frac{2}{3}\bigg(\frac{M^{2/3}}{9^{2/3}}\|\mathbf{h}\|_2^2\bigg)^{3/2} + \frac{1}{3}\bigg(\frac{9^{2/3}}{M^{2/3}}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big\|_2\bigg)^3 \\ &= \frac{2M}{27}\|\mathbf{h}\|_2^3 + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3,\end{aligned}$$

where the second inequality holds due to Young's inequality. ■

B.5. Proof of Lemma 28

In order to prove Lemma 28, we need the following two useful lemmas.

Lemma 33 *Under Assumption 1, if $M \ge 2L_2$, then we have*

$$\big\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\big\|_2 \le M\|\mathbf{h}\|_2^2 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 + \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^2 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2.$$

Lemma 34 *Under Assumption 1, if $M \ge 2L_2$, then we have*

$$-\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big) \le M\|\mathbf{h}\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + M\big|\|\mathbf{h}\|_2 - \|\mathbf{h}_t^s\|_2\big|.$$

Proof [Proof of Lemma 28] By the definition of μ, we can bound $\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\|_2^{3/2}$ and $0 \vee -L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big)\big]^3$ separately. To bound $\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\|_2^{3/2}$, applying Lemma 33 we have

$$\begin{aligned}&\big\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\big\|_2^{3/2} \\ &\le \bigg[M\|\mathbf{h}\|_2^2 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 + \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^2 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2\bigg]^{3/2} \\ &\le 2\bigg[M^{3/2}\|\mathbf{h}\|_2^3 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2^{3/2}\bigg],\end{aligned}$$

where the second inequality holds due to the following basic inequality $(a+b+c+d)^{3/2} \le 2(a^{3/2} + b^{3/2} + c^{3/2} + d^{3/2})$. To bound $-\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big)$, applying Lemma 26, we have

$$\begin{aligned}&-L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big)\big]^3 \\ &= -C_M^{3/2}M^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big)\big]^3 \\ &\le C_M^{3/2}M^{-3/2}\bigg[M\|\mathbf{h}\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + M\big|\|\mathbf{h}\|_2 - \|\mathbf{h}_t^s\|_2\big|\bigg]^3 \\ &\le 9C_M^{3/2}\bigg[M^{3/2}\|\mathbf{h}\|_2^3 + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 + M^{3/2}\big|\|\mathbf{h}\|_2 - \|\mathbf{h}_t^s\|_2\big|^3\bigg],\end{aligned}$$

where the second inequality is due to $(a+b+c)^3 \le 9(a^3 + b^3 + c^3)$. Since $9C_M^{3/2} > 2$, we have

$$\begin{aligned}\mu(\mathbf{x}_t^s + \mathbf{h}) &= \max\Big\{\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\|_2^{3/2}, -L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big)\big]^3\Big\} \\ &\le 9C_M^{3/2}\bigg[M^{3/2}\|\mathbf{h}\|_2^3 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 \\ &\qquad + \big\|\nabla m_t^s(\mathbf{h})\big\|_2^{3/2} + M^{3/2}\big|\|\mathbf{h}\|_2 - \|\mathbf{h}_t^s\|_2\big|^3\bigg],\end{aligned}$$

which completes the proof. ■

B.6. Proof of Lemma 29

Proof For any $\mathbf{x}_t^s, \mathbf{h}, \widehat{\mathbf{x}}^s$ and a constant $C > 0$, we have

$$\begin{aligned}&\|\mathbf{x}_t^s + \mathbf{h} - \widehat{\mathbf{x}}^s\|_2^3 \\ &\le \big(\|\mathbf{h}\|_2 + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2\big)^3 \\ &= \|\mathbf{h}\|_2^3 + 3\|\mathbf{h}\|_2^2 \cdot \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 + 3\|\mathbf{h}\|_2 \cdot \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \\ &= \|\mathbf{h}\|_2^3 + 3\big(C^{1/3}\|\mathbf{h}\|_2^2\big)\cdot\frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2}{C^{1/3}} + 3\big(C^{2/3}\|\mathbf{h}\|_2\big)\cdot\frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2}{C^{2/3}} + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \\ &\le \|\mathbf{h}\|_2^3 + 3\bigg(\frac{2}{3}\big(C^{1/3}\|\mathbf{h}\|_2^2\big)^{3/2} + \frac{1}{3}\bigg(\frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2}{C^{1/3}}\bigg)^3\bigg) \\ &\qquad + 3\bigg(\frac{1}{3}\big(C^{2/3}\|\mathbf{h}\|_2\big)^3 + \frac{2}{3}\bigg(\frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2}{C^{2/3}}\bigg)^{3/2}\bigg) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\end{aligned}$$

$$= \|\mathbf{h}\|_2^3 + \left(2C^{1/2}\|\mathbf{h}\|_2^3 + \frac{1}{C}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\right) + \left(C^2\|\mathbf{h}\|_2^3 + \frac{2}{C}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\right) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3$$
$$\leq 2C^2\|\mathbf{h}\|_2^3 + \left(1 + \frac{3}{C}\right)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|^3, \tag{71}$$

where the second inequality holds due to Young's inequality, the last inequality holds because $1 + 2\sqrt{C} \leq C^2$ when $C \geq 3/2$. ■

B.7. Proof of Lemma 30

Proof By induction, we have for any $0 \leq t \leq T$,

$$c_t = M\frac{(1+3/T)^{T-t} - 1}{1500T^2}.$$

Then for any $0 \leq t \leq T$,

$$2c_tT^2 \leq M\frac{2(1+3/T)^T}{1500} \leq M\frac{2 \cdot 27}{1500} < \frac{M}{24}.$$

■

Appendix C. Proof of Auxiliary Lemmas

In this section, we prove auxiliary lemmas used in Appendix B.

C.1. Proof of Lemma 31

Proof We have

$$\mathbb{E}\left\|\frac{1}{N}\sum_{i=1}^N \mathbf{a}_i\right\|_2^{3/2} = \frac{\mathbb{E}\|\sum_{i=1}^N \mathbf{a}_i\|_2^{3/2}}{N^{3/2}} \leq \frac{(\mathbb{E}\|\sum_{i=1}^N \mathbf{a}_i\|_2^2)^{3/4}}{N^{3/2}} = \frac{(\sum_{i=1}^N \mathbb{E}\|\mathbf{a}_i\|_2^2)^{3/4}}{N^{3/2}} = \frac{\left(\mathbb{E}\|\mathbf{a}_i\|_2^2\right)^{3/4}}{N^{3/4}}.$$

The first inequality holds due to Lemma 41 with $s = 3/2, t = 2$. The second equality holds due to $\mathbb{E}\mathbf{a}_i = 0$ and that $\mathbf{a}_i$ are independently identically distributed. ■

C.2. Proof of Lemma 32

Proof This proof is mainly adapted from Chen et al. (2012); Tropp (2016). First, Let $\{\mathbf{Y}_i' : i = 1, \ldots, N\}$ be an independent copy of the sequence $\{\mathbf{Y}_i : i = 1, \ldots, N\}$. We denote $\mathbb{E}_{\mathbf{Y}'}$ to be the expectation over the independent copy $\mathbf{Y}'$. Then $\mathbb{E}_{\mathbf{Y}'}\mathbf{Y}_i' = 0$ and

$$\mathbb{E}\left\|\sum_{i=1}^N \mathbf{Y}_i\right\|_2^q = \mathbb{E}\left\|\sum_{i=1}^N \mathbb{E}_{\mathbf{Y}'}(\mathbf{Y}_i - \mathbf{Y}_i')\right\|_2^q \leq \mathbb{E}\left[\mathbb{E}_{\mathbf{Y}'}\left\|\sum_{i=1}^N(\mathbf{Y}_i - \mathbf{Y}_i')\right\|_2^q\right] = \mathbb{E}\left\|\sum_{i=1}^N(\mathbf{Y}_i - \mathbf{Y}_i')\right\|_2^q. \tag{72}$$

The first equality holds due to $\mathbb{E}_{\mathbf{Y}'}\mathbf{Y}_i' = 0$, the first inequality holds because $\|\cdot\|_2^q$ is a convex function, and the second equality holds because we combine the iterated expectation into a single expectation.

Note that $\mathbf{Y}_i - \mathbf{Y}_i'$ has the same distribution as $\mathbf{Y}_i' - \mathbf{Y}_i$, thus the independent sequence $\{\xi_i(\mathbf{Y}_i - \mathbf{Y}_i') : 1 \leq i \leq n\}$ has the same distribution as $\{\mathbf{Y}_i - \mathbf{Y}_i' : 1 \leq i \leq N\}$, where ξ_i are independent Rademacher random variables, also independent with $\mathbf{Y}_i, \mathbf{Y}_i'$. Therefore,

$$\mathbb{E}\left\|\sum_{i=1}^N(\mathbf{Y}_i - \mathbf{Y}_i')\right\|_2^q = \mathbb{E}\left\|\sum_{i=1}^N \xi_i(\mathbf{Y}_i - \mathbf{Y}_i')\right\|_2^q. \tag{73}$$

Furthermore, we have

$$\mathbb{E}\left\|\sum_{i=1}^N \xi_i(\mathbf{Y}_i - \mathbf{Y}_i')\right\|_2^q \leq \mathbb{E}\left[2^{q-1}\left(\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_2^q + \left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i'\right\|_2^q\right)\right] = 2^q \cdot \mathbb{E}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_2^q. \tag{74}$$

The first inequality holds due to $\|\mathbf{A} - \mathbf{B}\|_2^q \leq (\|\mathbf{A}\|_2 + \|\mathbf{B}\|_2)^q \leq 2^{q-1}(\|\mathbf{A}\|_2^q + \|\mathbf{B}\|_2^q)$, where we let $\mathbf{A} = \sum_{i=1}^N \xi_i\mathbf{Y}_i, \mathbf{B} = \sum_{i=1}^N \xi_i\mathbf{Y}_i'$; the equality holds due to the identical distribution of $\{\xi\mathbf{Y}_i\}$ and $\{\xi\mathbf{Y}_i'\}$. Submitting (73), (74) into (72) yields

$$\mathbb{E}\left\|\sum_{i=1}^N \mathbf{Y}_i\right\|_2^q \leq 2^q \cdot \mathbb{E}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_2^q \tag{75}$$

Taking q-th root for both sides, we have

$$\left[\mathbb{E}\left\|\sum_{i=1}^N \mathbf{Y}_i\right\|_2^q\right]^{1/q} \leq 2\left[\mathbb{E}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_2^q\right]^{1/q}. \tag{76}$$

Next, we have the inequality chain:

$$\begin{aligned}
2\left[\mathbb{E}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_2^q\right]^{1/q} &\leq 2\left[\mathbb{E}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_{S_r}^q\right]^{1/q} \\
&= 2\left[\mathbb{E}_{\mathbf{Y}_i}\left(\mathbb{E}_{\xi_i}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_{S_r}^q\right)\right]^{1/q} \\
&\leq 2\left[\mathbb{E}_{\mathbf{Y}_i}\left(\mathbb{E}_{\xi_i}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_{S_r}^r\right)^{q/r}\right]^{1/q},
\end{aligned} \tag{77}$$

where the first inequality holds due to $\|\cdot\|_2 \leq \|\cdot\|_{S_r}$, the second inequality holds due to Lyapunov's inequality (Lemma 41), where we set $s = q, t = r$. Since $q < r$, then the second inequality holds. Note we have

$$2\left[\mathbb{E}_{\mathbf{Y}_i}\left(\mathbb{E}_{\xi_i}\left\|\sum_{i=1}^N \xi_i\mathbf{Y}_i\right\|_{S_r}^r\right)^{q/r}\right]^{1/q} \leq 2\sqrt{r}\left[\mathbb{E}\left\|\left(\sum_{i=1}^N \mathbf{Y}_i^2\right)^{1/2}\right\|_{S_r}^q\right]^{1/q}$$

$$
\begin{aligned}
&\le 2\sqrt{r}\Big[\mathbb{E}\Big(p^{1/r}\Big\|\Big(\sum_{i=1}^N \mathbf{Y}_i^2\Big)^{1/2}\Big\|_2\Big)^q\Big]^{1/q}\\
&\le 2\sqrt{er}\Big[\mathbb{E}\Big\|\Big(\sum_{i=1}^N \mathbf{Y}_i^2\Big)^{1/2}\Big\|_2^q\Big]^{1/q}\\
&= 2\sqrt{er}\Big[\mathbb{E}\Big\|\sum_{i=1}^N \mathbf{Y}_i^2\Big\|_2^{q/2}\Big]^{1/q}, \qquad (78)
\end{aligned}
$$

where the first inequality holds due to Proposition 42; the second inequality holds because $\|\mathbf{A}\|_{S_r} \le p^{1/r}\|\mathbf{A}\|_2$, where we set $\mathbf{A} = \big(\sum_{i=1}^N \mathbf{Y}_i^2\big)^{1/2}$ and p is the dimension of $\mathbf{A}$; the third inequality holds because $p^{1/r} \le p^{1/(2\log p)} = \sqrt{e}$.

Finally, we use Proposition 43 to bound (78). Since $\mathbf{Y}_i^2$ are positive-semidefinite and independent random matrices, we can set $\mathbf{W}_i$ in Proposition 43 as $\mathbf{W}_i = \mathbf{Y}_i^2$. Meanwhile, $q/2 \ge 1$, so we have

$$
\Big[\mathbb{E}\Big\|\sum_{i=1}^N \mathbf{Y}_i^2\Big\|_2^{q/2}\Big]^{2/q} \le \Big[\Big\|\sum_{i=1}^N \mathbb{E}\mathbf{Y}_i^2\Big\|_2^{1/2} + 2\sqrt{er}\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^q\big)^{1/q}\Big]^2,
$$

which immediately implies

$$
\Big[\mathbb{E}\Big\|\sum_{i=1}^N \mathbf{Y}_i^2\Big\|_2^{q/2}\Big]^{1/q} \le \Big\|\sum_{i=1}^N \mathbb{E}\mathbf{Y}_i^2\Big\|_2^{1/2} + 2\sqrt{er}\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^q\big)^{1/q}. \qquad (79)
$$

Submitting (77), (78),(79) into (76), we have the proof completed. ■

C.3. Proof of Lemma 33

Proof We have

$$
\begin{aligned}
\|\nabla F(\mathbf{x}_t^s+\mathbf{h})\|_2 &= \big\|\nabla F(\mathbf{x}_t^s+\mathbf{h}) - \nabla F(\mathbf{x}_t^s) - \nabla^2 F(\mathbf{x}_t^s)\mathbf{h} + \mathbf{v}_t^s + \mathbf{U}_t^s\mathbf{h}\\
&\quad + \big(\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\big) + \big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}\big\|_2\\
&\le \big\|\nabla F(\mathbf{x}_t^s+\mathbf{h}) - \nabla F(\mathbf{x}_t^s) - \nabla^2 F(\mathbf{x}_t^s)\mathbf{h}\big\|_2 + \Big\|\mathbf{v}_t^s + \mathbf{U}_t^s\mathbf{h} + \frac{M}{2}\|\mathbf{h}\|_2\mathbf{h}\Big\|_2\\
&\quad + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 + \big\|\big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}\big\|_2 + \frac{M}{2}\|\mathbf{h}\|_2^2, \qquad (80)
\end{aligned}
$$

where the inequality holds due to triangle inequality. In the following, we are going to bound the right-hand side of (80). For the first term in the right-hand side of (80), it can be bounded as

$$
\big\|\nabla F(\mathbf{x}_t^s+\mathbf{h}) - \nabla F(\mathbf{x}_t^s) - \nabla^2 F(\mathbf{x}_t^s)\mathbf{h}\big\|_2 \le \frac{L_2}{2}\|\mathbf{h}\|_2^2 \le \frac{M}{4}\|\mathbf{h}\|_2^2,
$$

where the first inequality holds due to Assumption 1 and the second inequality holds due to $2L_2 \le M$. For the second term in the the right hand side of (80), it equals to

$$
\Big\|\mathbf{v}_t^s + \mathbf{U}_t^s\mathbf{h} + \frac{M}{2}\|\mathbf{h}\|_2\mathbf{h}\Big\|_2 = \big\|\nabla m_t^s(\mathbf{h})\big\|_2.
$$

And the final term can be bounded as

$$
\big\|\big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}\big\|_2 \le \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 \cdot \|\mathbf{h}\|_2 \le \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^2 + \frac{M}{4}\|\mathbf{h}\|_2^2,
$$

where the last inequality is due to Young's inequality. Putting all these bounds together and submit them into (80), we have

$$
\big\|\nabla F(\mathbf{x}_t^s+\mathbf{h})\big\|_2 \le M\|\mathbf{h}\|_2^2 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 + \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^2 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2.
$$

■

C.4. Proof of Lemma 34

Proof We have

$$
\begin{aligned}
\nabla^2 F(\mathbf{x}_t^s+\mathbf{h}) &\succeq \nabla^2 F(\mathbf{x}_t^s) - L_2\|\mathbf{h}\|_2\mathbf{I}\\
&\succeq \mathbf{U}_t^s - \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2\mathbf{I} - L_2\|\mathbf{h}\|_2\mathbf{I}\\
&\succeq -\frac{M}{2}\|\mathbf{h}_t^s\|_2\mathbf{I} - \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2\mathbf{I} - L_2\|\mathbf{h}\|_2\mathbf{I},
\end{aligned}
$$

where the first inequality holds because $\nabla^2 F$ is L_2-Hessian Lipschitz, the last inequality holds due to (29) in Lemma 24. Thus we have

$$
\begin{aligned}
-\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s+\mathbf{h})\big) &\le \frac{M}{2}\|\mathbf{h}_t^s\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + L_2\|\mathbf{h}\|_2\\
&= \frac{M}{2}(\|\mathbf{h}_t^s\|_2 - \|\mathbf{h}\|_2) + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + (L_2 + M/2)\|\mathbf{h}\|_2\\
&\le M\|\mathbf{h}\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + M\big|\|\mathbf{h}_t^s\|_2 - \|\mathbf{h}\|_2\big|,
\end{aligned}
$$

where the last inequality holds because $L_2 \le M/2$. ■

Appendix D. Proof of Main Theoretical Results for Lite-SVRC

In this section, we provide the proofs of theoretical results of Lite-SVRC (Algorithm 2).

D.1. Proof of Theorem 18

For the simplification of notation, we define $\mathbf{e}_{\mathbf{v}}, \mathbf{e}_{\mathbf{U}}$ as follows

$$
\mathbf{e}_{\mathbf{v}} = \nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s, \quad \mathbf{e}_{\mathbf{U}} = \nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s, \qquad (81)
$$

where $\mathbf{v}_0^s = \mathbf{g}^s, \mathbf{U}_0^s = \mathbf{H}^s$. Before we state the proof, we present some technical lemmas that are useful in our analysis. Firstly, we give a sharp bound of $\mu(\mathbf{x}_{t+1}^s)$. A very crucial observation is that we can bound the norm of gradient $\|\nabla F(\mathbf{x}_{t+1}^s)\|_2$ and the smallest eigenvalue of Hessian $\lambda_{\min}(\nabla^2 F(\mathbf{x}_{t+1}^s))$ with $\|\mathbf{h}_t^s\|_2$, $\|\mathbf{e}_{\mathbf{v}}\|_2$ and $\|\mathbf{e}_{\mathbf{U}}\|_2$ defined in (81).

Lemma 35 *Under the same assumption as in Theorem 18, let $\mathbf{h}_t^s, \mathbf{x}_{t+1}^s, M_{s,t}$ be variables defined by Algorithm 2. Then we have*

$$\mu(\mathbf{x}_{t+1}^s) \le 9C_M^{3/2}\big(M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3 + \|\mathbf{e}_\mathbf{v}\|_2^{3/2} + M_{s,t}^{-3/2}\|\mathbf{e}_\mathbf{U}\|_2^3\big), \tag{82}$$

where C_M is the parameter in $M_{s,t} = C_M L_2$.

Lemma 35 suggests that to bound our target $\mathbb{E}\mu(\mathbf{x}_{t+1}^s)$, we only need to focus on $\mathbb{E}\|\mathbf{h}_t^s\|_2^3$, $\mathbb{E}\|\mathbf{e}_\mathbf{v}\|_2^{3/2}$ and $\mathbb{E}\|\mathbf{e}_\mathbf{U}\|_2^3$. The next lemma upper bound $F(\mathbf{x}_t^s) - F(\mathbf{x}_{t+1}^s)$ with $\mathbf{e}_\mathbf{v}, \mathbf{e}_\mathbf{U}$ and $\mathbf{h}_t^s$, which can be straightforwardly derived from the Hessian Lipschitz condition.

Lemma 36 *Under the same assumption as in Theorem 18, let $\mathbf{h}_t^s, \mathbf{x}_t^s, \mathbf{x}_{t+1}^s, M_{s,t}$ be variables defined by Algorithm 2. Then we have the following result:*

$$F(\mathbf{x}_{t+1}^s) \le F(\mathbf{x}_t^s) - M_{s,t}/12 \cdot \|\mathbf{h}_t^s\|_2^3 + C_1\big(\|\mathbf{e}_\mathbf{v}\|_2^{3/2}/M_{s,t}^{1/2} + \|\mathbf{e}_\mathbf{U}\|_2^3/M_{s,t}^2\big), \tag{83}$$

where $C_1 = 200$.

We can also bound $\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3$ with $\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3$ as follows.

Lemma 37 *Under the same assumption as in Theorem 18, let $\mathbf{h}_t^s, \mathbf{x}_t^s, \mathbf{x}_{t+1}^s, M_{s,t}$ be variables defined by Algorithm 2. Then we have the following result:*

$$\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 \le (1 + 3/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + 2n^{2/3}\|\mathbf{h}_t^s\|_2^3. \tag{84}$$

Finally, we bound the variance of $\widetilde{\mathbf{v}}_t^s$ and $\mathbf{U}_t^s$ as follows.

Lemma 38 *Under the same assumption as in Theorem 18, let $\mathbf{x}_t^s, \widetilde{\mathbf{v}}_t^s$ and $\widehat{\mathbf{x}}^s$ be the iterates defined in Algorithm 2. Then we have*

$$\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}\|\mathbf{e}_\mathbf{v}\|_2^{3/2} \le \frac{3L_1^{3/2}}{D_g^{3/4}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where D_g is the batch size parameter defined in (21) and $\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}$ takes expectation over $\widetilde{\mathbf{v}}_t^s$.

Lemma 39 *Under the same assumption as in Theorem 18, let $\mathbf{x}_t^s, \mathbf{U}_t^s$ and $\widehat{\mathbf{x}}^s$ be iterates defined in Algorithm 2. If the batch size of Hessian satisfies $B_h > 400\log d$, we have*

$$\mathbb{E}_{\mathbf{U}_t^s}\|\mathbf{e}_\mathbf{U}\|_2^{3/2} \le \frac{C_h L_2^3}{B_h^{3/2}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where $C_h = 1200(\log d)^{3/2}$ and $\mathbb{E}_{\mathbf{U}_t^s}$ only takes expectation over $\mathbf{U}_t^s$.

Lemmas 38 and 39 suggest that with carefully selection of batch size, both $\mathbb{E}\|\mathbf{e}_\mathbf{v}\|_2^{3/2}$ and $\mathbb{E}\|\mathbf{e}_\mathbf{U}\|_2^3$ can be bounded by $\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3$, which play similar roles to Lemmas 25 and 26 in the analysis of SVRC.

Proof [Proof of Theorem 18] We first define $R_t^s = \mathbb{E}[F(\mathbf{x}_t^s) + c_{s,t}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3]$, where $c_{s,t}$ is a number series defined as follows: $c_{s,T} = 0$ and for $s = 1, \ldots, S$, $t = 0, \ldots, T-1$,

$$c_{s,t} = c_{s,t+1}\big(1 + 3/n^{1/3}\big) + M_{s,t}/(500n). \tag{85}$$

Combining Lemmas 36 and 37, we can upper bound $F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3$. Specifically, (83) + $c_{s,t+1}\times$ (84) yields

$$\begin{aligned}
&F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 \\
&\le F(\mathbf{x}_t^s) - \|\mathbf{h}_t^s\|_2^3\big(M_{s,t}/12 - 2n^{2/3}c_{s,t+1}\big) + C_1\big(\|\mathbf{e}_\mathbf{v}\|_2^{3/2}/M_{s,t}^{1/2} + \|\mathbf{e}_\mathbf{U}\|_2^3/M_{s,t}^2\big) \\
&\quad + c_{s,t+1}(1 + 3/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3.
\end{aligned} \tag{86}$$

By Lemma 35, multiplying (82) with $\big(24 \times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}$ and adding it into (86) yields

$$\begin{aligned}
&F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(24 \times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1} \cdot \mu(\mathbf{x}_{t+1}^s) \\
&\le F(\mathbf{x}_t^s) + c_{s,t+1}\big(1 + 3/n^{1/3}\big)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \\
&\quad + \frac{(24C_1 + 1)}{24M_{s,t}^{1/2}} \cdot \|\mathbf{e}_\mathbf{v}\|_2^{3/2} + \frac{(24C_1 + 1)}{24M_{s,t}^2} \cdot \|\mathbf{e}_\mathbf{U}\|_2^3,
\end{aligned} \tag{87}$$

where we use the fact that $M_{s,t}/24 - 2n^{2/3}c_{s,t+1} > 0$ which can be easily verified by the definition in (85) and a similar argument in Appendix B.7. By Lemmas 38 and 39 we have

$$\mathbb{E}\|\mathbf{e}_\mathbf{v}\|_2^{3/2} \le \frac{3L_1^{3/2}}{D_g^{3/4}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \qquad \mathbb{E}\|\mathbf{e}_\mathbf{U}\|_2^3 \le \frac{C_h L_2^3}{B_h^{3/2}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where D_g, B_h are batch size parameters and $C_h = 1200(\log d)^{3/2}$. Taking expectation on (87), we obtain the following result

$$\begin{aligned}
&\mathbb{E}\big[F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(24 \times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1} \cdot \mu(\mathbf{x}_{t+1}^s)\big] \\
&\le \mathbb{E}\bigg[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \cdot \bigg(c_{s,t+1}\big(1 + 3/n^{1/3}\big) + \frac{6L_1^{3/2}C_1}{D_g^{3/4}M_{s,t}^{1/2}} + \frac{2L_2^3C_hC_1}{B_h^{3/2}M_{s,t}^2}\bigg)\bigg] \\
&\le \mathbb{E}\big[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \cdot \big(c_{s,t+1}\big(1 + 3/n^{1/3}\big) + M_{s,t}/(500n)\big)\big],
\end{aligned} \tag{88}$$

where in the last inequality we use the fact that $M_{s,t} = C_M L_2$, $D_g \ge C_1 n^{4/3}L_1^2/(L_2^2C_M^{4/3})$ and $B_h \ge 144(C_1C_h)^{2/3}n^{2/3}/C_M^2$. By the definition of $c_{s,t}$ in (85), (88) is equivalent to

$$\big(24 \times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1} \cdot \mathbb{E}(\mu(\mathbf{x}_{t+1}^s)) \le R_t^s - R_{t+1}^s. \tag{89}$$

Then we sum up (89) from $t = 0$ to $T - 1$, while yields

$$\sum_{t=0}^{T-1}\big(24 \times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1} \cdot \mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \le R_0^s - R_T^s.$$

Substituting $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s, \mathbf{x}_T^s = \widehat{\mathbf{x}}^{s+1}$ and $c_{s,T} = 0$ into the inequality above, we get

$$\sum_{t=0}^{T-1}\big(24 \times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1} \cdot \mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \le \mathbb{E}F(\widehat{\mathbf{x}}^s) - \mathbb{E}F(\widehat{\mathbf{x}}^{s+1}).$$

Then we take summation from $s=1$ to S, we have

$$\begin{aligned}\sum_{s=1}^{S}\sum_{t=0}^{T-1}\left(24\times 9C_M^{3/2}M_{s,t}^{1/2}\right)^{-1}\cdot\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] &\le \mathbb{E}F(\widehat{\mathbf{x}}^0)-\mathbb{E}F(\widehat{\mathbf{x}}^{S+1})\\ &\le F(\widehat{\mathbf{x}}^0)-\inf_{\mathbf{x}\in\mathbb{R}^d}F(\mathbf{x})\\ &=\Delta_F.\end{aligned}$$

Plugging $M_{s,t}=C_ML_2$ into the above inequality, we have

$$\sum_{s=1}^{S}\sum_{t=0}^{T-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)]\le 216C_M^2L_2^{1/2}\Delta_F. \tag{90}$$

In Algorithm 2, we choose $\mathbf{x}_{\text{out}}$ randomly over s and t, thus we have our result from (90):

$$\mathbb{E}[\mu(\mathbf{x}_{\text{out}})]\le\frac{216C_M^2L_2^{1/2}\Delta_F}{ST}.$$

This competes the proof. ■

D.2. Proof of Corollary 20

Now we provide the proof of our corollary for the sample complexity of Lite-SVRC.

Proof [Proof of Corollary 20] By the definition in (8) and the result in Theorem 18, to find an $(\epsilon,\sqrt{L_2\epsilon})$-approximate local minimum, we only need to make sure $216C_M^2L_2^{1/2}\Delta_F/(ST)\le\epsilon^{3/2}$. Setting $T=n^{1/3}$, it suffices to let $S=O(\max\{L_2^{1/2}\Delta_F/(\epsilon^{3/2}n^{1/3}),1\})$. We need to sample n Hessian at the beginning of each inner loop, and in each inner loop, we need to sample $B_h=\widetilde{O}(n^{2/3})$ Hessian matrices. Therefore, the total sample complexity of Hessian for Algorithm 2 is $S\cdot n+S\cdot T\cdot B_h=\widetilde{O}(n+n^{2/3}\cdot(\Delta_F\sqrt{L_2})/\epsilon^{3/2})$. The total amount of CSO calls is $O(ST)=O(\Delta_F\sqrt{L_2})/\epsilon^{3/2})$. ■

D.3. Proofs of Theorem 22 and Corollary 23

The proofs of the convergence of Lite-SVRC with an inexact cubic subproblem solver defined in Section 5 are almost the same as that of the convergence of SVRC. More specifically, the proof of Theorem 22 is the same as that of Theorem 13, and the proof of Corollary 23 is the same as that of Corollary 15. Therefore, we omit the proofs here for simplicity.

Appendix E. Proof of Technical Lemmas

In this section, we prove the technical lemmas used in the proof of Theorem 18.

E.1. Proof of Lemma 35

Proof [Proof of Lemma 35] Recall the definition of $\mu(\cdot)$ in (8). We need to upper bound $\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2}$ and $-L_2^{3/2}\lambda_{\min}^3(\nabla^2F(\mathbf{x}_{t+1}^s))$, which can be achieved by applying Lemmas 33 and 34 since we have $\mathbf{x}_{t+1}^s=\mathbf{x}_t^s+\mathbf{h}_t^s$. To bound $\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2}$, we apply Lemma 33:

$$\begin{aligned}&\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2}\\ &\le\Big[M_{s,t}\|\mathbf{h}_t^s\|_2^2+\|\nabla m_t^s(\mathbf{h}_t^s)\|_2+\|\nabla F(\mathbf{x}_t^s)-\widetilde{\mathbf{v}}_t^s\|_2+\frac{1}{M_{s,t}}\|\nabla^2F(\mathbf{x}_t^s)-\mathbf{U}_t^s\|_2^2\Big]^{3/2}\\ &\le 2\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3+\|\nabla F(\mathbf{x}_t^s)-\widetilde{\mathbf{v}}_t^s\|_2^{3/2}+\|\nabla m_t^s(\mathbf{h}_t^s)\|_2^{3/2}+M_{s,t}^{-3/2}\|\nabla^2F(\mathbf{x}_t^s)-\mathbf{U}_t^s\|_2^3\Big]\\ &\le 2\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3+\|\nabla F(\mathbf{x}_t^s)-\widetilde{\mathbf{v}}_t^s\|_2^{3/2}+M_{s,t}^{-3/2}\|\nabla^2F(\mathbf{x}_t^s)-\mathbf{U}_t^s\|_2^3\Big],\end{aligned}$$

where the second inequality holds due to the basic inequality $(a+b+c+d)^{3/2}\le 2(a^{3/2}+b^{3/2}+c^{3/2}+d^{3/2})$ and in the last inequality we use the fact that $\nabla m_t^s(\mathbf{h}_t^s)=0$. Next we bound $-M_{s,t}^{-3/2}\big[\lambda_{\min}\big(\nabla^2F(\mathbf{x}_{t+1}^s)\big)\big]^3$. Applying Lemma 34 with $\mathbf{h}=\mathbf{h}_t^s$, we have

$$\begin{aligned}-L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2F(\mathbf{x}_{t+1}^s)\big)\big]^3&\le C_M^{3/2}M_{s,t}^{-3/2}\big[M_{s,t}\|\mathbf{h}_t^s\|_2+\|\nabla^2F(\mathbf{x}_t^s)-\mathbf{U}_t^s\|_2\big]^3\\ &\le 9C_M^{3/2}M_{s,t}^{-3/2}\cdot\Big[\frac{M_{s,t}^3}{4}\|\mathbf{h}_t^s\|_2^3+\|\nabla^2F(\mathbf{x}_t^s)-\mathbf{U}_t^s\|_2^3\Big]\\ &\le 9C_M^{3/2}\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3+M_{s,t}^{-3/2}\|\nabla^2F(\mathbf{x}_t^s)-\mathbf{U}_t^s\|_2^3\Big],\end{aligned}$$

where the second inequality holds due to $(a+b+c)^3\le 9(a^3+b^3+c^3)$. Thus, we have

$$\begin{aligned}\mu(\mathbf{x}_{t+1}^s)&=\max\Big\{\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2},-L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2F(\mathbf{x}_{t+1}^s)\big)\big]^3\Big\}\\ &\le 9C_M^{3/2}\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3+\|\nabla F(\mathbf{x}_t^s)-\widetilde{\mathbf{v}}_t^s\|_2^{3/2}+M_{s,t}^{-3/2}\|\nabla^2F(\mathbf{x}_t^s)-\mathbf{U}_t^s\|_2^3\Big],\end{aligned}$$

which completes the proof. ■

E.2. Proof of Lemma 36

Proof [Proof of Lemma 36] Note that $\mathbf{x}_{t+1}^s=\mathbf{x}_t^s+\mathbf{h}_t^s$. Apply Lemma 40 and we have

$$\begin{aligned}F(\mathbf{x}_{t+1}^s)=F(\mathbf{x}_t^s+\mathbf{h}_t^s)&\le F(\mathbf{x}_t^s)+\langle\nabla F(\mathbf{x}_t^s),\mathbf{h}_t^s\rangle+\frac{1}{2}\langle\nabla^2F(\mathbf{x}_t^s)\mathbf{h}_t^s,\mathbf{h}_t^s\rangle+\frac{L_2}{6}\|\mathbf{h}_t^s\|_2^3\\ &=F(\mathbf{x}_t^s)+\langle\widetilde{\mathbf{v}}_t^s,\mathbf{h}_t^s\rangle+\frac{1}{2}\langle\mathbf{U}_t^s\mathbf{h}_t^s,\mathbf{h}_t^s\rangle+\frac{M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3+\langle\mathbf{e}_{\mathbf{v}},\mathbf{h}_t^s\rangle\\ &\quad+\frac{1}{2}\langle\mathbf{e}_{\mathbf{U}}\mathbf{h}_t^s,\mathbf{h}_t^s\rangle+\frac{L_2-M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3.\end{aligned}\tag{91}$$

Based on Lemma 24 for the sub-problem in cubic regularization, we have

$$\langle\widetilde{\mathbf{v}}_t^s,\mathbf{h}_t^s\rangle+\frac{1}{2}\langle\mathbf{U}_t^s\mathbf{h}_t^s,\mathbf{h}_t^s\rangle+\frac{M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3\le\frac{-M_{s,t}}{12}\|\mathbf{h}_t^s\|_2^3. \tag{92}$$

Meanwhile, we have following two bounds on $\langle \mathbf{e}_{\mathbf{v}}, \mathbf{h}_t^s \rangle$ and $\langle \mathbf{e}_{\mathbf{U}}\mathbf{h}_t^s, \mathbf{h}_t^s \rangle$ by Young's inequality:

$$\langle \mathbf{e}_{\mathbf{v}}, \mathbf{h}_t^s \rangle \le C_4 \|\mathbf{e}_{\mathbf{v}}\|_2 \cdot \frac{1}{C_4}\|\mathbf{h}_t^s\|_2 \le C_4^{3/2}\|\mathbf{e}_{\mathbf{v}}\|_2^{3/2} + \frac{1}{C_4^3}\|\mathbf{h}_t^s\|_2^3, \tag{93}$$

$$\langle \mathbf{e}_{\mathbf{U}}\mathbf{h}_t^s, \mathbf{h}_t^s \rangle \le C_5^2 \|\mathbf{e}_{\mathbf{U}}\|_2 \cdot \left(\frac{\|\mathbf{h}_t^s\|_2}{C_5}\right)^2 \le C_5^6\|\mathbf{e}_{\mathbf{U}}\|_2^3 + \frac{\|\mathbf{h}_t^s\|_2^3}{C_5^3}. \tag{94}$$

We set $C_4 = C_5 = (18/M_{s,t})^{1/3}$. Finally, because $L_2 \le M_{s,t}/2$, we have

$$\frac{L_2 - M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3 \le \frac{-M_{s,t}}{12}\|\mathbf{h}_t^s\|_2^3. \tag{95}$$

Substituting (92), (93), (94) and (95) into (91), we have the final result:

$$F(\mathbf{x}_{t+1}^s) \le F(\mathbf{x}_t^s) - \frac{M_{s,t}}{12}\|\mathbf{h}_t^s\|_2^3 + C_1\left(\frac{\|\mathbf{e}_{\mathbf{v}}\|_2^{3/2}}{M_{s,t}^{1/2}} + \frac{\|\mathbf{e}_{\mathbf{U}}\|_2^3}{M_{s,t}^2}\right), \tag{96}$$

where $C_1 = 200$. ■

E.3. Proof of Lemma 37

Proof [Proof of Lemma 37] Note that $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$, then we have

$$\begin{aligned}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 &\le \left(\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 + \|\mathbf{h}_t^s\|_2\right)^3 \\ &= \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + \|\mathbf{h}_t^s\|_2^3 + 3\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 \cdot \|\mathbf{h}_t^s\|_2 + 3\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \cdot \|\mathbf{h}_t^s\|_2^2. \end{aligned} \tag{97}$$

The inequality holds due to triangle inequality. Next, we bound $\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 \cdot \|\mathbf{h}_t^s\|_2$ and $\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \cdot \|\mathbf{h}_t^s\|_2^2$ by Young's inequality:

$$\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 \cdot \|\mathbf{h}_t^s\|_2 = \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2}{n^{2/9}} \cdot n^{2/9}\|\mathbf{h}_t^s\|_2 \le \frac{2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3}{3n^{1/3}} + \frac{n^{2/3}\|\mathbf{h}_t^s\|_2^3}{3}, \tag{98}$$

$$\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \cdot \|\mathbf{h}_t^s\|_2^2 = \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2}{n^{1/9}} \cdot n^{1/9}\|\mathbf{h}_t^s\|_2^2 \le \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3}{3n^{1/3}} + \frac{2n^{1/6}\|\mathbf{h}_t^s\|_2^3}{3}. \tag{99}$$

Substituting (98), (99) into (97), we have the result:

$$\begin{aligned}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 &\le (1 + 1/n^{1/3} + 2/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + (1 + 2n^{1/6} + n^{2/3})\|\mathbf{h}_t^s\|_2^3 \\ &\le (1 + 3/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + 2n^{2/3}\|\mathbf{h}_t^s\|_2^3,\end{aligned}$$

which completes the proof. ■

E.4. Proof of Lemma 38

Proof [Proof of Lemma 38] This proof is essentially the same as that of Lemma 25 in Section B.2. However, we replace the semi-stochastic gradient $\widetilde{\mathbf{v}}_t^s$ defined in (19) with $\mathbf{v}_t^s$ used in Lemma 25, which leads to the following inequality that is similar to (63):

$$\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}\|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2^{3/2} \le 1/B_{g;s,t}^{3/4}\left(\mathbb{E}\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) + \mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2^2\right)^{3/4}. \tag{100}$$

Next we bound $\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) + \mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2$. By Assumption 17, we have

$$\begin{aligned}\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) + \mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2 &\le \|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\|_2 + \|\mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2 \\ &= \|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\|_2 + \|\nabla F(\widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s)\|_2 \\ &\le 2L_1\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2. \end{aligned} \tag{101}$$

Finally, substituting (101) and $B_{g;s,t} = D_g/\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2$ into (100), we have

$$\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}\|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2^{3/2} \le \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^{3/2}}{D_g^{3/4}} \cdot \left(4L_1^2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2\right)^{3/4} \le \frac{2^{3/2}L_1^{3/2}}{D_g^{3/4}}\|\mathbf{x}_t^{s+1} - \widehat{\mathbf{x}}^s\|_2^3.$$

This completes the proof. ■

E.5. Proof of Lemma 39

Proof [Proof of Lemma 39] The proof of Lemma 39 is the same as that of Lemma 26 in Section B.3. Thus we omit it for simplicity. ■

Appendix F. Additional Lemmas and Propositions

It is obvious that Assumption 1 implies the Hessian Lipschitz assumption of F, which is also equivalent to the following lemma:

Lemma 40 *(Nesterov and Polyak, 2006) Suppose F is L_2-Hessian Lipschitz for some constant $L_2 > 0$, then we have*

$$\begin{aligned}\|\nabla^2 F(\mathbf{x}) - \nabla^2 F(\mathbf{y})\| &\le L_2\|\mathbf{x} - \mathbf{y}\|_2, \\ F(\mathbf{x} + \mathbf{h}) &\le F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{h}\rangle + \frac{1}{2}\langle \nabla^2 F(\mathbf{x})\mathbf{h}, \mathbf{h}\rangle + \frac{L_2}{6}\|\mathbf{h}\|_2^3, \\ \|\nabla F(\mathbf{x} + \mathbf{h}) - \nabla F(\mathbf{x}) - \nabla^2 F(\mathbf{x})\mathbf{h}\|_2 &\le \frac{L_2}{2}\|\mathbf{h}\|_2^2.\end{aligned}$$

Lemma 41 (Lyapunov's Inequality) *(Durrett, 2010) For a random variable X, when $0 < s < t$, it holds that*

$$(\mathbb{E}|X|^s)^{1/s} \le (\mathbb{E}|X|^t)^{1/t}.$$

The following two lemmas are matrix concentration inequalities.

Proposition 42 (Matrix Khintchine Inequality) *(Mackey et al., 2014) Suppose $r > 2$. Consider a finite sequence $\{\mathbf{A}_i, 1 \le i \le N\}$ of deterministic, self-adjoint matrices. Then*

$$\left[\mathbb{E}\left\|\sum_{i=1}^{N}\xi_i\mathbf{A}_i\right\|_{S_r}^{r}\right]^{1/r} \le \sqrt{r}\left\|\left[\sum_{i=1}^{N}\mathbf{A}_i^2\right]^{1/2}\right\|_{S_r},$$

where sequence ξ_i consists of independent Rademacher random variables.

Proposition 43 *(Chen et al., 2012) Let $q \ge 1$, and fix $r \ge \max\{q, 2\log p\}$. Consider $\mathbf{W}_1, ..., \mathbf{W}_N$ of independent, random, positive-definite matrices with dimension $p \times p$. Then*

$$\left[\mathbb{E}\left\|\sum_{i=1}^{N}\mathbf{W}_i\right\|_2^q\right]^{1/q} \le \left[\left\|\sum_{i=1}^{N}\mathbb{E}\mathbf{W}_i\right\|_2^{1/2} + 2\sqrt{er}\left(\mathbb{E}\max_i\|\mathbf{W}_i\|_2^q\right)^{1/(2q)}\right]^2.$$

References

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199. ACM, 2017.

Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In *Advances in Neural Information Processing Systems*, pages 2676–2687, 2018.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707, 2016.

Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*, pages 3720–3730, 2018.

Albert A Bennett. Newton's method in general analysis. *Proceedings of the National Academy of Sciences*, 2(10):592–598, 1916.

Dimitri P Bertsekas. *Nonlinear programming.* Athena scientific Belmont, 1999. ISBN 9781886529144.

Jose Blanchet, Coralia Cartis, Matt Menickelly, and Katya Scheinberg. Convergence rate analysis of a stochastic trust region method for nonconvex optimization. *arXiv preprint arXiv:1609.07428*, 2016.

Yair Carmon and John C Duchi. Gradient descent efficiently finds the cubic-regularized non-convex Newton step. *arXiv preprint arXiv:1612.00547*, 2016.

Yair Carmon and John C Duchi. Analysis of Krylov subspace solutions of regularized non-convex quadratic problems. In *Advances in Neural Information Processing Systems*, pages 10705–10715, 2018.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Trust-region and other regularisations of linear least-squares problems. *BIT Numerical Mathematics*, 49(1):21–53, 2009.

Coralia Cartis, Nicholas I Gould, and Philippe L Toint. Adaptive cubic regularisation methods for unconstrained optimization. part i: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011a.

Coralia Cartis, Nicholas I. M Gould, and Philippe L Toint. *Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity.* Springer-Verlag New York, Inc., 2011b.

Coralia Cartis, Nicholas IM Gould, and Ph L Toint. Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108, 2012.

Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM Journal on Optimization*, 23(3):1553–1574, 2013.

Richard Y Chen, Alex Gittens, and Joel A Tropp. The masked sample covariance estimator: an analysis using matrix concentration inequalities. *Information and Inference: A Journal of the IMA*, 1(1):2–20, 2012.

Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-region Methods.* Society for Industrial and Applied Mathematics, 2000. ISBN 0-89871-460-5.

Frank E Curtis, Daniel P Robinson, and Mohammadreza Samadi. A trust region algorithm with a worst-case iteration complexity of $O(\epsilon^{-3/2})$ for nonconvex optimization. *Mathematical Programming*, 162(1-2):1–32, 2017.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Rick Durrett. *Probability: theory and examples.* Cambridge university press, 2010.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 686–696, 2018.

Dan Garber and Elad Hazan. Fast and simple PCA via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle pointsonline stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Robert Gower, Nicolas Le Roux, and Francis Bach. Tracking the gradients using the Hessian: A new look at variance reducing stochastic methods. In *International Conference on Artificial Intelligence and Statistics*, pages 707–715, 2018.

Christopher J Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732, 2017.

Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 1042–1085. PMLR, 2018.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1895–1904. PMLR, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Aurelien Lucchi, Brian McWilliams, and Thomas Hofmann. A variance reduced stochastic Newton method. *arXiv preprint arXiv:1503.08316*, 2015.

Lester Mackey, Michael I Jordan, Richard Y Chen, Brendan Farrell, Joel A Tropp, et al. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3):906–945, 2014.

José Mario Martínez and Marcos Raydan. Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *Journal of Global Optimization*, 68(2):367–385, 2017.

Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic L-BFGS algorithm. In *Artificial Intelligence and Statistics*, pages 249–258, 2016.

Arkadii Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703, 2017.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pages 314–323, 2016a.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for smooth nonconvex optimization. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1971–1977. IEEE, 2016b.

Anton Rodomanov and Dmitry Kropotov. A superlinearly-convergent proximal Newton-type method for the optimization of finite sums. In *International Conference on Machine Learning*, pages 2597–2605, 2016.

Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

Clément W Royer and Stephen J Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM Journal on Optimization*, 28(2): 1448–1477, 2018. doi: 10.1137/17M1134329.

Shai Shalev-Shwartz. SDCA without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pages 747–754, 2016.

Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 2904–2913, 2018.

Joel A Tropp. The expected norm of a sum of independent random matrices: An elementary approach. In *High Dimensional Probability VII*, pages 173–202. Springer, 2016.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

H. Wai, W. Shi, A. Nedic, and A. Scaglione. Curvature-aided incremental aggregated gradient method. In *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 526–532, 2017. doi: 10.1109/ALLERTON.2017.8262782.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. Sample complexity of stochastic variance-reduced cubic regularization for nonconvex optimization. *arXiv preprint arXiv:1802.07372v1*, 2018a.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. Stochastic variance-reduced cubic regularization for nonconvex optimization. *arXiv preprint arXiv:1802.07372v2*, 2018b.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of Langevin dynamics based algorithms for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3125–3136, 2018a.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher R, and Michael W Mahoney. Sub-sampled Newton methods with non-uniform sampling. 2016.

Peng Xu, Farbod Roosta-Khorasan, and Michael W Mahoney. Second-order optimization for non-convex machine learning: An empirical study. *arXiv preprint arXiv:1708.07827*, 2017a.

Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *arXiv preprint arXiv:1708.07164*, 2017b.

Yi Xu, Jing Rong, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems*, pages 5531–5541, 2018b.

Chun Yu and Weixin Yao. Robust linear regression: A review and comparison. *Communications in Statistics-Simulation and Computation*, 46(8):6261–6282, 2017.

Yaodong Yu, Difan Zou, and Quanquan Gu. Saving gradient and negative curvature computations: Finding local minima more efficiently. *arXiv preprint arXiv:1712.03950*, 2017.

Yaodong Yu, Pan Xu, and Quanquan Gu. Third-order smoothness helps: Faster stochastic optimization algorithms for finding local minima. In *Advances in Neural Information Processing Systems*, pages 4526–4536, 2018.

Junyu Zhang, Lin Xiao, and Shuzhong Zhang. Adaptive stochastic variance reduction for subsampled Newton method with cubic regularization. *arXiv preprint arXiv:1811.11637*, 2018.

Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient Langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022, 2017.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularized Newton methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5990–5999. PMLR, 2018a.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduced gradient descent for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3922–3933, 2018b.

Journal of Machine Learning Research 20 (2019) 1-36 Submitted 1/19; Revised 7/19; Published 8/19

Gaussian Processes with Linear Operator Inequality Constraints

Christian Agrell CHRISAGR@MATH.UIO.NO
Department of Mathematics
University of Oslo
P.O. Box 1053 Blindern, Oslo N-0316, Norway

Group Technology and Research
DNV GL
P.O. Box 300, 1322 Høvik, Norway

Editor: Andreas Krause

Abstract

This paper presents an approach for constrained Gaussian Process (GP) regression where we assume that a set of linear transformations of the process are bounded. It is motivated by machine learning applications for high-consequence engineering systems, where this kind of information is often made available from phenomenological knowledge. We consider a GP f over functions on $\mathcal{X} \subset \mathbb{R}^n$ taking values in $\mathbb{R}$, where the process $\mathcal{L}f$ is still Gaussian when $\mathcal{L}$ is a linear operator. Our goal is to model f under the constraint that realizations of $\mathcal{L}f$ are confined to a convex set of functions. In particular, we require that $a \leq \mathcal{L}f \leq b$, given two functions a and b where $a < b$ pointwise. This formulation provides a consistent way of encoding multiple linear constraints, such as shape-constraints based on e.g. boundedness, monotonicity or convexity. We adopt the approach of using a sufficiently dense set of virtual observation locations where the constraint is required to hold, and derive the exact posterior for a conjugate likelihood. The results needed for stable numerical implementation are derived, together with an efficient sampling scheme for estimating the posterior process.

Keywords: Gaussian processes, Linear constraints, Virtual observations, Uncertainty Quantification, Computer code emulation

1. Introduction

Gaussian Processes (GPs) are a flexible tool for Bayesian nonparametric function estimation, and widely used for applications that require inference on functions such as regression and classification. A useful property of GPs is that they automatically produce estimates on prediction uncertainty, and it is often possible to encode prior knowledge in a principled manner in the modelling of prior covariance. Some early well-known applications of GPs are within spatial statistics, e.g. meteorology (Thompson, 1956), and in geostatistics (Matheron, 1973) where it is known as *kriging*. More recently, GPs have become a popular choice within probabilistic machine learning (Rasmussen and Williams, 2005; Ghahramani, 2015). Since the GPs can act as interpolators when observations are noiseless, GPs have also become the main approach for uncertainty quantification and analysis involving computer experiments (Sacks et al., 1989; Kennedy and O'Hagan, 2001).

©2019 Christian Agrell.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-065.html.

Often, the modeler performing function estimation has prior knowledge, or at least hypotheses, on some properties of the function to be estimated. This is typically related to the function shape with respect to some of the input parameters, such as boundedness, monotonicity or convexity. Various methods have been proposed for imposing these types of constraints on GPs (see Section 4.1 for a short review). For engineering and physics based applications, constraints based on integral operators and partial differential equations are also relevant (Jidling et al., 2017; Särkkä, 2011). What the above constraints have in common is that they are linear operators, and so any combination of such constraints can be written as a single linear operator. For instance, the constraints $a_1(\mathbf{x}) \leq f(\mathbf{x}) \leq b_1(\mathbf{x})$, $\partial f/\partial x_i \leq 0$ and $\partial^2 f/\partial x_j^2 \geq 0$ for some function (or distribution over functions) $f : X \to Y$, can be written as $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ for $a(\mathbf{x}) = [a_1(\mathbf{x}), -\infty, 0]$, $b(\mathbf{x}) = [b_1(\mathbf{x}), 0, \infty]$ and $\mathcal{L} : Y^X \to (Y^X)^3$ being the linear operator $\mathcal{L}f = [f, \partial f/\partial x_i, \partial^2 f/\partial x_j^2]$.

The motivation for including constraints is usually to improve predictions and to obtain a reduced and more realistic estimate on the uncertainty, the latter having significant impact for risk-based applications. For many real-world systems, information related to constraints in this form is often available from phenomenological knowledge. For engineering systems, this is typically knowledge related to some underlying physical phenomenon. Being able to make use of these constraint in probabilistic modelling is particularly relevant for high-consequence applications, where obtaining realistic uncertainty estimates in subsets of the domain where data is scarce is a challenge. Furthermore, information on whether these types of constraints are likely to hold given a set of observations is also useful for explainability and model falsification. For a broader discussion see (Agrell et al., 2018; Eldevik et al., 2018).

In this paper, we present a model for estimating a function f : $\mathbb{R}^{n_x} \to \mathbb{R}$ by a constrained GP (CGP) $f|D, a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$. Here D is a set of observations of $(\mathbf{x}_j, y_j)$, possibly including additive white noise, and $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ is a GP with mean $\mu(\mathbf{x})$ and covariance function $K(\mathbf{x}, \mathbf{x}')$ that are chosen such that existence of $\mathcal{L}f$ is ensured. Due to the linearity of $\mathcal{L}$, both $\mathcal{L}f|D$ and $f|D, \mathcal{L}f$ remain Gaussian, and our approach is based on modelling $f|D, \mathcal{L}f$ under the constraint $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$. To model the constraint that $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ for all inputs $\mathbf{x}$, we take the approach of using a finite set of input locations where the constraint is required to hold. That is, we require that $a(\mathbf{x}_v) \leq \mathcal{L}f(\mathbf{x}_v) \leq b(\mathbf{x}_v)$ for a finite set of inputs $\{\mathbf{x}_v\}$ called the set of *virtual observation locations*. With this approach the CGP is not guaranteed to satisfy the constraint on the entire domain, but a finite set of points $\{\mathbf{x}_v\}$ can be found so that the constraint holds globally with sufficiently high probability.

The model presented in this paper is inspired by the research on shape-constrained GPs, in particular (Wang and Berger, 2016; Da Veiga and Marrel, 2012, 2015; Riihimki and Vehtari, 2010; Golchi et al., 2015; Maatouk and Bay, 2017; López-Lopera et al., 2018). We refer to Section 4 for further discussion on these alternatives. In the case where $\mathcal{L} = \partial/\partial x_i$, our approach is most similar to that of Wang and Berger (2016), where the authors make use of a similar sampling scheme for noiseless GP regression applied to computer code emulation. Many of the approaches to constrained GPs, including ours, rely on the constraint to be satisfied at a specified set of virtual locations. The use of virtual constraint observations may seem *ad hoc* at first, as the set of virtual observation locations has to be dense enough to ensure that the constraint holds globally with sufficiently high probability. Inversion

of the covariance matrix of the joint GP may therefore be of concern, both because this scales with the number of observations cubed and because there is typically high serial correlation if there are many virtual observations close together. The general solution is then to restrict the virtual observation set to regions where the probability of occurrence of the constraint is low (Riihimki and Vehtari, 2010; Wang and Berger, 2016). According to Wang and Berger (2016), when they followed this approach in their experiments, they found that only a modest number of virtual observations were typically needed, that these points were usually rather disperse, and the resulting serial correlation was not severe. We draw the same conclusion in our experiments. There is also one benefit with the virtual observation approach, which is that implementation of constraints that only hold on subsets of the domain is straightforward.

For practical use of the model presented in this paper, we also pay special attention to numerical implementation. The computations involving only real observations or only virtual observations are separated, which is convenient when only changes to the constraints are made such as in algorithms for finding a sparse set of virtual observation locations or for testing/validation of constraints. We also provide the algorithms based on Cholesky factorization for stable numerical implementation, and an efficient sampling scheme for estimating the posterior process. These algorithms are based on derivation of the exact posterior of the constrained Gaussian process using a general linear operator, and constitutes the main contribution of this paper.

The paper is structured as follows: In Section 2 we state the results needed on GP regression and GPs under linear transformations. Our main results are given in Section 3, where we introduce the constrained GP (CGP) and present the model for GP regression under linear inequality constraints. In particular, given some training data, we derive the posterior predictive distribution of the CGP evaluated at a finite set of inputs, which is a compound Gaussian with a truncated Gaussian mean (Section 3.1). Section 3.2 presents an algorithm for sampling from the posterior, and parameter estimation is addressed in Section 3.3. Section 3.4 and Section 3.5 are dedicated to optimization of the set of virtual observation locations needed to ensure that the constraint holds with sufficiently high probability. Some relevant alternative approaches from the literature on GP's under linear constraints are discussed in Section 4, followed up by numerical examples considering monotonicity and boundedness constraints. A Python implementation is available at `https://github.com/cagrell/gp_constr`, together with the code used for the examples. We end with some concluding remarks in Section 5.

2. Gaussian Processes and Linear Operators

We are interested in GP regression on functions $\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}$ under the additional inequality constraint $a(\mathbf{x}) \leq \mathcal{L}\mathrm{f}(\mathbf{x}) \leq b(\mathbf{x})$ for some specified functions $a(\mathbf{x})$ and $b(\mathbf{x})$, and the class of linear operators $\{\mathcal{L}|\mathcal{L}\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_c}\}$. Here n_x and n_c are positive integers, and the subscripts are just used to indicate the relevant underlying space over $\mathbb{R}$. We will make use of the properties of GPs under linear transformations given below.

2.1. Gaussian Process Regression

We consider a Gaussian process $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ given as a prior over functions $\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}$, which is specified by its mean and covariance function

$$\begin{aligned} \mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] : \mathbb{R}^{n_x} \to \mathbb{R}, \\ K(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))] : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}. \end{aligned} \tag{1}$$

Let $\mathbf{x}$ denote a vector in $\mathbb{R}^{n_x}$ and X the $N \times n_x$ matrix of N such input vectors. The distribution over the vector $\mathbf{f}$ of N latent values corresponding to X is then multivariate Gaussian with

$$\mathbf{f}|X \sim \mathcal{N}(\mu(X), K(X, X)),$$

where $K(X, X')$ denotes the Gram matrix $K(X, X')_{i,j} = K(\mathbf{x}_i, \mathbf{x}'_j)$ for two matrices of input vectors X and X'. Given a set of observations $Y = [y_1, \ldots, y_N]^T$, and under the assumption that the relationship between the latent function values and observed output is Gaussian, $Y|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I_N)$, the predictive distribution for new observations X^* is still Gaussian with mean and covariance

$$\begin{aligned} \mathbb{E}[\mathbf{f}^*|X^*, X, Y] &= \mu(X^*) + K(X^*, X)[K(X, X) + \sigma^2 I_N]^{-1}(Y - \mu(X)), \\ \mathrm{cov}(\mathbf{f}^*|X^*, X, Y) &= K(X^*, X^*) - K(X^*, X)[K(X, X) + \sigma^2 I_N]^{-1} K(X, X^*). \end{aligned} \tag{2}$$

Here $\mathbf{f}^*|X^*$ is the predictive distribution of $f(X^*)$ and $\mathbf{f}^*|X^*, X, Y$ is the predictive posterior given the data X, Y. For further details see e.g. Rasmussen and Williams (2005).

2.2. Linear Operations on Gaussian Processes

Let $\mathcal{L}$ be a linear operator on realizations of $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$. As GPs are closed under linear operators (Rasmussen and Williams, 2005; Papoulis and Pillai, 2002), $\mathcal{L}f$ is still a GP [1]. We will assume that the operator produces functions with range in $\mathbb{R}^{n_c}$, but where the input domain $\mathbb{R}^{n_x}$ is unchanged. That is, the operator produces functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}^{n_c}$. This type of operators on GPs has also been considered by Särkkä (2011) with applications to stochastic partial differential equations. The mean and covariance of $\mathcal{L}f$ are given by applying $\mathcal{L}$ to the mean and covariance of the argument:

$$\begin{aligned} \mathbb{E}[\mathcal{L}f(\mathbf{x})] &= \mathcal{L}\mu(\mathbf{x}) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_c}, \\ \mathrm{cov}(\mathcal{L}f(\mathbf{x}), \mathcal{L}f(\mathbf{x}')) &= \mathcal{L}K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}^{n_c \times n_c}, \end{aligned} \tag{3}$$

1. We assume here that $\mathcal{L}f$ exists. For instance, if $\mathcal{L}$ involves differentiation then the process f must be differentiable. See e.g. (Adler, 1981) for details on proving existence.

and the cross-covariance is given as

$$\begin{aligned}\text{cov}(\mathcal{L}f(\mathbf{x}), f(\mathbf{x}')) &= \mathcal{L}K(\mathbf{x}, \mathbf{x}') : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}^{n_c}, \\ \text{cov}(f(\mathbf{x}), \mathcal{L}f(\mathbf{x}')) &= K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}^{n_c}.\end{aligned} \tag{4}$$

The notation $\mathcal{L}K(\mathbf{x}, \mathbf{x}')$ and $K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T$ is used to indicate when the operator acts on $K(\mathbf{x}, \mathbf{x}')$ as a function of $\mathbf{x}$ and $\mathbf{x}'$ respectively. That is, $\mathcal{L}K(\mathbf{x}, \mathbf{x}') = \mathcal{L}K(\mathbf{x}, \cdot)$ and $K(\mathbf{x}, \mathbf{x}')\mathcal{L} = \mathcal{L}K(\cdot, \mathbf{x}')$. With the transpose operator the latter becomes $K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T = (\mathcal{L}K(\cdot, \mathbf{x}'))^T$. In the following sections we make use of the predictive distribution (2), where observations correspond to the transformed GP under $\mathcal{L}$.

3. Gaussian Processes with Linear Inequality Constraints

Following Section 2.1 and Section 2.2, we let $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ be a GP over real valued functions on $\mathbb{R}^{n_x}$, and $\mathcal{L}$ a linear operator producing functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}^{n_c}$. The matrix X and the vector Y will represent N noise perturbed observations: $y_i = f(\mathbf{x}_i) + \varepsilon_i$ with ε_i i.i.d. $\mathcal{N}(0, \sigma^2)$ for $i = 1, \ldots, N$.

We would like to model the posterior GP conditioned on the observations X, Y, and on the event that $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ for two functions $a(\mathbf{x}), b(\mathbf{x}) : \mathbb{R}^{n_x} \to (\mathbb{R} \cup \{-\infty, \infty\})^{n_c}$, where $a_i(\mathbf{x}) < b_i(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$ and $i = 1, \ldots, n_c$. To achieve this approximately, we start by assuming that the constraint $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ only holds at a finite set of inputs $\mathbf{x}_1^v, \ldots, \mathbf{x}_S^v$ that we refer to as *virtual observation locations*. Later, we will consider how to specify the set of virtual observation locations such that the constraint holds for any $\mathbf{x}$ with sufficiently high probability. Furthermore, we will also assume that *virtual observations* of the transformed process, $\mathcal{L}f(\mathbf{x}_i^v)$, comes with additive white noise with variance σ_v^2. We can write this as $a(X^v) \leq \mathcal{L}f(X^v) + \varepsilon^v \leq b(X^v)$, where $X^v = [\mathbf{x}_1^v, \ldots, \mathbf{x}_S^v]^T$ is the matrix containing the virtual observation locations and ε^v is a multivariate Gaussian with diagonal covariance of elements σ_v^2.

We will make use of the following notation: Let $\widetilde{C}(X^v) \in \mathbb{R}^{S \times n_c}$ be the matrix with rows $(\widetilde{C}(X^v))_i = \mathcal{L}f(\mathbf{x}_i^v) + \varepsilon_i^v$ for i.i.d. $\varepsilon_i^v \sim \mathcal{N}(\mathbf{0}, \sigma_v^2 I_{n_c})$, and let $C(X^v)$ denote the event $C(X^v) := \cap_{i=1}^S \{a(\mathbf{x}_i^v) \leq (\widetilde{C}(X^v))_i \leq b(\mathbf{x}_i^v)\}$. $C(X^v)$ thus represents the event that the constraint $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) + \varepsilon^v \leq b(\mathbf{x})$ is satisfied for all points in X^v, and it is defined through the latent variable $\widetilde{C}(X^v)$.

In summary, the process we will consider is stated as

$$f|X, Y, X^v, C(X^v) := f|f(X) + \varepsilon = Y, a(X^v) \leq \mathcal{L}f(X^v) + \varepsilon^v \leq b(X^v),$$

where f is a Gaussian process, X, Y is the training data and X^v are the locations where the transformed process $\mathcal{L}f + \varepsilon^v$ is bounded. The additive noise ε and ε^v are multivariate Gaussian with diagonal covariance matrices of elements σ^2 and σ_v^2 respectively.

Here we assume that observations of all parts of $\mathcal{L}f$ comes with i.i.d. white noise with variance σ_v^2. The reason for this is mainly for numerical stability, where we in computations will choose a tiny variance to approximate noiseless observations. Similarly, σ^2 may be chosen as a fixed small number for interpolation in the standard GP regression setting. In the following derivations, the results for exact noiseless observations can be obtained by setting the relevant variance to zero.

We also assume that any sub-operator of $\mathcal{L}$ is constrained at the same set of virtual locations X^v. This is mainly for notational convenience, and this assumption will be relaxed in Section 3.5. In the following, we let N_v denote the total number of virtual observation locations. Here $N_v = S \cdot n_c$ for now, whereas we will later consider $N_v = \sum_{i=1}^{n_c} S_i$ where the i-th sub-operator is associated with S_i virtual observation locations.

3.1. Posterior Predictive Distribution

Our goal is to obtain the posterior predictive distribution $\mathbf{f}^*|X^*, X, Y, X^v, C(X^v)$. That is: the distribution of $\mathbf{f}^* = f(X^*)$ for some new inputs X^*, conditioned on the observed data $Y = f(X) + \varepsilon$ and the constraint $a(X^v) \leq \mathcal{L}f(X^v) + \varepsilon^v \leq b(X^v)$.

To simplify the notation we write $\mathbf{f}^*|Y, C$, excluding the dependency on inputs X, X^* and X^v (as well as any hyperparameter of the mean and covariance function). The posterior predictive distribution is given by marginalizing over the latent variable $\widetilde{C}$:

$$\begin{aligned}p(\mathbf{f}^*, C|Y) &= p(\mathbf{f}^*|C, Y)p(C|Y), \\ p(\mathbf{f}^*|C, Y) &= \int_{a(X^v)}^{b(X^v)} p(\mathbf{f}^*|\widetilde{C}, Y)p(\widetilde{C}|Y)d\widetilde{C}, \\ p(C|Y) &= \int_{a(X^v)}^{b(X^v)} p(\widetilde{C}|Y)d\widetilde{C},\end{aligned}$$

where the limits correspond to the hyper-rectangle in $\mathbb{R}^{N_v}$ given by the functions $a(\cdot)$ and $b(\cdot)$ evaluated at each $\mathbf{x}^v \in X^v$. The predictive distribution and the probability $p(C|Y)$ are given in Lemma 1. $p(C|Y)$ is of interest, as it is the probability that the constraint holds at X^v given the data Y.

In the remainder of the paper we will use the shortened notation $\mu^* = \mu(X^*)$, $\mu = \mu(X)$, $\mu^v = \mu(X^v)$ and $K_{X,X'} = K(X, X')$. For vectors with elements in $\mathbb{R}^{n_c}$, such as $\mathcal{L}\mu^v$, we interpret this elementwise. E.g. $\mathcal{L}\mu^v(X^v)$ is given by the column vector $[\mathcal{L}\mu(\mathbf{x}_1^v)_1, \ldots, \mathcal{L}\mu(\mathbf{x}_1^v)_{n_c}, \ldots, \mathcal{L}\mu(\mathbf{x}_S^v)_1, \ldots, \mathcal{L}\mu(\mathbf{x}_S^v)_{n_c}]$.

We start by deriving the posterior predictive distribution $\mathbf{f}^*$ at some new locations X^*. The predictive distribution is represented by a Gaussian, $\mathbf{f}^*|Y, C \sim \mathcal{N}(\mu(\mathbf{C}), \Sigma)$, for some fixed covariance matrix Σ and a mean $\mu(\mathbf{C})$ that depends on the random variable $\mathbf{C} = \widetilde{C}|Y, C$. The variable $\widetilde{C} = \mathcal{L}f(X^v) + \varepsilon^v$ remains Gaussian after conditioning on the observations Y, i.e. $\widetilde{C}|Y \sim \mathcal{N}(\nu_c, \Sigma_c)$ with some expectation ν_c and covariance matrix Σ_c that can be computed using (3, 4). Applying the constraints represented by the event C on the random variable $\widetilde{C}|Y$ just means restricting $\widetilde{C}|Y$ to lie in the hyper-rectangle defined by the bounds $a(X^v)$ and $b(X^v)$. This means that $\mathbf{C} = \widetilde{C}|Y, C$ is a truncated multivariate Gaussian, $\mathbf{C} \sim \mathcal{TN}(\nu_c, \Sigma_c, a(X^v), b(X^v))$. The full derivation of the distribution parameters of $\mathbf{C}$ and $\mathbf{f}^*|Y, C$ are given in Lemma 1 below, whereas Lemma 2 provides an alternative algorithmic representation suitable for numerical implementation.

Lemma 1 *The predictive distribution $\mathbf{f}^*|Y, C$ is a compound Gaussian with truncated Gaussian mean:*

$$\mathbf{f}^*|Y, C \sim \mathcal{N}(\mu^* + A(\boldsymbol{C} - \mathcal{L}\mu^v) + B(Y - \mu), \Sigma), \tag{5}$$

$$\boldsymbol{C} = \widetilde{C}|Y, C \sim \mathcal{TN}(\mathcal{L}\mu^v + A_1(Y-\mu), B_1, a(X^v), b(X^v)), \tag{6}$$

where $\mathcal{TN}(\cdot,\cdot,a,b)$ is the Gaussian $\mathcal{N}(\cdot,\cdot)$ conditioned on the hyper-rectangle $[a_1,b_1]\times\cdots\times[a_k,b_k]$, and

$$\begin{aligned} A_1 &= (\mathcal{L}K_{X^v,X})(K_{X,X}+\sigma^2 I_N)^{-1}, & B_1 &= \mathcal{L}K_{X^v,X^v}\mathcal{L}^T + \sigma_v^2 I_{N_v} - A_1 K_{X,X^v}\mathcal{L}^T, \\ A_2 &= K_{X^*,X}(K_{X,X}+\sigma^2 I_N)^{-1}, & B_2 &= K_{X^*,X^*} - A_2 K_{X,X^*}, \\ & & B_3 &= K_{X^*,X^v}\mathcal{L}^T - A_2 K_{X,X^v}\mathcal{L}^T, \end{aligned}$$
$$A = B_3 B_1^{-1}, \quad B = A_2 - AA_1, \quad \Sigma = B_2 - AB_3^T.$$

Moreover, the probability that the unconstrained version of $\boldsymbol{C}$ falls within the constraint region, $p(C|Y)$, is given by

$$p(C|Y) = p\left(a(X^v) \le \mathcal{N}(\mathcal{L}\mu^v + A_1(Y-\mu), B_1) \le b(X^v)\right), \tag{7}$$

and the unconstrained predictive distribution is

$$\mathbf{f}^*|Y \sim \mathcal{N}(\mu^* + A_2(Y-\mu), B_2).$$

The derivation in Lemma 1 is based on conditioning the multivariate Gaussian $(\mathbf{f}^*, Y, \widetilde{C})$, and the proof is given in Appendix A. For practical implementation the matrix inversions involved in Lemma 1 may be prone to numerical instability. A numerically stable alternative is given in Lemma 2 below.

In the following lemma, $Chol(K)$ is the lower triangular Cholesky factor of a matrix K. We also let $R = (P \setminus Q)$ denote the solution to the linear system $PR = Q$ for matrices P and Q, which may be efficiently computed when P is triangular using forward or backward substitution.

Lemma 2 *Let $L = Chol(K_{X,X}+\sigma^2 I_N)$, $v_1 = L \setminus K_{X,X^v}\mathcal{L}^T$ and $v_2 = L \setminus K_{X,X^*}$.*

Then the matrices in Lemma 1 can be computed as

$$\begin{aligned} A_1 &= (L^T \setminus v_1)^T, & B_1 &= \mathcal{L}K_{X^v,X^v}\mathcal{L}^T + \sigma_v^2 I_{N_v} - v_1^T v_1, \\ A_2 &= (L^T \setminus v_2)^T, & B_2 &= K_{X^*,X^*} - v_2^T v_2, \\ & & B_3 &= K_{X^*,X^v}\mathcal{L}^T - v_2^T v_1. \end{aligned}$$

Moreover, B_1 is symmetric and positive definite. By letting $L_1 = Chol(B_1)$ and $v_3 = L_1 \setminus B_3^T$ we also have

$$A = (L_1^T \setminus v_3)^T, \quad B = A_2 - AA_1, \quad \Sigma = B_2 - v_3^T v_3.$$

The proof is given in Appendix B. The numerical complexity of the procedures in Lemma 2 is $n^3/6$ for Cholesky factorization of $n \times n$ matrices and $mn^2/2$ for solving triangular systems where the unknown matrix is $n \times m$. In the derivation of Lemma 1 and Lemma 2, the order of operations was chosen such that the first Cholesky factor $L = Chol(K_{X,X}+\sigma^2 I_N)$ only depends on X. This is convenient in the case where the posterior $\mathbf{f}^*|Y, C$ is calculated multiple times for different constraints C or virtual observations X^v, but where the data X, Y remain unchanged.

3.2. Sampling from the Posterior Distribution

In order to sample from the posterior we can first sample from the constraint distribution (6), and then use these samples in the mean of (5) to create the final samples of $\mathbf{f}^*|Y, C$.

To generate k samples of the posterior at M new input locations, $[\mathbf{x}_1^*, \ldots, \mathbf{x}_M^*]^T = X^*$, we use the following procedure

Algorithm 3 *Sampling from the posterior distribution*

1. *Find a matrix Q s.t. $Q^T Q = \Sigma \in \mathbb{R}^{M\times M}$, e.g. by Cholesky or a spectral decomposition.*
2. *Generate $\widetilde{C}_k$, a $N_v \times k$ matrix where each column is a sample of $\widetilde{C}|Y, C$ from the distribution in (6).*
3. *Generate U_k, a $M \times k$ matrix with k samples from the standard normal $\mathcal{N}(\boldsymbol{0}, I_M)$.*
4. *The $M \times k$ matrix where each column in a sample from $\mathbf{f}^*|Y, C$ is then obtained by*

$$[\mu^* + B(Y-\mu)] \oplus_{col} [A(-\mathcal{L}\mu^v \oplus_{col} \widetilde{C}_k) + QU_k],$$

where $\oplus_{col}$ means that the $M \times 1$ vector on the left hand side is added to each column of the $M \times k$ matrix on the right hand side.

This procedure is based on the well-known method for sampling from multivariate Gaussian distributions, where we have used the property that in the distribution of $\mathbf{f}^*|Y, C$, only the mean depends on samples from the constraint distribution.

The challenging part of this procedure is the second step where samples have to be drawn from a truncated multivariate Gaussian. The simplest approach is by rejection sampling, i.e. generating samples from the normal distribution and rejection those that fall outside the bounds. In order to generate m samples with rejection sampling, the expected number of samples needed is $m/p(C|Y)$, where the acceptance rate is the probability $p(C|Y)$ given in (7). If the acceptance rate is low, then rejection sampling becomes inefficient, and an alternative approach such as Gibbs sampling (Kotecha and Djuric, 1999) is typically used. In our numerical experiments (presented in Section 4.2) we made use of a new method based on simulation via minimax tilting by Botev (2017), developed for high-dimensional exact sampling. Botev (2017) prove strong efficiency properties and demonstrate accurate simulation in dimensions $d \sim 100$ with small acceptance probabilities ($\sim 10^{-100}$), that take about the same time as one cycle of Gibbs sampling. For higher dimensions in the thousands, the method is used to accelerate existing Gibbs samplers by sampling jointly hundreds of highly correlated variables. In our experiments, we experienced that this method worked well in cases where Gibbs sampling was challenging. A detailed comparison with other sampling alternatives for an application similar to ours is also given in (López-Lopera et al., 2018). An important observation in Algorithm 3 is that for inference at a new set of input locations X^*, when the data X, Y and virtual observation locations X^v are unchanged, the samples generated in step *2* can be reused.

3.3. Parameter Estimation

To estimate the parameters of the CGP we make use of the marginal maximum likelihood approach (MLE). We define the marginal likelihood function of the CGP as

$$L(\theta) = p(Y, C|\theta) = p(Y|\theta)p(C|Y, \theta), \tag{8}$$

i.e. as the probability of the data Y and constraint C combined, given the set of parameters represented by θ. We assume that both the mean and covariance function of the GP prior (1) $\mu(\mathbf{x}|\theta)$ and $K(\mathbf{x}, \mathbf{x}'|\theta)$ may depend on θ. The log-likelihood, $l(\theta) = \ln p(Y|\theta) + \ln p(C|Y, \theta)$, is thus given as the sum of the unconstrained log-likelihood, $\ln p(Y|\theta)$, which is optimized in unconstrained MLE, and $\ln p(C|Y, \theta)$, which is the probability that the constraint holds at X^v given in (7).

In (Bachoc et al., 2018) the authors study the asymptotic distribution of the MLE for shape-constrained GPs, and show that for large sample sizes the effect of including the constraint in the MLE is negligible. But for small or moderate sample sizes the constrained MLE is generally more accurate, so taking the constraint into account is beneficial. However, due to the added numerical complexity in optimizing a function that includes the term $\ln p(C|Y, \theta)$, it might not be worthwhile. Efficient parameter estimation using the full likelihood (8) is a topic of future research. In the numerical experiments presented in this paper, we therefore make use of the unconstrained MLE. This also makes it possible to compare models with and without constraints in a more straightforward manner.

3.4. Finding the Virtual Observation Locations

For the constraint to be satisfied locally at any input location in some bounded set $\Omega \subset \mathbb{R}^{n_x}$ with sufficiently high probability, the set of virtual observation locations X^v has to be sufficiently dense. We will specify a target probability $p_{\text{target}} \in [0, 1)$ and find a set X^v, such that when the constraint is satisfied at all virtual locations in X^v, the probability that the constraint is satisfied for any $\mathbf{x}$ in Ω is at least p_{target}. The number of virtual observation locations needed depends on the smoothness properties of the kernel, and for a given kernel it is of interest to find a set X_v that is effective in terms of numerical computation. As we need to sample from a truncated Gaussian involving cross-covariances between all elements in X^v, we would like the set X^v to be small, and also to avoid points in X^v close together that could lead to high serial correlation.

Seeking an optimal set of virtual observation locations has also been discussed in (Wang and Berger, 2016; Golchi et al., 2015; Riihimki and Vehtari, 2010; Da Veiga and Marrel, 2012, 2015), and the intuitive idea is to iteratively place virtual observation locations where the probability that the constraint holds is low. The general approach presented in this section is most similar to that of Wang and Berger (2016). In Section 3.5 we extend this to derive a more efficient method for multiple constraints.

In order to estimate the probability that the constraint holds at some new location $\mathbf{x}^* \in \Omega$, we first derive the posterior distribution of the constraint process.

Lemma 4 *The predictive distribution of the constraint $\mathcal{L}f(\boldsymbol{x}^*)$ for some new input $\boldsymbol{x}^* \in \mathbb{R}^{n_x}$, condition on the data Y is given by*

$$\mathcal{L}f(\boldsymbol{x}^*)|Y \sim \mathcal{N}(\mathcal{L}\mu^* + \widetilde{A}_2(Y - \mu), \widetilde{B}_2), \tag{9}$$

and when $\mathcal{L}f(\boldsymbol{x}^*)$ is conditioned on both the data and virtual constraint observations, X, Y and $X^v, C(X^v)$, the posterior becomes

$$\mathcal{L}f(\boldsymbol{x}^*)|Y, C \sim \mathcal{N}(\mathcal{L}\mu^* + \widetilde{A}(\boldsymbol{C} - \mathcal{L}\mu^v) + \widetilde{B}(Y - \mu), \widetilde{\Sigma}). \tag{10}$$

Here L, v_1, A_1, B_1 and L_1 are defined as in Lemma 2 , $\boldsymbol{C}$ is the distribution in (6) *and*

$$\begin{aligned}
\widetilde{v}_2 &= L \setminus K_{X,\boldsymbol{x}^*}\mathcal{L}^T, & \widetilde{B}_2 &= \mathcal{L}K_{\boldsymbol{x}^*,\boldsymbol{x}^*}\mathcal{L}^T - \widetilde{v}_2^T\widetilde{v}_2, \\
\widetilde{A}_2 &= (L^T \setminus \widetilde{v}_2)^T, & \widetilde{B}_3 &= \mathcal{L}K_{\boldsymbol{x}^*,X^v}\mathcal{L}^T - \widetilde{v}_2^T v_1, \\
& & \widetilde{v}_3 &= L_1 \setminus \widetilde{B}_3^T, \\
\widetilde{A} = (L_1^T \setminus \widetilde{v}_3)^T, & \quad \widetilde{B} = \widetilde{A}_2 - \widetilde{A}A_1, & \widetilde{\Sigma} &= \widetilde{B}_2 - \widetilde{v}_3^T\widetilde{v}_3.
\end{aligned}$$

The proof is given in Appendix D. The predictive distribution in Lemma 4 was defined for a single input $\mathbf{x}^* \in \mathbb{R}^{n_x}$, and we will make use of the result in this context. But we could just as well consider an input matrix X^* with rows $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots$, where the only change in Lemma 4 is to replace $\mathbf{x}^*$ with X^*. In this case we also note that the variances, $\text{diag}(\widetilde{\Sigma})$, is more efficiently computed as $\text{diag}(\widetilde{\Sigma}) = \text{diag}(\mathcal{L}K_{X^*,X^*}\mathcal{L}^T) - \text{diag}(\widetilde{v}_2^T\widetilde{v}_2) - \text{diag}(\widetilde{v}_3^T\widetilde{v}_3)$ where we recall that $\text{diag}(v^T v)_i = \sum_j v_{i,j}^2$ for $v^T = [v_{i,j}]$.

Using the posterior distribution of $\mathcal{L}f$ in Lemma 4 we define the constraint probability $p_c : \mathbb{R}^{n_x} \to [0, 1]$ as

$$p_c(\mathbf{x}) = P\left(a(\mathbf{x}) - \nu < \xi(\mathbf{x}, X^v) < b(\mathbf{x}) + \nu\right), \tag{11}$$

where $\xi(\mathbf{x}, X^v) = \mathcal{L}f(\mathbf{x}^*)|Y$ for $X^v = \emptyset$ and $\xi(\mathbf{x}, X^v) = \mathcal{L}f(\mathbf{x}^*)|Y, C$ otherwise. The quantity ν is a non-negative fixed number that is included to ensure that it will be possible to increase p_c using observations with additive noise. When we use virtual observations $\widetilde{C}(\mathbf{x}) = \mathcal{L}f(\mathbf{x}^*) + \varepsilon^v$ that come with noise $\varepsilon^v \sim \mathcal{N}(0, \sigma_v^2)$, we can use $\nu = \max\{\sigma_v\Phi^{-1}(p_{\text{target}}), 0\}$ where $\Phi(\cdot)$ is the normal cumulative distribution function. Note that σ_v, and in this case ν, will be small numbers included mainly for numerical stability. In the numerical examples presented in this paper this noise variance was set to 10^{-6}.

In the case where $X^v = \emptyset$, computation of (11) is straightforward as $\xi(\mathbf{x}, X^v)$ is Gaussian. Otherwise, we will rely on the following estimate of $p_c(\mathbf{x})$:

$$\hat{p}_c(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m} P\left(a(\mathbf{x}) - \nu < (\mathcal{L}f(\mathbf{x})|Y, C_j) < b(\mathbf{x}) + \nu\right), \tag{12}$$

where $C_1, \ldots, C_m$ are m samples of $\mathbf{C}$ given in (6).

We outline an algorithm for finding a set of virtual observation locations X^v, such that the probability that the constraint holds locally at any $\mathbf{x} \in \Omega$ is at least p_{target} for some specified set $\Omega \subset \mathbb{R}^{n_x}$ and $p_{\text{target}} \in [0, 1)$. That is, $\min_{\mathbf{x}\in\Omega} p_c(\mathbf{x}) \geq p_{\text{target}}$. The algorithm can be used starting with no initial virtual observation locations, $X^v = \emptyset$, or using some pre-defined set $X^v \neq \emptyset$. The latter may be useful e.g. if the data X, Y is updated, in which case only a few additions to the previous set X^v might be needed.

Algorithm 5 *Finding locations of virtual observations X^v s.t. $\hat{p}_c(\boldsymbol{x}) \geq p_{target}$ for all $\boldsymbol{x} \in \Omega$.*

1. *Compute $L = Chol(K_{X,X} + \sigma^2 I_N)$.*

2. *Until convergence do:*

   (a) *If $X^v \neq \emptyset$ compute A_1 and B_1 as defined in Lemma 2, and generate m samples $C_1, \ldots, C_m$ of $\boldsymbol{C}$ given in* (6).

   (b) *If $X^v = \emptyset$ compute $(\boldsymbol{x}^*, p^*) = (\arg\min p_c(\boldsymbol{x}), p_c(\boldsymbol{x}^*))$. Otherwise compute $(\boldsymbol{x}^*, p^*) = (\arg\min \hat{p}_c(\boldsymbol{x}), \hat{p}_c(\boldsymbol{x}^*))$ with $\hat{p}_c$ defined as in* (12)*, using the samples generated in step (a).*

   (c) *Terminate if $p^* \geq p_{target}$, otherwise update $X^v \to X^v \cup \{\boldsymbol{x}^*\}$.*

The rate of convergence of Algorithm 5 relies on the probability that the constraint holds initially, $P\left(a(\mathbf{x}) < (\mathcal{L}f(\mathbf{x})|Y) < b(\mathbf{x})\right)$, and for practical application one may monitor p^* as a function of the number of virtual observation locations, $|X^v|$, to find an appropriate stopping criterion.

With the exception of low dimensional input $\mathbf{x}$, the optimization step $\mathbf{x}^* = \arg\min \hat{p}_c(\mathbf{x})$ is in general a hard non-convex optimization problem. But with respect to how $\mathbf{x}^*$ and p^* are used in the algorithm, some simplifications can be justified. First, we note that when computing $\hat{p}_c(\mathbf{x})$ with (12) for multiple $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2, \ldots$, the samples $C_1, \ldots, C_m$ are reused. It is also not necessary to find the the absolute minimum, as long as a *small enough* value is found in each iteration. Within the global optimization one might therefore decide to stop after the first occurrence of $\hat{p}_c(\mathbf{x})$ less than some threshold value. With this idea one could also search over finite candidate sets $\Omega \subset \mathbb{R}^{n_x}$, using a fixed number of random points in $\mathbb{R}^{n_x}$. This approach might produce a larger set X^v, but where the selection of $\mathbf{x}^*$ is faster in each iteration. Some of the alternative strategies for locating $\mathbf{x}^*$ in Algorithm 5 are studied further in our numerical experiments in Section 4.2.

With the above algorithm we aim to impose constraints on some bounded set $\Omega \subset \mathbb{R}^{n_x}$. Here Ω has to be chosen with respect to both training and test data. For a single boundedness constraint, it might be sufficient that the constraint only holds at the points $\mathbf{x} \in \mathbb{R}^{n_x}$ that will be used for prediction. But if we consider constraints related to monotonicity (see Example 1, Section 4.2), dependency with respect to the latent function's properties at the training locations is lost with this strategy. In the examples we give in this paper we consider a convex set Ω, in particular $\Omega = [0, 1]^{n_x}$, and assume that training data, test data and any input relevant for prediction lies within Ω.

3.5. Separating Virtual Observation Locations for Sub-operators

Let $\mathcal{L}$ be a linear operator defined by the column vector $[\mathcal{F}_1, \ldots, \mathcal{F}_k]$, where each $\mathcal{F}_i$ is a linear operator leaving both the domain and range of its argument unchanged, i.e. $\mathcal{F}_i$ produces functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}$, subjected to an interval constraint $[a_i(\mathbf{x}), b_i(\mathbf{x})]$. Until now we have assumed that the constrain holds at a set of virtual observation locations X^v, which means that $a_i(X^v) \leq \mathcal{F}_i f(X^v) \leq b_i(X^v)$ for all $i = 1, \ldots, k$.

However, it might not be necessary to constrain each of the sub-operators $\mathcal{F}_i$ at the same points $\mathbf{x}^v \in X^v$. Intuitively, constraints with respect to $\mathcal{F}_i$ need only be imposed at locations where $p(\mathcal{F}_i f(\mathbf{x}) \notin [a_i(\mathbf{x}), b_i(\mathbf{x})])$ is large. To accommodate this we let X^v be the concatenation of the matrices $X^{v,1}, \ldots, X^{v,k}$ and define $\mathcal{L}^T f(X^v) = [\mathcal{F}_1^T f(X^{v,1}), \ldots, \mathcal{F}_k^T f(X^{v,1})]^T$. This is equivalent to removing some of the rows in $\mathcal{L}(\cdot)(X^v)$, and all of the results in this paper still apply. In this setting we can improve the algorithm in Section 3.4 for finding the

set of virtual observation locations by considering each sub-operator individually. This is achieved using the estimated partial constraint probabilities, $p_{c,i}(\mathbf{x})$, that we defined as in (11) by considering only the i-th sub-operator. We may then use the estimate

$$\hat{p}_{c,i}(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} P\left(a_i(\mathbf{x}) - \nu < (\mathcal{L}f(\mathbf{x})|Y, C_j)_i < b_i(\mathbf{x}) + \nu\right), \tag{13}$$

where $(\mathcal{L}f(\mathbf{x})|Y, C_j)_i$ is the univariate Normal distribution given by the i-th row of $(\mathcal{L}f(\mathbf{x})|Y, C_j)$, and $C_1, \ldots, C_m$ are m samples of $\mathbf{C}$ given in (6) as before. Algorithm 5 can then be improved by minimizing (13) with respect to both $\mathbf{x}$ and $i = 1, \ldots k$. The details are presented in Appendix C, Algorithm 7.

3.6. Prediction using the Posterior Distribution

For the unconstrained GP in this paper where the likelihood is given by Gaussian white noise, the posterior mean and covariance is sufficient to describe predictions as the posterior remains Gaussian. It is also known that in this case there is a correspondence between the posterior mean of the GP and the optimal estimator in the Reproducing Kernel Hilbert Space (RKHS) associated with the GP (Kimeldorf and Wahba, 1970). This is a Hilbert space of functions defined by the positive semidefinite kernel of the GP. Interestingly, a similar correspondence holds for the constrained case. Maatouk et al. (2016) show that for constrained interpolation, the Maximum *A Posteriori* (MAP) or mode of the posterior is the optimal constrained interpolation function in the RKHS, and also illustrate in simulations that the unconstrained mean and constrained MAP coincide only when the unconstrained mean satisfies the constraint. This holds when the GP is constrained to a convex set of functions, which is the case in this paper where we condition on linear transformations of a function restricted to a convex set.

3.7. An Alternative Approach based on Conditional Expectations

Da Veiga and Marrel (2012, 2015) propose an approach for approximating the first two moments of the constrained posterior, $\mathbf{f}^*|Y, C$, using conditional expectations of the truncated multivariate Gaussian. This means, in the context of this paper, that the first two moments of $\mathbf{f}^*|Y, C$ are computed using the first two moments of the latent variable $\mathbf{C}$. To apply this idea using the formulation of this paper, we can make use of the following result.

Corollary 6 *Let the matrices A, B, Σ and the truncated Gaussian random variable $\boldsymbol{C}$ be as defined in Lemma 1, and let ν, Γ be the expectation and covariance of $\boldsymbol{C}$. Then the expectation and covariance of the predictive distribution $\boldsymbol{f}^*|Y, C$ are given as*

$$\begin{aligned} \mathbb{E}(\boldsymbol{f}^*|Y, C) &= \mu^* + A(\nu - \mathcal{L}\mu^v) + B(Y - \mu), \\ cov(\boldsymbol{f}^*|Y, C) &= \Sigma + A\Gamma A^T. \end{aligned} \tag{14}$$

Moreover, if $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{\Sigma}$ are the matrices defined in Lemma 4, then the expectation and variance of the predictive distribution of the constraint $\mathcal{L}f(\boldsymbol{x}^*)|Y, C$ are given as

$$\begin{aligned} \mathbb{E}(\mathcal{L}f(\boldsymbol{x}^*)|Y, C) &= \mathcal{L}\mu^* + \widetilde{A}(\nu - \mathcal{L}\mu^v) + \widetilde{B}(Y - \mu), \\ var(\mathcal{L}f(\boldsymbol{x}^*)|Y, C) &= \widetilde{\Sigma} + \widetilde{A}\Gamma\widetilde{A}^T. \end{aligned} \tag{15}$$

The results follows directly from the distributions derived in Lemmas 1 and 4, and moments of compound distributions. A proof is included in Appendix E for completeness.

Da Veiga and Marrel (2012, 2015) make use of a Genz approximation (Genz, 1992, 1997) to compute ν, Γ for inference using (14). They also introduce a crude but faster correlation-free approximation that can be used in the search for virtual observation locations. With this approach, (15) is used where ν, Γ are computed under the assumption that $\text{cov}(\widetilde{C}|Y)$ is diagonal. We can state this approximation as follows:

$$\nu_i \approx m_i + s_i \frac{\phi(\widetilde{a_i}) - \phi(\widetilde{b_i})}{\Phi(\widetilde{b_i}) - \Phi(\widetilde{a_i})}, \quad \Gamma_{i,i} \approx s_i^2 \left[1 + \frac{\widetilde{a_i}\phi(\widetilde{a_i}) - \widetilde{b_i}\phi(\widetilde{b_i})}{\Phi(\widetilde{b_i}) - \Phi(\widetilde{a_i})} - \left(\frac{\phi(\widetilde{a_i}) - \phi(\widetilde{b_i})}{\Phi(\widetilde{b_i}) - \Phi(\widetilde{a_i})} \right)^2 \right],$$

where m_i is the i-th component of $\mathbb{E}(\widetilde{C}|Y) = \mathcal{L}\mu^v + A_1(Y - \mu)$, $s_i = \sqrt{\text{cov}(\widetilde{C}|Y)_{i,i}} = \sqrt{(B_1)_{i,i}}$, $\widetilde{a_i} = (a(X^v)_i - m_i)/s_i$, $\widetilde{b_i} = (b(X^v)_i - m_i)/s_i$, ϕ and Φ are the pdf and cdf of the standard normal distribution and Γ is diagonal with elements $\Gamma_{i,i}$. We will make use of these approximations in some of the examples in Section 4.2 for comparison.

3.8. Numerical Considerations

For numerical implementation, we discuss some key considerations with the proposed model. One of the main issues with implementation of GP models in terms of numerical stability is related to covariance matrix inversion, which is why alternatives such as Cholesky factorization are recommended in practice. This does however not alleviate problems related to ill-conditioned covariance matrices. This is a common problem in computer code emulation (zero observational noise) in particular, where training points might be 'too close to each other' in terms of the covariance function, leaving the covariance matrix close to degenerate as some of the observations become redundant. A common remedy is to introduce a 'nugget' term on the diagonal entries of the covariance matrix, in the form of additional white noise on the observations. This means using a small $\sigma > 0$ instead of $\sigma = 0$ in Equation (2), even when the observations are noiseless. In terms of matrix regularization this is equivalent to Tikhonov regularization. See for instance Ranjan et al. (2010) and Andrianakis and Challenor (2012) which give a detailed discussion and recommendations for how to choose appropriate value for σ. In practice, a fixed small value is often used without further analysis, as long as the resulting condition number is not too high. This approach can be justified since the use of a nugget term has a straightforward interpretation, as opposed to other alternatives such as pseudoinversion. In our experiments on noiseless regression we fix $\sigma^2 = 10^{-6}$, as the error introduced by adding a variance of 10^{-6} to the observations is negligible.

Similarly, for the virtual observations used in this paper we make use of the noise parameter σ_v to avoid ill-conditioning of the matrix B_1 defined in Lemma 1. B_1 is the covariance matrix of the transformed GP, $\widetilde{C}|Y$, and B_1^{-1} together with $(K_{X,X} + \sigma^2 I_N)^{-1}$ are needed for all the posterior computations that involve constraints. The virtual noise parameter σ_v has a similar interpretation as σ, but where the artificial added noise acts on observations of the transformed process. Here $\sigma_v = 0$ means that the constraints are enforced with probability 1, $\sigma_v > 0$ implies that the constraints are enforced in a soft way, and $\sigma_v \to \infty$ provides no constraint at all. In the numerical examples presented in this

paper, a fixed value $\sigma_v^2 = 10^{-6}$ has been used to approximate hard constraints with an error we find negligible.

As for computational complexity, we may start by first looking at the operations involved in computing the posterior predictive distribution at M inputs $\mathbf{x}_1^*, \ldots, \mathbf{x}_M^*$ (including covariances), using Lemma 2. We first make note of the operations needed in the unconstrained case, i.e. standard GP regression with Gaussian noise, for comparison. If there are $N \geq M$ observations in the training set, then the complexity is dominated by the Cholesky factorization $L = Chol(K_{X,X} + \sigma^2 I_N)$, which require an order of N^3 operations and N^2 in memory. The Cholesky factor may be stored for subsequent predictions. Then, to compute the posterior predictive distribution at M new inputs, the number of operations needed is dominated by matrix multiplication and solving triangular systems, of orders NM^2 and N^2M. When a number N_v of virtual observation locations are included, we are essentially dealing with the same computations as the standard GP regression, but with $N + N_v$ number of observations. I.e. the computations involved are of order $(N + N_v)^3$ in time and $(N + N_v)^2$ in memory. The order of operations in Lemma 2 was chosen such that the Cholesky factor L that only depends on the training data can be reused. For a new set X^v of size N_v, the computations needed for prediction at M new locations X^* will only require the Cholesky factorization $L_1 = Chol(B_1)$ of order N_v^3. When both L and L_1 are stored, the remaining number of operations will be of order N^2M or N_v^2M for solving triangular systems, and NM^2, N_vM^2 or NMN_v for matrix multiplications.

In order to sample from the posterior using Algorithm 3, some additional steps are required. After the computations of Lemma 2 we continue to factorize the $M \times M$ covariance matrix Σ and generate samples from the truncated Gaussian $\widetilde{C}|Y, C$. The complexity involved in sampling from this N_v-dimensional truncated Gaussian depends on the sampling method of choice, see Section 3.2. We can combine k of these samples with k samples from a standard normal $\mathcal{N}(\mathbf{0}, I_M)$ to obtain samples of the final posterior, using an order of $MN_vk + M^2k$ operations. The total procedure of generating k samples at $M \leq N$ new inputs is therefore dominated by matrix operations of order $(N + N_v)^3$, MN_vk and M^2k, together with the complexity involved with sampling from a N_v-dimensional truncated Gaussian. For subsequent prediction it is convenient to here also reuse the samples generated from the truncated Gaussian, together with results that only involve X and X^v. This means storing matrices of size $N_v \times k$, $N \times N$ and $N_v \times N_v$. The remaining computations are then dominated by operations of order N^2M, N_v^2M, NM^2, N_vM^2, NMN_v, MN_vk, and M^2k. In the algorithms used to find virtual observation locations, Algorithm 5 and 7, we make sure to reuse computations that only involve the training data in each iteration of $N_v = 1, 2, \ldots$. This means that in addition to the previously stated operations, we need to perform Cholesky factorization of order N_v^3 and generate samples from a N_v-dimensional truncated Gaussian. This is initially very cheap, but becomes the main numerical challenge when N_v grows large. As the purpose of these algorithms is to find a small set X^v, that also avoids sampling issues due to serial correlation, we found it useful to output the minimal constraint probability p^* found in each iteration to reveal if the stopping criterion used (in terms of p_{target} or a maximum number of iterations) was unrealistic in practice.

4. Gaussian Process Modelling with Boundedness and Monotonicity Constraints

In this section we present some examples related to function estimation where we assume that the function and some of its partial derivatives are bounded. This is the scenario considered in the literature on shape-constrained GPs, and alternative approaches to GPs under linear constraints are usually presented in this setting. We start by a brief discussion on related work, followed by some numerical experiments using boundedness and monotonicity constraints. The numerical experiments were performed using the Python implementation available at `https://github.com/cagrell/gp_constr`.

4.1. Related Work

We give a brief overview of some alternative and related approaches to constrained GPs. For the approaches that rely on imposing constraints at a finite set of virtual observation locations, we recall that the constraint probability can be used in the search for a suitable set of virtual observation locations. The constraint probability is the probability that the constraint holds at an arbitrary input $\mathbf{x}$, $p_c(\mathbf{x})$ given in (11). Some key characteristics of the approaches that make use of virtual observations are summarized in Table 1.

The related work most similar to the approach presented in this paper is that of Wang and Berger (2016) and Da Veiga and Marrel (2012, 2015). Wang and Berger (2016) make use of a similar sampling scheme for noiseless GP regression applied to computer code emulation. A Gibbs sampling procedure is used for inference and to estimate the constraint probability $p_c(\mathbf{x})$ in the search for virtual observation locations. The approach of Da Veiga and Marrel (2012, 2015) is based on computation of the posterior mean and covariance of the constrained GP, using the equations that are also restated in this paper in Corollary 6. They make use of a Genz approximation for inference (Genz, 1992, 1997), and also introduce a crude but faster correlation-free approximation that can be used in the search for virtual observation locations. The approach of Da Veiga and Marrel (2012, 2015) is discussed further in the numerical experiments below, where we illustrate the idea in Example 1 and in Example 2 study an approximation of the posterior constrained GP using the constrained moments with a Gaussian distribution assumption. A major component in (Da Veiga and Marrel, 2012, 2015), (Wang and Berger, 2016) and this paper is thus computation involving the truncated multivariate Gaussian. Besides the choice of method for sampling from this distribution, the main difference with our approach is that we leverage Cholesky factorizations and noisy virtual observations for numerical stability.

A different approach that also make use of virtual observations is that of Riihimki and Vehtari (2010), where a *probit* likelihood is used to represent interval observations of the derivative process to impose monotonicity. They then make use of Expectation Propagation (EP) to approximate the posterior with a multivariate Gaussian. As pointed out by Golchi et al. (2015), the Gaussian assumption is questionable if the constraint (in this case monotonicity) does not hold with high probability a priori. Golchi et al. (2015) proceeds to develop a fully Bayesian procedure for application to computer experiments by the use of Sequentially Constrained Monte Carlo Sampling (SCMC). A challenge with this approach however is that finding a suitable set of virtual observation locations is difficult. Our experience, in agreement with (Wang and Berger, 2016; Da Veiga and Marrel, 2012,

	Virtual obs. likelihood	**Inference strategy**	**Strategy for finding X^v**
Agrell (2019)	Indicator + noise	Sampling (Minimax tilting)	Based on estimating $p_c(\mathbf{x})$ from samples
Wang and Berger (2016)	Indicator	Sampling (Gibbs)	Based on estimating $p_c(\mathbf{x})$ from samples
Da Veiga and Marrel (2012, 2015)	Indicator	Moment approximation (Genz)	Based on approximating $p_c(\mathbf{x})$ assuming Gaussian posterior distribution
Riihimki and Vehtari (2010)	Probit	Expectaion Propagation	Based on approximating $p_c(\mathbf{x})$ assuming Gaussian posterior distribution
Golchi et al. (2015)	Probit	SCMC	NA

Table 1: Summary of alternative approaches that make use of virtual observations. The table compares the likelihood used for virtual observations, the method used for inference and to determine the set of virtual observation locations X^v.

2015; Riihimki and Vehtari, 2010), is that for practical applications in more than a few dimensions, such a strategy is essential to avoid numerical issues related to high serial correlation, and also to reduce the number of virtual observation locations needed. It is also worth noting that a strategy that decouples computation involving training data and virtual observation locations from inference at new locations is beneficial. For the approaches discussed herein that rely on sampling/approximation related to the truncated multivariate Gaussian, the samples/approximations can be stored and reused as discussed in Section 3.8.

There are also some approaches to constrained GPs that are not based on the idea of using virtual observations. An interesting approach by Maatouk and Bay (2017), that is also followed up by López-Lopera et al. (2018), is based on modelling a conditional process where the constraints hold in the entire domain. They achieve this through finite-dimensional approximations of the GP that converge uniformly pathwise. With this approach, sampling from a truncated multivariate Gaussian is also needed for inference, in order to estimate the coefficients of the finite-dimensional approximation that arise from discretization of the input space. The authors give examples in 1D and 2D, but note that due to the structure of the approximation, the approach will be time consuming for practical applications in higher dimensions. There are also other approaches that consider special types of shape constraints, but where generalization seems difficult. See for instance (Abrahamsen and Benth, 2001; Yoo and Kyriakidis, 2006; Michalak, 2008; Kleijnen and Beers, 2013; Lin and Dunson, 2014; Lenk and Choi, 2017).

4.2. Numerical Experiments

In this section we will make us of the following constraints:

- $a_0(\mathbf{x}) \leq f(\mathbf{x}) \leq b_0(\mathbf{x})$
- $a_i(\mathbf{x}) \leq \partial f/\partial x_i(\mathbf{x}) \leq b_i(\mathbf{x})$

for all $\mathbf{x}$ in some bounded subset of $\mathbb{R}^{n_x}$, and $i \in \mathcal{I} \subset \{1, \ldots, n_x\}$. Without loss of generality we assume that the constrains on partial derivatives are with respect to the first k components of $\mathbf{x}$, i.e. $\mathcal{I} = \{1, \ldots, k\}$ for some $k \leq n_x$.

As the prior GP we will assume a constant mean $\mu = 0$ and make use of either the RBF or Matérn 5/2 covariance function. These are stationary kernels of the form

$$K(\mathbf{x}, \mathbf{x}') = \sigma_K^2 k(r), \; r = \sqrt{\sum_{i=1}^{n_x} \left(\frac{x_i - x_i'}{l_i} \right)^2}, \tag{16}$$

with variance parameter σ_K^2 and length scale parameters l_i for $i = 1, \ldots, n_x$. The radial basis function (RBF), also called squared exponential kernel, and the Matérn 5/2 kernel are defined through the function $k(r)$ as

$$k_{\text{RBF}}(r) = e^{-\frac{1}{2}r^2} \text{ and } k_{\text{Matérn } 5/2}(r) = (1 + \sqrt{5}r + \frac{5}{3}r^2)e^{-\sqrt{5}r}.$$

In general, the kernel hyperparameters σ_K^2 and l_i are optimized together with the noise variance σ through MLE. In the examples that consider noiseless observations, the noise variance is not estimated, but set to a small fixed value as discussed in Section 3.8. With the above choice of covariance function, existence of the transformed GP is ensured. In fact, the resulting process is infinitely differentiable using the RBF kernel (see Adler, 1981, Theorem 2.2.2) and twice differentiable with the Matérn 5/2. These prior GP alternatives were chosen as they are the most commonly used in the literature, and thus a good starting point for illustrating the effect of including linear constraints. We note that although it is not in general possible to design mean and covariance functions that produce GPs that satisfy the constraints considered in this paper, one could certainly ease numerical computations by selecting a GP prior based on the constraint probability $p(C|Y, \theta)$ in (7), and for instance make us of a mean function that is known to satisfy the constraint.

If we let $\mathcal{F}^0 f = f$, $\mathcal{F}^i f = \partial f/\partial x_i$, and $X^{v,i}$ be the set of S_i virtual observations corresponding to the i-th operator $\mathcal{F}^i$, then we can make use of the formulation in Section 3.5 and equations from Appendix C to obtain

$$\mathcal{L}\mu^v = [\mu \mathbf{1}_{S_0}, \mathbf{0}_{S_{[1,k]}}]^T,$$

where $\mathbf{1}_{S_1}$ is the vector $[1, \ldots, 1]^T$ of length S_1 and $\mathbf{0}_{S_{[1,k]}}$ is the vector $[0, \ldots, 0]^T$ of length $S_{[1,k]} = -S_0 + \sum S_i$. Furthermore,

$$K_{X,X^v}\mathcal{L}^T = \left[K_{X,X^{v,0}}, (K^{1,0}_{X^{v,1},X})^T, \ldots, (K^{k,0}_{X^{v,k},X})^T \right],$$

$$K_{X^*,X^v}\mathcal{L}^T = \left[K_{X^*,X^{v,0}}, (K^{1,0}_{X^{v,1},X^*})^T, \ldots, (K^{k,0}_{X^{v,k},X^*})^T \right],$$

$$\mathcal{L}K_{X^v,X^v}\mathcal{L}^T = \begin{bmatrix} K_{X^{v,0},X^{v,0}} & (K^{1,0}_{X^{v,1},X^{v,0}})^T & \ldots & (K^{k,0}_{X^{v,k},X^{v,0}})^T \\ K^{1,0}_{X^{v,1},X^{v,0}} & K^{1,1}_{X^{v,1},X^{v,1}} & \ldots & K^{1,k}_{X^{v,1},X^{v,k}} \\ \vdots & \vdots & \ddots & \vdots \\ K^{k,0}_{X^{v,k},X^{v,0}} & K^{k,1}_{X^{v,k},X^{v,1}} & \ldots & K^{k,k}_{X^{v,k},X^{v,k}} \end{bmatrix},$$

where we have used the notation

$$K^{i,0}(\mathbf{x}, \mathbf{x}') = \frac{\partial}{\partial x_i} K(\mathbf{x}, \mathbf{x}') \text{ and } K^{i,j}(\mathbf{x}, \mathbf{x}') = \frac{\partial^2}{\partial x_i \partial x_i'} K(\mathbf{x}, \mathbf{x}').$$

The use of constraints related to boundedness and monotonicity is illustrated using three examples of GP regression. Example 1 considers a function $\mathrm{f} : \mathbb{R} \to \mathbb{R}$ subjected to boundedness and monotonicity constraints. In Example 2 a function $\mathrm{f} : \mathbb{R}^4 \to \mathbb{R}$ is estimated under the assumption that information on whether the function is monotone increasing or decreasing as a function of the first two inputs is known, i.e. $\mathrm{sgn}(\partial \mathrm{f}/\partial x_1)$ and $\mathrm{sgn}(\partial \mathrm{f}/\partial x_2)$ are known. In Example 3 we illustrate how monotonicity constraints in multiple dimensions can be used in prediction of pressure capacity of pipelines.

4.2.1. **Example 1**: Illustration of Boundedness and Monotonicity in 1D

As a simple illustration of imposing constraints in GP regression, we first consider the function $\mathrm{f} : \mathbb{R} \to \mathbb{R}$ given by $\mathrm{f}(x) = \frac{1}{3}[\tan^{-1}(20x - 10) - \tan^{-1}(-10)]$. We assume that the function value is known at 7 input locations given by $x_i = 0.1 + 1/(i + 1)$ for $i = 1, \ldots, 7$. First, we assume that the observations are noiseless, i.e. $\mathrm{f}(x_i)$ is observed for each x_i. Estimating the function that interpolates at these observations is commonly referred to as *emulation*, which is relevant when dealing with data from computer experiments. Our function $\mathrm{f}(x)$ is both bounded and increasing on all of $\mathbb{R}$. In this example we will constrain the GP to satisfy the conditions that for $x \in [0, 1]$, we have that $\mathrm{df}/\mathrm{d}x \geq 0$ and $a(x) \leq \mathrm{f}(x) \leq b(x)$ for $a(x) = 0$ and $b(x) = \frac{1}{3}\ln(30x + 1) + 0.1$. The function is shown in Figure 1 together with the bounds and the 7 observations.

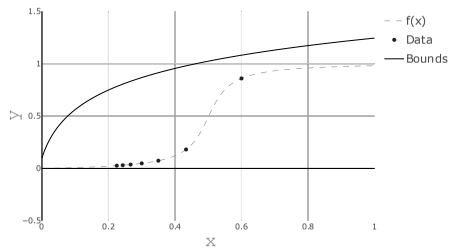


Figure 1: Function to emulate in Example 1

We select an RBF kernel (16) with parameters $\sigma_K = 0.5$ (variance) and $l = 0.1$ (length scale). To represent noiseless observations we set $\sigma^2 = 10^{-6}$, where σ^2 is the noise variance in the Gaussian likelihood. The assumed noise on virtual observations will also be set to 10^{-6}. To illustrate the effect of adding constraints we show the constrained GP using only boundedness constraint, only monotonicity constraint and finally when both constraints are imposed simultaneously. Figure 2 shows the resulting GPs. Algorithm 7 was used with a target probability $p_{target} = 0.99$ to determine the virtual observation locations that are indicated in the figures, and the posterior mode was computed by maximizing a Gaussian

kernel density estimator over the samples generated in Algorithm 3. For both constraints, 17 locations was needed for monotonicity and only 3 locations was needed to impose boundedness when the virtual locations for both constraints where optimized simultaneously. This is reasonable, as requiring $f(0) > 0$ is sufficient to ensure $f(x) > 0$ for $x \geq 0$ when f is increasing, and similarly requiring $f(x^v) < b(x^v)$ for some few points $x^v \in [0.6, 1]$ should suffice. But note that Algorithm 7 finds the virtual observation locations for both constraints simultaneously. Here $x^v = 0$ for boundedness was first identified, followed by some few points for monotonicity, followed by a new point x^v for boundedness etcetera.

For illustration purposes none of the hyperparameters of the GP were optimized. Moreover, for data sets such as the one in this example using plug-in estimates obtained from MLE generally not appropriate due to overfitting. Maximizing the marginal likelihood for the unconstrained GP gives a very poor model upon visual inspection ($\sigma_K = 0.86, l = 0.26$). However, it was observed that the estimated parameters for the constrained model (using Eq. (8)) gives estimates closer to the selected prior which seems more reasonable ($\sigma_K = 0.42, l = 0.17$), and hence the inclusion of the constraint probability, $p(C|Y, \theta)$, in the likelihood seems to improve the estimates also for the unconstrained GP.

We may also assume that the observations come with Gaussian white noise, which in terms of numerical stability is much less challenging than interpolation. Figure 3 shows the resulting GPs fitted to 50 observations. The observations were generated by sampling $x_i \in [0.1, 0.8]$ uniformly, and y_i from f$(x_i)+\varepsilon_i$ where ε_i are i.i.d. zero mean Gaussian with variance $\sigma^2 = 0.04$. Both GPs were optimized using plug-in estimates of hyperparameters (σ_K, l, σ^2) given by maximizing the marginal likelihood. These are $(\sigma_K = 0.34, l = 0.32, \sigma^2 = 0.053)$ for the constrained case and $(\sigma_K = 0.34, l = 0.23, \sigma^2 = 0.040)$ for the unconstrained case. We observe that the estimated noise variance is larger in the constrained model than the unconstrained where this estimate is exact.

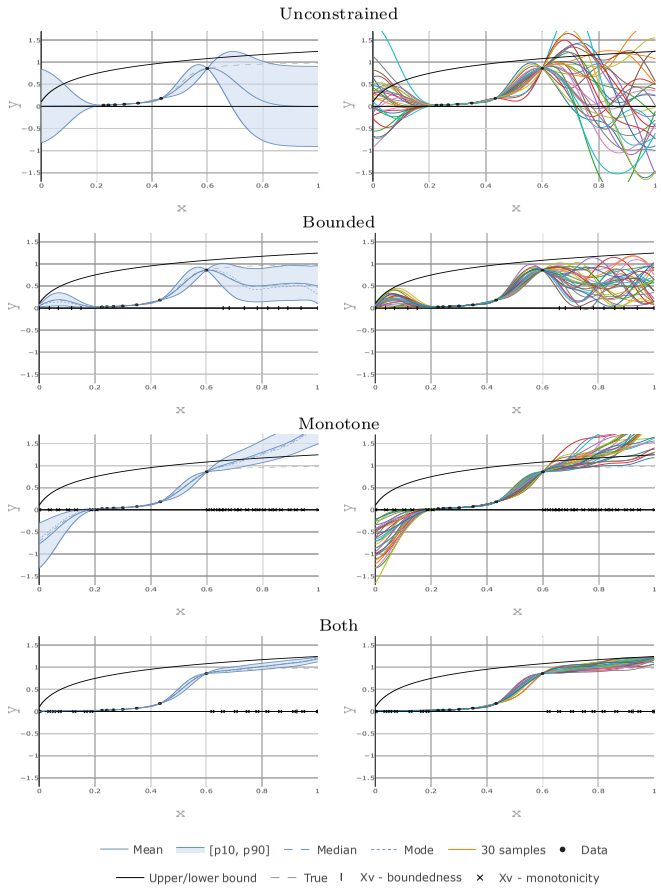


Figure 2: The GP with parameters $\sigma_K = 0.5$ (variance) and $l = 0.1$ (length scale) used in Example 1. The virtual observation locations are indicated by markers on the x-axis.

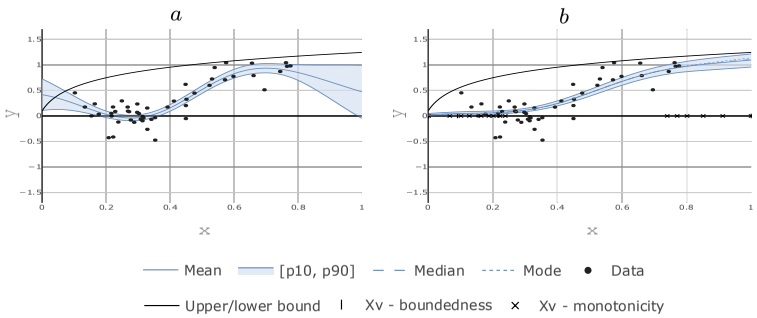


Figure 3: Unconstrained (a) and constrained (b) GPs fitted to 50 observations with Gaussian noise. The predictive distributions are shown, i.e. the distribution of $f(x)$ where $y = f(x) + \varepsilon$.

Da Veiga and Marrel (2015) propose to use estimates of the posterior mean and variance of $\mathcal{L}f(\mathbf{x})|Y, C$ to estimate the constraint probability $p_c(\mathbf{x})$ assuming a Gaussian distribution. They also introduce the faster correlation-free approximation, where the parameters are estimated under the assumption that observations of $\mathcal{L}f(\mathbf{x})|Y$ at different input locations $\mathbf{x}$ are independent (see Section 3.7). In Figure 4 we plot estimates of $p_{c,i}(\mathbf{x})$, for the boundedness and monotonicity constraint individually, using the approach in this paper (13) and the two moment based approximations. The plots were generated first after a total of 5 and then 10 virtual observations locations had been included in the model with both constraints. As we are mainly interested in finding $\mathbf{x}^* = \arg\min p_{c,i}(\mathbf{x})$, Figure 4 indicates that the moment based approximations are appropriate initially. However, as more virtual observation locations are included, the correlation-free assumption becomes questionable. But it could still serve as a useful starting point, and in a strategy based on checking the approximation error from time to time, it should still be possible to take advantage of the computational savings offered by the correlation-free approximation.

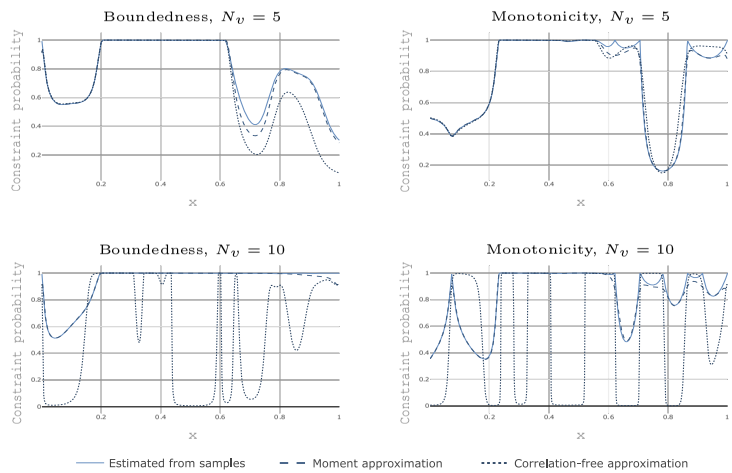


Figure 4: Constraint probability $p_c(\mathbf{x})$ computed using the estimate (13) together with the moment based approximations from Da Veiga and Marrel (2015). The constraint probability is shown for monotonicity and boundedness, where N_v is the total number of virtual observation locations used in the model.

4.2.2. **Example 2**: 4D Robot Arm Function

In this example we consider emulation of a function $\mathrm{f} : \mathbb{R}^4 \to \mathbb{R}$, where we assume that the sign of the first two partial derivatives, $\mathrm{sgn}(\partial \mathrm{f}/\partial x_1)$ and $\mathrm{sgn}(\partial \mathrm{f}/\partial x_2)$, are known. The function to emulate is

$$\mathrm{f}(\mathbf{x}) = \sum_{i=1}^{m} L_i \cos\left(\sum_{j=1}^{i} \tau_j\right),$$

for $m = 2$, and $\mathbf{x} = [L_1, L_2, \tau_1, \tau_2]$. The function is inspired by the robot arm function often used to test function estimation (An and Owen, 2001). Here $\mathrm{f}(\mathbf{x})$ is the y-coordinate of a two dimensional robot arm with m line segments of length $L_i \in [0, 1]$, positioned at angle $\tau_i \in [0, 2\pi]$ with respect to the horizontal axis. The constraints on the first two partial derivatives thus implies that it is known whether or not the arm will move further away from the x-axis, as a function of the arm lengths, L_1 and L_2, for any combination of τ_1 and τ_2.

In this experiment we first fit an unconstrained GP using 40 observations taken from a Latin hypercube sample over the input space $[0, 1]^2 \times [0, 2\pi]^2$. A Matérn 5/2 covariance function is used with plug-in MLE hyperparameters. Then, a total of 80 virtual observation locations are found using the procedure in Algorithm 7, where we search over a finite candi-

date set of 1000 locations in the minimization of the constraint probability. We repeat this procedure 100 times and report performance using the predictivity coefficient Q^2, predictive variance adequation (PVA) and the average width of 95% confidence intervals (AWoCI).

Given a set of tests $y_1, \ldots, y_{n_{test}}$ and predictions $\hat{y}_1, \ldots, \hat{y}_{n_{test}}$, Q^2 is defined as

$$Q^2 = 1 - \sum_{i=1}^{n_{test}} (\hat{y}_i - y_i)^2 / \sum_{i=1}^{n_{test}} (\bar{y} - y_i)^2,$$

where $\bar{y}$ is the mean of $y_1, \ldots, y_{n_{test}}$. In our experiments the predictions $\hat{y}_i$ are given by the posterior mean of the GP. The PVA criterion is defined as

$$\text{PVA} = \left| \log \left(\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \frac{(\hat{y}_i - y_i)^2}{\hat{\sigma}_i^2} \right) \right|,$$

where $\hat{\sigma}_i^2$ is the predictive variance. This criterion evaluates the quality of the predictive variances and to what extent confidence intervals are reliable. The smaller the PVA is, the better (Bachoc, 2013). In addition to this criterion, it is also useful to evaluate the size of confidence intervals. For this we compute the average width of 95% confidence intervals

$$\text{AWoCI} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (p_{0.975}^{(i)} - p_{0.025}^{(i)}),$$

where $p_{0.975}^{(i)}$ and $p_{0.025}^{(i)}$ are the predicted 97.5% and 2.5% percentiles.

The result of 100 predictions for one single experiment is shown in Figure 5. As expected, the estimated prediction uncertainty is reduced significantly using the constrained model, and single predictions given by the posterior mean are also improved. In Table 2 we summarize the results from running 100 of these experiments. In each experiment, Q^2, PVA and AWoCI was computed from prediction at 1000 locations sampled uniformly in the domain. We also report the probability that the constraint holds in the unconstrained GP, $p(C|Y)$ given in (7), and the CPU time in seconds used to generate 10^4 samples from the posterior on an Intel® Core™ i5-7300U 2.6GHz CPU. For comparison, we also include predictions from moment-based approximations using the approach of Da Veiga and Marrel (2012, 2015). We study in particular their approach for finding the set of virtual observation locations, as discussed in Section 3.7 and illustrated in the previous example. In total, the following alternatives are considered:

1. **Unconstrained**: The initial GP without constraints.
2. **Constrained**: The constrained GP using the approach presented in this paper.
3. **Moment approx. 1**: Using the sampling scheme of this paper for inference, but where the moment based approximation is used in the search for virtual observation locations.
4. **Moment approx. 2**: Using moment approximation for both inference and searching for virtual observation locations. This is one of the procedures from Da Veiga and Marrel (2012, 2015).

5. **Correlation-free approx.**: Same as **Moment approx. 1** but where the correlation-free approximation is used in the search for virtual observation locations.

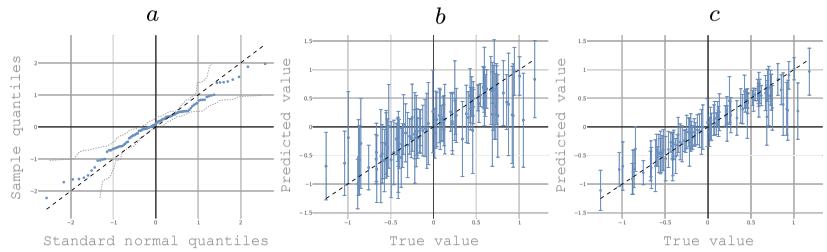


Figure 5: Figure a shows a qq-plot with 95% confidence band of 100 normalized residuals $(y_i - \mu_i)/(\sigma_i)$, where μ_i and σ_i^2 are the mean and variance of the predictive distribution of the unconstrained GP. In Figure b, predictions vs the true function value is shown together with a $[0.025, 0.975]$ (95%) percentile interval for the unconstrained GP. The same type of figure is shown in c for the constrained GP.

In Table 2 we see that the use of constraints is beneficial in terms of both a higher Q^2 (better predictive performance) and a smaller PVA (higher quality of predictive variances). With the exception of 'Moment approx. 2', the inclusion of constraints provides significant uncertainty reduction as the width of 95% confidence intervals (AWoCI) are reduce by almost a factor of 2 on average. A box plot showing AWoCI from the 100 experiments is also shown in Figure 6. We see that the different approaches for estimating the constraint probability, $p_c(\mathbf{x})$, in the search for virtual observation locations work equally well. The Gaussian assumption on the posterior $\mathbf{f}^*|Y, C$ on the other hand is not optimal, as it tends to overestimate the uncertainty in this example.

	$p(C\|Y)$	T_s	PVA	Q^2	AWoCI
Unconstrained			3.03	0.7558	0.99
Constrained	4.1E-34	24.8	2.85	0.8842	0.54
Moment approx. 1	2.4E-36	25.2	2.84	0.8844	0.54
Moment approx. 2	2.4E-36	25.2	2.84	0.8844	0.83
correlation-free approx.	8.6E-37	21.1	2.91	0.8775	0.55

Table 2: Average values from 100 experiments of the robot arm function. T_s is the CPU time in seconds used to generate 10^4 samples.

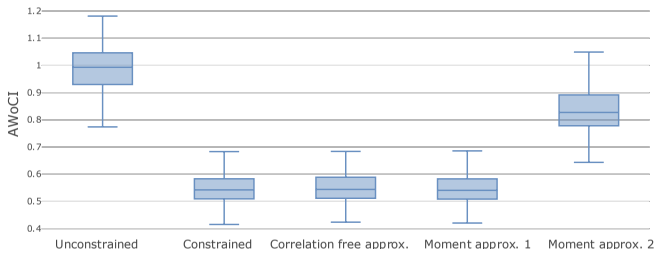


Figure 6: Average width of confidence intervals (AWoCI) from 100 experiments of the robot arm function.

4.2.3. **Example 3**: Pipeline Pressure Capacity

In this example we consider a model for predicting the pressure capacity of a steel pipeline with defects due to corrosion. As corrosion is one of the major threats to the integrity of offshore pipelines, experiments are carried out to understand how metal loss due to corrosion affects a pipeline's capacity with respect to internal pressure (Sigurdsson et al., 1999; Amaya et al., 2019). These include full scale burst tests and numerical simulation through Finite Element Analysis (FEA). Results from this type of experiments serve as the basis for current methodologies used in the industry for practical assessment of failure probabilities related to pipeline corrosion, such as ASME B31G or DNVGL-RP-F101. We consider experiments related to a single rectangular shaped defect, which is essential to these methodologies.

To simulate synthetic experiments of the burst capacity of a pipeline with a rectangular defect, we will use the simplified capacity equation given in in (RP-F101 DNV GL, 2017). The maximum differential pressure (capacity in MPa) the pipeline can withstand without

bursting is in the simplified equation given as

$$P_{cap}(\sigma_u, D, t, d, l) = 1.05\frac{2t\sigma_u}{D-t}\frac{1-d/t}{1-\frac{d/t}{Q}}, \quad Q = \sqrt{1+0.31\frac{l^2}{Dt}},$$

where $\sigma_u \in [450, 550]$ (MPa) is the ultimate tensile strength of the material, $D \in [10t, 50t]$ (mm) and $t \in [5, 30]$ (mm) are the outer diameter and wall thickness of the pipeline, and $d \in [0, t]$ (mm) and $l \in [0, 1000]$ (mm) are the depth and length of the rectangular defect.

From the physical phenomenon under consideration, we know that the capacity of the pipeline will decrease if the size of the defect were to increase. Similarly, we know that the pipeline capacity increases with a higher material strength or wall thickness, and decreases as a function of the diameter, all else kept equal. In the form of partial derivatives we can express this information as: $\frac{\partial P_{cap}}{\partial d} < 0$, $\frac{\partial P_{cap}}{\partial l} < 0$, $\frac{\partial P_{cap}}{\partial \sigma_u} > 0$, $\frac{\partial P_{cap}}{\partial t} > 0$ and $\frac{\partial P_{cap}}{\partial D} < 0$.

For convenience we will transform the input variables to the unit hypercube. Let $\mathbf{x}$ denote the transformed input vector $\mathbf{x} = [x_1, \ldots, x_5]$, where $x_1 = (\sigma_u - 450)/(550 - 450)$, $x_2 = (D/t - 10)/(50 - 10)$, $x_3 = (t - 5)/(30 - 5)$, $x_4 = d/t$ and $x_5 = l/1000$. We will make use of the function

$$f(\mathbf{x}) = P_{cap}(\mathbf{x}) \text{ for } \mathbf{x} \in [0, 1]^5,$$

and assume that the burst capacity observed in an experiment is $f(\mathbf{x}) + \varepsilon$, where ε is a zero mean Normal random variable with variance $\sigma^2 = 4$. The constraints on the partial derivatives after the transformation becomes: $\frac{\partial f}{\partial x_1} > 0$, $\frac{\partial f}{\partial x_2} < 0$, $\frac{\partial f}{\partial x_3} > 0$, $\frac{\partial f}{\partial x_4} < 0$ and $\frac{\partial f}{\partial x_5} < 0$ for $\mathbf{x} \in [0, 1]^5$.

In this example we thus have five constraints available, represented by bounds on the partial derivative of $f(\mathbf{x})$ w.r.t. x_i for $i = 1, \ldots, 5$. Besides studying the effect of including all five constraints, we will test some different alternatives using a smaller number of constraints, and also lower input dimensions. To simulate a lower dimensional version of the capacity equation, we can consider only the fist n_x input variables and keep the remaining variables fixed. We consider $n_x = 3, 4$ and 5 where we fix $x_i = 0.5$ for all $i > n_x$. For each of these scenarios we will consider n_x and $n_x - 1$ number of constraints. We let n_c denote the number of constraints, where using n_c constraints means that the bound on $\partial f / \partial x_i$ is included for $i = 1, \ldots, n_c$.

In each experiment we start by generating a training set of $N = 5n_x$ or $N = 10n_x$ LHS samples from $[0, 1]^{n_x}$. As in the previous example in Section 4.2.2, we fit a zero mean GP using a Matérn 5/2 covariance function and plug-in hyperparameters by MLE. We search over a candidate set consisting of 2500 uniform samples from $[0, 1]^{n_x}$ iteratively to update the set of virtual observation locations, until the constraint probability at all locations in the candidate set, and for each constraint, is at least 0.7. To check whether this is a reasonable stopping criterion we finish by minimizing the constraint probability for each constraint, using the differential evolution (Storn and Price, 1997) global optimization algorithm available in (SciPy Jones et al., 2001–).

Table 3 shows the results for different combinations of input dimensionality n_x, number of constraints n_c and number of training samples N, where the results in each row is computed from 100 experiments. As in the previous example we report $p(C|Y)$, PVA, Q^2 and AWoCI, and the CPU time spent generating samples for prediction (T_s). We also report the average CPU time used in the search for a new virtual observation location and

n_x	n_c	N	N_v	T_v	$p(C\|Y)$	$p_{c,\min}$	T_s	PVA	Q^2	AWoCI
3	2	15	3.6	0.6	2.6E-01	0.79	0.05	0.94 (0.89)	0.95 (0.95)	3.9 (6.2)
3	2	30	3.5	0.6	2.5E-01	0.78	0.04	0.89 (0.87)	0.97 (0.97)	3.0 (4.8)
3	3	15	5.8	0.9	1.2E-01	0.74	0.09	1.47 (1.23)	0.95 (0.95)	3.7 (6.1)
3	3	30	3.9	0.9	2.2E-01	0.76	0.04	0.79 (0.79)	0.97 (0.97)	3.1 (5.0)
4	3	20	11.8	0.9	1.5E-02	0.67	0.19	1.40 (1.29)	0.87 (0.92)	5.5 (9.4)
4	3	40	11.7	0.9	6.6E-03	0.71	0.18	0.51 (0.52)	0.97 (0.97)	4.1 (6.9)
4	4	20	13.6	1.2	6.9E-03	0.65	0.49	1.56 (1.31)	0.91 (0.91)	5.5 (9.6)
4	4	40	12.8	1.2	2.7E-03	0.69	0.19	0.50 (0.48)	0.97 (0.97)	4.0 (6.7)
5	4	25	14.8	1.2	6.3E-03	0.66	0.22	1.03 (1.08)	0.85 (0.83)	8.3 (14.3)
5	4	50	17.4	1.2	1.2E-03	0.66	0.26	0.73 (0.78)	0.90 (0.90)	6.8 (11.5)
5	5	25	15.5	1.5	3.1E-03	0.65	0.24	1.12 (1.10)	0.82 (0.81)	8.4 (14.4)
5	5	50	20.2	1.6	1.1E-03	0.61	0.35	0.67 (0.77)	0.90 (0.90)	6.5 (11.3)

Table 3: Average values from 100 experiments with input dimensionality n_x, number of constraints n_c and number of training samples N. Values in parenthesis correspond to the unconstrained model. Here $p_{c,\min}$ is the minimum of the constraint probability for any constraint over the entire domain after a total of N_v virtual observation locations have been included. T_v is the average CPU time in seconds used to find each of the N_v points using 10^3 samples, and T_s is the CPU time in seconds used to generate 10^4 samples of the final model for prediction.

the minimum constraint probability, $p_{c,\min} = \min_{i=1,\dots n_c} \min_{\mathbf{x}\in[0,1]^{n_x}} \hat{p}_{c,i}(\mathbf{x})$ (13), computed with differential evolution. Here we make use of 10^3 samples to compute the estimate $\hat{p}_{c,i}(\mathbf{x})$, whereas 10^4 samples are used for the final prediction.

From Table 3 we first notice that the number of virtual observation locations (N_v) determined by the searching algorithm is fairly low. One might interpret this as an indication that the unconstrained GP produces samples that are likely to agree with the monotonicity constraints, except for at a few locations. As a result, computation that involve sampling from the truncated multivariate Gaussian is efficient. Still, we see that inclusion of the constraints has an effect on uncertainty estimates as the AWoCI is reduced by a factor of around 1.6 in each experiment, whereas PVA and Q^2 are fairly similar for the unconstrained and constrained model overall. We also notice that the smallest constraint probability found in the domain using a global optimization technique is reduced when the number of constraints or dimensionality is increased. This is expected, as we only considered a finite candidate set and not the entire domain when searching for the location minimizing the constraint probability. Hence, if we really want to achieve a minimal constraint probability larger than 0.7 in 5 dimensions, more than 2500 samples in the candidate set would be needed with this strategy, or a global optimizer could be used to identify the remaining virtual observation locations needed.

For the application considered in this example, where uncertainty in the prediction is key to risk assessment, we argue that the effect the constraints have on uncertainty estimates makes the inclusion of constraints worthwhile. Modern engineering methodologies that

make use of capacity predictions as the one illustrated in this example are usually derived in the context of Structural Reliability Analysis (SRA), where the capacity is combined with a probabilistic representation of load (in this case differential pressure) to estimate the probability of failure (Madsen et al., 2006).

Alternative methods based on conservative estimates to ensure sufficient safety margin between load and capacity are also common. For the application considered herein, this would typically mean using a lower percentile instead of the posterior mean in order to represent a conservative capacity. The inclusion of constraints can therefore help to avoid unnecessary conservatism due to unphysical scenarios, that are not realistic but have positive probability in the unconstrained model.

Finally, we note that the constraints used in this example are not from differentiating the equation used as stand-in for experiments, but from knowledge related to the underlying physical phenomenon. The constraints therefore remain applicable, were the experiments to come from physical full-scale tests. This naturally also holds in applications to computer code emulation, where we would set the noise term ε to zero in this example if we were to assume that the capacity experiments came from a numerical (FEA) simulation. With results from this type of numerical simulation, a noise parameter is usually added to the simulation output as well, to represent model uncertainty as the numerical simulation is not a perfect representation of the real physical phenomenon. Very often the model uncertainty is represented by a univariate Gaussian. An interesting alternative here is to instead account for the model uncertainty as observational noise in the GP, where the use of constraints may help to obtain a more realistic model uncertainty as well.

5. Discussion

The model presented in this paper provides a consistent approach to GP regression under multiple linear constraints. The computational framework used is based on a sampling scheme which is exact in the limit. However, sampling strategies like the one in this paper can be too numerically demanding as opposed to approximation methods such as Laplace approximations, variational Bayesian inference, expectation propagation etcetera. The choice of using a sampling-based approach came from the author's intended use, which relates to machine learning for high-risk and safety-critical engineering applications (Agrell et al., 2018). For these applications, a proper treatment of uncertainty with respect to risks and the overall reliability of the system under consideration is essential. Making predictions based on past observations in this setting is challenging, as the consequence of wrong predictions may be catastrophic. In addition, critical consequences often relate to infrequent or low probability events, where relevant data is naturally scarce. However, there is usually additional knowledge available, and todays methods for assessing risk tend to rely heavily on understanding the underlying physical phenomenon. We gave an example in Section 4.2.3 considering prediction of the burst capacity of a pipeline, that may serve as a component in a larger model of system reliability. Such models are often graphical, e.g. Bayesian networks, that are derived from known causal dependencies. In this scenario it is essential that the accuracy of numerical estimation- or approximation methods can be assessed. In the case where simulation-based methods cannot be used due to computational limitations, they still serve as a useful benchmark that can help in the development and assessment of

suitable approximation-based algorithms. As for the simulation scheme in this paper, the only computational burden lies in sampling from a truncated multivariate Gaussian. As this is a fairly general problem, multiple good samplers exist for this purpose. We found the method of Botev (2017) to work particularly well for our applications, as it provides exact sampling in a relevant range of dimensions where many alternative sampling schemes fail. Based on a comparison made by López-Lopera et al. (2018), we see that the method based on Hamiltonian Monte Carlo by Pakman and Paninski (2012) may also be appropriate.

As we discuss briefly in Section 3.3, estimation of hyperparameters becomes challenging when the term $p(C|Y,\theta)$ enters the likelihood. Moreover, as our approach is based on the use of virtual observation locations, we are aware that the task of estimating or optimizing model hyperparameters in general is not well defined. This is because the likelihood depends both on the hyperparameters and the set of virtual observation locations (Eq. 8). This problem is neglected in the literature on shape-constrained GPs, where it is either assumed that the virtual observation locations are known a priori (for low input dimension selecting a space filling sufficiently dense design is unproblematic), or the hyperparameters are addressed independently of these. To our knowledge the problem of simultaneously estimating hyperparameters and virtual observation locations has not yet been addressed. A rather simplistic approach is to iterate between estimating hyperparameter and the set of virtual observation locations. However, for higher input dimensions this might be problematic altogether, in which case sparse approximations may be needed to deal with a large set of virtual observation locations. In this setting, it might be more fruitful to view the virtual observation locations as additional hyperparameters, in a model approximating the posterior corresponding to an sufficiently dense set of virtual observation locations, e.g. as in the inducing points framework for scaling GPs to large data sets (de G. Matthews et al., 2016). This is a topic of further research.

With the approach in this paper, we make use of the probability $p(C|Y)$, which is interesting in its own for investigating whether constraints such as e.g. monotonicity are likely to hold given a set of observations. Alternatively, inference on the constraint noise parameter σ_v can provide similar type of information. Ideally, we choose a small fixed value for σ_v to avoid numerical instability, as discussed in Section 3.8. But in extreme cases, with conflicting constraints or observations that contradict constraints with high probability, the model may still experience numerical issues. We argue that models that 'break' under these circumstances are preferred as it reveals that either 1) there is something wrong with the observations, or 2) there is something wrong with the constraints and hence our knowledge of the underlying phenomenon (Agrell et al., 2018). It would nevertheless be better if more principled ways of investigating such issues were available. In our experiments we observed that the conditional likelihood, $p(Y|C)$, in general is decreasing as a function of σ_v, whereas this was not the case for an invalid constraint assuming a monotonic *decreasing* function in Example 1. Hence, σ_v might provide useful information in this manner. The estimated partial constraint probabilities $\hat{p}_{c,i}(\mathbf{x})$ can also be useful for revealing such issues, for instance by monitoring the intermediate minimum values p_i^* computed in Algorithm 7 as new virtual observation locations are added.

Finally, we note that as the model presented in this paper relies on conditioning on a transformed GP with values in $\mathbb{R}^{n_c}$, it could be extended to multi-output GPs over functions

$\mathrm{f}:\mathbb{R}^{n_x}\to\mathbb{R}^{n_y}$ in a natural way. But for non-Gaussian likelihoods, or applications with large or high-dimensional data, other approximation based alternatives are needed.

Acknowledgments

This work has been supported by grant 276282 from the Norwegian Research Council and DNV GL Group Technology and Research. The research is part of an initiative on applying constraints based on phenomenological knowledge in probabilistic machine learning for high-risk applications, and the author would like to thank colleagues at DNV GL and the University of Oslo for fruitful discussions on the topic. A special thanks to Arne B. Huseby, Simen Eldevik, Andreas Hafver, and the editor and reviewers of JMLR for insightfull comments that have greatly improved the paper.

Appendix A. Proof of Lemma 1

Proof. We start by observing that $(\mathbf{f}^*,\widetilde{C},Y)$ is jointly Gaussian with mean and covariance

$$\mathbb{E}([\mathbf{f}^*,\widetilde{C},Y]^T)=[\mu^*,\mathcal{L}\mu^v,\mu]^T, \tag{17}$$

$$\mathrm{cov}([\mathbf{f}^*,\widetilde{C},Y]^T)=\begin{bmatrix} K_{X^*,X^*} & K_{X^*,X^v}\mathcal{L}^T & K_{X^*,X} \\ \mathcal{L}K_{X^v,X^*} & \mathcal{L}K_{X^v,X^v}\mathcal{L}^T+\sigma_v^2 I_{N_v} & \mathcal{L}K_{X^v,X} \\ K_{X,X^*} & K_{X,X^v}\mathcal{L}^T & K_{X,X}+\sigma^2 I_N \end{bmatrix}. \tag{18}$$

By first conditioning on Y we obtain

$$\begin{matrix}\mathbf{f}^* \\ \widetilde{C}\end{matrix}\,\Bigg|\,Y\sim\mathcal{N}\left(\begin{bmatrix}\mu^*+A_2(Y-\mu)\\ \mathcal{L}\mu^v+A_1(Y-\mu)\end{bmatrix},\begin{bmatrix}B_2 & B_3\\ B_3^T & B_1\end{bmatrix}\right), \tag{19}$$

for $A_1=(\mathcal{L}K_{X^v,X})(K_{X,X}+\sigma^2 I_N)^{-1}$, $A_2=K_{X^*,X}(K_{X,X}+\sigma^2 I_N)^{-1}$, $B_1=\mathcal{L}K_{X^v,X^v}\mathcal{L}^T+\sigma_v^2 I_{N_v}-A_1K_{X,X^v}\mathcal{L}^T$, $B_2=K_{X^*,X^*}-A_2K_{X,X^*}$, and $B_3=K_{X^*,X^v}\mathcal{L}^T-A_2K_{X,X^v}\mathcal{L}^T$.

Conditioning on $\widetilde{C}$ then gives

$$\mathbf{f}^*|Y,\widetilde{C}\sim\mathcal{N}\left(\mu^*+A(\widetilde{C}-\mathcal{L}\mu^v)+B(Y-\mu),\Sigma\right), \tag{20}$$

for $A=B_3B_1^{-1}$, $B=A_2-AA_1$ and $\Sigma=B_2-AB_3^T$.

Similarly, we may derive $\widetilde{C}|Y$ by observing that the joint distribution of $\widetilde{C},Y$ is given by removing the first row in (17) and the first row and column in (18). Hence,

$$\widetilde{C}|Y\sim\mathcal{N}\left(\mathcal{L}\mu^v+A_1(Y-\mu),B_1\right). \tag{21}$$

The constrained posterior of $\widetilde{C}$ is obtained by applying the constraint C to the posterior, and hence $\widetilde{C}|Y,C$ becomes a truncated Gaussian with the same mean and variance as in (21), and the bounds $a(X^v)$ and $b(X^v)$ given by C. Similarly, $\mathbf{f}^*|Y,C$ is obtained by replacing $\widetilde{C}$ in (20) with $\widetilde{C}|Y,C$. Finally, the probability $p(C|Y)$ is just the probability that $\widetilde{C}|Y$ given in (21) falls within the bounds given by C, and the unconstrained distribution remains the same as (2). ■

Appendix B. Proof of Lemma 2

Proof. The equations in Lemma 2 can be verified by simply inserting L, v_1 and v_2 and check against the expressions in Lemma 1. We show this for A_1 and B_1, and the results for the remaining matrices are proved by applying the same procedures. In order to factorize B_1, we use that B_1 is the covariance matrix of a Gaussian random variable (see Equation 21 in Appendix A), and must therefore be symmetric and positive definite.

To show that $A_1 = (L^T \setminus v_1)^T$ we use that $v_1 = L \setminus K_{X,X^v}\mathcal{L}^T \Rightarrow Lv_1 = K_{X,X^v}\mathcal{L}^T$. Hence,

$$\begin{aligned}
&A_1 = (L^T \setminus v_1)^T \\
&\Rightarrow L^T A_1^T = v_1 = L \setminus K_{X,X^v}\mathcal{L}^T \\
&\Rightarrow LL^T A_1^T = K_{X,X^v}\mathcal{L}^T \\
&\Rightarrow A_1 = ((LL^T)^{-1}K_{X,X^v}\mathcal{L}^T)^T = (\mathcal{L}K_{X^v,X})(K_{X,X} + \sigma^2 I_N)^{-1},
\end{aligned}$$

where we have used that $(K_{X,X^v}\mathcal{L}^T)^T = \mathcal{L}K_{X^v,X}$ and $LL^T = K_{X,X} + \sigma^2 I_N$.

To show that $B_1 = \mathcal{L}K_{X^v,X^v}\mathcal{L}^T + \sigma_v^2 I_{N_v} - v_1^T v_1$ we need to show that $v_1^T v_1 = A_1 K_{X,X^v}\mathcal{L}^T$, which is trivial

$$\begin{aligned}
v_1^T v_1 &= (L^{-1}K_{X,X^v}\mathcal{L}^T)^T(L^{-1}K_{X,X^v}\mathcal{L}^T) \\
&= \mathcal{L}K_{X^v,X}(LL^T)^{-1}K_{X,X^v}\mathcal{L}^T \\
&= A_1 K_{X,X^v}\mathcal{L}^T.
\end{aligned}$$

■

Appendix C. Algorithm for Finding Virtual Observation Locations based on Individual Sub-operators

We present the details of the algorithm for finding virtual observation locations introduced in Section 3.5. Here we let $\mathcal{L}$ be a linear operator defined by the column vector $[\mathcal{F}_1, \ldots, \mathcal{F}_k]$, where $\mathcal{F}_i$ produces functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}$, subjected to an interval constraint $[a_i(\mathbf{x}), b_i(\mathbf{x})]$. We would like to impose constraints related to the i-th sub-operator only at locations where $p(\mathcal{F}_i f(\mathbf{x}) \notin [a_i(\mathbf{x}), b_i(\mathbf{x})])$ is not sufficiently small. For this we let X^v be the concatenation of the matrices $X^{v,1}, \ldots, X^{v,k}$ and define $\mathcal{L}^T f(X^v) = [\mathcal{F}_1^T f(X^{v,1}), \ldots, \mathcal{F}_k^T f(X^{v,1})]^T$. The matrices needed to make use of Lemma 1 and Lemma 2 are $\mathcal{L}\mu^v$, $K_{X,X^v}\mathcal{L}^T$, $K_{X^*,X^v}\mathcal{L}^T$, and $\mathcal{L}K_{X^v,X^v}\mathcal{L}^T$. Using that $\mathcal{F}_i f(X^v) = \mathcal{F}_i f(X^{v,i})$, these are given by

$$\mathcal{L}\mu^v = \begin{bmatrix} \mathcal{F}_1\mu(X^{v,1}) \\ \vdots \\ \mathcal{F}_k\mu(X^{v,k}) \end{bmatrix}, \quad K_{X,X^v}\mathcal{L}^T = \begin{bmatrix} K_{X,X^{v,1}}\mathcal{F}_1^T \\ \vdots \\ K_{X,X^{v,k}}\mathcal{F}_k^T \end{bmatrix},$$

where $K_{X^*,X^v}\mathcal{L}^T$ also is given by the above equation for $X = X^*$. Finally, $\mathcal{L}K_{X^v,X^v}\mathcal{L}^T$ is the block matrix with blocks

$$(\mathcal{L}K_{X^v,X^v}\mathcal{L}^T)_{i,j} = \mathcal{F}_i K_{X^{v,i},X^{v,j}}\mathcal{F}_j^T.$$

We want to improve the algorithm in Section 3.4 for finding the set of virtual observation locations by considering each sub-operator individually. To do this we make use estimated partial constraint probabilities (given in (13) and restated below).

$$\hat{p}_{c,i}(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m} P\left(a_i(\mathbf{x}) - \nu < (\mathcal{L}f(\mathbf{x})|Y, C_j)_i < b_i(\mathbf{x}) + \nu\right),$$

where $(\mathcal{L}f(\mathbf{x})|Y, C_j)_i$ is the univariate Normal distribution given by the i-th row of $(\mathcal{L}f(\mathbf{x})|Y, C_j)$ and $C_1, \ldots, C_m$ are m samples of $\mathbf{C}$ given in (6) as before. For the individual sub-operators $\mathcal{F}_i$, the set of virtual observations X_i^v needed to ensure that $\hat{p}_{c,i}(\mathbf{x}) \geq p_{\text{target}}$ can then be found using the following algorithm.

Algorithm 7 *Finding locations of virtual observations X_i^v s.t. $\hat{p}_{c,i}(\boldsymbol{x}) \geq p_{target}$ for all $\boldsymbol{x} \in \Omega$ and all sub-operators $\mathcal{F}_1, \ldots, \mathcal{F}_k$.*

1. *Compute $L = Chol(K_{X,X} + \sigma^2 I_N)$.*
2. *Until convergence do:*
   - *(a) If $X^v \neq \emptyset$ compute A_1 and B_1 as defined in Lemma 2, and generate m samples $C_1, \ldots, C_m$ of $\boldsymbol{C}$ given in (6).*
   - *(b) If $X^v = \emptyset$ compute $(\boldsymbol{x}_i^*, p_i^*) = (\arg\min p_{c,i}(\boldsymbol{x}), p_{c,i}(\boldsymbol{x}^*))$. Otherwise compute $(\boldsymbol{x}_i^*, p_i^*) = (\arg\min \hat{p}_{c,i}(\boldsymbol{x}), \hat{p}_{c,i}(\boldsymbol{x}^*))$, for all $i = 1, \ldots, k$ with $\hat{p}_{c,i}$ defined as in (13) using the samples generated in step (a).*
   - *(c) Let $(\boldsymbol{x}^*, p^*, j)$ correspond to the smallest probability: $p^* = p_j^* = min_i\ p_i^*$.*
   - *(d) Terminate if $p^* \geq p_{target}$, otherwise update $X_j^v \to X_j^v \cup \{\boldsymbol{x}^*\}$.*

Appendix D. Proof of Lemma 4

Proof. This follows exactly from the proofs of Lemma 1 and Lemma 2 by replacing $\mathbf{f}^* \to \mathcal{L}f(\mathbf{x}^*)$, which implies $\mu^* \to \mathcal{L}\mu^*$, $K_{X^*,X} \to \mathcal{L}K_{\mathbf{x}^*,X}$, $K_{X^*,X^*} \to \mathcal{L}K_{\mathbf{x}^*,\mathbf{x}^*}\mathcal{L}^T$ and $K_{X^*,X^v}\mathcal{L}^T \to \mathcal{L}K_{\mathbf{x}^*,X^v}\mathcal{L}^T$.

■

Appendix E. Proof of Corollary 6

Proof. We show the derivation of the expectation and covariance of $\mathbf{f}^*|Y, C$ as the derivations for $\mathcal{L}f(\mathbf{x}^*)|Y, C$ are equivalent. From Lemma 1 we have that

$$\mathbf{f}^*|Y, C \sim \mathcal{N}(\mu^* + A(\mathbf{C} - \mathcal{L}\mu^v) + B(Y - \mu), \Sigma).$$

If we let ν, Γ be the expectation and covariance of $\mathbf{C}$, then

$$\begin{aligned}
\mathbb{E}[\mathbf{f}^*|Y, C] = \mathbb{E}_{\mathbf{C}}\left[\mathbb{E}[\mathbf{f}^*|Y, \mathbf{C}]\right] &= \mathbb{E}_{\mathbf{C}}\left[\mu^* + A(\mathbf{C} - \mathcal{L}\mu^v) + B(Y - \mu)\right] \\
&= \mu^* + A(\nu - \mathcal{L}\mu^v) + B(Y - \mu),
\end{aligned}$$

and

$$\begin{aligned}\mathrm{cov}[\mathbf{f}^*|Y,C] &= \mathbb{E}_{\mathbf{C}}\left[\mathrm{cov}[\mathbf{f}^*|Y,\mathbf{C}]\right] + \mathrm{cov}_{\mathbf{C}}[\mathbb{E}[\mathbf{f}^*|Y,\mathbf{C}]] \\ &= \mathbb{E}_{\mathbf{C}}[\Sigma] + \mathrm{cov}_{\mathbf{C}}[\mu^* + A(\mathbf{C} - \mathcal{L}\mu^v) + B(Y-\mu)] \\ &= \Sigma + \mathrm{cov}_{\mathbf{C}}[A\mathbf{C}] = \Sigma + A\Gamma A^T.\end{aligned}$$

■

References

Petter Abrahamsen and Fred Espen Benth. Kriging with inequality constraints. *Mathematical Geology*, 33(6):719–744, Aug 2001.

Robert J. Adler. *The Geometry of Random Fields*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. J. Wiley, 1981.

Christian Agrell, Simen Eldevik, Andreas Hafver, Frank Børre Pedersen, Erik Stensrud, and Arne Huseby. Pitfalls of machine learning for tail events in high risk environments. In Stein Haugen, Anne Barros, Coen van Gulijk, Trond Kongsvik, and Jan Erik Vinnem, editors, *Safety and Reliability Safe Societies in a Changing World - Proceedings of ESREL 2018*. CRC Press, june 2018.

Rafael Amaya, Mauricio Sanchez-Silva, Emilio Bastidas-Arteaga, Franck Schoefs, and Felipe Munoz. Reliability assessments of corroded pipelines based on internal pressure - A review. *Engineering Failure Analysis*, 98, 01 2019.

Jian An and Art Owen. Quasi-regression. *Journal of Complexity*, 17(4):588 – 607, 2001.

Ioannis Andrianakis and Peter G. Challenor. The effect of the nugget on gaussian process emulators of computer models. *Computational Statistics & Data Analysis*, 56(12):4215 – 4228, 2012.

François Bachoc. Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55 – 69, 2013.

François Bachoc, Agnes Lagnoux, and Andrés F. López-Lopera. Maximum likelihood estimation for gaussian processes under inequality constraints. working paper or preprint, August 2018.

Zdravko I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148, 2017.

Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. *Annales de la faculté des sciences de Toulouse Mathématiques*, 21(3):529–555, 4 2012.

Sébastien Da Veiga and Amandine Marrel. Gaussian process regression with linear inequality constraints. working paper or preprint, 10 2015.

Alexander G. de G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 231–239. PMLR, 09–11 May 2016.

DNV GL. Reccomended Practice: Corroded pipelines DNVGL-RP-F101. *DNV GL, Høvik, Norway*, 2017.

Simen Eldevik, Christian Agrell, Andreas Hafver, and Frank B. Pedersen. AI + Safety: Safety implications for artificial intelligence and why we need to combine casual- and data-driven models. 08 2018. [Online position paper by DNV GL Group Technology and Research; https://ai-and-safety.dnvgl.com/, posted 28-August-2018].

Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149, 1992.

Alan Genz. Comparison of methods for the computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 11, 04 1997.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521 (7553):452–459, 2015.

Shirin Golchi, D R. Bingham, H Chipman, and David Campbell. Monotone emulation of computer experiments. *SIAM/ASA Journal on Uncertainty Quantification*, 3:370–392, 01 2015.

Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B Schön. Linearly constrained gaussian processes. pages 1215–1224. Curran Associates, Inc., 2017.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

George S. Kimeldorf and Grace Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 41(2):495–502, 04 1970.

Jack P. C. Kleijnen and Wim C. M. Van Beers. Monotonicity-preserving bootstrapped kriging metamodels for expensive simulations. *JORS*, 64:708–717, 2013.

Jayesh H. Kotecha and Petar M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 3, pages 1757–1760 vol.3, March 1999.

Peter Lenk and Taeryon Choi. Bayesian analysis of shape-restricted functions using gaussian process priors. *Statistica Sinica*, 27:43–69, 2017.

Lizhen Lin and David B. Dunson. Bayesian monotone regression using gaussian process projection. *Biometrika*, 101(2):303–317, 2014.

Andrés López-Lopera, François Bachoc, Nicolas Durrande, and Olivier Roustant. Finite-dimensional gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255, 2018.

Hassan Maatouk and Xavier Bay. Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582, Jul 2017.

Hassan Maatouk, Laurence Grammont, and Xavier Bay. Generalization of the kimeldorf-wahba correspondence for constrained interpolation. *Electronic Journal of Statistics*, 10 (1):1580–1595, 2016.

Henrik O. Madsen, Steen Krenk, and Niels C. Lind. *Methods of Structural Safety*. Dover Civil and Mechanical Engineering Series. Dover Publications, 2006.

Georges Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, 1973.

Anna Michalak. A gibbs sampler for inequality-constrained geostatistical interpolation and inverse modeling. *Water Resour. Res*, 44, 09 2008.

Ari Pakman and Liam Paninski. Exact hamiltonian monte carlo for truncated multivariate gaussians. *Journal of Computational and Graphical Statistics*, 23, 08 2012.

Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Higher Education, 4 edition, 2002.

Pritam Ranjan, Ronald Haynes, and Richard Karsten. A computationally stable approach to gaussian process interpolation of deterministic computer simulation data. *Technometrics*, 53, 03 2010.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Jaakko Riihimki and Aki Vehtari. Gaussian processes with monotonicity information. *Journal of Machine Learning Research - Proceedings Track*, 9:645–652, 01 2010.

Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statist. Sci.*, 4(4):409–423, 11 1989.

Simo Särkkä. Linear operators and stochastic partial differential equations in gaussian process regression. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 151–158, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Gudfinnur Sigurdsson, Espen H. Cramer, Ola H. Bjørnøy, B. Fu, and D. Ritchie. Background to DNV RP-F101 Corroded pipelines. In *Proceedings of the* 18th *international conference on offshore mechanics and arctic engineering, OMAE, Newfoundland, Canada*. American Society of Mechanical Engineers, U.S., 1999.

Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4): 341–359, Dec 1997.

Philip Duncan Thompson. Optimum smoothing of two-dimensional fields. *Tellus*, 8(3): 384–393, 1956.

Xiaojing Wang and James O. Berger. Estimating shape constrained functions using gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4:1–25, 01 2016.

Eun-Hye Yoo and Phaedon C. Kyriakidis. Area-to-point kriging with inequality-type data. *Journal of Geographical Systems*, 8(4):357–390, Oct 2006.

Journal of Machine Learning Research 20 (2019) 1-37 Submitted 4/19; Revised 8/19; Published 8/19

Local Regularization of Noisy Point Clouds: Improved Global Geometric Estimates and Data Analysis

Nicolás García Trillos GARCIATRILLO@WISC.EDU
Department of Statistics
University of Wisconsin-Madison
Madison, WI 53706, USA

Daniel Sanz-Alonso SANZALONSO@UCHICAGO.EDU
Ruiyi Yang YRY@UCHICAGO.EDU
Department of Statistics
University of Chicago
Chicago, IL 60637, USA

Editor: Sayan Mukherjee

Abstract

Several data analysis techniques employ similarity relationships between data points to uncover the intrinsic dimension and geometric structure of the underlying data-generating mechanism. In this paper we work under the model assumption that the data is made of random perturbations of feature vectors lying on a low-dimensional manifold. We study two questions: how to define the similarity relationships over noisy data points, and what is the resulting impact of the choice of similarity in the extraction of global geometric information from the underlying manifold. We provide concrete mathematical evidence that using a local regularization of the noisy data to define the similarity improves the approximation of the hidden Euclidean distance between unperturbed points. Furthermore, graph-based objects constructed with the locally regularized similarity function satisfy better error bounds in their recovery of global geometric ones. Our theory is supported by numerical experiments that demonstrate that the gain in geometric understanding facilitated by local regularization translates into a gain in classification accuracy in simulated and real data.

Keywords: manifold denoising, metric estimation, spectral convergence, graph Laplacian

1. Introduction

Several techniques for the analysis of high dimensional data build on the observation that data-generating mechanisms can often be described by few degrees of freedom. In this paper we study graph-based methods that employ similarity relationships between data points to uncover the low intrinsic dimension and geometric structure of datasets. Graph-based learning provides a well-balanced compromise between accuracy and interpretability Coifman and Lafon (2006), and is popular in a variety of unsupervised and semi-supervised tasks Zhu (2005); Von Luxburg (2007). These methods have been extensively analyzed in the idealized setting where the data is sampled from a low-dimensional manifold and

†. All authors contributed equally to this work.

©2019 Nicolás García Trillos, Daniel Sanz-Alonso, and Ruiyi Yang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-261.html.

similarities are computed using the ambient Euclidean distance or the geodesic distance, see e.g. Coifman and Lafon (2006); Singer (2006); Burago et al. (2014); García Trillos et al. (2018). The manifold setting is truthful in spirit to the presupposition that data arising from structured systems may be described by few degrees of freedom, but it is not so in that the data are typically noisy. The aim of this paper is to provide new mathematical theory under the more general and realistic model assumption that the data consist of random perturbations of low-dimensional features lying on a manifold.

By relaxing the manifold assumption we bring forward two fundamental questions that are at the heart of graph-based learning but have not been accounted for by previous theory. First, how to define the inter-point similarities between noisy data points in order to approximate the Euclidean distances between unperturbed data-points? Second, is it possible to recover global geometric features of the manifold from suitably-defined similarities between noisy data points? We will show by rigorous mathematical reasoning that:

(i) Denoising inter-point distances leads to an improved approximation of the hidden Euclidean distance between unperturbed points. We illustrate this general idea by analyzing a simple, easily-computable similarity defined in terms of a local-regularization of the noisy dataset.

(ii) Graph-based objects defined via locally regularized similarities can be guaranteed to satisfy improved error bounds in the recovery of global geometric properties. We illustrate this general idea by showing the spectral approximation of an unnormalized ε-graph Laplacian to a Laplace operator defined on the underlying manifold.

In addition to giving theoretical support for the denoising of point clouds, we study the practical use of local regularization in classification problems. Our analytically tractable local-regularization depends on a parameter that modulates the amount of localization, and our analysis suggests the appropriate scaling of said parameter with the level noise level. In our numerical experiments we show that in semi-supervised classification problems this parameter may be chosen by cross-validation, ultimately producing classification rules with improved accuracy. Finally, we propose two alternative denoising methods with similar empirical performance that are sometimes easier to implement. In short, the improved recovery of the geometric structure of the underlying point cloud facilitated by (local) regularization translates into improved graph-based data analysis, and the results seem to be robust to the choice of methodology.

1.1 Framework

We assume a data model

$$y_i = x_i + z_i, \tag{1}$$

where the *unobserved* points x_i are sampled from an unknown m-dimensional manifold $\mathcal{M}$, the vectors $z_i \in \mathbb{R}^d$ represent noise, and $\mathcal{Y}_n = \{y_1, \ldots, y_n\} \subseteq \mathbb{R}^d$ is the observed data. Further geometric and probabilistic structure will be imposed to prove our main results—see Assumptions 1 and 2 below. Our analysis is motivated by the case, often found in applications, where the number n of data points and the ambient space dimension d are large, but the underlying intrinsic dimension m is small or moderate. Thus, the data-generating mechanism is described (up to a noisy perturbation) by $m \ll d$ degrees of

freedom. We aim to uncover geometric properties of the underlying manifold $\mathcal{M}$ from the observed data $\mathcal{Y}_n$ by using *similarity* graphs. The set of vertices of these graphs will be identified with the set $[n] := \{1, \ldots, n\}$—so that the i-th node corresponds to the i-th data point—and the weight $W(i,j)$ between the i-th and j-th data-point will be defined in terms of a similarity function $\delta : [n] \times [n] \to [0, \infty)$.

The first question that we consider is how to choose the similarity function so that $\delta(i,j)$ approximates the hidden Euclidean distance $\delta_{\mathcal{X}_n}(i,j) := |x_i - x_j|$. Full knowledge of the Euclidean distance between the latent variables x_i would allow to recover, in the large n limit, global geometric features of the underlying manifold. This motivates the idea of denoising the observed point cloud $\mathcal{Y}_n$ to approximate the hidden similarity function $\delta_{\mathcal{X}_n}$. Here we will study a family of similarity functions based on the Euclidean distance between local averages of points in $\mathcal{Y}_n$, i.e. averages of the local measures. We define a denoised dataset $\bar{\mathcal{Y}}_n = \{\overline{y}_1, \ldots, \overline{y}_n\}$ by locally averaging the original dataset, and we then define an associated similarity function

$$\delta_{\bar{\mathcal{Y}}_n}(i,j) := |\overline{y}_i - \overline{y}_j|.$$

In its simplest form, $\overline{y}_i$ is defined by averaging all points in $\mathcal{Y}_n$ that are inside the ball of radius $r > 0$ centered around y_i, that is,

$$\overline{y}_i := \frac{1}{N_i} \sum_{j \in \mathcal{A}_i} y_j, \tag{2}$$

where N_i is the cardinality of $\mathcal{A}_i := \{j \in [n] : y_j \in B(y_i, r)\}$. As discussed in Subsection 1.3, this corresponds to one step of the mean-shift algorithm Fukunaga and Hostetler (1975). Note that $\bar{\mathcal{Y}}_n$ (and the associated similarity function $\delta_{\bar{\mathcal{Y}}_n}$) depends on r, but we do not include said dependence in our notation for simplicity. Other possible local and non-local averaging approaches may be considered. We will only analyze the choice made in (2) and we will explore other constructions numerically. Introducing the notation

$$\delta_{\mathcal{X}_n}(i,j) = |x_i - x_j|, \quad \delta_{\mathcal{Y}_n}(i,j) = |y_i - y_j|,$$

the first question that we study may be formalized as understanding when, and to what extent, the similarity function $\delta_{\bar{\mathcal{Y}}_n}$ is a better approximation than $\delta_{\mathcal{Y}_n}$ (the standard choice) to the hidden similarity function $\delta_{\mathcal{X}_n}$. An answer is given in Theorem 1 below.

The second question that we investigate is how an improvement in the approximation of the hidden similarity function affects the approximation of the Laplace Beltrami operator on the underlying manifold $\mathcal{M}$. Specifically, we study how the spectral convergence of graph-Laplacians constructed with noisy data may be improved by local regularization of the point cloud. For concreteness, our theoretical analysis is focused on ε-graphs and unnormalized graph-Laplacians, but we expect our results to generalize to other graphs and graph-Laplacians—evidence to support this claim will be given through numerical experiments. We now summarize the necessary background to formalize this question. For a given similarity $\delta : [n] \times [n] \to [0, \infty)$ and a parameter $\varepsilon > 0$, we define a weighted graph $\Gamma_{\delta,\varepsilon} = ([n], W)$ by setting the weight between the i-th and j-th node to be

$$W(i,j) := \frac{2(m+2)}{\alpha_m \varepsilon^{m+2} n} \mathbb{1}\{\delta(i,j) < \varepsilon\}, \tag{3}$$

where α_m is the volume of the m-dimensional Euclidean unit ball. Associated to the graph $\Gamma_{\delta,\varepsilon}$ we define the unnormalized graph Laplacian matrix

$$\Delta_{\delta,\varepsilon} := D - W \in \mathbb{R}^{n \times n}, \tag{4}$$

where D is a diagonal matrix with diagonal entries

$$D(i,i) := \sum_{j=1}^{n} W(i,j).$$

The motivation for the scaling in (3) is so that $\Delta_{\delta,\epsilon}$ matches the scale of the Laplace-Beltrami operator (see for example Burago et al. (2014)). For the rest of the paper we shall denote $\Gamma_{\mathcal{X}_n,\varepsilon} := \Gamma_{\delta_{\mathcal{X}_n},\varepsilon}$ and $\Delta_{\mathcal{X}_n,\varepsilon} := \Delta_{\delta_{\mathcal{X}_n},\varepsilon}$. We use analogous notation for $\mathcal{Y}_n$ and $\bar{\mathcal{Y}}_n$. The second question that we consider may be formalized as understanding when, and to what extent, $\Delta_{\bar{\mathcal{Y}}_n}$ provides a better approximation (in the spectral sense) than $\Delta_{\mathcal{Y}_n}$ to a Laplace operator on the manifold $\mathcal{M}$. An answer is given in Theorem 3 below.

1.2 Main results

In this subsection we state our main theoretical results. We first impose some geometric conditions on the underlying manifold $\mathcal{M}$.

Assumption 1 *$\mathcal{M}$ is a smooth, oriented, compact manifold with no boundary and intrinsic dimension m, embedded in $\mathbb{R}^d$. Moreover, $\mathcal{M}$ has injectivity radius $\geq i_0$, maximum of the absolute value of sectional curvature $\leq K$, and reach $\geq R$. Finally, we assume that $\mathcal{M}$'s total volume is normalized and equal to one.*

Loosely speaking, the injectivity radius determines the range of the exponential map (which will be an important tool in our analysis and will be reviewed in the next section) and the sectional curvature controls the metric distortion induced by the exponential map, and thereby its Jacobian. The reach R can be thought of as an (inverse) conditioning number of the manifold and controls its second fundamental form; it can also be interpreted as a measure of extrinsic curvature—see, e.g. Aamari et al. (2019), Federer (1959) for technical background. The significance of these geometric quantities and their role in our analysis will be further discussed in Section 2.

Next we impose further probabilistic structure into the data model (1). We assume that the pairs (x_i, z_i) are i.i.d. samples of the random vector $(X, Z) \sim \boldsymbol{\mu} \in \mathcal{P}(\mathcal{M} \times \mathbb{R}^d)$. Let μ and μ_x be, respectively, the marginal distribution of X and the conditional distribution of Z given $X = x$. We assume that μ is absolutely continuous with respect to the Riemannian volume form of $\mathcal{M}$ with density $p(x)$, i.e.,

$$d\mu(x) = p(x) d\mathrm{vol}_{\mathcal{M}}(x). \tag{5}$$

Furthermore, we assume that μ_x is supported on $T_x\mathcal{M}^\perp$ (the orthogonal complement of the tangent space $T_x\mathcal{M}$) and that it is absolutely continuous with respect to the $(d-m)$-dimensional Hausdorff measure $\mathcal{H}^{d-m}$ restricted to $T_x\mathcal{M}^\perp$ with density $p(z|x)$, i.e.,

$$d\mu_x(z) = p(z|x) d\mathcal{H}^{d-m}(z).$$

To ease the notation we will write dz instead of $\mathcal{H}^{d-m}(dz)$. We make the following assumptions on these densities.

Assumption 2 *It holds that:*

(i) The density $p(x)$ is of class $C^2(\mathcal{M})$ and is bounded above and below by positive constants:

$$0 < p_{min} \leq p(x) \leq p_{max}, \quad \forall x \in \mathcal{M}.$$

(ii) For all $x \in \mathcal{M}$,

$$\int z p(z|x) dz = 0.$$

Moreover, there is $\sigma < R$ such that $p(z|x) = 0$ for all z with $|z| \geq \sigma$.

Note that the assumption on $p(z|x)$ ensures that the noise is centered and bounded by a constant σ. While the assumption that the noise is bounded and orthogonal to the manifold can be relaxed, we choose not to do so here to streamline our results and proofs.

In our first main theorem we study the approximation of the similarity function $\delta_{\mathcal{X}_n}$ by $\delta_{\bar{\mathcal{Y}}_n}$. We consider points x_i and x_j that are close with respect to the geodesic distance $d_{\mathcal{M}}$ on the manifold, and show that local regularization improves the approximation of the hidden similarity provided that n is large and the noise level σ is small. The local regularity parameter r needs to be suitably scaled with σ. We make the following standing assumption linking both parameters; we refer to Remark 2 below for a discussion on the optimal scaling of r with σ, and to our numerical experiments for practical guidelines.

Assumption 3 *The localization parameter r and the noise level σ satisfy*

$$\sigma \leq \frac{R}{16m}, \; r \leq \min\left\{ i_0, \frac{1}{\sqrt{K}}, \sqrt{\frac{\alpha_m}{2CmK}}, \sqrt{\frac{R}{32}} \right\}, \text{ and } \sigma \leq \frac{1}{3} r, \tag{6}$$

where C is a universal constant, α_m denotes the volume of the Euclidean unit ball in $\mathbb{R}^m$, and i_0, R, and K are as in Assumption 1.

In words, Assumption 3 requires both r and σ to be sufficiently small, and r to be larger than σ.

Now we are ready to state the first main result.

Theorem 1 *Under Assumptions 1, 2 and 3, with probability at least $1 - 4ne^{-cnr^{\max\{2m,m+4\}}}$, for all x_i and x_j with $d_{\mathcal{M}}(x_i, x_j) \leq r$ we have*

$$\left|\delta_{\mathcal{X}_n}(i,j) - \delta_{\bar{\mathcal{Y}}_n}(i,j)\right| \leq C_{\mathcal{M}} \left(r^3 + r\sigma + \frac{\sigma^2}{r} \right), \tag{7}$$

where $c = \min\left\{ \frac{\alpha_m^2 p_{min}^2}{4^{m+2}}, \frac{1}{16} \right\}$ and $C_{\mathcal{M}}$ is a constant depending on m, K, R, a uniform bound on the change in second fundamental form of $\mathcal{M}$, and on the regularity of the density p.

Remark 2 *Theorem 1 gives concrete evidence of the importance of the choice of similarity function. For the usual Euclidean distance between observed data, $\delta_{\mathcal{Y}_n}$, one can only guarantee that*

$$\left|\delta_{\mathcal{X}_n}(i,j) - \delta_{\mathcal{Y}_n}(i,j)\right| \leq 2\sigma,$$

which follows from

$$\big||x_i - x_j| - |y_i - y_j|\big| \leq |z_i - z_j| \leq 2\sigma.$$

However, if we choose $r \propto \sigma^{1/2}$, then the error in (7) is of order $\sigma^{3/2}$, which is a considerably smaller quantity in the small noise limit.

Our second main result translates the local similarity bound from Theorem 1 into a global geometric result concerning the spectral convergence of the graph Laplacian to the Laplace operator formally defined by

$$\Delta_{\mathcal{M}} f = -\frac{1}{p} \mathrm{div}\big(p^2 \nabla f\big), \tag{8}$$

where div and ∇ denote the divergence and gradient operators on the manifold and p is the sampling density of the hidden point cloud $\mathcal{X}_n$, as introduced in Equation (5). It is intuitively clear that the spectral approximation of the discrete graph-Laplacian to the continuum operator $\Delta_{\mathcal{M}}$ necessarily rests upon having a sufficient number of samples from μ (defined in (5)). In other words, the empirical measure $\mu_n = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ needs to be close to μ, the sampling density of the hidden dataset. We characterize the closeness between μ_n and μ by the ∞-OT transport distance, defined as

$$d_\infty(\mu_n, \mu) := \min_{T: T_\sharp \mu = \mu_n} \operatorname*{esssup}_{x \in \mathcal{M}} \; d_{\mathcal{M}}\big(x, T(x)\big),$$

where $T_\sharp \mu$ denotes the push-forward of μ by T, that is, $T_\sharp \mu = \mu\big(T^{-1}(U)\big)$ for any Borel subset U of $\mathcal{M}$. Theorem 2 in García Trillos et al. (2018) shows that for every $\beta > 1$, with probability at least $1 - C_{\beta,\mathcal{M}} n^{-\beta}$,

$$d_\infty(\mu_n, \mu) \leq C_{\mathcal{M}} \frac{\log(n)^{p_m}}{n^{1/m}},$$

where $p_m = 3/4$ if $m = 2$ and $p_m = 1/m$ for $m \geq 3$. This is the high probability scaling of $d_\infty(\mu_n, \mu)$ in terms of n.

We introduce some notation before stating our second main result. Let $\lambda_\ell(\Gamma_{\delta,\varepsilon})$ be the ℓ-th smallest eigenvalue of the unnormalized graph-Laplacian $\Delta_{\delta,\varepsilon}$ defined in Equation (4), and let $\lambda_\ell(\mathcal{M})$ be the ℓ-th smallest eigenvalue of the continuum Laplace operator defined in Equation (8).

Theorem 3 *Suppose that Assumptions 1, 2, and 3 hold. Suppose further that ε is small enough (but not too small) so that*

$$\begin{gathered} \max\left\{ (m+5) d_\infty(\mu_n, \mu), 2Cm\eta \right\} < \varepsilon < \min\left\{ 1, \frac{i_0}{10}, \frac{1}{\sqrt{mK}}, \frac{R}{\sqrt{27m}} \right\}, \\ \big(\sqrt{\lambda_\ell(\mathcal{M})} + 1\big)\varepsilon + \frac{d_\infty(\mu_n, \mu)}{\varepsilon} < \tilde{c}_p, \end{gathered} \tag{9}$$

where $\tilde{c}_p$ is a constant that only depends on m and the regularity of the density p, C is a universal constant, and

$$\eta = C_{\mathcal{M}}\left(r^3 + r\sigma + \frac{\sigma^2}{r}\right)$$

is the bound in (7)*. Then, with probability at least* $1 - 4ne^{-cnr^{\max\{2m,m+4\}}}$*, for all* $\ell = 1, 2, 3, \ldots,$

$$\frac{|\lambda_\ell(\Gamma_{\bar{\mathcal{Y}}_n,\varepsilon}) - \lambda_\ell(\mathcal{M})|}{\lambda_\ell(\mathcal{M})} \leq \tilde{C}\left(\frac{\eta}{\varepsilon} + \frac{d_\infty(\mu_n,\mu)}{\varepsilon} + (1+\sqrt{\lambda_\ell(\mathcal{M})})\varepsilon + \left(K + \frac{1}{R^2}\right)\varepsilon^2\right),$$

where $\tilde{C}$ only depends on m and the regularity of p, and $c = \min\left\{\frac{\alpha_m^2 p_{min}^2}{4^{m+2}}, \frac{1}{16}\right\}$.

Remark 4 *We will see in Section 3 that Theorem 3 follows by plugging the probabilitstic estimate* (7) *into a modification of a deterministic result from (García Trillos et al., 2018, Corollary 2), which we present for the convenience of the reader in Theorem 20. We remark that any improvement of Theorem 20 would immediately translate into an improvement of our Theorem 3. As discussed in Remark 2, local regularization enables a smaller η than if no regularization is performed. This in turn allows one to choose, for a given error tolerance, a smaller connectivity ε, leading to a sparser graph that is computationally more efficient. Note also that the bound in Theorem 3 does not depend on the ambient space dimension d, but only on the intrinsic dimension m of the data.*

Remark 5 *Theorem 3 concretely shows how an improvement in metric approximation translates into an improved estimation of global geometric quantities. We have restricted our attention to analyzing eigenvalues of a Laplacian operator, but we remark that the idea goes beyond this particular choice. For example, one can conduct an asymptotic analysis illustrating the effect of changing the similarity function in the approximation of other geometric quantities of interest like Cheeger cuts. Such analysis could be carried out using the variational convergence approach from García Trillos and Slepčev (2016). Finally, we remark that it is possible to study convergence of* eigenvectors *of graph Laplacians following the results in García Trillos et al. (2018).*

1.3 Related and Future Work

Graph-based learning algorithms include spectral clustering, total variation clustering, graph-Laplacian regularization for semi-supervised learning, graph based Bayesian semi-supervised learning. A brief and incomplete summary of methodological and review papers is Shi and Malik (2000); Ng et al. (2002); Belkin and Niyogi (2004); Zhou and Schölkopf (2005); Spielman and Teng (2007); Von Luxburg (2007); Zhu (2005); Bertozzi et al. (2018). These algorithms involve either a graph Laplacian, the graph total variation, or Sobolev norms involving the graph structure. The large sample $n \to \infty$ theory studying the behavior of some of the above methodologies has been analyzed without reference to the intrinsic dimension of the data Von Luxburg et al. (2008) and in the case of points laying *on* a low dimensional manifold, see e.g. Belkin et al. (2006); Garcia Trillos and Sanz-Alonso (2018); Garcia Trillos et al. (2017) and references therein. Some papers that account for *both* the noisy and low

intrinsic dimensional structure of data are Niyogi et al. (2008); Little and Maggioni (2017); Agapiou et al. (2017); Weed and Bach (2017); Genovese et al. (2012); Aamari and Levrard (2019). For example, Niyogi et al. (2008) studies the recovery of the homology groups of submanifolds from noisy samples. We use the techniques for the analysis of spectral convergence of graph-Laplacians introduced in Burago et al. (2014) and further developed in García Trillos et al. (2018). The results in the latter reference would allow to extend our analysis to other graph Laplacians, but we do not pursue this here for conciseness.

We highlight that the denoising by local regularization occurs at the level of the dataset. That is, rather than denoising each of the observed features individually, we analyze denoising by averaging different data points. In practice combining both forms of denoising may be advantageous. For instance, when each of the data points corresponds to an image, one can first denoise each image at the pixel level and then do regularization at the level of the dataset as proposed here. In this regard, our regularization at the level of the data-set is similar to applying a filter at the level of individual pixels Tukey and Tukey (1988). The success of non-local filter image denoising algorithms suggests that non-local methods may be also of interest at the level of the dataset, but we expect this to be application-dependent. Finally, while in this paper we only consider first-order regularization based on averages, a topic for further research is the analysis of local PCA regularization Little and Maggioni (2017), incorporating covariance information.

With the same motivation for our work, in Mémoli et al. (2018) a general construction of metrics on noisy datasets was proposed. The so called *Wasserstein transform* associates to each of the data points a "local" probability distribution, and defines a new metric on the data by computing the Wasserstein distance between the corresponding local measures. A particular construction of local measures closely related to the metric we study here assigns to each observation the empirical measure of the observations restricted to a ball of certain radius around the given data point. The authors of Mémoli et al. (2018) propose the Wasserstein transform as a way to generalize the *mean-shift algorithm* and they study how it alleviates the so called *chaining effect* in single linkage clustering. The aim of our work is to provide quantitative evidence of the effect that changing the metric on noisy datasets has on graph-based spectral clustering algorithms. The success of these algorithms hinges on their ability to capture the geometry of the underlying data generating model.

It is worth noting the parallel between the local regularization that we study here and mean-shift and mode seeking methods Chen et al. (2016); Fukunaga and Hostetler (1975). As a matter of fact the points $\overline{x}_i$ that we construct here correspond to one step in the standard mean shift algorithm. However, we notice that our goal is not to run mean shift for mode seeking, but rather, as a way to construct a metric that better captures the underlying "true" geometric structure of the data that was blurred by noise. This paralellism with mean-shift techniques (or the more general Wasserstein transform in Mémoli et al. (2018)) suggests the idea of doing local averaging iteratively. Of course, it is important to notice that unless one prevents points to move tangentially to $\mathcal{M}$ (as discussed in Wang and Carreira-Perpinán (2010)), a large number of iterations would result in points collapsing to a finite number of local modes.

Local regularization may be also interpreted as a form of dictionary learning, where each data-point is represented in terms of its neighbors. For specific applications it may be of interest to restrict (or extend) the dictionary used to represent each data point Haddad et al.

(2014). Finally we refer to Hein and Maier (2007) for alternative techniques on manifold denoising.

1.4 Outline

The paper is organized as follows. In Section 2 we formalize the geometric setup and prove Theorem 1. Section 3 contains the proof of Theorem 3 and a lemma that may be of independent interest. Finally, Section 4 includes several numerical experiments. In the Appendix we prove a technical lemma that serves as a key ingredient in proving Theorem 1.

2. Distance Approximation

In this section we prove Theorem 1. We start with Subsection 2.1 by giving some intuition on the geometric conditions imposed in Assumption 1 and introducing the main geometric tools in our analysis. In Subsection 2.2 we decompose the approximation error between the similarity functions $\delta_{\bar{\mathcal{Y}}_n}$ and $\delta_{\mathcal{X}_n}$ into three terms, which are bounded in Subsections 2.3, 2.4, and 2.5.

2.1 Geometric Preliminaries

In this subsection we set our notation and provide some background on geometric concepts used in the remainder of this paper.

2.1.1 Basic Notation

For each $x \in \mathcal{M}$ we let $T_x\mathcal{M}$ be the tangent plane of $\mathcal{M}$ at x centered at the origin. In particular, $T_x\mathcal{M}$ is a m-dimensional subspace of $\mathbb{R}^d$, and we denote by $T_x\mathcal{M}^\perp$ its orthogonal complement. We will use $\mathrm{vol}_\mathcal{M}$ to denote the Riemannian volume form of $\mathcal{M}$. We will denote by $|x-\tilde{x}|$ the Euclidean distance between arbitrary points in $\mathbb{R}^d$ and denote by $d_\mathcal{M}(x,\tilde{x})$ the geodesic distance between points in $\mathcal{M}$. We denote by B_x balls in $T_x\mathcal{M}$ and by $B_\mathcal{M}$ balls in the manifold $\mathcal{M}$ (with respect to the geodesic distance). Also, unless otherwise specified B, without subscripts will be used to denote balls in $\mathbb{R}^d$. We denote by α_m the volume of the unit Euclidean ball in $\mathbb{R}^m$. Throughout the rest of the paper we use R, i_0 and K to denote the reach, injectivity radius, and maximum absolute curvature of $\mathcal{M}$, as in Assumption 1. We now describe at an intuitive level the role that these quantities play in our analysis.

2.1.2 The Reach

The reach of a closed submanifold $\mathcal{M}$ is the largest value $t \in [0,\infty]$ such that the projection map onto $\mathcal{M}$ is well defined on $\{x \in \mathbb{R}^d : \inf_{\tilde{x}\in\mathcal{M}}|\tilde{x}-x| < t\}$, i.e., every point in the tubular neighborhood around $\mathcal{M}$ of width t has a unique closest point in $\mathcal{M}$. Our assumption that the noise level satisfies $\sigma < R$ guarantees that x_i is the (well-defined) projection of y_i onto the manifold. The reach can be thought of as an inverse conditioning number for the manifold Niyogi et al. (2008). We will use that the inverse of the reach provides a uniform upper bound on the second fundamental form (see Lemma 12).

2.1.3 Exponential Map, Injectivity Radius and Sectional Curvature

We will make use of the exponential map exp, which for every $x \in \mathcal{M}$ is a map

$$\exp_x : B_x(0,i_0) \to B_\mathcal{M}(x,i_0)$$

where i_0 is the injectivity radius for the manifold $\mathcal{M}$. We recall that the exponential map $\exp_x$ takes a vector $v \in T_x\mathcal{M}$ and maps it to the point $\exp_x(v) \in \mathcal{M}$ that is at geodesic distance $|v|$ from x along the unit speed geodesic that at time $t=0$ passes through x with velocity $v/|v|$. The injectivity radius i_0 is precisely the maximum radius of a ball in $T_x\mathcal{M}$ centered at the origin for which the exponential map is a well defined diffeomorphism for every x. We denote by J_x the Jacobian of the exponential map $\exp_x$. Integrals with respect to $d\mathrm{vol}_\mathcal{M}$ can then be written in terms of integrals on $T_x\mathcal{M}$ weighted by the function J_x. More precisely, for an arbitrary test function $\varphi : \mathcal{M} \to \mathbb{R}$,

$$\int_{B_\mathcal{M}(x,i_0)} \varphi(\tilde{x}) d\mathrm{vol}_\mathcal{M}(\tilde{x}) = \int_{B_x(0,i_0)} \varphi\big(\exp_x(v)\big) J_x(v) dv.$$

For fixed $0 < r \le \min\{i_0, 1/\sqrt{K}\}$ one can obtain bounds on the metric distortion by the exponential map $\exp_x\colon B_x(0,r) \subseteq T_x\mathcal{M} \to \mathcal{M}$ ((do Carmo, 1992, Chapter 10) and (Burago et al., 2014, Section 2.2)), and thereby guarantee the existence of a universal constant C such that, for $|v| \le r$,

$$(1+CmK|v|^2)^{-1} \le J_x(v) \le (1+CmK|v|^2). \tag{10}$$

An immediate consequence of the previous inequalities is

$$|\mathrm{vol}(B_\mathcal{M}(x,r)) - \alpha_m r^m| \le CmKr^{m+2}, \tag{11}$$

where we recall α_m is the volume of the unit ball in $\mathbb{R}^m$. Equations (10) and (11) will be used in our geometric and probabilistic arguments and motivate our assumptions on the choice of local regularization parameter r in terms of the injectivity radius and the sectional curvature.

2.2 Local Distributions

Next we study the local behavior of (X,Z). To characterize its local distribution, it will be convenient to introduce the following family of probability measures.

Definition 6 *Let y be a vector in $\mathbb{R}^d$ whose distance to $\mathcal{M}$ is less than R. Let x be the projection of y onto $\mathcal{M}$. We say that the random variable $(\tilde{X},\tilde{Z})$ has the distribution $\boldsymbol{\mu}_y$ provided that*

$$\mathbb{P}\big((\tilde{X},\tilde{Z}) \in A_1 \times A_2\big) := \mathbb{P}\big((X,Z) \in A_1 \times A_2 | X+Z \in B(y,r)\big),$$

for all Borel sets $A_1 \subseteq \mathcal{M}$ $A_2 \subseteq \mathbb{R}^d$, where in the above (X,Z) is distributed according to $\boldsymbol{\mu}$.

In the remainder we use $\boldsymbol{\mu}_i$ as shorthand notation for $\boldsymbol{\mu}_{y_i}$. As for the original measure $\boldsymbol{\mu}$, we characterize $\boldsymbol{\mu}_i$ in terms of a marginal and conditional distribution. We introduce the density $\widetilde{p}_i : \mathcal{M} \to \mathbb{R}$ given by

$$\widetilde{p}_i(x) := \frac{\mathbb{P}_i\big(X + Z \in B(y_i, r) | X = x\big)}{\mathbb{P}_i\big(X + Z \in B(y_i, r)\big)} \cdot p(x), \tag{12}$$

and define

$$\widetilde{p}_i(z|x) = \frac{\mathbb{1}_{x+z \in B(y_i,r)}}{\mathbb{P}_i\big(X + Z \in B(y_i, r) | X = x\big)} \cdot p(z|x), \tag{13}$$

where in the above and in the remainder we use $\mathbb{E}_i$ and $\mathbb{P}_i$ to denote conditional expectation and conditional probability given (x_i, z_i). It can be easily shown that these functions correspond to the marginal density of $\tilde{X}_i$ and the conditional density of $\tilde{Z}_i$ given $\tilde{X}_i = x$, where $(\tilde{X}_i, \tilde{Z}_i) \sim \boldsymbol{\mu}_i$. The distribution $\boldsymbol{\mu}_i$ is of relevance because by definition of $\overline{y}_i$ one has

$$\mathbb{E}_i[\overline{y}_i] = \mathbb{E}_i[\tilde{X}_i + \tilde{Z}_i].$$

Now we are ready to introduce the main decomposition of the error between the similarity functions $\delta_{\widetilde{\mathcal{Y}}_n}$ and $\delta_{\mathcal{X}_n}$. Using the triangle inequality we can write

$$\big||x_i - x_j| - |\overline{y}_i - \overline{y}_j|\big| \leq \big|\mathbb{E}_i[\tilde{X}_i] - x_i - (\mathbb{E}_j[\tilde{X}_j] - x_j)\big| \tag{14}$$

$$+ \big|\mathbb{E}_j[\tilde{Z}_j]\big| + \big|\mathbb{E}_i[\tilde{Z}_i]\big| \tag{15}$$

$$+ \big|\mathbb{E}_i[\overline{y}_i] - \overline{y}_i\big| + \big|\mathbb{E}_j[\overline{y}_j] - \overline{y}_j\big|. \tag{16}$$

In the next subsections we bound each of the terms (15) (expected conditional noise), (14) (difference in *geometric* bias), and (16) (sampling error). As we will see in Subsection 2.5 we can control both terms in (16) with very high probability using standard concentration inequalities. The other three terms are deterministic quantities that can be written in terms of integrals with respect to the distributions $\tilde{\mu}_i$ and $\tilde{\mu}_j$. To study these integrals it will be convenient to introduce two quantities $r_- < r < r_+$ (independent of $i = 1, \ldots, n$) satisfying:

i) For all $x \in \mathcal{M}$ with $d_{\mathcal{M}}(x, x_i) > r_+$ we have

$$\mathbb{P}_i\big(X + Z \in B(y_i, r) | X = x\big) = 0.$$

Equivalently, the density $\tilde{p}_i(x)$ is supported in $\overline{B_{\mathcal{M}}(x_i, r_+)}$.

ii) For all x with $d_{\mathcal{M}}(x, x_i) < r_-$ we have

$$\mathbb{P}_i\big(X + Z \in B(y_i, r) | X = x\big) = 1.$$

It should be noted that the choice of both r_- and r_+ depends on r. In Appendix A we present the proof of the following lemma giving estimates for r_+ and r_-.

Lemma 7 (Bounds for r_+ and r_-) *Under Assumption 3, the quantities*

$$r_- := r\left(\sqrt{1 + \frac{4\sigma}{R} + \frac{16\sigma^2}{r^2}} + \frac{m\sigma}{R}\right)^{-1},$$

$$r_+ := r\left(\sqrt{1 - \frac{8r^2}{R} - \frac{4\sigma}{R}} - \frac{m\sigma}{R}\right)^{-1},$$

satisfy properties i) and ii). Furthermore,

$$r_+ - r_- \leq C_{m,R}\left(r^3 + r\sigma + \frac{\sigma^2}{r}\right), \; C_{m,R} := \max\left\{\frac{8m + 32}{R}, 64\right\}$$

and

$$\frac{1}{2} r_+ \leq r \leq 2r_-. \tag{17}$$

2.3 Bounding Expected Conditional Noise

Proposition 8 *Suppose that Assumptions 1 and 2 hold. Then,*

$$\big|\mathbb{E}_i[\tilde{Z}_i]\big| \leq C_{m,p} \frac{\sigma}{r}(r_+ - r_-), \qquad C_{m,p} := \frac{4^{m+1} p_{max}}{m p_{min}}.$$

Proof Using the definition of r_+,

$$\begin{aligned}
\mathbb{E}_i[\tilde{Z}_i] &= \int_{B_{\mathcal{M}}(x_i, r_+)} \int z \tilde{p}_i(z|x) dz \; \tilde{p}_i(x) \, \mathrm{dvol}_{\mathcal{M}}(x) \\
&= \int_{B_{\mathcal{M}}(x_i, r_-)} \int z \tilde{p}_i(z|x) dz \; \tilde{p}_i(x) \, \mathrm{dvol}_{\mathcal{M}}(x) \\
&+ \int_{B_{\mathcal{M}}(x_i, r_+) \setminus B_{\mathcal{M}}(x_i, r_-)} \int z \tilde{p}_i(z|x) dz \, \tilde{p}_i(x) \mathrm{dvol}_{\mathcal{M}}(x).
\end{aligned}$$

The first integral is the zero vector because for $x \in B_{\mathcal{M}}(x_i, r_-)$, we have $\tilde{p}(z|x) \propto p(z|x)$ and $p(z|x)$ is assumed to be centered. Therefore,

$$\begin{aligned}
\big|\mathbb{E}_i[\tilde{Z}_i]\big| &\leq \sigma \int_{B_{\mathcal{M}}(x_i, r_+) \setminus B_{\mathcal{M}}(x_i, r_-)} \tilde{p}_i(x) \mathrm{dvol}_{\mathcal{M}}(x) \\
&= \frac{\sigma}{\mathbb{P}_i\big(X + Z \in B(y_i, r)\big)} \int_{B_{\mathcal{M}}(x_i, r_+) \setminus B_{\mathcal{M}}(x_i, r_-)} p(x) \mathrm{dvol}_{\mathcal{M}}(x) \\
&\leq \frac{\sigma p_{max}}{\mathbb{P}_i\big(X + Z \in B(y_i, r)\big)} \int_{B_{\mathcal{M}}(x_i, r_+) \setminus B_{\mathcal{M}}(x_i, r_-)} \mathrm{dvol}_{\mathcal{M}}(x) \\
&\leq \frac{\sigma p_{max}}{\mathbb{P}_i\big(X + Z \in B(y_i, r)\big)} \int_{B_{x_i}(0, r_+) \setminus B_{x_i}(0, r_-)} J_{x_i}(v) dv \\
&\leq \frac{2\alpha_m \sigma p_{max}}{\mathbb{P}_i\big(X + Z \in B(y_i, r)\big)} (r_+^m - r_-^m) \\
&\leq \frac{2\alpha_m \sigma p_{max}}{m \mathbb{P}_i\big(X + Z \in B(y_i, r)\big)} (r_+ - r_-) r_+^{m-1},
\end{aligned}$$

where we have used (10) and the assumptions on r to say (in particular) that $J_{x_i}(v) \leq 2$, and also the fact that, for $t > s > 0$,

$$t^m - s^m = \int_s^t \frac{u^{m-1}}{m} du \leq (t - s) \frac{t^{m-1}}{m}.$$

Finally, notice that

$$\mathbb{P}_i\big(X+Z \in B(y_i,r)\big) \geq \mathbb{P}_i\big(X \in B_{\mathcal{M}}(x_i,r_-)\big) = \int_{B_{x_i}(0,r_-)} p\big(\exp_{x_i}(v)\big)J_{x_i}(v)dv \geq \frac{1}{2}p_{min}\alpha_m r_-^m,$$

where again we have used (10) to conclude (in particular) that $J_{x_i}(v) \geq 1/2$. The result now follows by (17).

■

2.4 Bounding Difference in Geometric Bias

In terms of r_+ and r_-, the difference $\mathbb{E}_i[\tilde{X}_i] - x_i$ (and likewise $\mathbb{E}_j[\tilde{X}_j] - x_j$) can be written as:

$$\begin{aligned}
\mathbb{E}_i[\tilde{X}_i] - x_i =& \int_{B_{\mathcal{M}}(x_i,r_+)} (x-x_i)\tilde{p}_i(x)d\mathrm{vol}_{\mathcal{M}}(x) \\
=& \int_{B_{x_i}(0,r_+)} \big(\exp_{x_i}(v) - x_i\big)\tilde{p}_i\big(\exp_x(v)\big)J_{x_i}(v)dv \\
=& \int_{B_{x_i}(0,r_+)} \big(\exp_{x_i}(v) - x_i\big)\tilde{p}_i\big(\exp_x(v)\big)dv + \\
& \int_{B_{x_i}(0,r_+)} \big(\exp_{x_i}(v) - x_i\big)\tilde{p}_i\big(\exp_x(v)\big)\big(J_{x_i}(v)-1\big)dv \\
=& \frac{1}{\mathbb{P}_i\big(X+Z\in B(y_i,r)\big)} \int_{B_{x_i}(0,r_-)} \big(\exp_{x_i}(v) - x_i\big)p\big(\exp_x(v)\big)dv + \\
& \int_{B_{x_i}(0,r_+)\setminus B_{x_i}(0,r_-)} \big(\exp_{x_i}(v) - x_i\big)\tilde{p}_i\big(\exp_x(v)\big)dv + \\
& \int_{B_{x_i}(0,r_+)} \big(\exp_{x_i}(v) - x_i\big)\tilde{p}_i\big(\exp_x(v)\big)\big(J_{x_i}(v)-1\big)dv \\
:=& \frac{1}{\mathbb{P}_i\big(X+Z\in B(y_i,r)\big)} \int_{B_{x_i}(0,r_-)} \big(\exp_{x_i}(v) - x_i\big)p\big(\exp_x(v)\big)dv + \xi_i,
\end{aligned}$$

where the second to last equality follows from (12). To further simplify the expression for $x_i - \mathbb{E}_i[\tilde{X}_i]$ let us define

$$b_i := \int_{B_{x_i}(0,r_-)} \big(\exp_{x_i}(v) - x_i\big)p\big(\exp_x(v)\big)dv.$$

It follows that

$$\begin{aligned}
\big|\mathbb{E}_i[\tilde{X}_i] - x_i - (\mathbb{E}_j[\tilde{X}_j] - x_j)\big| &\leq \Big|\frac{b_i}{P_i} - \frac{b_j}{P_j}\Big| + \big|\xi_i\big| + \big|\xi_j\big| \\
&\leq \Big|\frac{1}{P_i} - \frac{1}{P_j}\Big|\big|b_i\big| + \frac{1}{P_j}\big|b_i - b_j\big| + \big|\xi_i\big| + \big|\xi_j\big|,
\end{aligned} \tag{18}$$

where in the above

$$P_i := \mathbb{P}_i\big(X+Z\in B(y_i,r)\big), \quad P_j := \mathbb{P}_j\big(X+Z\in B(y_j,r)\big).$$

Lemma 9 *The following hold.*

i) *The terms P_i satisfy*

$$\frac{1}{2}p_{min}\alpha_m r_-^m \leq P_i.$$

ii) *The terms ξ_i satisfy:*

$$\big|\xi_i\big| \leq C_1(r_+ - r_-) + C_2 r^3,$$

where, up to universal multiplicative constants,

$$C_1 = \frac{4^{m+1}p_{max}}{mp_{min}}, \quad C_2 = 4^{m+3}mK\frac{p_{max}}{p_{min}}.$$

iii) *Suppose that $d_{\mathcal{M}}(x_i,x_j) \leq r$. Then,*

$$|P_i - P_j| \leq C_3 r^{m+1} + C_4(r_+ - r_-)r^{m-1} + C_5 r^{m+2},$$

where, up to universal multiplicative constants,

$$C_3 = C_p\alpha_m, \quad C_4 = \frac{2^{m-1}\alpha_m p_{max}}{m}, \quad C_5 = mKp_{max}\alpha_m.$$

and C_p only depends on bounds on the first derivatives of the density p.

Proof The first inequality was already obtained at the end of the proof of Proposition 8. For the second inequality recall that

$$\begin{aligned}
\xi_i =& \int_{B_{x_i}(0,r_+)\setminus B_{x_i}(0,r_-)} \big(x_i - \exp_{x_i}(v)\big)\tilde{p}_i\big(\exp_{x_i}(v)\big)dv + \\
& \int_{B_{x_i}(0,r_+)} \big(x_i - \exp_{x_i}(v)\big)\tilde{p}_i\big(\exp_{x_i}(v)\big)[J_{x_i}(v)-1]dv := I_1 + I_2.
\end{aligned}$$

For the first term we notice that $|x_i - \exp_{x_i}(v)| \leq d_{\mathcal{M}}(x_i,\exp_{x_i}(v)) \leq r_+$. Thus using i) and the definition of $\tilde{p}_i$ we have

$$|I_1| \leq \frac{r_+ p_{max}\alpha_m}{\mathbb{P}_i\big(X+Z\in B(y_i,r)\big)}(r_+^m - r_-^m) \leq \frac{4^{m+1}p_{max}}{mp_{min}}(r_+ - r_-).$$

For the second term we use i) and (10) to see that

$$|I_2| \leq \frac{CmKp_{max}\alpha_m}{\mathbb{P}_i\big(X+Z\in B(y_i,r)\big)}r_+^{m+3} \leq C4^{m+3}mK\frac{p_{max}}{p_{min}}r^3.$$

For iii) we notice that by definition of r_- and r_+ we can write

$$\mathbb{P}_i\big(X\in B_{x_i}(0,r_-)\big) - \mathbb{P}_j\big(X\in B_{x_j}(0,r_+)\big) \leq P_i - P_j \leq \mathbb{P}_i\big(X\in B_{x_i}(0,r_+)\big) - \mathbb{P}_j\big(X\in B_{x_j}(0,r_-)\big),$$

and in particular it is enough to bound $H_{ij} := \big|\mathbb{P}_i\big(X \in B_{\mathcal{M}}(x_i, r_+)\big) - \mathbb{P}_j\big(X \in B_{\mathcal{M}}(x_j, r_-)\big)\big|$. We can expand H_{ij} as follows.

$$\begin{aligned} H_{ij} &= \int_{B_{x_i}(0,r_-)} p\big(\exp_{x_i}(v)\big)dv - \int_{B_{x_j}(0,r_-)} p\big(\exp_{x_j}(\tilde{v})\big)d\tilde{v} \\ &+ \int_{B_{x_i}(0,r_+)\setminus B_{x_i}(0,r_-)} p\big(\exp_{x_i}(v)\big)dv \\ &+ \int_{B_{x_i}(0,r_-)} p\big(\exp_{x_i}(v)\big)\big(J_{x_i}(v)-1\big)dv - \int_{B_{x_j}(0,r_-)} p\big(\exp_{x_j}(\tilde{v})\big)\big(J_{x_j}(\tilde{v})-1\big)d\tilde{v} \\ &:= \mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_3. \end{aligned}$$

By a similar argument as above, we can bound $\mathcal{I}_2$ and $\mathcal{I}_3$ by

$$\begin{aligned} |\mathcal{I}_2| &\le p_{max}\alpha_m(r_+^m - r_-^m) \le \frac{2^{m-1}}{m}\alpha_m p_{max}(r_+ - r_-)r^{m-1}, \\ |\mathcal{I}_3| &\le 2CmKp_{max}\alpha_m r_-^{m+2} \le 2CmKp_{max}\alpha_m r^{m+2}. \end{aligned}$$

Finally, we notice that we can identify $B_{x_i}(0,r_-)$ with $B_{x_j}(0,r_-)$. From the assumed smoothness on p (which in particular is C^1) we see that for any $v \in B_{x_i}(0,r_-)$ we have

$$\big|p\big(\exp_{x_i}(v)\big) - p\big(\exp_{x_j}(v)\big)\big| \le C_p d_{\mathcal{M}}\big(\exp_{x_i}(v), \exp_{x_j}(v)\big) \le 3C_p r.$$

Then it follows that $|\mathcal{I}_1| \le 3C_p\alpha_m r^{m+1}$ and we get the desired result. ■

We now bound the difference $\big|b_i - b_j\big|$ for nearby points x_i, x_j, where we recall that

$$b_i := \int_{B_{x_i}(0,r_-)} \big(\exp_{x_i}(v) - x_i\big)p\big(\exp_{x_i}(v)\big)\,dv.$$

Proposition 10 *Suppose that x_i and x_j are such that $d_{\mathcal{M}}(x_i, x_j) \le r$. Then,*

$$\big|b_i - b_j\big| \le Cr^{m+3},$$

where the constant C can be written as

$$C = p_{max}\alpha_m\left(\frac{6\sqrt{m}}{R^2} + \Big(1 + \frac{4}{R}\Big)C_{\mathcal{M}}\right) + \frac{C_p}{R}\alpha_m,$$

where C_p is a constant that depends on bounds on first and second derivatives of the density p, and $C_{\mathcal{M}}$ is a constant that depends only on the change in second fundamental form along $\mathcal{M}$ (a third order term).

As we will see Proposition 10 can be proved combining several ideas from differential geometry. We present the required auxiliary results as we develop the proof of the proposition.

We start by conveniently writing b_i and b_j in a way that facilitates their direct comparison. Indeed, for any given $v \in B_{x_i}(0,r_-)$ let us consider the curves

$$\gamma_{v,i}(t) := \exp_{x_i}\left(t\frac{v}{|v|}\right), \quad t \in [0, |v|],$$

and

$$t \in [0, |v|] \mapsto x_i + t \in [0, |v|].$$

Thus, $\gamma_{v,i}$ is an arc-length parameterized geodesic on $\mathcal{M}$ that starts at the point x_i and at time $|v|$ passes though the point $\exp_{x_i}(v)$. Its initial velocity $\dot{\gamma}_{v,i}(0)$ is the vector $v/|v|$. On the other hand, while the second curve does not stay in $\mathcal{M}$ for $t > 0$, it does have the same starting point and velocity as $\gamma_{v,i}$. We can use the fundamental theorem of calculus to write:

$$\exp_{x_i}(v) - (x_i + v) = \int_0^{|v|}\left(\dot{\gamma}_{v,i} - \frac{v}{|v|}\right)dt,$$

as well as

$$\dot{\gamma}_{v,i}(t) - \frac{v}{|v|} = \int_0^t \ddot{\gamma}_{v,i}(s)ds, \quad \forall t \in [0, |v|]. \tag{19}$$

In particular, we have the second order representation

$$\exp_{x_i}(v) - x_i = v + \int_0^{|v|}\int_0^t \ddot{\gamma}_{v,i}(s)dsdt. \tag{20}$$

As a consequence of the previous formula we can rewrite b_i as

$$\begin{aligned} b_i &= \int_{B_{x_i}(0,r_-)} \big(\exp_{x_i}(v) - x_i\big)p\big(\exp_{x_i}(v)\big)dv \\ &= \int_{B_{x_i}(0,r_-)} p\big(\exp_{x_i}(v)\big)\int_0^{|v|}\int_0^t \ddot{\gamma}_{v,i}(s)dsdtdv + \int_{B_{x_i}(0,r_-)} vp\big(\exp_{x_i}(v)\big)dv. \end{aligned} \tag{21}$$

Completely analogous definitions and statements can be introduced to represent b_j.

With the objective of using the formula (21) to compare b_i and b_j we relate vectors in $T_{x_i}\mathcal{M}$ with vectors in $T_{x_j}\mathcal{M}$ by a convenient linear isometry $F_{ij} : T_{x_i}\mathcal{M} \mapsto T_{x_j}\mathcal{M}$ constructed using parallel transport.

Lemma 11 *Suppose that x_i and x_j are such that $d_{\mathcal{M}}(x_i, x_j) \le r$. Let $\phi : t \in [0, d_{\mathcal{M}}(x_i, x_j)] \mapsto \phi(t) \in \mathcal{M}$, be the arc-length parameterized geodesic starting at x_i at time zero and passing through x_j at time $t = d_{\mathcal{M}}(x_i, x_j)$. For an arbitrary vector $v \in T_{x_i}\mathcal{M}$ let V_v be the (unique) vector field along ϕ that solves the ODE*

$$\begin{cases} \frac{D}{dt}V_v(t) = 0, & t \in \big(0, d_{\mathcal{M}}(x_i, x_j)\big), \\ V_v(0) = v, \end{cases},$$

where $\frac{D}{dt}$ denotes the covariant derivative (on $\mathcal{M}$) along the curve ϕ. Then, the map F_{ij} defined by

$$F_{ij} : v \longmapsto \tilde{v} := V_v\big(d_{\mathcal{M}}(x_i, x_j)\big)$$

is a linear isometry. Moreover,

$$|v - \tilde{v}| \le \frac{1}{R}|v|d_{\mathcal{M}}(x_i, x_j), \quad \forall v \in T_{x_i}\mathcal{M}. \tag{22}$$

Proof First note that F_{ij} is a linear isometry since the ODE defining V_v is linear and the vector fields V_v are parallel to the curve ϕ by definition. To get the estimate (22) we can use the fundamental theorem of calculus and write

$$\tilde{v} = v + \int_0^t \dot{V}_v(s)ds,$$

where $t := d_{\mathcal{M}}(x_i, x_j)$. The fact that V_v is parallel along the curve ϕ implies that $\dot{V}_v(s) \in T_{\phi(s)}\mathcal{M}^{\perp}$ and furthermore that for arbitrary unit norm η with $\eta \in T_{\phi(s)}\mathcal{M}^{\perp}$ we have

$$|\langle \dot{V}_v(s), \eta\rangle| = |\langle S_\eta(V_v(s)), \dot{\phi}(s)\rangle| \leq \|S_\eta\||V_v(s)||\dot{\phi}(s)| = \|S_\eta\||v|,$$

where S_η is the so called *shape operator* representing the second fundamental form (see Proposition 2.3. Chapter 6 in do Carmo (1992)). The relevance of the previous inequality is that when combined with Proposition 6.1 in Niyogi et al. (2008) (which shows that the operator norm of the second fundamental form is bounded by $1/R$) it implies that

$$|\dot{V}_v(s)| \leq \frac{|v|}{R}, \quad \forall s \in [0, t].$$

Therefore,

$$|\tilde{v} - v| \leq \int_0^t |\dot{V}_v(s)|ds \leq \frac{|v|}{R} d_{\mathcal{M}}(x_i, x_j),$$

establishing in this way the desired bound. ■

From now on, for a given $v \in B_{x_i}(0, r_-)$ we let $\tilde{v} \in B_{x_j}(0, r_-)$ be its image under F_{ij}. We consider the curve

$$\gamma_{\tilde{v},j}(t) := \exp_{x_j}\left(t\frac{\tilde{v}}{|\tilde{v}|}\right), \quad t \in [0, |\tilde{v}|],$$

where we recall that $|v| = |\tilde{v}|$ because F_{ij} is a linear isometry. We can then make a change of variables and write b_j as

$$b_j = \int_{B_{x_i}(0,r_-)} p\big(\exp_{x_j}(\tilde{v})\big) \int_0^{|v|} \int_0^t \ddot{\gamma}_{\tilde{v},j}(s)dsdtdv + \int_{B_{x_i}(0,r_-)} \tilde{v}p\big(\exp_{x_j}(\tilde{v})\big)dv. \tag{23}$$

In the next lemma we find bounds for the norms of accelerations.

Lemma 12 *Let $v \in B_{x_i}(0, r_-)$ and let $\tilde{v}$ be as in Lemma 11. Then, for all $t \in [0, |v|]$ we have*

$$|\ddot{\gamma}_{v,i}(t)| \leq \frac{1}{R},$$

and

$$|\dot{\gamma}_{v,i}(t) - \dot{\gamma}_{\tilde{v},j}(t)| \leq 2\frac{|v|}{R} + \frac{d_{\mathcal{M}}(x_i, x_j)}{R}.$$

Proof The first inequality appears in the proof of Proposition 2 in Niyogi et al. (2008) and is obtained in a completely analogous way as we obtained the bound for $\dot{V}_v$ in the proof of Lemma 11 (given that unit speed geodesics are auto parallel).

To prove the second estimate, we notice that from the first bound and (19) it follows that

$$\left|\dot{\gamma}_{v,i}(t) - \frac{v}{|v|}\right| \leq \frac{|v|}{R}, \quad \forall t \in [0, |v|].$$

Naturally, a similar inequality holds for $\gamma_{\tilde{v},j}$. Using Lemma 11 we conclude that for all $t \in [0, |v|]$ (recall that $|v| = |\tilde{v}|$)

$$\begin{aligned}
|\dot{\gamma}_{v,i}(t) - \dot{\gamma}_{\tilde{v},j}(t)| &\leq \left|\frac{v}{|v|} - \frac{\tilde{v}}{|\tilde{v}|}\right| + \left|\dot{\gamma}_{v,i}(t) - \frac{v}{|v|}\right| + \left|\dot{\gamma}_{\tilde{v},i}(t) - \frac{\tilde{v}}{|\tilde{v}|}\right| \\
&\leq \frac{1}{|v|}|v - \tilde{v}| + 2\frac{|v|}{R} \\
&\leq \frac{d_{\mathcal{M}}(x_i, x_j)}{R} + 2\frac{|v|}{R}.
\end{aligned}$$

■

From our assumption that the density p was in $C^2(\mathcal{M})$ it follows that

$$\begin{aligned}
p\big(\exp_{x_i}(v)\big) &= p(x_i) + \langle \nabla p(x_i), v\rangle + R_i(v), \\
p\big(\exp_{x_j}(\tilde{v})\big) &= p(x_j) + \langle \nabla p(x_j), \tilde{v}\rangle + R_j(\tilde{v}),
\end{aligned}$$

where $\nabla p(x_i)$ is the gradient (in $\mathcal{M}$) of p at the point x_i, and the remainder terms satisfy

$$\max\{|R_i(v)|, |R_j(\tilde{v})|\} \leq C_p|v|^2,$$

for a constant C_p that depends on a uniform bound on second derivatives of p. Likewise,

$$\max\{|p(x_i) - p(x_j)|, |\nabla p(x_i) - \nabla p(x_j)|\} \leq C_p d_{\mathcal{M}}(x_i, x_j).$$

Plugging the previous identities in the expressions (21) and (23), using

$$\int_{B_{x_i}(0,r_-)} p(x_i)vdv = 0, \quad \int_{B_{x_i}(0,r_-)} p(x_j)\tilde{v}d\tilde{v} = 0,$$

inequality (22), the bound on accelerations from Lemma 12, and finally Assumption 2, we can conclude that

$$\begin{aligned}
\big|b_i - b_j\big| &\leq \int_{B_{x_i}(0,r_-)} \int_0^{|v|} \int_0^t \big|p(x_j)\ddot{\gamma}_{\tilde{v},j}(s) - p(x_i)\ddot{\gamma}_{v,i}(s)\big|dsdtdv + \frac{C_p}{R}\alpha_m r^{m+2}\big(r + d_{\mathcal{M}}(x_i, x_j)\big) \\
&\leq p_{max} \cdot \int_{B_{x_i}(0,r_-)} \int_0^{|v|} \int_0^t \big|\ddot{\gamma}_{\tilde{v},j}(s) - \ddot{\gamma}_{v,i}(s)\big|dsdtdv \\
&+ \int_{B_{x_i}(0,r_-)} \int_0^{|v|} \int_0^t \big|p(x_j) - p(x_i)\big|\big|\ddot{\gamma}_{v,i}(s)\big|dsdtdv + \frac{C_p}{R}\alpha_m r^{m+2}\big(r + d_{\mathcal{M}}(x_i, x_j)\big) \\
&\leq p_{max} \cdot \int_{B_{x_i}(0,r_-)} \int_0^{|v|} \int_0^t \big|\ddot{\gamma}_{\tilde{v},j}(s) - \ddot{\gamma}_{v,i}(s)\big|dsdtdv + \frac{C_p}{R}\alpha_m r^{m+2}\big(r + d_{\mathcal{M}}(x_i, x_j)\big).
\end{aligned} \tag{24}$$

In the above, C_p is a constant that depends on derivatives of p of order 1 and order 2 (and in particular is equal to zero when p is constant) and α_m is the volume of the m-dimensional unit ball.

Proposition 10 now follows from the next lemma where we bound the difference of accelerations.

Lemma 13 *Let $v \in B_{x_i}(0, r_-)$ and let $\tilde{v}$ be as in Lemma 11. Then, for all $t \in [0, |v|]$ we have*

$$\left|\ddot{\gamma}_{v,i}(t) - \ddot{\gamma}_{\tilde{v},j}(t)\right| \leq \left(2\frac{\sqrt{m}}{R^2} + C_{\mathcal{M}}\right)\left(2|v| + d_{\mathcal{M}}(x_i, x_j)\right) + 2C_{\mathcal{M}}\left(\frac{|v|}{R} + \frac{d_{\mathcal{M}}(x_i, x_j)}{R}\right),$$

where $C_{\mathcal{M}}$ is a constant that depends on $\mathcal{M}$ (a third order term).

Proof For a fixed $t \in [0, |v|]$ we let

$$x := \gamma_{v,i}(t), \quad \tilde{x} := \gamma_{\tilde{v},j}(t).$$

We start by constructing a convenient linear map

$$\eta \in T_x\mathcal{M}^{\perp} \mapsto \tilde{\eta} \in T_{\tilde{x}}\mathcal{M}^{\perp}.$$

For this purpose we use an *orthonormal frame* $E_1, \ldots, E_m$ on a neighborhood (in $\mathcal{M}$) of x containing the geodesic connecting x and $\tilde{x}$. The frame is constructed by parallel transporting an orthonormal basis $E_1(x), \ldots, E_m(x)$ of $T_x\mathcal{M}$ along geodesics emanating from x. Now, associated to $\eta \in T_x\mathcal{M}^{\perp}$ we define the (normal) vector field N_η by

$$N_\eta := \eta - \sum_{l=1}^{m} \langle E_l, \eta\rangle E_l.$$

Equivalently, N_η can be written as

$$N_\eta(z) = \Pi_z(\eta),$$

where for a point $z \in \mathcal{M}$, Π_z denotes the projection onto $T_z\mathcal{M}^{\perp}$ (the orthogonal complement of the tangent plane at z).

Let $\phi_{x\tilde{x}}$ be the arc-length parameterized geodesic with $\phi_{x\tilde{x}}(0) = x$ and $\phi_{x\tilde{x}}(\tilde{t}) = \tilde{x}$. We restrict the vector field N_η to the curve $\phi_{x,\tilde{x}}$ and abuse notation slightly to write $N_\eta(s)$ and $E_j(s)$ for the value of the vector fields at the point $\phi_{x\tilde{x}}(s)$. We let $\tilde{\eta} := N_\eta(\tilde{t})$ and notice that

$$|\eta - \tilde{\eta}| = \left(\sum_{l=1}^{m} \langle E_l(\tilde{t}), \eta\rangle^2\right)^{1/2} = \left(\sum_{l=1}^{m} \langle E_l(\tilde{t}) - E_l(0), \eta\rangle^2\right)^{1/2} \leq \frac{\sqrt{m} d_{\mathcal{M}}(x, \tilde{x})|\eta|}{R}, \tag{25}$$

where in the last line we have used that $|E_l(\tilde{t}) - E_l(0)| \leq \frac{\tilde{t}}{R}$ (proved in the exact same way as (22)).

Let $\eta \in T_x\mathcal{M}^{\perp}$ be a unit norm vector and let $\tilde{\eta}$ be as constructed before. Since N_η is a normal vector field which locally extends η we can follow the characterization for the shape operator in Proposition 2.3 Chapter 6 in do Carmo (1992) and deduce that:

$$\langle \ddot{\gamma}_{v,i}(t), \eta\rangle = \langle S_\eta(\dot{\gamma}_{v,i}), \dot{\gamma}_{v,i}(t)\rangle = \langle \frac{d}{dt} N_\eta(\gamma_{v,i}(t)), \dot{\gamma}_{v,i}(t)\rangle.$$

Moreover, the smoothness of the manifold $\mathcal{M}$ allows us to extend N_η smoothly to a neighborhood in $\mathbb{R}^d$ of x and $\tilde{x}$ (we also use N_η to represent the extension). Indeed, for any point z in a tubular neighborhood of $\mathcal{M}$ of width smaller than R we can define

$$N_\eta(z) := N_\eta(Proj_{\mathcal{M}}(z)),$$

where $Proj_{\mathcal{M}}$ is the projection onto $\mathcal{M}$ (which is well defined for points within distance R from $\mathcal{M}$). The smoothness of N_η in particular implies that

$$\|DN_\eta(x) - DN_\eta(\tilde{x})\| \leq C_{\mathcal{M}}|x - \tilde{x}| \leq C_{\mathcal{M}} d_{\mathcal{M}}(x, \tilde{x}),$$

$$\|DN_\eta(\tilde{x})\| \leq C_{\mathcal{M}},$$

where DN_η is the matrix of derivatives of the vector field N_η, and where $C_{\mathcal{M}}$ is some constant that only depends on $\mathcal{M}$. With this extension at hand, we can then use the chain rule and write:

$$\langle \ddot{\gamma}_{v,i}(t), \eta\rangle = \langle DN(x)\dot{\gamma}_{v,i}(t), \dot{\gamma}_{v,i}(t)\rangle,$$

and in a similar fashion

$$\langle \ddot{\gamma}_{\tilde{v},j}(t), \eta\rangle = \langle \ddot{\gamma}_{\tilde{v},j}(t), \eta - \tilde{\eta}\rangle + \langle \ddot{\gamma}_{\tilde{v},j}(t), \tilde{\eta}\rangle = \langle \ddot{\gamma}_{\tilde{v},j}(t), \eta - \tilde{\eta}\rangle + \langle -DN(\tilde{x})\dot{\gamma}_{\tilde{v},j}(t), \dot{\gamma}_{\tilde{v},j}(t)\rangle.$$

Using the triangle and Cauchy-Schwarz inequalities we obtain

$$\begin{aligned}|\langle \ddot{\gamma}_{v,i}(t) - \ddot{\gamma}_{\tilde{v},j}(t), \eta\rangle| \leq & |\ddot{\gamma}_{\tilde{v},j}||\eta - \tilde{\eta}| + \|DN(x) - DN(\tilde{x})\||\dot{\gamma}_{v,i}|^2 + \\ & \|DN(\tilde{x})\||\dot{\gamma}_{v,i} - \dot{\gamma}_{\tilde{v},j}|(|\dot{\gamma}_{\tilde{v},j}| + |\dot{\gamma}_{v,i}|) \\ \leq & \frac{\sqrt{m}}{R^2} d_{\mathcal{M}}(x, \tilde{x}) + C_{\mathcal{M}} d_{\mathcal{M}}(x, \tilde{x}) + 2C_{\mathcal{M}}\left(\frac{|v|}{R} + \frac{d_{\mathcal{M}}(x_i, x_j)}{R}\right).\end{aligned}$$

Since the above inequality holds for all $\eta \in T_x\mathcal{M}^{\perp}$ with norm one, we conclude that

$$|\Pi_x(\ddot{\gamma}_{v,i}(t)) - \Pi_x(\ddot{\gamma}_{\tilde{v},j}(t))| \leq \frac{\sqrt{m}}{R^2} d_{\mathcal{M}}(x, \tilde{x}) + C_{\mathcal{M}} d_{\mathcal{M}}(x, \tilde{x}) + 2C_{\mathcal{M}}\left(\frac{|v|}{R} + \frac{d_{\mathcal{M}}(x_i, x_j)}{R}\right),$$

where we recall Π_x represents the projection onto $T_x\mathcal{M}^{\perp}$. Moreover, since $\ddot{\gamma}_{v,i}(t)$ is the acceleration of a unit speed geodesic passing through x, we know that $\ddot{\gamma}_{v,i}(t) \in T_x\mathcal{M}^{\perp}$, so that $\Pi_x(\ddot{\gamma}_{v,i}) = \ddot{\gamma}_{v,i}$. Similarly we have $\Pi_{\tilde{x}}(\ddot{\gamma}_{\tilde{v},j}) = \ddot{\gamma}_{\tilde{v},j}$ (where $\Pi_{\tilde{x}}$ represents projection onto $T_{\tilde{x}}\mathcal{M}^{\perp}$) . Hence

$$|\ddot{\gamma}_{v,i}(t) - \ddot{\gamma}_{\tilde{v},j}(t)| \leq |\Pi_x\ddot{\gamma}_{v,i}(t) - \Pi_x\ddot{\gamma}_{\tilde{v},j}(t)| + |\Pi_x\ddot{\gamma}_{\tilde{v},j}(t) - \ddot{\gamma}_{\tilde{v},j}(t)|, \tag{26}$$

and so it remains to find a bound for $|\Pi_x \ddot{\gamma}_{\tilde{v},j}(t) - \ddot{\gamma}_{\tilde{v},j}(t)|$. We can write

$$\Pi_x \ddot{\gamma}_{\tilde{v},j} = \ddot{\gamma}_{\tilde{v},j} - \sum_{l=1}^{m} \langle \ddot{\gamma}_{\tilde{v},j}, E_l(0)\rangle E_l(0).$$

Therefore,

$$|\ddot{\gamma}_{\tilde{v},j} - \Pi_x \ddot{\gamma}_{\tilde{v},j}| = \left(\sum_{l=1}^{m} \langle \ddot{\gamma}_{\tilde{v},j}, E_l(0)\rangle^2\right)^{1/2} = \left(\sum_{l=1}^{m} \langle \ddot{\gamma}_{\tilde{v},j}, E_l(0) - E_l(\tilde{t})\rangle^2\right)^{1/2} \leq \sqrt{m}\frac{d_{\mathcal{M}}(x,\tilde{x})}{R^2}.$$

Putting everything together we deduce that

$$\begin{aligned}|\ddot{\gamma}_{v,i} - \ddot{\gamma}_{\tilde{v},j}| &\leq (2\frac{\sqrt{m}}{R^2} + C_{\mathcal{M}})d_{\mathcal{M}}(x,\tilde{x}) + 2C_{\mathcal{M}}\left(\frac{|v|}{R} + \frac{d_{\mathcal{M}}(x_i,x_j)}{R}\right)\\ &\leq (2\frac{\sqrt{m}}{R^2} + C_{\mathcal{M}})(2|v| + d_{\mathcal{M}}(x_i,x_j)) + 2C_{\mathcal{M}}\left(\frac{|v|}{R} + \frac{d_{\mathcal{M}}(x_i,x_j)}{R}\right),\end{aligned}$$

where in the last step we have used the triangle inequality

$$d_{\mathcal{M}}(x,\tilde{x}) \leq d_{\mathcal{M}}(x,x_i) + d_{\mathcal{M}}(x_i,x_j) + d_{\mathcal{M}}(x_j,\tilde{x}) \leq 2|v| + d_{\mathcal{M}}(x_i,x_j).$$

■

Remark 14 *Notice that the computations in the proof of Proposition 10 also show that*

$$|b_i| \leq Cr^{m+2}, \qquad i = 1,\ldots,n.$$

Indeed, this can be seen directly from (21)*, Lemma 12 (which bounds the acceleration term), and the fact that the first term on the right-hand side of the following expression drops by symmetry:*

$$\int_{B_{x_i}(0,r_-)} p(\exp_{x_i}(v))v dv = p(x_i)\int_{B_{x_i}(0,r_-)} v dv + \int_{B_{x_i}(0,r_-)} (\langle \nabla p(x_i), v\rangle + R_i(v))v dv.$$

2.5 Bounding Sampling Error

We will make use of two concentration inequalities to bound the sampling error. We first recall Hoeffding's inequality.

Lemma 15 (Hoeffding's inequality) *Let $w_1,\ldots,w_n$ be i.i.d samples from a random variable w taking values in the interval [0,1] and let $\overline{w}$ be the sample average. Then,*

$$\mathbb{P}\left(|\overline{w} - \mathbb{E}[\overline{w}]| > t\right) \leq 2e^{-2nt^2}.$$

The next is a generalization for random vectors that follows directly from the simple and elegant work Pinelis (1992) (more precisely, Theorem 3).

Lemma 16 *Let $W_1,\ldots,W_n$ be i.i.d samples from a random vector W such that $|W| \leq M$ for some constant M, and $\mathbb{E}[W] = 0$. Let $\overline{W}$ be the sample average. Then,*

$$\mathbb{P}\left(\left|\overline{W} - \mathbb{E}[\overline{W}]\right| > \sqrt{\frac{M^2}{n}t}\right) \leq 2e^{-t^2/16}.$$

Proposition 17 *Suppose Assumption 3 holds. Then,*

$$\mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{2^{m+4}}{\alpha_m p_{min}}}r^3\right) \leq 4e^{-cnr^{\max\{2m,m+4\}}}, \text{ where } c = \min\left\{\frac{\alpha_m^2 p_{min}^2}{4^{m+2}}, \frac{1}{16}\right\}.$$

In particular, if $nr^{\max\{2m,m+4\}} \gg 1$, then $\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| \leq \sqrt{\frac{2^{m+4}}{\alpha_m p_{min}}}r^3$ with high probability.

Proof Let N_i be the number of points in $B(y_i,r)$. Notice that $\tilde{x}_i + \tilde{z}_i - \mathbb{E}_i[\tilde{X}_i + \tilde{Z}_i]$ is centered and bounded by $2r$ in norm, and $\overline{y}_i = \overline{\tilde{x}_i + \tilde{z}_i}$. Then Lemma 16 implies

$$\mathbb{P}_i\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{4r^2}{N_i}t}\,\middle|\, N_i\right) \leq 2e^{-t^2/16}.$$

By the law of iterated expectations it follows that

$$\mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{4r^2}{N_i}t}\right) \leq 2e^{-t^2/16}.$$

Next note that N_i, the number of points y_j in $B(y_i,r)$, can be bounded below by $\widetilde{N}_i$, the number of points x_j that lie in the ball $B_{\mathcal{M}}(x_i,r_-)$. Thus,

$$\mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{4r^2}{\widetilde{N}_i}t}\right) \leq 2e^{-t^2/16}. \tag{27}$$

Now we find probabilistic bound for $\widetilde{N}_i$. Let $w_j = \mathbb{1}\{x_j \in B(x_i,r_-)\}$. Then given x_i, the w_j are i.i.d samples from Bernoulli(q_i), where $q_i = \mu(B_{\mathcal{M}}(x_i,r_-))$. Lemma 15 implies

$$\mathbb{P}_i\left(\left|\widetilde{N}_i - nq_i\right| > nt\,\middle|\, x_i\right) \leq 2e^{-2nt^2}.$$

Again by the law of iterated expectation and rearranging terms, we have

$$\mathbb{P}\left(\widetilde{N}_i < n(q_i - t)\right) \leq 2e^{-2nt^2}. \tag{28}$$

Combining (27) and (28), we obtain

$$\begin{aligned}\mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{4r^2}{n(q_i-s)}t}\right) &= \mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{4r^2}{n(q_i-s)}t}, \widetilde{N}_i < n(q_i-s)\right)\\ &\quad + \mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{4r^2}{n(q_i-s)}t}, \widetilde{N}_i \geq n(q_i-s)\right)\\ &\leq \mathbb{P}\left(\widetilde{N}_i < n(q_i-s)\right) + \mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{4r^2}{\widetilde{N}_i}t}\right)\\ &\leq 2e^{-2ns^2} + 2e^{-t^2/16}.\end{aligned}$$

Under Assumption 3, (11) implies $q_i \geq \frac{\alpha_m p_{min}}{2^{m+1}} r^m$. Taking $s = \frac{\alpha_m p_{min}}{2^{m+2}} r^m$ and $t = \sqrt{nr^{m+4}}$, we see that

$$\mathbb{P}\left(\left|\overline{y}_i - \mathbb{E}_i[\overline{y}_i]\right| > \sqrt{\frac{2^{m+4}}{\alpha_m p_{min}} r^3} \right) \leq 2e^{-\frac{\alpha_m^2 p_{min}^2}{4^{m+2}} nr^{2m}} + 2e^{-nr^{m+4}/16} \leq 4e^{-cnr^{\max\{2m,m+4\}}},$$

where $c = \min\left\{\frac{\alpha_m^2 p_{min}^2}{4^{m+2}}, \frac{1}{16}\right\}$. The result then follows. ■

Theorem 1 now follows by combining Lemma 9, Propositions 8, 10, and Proposition 17 together with a union bound.

3. From Local Regularization to Global Estimates

In this section we use the local estimates (7) to show spectral convergence of $\Delta_{\bar{\mathcal{Y}}_n,\varepsilon}$ towards the continuum Laplace-Beltrami operator. We first make some definitions. Recall that the graph $\Gamma_{\delta,\varepsilon} = ([n], W)$ has weights

$$W(i,j) = \frac{2(m+2)}{\alpha_m \varepsilon^{m+2} n} \mathbb{1}\{\delta(i,j) < \varepsilon\},$$

where m is the dimension of $\mathcal{M}$ and α_m is the volume of the $m-$dimensional Euclidean unit ball. For a function $u : [n] \to \mathbb{R}$, we denote its value on the i-th node as $u(i)$. We then define the discrete Dirichlet energy of u as

$$E_{\delta,\varepsilon}[u] = \frac{m+2}{\alpha_m \varepsilon^{m+2} n} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1}\{\delta(i,j) < \varepsilon\} |u(i) - u(j)|^2$$

and the L^2 norm of u as

$$\|u\|^2 = \frac{1}{n} \sum_{i=1}^{n} |u(i)|^2.$$

Given that $\Delta_{\delta,\varepsilon}$ is a positive semi-definite operator, we can use the minimax principle (see for example Lieb et al. (2001)) to write

$$\lambda_\ell(\Gamma_{\delta,\varepsilon}) = \min_{L} \max_{u \in L \setminus \{0\}} \frac{E_{\delta,\varepsilon}[u]}{\|u\|^2},$$

where $\lambda_\ell(\Gamma_{\delta,\varepsilon})$ is the ℓ-th smallest eigenvalue of $\Delta_{\Gamma_{\delta,\varepsilon}}$ and the minimum is taken over all subspaces L of dimension ℓ. The following lemma compares the eigenvalues of the discrete graphs constructed using $\delta_{\mathcal{X}_n}$ and $\delta_{\bar{\mathcal{Y}}_n}$.

Lemma 18 *Let η be the bound in (7) so that for all i, j with $d_{\mathcal{M}}(x_i, x_j) \leq r$ we have*

$$\left|\delta_{\mathcal{X}_n}(i,j) - \delta_{\bar{\mathcal{Y}}_n}(i,j)\right| \leq \eta.$$

Suppose that ε is chosen so that $\varepsilon \geq 2Cm\eta$, for some universal constant C. Then,

$$\left(1 - Cm\frac{\eta}{\varepsilon}\right)\lambda_\ell(\Gamma_{\mathcal{X}_n,\varepsilon-\eta}) \leq \lambda_\ell(\Gamma_{\bar{\mathcal{Y}}_n,\varepsilon}) \leq \left(1 + Cm\frac{\eta}{\varepsilon}\right)\lambda_\ell(\Gamma_{\mathcal{X}_n,\varepsilon+\eta}). \tag{29}$$

Proof We first compare the Dirichlet energies. Since $\delta_{\mathcal{X}_n}(i,j) < \delta_{\bar{\mathcal{Y}}_n}(i,j) + \eta$, we have

$$\begin{aligned} E_{\bar{\mathcal{Y}}_n,\varepsilon}[u] &= \frac{m+2}{\alpha_m \varepsilon^{m+2} n} \sum_i \sum_j \mathbb{1}\{\delta_{\bar{\mathcal{Y}}_n}(i,j) < \varepsilon\} |u_i - u_j|^2 \\ &\leq \frac{m+2}{\alpha_m \varepsilon^{m+2} n} \sum_i \sum_j \mathbb{1}\{\delta_{\mathcal{X}_n}(i,j) < \varepsilon + \eta\} |u_i - u_j|^2 \\ &= \left(\frac{\varepsilon+\eta}{\varepsilon}\right)^{m+2} E_{\mathcal{X}_n,\varepsilon+\eta}[u] \\ &\leq \left(1 + Cm\frac{\eta}{\varepsilon}\right) E_{\mathcal{X}_n,\varepsilon+\eta}[u]. \end{aligned} \tag{30}$$

Now we use the minimax principle to show the upper-bound on (29). Let $u_1, \ldots, u_\ell$ be the first l eigenvectors of $\Delta_{\mathcal{X}_n,\varepsilon+\eta}$ and let $L = \text{span}\{u_1, \ldots, u_\ell\}$. Then $\dim L = \ell$ and for any $u \in L$, $E_{\mathcal{X}_n,\varepsilon+\eta}[u] \leq \lambda_\ell(\Gamma_{\mathcal{X}_n,\varepsilon+\eta})\|u\|^2$. Then by (30), we have

$$\lambda_\ell(\Gamma_{\bar{\mathcal{Y}}_n,\varepsilon}) \leq \max_{L \setminus 0} \frac{E_{\bar{\mathcal{Y}}_n,\varepsilon}[u]}{\|u\|^2} \leq \left(1 + Cm\frac{\eta}{\varepsilon}\right) \max_{L \setminus 0} \frac{E_{\mathcal{X}_n,\varepsilon+\eta}[u]}{\|u\|^2} = \left(1 + Cm\frac{\eta}{\varepsilon}\right)\lambda_\ell(\Gamma_{\mathcal{X}_n,\varepsilon+\eta}).$$

By a similar argument applied to $\Gamma_{\mathcal{X}_n,\varepsilon-\eta}$ and $\Gamma_{\bar{\mathcal{Y}}_n,\varepsilon}$, we get the lower-bound in (29). ■

Remark 19 *With the convergence of eigenvalues and the relationship between the Dirichlet energies it is also possible to make statements about convergence of eigenvectors (or better yet, spectral projections).*

The spectral convergence towards the continuum (Theorem 3) is a consequence of the following theorem, proved in (García Trillos et al., 2018, Corollary 2).

Theorem 20 *Let d_∞ be the ∞-OT distance between μ_n and μ. Suppose ε satisfies the conditions in Equation (9) and that Assumptions 1 and 2 hold. Then*

$$\frac{|\lambda_\ell(\Gamma_{\mathcal{X}_n,\varepsilon}) - \lambda_\ell(\mathcal{M})|}{\lambda_\ell(\mathcal{M})} \leq \tilde{C}\left(\frac{d_\infty}{\varepsilon} + \left(1 + \sqrt{\lambda_\ell(\mathcal{M})}\right)\varepsilon + \left(K + \frac{1}{R^2}\right)\varepsilon^2\right),$$

where $\tilde{C}$ only depends on m and the regularity of p.

Combining Lemma 18 and Theorem 20 gives Theorem 3.

4. Numerical Experiments

In this section we present a series of numerical experiments where we conduct local regularization on three different datasets. In Subsection 4.1 we consider a toy example with artificial data generated by perturbing points sampled uniformly from the unit, two-dimensional sphere embedded in $\mathbb{R}^d$ with $d = 100$. We show that the approximation of the hidden Euclidean distances between unperturbed points is significantly improved by locally regularizing the data, and that this improvement translates into better spectral approximation of the spherical Laplacian. Our numerical findings corroborate the theory developed in the previous two sections. In Subsection 4.2 we consider the two-moon and MNIST datasets and show that graphs constructed with locally regularized data can be used to improve the performance of a simple graph-based optimization method for semi-supervised classification.

4.1 Distance & Spectrum

Here we study the effect of local regularization on distance approximation and spectral convergence, as an illustration of the results from Sections 2 and 3. In our toy model we consider uniform samples from the unit two-dimensional sphere $\mathcal{M} = \mathcal{S}$ embedded in $\mathbb{R}^d$, with $d = 100$. The motivation for such a choice is that the eigenvalues of the associated Laplace-Beltrami operator on $\mathcal{S}$ are known explicitly (see for example Olver). Indeed, after appropriate normalization, $\Delta_{\mathcal{S}}$ admits eigenvalues $\ell(\ell+1), \ell \in \mathbb{N}$, with corresponding multiplicity $2\ell+1$.

The dataset is generated by sampling $n = 3000$ points x_i uniformly from the sphere and adding uniform noise z_i normal to the tangent plane, and bounded by σ in norm. To be more precise, the noise is normal to the sphere for the first three dimensions and uniform in all directions for the rest dimensions. Local regularization is performed by taking $r \propto \sqrt{\sigma}$ and the graph is constructed with $\varepsilon = 2n^{-1/4}$. The optimal proportion constant in r is not obvious from our theory and in the experiments below we choose $r = \sqrt{\sigma}/3$ for $\sigma = 0.1$ and $r = \sqrt{\sigma}$ for the rest of the σ's. We first show that the $\bar{y}_i$ give a better approximation of the pairwise distances of the x_i than the y_i do. We only consider those nodes i, j such that $\delta_{\mathcal{X}_n}(i,j) < \varepsilon$ (i.e. the nodes that are relevant for the construction of the graph Laplacians). More precisely, let $D_{\mathcal{X}_n}$ be the matrix whose ijth entry is $\delta_{\mathcal{X}_n}(i,j)\mathbb{1}\{\delta_{\mathcal{X}_n}(i,j) < \varepsilon\}$. Similarly, we define $[D_{\mathcal{Y}_n}]_{ij} = \delta_{\mathcal{Y}_n}(i,j)\mathbb{1}\{\delta_{\mathcal{X}_n}(i,j) < \varepsilon\}$ and $[D_{\bar{\mathcal{Y}}_n}]_{ij} = \delta_{\bar{\mathcal{Y}}_n}(i,j)\mathbb{1}\{\delta_{\mathcal{X}_n}(i,j) < \varepsilon\}$. In Table 1 we compare the entrywise ∞-norm of the $D_{\mathcal{X}_n} - D_{\mathcal{Y}_n}$ and $D_{\mathcal{X}_n} - D_{\bar{\mathcal{Y}}_n}$ for different values of σ. We see that the improvement is substantial.

	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	$\sigma = 0.7$	$\sigma = 0.9$
$\lVert D_{\mathcal{X}_n} - D_{\mathcal{Y}_n} \rVert_\infty$	0.095	0.298	0.505	0.723	0.937
$\lVert D_{\mathcal{X}_n} - D_{\bar{\mathcal{Y}}_n} \rVert_\infty$	0.084	0.090	0.087	0.093	0.144

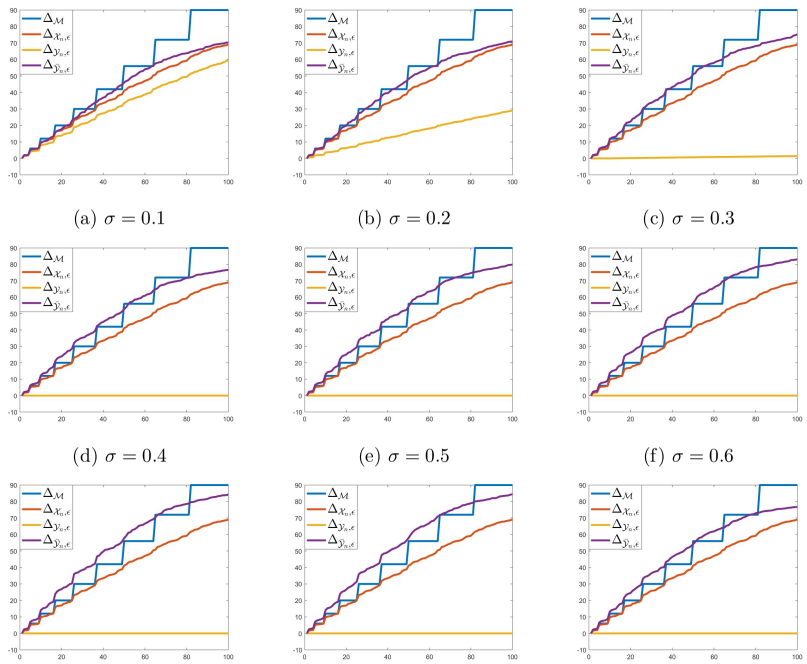
Table 1: Entrywise ∞-norm of $D_{\mathcal{X}_n} - D_{\mathcal{Y}_n}$ and $D_{\mathcal{X}_n} - D_{\bar{\mathcal{Y}}_n}$ on $\mathcal{S}$ for several σ's.

Next we study the spectral approximation of Laplacians by comparing the spectra of $\Delta_{\mathcal{X}_n,\varepsilon}$, $\Delta_{\mathcal{Y}_n,\varepsilon}$ with that of $\Delta_{\bar{\mathcal{Y}}_n,\varepsilon}$. Note that since the x_i are uniformly distributed, the density p on $\mathcal{S}$ that they are sampled from is constant and equal to $\frac{1}{\mathrm{vol}\mathcal{M}}$. So for the spectra of the graph Laplacians to match in scale with that of $\Delta_{\mathcal{S}}$, the weights should be rescaled according to

$$W(i,j) = \frac{2(m+2)\mathrm{vol}(\mathcal{M})}{\alpha_m \varepsilon^{m+2} n},$$

where $\mathrm{vol}(\mathcal{M})$ is the volume of the manifold and equals 4π in this case. In Figure 1 we compare the first 100 eigenvalues of $\Delta_{\mathcal{X}_n,\varepsilon}$, $\Delta_{\mathcal{Y}_n,\varepsilon}$, and $\Delta_{\bar{\mathcal{Y}}_n,\varepsilon}$ with the continuum spectrum. We see that when the noise size is large, the Euclidean graph Laplacian $\Delta_{\mathcal{Y}_n,\varepsilon}$ does not give a meaningful approximation of the continuum spectrum, while the locally regularized version $\Delta_{\bar{\mathcal{Y}}_n,\varepsilon}$ still performs well.

Remark 21 *While our theory in Section 2 suggests the choice $r \propto \sqrt{\sigma}$ in the small r and large n limit, for practical purposes some other scalings may give better results. Indeed for*



(a) $\sigma = 0.1$ (b) $\sigma = 0.2$ (c) $\sigma = 0.3$

(d) $\sigma = 0.4$ (e) $\sigma = 0.5$ (f) $\sigma = 0.6$

(g) $\sigma = 0.7$ (h) $\sigma = 0.8$ (i) $\sigma = 0.9$

Figure 1: Comparison of spectra of continuum Laplacian, $\Delta_{\mathcal{X}_n,\varepsilon}$, $\Delta_{\mathcal{Y}_n,\varepsilon}$ and $\Delta_{\bar{\mathcal{Y}}_n,\varepsilon}$ for different values of σ.

the above σ's, choosing $r = \sigma$ seems to give better spectral approximation. The choice of the local-regularization parameter will be further investigated in Subsection 4.2.2 in the context of a semi-supervised classification task, where a data-driven (cross-validation) approach can be used.

4.2 Classification

In this subsection we demonstrate the practical use of local regularization by applying it to classification problems. To show the potential benefits, we consider synthetic and real datasets, namely the two moons and MNIST datasets. Since in one of our experiments we study a real dataset, where in general the connectivity parameter in an ε graph is hard to tune, we instead consider fully connected graphs with self-tuning weights. Precisely, given a similarity $\delta : [n] \times [n] \to [0, \infty)$ we define, following Zelnik-Manor and Perona (2005), the

weights by

$$W(i,j) = \exp\left(-\frac{\delta(i,j)^2}{2\tau(i)\tau(j)}\right), \tag{31}$$

where $\tau(i)$ is the similarity between the i-th data point and its K-th nearest neighbor with respect to the distance δ. As before, we denote by $\Gamma_{\mathcal{X}_n}, \Gamma_{\mathcal{Y}_n}$ and $\Gamma_{\bar{\mathcal{Y}}_n}$ the graphs constructed with similarities $\delta_{\mathcal{X}_n}, \delta_{\mathcal{Y}_n}$, and $\delta_{\bar{\mathcal{Y}}_n}$. Instead of specifying a universal ε representing the connectivity length-scale, the neighborhood for each point is selected from using the local geometry which varies in space. It amounts to choosing different values of ε adaptively depending on the local scale, as proposed in Zelnik-Manor and Perona (2005). Since the $\tau(i)$ are defined by considering K-nearest neighbors, a natural variant of the above fully connected graph is to set the weights to be 0 whenever x_i and x_j are not among the K-nearest neighbors of each other. In other words, we can construct a (symmetrized) K-NN graph with the same K as in the definition of $\tau(i)$ and the nonzero weights are the same as above. It turns out that empirically this K-NN version can improve the classification performance substantially, but to illustrate the local regularization idea, we will present results for both graph constructions. We shall denote these two types of graphs as fully-connected and K-NN variants for brevity, or fully and K-NN for short.

In the following, we focus on the semi-supervised learning setting where we are given n data points with the first J being labeled. The classification is done by minimizing a probit functional as explained below. Let Δ_δ be a normalized graph Laplacian constructed on the dataset, which will be constructed using $\mathcal{X}_n$, $\mathcal{Y}_n$ and $\bar{\mathcal{Y}}_n$ and $\Delta_\delta = I - D^{-1/2}WD^{-1/2}$ as compared with (4). Let (λ_i, q_i), $i = 1, \ldots, n$ be the associated eigenvalue-eigenvector pairs, and let $U = \text{span}\{q_2, \ldots, q_n\}$. The classifier is set to be the sign of the minimizer u of the functional

$$\mathcal{J}(u) := \frac{1}{2c}\langle u, \Delta_\delta u\rangle - \sum_{j=1}^{J} \log\Big(\Phi(y(j)u(j);\gamma)\Big), \text{ with } c := n\Big(\sum_{i=2}^{n} \lambda_i^{-1}\Big)^{-1},$$

where $\{y(j)\}_{j=1}^{J}$ is the vector of labels and Φ is the cdf of $\mathcal{N}(0,\gamma^2)$. The functional $\mathcal{J}$ can be interpreted as the negative log posterior in a Bayesian setting, as discussed in Bertozzi et al. (2018). Throughout our experiments we set $\gamma = 0.1$.

4.2.1 Two Moons

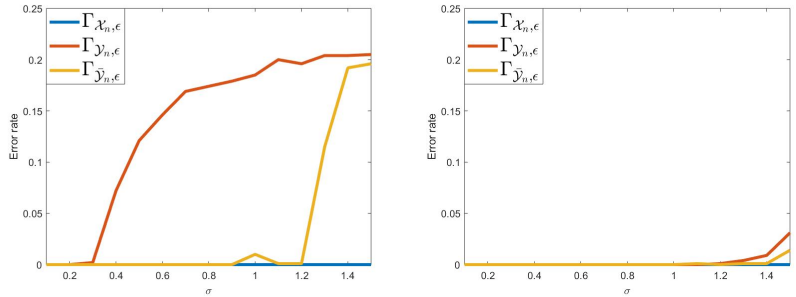
We first study the two moons dataset (Bühler and Hein (2009)), which is generated by sampling points uniformly from two semi-circles of unit radius centered at $(0,0)$ and $(1,0.5)$ and then embedding the dataset in $\mathbb{R}^d$, with $d = 100$. We then perturb the data by adding uniform noise with norm bounded by σ. As before, the noise in the first two dimensions are normal to the semicirlces; the noise is taken to be uniform in the ambient space in the remaining dimensions. In addition to the semi-supervised setting, we also examine the unsupervised case.

We consider $n = 1000$ points 1% of which have labels and we set $K = 10$. As pointed out in Remark 21, we choose the regularization parameter r to be equal to σ. We compare the approximation of distance matrix and classification performance on $\mathcal{X}_n, \mathcal{Y}_n$, and $\bar{\mathcal{Y}}_n$'s,

as in Table 2 and Figure 4. Instead of comparing nodes that are within $\delta_{\mathcal{X}_n}$-distance ε, we consider nodes that are K-nearest neighbors of each other with respect to $\delta_{\mathcal{X}_n}$. As before, the regularized points $\bar{\mathcal{Y}}_n$ approximate the pairwise distances better and moreover, they improve the classification performance. Especially for the fully-connected case, we see that $\bar{\mathcal{Y}}_n$ is able to capture the exact correct labeling as the clean data does for moderate σ's, while the noisy data $\mathcal{Y}_n$ is making mistakes even when σ is as small as 0.3.

	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	$\sigma = 0.7$	$\sigma = 0.9$
$\lVert D_{\mathcal{X}_n} - D_{\mathcal{Y}_n}\rVert_\infty$	0.109	0.344	0.589	0.795	0.996
$\lVert D_{\mathcal{X}_n} - D_{\bar{\mathcal{Y}}_n}\rVert_\infty$	0.064	0.164	0.240	0.372	0.431

Table 2: Entrywise ∞-norm of $D_{\mathcal{X}_n} - D_{\mathcal{Y}_n}$ and $D_{\mathcal{X}_n} - D_{\bar{\mathcal{Y}}_n}$ on two moons for different values of σ.

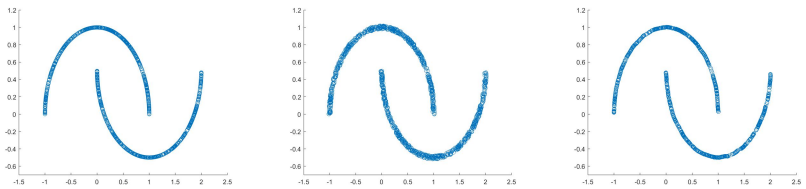


(a) Fully-connected. (b) K-NN variant.

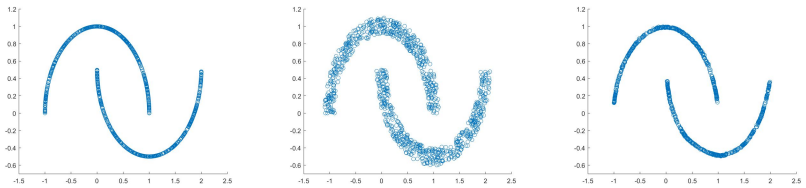
Figure 2: Classification error rates for $\Gamma_{\mathcal{X}_n}$, $\Gamma_{\mathcal{Y}_n}$ and $\Gamma_{\bar{\mathcal{Y}}_n}$ on two moons for different values of σ.

For further understanding, in Figure 3 we plot the first two coordinates of the points in $\mathcal{X}_n$, $\mathcal{Y}_n$ and $\bar{\mathcal{Y}}_n$ for large values of σ. We see that after local regularization, the first two coordinates of $\bar{\mathcal{Y}}_n$ lie almost on the underlying manifold. The denoising effect of local regularization is apparent. Furthermore, we observe that the semicircles for $\bar{\mathcal{Y}}_n$ are "shorter" than those of $\mathcal{X}_n$. In other words, points near the ends are pulled away from the boundaries. Moreover, if one looks carefully at the plots for $\bar{\mathcal{Y}}_n$, points are denser near the top and bottom. This illustrates that local regularization not only reduces noise, but also moves points to regions of high probability. We refer to Chen et al. (2016); Fukunaga and Hostetler (1975) and the references therein for some discussion on mean-shift and mode-seeking type algorithms.

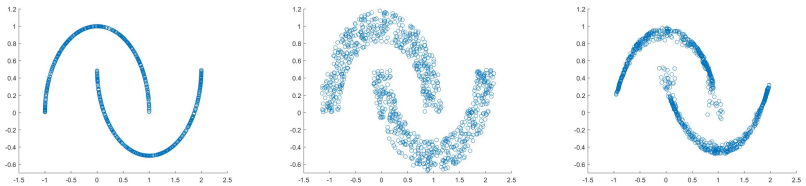
Remark 22 *The two moons dataset is sampled from a manifold with boundaries, and so our theory does not directly apply. However, the numerical results seem to suggest that our theory continues to hold in this setting.*



(a) $\sigma = 0.1$.



(b) $\sigma = 0.5$.



(c) $\sigma = 0.9$.

Figure 3: Visualization of the point clouds $\mathcal{X}_n$, $\mathcal{Y}_n$, and $\bar{\mathcal{Y}}_n$. Each row contains scatter plots of the first two coordinates of the points in the datasets $\mathcal{X}_n$, $\mathcal{Y}_n$, and $\bar{\mathcal{Y}}_n$.

Remark 23 *Additional numerical experiments not shown here suggest that applying local regularization within unsupervised spectral clustering gives qualitatively similar results to those shown in Figure 2 for a semi-supervised setting.*

4.2.2 MNIST

In this subsection we apply local regularization on the MNIST data-set of hand-written digits (LeCun (1998)). Each digit is described by a 784-dimensional vector, but the number of degrees of freedom of the data-generating mechanism is much smaller. For instance, in Hein and Audibert (2005) the authors estimate the intrinsic dimension of the digits 1 from MNIST to be 8. However, unlike in the previous examples, here there is no explicitly avail-

able underlying manifold from which the digits are sampled. Instead of adding additional noise to the dataset, we directly apply local regularization to the digits and show that doing so improves their binary classification. Since the level of noise is unknown choosing the localization parameter r cannot be guided by the theory and in our experiments, we tune it by performing 2-fold cross validation on the label sets. When there are few labels, we repeatedly generate holdout sets and compare the overall error. Due to this practical difficulty of tuning r, we propose two variants of $\Gamma_{\bar{\mathcal{Y}}_n}$ that can serve as alternatives in practice.

We study the classification performance of $\Gamma_{\bar{\mathcal{Y}}_n}$ for different pairs of digits. We consider a semi-supervised learning problem with $n = 1000$ images and $K = 20$ for the K-nearest neighbor variant. Table 3 shows the classification error percentage for four different pairs of digits when 4% of the digits are labeled. Table 4 shows the decrease on the classification error on the pair 4&9 as the percentage of labeled digits is increased.

Fully	3&8	5&8	4&9	7&9	K-NN	3&8	5&8	4&9	7&9
$\Gamma_{\mathcal{Y}_n}$	27.7%	48.0%	48.0%	48.0%	$\Gamma_{\mathcal{Y}_n}$	7.6%	5.5%	13.3%	7.3%
$\Gamma_{\bar{\mathcal{Y}}_n}$	13.4%	17.4%	30.0%	15.3%	$\Gamma_{\bar{\mathcal{Y}}_n}$	6.0%	3.6%	9.6%	5.4%

Table 3: Classification error for different pairs of digits 3&8, 5&8, 4&9, and 7&9.

Fully	4%	8%	12%	16%	K-NN	4%	8%	12%	16%
$\Gamma_{\mathcal{Y}_n}$	48.0%	42.7%	38.8%	29.4%	$\Gamma_{\mathcal{Y}_n}$	13.3%	10.9%	7.6%	5.1%
$\Gamma_{\bar{\mathcal{Y}}_n}$	30.0%	26.1%	21.9%	18.2%	$\Gamma_{\bar{\mathcal{Y}}_n}$	9.6%	6.4%	6.0%	4.5%

Table 4: Classification error for 4&9 with different number of labels.

(a) Threes in MNIST. (b) Eights in MNIST.

Figure 4: Visualization of the regularization effects. The second row is the regularized version of the corresponding image in the first row. While arguably more blurred, the digits in the second row are more homogeneous within each group, making classification easier.

Again the K-NN variant performs much better than the fully-connected graph. As in Table 3, we see that except for the pair 3&8, the classification error for the other three pairs with $\Gamma_{\mathcal{Y}_n}$ is 480: after respecting the 40 labels, the other 960 images are classified as part of the same group. However, after regularization, the classification error is greatly reduced with $\Gamma_{\bar{\mathcal{Y}}_n}$. The same is true when we use the K-NN variant, but the improvement is smaller. Similarly as in Table 4, the improvement for local regularization becomes less dramatic as we go from the fully-connect graph to its K-NN variant and as the number of labels increases. This implies that there is certainly a limit for the improvement that local regularization can provide. Moreover, such improvement is most effective when label information is limited and one has to extract information from the geometry. Our theory and our experiments show that local regularization improves the recovery of geometric information and thereby

boosts the classification performance in that scenario. We present a visualization of the effect of local regularization in Figure 3. The two rows represent the image before and after local regularization respectively. We can see that especially for the eights, many of the images get "fixed" after regularization. Moreover, at a high level, images within each group in the second row look more similar among themselves than those in the first row. Because of this we expect the classification to be better.

Remark 24 *The four chosen pairs of digits are the hardest pairs to classify but local regularization can improve the performance for other pairs too. For unsupervised spectral clustering, local regularization still gives improvement, but using cross validation to choose r is no longer possible.*

4.3 Future Directions

As mentioned above, the practical choice of r can be challenging. We propose two alternatives that may be easier to work with and investigate their competence on the MNIST dataset.

4.3.1 k-NN regularization

This is a natural variant of $\Gamma_{\bar{\mathcal{Y}}_n}$ based on k-nearest neighbor regularization. Instead of specifying a neighborhood of y_i of radius r, we simply regress the data by averaging over its k nearest neighbors. Here k is not necessarily the same as K (the number of neighbors used to construct a similarity graph). Conceptually, choosing k amounts to setting different values of r at different points in such a way that the resulting neighborhoods contain roughly the same number of points. This construction is easier to work with since k is in general easier to tune than r.

4.3.2 Self-tuning regularization

This is a global regularization variant that does not require hyper parameters. Instead of averaging over a neighborhood of radius r, we take a global weighted average of the whole point cloud, where the weights are proportional to the similarities between the y_i. More specifically, we define a new distance in terms of the points $\hat{y}_i$, where

$$\hat{y}_i = \sum_{j=1}^{n} W(i,j) y_j,$$

and $W(i,j)$ is the defined as in (31). We see that points far from y_i have small contribution in the definition of $\hat{y}_i$ and so essentially one ends up summing over points in a neighborhood that is implicitly specified by the similarities. For points close to y_i, the weights are roughly on the same order. Hence $\hat{y}_i$ can be seen approximately as $\bar{y}_i$ plus a small contribution from points that are far from y_i. We expect this construction to behave a little worse than the $\Gamma_{\bar{\mathcal{Y}}_n}$ with optimal r. However, the fact that this construction does not require the tuning of any hyper-parameter makes it an appealing choice. Table 6 compares the classification performance of all graphs mentioned above (with the four different choices of distance function, and the two alternatives to build similarity graphs).

Fully	3&8	5&8	4&9	7&9	K-NN	3&8	5&8	4&9	7&9
$\Gamma_{\mathcal{Y}_n}$	27.7%	48.0%	48.0%	48.0%	$\Gamma_{\mathcal{Y}_n}$	7.6%	5.5%	12.8%	7.3%
$\Gamma_{\bar{\mathcal{Y}}_n}$	13.4%	17.4%	36.9%	15.3%	$\Gamma_{\bar{\mathcal{Y}}_n}$	6.9%	3.6%	9.7%	5.4%
k-NN	11.5%	7.4%	43.1%	18.3%	k-NN	5.3%	5.9%	9.6%	6.1%
self-tuning	16.1%	13.9%	33.4%	26.3%	self-tuning	7.6%	3.1%	8.8%	5.6%

Table 5: Comparison of classification errors with 4% labeled data.

Remark 25 *The idea of using labels to learn r (or k) can be understood as a specific instance of a more general idea: to use labels to better inform the learning of the underlying geometry of a dataset. What is more, one can try to simultaneously learn the geometry of the input space with the learning of the labeling function, instead of looking at these two problems in sequential form. This will be the topic of future research.*

Acknowledgments

The work of NGT and DSA was supported by the NSF Grant DMS-1912818/1912802. The authors are thankful to Facundo Mémoli for enlightening discussions, and two anonymous referees for their valuable feedback. Part of this manuscript was completed when the first author visited the Department of Statistics at the University of Chicago; NGT would like to thank the department for their hospitality.

Appendix A. Estimating r_- and r_+

A.1 Estimating r_-

We want to find values of $t > \frac{r}{2}$ for which for all $v \in T_{x_i}\mathcal{M}$ with $|v| \leq t$, and for all $\eta \in T_{\exp_{x_i}(v)}\mathcal{M}^\perp$ with $|\eta| \leq \sigma$ we have

$$|\exp_{x_i}(v) + \eta - y_i| < r.$$

We will later take the maximum value of t for which this holds and set r_- to be this maximum value.

Let $x = \exp_{x_i}(v)$. First, with the parallel transport map used in the proof of the geometric bias estimates (as in (25)) we can associate a vector $\tilde{\eta} \in T_{x_i}\mathcal{M}^\perp$ to a vector $\eta \in T_x\mathcal{M}^\perp$ with norm less than σ, for which

$$|\eta - \tilde{\eta}| \leq \frac{m}{R}\sigma t.$$

Now,

$$\begin{aligned}
|x + \eta - y_i| &\leq |x - x_i + \hat{\eta} - z_i| + |\eta - \hat{\eta}| \\
&= \left(|x - x_i|^2 + 2\langle x - x_i, \hat{\eta} - z_i\rangle + |\hat{\eta} - z_i|^2|\right)^{1/2} + |\eta - \hat{\eta}| \\
&\leq \left(|x - x_i|^2 + 2\langle x - x_i, \hat{\eta} - z_i\rangle + |\hat{\eta} - z_i|^2|\right)^{1/2} + \frac{m}{R}\sigma t.
\end{aligned}$$

We have

$$|x - x_i| \leq d_{\mathcal{M}}(x, x_i) = |v| \leq t,$$

and also

$$\langle x - x_i, \hat{\eta} - z_i \rangle = \langle x - (x_i + v), \hat{\eta} - z_i \rangle,$$

as it follows from the fact that $\eta, z_i \in T_{x_i}\mathcal{M}^{\perp}$ and $v \in T_{x_i}\mathcal{M}$. Using this, Cauchy-Schwartz, and (20) we conclude that

$$|\langle x - (x_i + v), \hat{\eta} - z_i \rangle| \leq 2\sigma |x - (x_i + v)| \leq 2\sigma \frac{|v|^2}{R},$$

and hence

$$\begin{aligned} |x + \eta - y_i| &\leq \left(t^2 + \frac{4}{R}\sigma t^2 + 4\sigma^2 \right)^{1/2} + \frac{m\sigma t}{R} \\ &= t \left(\sqrt{1 + \frac{4\sigma}{R} + \frac{4\sigma^2}{t^2}} + \frac{m\sigma}{R} \right) \\ &\leq t \left(\sqrt{1 + \frac{4\sigma}{R} + \frac{16\sigma^2}{r^2}} + \frac{m\sigma}{R} \right). \end{aligned}$$

From the above it follows that r_- defined as

$$r_- := r \left(\sqrt{1 + \frac{4\sigma}{R} + \frac{16\sigma^2}{r^2}} + \frac{m\sigma}{R} \right)^{-1}, \tag{32}$$

satisfies the desired properties and moreover

$$r - r_- \leq r \left(1 - \left(\sqrt{1 + \frac{4}{R}\sigma + \frac{16\sigma^2}{r^2}} + \frac{m\sigma}{R} \right)^{-1} \right). \tag{33}$$

A.2 Estimating r_+

To estimate r_+, we need the following lemma proved in García Trillos et al. (2018).

Lemma 26 *Suppose $x, \tilde{x} \in \mathcal{M}$ are such that $|x - \tilde{x}| \leq R/2$. Then*

$$|x - \tilde{x}| \leq d_{\mathcal{M}}(x, \tilde{x}) \leq |x - \tilde{x}| + \frac{8}{R}|x - \tilde{x}|^3. \tag{34}$$

To construct r_+ we find values of t with $2r \geq r + \sigma > t > 0$ such that if $|v| > t$ then

$$|\exp_{x_i}(v) + \sigma\eta - y_i| \geq r$$

for all $\eta \in T_{\exp_{x_i}(v)}\mathcal{M}^{\perp}$ of norm no larger than σ.

As in the construction of r_- we let $x := \exp_{x_i}(v)$. Similar computations give

$$\begin{aligned} |x + \eta - y_i| &\geq |x - x_i + \hat{\eta} - z_i| - |\eta - \hat{\eta}| \\ &= \left(|x - x_i|^2 + 2\langle x - x_i, \hat{\eta} - z_i \rangle + |\hat{\eta} - z_i|^2 | \right)^{1/2} - |\eta - \hat{\eta}| \\ &\geq \left(|x - x_i|^2 + 2\langle x - x_i, \hat{\eta} - z_i \rangle + |\hat{\eta} - z_i|^2 | \right)^{1/2} - \frac{m}{R}\sigma|v| \\ &\geq \left(\left(|v| - \frac{1}{R}|v|^3 \right)^2 - \frac{4}{R}\sigma|v|^2 \right)^{1/2} - \frac{m}{R}\sigma|v| \\ &\geq |v| \left(\sqrt{1 - \frac{2|v|^2}{R} - \frac{4\sigma}{R}} - \frac{m\sigma}{R} \right) \\ &\geq |v| \left(\sqrt{1 - \frac{8r^2}{R} - \frac{4\sigma}{R}} - \frac{m\sigma}{R} \right) \\ &\geq t \left(\sqrt{1 - \frac{8r^2}{R} - \frac{4\sigma}{R}} - \frac{m\sigma}{R} \right), \end{aligned}$$

where in the third inequality we have used (34) to conclude that

$$|\exp_{x_i}(v) - x_i| \geq d_{\mathcal{M}}(x, x_i) - C(d_{\mathcal{M}}(x, x_i))^3 = |v| - C|v|^3.$$

We can then take t to be such that the right hand side of (33) is equal to r. That is, we can take

$$r_+ := r \left(\sqrt{1 - \frac{8r^2}{R} - \frac{4\sigma}{R}} - \frac{m\sigma}{R} \right)^{-1}.$$

From these estimates we see that

$$r_+ - r_- \leq c \left(r^3 + r\sigma + \frac{\sigma^2}{r} \right).$$

References

E. Aamari and C. Levrard. Nonasymptotic rates for manifold, tangent space and curvature estimation. *The Annals of Statistics*, 47(1):177–204, 2019.

E. Aamari, J. Kim, F Chazal, B Michel, A Rinaldo, L Wasserman, et al. Estimating the reach of a manifold. *Electronic Journal of Statistics*, 13(1):1359–1399, 2019.

S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017.

M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

A. L. Bertozzi, X. Luo, A. M. Stuart, and K. C. Zygalakis. Uncertainty quantification in graph-based classification of high dimensional data. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):568–595, 2018.

T. Bühler and M. Hein. Spectral clustering based on the graph p-Laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 81–88. ACM, 2009.

D. Burago, S. Ivanov, and Y. Kurylev. A graph discretization of the Laplace-Beltrami operator. *J. Spectr. Theory*, 4:675–714, 2014.

Y.-C. Chen, C. R. Genovese, L. Wasserman, et al. A comprehensive approach to mode clustering. *Electronic Journal of Statistics*, 10(1):210–241, 2016.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

M. P. do Carmo. *Riemannian geometry*. Mathematics: Theory & Applications. Birkhäuser Boston, Inc., Boston, MA, 1992. ISBN 0-8176-3490-8. doi: 10.1007/978-1-4757-2201-7. URL `https://doi.org/10.1007/978-1-4757-2201-7`. Translated from the second Portuguese edition by Francis Flaherty.

H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93 (3):418–491, 1959.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1): 32–40, 1975.

N. Garcia Trillos and D. Sanz-Alonso. Continuum limits of posteriors in graph bayesian inverse problems. *SIAM Journal on Mathematical Analysis*, 50(4):4020–4040, 2018.

N. García Trillos and D. Slepčev. Continuum limit of total variation on point clouds. *Archive for rational mechanics and analysis*, 220(1):193–241, 2016.

N. Garcia Trillos, Z. Kaplan, T. Samakhoana, and D. Sanz-Alonso. On the consistency of graph-based Bayesian learning and the scalability of sampling algorithms. *arXiv preprint arXiv:1710.07702*, 2017.

N. García Trillos, M. Gerlach, M. Hein, and D. Slepčev. Spectral convergence of empirical graph Laplacians. *Preprinr*, 2018.

C. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Minimax manifold estimation. *Journal of machine learning research*, 13(May):1263–1291, 2012.

A. Haddad, D. Kushnir, and R. R. Coifman. Texture separation via a reference set. *Applied and Computational Harmonic Analysis*, 36(2):335–347, 2014.

M. Hein and J. Audibert. Intrinsic dimensionality estimation of submanifolds in r d. In *Proceedings of the 22nd international conference on Machine learning*, pages 289–296. ACM, 2005.

M. Hein and M. Maier. Manifold denoising. In *Advances in Neural Information Processing Systems 19*, pages 561–568, Cambridge, MA, USA, September 2007. Max-Planck-Gesellschaft, MIT Press.

Y. LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

E. H. Lieb, M. Loss, et al. Graduate studies in mathematics. *Analysis*, 14, 2001.

A. V. Little and L. Maggioni, M.and Rosasco. Multiscale geometric methods for data sets i: Multiscale svd, noise and curvature. *Applied and Computational Harmonic Analysis*, 43(3):504–567, 2017.

F. Mémoli, Z. T. Smith, and Z. Wan. The wasserstein transform. *CoRR*, abs/1810.07793, 2018. URL `http://arxiv.org/abs/1810.07793`.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

P. J. Olver. *Introduction to Partial Differential Equations*. Springer.

I. Pinelis. An approach to inequalities for the distributions of infinite-dimensional martingales. In *Probability in Banach spaces, 8 (Brunswick, ME, 1991)*, volume 30 of *Progr. Probab.*, pages 128–134. Birkhäuser Boston, Boston, MA, 1992.

J. Shi and J. Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.

A. Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006.

D. A Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2-3):284–305, 2007.

J. W. Tukey and P. A. Tukey. Computer graphics and exploratory data analysis: An introduction. *The Collected Works of John W. Tukey: Graphics: 1965-1985*, 5:419, 1988.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

U. Von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

W. Wang and M. A. Carreira-Perpinán. Manifold blurring mean shift algorithms for manifold denoising. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1759–1766. IEEE, 2010.

J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *arXiv preprint arXiv:1707.00087*, 2017.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2005.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Joint Pattern Recognition Symposium*, pages 361–368. Springer, 2005.

X. Zhu. Semi-supervised learning literature survey. 2005.

Journal of Machine Learning Research 20 (2019) 1-68 Submitted 3/17; Published 9/19

Multiclass Boosting: Margins, Codewords, Losses, and Algorithms

Mohammad Saberian SABERIAN@UCSD.EDU
Statistical Visual Computing Laboratory,
University of California, San Diego
La Jolla, CA 92039, USA

Nuno Vasconcelos NUNO@UCSD.EDU
Statistical Visual Computing Laboratory,
University of California, San Diego
La Jolla, CA 92039, USA

Editor: Koby Crammer

Abstract

The problem of multiclass boosting is considered. A new formulation is presented, combining multi-dimensional predictors, multi-dimensional real-valued codewords, and proper multiclass margin loss functions. This leads to a number of contributions, such as maximum capacity codeword sets, a family of proper and margin enforcing losses, denoted as $\gamma - \phi$ losses, and two new multiclass boosting algorithms. These are descent procedures on the functional space spanned by a set of weak learners. The first, CD-MCBoost, is a coordinate descent procedure that updates one predictor component at a time. The second, GD-MCBoost, a gradient descent procedure that updates all components jointly. Both MCBoost algorithms are defined with respect to a $\gamma - \phi$ loss and can reduce to classical boosting procedures (such as AdaBoost and LogitBoost) for binary problems. Beyond the algorithms themselves, the proposed formulation enables a unified treatment of many previous multiclass boosting algorithms. This is used to show that the latter implement different combinations of optimization strategy, codewords, weak learners, and loss function, highlighting some of their deficiencies. It is shown that no previous method matches the support of MCBoost for real codewords of maximum capacity, a proper margin-enforcing loss function, and any family of multidimensional predictors and weak learners. Experimental results confirm the superiority of MCBoost, showing that the two proposed MCBoost algorithms outperform comparable prior methods on a number of datasets.

Keywords: Boosting, Multiclass Boosting, Multiclass Classification, Margin Maximization, Loss Function.

1. Introduction

Boosting is a popular approach for classifier design. It is a simple and effective procedure to learn strong decision rules by combination of weak learners. However, most boosting algorithms were designed primarily for binary classification. In many cases, the extension to M-ary problems (of $M > 2$) is not straightforward. There are many ways to interpret boosting (Schapire and Freund, 2012) and these have been used to justify different multiclass extensions. In this work, we consider the view of boosting as a method for empirical risk minimization, using some optimization procedure (usually gradient descent) on the functional space spanned by a set of weak learners (Friedman et al., 1998; Mason et al., 2000). Under this view, the effectiveness of a boosting algorithm is determined by the underlying choices of optimization method, weak learner pool, and risk function. The latter can be further decomposed into a choice of a loss function and an encoding of class labels. All of these are fairly well understood in the binary setting. In some cases, such as the selection of codeword labels, there is virtually universal agreement in the literature, where most algorithms use -1 as the label of "negative" examples and $+1$ for "positives". On others, there is wide but not universal agreement. For example, while it can be shown that most boosting algorithms implement some form of functional gradient descent (Mason et al., 2000; Zhang, 2004; Buja et al., 2006; Masnadi-Shirazi and Vasconcelos, 2008), there have also been proposals to use second order optimization methods, based on Taylor series expansions (Saberian et al., 2010) and Newton's method (Friedman et al., 1998). The remaining aspects are less consensual. Most algorithms differ in terms of the loss function that defines the risk for which they are optimal. Several loss properties, such as encouraging large margins (Vapnik, 1998) and allowing the recovery of class probabilities (Zhang, 2004; Buja et al., 2006; Mease and Wyner, 2008; Masnadi-Shirazi and Vasconcelos, 2008; Reid and Williamson, 2010) have been identified as important, and shown to hold for a large family of functions (Masnadi-Shirazi and Vasconcelos, 2015). Some members of this family, such as the exponential loss of AdaBoost (Freund and Schapire, 1997), the logit loss of LogitBoost (Friedman et al., 1998), the hinge loss of support vector machines (Vapnik, 1998) or the Savage loss of SavageBoost (Masnadi-Shirazi and Vasconcelos, 2008), have been widely used in practice. Finally, weak learners tend to vary with the application. Popular choices include weak classifiers, such as the decision stumps or shallow decision trees commonly used in computer vision (Viola and Jones, 2004; Dollar et al., 2012), and regression models (Friedman et al., 1998).

To date, there have been no comprehensive efforts to extend this understanding to the M-ary case. While boosting components such as the optimization strategy generalize in a straightforward manner, others do not. In particular, many label encodings and loss functions are possible. There is limited understanding of what constitutes a good encoding or when a loss function is proper (allows the recovery of class conditional probabilities). As we will see later on, some popular choices for these components are quite suboptimal. In fact, for multiclass boosting, it is not even easy to guarantee that a set of multiclass weak learners can be boosted. For binary classification, it is well known this is possible as long as a weak learner with "less than 50% error" can be found in all bosting iterations. Since this only requires a weak learner marginaly better than random guessing, the condition can be met trivially. However, for M-ary classification, the accuracy of a random classifier is only $\frac{100}{M}\%$. In this case, straightforward extensions of binary boosting algorithms that require multiclass "weak" learners with "less than 50% error", such as the well known AdaBoost.M1 algorithm (Freund and Schapire, 1996), are too difficult to implement. In practice, they tend to stop prematurely and fail to produce strong ensemble decision rules. Due to all this, many of the existing approaches to multiclass boosting involve some form of reduction of the M-class problem to a collection of binary problems. While the most popular are the well known "one vs all" (Nilsson, 1965) and "all pairs" (Hastie and Tibshirani, 1998) classification architectures, many boosting algorithms based on this type of representation have been proposed. A smaller number of true multiclass methods, such as AdaBoost.M1(Freund and Schapire, 1996; Eibl and Schapire, 2005), SAMME (Zhu et al., 2009), and AdaBoost.MM (Mukherjee and Schapire, 2013) have also been proposed. These boost multiclass weak learners, usually decision trees. The limitations of these methods are discussed in Section 2.

©2019 Mohammad Saberian and Nuno Vasconcelos.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-137.html.

In this work, we study multiclass boosting with the goal of an integrated understanding of the roles of the optimization strategy, label codewords, weak learners, and multiclass risk. This leads to a new formulation of the problem based on 1) multi-dimensional predictors, 2) multi-dimensional real valued codewords, and 3) proper multiclass margin loss functions. We start by studying the role of the label encoding, showing that the selected codewords impose an upper bound on the maximum margin achievable by any predictor. This is denoted the margin capacity bound. We then define a family of losses, denoted $\gamma - \phi$ losses, which extend the classical margin losses used by binary boosting algorithms. These losses connect the classification margin to a set of dot-products between a multidimensional predictor and a codeword set, enabling the formulation of multiclass boosting as a margin maximization problem in multidimensional functional space. This objective is formulated through an empirical risk that combines a $\gamma - \phi$ loss and a margin capacity codeword set. Two algorithms are then derived to solve this optimization. The first, denoted CD-MCBoost, implements a functional coordinate descent procedure. CD-MCBoost supports any type of weak learners, updating one component of the predictor per boosting iteration. This method has some similarities to binary reduction procedures but 1) uses real-valued codewords and 2) learns all predictor components jointly. The second, denoted GD-MCBoost, implements functional gradient descent, in a space of multidimensional weak learners, updating all predictor components simultaneously. Both MCBoost algorithms reduce to classical boosting algorithms (such as AdaBoost or LogitBoost) for binary problems, depending on the choice of $\gamma - \phi$ loss. They are also shown to exhibit classical boosting properties, such as seeking the weak learner of maximum margin on a reweighted training sample at each iteration and well defined boostability conditions. These properties are, however, shown to hold more generally, as would be expected of the multiclass setting. With respect to weights, MCBoost emphasizes not only the most difficult examples but also the most difficult classes, at each boosting iteration. With regards to boostability, MCBoost is shown to boost any set of weak learners with better than multiclass chance performance, i.e. less than $M/100\%$ error.

In the remainder of the paper, we consider the design of good codeword sets and $\gamma - \phi$ loss functions for MCBoost algorithms. We start by considering the design of a set of codewords of maximum capacity, for any predictor dimension d. We derive necessary and sufficient conditions for a codeword set to achieve this bound and show that such a codeword set is guaranteed to exist whenever $d > M - 1$. A procedure to generate the associated codewords is then presented. We next consider the case of low-dimensional predictors, with $d \leq M - 1$. We show that, while there are no guarantees of meeting the capacity bound in this case, it is possible to generate codeword sets that are optimal in a related sense, the max-min codeword distance. A procedure to generate such codeword sets is again provided. We next consider the interplay between codewords and risk, by studying several properties of $\gamma - \phi$ losses. In particular, we derive conditions under which these losses are margin enforcing and show that they are proper under mild conditions on the codeword set. These conditions are shown to hold for all codeword sets that achieve the margin capacity bound. By relating $\gamma - \phi$ losses to the binary margin losses of classical boosting algorithms, we then derive extensions of the latter to the multiclass setting. This leads to natural multiclass extensions of algorithms like AdaBoost (Freund and Schapire, 1997), LogitBoost (Friedman et al., 1998), and SavageBoost (Masnadi-Shirazi and Vasconcelos, 2008). The theoretical framework now introduced is finally used to place previous multiclass boosting algorithms in a common footing. This exercise shows that all algorithms implement different combinations of optimization strategy, codewords, weak learners, and loss function. It also highlights some of the deficiencies of these

algorithms, explaining why they fail under some settings. In particular, it is shown that no previous method matches the support of MCBoost for real codewords of maximum capacity, a proper margin-enforcing loss function, multidimensional predictors and any class of weak learners. Experimental results confirm the superiority of MCBoost, showing that the two proposed MCBoost algorithms outperform comparable prior methods on a number of datasets.

The paper is organized as follows. In section 2, we briefly review prior work on multiclass boosting. The foundations of the proposed formulation are introduced in Section 3, where we propose a set of multiclass definitions for class labels, predictor, and margin. The MCBoost algorithms are derived in Section 4, where we also discuss properties such as weighting mechanisms and weak learners. The problem of optimal codeword design is then discussed in Section 5, where we introduce the notions of margin capacity and derive necessary and sufficient conditions for maximum capacity and max-min distance codeword sets. Section 6 is devoted to weak learners, analyzing issues such as boostability or the role of classification vs. real valued learners. Section 7 discusses various properties of interest for $\gamma - \phi$ losses, and introduces a number of losses that meet these properties, leading to the multiclass extensions of various classical boosting algorithms. These are compared to previous multiclass boosting algorithms in Section 8, where existing methods are studied in light of the proposed boosting framework. Finally, experimental results are discussed in Section 9 and some conclusions drawn in Section 10.

2. Related work

In this section, we briefly review previous work in multiclass boosting.

2.1. Origins

The problem of multiclass classification has attracted significant attention since the early days of machine learning. The first, and still popular, method for designing an M-class classifier is to learn M binary classifiers, each separating one class form the remaining (Nilsson, 1965). At the test time, a prediction score is computed by each binary classifier and the class of highest score is selected. This is known as "one-vs-all" (OVA) classification. Later, (Sejnowski and Rosenberg, 1987) extended the idea by assigning a binary string of length l to each class and learning l binary classifiers to predict the bits of those strings. While this is similar to OVA, the class selected at test time is that whose string is closest, in the Hamming distance sense, to that recovered by the binary classifiers. This was, to the best of our knowledge, the first attempt to represent classes by multidimensional codewords (binary strings in this case). (Dietterich and Bakiri, 1995) improved this idea by introducing an error correcting output code (ECOC). This consists of using techniques from the error correction literature to design a codeword set resistant to errors. It made it possible to recover the true class even if a few of the binary classifiers produced erroneous bits. (Hastie and Tibshirani, 1998) suggested an alternative approach, by designing $\frac{M(M-1)}{2}$ classifiers to discriminate between all pairs of classes. At test time, a class was selected through a vote among these classifiers. (Allwein et al., 2001) unified all these binary classification methods, showing that they all reduce the multiclass problem into binary sub-problems, for a specific choice of coding matrix.

2.2. Reductions to binary

Upon the introduction of AdaBoost (Freund and Schapire, 1997), there was a substantial effort to extend this simple and effective binary classification algorithm to multiclass problems. Like the methods above, many of these approaches are based on a reduction to a collection of binary classification problems. These can be grouped into three main sub-classes.

The first subclass follows the ECOC approach of (Dietterich and Bakiri, 1995). (Schapire, 1997) extended this approach by combining it with a pseudo-loss previously introduced to derive AdaBoost.M2 (Freund and Schapire, 1996). The combination of the two components, through the AdaBoost.OC algorithm, enabled the joint learning of the binary sub-classifiers. This work also proposed using binary random codes and codes designed to have the best error correction performance, using "max-cut" algorithms. A number of algorithms, including AdaBoost.ECC (Guruswami and Sahai, 1999; Allwein et al., 2001), AdaBoost.SECC (Sun et al., 2005), AdaBoost.ERP (Li), AdaBoost.SIP (Zhang et al., 2009) and HingeBoost (Gao and Koller, 2011), were then proposed to generalize or improve the combination of boosting and error correction. For example, (Guruswami and Sahai, 1999) modified the pseudo-loss of AdaBoost.OC and proposed AdaBoost.ECC, which was shown to have better generalization guarantees and performance. (Allwein et al., 2001) studied the impact of the codeword distance function, proposing a loss-based distance to replace the Hamming distance, during both training and classification. (Sun et al., 2005) connected these methods to the margin framework, showing that AdaBoost.OC and AdaBoost.ECC were in fact maximizing multiclass definitions of the margin. The problem of finding the optimal binary coding matrix was studied in (Li; Zhang et al., 2009; Gao and Koller, 2011), which proposed optimization methods to find a good set of codes for each boosting iteration. However, (Crammer and Singer, 2002b) showed that the problem of determining the optimal binary coding matrix is NP-hard, and suggested the use of real-valued codes. In general, the performance of these algorithms depends on two factors: 1) the error correction performance of the coding matrix and the weighting algorithms used to train the binary classifiers. In practice, the optimization of these two factors often requires extensive computation, for both training and classification. This is not surprising since, as pointed out by (Crammer and Singer, 2002b), the optimization problem is NP-hard.

The second subclass of binary reductions originated with AdaBoost.M2 (Freund and Schapire, 1997, 1996). This algorithm couples a class identifier $c \in \mathcal{C}$ with the example $x \in \mathcal{X}$, learning a real valued predictor $f : \mathcal{X} \times \mathcal{C} \to \mathbb{R}$ in the product space of examples and class labels. This predictor produces a score for each example x and class c. It is learned by minimizing a pseudo-loss function defined over all pairs of examples and incorrect labels, so as to penalize classification errors. At the test time, f is evaluated for all pairs of example x and classes c. The class of largest score is finally selected. (Schapire and Singer, 1999) used this approach to extend AdaBoost.M2 into AdaBoost.MR, which addressed multi-label problems. The procedure was also improved by the introduction of AdaBoost.MH, whose weights are updated using the Hamming loss. More recently, (Kuznetsov et al., 2014) introduced new upper bounds and improved algorithms for this class of approaches. As shown by (Friedman et al., 1998), coupling examples with class labels effectively converts the multiclass problem into a binary problem. This is, however, an extremely non-linear problem, since very similar examples, namely all (x, c) of common x, have very different scores (depending on c). In result, these binary reductions require weak learners of high discriminant power, usually deep decision trees. This makes the boosting algorithm prone to over-fitting.

The final subclass of binary reductions originated with the multiclass LogitBoost method of (Friedman et al., 1998). This is based on the statistical view of Boosting as gradient descent in functional space and learns additive logistic regression models for each class, using binary LogitBoost. While these regressors are binary predictors, the key difference to OVA is that they are learned jointly, and must add up to zero. (Huang et al., 2007) extended this framework to a multiclass version of GentleBoost, proposing the GAMBLE algorithm. This was further extended for any Fisher consistent loss function by the AdaBoost.ML algorithm of (Zou et al., 2008). However, it has been shown that scores of the binary regression classifiers do not represent true class probabilities (Mease and Wyner, 2008).

2.3. Boosting multiclass base learners

In contrast to the large literature in binary reductions of the multiclass boosting problem, there have been relatively few attempts to produce true multiclass boosting algorithms. The earliest such attempt is AdaBoost.M1 (Freund and Schapire, 1997, 1996), which directly extends AdaBoost to the multiclass setting. However, by maintaining the AdaBoost boostability requirement of "less than 50% error weak learners," this method has the limitations discussed in Section 1. A few attempts have been made to relax these weak learner requirements, such as the works of (Eibl and Schapire, 2005) and (Mukherjee and Schapire, 2013). The latter provided the most extensive treatment, introducing a game theoretic analysis of the necessary conditions for boostable base learners, proposing a less strict base learner selection criterion. However, the optimal criteria for selection of base learners is still an open problem. Alternatively, (Zhu et al., 2009) devoted more attention to the loss function, attacking the problem through the definition of an exponential loss for multiclass classification in multi-dimensional functional space. They then proposed a gradient descent procedure for this optimization, which was denoted as SAMME boosting. While this has some similarity to the framework now introduced, we show in Section 8 that the exponential loss function is not margin enforcing. Hence SAMME is not a maximum margin algorithm.

3. Multiclass boosting

In this work, we seek multiclass boosting algorithms that do not rely on binary reductions. We start by reviewing the fundamental ideas behind the classical use of boosting for the design of *binary* classifiers, and then extend these ideas to the multiclass setting.

3.1. Binary classification

A binary classifier, $F(x)$, implements a *decision rule* that maps examples $x \in \mathcal{X}$ to classes $c \in \{1, 2\}$. The classifier is optimal when this decision rule minimizes some *classification risk*. A classical risk is the probability of classification error, which is minimized by the Bayes decision rule

$$F(x) = \arg \min_{c \in \{1,2\}} P_{C|X}(c|x). \tag{1}$$

This rule is not easy to implement, due to the difficulty of estimating the probabilities $P_{C|X}(c|x)$. Large margin methods, such as boosting, avoid this difficulty by adopting alternative risks. They

implement the classifier as

$$F(x) = \begin{cases} 1 & \text{if} \quad f^*(x) < 0 \\ 2 & \text{if} \quad f^*(x) > 0. \end{cases} \tag{2}$$

where $f^*(x) : \mathcal{X} \to \mathbb{R}$ is the *continuous valued predictor*

$$f^*(x) = \arg\min_f R_L(f) \tag{3}$$

that minimizes the risk

$$R_L(f) = E_{X,C}\{L[y^c, f(x)]\} \tag{4}$$

defined by a *loss function* $L[.,.]$ and a set of *class labels* y^c, where y^c is the label of class $c \in \{1, 2\}$. The loss $L[.,.]$ is Bayes consistent if the minimization of (4) results in the Bayes decision rule, i.e. (1) and (2) are equivalent.

To learn the optimal classifier, the risk of (4) is estimated by the empirical risk

$$\overline{R}_L(f) = \frac{1}{n}\sum_{i=1}^{n} L[y^{c_i}, f(x_i)] \tag{5}$$

over a training sample $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^n$. Large margin methods use the *labels* $y^1 = -1$ and $y^2 = 1$ and a Bayes consistent loss function that only depends on the *classification margin* $y^c f(x)$, i.e.

$$L[y^c, f(x)] = L[y^c f(x)]. \tag{6}$$

This guarantees that the classifier has good generalization for finite training samples (Vapnik, 1998). Boosting learns the optimal predictor $f^*(x) : \mathcal{X} \to \mathbb{R}$ as the solution of

$$\begin{cases} \min_{f(x)} & \overline{R}_L(f) \\ s.t & f(x) \in span(\mathcal{H}) \end{cases} \tag{7}$$

where $\overline{R}_L(f)$ is the empirical risk of (5), and $\mathcal{H} = \{h_1(x), \ldots, h_r(x)\}$, a set of weak learners $h_i(x) : \mathcal{X} \to \mathbb{R}$. The optimization is carried out by gradient descent in the functional space $span(\mathcal{H})$ of linear combinations of $h_i(x)$ (Friedman et al., 1998; Mason et al., 2000; Saberian et al., 2010). The extension of binary boosting to the multiclass setting requires multiclass definitions of class labels, predictor, margin, decision rule, loss function and risk minimization procedure.

3.2. Multiclass extensions

The definition of the classification labels as $y^c = \pm 1$ plays a significant role in the binary formulation. One of the difficulties of the multiclass extension is that these labels do not have an obvious generalization. For M-ary classification, $c \in \{1, \ldots, M\}$, each class c must be mapped into a distinct class label $y^c \in \mathcal{Y} = \{y^1, \ldots, y^M\}$. This label can be thought of as a *codeword* that identifies the class. In the binary case, the predictor is a real valued function, i.e. $f(x) \in \mathbb{R}$, and the codewords ± 1 are the two directions on the line. To generalize these concepts to the multiclass setting, we introduce a multi-dimensional predictor $f(x) \in \mathbb{R}^d$ and codewords y^k which are directions in this space

$$y^c \in \mathbb{R}^d, \quad \|y^c\| = 1. \tag{8}$$

Given a multi-dimensional predictor and and a set of multi-dimensional codewords[1], the multiclass margin is defined as follows.

Definition 1 *Let $y^k \in \mathbb{R}^d, k \in \{1, \ldots, M\}$, be the set of codewords of an M-ary classifier of predictor $f : \mathcal{X} \to \mathbb{R}^d$. The margin of example x with respect to class k is*

$$\begin{aligned} \mathcal{M}(y^k, f(x)) &= \min_{l \neq k} \left[u^k - u^l\right] & (9) \\ &= u^k - \max_{l \neq k} u^l, & (10) \end{aligned}$$

where

$$u^k = \frac{1}{2}\left\langle y^k, f(x)\right\rangle, \tag{11}$$

is the component of $f(x)$ along codeword y^k and $< .,. >$ is the Euclidean dot-product. The quantity

$$\left[u^k - u^l\right] = \frac{1}{2}\left\langle y^k - y^l, f(x)\right\rangle \tag{12}$$

is the lth margin component of $f(x)$ with respect to class k.

This definition is closely related to previous definitions of multiclass margin in the literature. For example, it generalizes that of (Guermeur, 2007), where the codewords y^k are restricted to binary vectors in the canonical basis of $\mathbb{R}^d$, and is a special case of that of (Allwein et al., 2001), where the dot products $\langle y^k, f(x)\rangle$ are replaced by a generic function of f, x, and k. Furthermore, when $M = 2$ and $y^1 = -y^2 = 1$,

$$\mathcal{M}(y^k, f(x)) = \frac{1}{2}[y^k f(x) - \max_{l \neq k} y^l f(x)] = \frac{1}{2}[y^k f(x) + y^k f(x)] = y^k f(x), \tag{13}$$

and (9) is identical to the classic definition of binary margin. Similarly to the binary case, it is possible to define the margin of a predictor for a particular dataset, as follows.

Definition 2 *The margin of a predictor $f(.)$ with respect to a set of codewords $\mathcal{Y} = \{y^1, \ldots, y^M\}$ and examples $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^n$ is*

$$\mathcal{M}_p(\mathcal{D}, \mathcal{Y}, f) = \min_{(x_i, y^{c_i}) \in \mathcal{D}} \mathcal{M}(y^{c_i}, f(x_i)). \tag{14}$$

This can be seen as a measure of the distance between the classification boundary and the point closest to it.

To extent the binary decision rule of (2) to the multiclass case, we start by noting that, for a binary classifier with $y^1 = 1$ and $y^2 = -1$, it can be written as

$$F(x) = \arg\max_{k=\{1,2\}} y^k f^*(x), \tag{15}$$

i.e. the classifier simply chooses the class of largest margin for example x. This has the following straightforward extension.

1. At this point, we assume that a good set of codeword exists and is known. In Section 5 we will discuss how the selection of codewords affects learning performance and procedures for determining the optimal ones.

Definition 3 *Consider an M-ary classification problem with codewords $y^k \in \mathbb{R}^d, k \in \{1, \ldots, M\}$. A maximum margin classifier of predictor $f : \mathcal{X} \to \mathbb{R}^d$ implements the decision rule*

$$F(x) = \arg \max_{k \in \{1,\ldots,M\}} \mathcal{M}(y^k, f(x)). \tag{16}$$

The following result shows that this is equivalent to selecting the class along whose codeword $f(x)$ has the largest component.

Lemma 1 *The decision rule of the maximum margin classifier of (16) is equivalent to*

$$F(x) = \arg \max_{k \in \{1,\ldots,M\}} \left\langle y^k, f(x) \right\rangle. \tag{17}$$

Proof See Appendix A.1. ■

A corollary of this result is that, as is the case for binary classification, an example x of class c is correctly classified by the max margin classifier if and if and only if the example margin of x with respect to class c is positive.

Corollary 1 *Let c be the class of example x and $f(x)$ the predictor of a maximum margin classifier $F(x)$. Then $F(x) = c$ if and only if*

$$\mathcal{M}(y^c, f(x)) > 0. \tag{18}$$

Proof See Appendix A.2 ■

Finally, a maximum margin classifier classifies all examples in a dataset $\mathcal{D}$ correctly if and only if its predictor margin with respect to $\mathcal{D}$ is positive.

Corollary 2 *Let $f(.)$ be the predictor of a maximum margin classifier $F(x)$ and $\mathcal{D}$ a set of examples (x_i, c_i). $f(.)$ classifies all $x_i \in \mathcal{D}$ correctly if and only if*

$$\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y}) > 0. \tag{19}$$

Proof See Appendix A.3. ■

These corollaries extend the equivalent properties of binary large margin predictors to the multiclass case.

4. Multiclass Boosting Algorithms

In this section we introduce two multiclass boosting algorithms. Both are gradient descent procedures for the minimization of a multiclass empirical risk in the functional space of linear combinations of weak learners and can be seen as generalizations of GradientBoost (Mason et al., 2000). We start by extending the definitions of risk and margin loss to the multiclass setting.

Table 1: Binary losses and corresponding γ and ϕ functions.

Name	$\xi(v)$	$\gamma(v)$	$\phi(v)$
Exponential	$\exp(-v)$	v	$\exp(-v)$
Logistic	$\log(1+e^{-2v})$	$\log(1+v)$	$\exp(-2v)$
Savage	$\frac{1}{(1+e^{2v})^2}$	$\left(\frac{v}{1+v}\right)^2$	$\exp(-2v)$

4.1. $\gamma - \phi$ losses

Given a set of codewords $\mathcal{Y}$, the optimal multiclass predictor $f^*(x)$ minimizes the classification risk

$$R_{L_M}(f) = E_{X,C}\{L_M[y^c, f(x)]\}, \tag{20}$$

where c is the class of example x, y^c its codeword and $L_M[y^c, f(x)]$ the loss of prediction $f(x)$. For classifier design, this is approximated by the empirical estimate

$$\overline{R}_{L_M}(f) = \frac{1}{n} \sum_{i=1}^{n} L_M[y^{c_i}, f(x_i)], \tag{21}$$

derived from a training sample $\mathcal{D} = (x_i, c_i)_{i=1}^n$. When the minimization of (21) encourages predictors for which the margin of (14) is large, $L_M[.,.]$ is denoted a *margin loss*. This property guarantees that the optimal predictor has good generalization beyond the training set.

For binary classification, margin losses are monotonically decreasing functions of the margin $y^c f$. The natural multiclass extension would be to consider decreasing functions of the margin, now defined in (9), i.e.

$$L_M[y^c, f(x)] = \chi\left(\min_{l \neq c}(u^c - u^l)\right), \tag{22}$$

with u^k as in (11), for some monotonically decreasing function χ. This, however, is a non-differentiable function of the predictor f. We avoid this difficulty by considering the set of $\gamma - \phi$ losses.

Definition 4 *Let $\mathcal{Y} = \{y^1, \ldots, y^M\} \in \mathbb{R}^d$ be a set of codewords, $f(x) : \mathcal{X} \to \mathbb{R}^d$ a predictor. A $\gamma - \phi$ loss is defined as*

$$L_M^{\gamma-\phi}[y^c, f(x)] = \gamma\left(\sum_{l=1, l\neq c}^{M} \phi\left(u^c - u^l\right)\right), \tag{23}$$

where $\phi : \mathbb{R} \to \mathbb{R}^+$ and $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ are strictly positive and u^j is defined in (11).

We leave a theoretical discussion of the properties of $\gamma - \phi$ losses to Section 7. For now, we simply point out that this set includes a large family of losses. In fact, for binary classification with the classic labels $y^1 = -y^2 = -1$, $u^1 = -u^2 = -\frac{1}{2}f(x)$ and $L_M^{\gamma-\phi}[y^c, f(x)]$ reduces to

$$\begin{aligned} L_2^{\gamma-\phi}[y^c, f(x)] &= \gamma\left(\sum_{k=1|k\neq c}^{2} \phi\left[u^c - u^k\right]\right) = \gamma\left(\phi\left[\frac{(y^c - (-y^c))f}{2}\right]\right) \\ &= \gamma\left(\phi(y^c f)\right) = \xi(y^c f), \end{aligned} \tag{24}$$

Algorithm 1 CD-MCBoost and GD-MCBoost

Input: Number of classes M, dimension d, codeword set $\mathcal{Y} = \{y^1, \ldots, y^M\} \in \mathbb{R}^d$, boosting iterations N and dataset $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^n$ of examples x_i and class labels $c_i \in \{1, \ldots, M\}$.
Initialization: set $t = 0$, and $f^t = 0 \in \mathbb{R}^d$

CD-MCBoost
while $t < N$ **do**
  Compute w_i with (29)
  for $j = 1$ to d **do**
    Find $\hat{g}_j(x)$, $\hat{\alpha}_j$ using (39) and (40)
  end for
  Set $j^* = \arg\min_j \overline{R}[f^t(x) + \hat{\alpha}_j \hat{g}(x)\mathbf{1}_j]$
  Update $f^{t+1}(x) = f^t(x) + \hat{\alpha}_{j^*}\hat{g}_{j^*}(x)\mathbf{1}_j$
  $t = t + 1$
end while

GD-MCBoost
while $t < N$ **do**
  Compute w_i with (29)
  Find $g^*(x)$, α^* using (32) and (33)
  Update $f^{t+1}(x) = f^t(x) + \alpha^* g^*(x)$
  $t = t + 1$
end while

Output: decision rule: $F(x) = \arg\max_{y^k} \left\langle f^N(x), y^k \right\rangle$

where $\xi = \gamma \circ \phi$ is a composite function. Table 1 shows that the exponential loss of AdaBoost (Freund and Schapire, 1997), the logistic loss of LogitBoost (Friedman et al., 1998), and the Savage loss of (Masnadi-Shirazi and Vasconcelos, 2008) can all be interpreted as $\gamma - \phi$ losses with different choices of γ and ϕ. It should be noted that these decompositions are not unique. In all cases, ξ could be equally decomposed into $\gamma(v) = v$ and $\phi = \xi$. The key property is that ξ is a monotonically decreasing function. In Section 7 we show that $\gamma - \phi$ losses are margin enforcing whenever γ is strictly increasing and ϕ is strictly decreasing. This implies that $\xi = \gamma \circ \phi$ is decreasing.

4.2. Gradient descent

The first boosting algorithm is a gradient descent procedure to seek the optimal predictor $f^*(x) = [f_1^*(x), \ldots, f_d^*(x)]$ of the optimization problem

$$\begin{cases} \min_{f(x)} & \overline{R}_{L_M^{\gamma-\phi}}[f(x)] \\ s.t & f(x) \in span(\overline{\mathcal{H}}), \end{cases} \tag{25}$$

where $\overline{\mathcal{H}} = \{\overline{h}_1(x), \ldots, \overline{h}_r(x)\}$ is a set of multivariate weak learners,

$$\overline{h}_i(x) : \mathcal{X} \to \mathbb{R}^d. \tag{26}$$

Let $f^t(x) = [f_1^t(x), \ldots, f_d^t(x)]$ be the predictor available after t boosting iterations. At iteration $t+1$, a step is given along the direction $g(x) \in \overline{\mathcal{H}}$ of largest decrease of the risk $\overline{R}_{L_M^{\gamma-\phi}}[f(x)]$. This is determined by the directional derivative of $\overline{R}_{L_M^{\gamma-\phi}}[f(x)]$ along the functional $g : \mathcal{X} \to \mathbb{R}^d$, at point $f(x) = f^t(x)$ (Frigyik et al., 2008),

$$\delta\overline{R}_{L_M^{\gamma-\phi}}[f^t; g] = \left. \frac{\partial \overline{R}_{L_M^{\gamma-\phi}}[f^t + \epsilon g]}{\partial \epsilon} \right|_{\epsilon=0}. \tag{27}$$

As shown in Appendix B.1,

$$-\delta\overline{R}_{L_M^{\gamma-\phi}}[f^t; g] = \frac{1}{2n}\sum_{i=1}^n w_i \left\langle g(x_i), y^{c_i} - \sum_{k \neq c_i} y^k \tau_k(x_i, c_i) \right\rangle, \tag{28}$$

where

$$w_i = -\gamma' \left(\sum_{k \neq c_i} \phi \left[\frac{1}{2} \left\langle f^t(x_i), y^{c_i} - y^k \right\rangle \right] \right) \left(\sum_{k \neq c_i} \phi' \left[\frac{1}{2} \left\langle f^t(x_i), y^{c_i} - y^k \right\rangle \right] \right), \tag{29}$$

and

$$\tau_k(x, c) = \frac{\phi' \left[\frac{1}{2} \left\langle f^t(x), y^c - y^k \right\rangle \right]}{\sum_{k \neq c} \phi' \left[\frac{1}{2} \left\langle f^t(x), y^c - y^k \right\rangle \right]}. \tag{30}$$

The direction of steepest descent is the weak learner

$$g^*(x) = \arg\min_{g \in \overline{\mathcal{H}}} \delta\overline{R}[f^t(x); g(x)] \tag{31}$$

$$= \arg\max_{g \in \overline{\mathcal{H}}} \sum_{i=1}^n w_i \left\langle g(x_i), y^{c_i} - \sum_{k \neq c_i} y^k \tau_k(x_i, c_i) \right\rangle, \tag{32}$$

and the optimal step size along this direction

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}} \overline{R}_{L_M^{\gamma-\phi}}[f^t(x) + \alpha g^*(x)]. \tag{33}$$

Note that α^* may not have a closed form and a line search might be required. The predictor is finally updated according to

$$f^{t+1}(x) = f^t(x) + \alpha^* g^*(x). \tag{34}$$

This procedure is summarized in Algorithm 1-right, and denoted Gradient Descent Multiclass Boosting (GD-MCBoost).

4.3. Coordinate descent

Alternatively, (21) can be minimized by learning a linear combination of scalar functions. In this case, the optimization problem is

$$\begin{cases} \min_{f_1(x), \ldots, f_d(x)} & \overline{R}_{L_M^{\gamma-\phi}}([f_1(x), \ldots, , f_d(x)]) \\ s.t & f_j(x) \in span(\mathcal{H}) \quad \forall j = 1, \ldots, d, \end{cases} \tag{35}$$

where $\mathcal{H} = \{h_1(x), \ldots, h_r(x)\}$ is a set of *scalar* weak learners $h_i(x) : \mathcal{X} \to \mathbb{R}$. Let $f^t(x)$ be the predictor available after t boosting iterations. At iteration $t+1$ a single component $f_j(x)$ of $f(x)$ is updated with a step in the direction of the scalar functional g that most decreases the risk $\overline{R}_{L_M^{\gamma-\phi}}[f_1^t, \ldots, f_j^t + \alpha_j^* g, \ldots, f_d^t]$. For this, we consider the functional derivative of $\overline{R}_{L_M^{\gamma-\phi}}[f(x)]$

along the direction of the functional $g : \mathcal{X} \to \mathbb{R}$, at point $f(x) = f^t(x)$, with respect to the jth component $f_j(x)$ of $f(x)$

$$\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; j, g] = \left. \frac{\partial \overline{R}_{L_M^{\gamma-\phi}}[f^t + \epsilon g \mathbf{1}_j]}{\partial \epsilon} \right|_{\epsilon=0}, \tag{36}$$

where $\mathbf{1}_j \in \mathbb{R}^d$ is a vector whose j^{th} element is one and the remaining zero, i.e. $f^t + \epsilon g \mathbf{1}_j = [f_1^t, \ldots, f_j^t + \epsilon g, \ldots, f_d^t]$. As shown in Appendix B.2,

$$-\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; j, g] = -\frac{1}{2} \sum_{i=1}^n w_i g(x_i) \left\langle \mathbf{1}_j, y^{c_i} - \sum_{k \neq c_i} y^k \tau_k(x_i, c_i) \right\rangle, \tag{37}$$

where w_i is as in (29) and $\tau_k(x, c)$ as in (30). The direction of largest descent is then

$$g^*(x) = \arg\min_{g \in \mathcal{H}} \delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; j, g] \tag{38}$$

$$= \arg\max_{g \in \mathcal{H}} \sum_{i=1}^n w_i g(x_i) \left\langle \mathbf{1}_j, y^{c_i} - \sum_{k \neq c_i} y^k \tau_k(x_i, c_i) \right\rangle, \tag{39}$$

and the optimal step size along this direction

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}} \overline{R}[f^t(x) + \alpha g^*(x) \mathbf{1}_j]. \tag{40}$$

Again, this step size may not have a closed form and a line search might be required. The predictor is finally updated with

$$f^{t+1} = f^t(x) + \alpha^* g^*(x) \mathbf{1}_j = [f_1^t, \ldots, f_j^t + \alpha^* g^*, \ldots, f_d^t]. \tag{41}$$

This procedure is summarized in Algorithm 1-left and denoted Coordinate Descent Multiclass Boosting (CD-MCBoost). In each iteration of CD-MCBoost, the best weak learner update is found for each of the d coordinates and the coordinate whose update results in the largest reduction of the risk is then chosen. As can be seen from Algorithms 1, MC-Boost algorithms share the simplicity of implementation (a few lines of code) of binary boosting. The main differences between the two are 1) the use of multi-dimensional predictor and codewords, and 2) the weak learner selection rule. We leave an analysis of the role of codewords to Section 5 and consider the weak learner selection rule next.

4.4. Weak learner selection and boosting weights

The update equations of the two MCBoost algorithms are a natural generalization of the update equations of binary boosting. From (32) and (39), the update equation can, in both cases, be written as

$$g^*(x) = \arg\max_p \frac{1}{n} \sum_{i=1}^n w_i \hat{\mathcal{M}}_{f_t}(y^{c_i}, g(x_i)) \tag{42}$$

where

$$\hat{\mathcal{M}}_{f_t}(y^c, g(x)) = \frac{1}{2} \left\langle g(x), y^c - \sum_{k \neq c} \tau_k^t(x, c) y^k \right\rangle. \tag{43}$$

The only difference is that, while for GD-MCBoost $g(x)$ is a generic vector $g(x) \in \mathbb{R}^d$, for CD-MCBoost it is a vector of the form $g(x)\mathbf{1}_j$ with $g(x) \in \mathbb{R}$. The two algorithms are thus conceptually equivalent. We next show that they are conceptually equivalent to binary boosting algorithms in the sense that they seek, at each iteration, the weak learner of largest margin on a reweighted training sample. The only difference is that MCBoost implements two weighting mechanisms. The first one, implemented through the weights w_i, emphasizes difficult examples. The second one, implemented through $\hat{\mathcal{M}}_{f_t}$, emphasizes difficult classes.

When the $\gamma - \phi$ loss is margin enforcing, the first weighting mechanism of MCBoost assigns to each example a weight inversely proportional to how well the currently predictor classifies the example. This can be seen by rewriting (29) as

$$w_i = -\gamma' \left(\sum_{k \neq c_i} \phi(v^k) \right) \left(\sum_{k \neq c_i} \phi'(v^k) \right) \Bigg|_{v^k = \frac{1}{2}\left\langle f^t(x_i), y^{c_i} - y^k \right\rangle} \tag{44}$$

and using Theorem 5 (see Section 7), which states that a $\gamma - \phi$ loss is margin enforcing when $\xi(v) = \gamma \circ \phi(v)$ is a decreasing function. Since $\xi(v) \geq 0$, by definition of $\gamma - \phi$ loss, the derivative of $\xi(v)$ must approach zero for large positive values of v, i.e. $\gamma'(\phi(v))\phi'(v) \to 0$ for large positive v. Hence, w_i is a decreasing function of the margin components $< f^t(x_i), y^{c_i} - y^k >$ and close to zero when the smallest component is positive and large. Since, from (9), this implies a large margin, examples of larger margin under the current predictor $f^t(x)$ will receive smaller weights than examples of smaller margin. As is common in boosting, this focuses the learning resources in the examples that are poorly classified by $f^t(x)$. Hence, the first weighting mechanism is the multiclass generalization of the weighting mechanism of binary boosting.

The second weighting mechanism appears on $\hat{\mathcal{M}}_{f_t}(y^c, g(x))$. By rewriting this quantity as

$$\hat{\mathcal{M}}_{f_t}(y^c, g(x)) = \frac{1}{2} \sum_{k \neq c} \tau_k^t(x, c) \left[\left\langle g(x), y^c - y^k \right\rangle \right] \tag{45}$$

and comparing to (9), it can be seen that $\hat{\mathcal{M}}_{f_t}(y^c, g(x))$ is an estimate of the margin $\mathcal{M}(y^c, g(x))$. Rather than taking the minimum of the margin components $\left\langle g(x), y^c - y^k \right\rangle$ over the classes $k \neq c$, this estimate is a weighted average of these components, assigning weight $\tau_k^t(x, c)$ to class k. Hence, $\tau_k^t(x, c)$ is a weighting mechanism that favors some classes over others. To interpret this mechanism it is useful to consider the case where $\phi(v) = e^{-v}$ and (30) reduces to

$$\tau_k^t(x, c) = \frac{e^{\left[-\frac{1}{2}\left\langle f^t(x), y^c - y^k \right\rangle\right]}}{\sum_{k \neq c} e^{-\left[\frac{1}{2}\left\langle f^t(x), y^c - y^k \right\rangle\right]}} = \frac{e^{\left[\frac{1}{2}\left\langle f^t(x), y^k \right\rangle\right]}}{\sum_{k \neq c} e^{\left[\frac{1}{2}\left\langle f^t(x), y^k \right\rangle\right]}}, \tag{46}$$

i.e. the soft-min of the margin components (and the soft-max of codeword projections). More generally, when $\phi'(v)$ is any decreasing function, $\tau_k^t(x, c)$ is a generalized soft-min operator. It assigns larger weights to classes of smaller margin component and smaller weights to classes of larger margin component. The choice of ϕ function controls the softness of the weight assignment. For example, if $\phi(v) = e^{-\alpha v}$ with $\alpha > 0$, the softness of the assignments is controlled by the choice of α. For large α the weights are harder, i.e. closer to one for the smallest margin component and zero for all others. For smaller α the weights are more uniform. Hence, for this choice of ϕ function,

the soft-min of (45) approaches the min of (9) as $\alpha \to \infty$. More generally, up to the approximation of the min by the soft-min operator,

$$\hat{\mathcal{M}}_{f_t}(y^c, g(x)) = \frac{1}{2}\sum_{k\neq c} \tau_k^t(x,c)\left[\left\langle g(x), y^c - y^k\right\rangle\right] \quad (47)$$

$$\approx \frac{1}{2}\langle g(x), y^c - \overline{y}\rangle \quad (48)$$

where

$$\overline{y} = \arg\min_{y_j \neq y^c} \langle f_t(x), y^c - y^j\rangle = \arg\max_{y_j \neq y^c} \langle f_t(x), y^j\rangle. \quad (49)$$

This is the multiclass margin of weak learner $g(x)$ under the alternative margin definition $\hat{\mathcal{M}}_{f_t}(y^c, g(x))$. Comparing to the original definition of (9), which can be written as

$$\mathcal{M}(y^c, g(x)) = \frac{1}{2}\left\langle g(x), y^c - \overline{\overline{y}}\right\rangle \qquad \text{where} \qquad \overline{\overline{y}} = \arg\min_{y_j \neq y^c} \langle g(x), y^c - y_j\rangle, \quad (50)$$

$\hat{\mathcal{M}}_{f^t}(y^c, g(x))$ restricts the margin of $g(x)$ to the worst case codeword $\overline{y}$ for the current predictor $f^t(x)$. The strength of this restriction is determined by the soft-min operator. If $< f^t(x), y^c - \overline{y} >$ is much smaller than $< f^t(x), y^c - y_j >, y^j \neq \overline{y}$, $\tau_k^t(x,c)$ closely approximates the minimum operator. Otherwise, the remaining codewords also contribute to (45). In summary, $\tau_k^t(x,c)$ is a set of class weights that emphasizes classes of small margin for $f^t(x)$. In addition, $\hat{\mathcal{M}}_{f^t}(y^c, g(x))$ is an estimate of the margin of $g(x)$ under the restriction to the most difficult classes for the current predictor $f^t(x)$. When combined with (29) this leads to a weighting mechanism that emphasizes difficult examples (through the weights w) and difficult classes (through the margin $\hat{\mathcal{M}}_{f^t}$).

4.5. GD vs. CD-MCBoost

Since (29) and (30) only depend on the current predictor $f^t(x)$ and not on the weak learner $g(x)$, the computation of weights is identical for GD-MCBoost and CD-MCBoost. The only difference thus resides on the margin estimate of (45). While GD-MCBoost simply chooses the vector weak learner $g(x)$ with (42)-(45), CD-MCBoost uses

$$g^*(x) = \arg\max_{g\in\mathcal{H}} \max_j \frac{1}{n}\sum_{i=1}^n w_i \hat{\mathcal{M}}_{f^t}(y^{c_i}, g(x_i), j). \quad (51)$$

In this case, the approximation of (48) is

$$\hat{\mathcal{M}}_{f^t}(y^c, g(x), j) \approx \frac{g(x)}{2}\left[\langle \mathbf{1}_j, y^c\rangle - \langle \mathbf{1}_j, \overline{y}\rangle\right] \quad (52)$$

$$= g(x)\frac{y_j^c - \overline{y}_j}{2}, \quad (53)$$

and $\mathcal{M}_{f^t}(y^c, g(x), j)$ is *a measure of the margin of* $g(x)$ *under the* j^{th} *coordinate of the codewords that determine the margin of the current predictor* $f^t(x)$. Note that, for binary codewords,

$$\mathcal{M}_{f^t}(y^{c_i}, g(x_i), j) \approx \begin{cases} 0, & y_j^{c_i} = \overline{y}_j \\ y_j^{c_i} g(x_i) & y_j^{c_i} \neq \overline{y}_j. \end{cases} \quad (54)$$

This is the standard definition of margin for a scalar weak learner $g(x)$, but discards the points for which $\overline{y}_j = y_j^{c_i}$. In this way, the boosting algorithms modulates the emphasis on poorly classified points in a coordinate-by-coordinate manner.

5. Optimal codewords

So far, we have assumed the existence of a good codeword set $\mathcal{Y} = \{y^1, \ldots, y^M\}$. In this section we study the impact of the codeword set on the performance of MCBoost algorithms.

5.1. The role of codewords

In both MCBoost algorithms, the predictor is initialized with $f(x) = 0 \quad \forall x$, i.e. all examples are mapped to the origin. By minimizing (21), MCBoost then learns a predictor $f(x)$ of maximum margin. From (10), the margin of example x_i is maximal when the projection of $f(x_i)$ is maximal along correct class codeword, y^{c_i}, and minimal along the remaining codewords. If the margin of x_i is positive, it can be increased by increasing the magnitude of the projection of $f(x_i)$ along y^{c_i}. Hence, both MCBoost algorithms seek a predictor $f(x)$ that, for all i, $f(x_i)$ 1) is as aligned as possible with y^{c_i} and 2) has the largest possible magnitude along this direction. Hence, starting from the origin, both MCBoost algorithms *push all example predictions outward, in the direction of the corresponding class codewords*. The effectiveness of this mechanism depends on the structure of the codeword set. For example, if two classes were to share a codeword, it would be impossible to distinguish them with (16) or (17). This implies that some codeword sets result in larger margins for the decision rule of (17) than others.

Intuitively, the margin capacity will increase if the codewords are better separated. This suggests that better performance should be possible for larger values of the codeword dimension d. However, since increasing d increases the complexity of the decision rule, there is a trade-off between margin capacity and implementation complexity. For a fixed d, it remains to determine how to best separate the codewords, so as to maximize the margin capacity, and whether a solution to this problem, i.e. an optimal codeword set, exists.

5.2. Optimal codeword sets

To find the optimal set of codewords, we start by noting that there are several factors that impact the margin, (14), of the learned predictor by MCBoost. The first is the complexity and separability of the underlying problem, i.e., if the problem is not separable, there will never be a classifier with 100% accuracy and the margin of (14) will always be negative. The second is the set of weak learners, i.e. if the set of weak learners is limited then the space of their linear combinations won't be rich enough to have a good classifier. Finally as mentioned above the set of codewords can also impact the margin.

To focus on the impact of the codewords, we assume that 1) the problem is separable and, 2) our weak learners are rich enough and there exists a linear combination of them, f^*, that can separate training examples with 100% accuracy. Under these assumptions, according to (14), the margin of f^* is the positive and we can arbitrarily increase it by increasing norm of f^*. Therefore we can achieve arbitrarily large margin for any codewords set. To resolve this problem we assume that norm of f^* is bounded, e.g., $\|f^*\| = 1$ [2].

Definition 5 *A predictor $f(x)$ is normalized if $||f(x)|| = 1, \forall x$. $\mathcal{F}$ is the set of normalized predictors.*

2. Note that this normalization constraint is only required for finding the optimal set of codewords and is not required in MCBoost algorithm.

Finally note that (9) is invariant to the addition of a constant to all codewords. Also if the margin is positive you can arbitrarily increase it by increasing the codewords norms. To remove these issues, we complement the unit norm constraint of (8) with the constraint that the codewords be centered, i.e., add up to zero.

Definition 6 *A set of vectors $\mathcal{Y} = \{y^1, \ldots, y^M\} \in \mathbb{R}^d$, is denoted a centered (M, d) codeword set if*

$$\sum_{k=1}^{M} y^k = 0, \quad \|y^k\| = 1 \quad \forall k = 1, \ldots, M. \tag{55}$$

The set of all centered (M, d) codeword sets is denoted $\mathcal{S}(M, d)$.

In the rest of this section, we assume that codeword sets are always centered, simply referring to a centered (M, d) codeword set as "an (M, d) codeword set,".

Definition 7 *Consider an M-ary classification problem with a codeword set $\mathcal{Y} \in \mathcal{S}(M, d)$. The margin capacity of $\mathcal{Y}$ is*

$$\mathcal{C}[\mathcal{Y}] \quad = \quad \min_{k=1,\ldots,M} \mathcal{M}(y^k, \xi^k), \tag{56}$$

where

$$\xi^k = \arg\max_{\|v\|=1} \mathcal{M}(y^k, v). \tag{57}$$

is the predictor direction of largest margin for class k.

The margin capacity $\mathcal{C}[\mathcal{Y}]$ can then be interpreted as the maximum margin achievable by any predictor in $\mathcal{F}$ using codewords $\mathcal{Y}$ on any dataset $\mathcal{D}$. Note, from (14), that it is the margin, with respect to $\mathcal{Y}$ and $\mathcal{D}$, of a predictor which maps all examples from class k into the direction ξ^k of largest margin. A large capacity implies that the codeword set is such that a large margin can be achieved for all classes. A small capacity implies that there is at least one class for which the largest achievable margin is small. We define the optimal codeword set as that of largest capacity.

Definition 8 *$\mathcal{Y}^* \in \mathcal{S}(M, d)$ is a codeword set of maximum capacity if*

$$\mathcal{Y}^* = \arg\max_{\mathcal{Y} \in \mathcal{S}(M,d)} \mathcal{C}[\mathcal{Y}]. \tag{58}$$

To formalize this discussion, we start by deriving an upper bound on the margin achievable along any direction of largest margin.

Theorem 1 *Let $\mathcal{Y} \in \mathcal{S}(M, d)$ be a codeword set with directions of largest margin ξ^k, $k \in \{1, \ldots, M\}$ as defined in (57). Then, $\forall k$*

$$\mathcal{M}(y^k, \xi^k) \leq \frac{M}{2(M-1)}, \tag{59}$$

Proof See Appendix C.1. ■

An immediate consequence of (56) and (59) is that, for any codeword set $\mathcal{Y} \in \mathcal{S}(M, d)$,

$$\mathcal{C}[\mathcal{Y}] \leq \frac{M}{2(M-1)}. \tag{60}$$

The right hand side of this inequality is denoted the capacity bound of $\mathcal{S}(M, d)$. The following result characterizes the codeword sets that achieve this bound.

Theorem 2 *A codeword set $\mathcal{Y} \in \mathcal{S}(M, d)$ meets the capacity bound of (60) with equality if and only if its directions of largest margin are $\xi^k = y^k, \forall k$ and*

$$\left\langle y^k, y^l \right\rangle = -\frac{1}{M-1} \quad \forall k, l, k \neq l. \tag{61}$$

Proof See Appendix C.2. ■

The theorem shows that a codeword set meets the capacity bound if and only if it has the directions of largest margin as codewords. We next derive the conditions under which such a set of codewords exists.

Theorem 3 *$\mathcal{S}(M, d)$ contains a set of codewords $\mathcal{Y}^c(M, d)$ that meets the capacity bound if and only if $d \geq M - 1$. In this case, the codewords in $\mathcal{Y}^c(M, d)$ are the vertexes of a regular simplex in $\mathbb{R}^d$.*

Proof See Appendix C.3. ■

The proof of the theorem is constructive, providing a procedure to determine the codeword set $\mathcal{Y}^c(M, d)$. The main results of this section are summarized in the following corollary, which is a straightforward consequence of Theorems 1 and 3.

Corollary 3 *Let $\mathcal{Y}^*$ be a codeword set of maximum capacity in $\mathcal{S}(M, d)$. If $d \geq M - 1$, the codewords $(y*)^k$ of $\mathcal{Y}^*$ are the vertexes of a regular simplex in $\mathbb{R}^d$. In this case, the directions of largest margin are $\xi^k = (y^*)^k$,*

$$\left\langle (y^*)^k, (y^*)^l \right\rangle = -\frac{1}{M-1}, \ \forall k, l \neq k \tag{62}$$

and

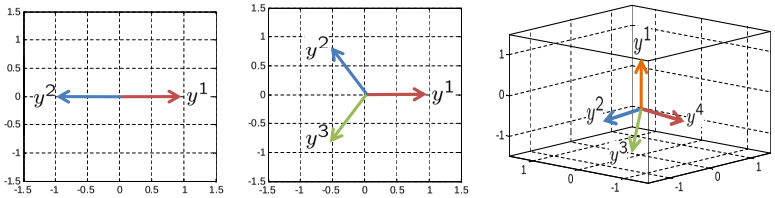
$$\mathcal{C}[\mathcal{Y}^*] = \frac{M}{2(M-1)}. \tag{63}$$

In summary, the capacity bound of (60) is met whenever $d \geq M - 1$. Since this bound only depends on the number of classes M, not in the dimension d, and the decision rule of (17) has linear complexity in d, there is usually no benefit in adopting codewords of dimension larger than $M - 1$. Hence, when the classification problem has no further constraints, it is natural to rely on codeword sets of dimension

$$d = M - 1. \tag{64}$$

Figure 1 presents the optimal codeword sets for various M. Note that in the binary case, $M = 2$, the optimal codewords are the classical $\{+1, -1\}$ labels[3].

3. A set of Matlab scripts that determine the optimal codeword set for any M is available at http://www.svcl.ucsd.edu/publications/conference/2014/icml/ICML_2014_guess_averse_code_data.zip



$(M = 2, d = 1)$ $(M = 3, d = 2)$ $(M = 4, d = 3)$

Figure 1: Codewords of maximal capacity for different values of M.

5.3. Low-dimensional predictors

When $d < M - 1$, there is no guarantee that a codeword set of maximum capacity will achieve the capacity bound. Nevertheless, the choice of $d < M - 1$ can be appealing for applications where it is critical to use low-dimensional predictors. In this case, from (9)-(11), (56), (57), and (58), the search for the codeword set of maximum capacity requires the solution of

$$\mathcal{Y}^* = \arg \max_{\mathcal{Y} \in \mathcal{S}(M,d)} \min_{k=1,\ldots,M} \max_{||v||=1} \min_{l \neq k} \left[\left\langle y^k, v \right\rangle - \left\langle y^l, v \right\rangle \right]. \tag{65}$$

This is a non-trivial optimization for which, to the best of our knowledge, there are no efficient algorithms. Hence, it is of interest to consider alternative optimality criteria. One possibility is the max-min codeword distance criterion.

Definition 9 *Let $\mathcal{Y}$ be a codeword set in $\mathcal{S}(M, d)$. The minimum distance of $\mathcal{Y}$ is*

$$d_{min}[\mathcal{Y}] = \min_{k, l \neq k} \|y^k - y^l\|^2. \tag{66}$$

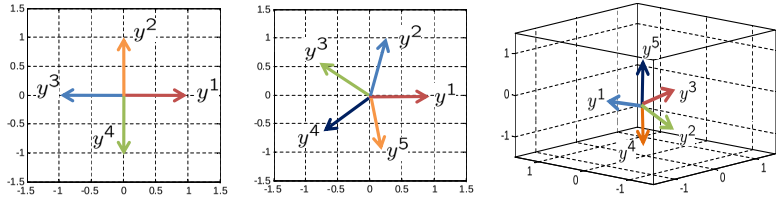
$\mathcal{Y}^*$ is a max-min distance codeword set in $\mathcal{S}(M, d)$ if

$$\mathcal{Y}^* = \arg \max_{\mathcal{Y} \in \mathcal{S}(M,d)} d_{min}[\mathcal{Y}]. \tag{67}$$

Max-min distance codeword sets have various appealing properties. First, the maximization of $d_{min}[\mathcal{Y}]$ is intuitive. From (17), the decision rule of the maximum margin classifier is based on the projections $\langle f, y^k \rangle$ of the predictor f along the codewords. If the codewords are similar, the same will hold for the projections, resulting in lower margins. Second, the problem of (67) is equivalent to determining the maximum diameter of M equal circles placed on the surface of the unit sphere without overlap. This is known as the *Tammes* problem (Tammes, 1930). While it does not have closed-form solution, or even a unique solution (any rotation of a valid solution is a valid solution), its numerical solution is much simpler than that of (65). Finally, as shown in the following result, there is a close relationship between the capacity and the minimum distance of any codeword set in $\mathcal{S}(M, d)$.

Lemma 2 *Let $\mathcal{Y}$ be a codeword set in $\mathcal{S}(M, d)$. Then*

$$\frac{1}{4} d_{min}[\mathcal{Y}] \leq \mathcal{C}[\mathcal{Y}] < \frac{1}{2} d_{min}[\mathcal{Y}] \tag{68}$$



$(M = 4, d = 2)$ $(M = 5, d = 2)$ $(M = 5, d = 3)$

Figure 2: max-min codeword sets for different values of d and M.

and

$$d_{min}[\mathcal{Y}] \leq \frac{2M}{M-1}. \tag{69}$$

Proof See Appendix D.1. ■

The following theorem uses this result to show that the optimal codeword sets under the optimality criteria of Definitions 8 and 9 are identical when $d > M - 1$.

Theorem 4 *Let $\mathcal{Y}^*$ be a codeword set in $\mathcal{S}(M, d)$. If $d \geq M - 1$ then $\mathcal{Y}^*$ is a codeword set of maximum capacity, i.e.*

$$\mathcal{C}[\mathcal{Y}^*] = \frac{M}{2(M-1)}, \tag{70}$$

if and only if $\mathcal{Y}^*$ is a codeword set of max-min distance, i.e.

$$d_{min}[\mathcal{Y}^*] = \frac{2M}{M-1}. \tag{71}$$

Proof See Appendix D.2. ■

In summary, in the regime of $d \geq M - 1$, the vertexes of a regular simplex in $\mathbb{R}^d$ are both a maximum capacity and a max-min distance codeword sets of $\mathcal{S}(M, d)$. These codeword sets achieve both the capacity and max-min distance bounds of $\mathcal{S}(M, d)$. The main difference between the two optimality criteria is the difficulty of finding an optimal solution for $d < M - 1$. While the optimization problem of (65) is difficult, there are many algorithms for the solution of (67). Our implementation uses a solver based on the barrier method (Nocedal and Wright, 1999). Figure 2 presents max-min codewords sets for different values of M and d.

5.4. Complexity vs. capacity

In general, the complexity of an M-ary predictor increases with the number M of classes. For example, the "one vs all" architecture requires the evaluation of one predictor per class, i.e. has complexity $O(M)$, while the "all pairs" architecture has complexity $O(M^2)$. While the decision rule $F(x)$ of (17) always has complexity $O(M)$, this consists of computing M d-dimensional dot-products and can be usually be performed efficiently for large values of M, as long as d is small.

Hence, for MCBoost, predictor complexity is mostly a function of the predictor dimension d, not the number of classes. In fact, MCBoost can solve any M-ary classification problem with a predictor of dimension as low as $d = 2$. Obviously, this carries a penalty, by limiting the margin capacity of the codeword set. While, as discussed in Theorem 3, the margin capacity can be achieved for any dimension $d \geq M - 1$, smaller dimensions typically lead to reduced classification accuracy. Hence, the dimension d can be seen as a parameter that controls the trade-off between predictor complexity and accuracy. Note that complexity does not need to refer simply to the number of computations. For $d < M$, the act of learning the predictor $f(x)$ can be seen as a form of discriminant dimensionality reduction. This could be of interest for applications such as hashing (Datar et al., 2004; Torralba et al., 2008; Kulis and Darrell, 2009; Gong et al., 2012; Liu et al., 2012), where it is important to represent information from many classes by short codes. In this case, MCBoost can directly learn a discriminant *representation* of low complexity.

6. Weak Learners

In this section, we investigate the role of weak learners in MCBoost. We consider the questions of which weak learner families can be boosted, the role of classification versus real-valued learners, and how these connect to interpretations of boosting as voting rules. While we limit the discussion to GD-MCBoost, for brevity, all considerations extend to CD-MCBoost.

6.1. Boostability

The selection of weak learner family for a boosting algorithm is usually guided by a boostability condition. A weak learner family is boostable if it guarantees zero training error after a sufficient number of iterations of the boosting algorithm. For binary classification, this holds if the family contains a weak learner with better than random performance, i.e. error rate smaller than 50%, for any weight distribution over training examples. Existing multiclass boosting algorithms are usually designed for specific classes of weak learners, e.g regression or decision stumps, and require this boostability condition. However, while trivial for binary classification, the "less than chance error" condition can be non-trivial to prove for multiclass problems, where chance level error is $M^{-1} \times 100\%$. For example, (Mukherjee and Schapire, 2013) used a fairly sophisticated game theoretic analysis to derive boostability conditions in terms of the error rate of a set of weak learners.

Unlike most of these algorithms, MCBoost has a fairly straightforward boostability condition for any convex $\gamma - \phi$ loss. It suffices that, at any iteration t, there exists a weak learner $h \in \mathcal{H}$ for which the directional derivative of (28) satisfies

$$-\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; h] > \delta, \tag{72}$$

for some fixed $\delta > 0$. In this case, it is always possible to reduce the classification risk by adding h to the weak learner ensemble and, by convexity, the algorithm will converge to the minimum training error, which is zero for separable data. In summary, a set of weak learners $\mathcal{H}$ is boostable by MCBoost if, under any weight distribution over training examples, it contains a member h for which (72) holds with $\delta > 0$.

Similarly to binary boosting, this condition can be expressed in terms of the performance of a set of random weak learners. Let $\rho(x)$ be a weak learner with random output, i.e.

$$P(\rho(x) = y^k) = \frac{1}{M} \quad \forall x, k = 1, \ldots, M. \tag{73}$$

It follows that, for any given x_i, $\epsilon_i = \rho(x_i)$ is a sample from a random vector ϵ uniformly distributed over the codeword set $\{y^1, \ldots, y^M\}$. The derivative of (28)

$$-\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; \rho] = \frac{1}{2n} \sum_{i=1}^{n} w_i \left\langle \epsilon_i, y^{c_i} - \sum_{k \neq c_i} y^k \tau_k(x_i, c_i) \right\rangle \tag{74}$$

has expected value

$$E_\epsilon \left\{ -\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; \rho] \right\} = \frac{1}{2n} \sum_{i=1}^{n} w_i \left\langle E_\epsilon \{\epsilon_i\}, y^{c_i} - \sum_{k \neq c_i} y^k \tau_k(x_i, c_i) \right\rangle, \tag{75}$$

Since

$$E_\epsilon \{\epsilon_i\} = \sum_{k=1}^{M} \frac{y^k}{M} = 0, \tag{76}$$

where the second equality follows from (55), it follows that

$$E_\epsilon \left\{ -\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; \rho] \right\} = 0. \tag{77}$$

Hence, the boostability condition of (72) can be written as

$$-\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; h] > E_\epsilon \left\{ -\delta \overline{R}_{L_M^{\gamma-\phi}}[f^t; \rho] \right\} + \delta, \tag{78}$$

with $\delta > 0$. A weak learner family is boostable if, for any weight distribution, it contains a weak learner h that beats the random learner ρ. Hence, as is the case for binary boosting, MCBoost can boost any set of any weak learners h whose performance is slightly better than random, i.e. has "less than $1/M \times 100\%$ error," for any weight distribution. Note that this result is valid for any set $\mathcal{H}$ of weak learners. In fact, the weak learners do not even have to be classifiers. CD-Boost can be applied to any set of real-valued functions $h : \mathcal{X} \to \mathbb{R}$ and GD-MCBoost to any set of multi-dimensional functions $\overline{h} : \mathcal{X} \to \mathbb{R}^d$.

6.2. Classification weak learners and voting rules

Obviously, MCBoost can be implemented with classification weak learners. In this case, the algorithm can be presented as a voting rule, as in the classical presentation of AdaBoost. In appendix E.1 we derive the GD-MCBoost equations for the case where the weak learner space $\overline{\mathcal{H}}$ is a space of multiclass classifiers, i.e. $h : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y} = \{y^1, \ldots, y^M\}$ is the codeword set of the classification problem. We show that, when this is the case, the learned predictor has the form

$$f(x) = \sum_k y^k \sum_{t | g_t(x) = y^k} \alpha_t \tag{79}$$

if $\mathcal{Y}$ is a codeword set of maximum capacity in $\mathcal{S}(M, d)$ and

$$f_k(x) = \sum_{t | g_t(x) = \mathbf{1}_k} \alpha_t. \tag{80}$$

if $\mathcal{Y}$ is a set of canonical codewords $\mathbf{1}_j$. In either case, denoting $I(g(x) = y^k)\alpha$ as a vote of strength α from weak learner $g(x)$ for class k, $f(x)$ can be seen as a weighted average of the class codewords, where each class is weighted by the strength of its weak learner votes.

It is also shown that, although these predictors have different codeword projections, namely

$$\left\langle f(x), y^j \right\rangle = \sum_{t|g_t(x)=y^j} \alpha_t - \frac{1}{M-1} \sum_{k\neq j} \sum_{t|g_t(x)=y^k} \alpha_t, \tag{81}$$

for the maximum capacity codeword set and

$$\left\langle f(x), y^j \right\rangle = \sum_{t|g_t(x)=\mathbf{1}_j} \alpha_t \tag{82}$$

for canonical codewords, the margin components are similar up to a rescaling, namely

$$\left\langle f(x), y^c - y^k \right\rangle = \frac{M}{M-1} \left(\sum_{t|g_t(x)=y^c} \alpha_t - \sum_{t|g_t(x)=y^k} \alpha_t \right), \tag{83}$$

for a maximum capacity set and

$$\left\langle f(x), y^c - y^k \right\rangle = \sum_{t|g_t(x)=\mathbf{1}_c} \alpha_t - \sum_{t|g_t(x)=\mathbf{1}_k} \alpha_t \tag{84}$$

for a canonical set. In summary, for maximum capacity codewords, the projection of $f(x)$ along codeword y^j is the difference between the strength of the weak learner votes for class j and the average vote strength for the remaining classes. Hence, the projection is large if and only if the class receives votes of strength well above average. On the other hand, for canonical codewords, the projection reduces to the strength of the weak learner votes for class j. The margin components are always larger under the maximum capacity codeword set, but the difference vanishes for large M. Nevertheless, due to the similar form of the margin components, the decision rule of (17) is the same in the two cases and consists of choosing the class of strongest vote

$$F(x) = \arg\max_k \sum_{t|g_t(x)=y^k} \alpha_t. \tag{85}$$

The margin condition of (18) thus holds if and only if this is the true class y^c.

With regards to learning, it is shown in appendix E.1 that the steepest descent direction of (32) is also

$$g^*(x) = \arg\min_{g\in\mathcal{H}} \sum_{i|g(x_i)\neq y^{c_i}} w_i \left(1 + \sum_{j\neq c_i} \tau_j(x_i, c_i) I(g(x_i) = y^j) \right), \tag{86}$$

in the two cases. The weak learner selection rule reduces to selecting the classifier $g(x)$ of lowest error rate on a dataset where each point x_i is reweighted proportionally to how poorly it is classified by the current predictor $f^t(x)$ (which is captured by w_i) and the agreement between the (incorrect) class to which it is assigned by $g(x)$ and $f^t(x)$ (captured by $\sum_{j\neq c_i} \tau_j(x_i, c_i) I(g(x_i) = y^j)$). Note that, even though the weak learner selection rule is identical for the two codeword sets, the algorithms are not the same, since the weights w_i and $\tau_j(x_i, c_i)$ are determined by the margin components, according to (29)-(30),

6.3. Classification vs. real valued learners

Finally, we consider the question of whether anything is lost by using a set of classification weak learners. Consider the implementation of MCBoost in a space $\mathcal{H}$ of continuous weak learners. Let the direction of steepest descent of the risk at a given iteration be $h(x) \in \mathcal{H}$. Then, it is possible to define a classification weak learner

$$g(x) = y^k, \qquad k = \arg\max_j < h(x), y^j > . \tag{87}$$

It follows that

$$\int \langle g(x), h(x) \rangle \, dx = \int \left\langle y^{\arg\max_j <h(x),y^j>}, h(x) \right\rangle dx \tag{88}$$

$$= \int \arg\max_j \left\langle h(x), y^j \right\rangle dx. \tag{89}$$

Since $\arg\max_j \left\langle h(x), y^j \right\rangle \geq \left\langle \frac{y^j+y^l}{2}, y^j \right\rangle, \forall l \neq j$, and

$$\left\langle \frac{y^j+y^l}{2}, y^j \right\rangle = \begin{cases} \frac{1}{2} & \text{if } \mathcal{Y} \text{ is a canonical codeword set} \\ \frac{1}{2} - \frac{1}{2(M-1)} & \text{if } \mathcal{Y} \text{ is a maximum capacity codeword set,} \end{cases} \tag{90}$$

the inequality $\arg\max_j \left\langle h(x), y^j \right\rangle > 0$ holds for both classes of codewords, whenever $M > 2$. Hence,

$$\int \langle g(x), h(x) \rangle \, dx > 0 \tag{91}$$

and $g(x)$ is a descent direction, albeit not the steepest descent direction, in $\mathcal{H}$. It follows that the family of weak learners $g(x)$ is boostable. In summary, for either canonical or maximum capacity codeword sets, whenever there is a boostable set $\mathcal{H}$ of generic weak learners, there is also a boostable family of classification weak learners, given by (87). The only penalty in adopting the latter is that, because MCBoost is no longer a steepest descent procedure in $\mathcal{H}$, its convergence can be slower.

7. Loss functions

So far, we have considered a generic $\gamma - \phi$ loss function. In this section, we derive conditions under which $\gamma - \phi$ losses belong to some families with desirable properties for classification, such as margin or proper losses. We then design specific $\gamma - \phi$ losses functions in these families and use them to implement different MCBoost algorithms.

7.1. Margin losses

An important question is under which conditions $\gamma - \phi$ losses are margin enforcing, i.e. when the minimization of the empirical risk of a $\gamma - \phi$ loss encourages predictors of large margin $\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y})$. The following result shows that this holds whenever γ is strictly increasing and ϕ is strictly positive and decreasing.

Theorem 5 *Consider a training set $\mathcal{D}$, a codeword set $\mathcal{Y} = \{y^1, \ldots, y^M\}$, a predictor $f(x)$. Let $\overline{R}_{L_M^{\gamma-\phi}}(f)$ be the empirical risk of (21) for a $\gamma - \phi$ loss as in (23), such that γ is strictly increasing and ϕ is strictly positive and decreasing. Then*

$$\overline{R}_{L_M^{\gamma-\phi}}(f) \quad > \quad \frac{1}{n}\gamma\left(\phi[\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y})]\right), \tag{92}$$

where $\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y})$ is the margin of predictor f - as defined in (14).

Proof See Appendix F.1. ■

Since γ is strictly increasing and ϕ is strictly decreasing, $\xi = \gamma \circ \phi$ is strictly decreasing. It follows from (92) that the minimization of $\overline{R}_{L_M^{\gamma-\phi}}(f)$ forces large margins and the $\gamma - \phi$ loss is margin enforcing. Note that the minimization of a decreasing ξ is similar to the definition of margin loss for binary classification, where $\xi = \gamma \circ \phi$ is always decreasing. The main difference is that the components γ and ϕ now have different roles. While ϕ operates on each margin component, γ is applied to the aggregate of all the components. Hence, it no longer suffices to specify ξ, but γ and ϕ have to be specified explicitly.

7.2. Proper losses

A loss $L_M[.,.]$ is said to be *proper* if it is possible to recover the class posterior probabilities $P_{C|X}(c|x), \quad C = 1, \ldots, M$, for all x from any predictor $f^*(x)$ optimal under that loss. This property is important for applications that require a confidence score for the classification. It is natural to ask under which conditions are $\gamma - \phi$ losses proper. For simplicity, we adopt the notation

$$\eta_k(x) = P_{C|X}(k|x), \tag{93}$$

for the posterior class probabilities and represent a codeword set $\mathcal{Y} = \{y^1, \ldots, y^M\} \in \mathbb{R}^d$ by a code matrix $\mathbf{Y} \in \mathbb{R}^{d \times M}$, whose columns are the codewords in $\mathcal{Y}$, i.e.

$$\mathbf{Y} = [y^1, \ldots, y^M]. \tag{94}$$

We also resort to the well known result that, to minimize (20), it suffices to determine the predictor $f^*(x)$ of minimum conditional risk

$$R_{L_M^{\gamma-\phi}}(f|x) \quad = \quad E_{C|X}\{L_M^{\gamma-\phi}[y^c, f(x)]|x\}, \tag{95}$$

for all x. The following theorem characterizes the set of minimizers of the conditional risk of a $\gamma - \phi$ loss.

Theorem 6 *Let $\mathcal{Y} = \{y^1, \ldots, y^M\} \in \mathbb{R}^d$ be a codeword set, $f(x) : \mathcal{X} \to \mathbb{R}^d$ a predictor, $\eta_k(x)$ the posterior probabilities of (93), and $R_{L_M^{\gamma-\phi}}(f|x)$ the conditional risk of (95) when $L_M^{\gamma-\phi}[.,.]$ is a $\gamma - \phi$ loss with differentiable γ, ϕ. In this case any minimizer, f^*, of (95) is a solution of*

$$\mathbf{Y}\mathbf{Q}_{f^*}^{\phi}\mathbf{\Gamma}_{f^*}^{\gamma-\phi}\eta = 0, \tag{96}$$

where $\mathbf{Y} \in \mathbb{R}^{d \times M}$ is the code matrix of (94), $\eta = [\eta_1, \ldots, \eta_M]^T$ the vector of posterior probabilities, $\mathbf{Q}_f^{\phi}, \mathbf{\Gamma}_f^{\gamma-\phi} \in \mathbb{R}^{M \times M}$ with

$$\mathbf{Q}_f^{\phi}(k,l) = \begin{cases} -\phi'(u^l - u^k) & k \neq l \\ \sum_{j \neq k} \phi'(u^k - u^j) & k = l, \end{cases} \tag{97}$$

$$\mathbf{\Gamma}_f^{\gamma-\phi}(k,l) = \begin{cases} \gamma'\left(\sum_{l \neq k} \phi(u^k - u^l)\right) & k = l \\ 0 & k \neq l, \end{cases} \tag{98}$$

u^j as defined in (11) and we have omitted the dependency of all terms on x for notational simplicity.

Proof See Appendix G.1. ■

Since (96) holds for any minimizer f^* of (95), the posterior probability vector η can be uniquely recovered from f^* only when the null space of $\mathbf{Y}\mathbf{Q}_{f^*}^{\phi}\mathbf{\Gamma}_{f^*}^{\gamma-\phi}$ contains a single probability vector. The following lemma provides sufficient conditions for this to holds. In the following results, we use $|\mathcal{A}|$ to denote the dimensionality of a space $\mathcal{A}$, and $Rank(\mathbf{A}), Null(\mathbf{A})$, and $Range(\mathbf{A})$ to denote the rank, null space, and column space of matrix $\mathbf{A}$, respectively.

Lemma 3 *Let $\mathcal{Y} = \{y^1, \ldots, y^M\} \in \mathbb{R}^d$ be a codeword set, $\mathbf{Y}$ the corresponding matrix of (94) and $\mathbf{Q}_f^{\phi}, \mathbf{\Gamma}_f^{\gamma-\phi}$ as defined in (97), (98) respectively. The following statements hold.*

1. *If $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is strictly monotonic then*

$$Null(\mathbf{Y}\mathbf{Q}_f^{\phi}\mathbf{\Gamma}_f^{\gamma-\phi}) = Null(\mathbf{Y}\mathbf{Q}_f^{\phi}) \quad \textit{and} \quad Null(\mathbf{Q}_f^{\phi}\mathbf{\Gamma}_f^{\gamma-\phi}) = Null(\mathbf{Q}_f^{\phi}).$$

2. *If $Rank(\mathbf{Y}) \leq M - 2$ then $|Null(\mathbf{Y}\mathbf{Q}_f^{\phi})| \geq 2$*

3. *If $Rank(\mathbf{Y}) \geq M - 1$ then $Null(\mathbf{Y}\mathbf{Q}_f^{\phi}) = Null(\mathbf{Q}_f^{\phi})$.*

4. *If $\phi : \mathbb{R} \to \mathbb{R}^+$ is strictly monotonic then $|Null(\mathbf{Q}_f^{\phi})| = 1$.*

Proof See Appendix G.2. ■

The lemma shows that, if $Rank(\mathbf{Y}) \geq M-1$ and ϕ, γ are strictly monotonic, the set of vectors η for which (96) holds is a one-dimensional space. It remains to verify if this space contains a probability vector, i.e. a vector of non-negative entries that sum to one. The following lemma shows that this is indeed the case.

Lemma 4 *Let $\eta = [\eta_1, \ldots, \eta_M]^T$ and $\mathbf{Q}_f^{\phi}, \mathbf{\Gamma}_f^{\gamma-\phi}$ be as defined in (97) and (98), respectively. If $\mathbf{Q}_f^{\phi}\mathbf{\Gamma}_f^{\gamma-\phi}\eta = 0$ then*

$$\eta_k = \frac{\sum_{l=1}^M \eta_l \pi_l \phi'(u^l - u^k)}{\pi_k \sum_{l=1}^M \phi'(u^k - u^l)} \quad \forall k, \tag{99}$$

where

$$\pi_j = \gamma'\left(\sum_{l \neq j} \phi(u^j - u^l)\right) \ \forall j. \tag{100}$$

Furthermore, if $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ and $\phi : \mathbb{R} \to \mathbb{R}^+$ are strictly monotonic and $\eta \neq 0$ then all η_k have the same sign.

Proof See Appendix G.3. ■

Together, Lemmas 3 and 4 state that, for any code matrix $\mathbf{Y}$ such that $Rank(\mathbf{Y}) \geq M-1$ and any pair of strictly monotonic functions ϕ, γ, the space of η that satisfy (96) is one dimensional and composed by vectors that satisfy (99). Since, for any vector $\eta \neq \mathbf{0}$ in this subspace, all η_k have the same sign, $\sum_k \eta_k \neq 0$ and thus $\bar{\eta} = \frac{\eta}{\sum_k \eta_k}$ is a probability vector. Under these conditions, the class posterior probabilities of (96) can be recovered from the optimal predictor f^* as follows.

Theorem 7 *Let $\mathcal{Y} = \{y^1, \ldots, y^M\} \in \mathbb{R}^d$ be a codeword set of matrix $\mathbf{Y}$, $L_M^{\gamma-\phi}[.,.]$ a $\gamma-\phi$ loss, $R_{L_M^{\gamma-\phi}}(f|x)$ the conditional risk of (95), and η the vector of the posterior probabilities of (93). If $Rank(\mathbf{Y}) \geq M-1$ and (ϕ, γ) is a pair of strictly monotonic differentiable functions, then η can be recovered from any minimizer, f^*, of $R_{L_M^{\gamma-\phi}}(f|x)$ with*

$$\eta = \left(\mathbf{\Omega}_f^{\gamma-\phi}\right)^{-1} \mathbf{e}_1, \tag{101}$$

where $\mathbf{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^M$ and

$$\mathbf{\Omega}_f^{\gamma-\phi}(k,l) = \begin{cases} 1 & k=1 \\ -\pi_l \phi'(u^l - u^k) & k \neq l \\ \pi_k \sum_{j \neq k} \phi'(u^k - u^j) & k = l, \end{cases} \tag{102}$$

with u^k, π_k as defined in (11) and (100), respectively.

Proof See Appendix G.4 ■

Theorems 3 and 4 show that, for an M-ary classification problem, a codeword space of dimension $M-1$ is the smallest to contain a set of codewords that achieve either the margin capacity or maximum distance bounds. In this case, the optimal codeword set consists of the M vertexes of the regular simplex in $\mathbb{R}^{M-1}$ and can be obtained with the procedure of (Coxeter, 1973). The associated matrix $\mathbf{Y}$ has rank $M-1$ and the codewords are the directions of largest margin for each of the M classes. Theorem 7 shows that the combination of the $M-1$ dimensional simplex codewords with strictly monotonic γ, ϕ functions is a sufficient condition for a proper $\gamma-\phi$ loss. In this case, the posterior probability vector η can be recovered from any minimizer f^* of the risk, using (101).

7.3. Relations to binary classification

A third question of interest is how the results above generalize previously known properties of binary losses. For binary classification, where $M=2$ and $y^1 = -y^2 = 1$, it follows from (102) that

$$\begin{aligned} \mathbf{\Omega}_f^{\gamma-\phi} &= \begin{bmatrix} 1 & 1 \\ -\phi'(u^1-u^2)\gamma'(\phi(u^1-u^2)) & \phi'(u^2-u^1)\gamma'(\phi(u^2-u^1)), \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ -\xi'(f) & \xi'(-f), \end{bmatrix}, \end{aligned} \tag{103}$$

Table 2: Loss, link, and inverse link, of various boosting algorithms.

Algorithm	$\xi(v)$	$\zeta_{\gamma-\phi}(\eta)$	$\zeta_{\gamma-\phi}^{-1}(v)$
AdaBoost (Freund and Schapire, 1997)	$\exp(-v)$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$\frac{e^{2v}}{1+e^{2v}}$
LogitBoost (Friedman et al., 1998)	$\log(1+e^{-2v})$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$\frac{e^{2v}}{1+e^{2v}}$
SavageBoost (Masnadi-Shirazi and Vasconcelos, 2008)	$\frac{1}{(1+e^{2v})^2}$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$\frac{e^{2v}}{1+e^{2v}}$

where $f = u^1 - u^2$ and $\xi = \gamma \circ \phi$. Using (103), the solution of (101) is then

$$\eta_1 = \frac{\xi'(-f)}{\xi'(f) + \xi'(-f)} \tag{104}$$

$$\eta_2 = \frac{\xi'(f)}{\xi'(f) + \xi'(-f)}, \tag{105}$$

where we omitted the dependence on x. Defining $\eta = \eta_1$, the two equations can be rewritten as

$$\eta \xi'(f) = (1-\eta)\xi'(-f). \tag{106}$$

This is a popular equation in the binary classification literature, where it is known as a sufficient condition for f to minimize the classification risk associated with the binary margin loss $\xi(.)$ (Zhang, 2004; Reid and Williamson, 2010). It is usually possible to solve (106) for the optimal f. This is given by

$$f = \zeta_{\gamma-\phi}(\eta), \tag{107}$$

for some function $\zeta_{\gamma-\phi}$, which is denoted the link function of the loss ξ. The link function plays an important role in the recovery of the posterior probabilities η_k because, for a proper loss ϕ, $\zeta_{\gamma-\phi}$ is invertible and

$$\eta = \zeta_{\gamma-\phi}^{-1}(f). \tag{108}$$

Hence, probabilities $\eta(x)$ can be recovered by simply feeding the classifier predictions $f(x)$ through the inverse of the link. Table 2 lists the loss, link, and inverse link of the margin losses that underlie various popular boosting algorithms.

For M-ary classification, (99) generalizes the inverse link relationship of (108). The main difficulty of the M-ary extension is that the inverse link does not always have closed-form. In fact, while the exact decomposition of the loss ξ into the components γ and ϕ does not affect the link for binary classification, this is not the case for M-ary classification. Consider, for example, the exponential and logistic losses of Table 1. For all these losses, $\phi(v) = e^{\alpha v}$ with $\alpha \in \mathbb{R}^-$. Consider next a multiclass $\gamma-\phi$ loss whose $\phi(.)$ function is in this family. From (99)

$$\begin{aligned} \eta_k &= \frac{\sum_{l=1}^M \eta_l \pi_l \alpha e^{\alpha(u^l - u^k)}}{\pi_k \sum_{l=1}^M \alpha e^{\alpha(u^k - u^l)}} \\ &= \frac{e^{-\alpha u^k} \sum_{l=1}^M \eta_l \pi_l e^{\alpha u^l}}{\pi_k e^{\alpha u^k} \sum_{l=1}^M e^{-\alpha u^l}} \\ &\propto e^{-2\alpha u^k - \log \pi_k}, \end{aligned} \tag{109}$$

Table 3: Expressions of γ, ϕ, η_k, and π_k for multiclass generalizations of the binary $\gamma - \phi$ losses of Table 1. In all cases, $p_k = \frac{e^{2u^k}}{\sum_i e^{2u^i}}$.

name	multiclass loss	$\gamma(v)$	$\phi(v)$	η_k	π_k
Exponential	$\sum_{l\neq c} e^{-(u^c-u^l)}$	v	e^{-v}	p_k	1
Logistic	$\log\left(1+\sum_{l\neq c} e^{-2(u^c-u^l)}\right)$	$\log(1+v)$	e^{-2v}	p_k	p_k
Savage	$\left(1-\frac{1}{1+\sum_{l\neq c} e^{-2(u^c-u^l)}}\right)^2$	$\left(\frac{v}{1+v}\right)^2$	e^{-2v}	$\frac{1}{1+\sum_{j\neq k}\frac{1-p_k}{1-p_j}}$	$2p_k^2(1-p_k)$

where, from (100),

$$\pi_k = \gamma'\left(e^{\alpha u^k}\sum_{l=1}^{M} e^{-\alpha u^l} - 1\right). \tag{110}$$

Since η is a probability vector, it follows that

$$\eta_k = \frac{e^{-2\alpha u^k - \log \pi_k}}{\sum_i e^{-2\alpha u^i - \log \pi_i}}. \tag{111}$$

Hence, the inverse link of any $\gamma - \phi$ loss of exponential ϕ function is the softmax mapping $\rho : \mathbb{R}^M \to [0,1]^M$,

$$\rho_k(v) = \frac{e^{v_k}}{\sum_i e^{v_i}}, \tag{112}$$

of the class scores u^i rescaled according to

$$\eta = \rho(-2\alpha u - \kappa(u)) \tag{113}$$

$$u = \frac{1}{2}\left[< y^1, f >, \ldots, < y^M, f >\right]^T \tag{114}$$

$$\kappa(u) = \left[\log \pi_1(u), \ldots, \log \pi_M(u)\right]^T. \tag{115}$$

In summary, whenever the recovery of the posterior probability vector η is of interest, the choice of $\phi(v) = e^{\alpha v}$ is desirable, as it guarantees a closed form-solution for the inverse link. The choice of $\gamma(.)$ function simply determines the rescaling $\kappa(u)$ of the scores, through (110). This suggests a very natural generalization of the binary losses of Table 1, which is given in Table 3. Note that, in all cases, $\gamma \circ \phi$ is a decreasing function. Hence, by Theorem 5, all losses are margin enforcing. The table also presents the values of π_k and η_k for all losses. Since in all cases the probabilities are

$$\eta_k = \frac{e^{2u^k}}{\sum_i e^{2u^i}}, \tag{116}$$

i.e. directly proportional to the class scores u^k, the decision rule of (17) is identical to the Bayes decision rule (1).

Table 4: Expressions of the MCBoost weights of (29) and (30) for the multiclass losses of of Table 3.

name	multiclass loss	w_i	$\tau_k(x_i, c_i)$
Exponential	$\sum_{l\neq c} e^{-(u^c-u^l)}$	$\sum_{k\neq c_i} e^{-\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}$	$\frac{e^{-\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}{\sum_{k\neq c_i} e^{-\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}$
Logistic	$\log\left(1+\sum_{l\neq c} e^{-2(u^c-u^l)}\right)$	$\frac{2\sum_{k\neq c_i} e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}{1+\sum_{k\neq c_i} e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}$	$\frac{e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}{\sum_{k\neq c_i} e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}$
Savage	$\left(1-\frac{1}{1+\sum_{l\neq c} e^{-2(u^c-u^l)}}\right)^2$	$\frac{4\left(\sum_{k\neq c_i} e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}\right)^2}{\left(1+\sum_{k\neq c_i} e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}\right)^3}$	$\frac{e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}{\sum_{k\neq c_i} e^{-\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle}}$

7.4. Multiclass boosting algorithms

A final important question is how the discussion above can be used to derive multiclass boosting algorithms that generalize the ones available in the literature for binary classification. In Section 4.4, we saw that the update equations of the two MCBoost algorithms are given by (42)-(44) and (30). In the binary setting ($M = 2$) with $y^1 = 1, y^2 = -1$, (30) reduces to $\tau_k^t(x, c) = 1$ for $k \neq c$ and

$$\hat{\mathcal{M}}_{f^t}(y^c, g(x)) = \frac{1}{2}\left[g(x)y^c - g(x)(-y^c)\right] = y^c g(x), \tag{117}$$

reduces to the standard definition of margin. Furthermore, from (44)

$$w_i = -\gamma'\left(\phi(y^{c_i} f^t(x_i))\right)\left(\phi'(y^{c_i} f^t(x_i))\right) = -\xi'(y^{c_i} f^t(x_i)), \tag{118}$$

where $\xi = \gamma \circ \phi$ is a decreasing function. This leads to the classical weak learner selection rule

$$g^*(x) = \arg\max_{g\in\overline{\mathcal{H}}} \frac{1}{n}\sum_{i=1}^{n} w_i \mathcal{M}_{f^t}(y^{c_i}, g(x_i)) \tag{119}$$

$$= \arg\max_{g\in\overline{\mathcal{H}}} \frac{1}{n}\sum_{i=1}^{n} y^{c_i} w_i g(x_i), \tag{120}$$

of binary boosting algorithms. Hence, when $M = 2$, both GD-MCBoost and CD-MCBoost reduce to binary boosting with loss $\xi(v) = \gamma(\phi(v))$. It follows that, by using the losses of Table 2, MCBoost can implement all popular binary boosting algorithms, including AdaBoost (Freund and Schapire, 1997), LogitBoost (Friedman et al., 1998) and SavageBoost (Masnadi-Shirazi and Vasconcelos, 2008). Furthermore, because these are special cases of the losses of Table 3, MCBoost algorithms derived with these losses can be seen as multiclass extensions of AdaBoost and LogitBoost. Table 4 summarizes the expressions of the weights w_i and $\tau_k(x_i, c_i)$ that define these algorithms.

8. Comparison to previous learning algorithms

In this section we compare MCBoost learning algorithms in the literature. We start by comparing MCBoost with the current multiclass boosting algorithms and then compare it with the kernel-SVM.

Table 5: Comparison of multiclass boosting algorithms.

Algorithm	Optimization	Codewords	Weak Learners	Loss
OVA	CD	canonical	any	L_{ova} (225)
AdaBoost.M1	GD	canonical	classification	$L_{Ada.M1}$ (238)
SAMME	GD	max-capacity	classification	L_{SAMME} (240)
AdaBoost.MH	GD	canonical	product	L_{ova} (225)
AdaBoost.ECC	CD	binary	any	$\gamma(v) = v, \phi(v) = e^{-2v}$
LogitBoost	CD	canonical	regression	Logistic $\gamma - \phi$
CD-MCBoost	CD	max-capacity	any	any $\gamma - \phi$
AdaBoost.M2/MR	GD	canonical	product	Exponential $\gamma - \phi$
AdaBoost.MM	GD	canonical	classification	$\gamma(v) = v, \phi(v) = e^{-2v}$
GD-MCBoost	GD	max-capacity	any	any $\gamma - \phi$

8.1. Comparison to boosting algorithms

Many of the current multiclass boosting algorithms are special instances of MCBoost using different, and sometimes sub-optimal, codewords, weak learners, loss functions, or optimization strategy (CD v. GD). We discuss these relationships in detail in Appendix H, where we show that these multiclass boosting algorithms can be cast as special cases of, or approximations to, MCBoost. The differences between them are summarized in Table 5. To simplify comparisons, the table is organized in three sections. The top section contains algorithms that do not use $\gamma - \phi$ losses. These are used by the algorithms in the middle and bottom sections, which differ in optimization strategy. It can be seen that all algorithms are either gradient descent (GD) or coordinate descent (CD) procedures. They also differ in the codeword set used to label the different classes. While some use a maximum capacity codeword set, most rely on canonical codewords, or codeword choices (ECOC) of margin capacity below the bound of (59). With regards to weak learners, most algorithms adopt classification learners $h : \mathcal{X} \to \mathcal{Y} = \{y^1, \ldots, y^M\}$, but there also have been proposals for linear regressors (multiclass LogitBoost), or the product learners $h : \mathcal{X} \times \{1, \ldots, M\} \to \mathbb{R}$ of AdaBoost.M2, AdaBoost.MR, and AdaBoost.MH. Finally, while most algorithms use a special form of the $\gamma - \phi$ loss of (23), a few (OVA, AdaBoost.MH) rely on the OVA loss of (225), while others (Ada.M1,SAMME) minimize the exponential losses of (238) and (240).

As discussed in the previous sections, many of the configurations in the table can result in sub-optimal performance. In the bottom two sections, it is worth noting the codewords of AdaBoost.ECC, which do not offer guarantees in terms of margin capacity, and the product weak learners of AdaBoost.M2 and AdaBoost.MR, which can lead to slow convergence. These problems can be compounded by the choice of loss function. In fact, a consistent experimental observation is that all algorithms derived from losses outside of the $\gamma - \phi$ family (top section of the table) tend to produce sub-optimal classifiers. In Section H.1.1, we show that the combination of the L_{ova} loss and canonical codewords, used by OVA and AdaBoost.MH, encourages the independent learning of coordinate classifiers. This frequently leads to uncalibrated scores across classes and poor classification performance. On the other hand, the losses

$$L^{\alpha}[y^c, f(x)] = e^{-\alpha \langle y^c, f(x) \rangle}, \tag{121}$$

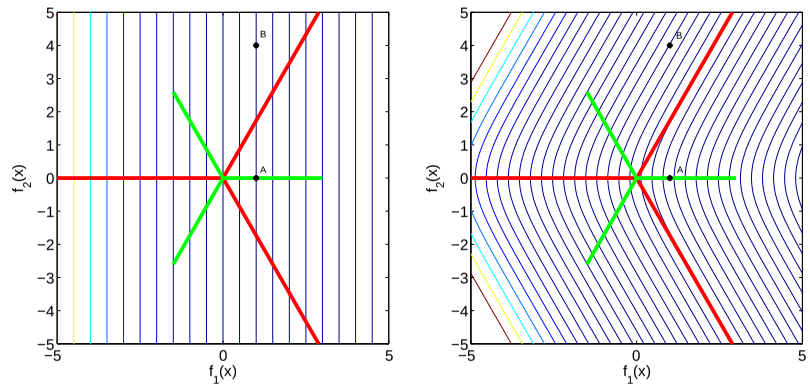


Figure 3: Loss functions $L^{\alpha}[y^1, f(x)]$ for a classification problem with $M = 3$, $d = 2$, and the max-capacity codewords of Figure 1 (center), where $y^1 = (0, 1)$. Left: exponential loss of (125). Right: $\gamma - \phi$ loss of (126). In both plots $\alpha = 1$, codewords are shown in green and class boundaries in red.

used by AdaBoost.M1 ($\alpha = 1$) and SAMME ($\alpha = 1/M$) are vaguely similar to a $\gamma - \alpha$ loss with $\gamma(v) = v$ and $\phi(v) = e^{-\alpha v}$

$$L^{\gamma-\phi}[y^c, f(x)] = \sum_{l \neq c} e^{-\alpha\left(\langle y^c, f(x) \rangle - \langle y^l, f(x) \rangle\right)}. \tag{122}$$

However, these losses do not consider the pair-wise difference between class scores. Hence, while it has been shown that, asymptotically, the minimizer of the risk defined by these losses implements Bayes rule (Zhu et al., 2009), these losses do not encourage large values of the margin of (9).

Consider, for example, the case where $M = 3$, $d = 2$, and the codeword set is the max-capacity set shown in the center of Figure 1, i.e.

$$y^1 = (1, 0) \quad y^2 = (-1/2, \sqrt{3}/2) \quad y^3 = (-1/2, -\sqrt{3}/2). \tag{123}$$

In this case, as shown in Figure 3, the class boundaries are the lines of direction

$$l^{12} = (1/2, \sqrt{3}/2) \quad l^{13} = (1/2, -\sqrt{3}/2) \quad l^{23} = (-1, 0), \tag{124}$$

where l^{ij} is the direction of the boundary between classes i and j. The loss associated with y^1 is then

$$L^{\alpha}[y^1, f(x)] = e^{-\alpha f_1(x)} \tag{125}$$

for the exponential loss and

$$L^{\gamma-\phi}[y^1, f(x)] = e^{-\alpha \langle d^{12}, f(x) \rangle} + e^{-\alpha \langle d^{13}, f(x) \rangle} \tag{126}$$

with

$$d^{12} = y^1 - y^2 = (3/2, -\sqrt{3}/2) \quad d^{13} = y^1 - y^3 = (3/2, \sqrt{3}/2). \tag{127}$$

The contour plots of these functions, shown in Figure 3, illustrate the limitations of the exponential loss. While this loss has contours orthogonal to y^1, it does not take into account the remaining codewords. In result, its contours cross the class boundaries. This implies that many classification errors (e.g. the prediction output marked as 'B') receive the *same* penalty as correct decisions (prediction marked as 'A'), which is clearly sub-optimal for classification. More generally, the loss functions of (121) produce poor classifiers whenever codewords are not aligned with the coordinate axes. This predicts poor performance for methods like SAMME. The empirical studies of (Mukherjee and Schapire, 2013) and Section 9 confirm this prediction. On the other hand, by aligning the loss contours with the decision boundaries, the $\gamma - \phi$ loss eliminates this problem. Note how it assigns a larger penalty to 'B' than 'A'.

These observations support the claim that the design of a multiclass boosting algorithm requires the careful *joint* selection of optimization strategy, codewords, weak learners, and loss function. For example, AdaBoost.M1 combines the loss of (121) and (axis-aligned) canonical codewords. However, the poor choice of step size along the gradient direction, see (236), leads to a boostability condition that demands weak learners with "better than 50%" error. Since this is unfeasible for most problems, the algorithm frequently stops prematurely. SAMME corrects this problem, but through the introduction of a set of max-capacity codewords that are no-longer axis-aligned. The mismatch between the loss of (121) and these codewords in turn compromises it performance. MCBoost eliminates these problems, supporting any max-capacity codeword set and any family of weak learners, as long as a $\gamma - \phi$ loss is used. In fact, MCBoost even establishes a connection between boosting and other machine learning algorithms, such as support vector machines (SVMs).

8.2. MCBoost and the SVM

Given a training set $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^n$ kernel support vector machines (K-SVMs) first map training examples x_i into a reproducing kernel Hilbert space (RKHS), using the mapping $\mathbf{\Phi}(x) = \mathbf{K}(x, .)$ induced by the kernel $\mathbf{K}(., .)$. The linear classifier that maximizes the margin in this space is then learned, so as to separate examples of different classes (Vapnik, 1998). As is the case for boosting, multiclass K-SVMs can be learned in multiple ways, including the reduction to several binary subproblems, as in the 'one vs all', 'all pairs' or 'ECOC' methods (Scholkopf et al., 1995; Blanz et al., 1996; Allwein et al., 2001). These are usually sub-optimal. In this work, we consider direct multiclass methods (Vapnik, 1998; Weston and Watkins, 1999; Hsu and Lin, 2002; Crammer and Singer, 2002a).

Under the direct formulation, the multiclass K-SVM solves an M-ary classification problem by learning the M linear discriminants $\mathbf{w}_l^*$, $l = 1, \ldots, M$ that maximize a risk based on the multiclass margin

$$\mathcal{M}_{K-SVM}(x_i, \mathbf{w}_{c_i}) = \frac{1}{2}\left[\langle \Phi(x_i), \mathbf{w}_{c_i}\rangle - \max_{l \neq c_i} \langle \Phi(x_i), \mathbf{w}_l\rangle\right], \tag{128}$$

so as to implement the decision rule

$$F_{SVM}(x) = \arg\max_{l=1,\ldots,M} \langle \Phi(x), \mathbf{w}_l^*\rangle. \tag{129}$$

The comparison of (128) with (9) and (129) with (17) reveals a parallel with MCBoost, where

- the kernel mapping $\Phi(x)$ of the SVM plays the role of the predictor $f(x)$ of boosting,

- the linear discriminants $\mathbf{w}_l$ of the SVM are equivalent to the class codewords y^l of boosting.

This parallel can be taken even further, by considering the SVM optimization. While several formulations have been proposed to maximize (128) (Vapnik, 1998; Weston and Watkins, 1999; Hsu and Lin, 2002; Crammer and Singer, 2002a), the most popular (Vapnik, 1998) is to find the optimal linear discriminants $\mathbf{w}_l^*$ by solving

$$\left\{\begin{array}{ll} \min_{\mathbf{w}_l,\ldots,\mathbf{w}_M} & \sum_{l=1}^M \|\mathbf{w}_l\|_2^2 + C\sum_i \xi_i \\ s.t & \langle \Phi(x_i), \mathbf{w}_{c_i}\rangle - \langle \Phi(x_i), \mathbf{w}_l\rangle \geq 1 - \xi_i \\ & \forall (x_i, c_i) \in \mathcal{D}, l \neq c_i, \\ & \xi_i \geq 0 \quad \forall i. \end{array}\right. \tag{130}$$

Rewriting the constraints as

$$\xi_i \geq \max[0, (1 - \langle \Phi(x_i), \mathbf{w}_{c_i}\rangle - \max_{l \neq c_i} \langle \Phi(x_i), \mathbf{w}_l\rangle)],$$

and using the fact that the objective function is monotonically increasing in ξ_i, this is identical to solving the problem

$$\min_{\mathbf{w}_l,\ldots,\mathbf{w}_M} \quad \overline{R}_{svm} + \lambda \sum_{l=1}^M \|\mathbf{w}_l\|_2^2, \tag{131}$$

where

$$\overline{R}_{svm} = \sum_i \left\lfloor \langle \Phi(x_i), \mathbf{w}_{c_i}\rangle - \max_{l \neq c_i} \langle \Phi(x_i), \mathbf{w}_l\rangle \right\rfloor_+, \tag{132}$$

$\lfloor x \rfloor_+ = \max(0, 1 - x)$ is the hinge loss, and $\lambda = 1/C$. Hence, the SVM minimizes the risk $\overline{R}_{svm}$ subject to a regularization constraint on $\sum_l \|\mathbf{w}_l\|_2^2$. Note that, under the parallel above, $\overline{R}_{svm}$ is the risk $\overline{R}_{L_M}(\mathbf{w})$ of (21) using the margin loss $L_M[\mathbf{w}_c, \Phi(x)]$ of (22), and the hinge loss as χ function, i.e. $\chi(v) = \lfloor v \rfloor_+$. It follows that the parallel is exact, i.e. the methods are mathematically identical, up to the replacement of $L_M[., .]$ by the $\gamma - \phi$ loss of (23).

In summary, boosting and SVMs optimize very similar costs, but with different optimization strategies. In SVM learning, examples are first mapped into the RKHS, using a *pre-defined* transformation based on the kernel. The SVM then finds a set of linear discriminants that maximizes the margin of examples in this RKHS. On the other hand, boosting pre-defines the linear discriminants as the codewords y^c. It then finds a mapping of the examples, the predictor $f(x)$, that maximizes the margin with respect to these codewords. Both the SVM discriminants and the boosting predictor are found by solving optimization problems that are equivalent, up to slight differences in loss function ($L_M[., .]$ vs. $\gamma - \phi$ loss) and regularization. While the SVM uses an explicit regularization constraint on the norm of the discriminants $\mathbf{w}_l$, boosting uses two implicit constraints on the complexity of the predictor $f(x)$: it 1) restricts $f(x)$ to a linear combination of *weak* learners and 2) limits the number of weak learners in this combination to the number of boosting iterations.

While this shows that there is a great degree of similarity between the two approaches, there is also a significant difference. This follows from the fact that it is much easier to design a good set of codewords than a good kernel. As we have shown in Section 5, under the boosting strategy it is possible to first design a set of codewords of maximum margin capacity and then find the predictor that maximizes the margin with respect to these codewords. This is unlike the SVM strategy, where it is quite difficult to determine an optimal kernel. In practice, the SVM kernel is usually chosen by cross-validation within a relatively small set of functions. In summary, while the SVM exploits a

Table 6: Characteristics of the used UCI datasets

#Id	Data Name	#Training	#Testing	#Attributes	# Classes
1	*Landsat Satellite*	4,435	2,000	36	6
2	*Letter*	16,000	4,000	16	26
3	*Pen Digit*	7,494	3,498	16	10
4	*Poker*	25,010	1,000	10	10
5	*Optical Digit*	3,823	1,797	64	10
6	*Shuttle*	43,500	14,500	9	7
7	*Isolet*	6,238	1,559	617	26
8	*Vehicle*	692	154	18	4
9	*Vertebral*	239	71	6	3
10	*Image Segmentation*	210	2,100	19	7
11	*Ecoli*	258	78	7	8
12	*Breast Tissue*	81	25	9	6

single tool for margin maximization (finding the optimal linear discriminants), boosting decouples the margin maximization into a sequence of two optimizations 1) *determination of a codeword set* (linear discriminants) of maximum margin capacity and 2) *learning of an example mapping* (predictor) of maximum margin with respect to these codewords.

9. Evaluation

Several experiments were conducted to evaluate the performance of MCBoost algorithms. Unless otherwise noted, these were implemented with $d = M - 1$, a max capacity codeword set composed by the vertexes of a regular simplex in $\mathbb{R}^{M-1}$, and the exponential $\gamma - \phi$ loss of Table 3 ($\gamma(v) = v$ and $\phi(v) = e^{-v}$)[4].

9.1. Data

The first experiment was based on a synthetic dataset, for which the optimal decision rule is known. This is a three class problem, with two-dimensional Gaussian classes of means

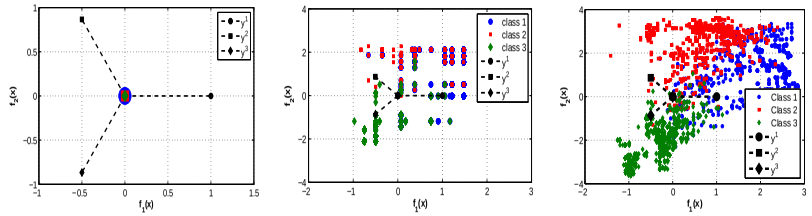
$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \end{bmatrix} \tag{133}$$

and covariances

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}, \begin{bmatrix} 0.4 & 0.1 \\ 0.1 & 0.8 \end{bmatrix} \tag{134}$$

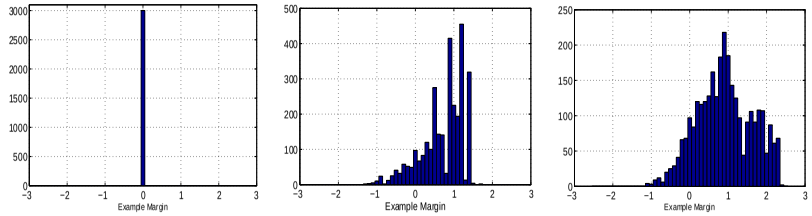
respectively. Training and test sets of $1,000$ examples each were randomly sampled and the Bayes rule computed in closed form (Duda et al., 2001). This rule achieved an error rate of 11.67% in the training and 11.13% in the test set, suggesting a Bayes error of about 11%. The remaining experiments were based on the twelve UCI datasets of Table 6. For these, we used the training/test set split provided by the dataset, whenever possible. If a split was unavailable, 20% of the examples were randomly selected for testing.

4. A Matlab implementation of CD-MCBoost and GD-MCBoost is available from http://www.svcl.ucsd.edu/publications/conference/2014/icml/ICML_2014_guess_averse_code_data.zip. A C++ implementation of GD-MCBoost for deep convolutional neural networks (Moghimi et al., 2016) integrated with the CAFFE library (Jia et al., 2014) is available from https://github.com/mmoghimi/BoostCNN.



a) $t = 0$ b) $t = 10$ c) $t = 100$

Figure 4: Classifier predictions $f^t(x_i)$ of CD-MCBoost, on the test set, after $t = 0, 10, 100$ boosting iterations.



a) $t = 0$ b) $t = 10$ c) $t = 100$

Figure 5: Histogram of the example margins $\mathcal{M}(y^{c_i}, f^t(x_i))$ using CD-MCBoost prediction, on the test set, after $t = 0, 10, 100$ boosting iterations.

9.2. Synthetic data

We start with the synthetic dataset. A classifier was learned with CD-MCBoost, using decision stumps as weak learners. Figures 4 and 5 show the predictions[5] $f^t(x_i)$ and histograms of the associated example margins $\mathcal{M}(y^{c_i}, f^t(x_i))$ for all test examples, after $t = 0, 10$, and 100 iterations, respectively. While the predictor initially maps all examples to the origin, $f^0(x_i) = [0, 0]^T \forall x_i$, CD-MCBoost produces predictions that are gradually more aligned with the class codewords, shown as dashed lines in Figure 4. Note how the example predictions, whose color reflects their class label, are pushed out, filling up the space that surrounds the codewords of the corresponding classes. This alignment of $f^t(x_i)$ with y^{c_i} guarantees that $f^t(x_i)$ will have a larger dot product with y^{c_i} than with the remaining codewords, leading to the correct classification of x_i by the decision rule of (17). Similarly, since $f^0(x_i) = [0, 0]^T \forall x_i$, all examples have zero margin in the initial iterations. However, as shown by the histograms of Figure 5, the margins $\mathcal{M}(y^{c_i}, f^t(x_i))$ increase with the boosting iteration t. Note that, as the example predictions expand in Figure 4, both the percentage of examples with positive margin and the strength of their margins increase in Figure 5. This implies that, in addition to a decrease in error rate, the decision rule generalizes better. Finally, the test set error rate was 11.30% after 100 iterations, which is very close to the estimates of the Bayes error rate of 11%.

5. We emphasize the fact that these are plots of $f^t(x) \in \mathbb{R}^2$, not $x \in \mathbb{R}^2$.

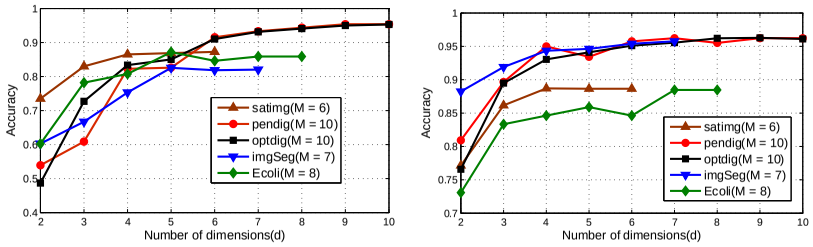


Figure 6: Effect of the dimension of codewords on the accuracy of the trained classifiers. Left: CD-MCBoost with decision stumps. Right: GD-MCBoost with trees of depth 2.

9.3. Effect of codeword dimension, d

The theoretical analysis of Section 5 suggests that there is no benefit to using a predictor dimensionality d larger than $M-1$, where M is the number of classes. Since larger dimensionality increases computation and over-fitting potential, this implies that $d = M-1$ is the best selection for the predictor dimension. In this section, we report on an empirical study of the effect of this parameter on the performance of MCBoost.

We considered 5 UCI datasets, *Landsat, Pen Digit, Optical Digit, Image Segmentation, and Ecoli,* with $M \in [6, 10]$ classes. For each dataset, we learned classifiers with predictors of dimensionality d ranging from 2 to M. A max-min codeword set was used for each value of d. Note that these codewords are identical to the optimal set for $d = M-1$. All classifiers were learned with 200 iterations of MCBoost. We considered both CD-MCBoost and GD-MCBoost, using decision stumps and trees of depth 2 as weak learners, respectively. Figure 6 shows the accuracy of the learned classifiers as a function of the dimension d, for each of the five datasets.

A number of conclusions can be drawn from the figure. First, increasing d usually leads to a better classifier, for both CD-MCBoost and GD-MCBoost. This is not surprising, since for larger d it is possible to increase the minimum mutual distance d_{min} between codewords. Since, from Lemma 2, this yields a larger margin capacity, MCBoost can produce predictors of larger margin. This explains the performance improvement. Second, since increasing d increases classification complexity, the curves of Figure 6 also depict the trade-off between accuracy and complexity of the MCBoost classifiers. Third, all curves saturate as D approaches the value of $M-1$, confirming the theoretical prediction for the lack of benefit of higher dimensions. This follows from the fact that the margin capacity does not increase beyond $d = M-1$. Finally, it is interesting that most curves saturate well below this point. Note, for example, that the GD-MCBoost performance saturates at a dimension between 50% and 70% of the number M of classes, for most datasets. This implies that it is possible to perform "dimensionality reduction on the output space," without a noticeable cost in classification accuracy. In this sense, the predictor $f(x)$ could be interpreted as an extractor of low dimensional features without loss of discriminant power. This could be interesting for applications where classification has to be performed under complexity constraints.

Table 7: Classification accuracy (%) of coordinate descent boosting algorithms with decision stump weak learners. CD-MCBoost implemented with the exponential $\gamma - \phi$ loss.

Data set id	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
AdaBoost-OVA	88.0	83.3	95.0	50.9	**95.0**	79.2	**95.3**	68.2	83.1	67.8	85.9	32.0
AdaBoost-ECC	**88.2**	77.8	94.9	**51.6**	94.8	79.2	93.6	69.5	83.1	**82.7**	84.6	**44.0**
CD-MCBoost	87.0	**84.0**	**95.4**	51.5	**95.0**	79.2	95.0	**71.4**	83.1	81.4	**85.9**	**44.0**

Table 8: Classification accuracy (%) of coordinate descent boosting algorithms with regression weak learners. CD-MCBoost implemented with the logistic $\gamma - \phi$ loss.

Data set id	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
LogitBoost	84.4	77.5	92.9	50.8	89.6	94.7	**96.3**	81.2	87.3	**93.0**	85.9	**76.0**
CD-MCBoost	**84.5**	77.5	92.9	50.8	**95.1**	**96.7**	95.6	81.2	87.3	92.3	**87.2**	72.0

9.4. Comparison to other multiclass Boosting methods

We finish with an experimental comparison of MCBoost to previous boosting algorithms, which complements the theoretical analysis of Section 8.

9.4.1. CD-MCBOOST

We started by comparing the coordinate descent (CD) methods of Table 5, namely OVA, AdaBoost.ECC (Guruswami and Sahai, 1999) and multiclass LogitBoost (Friedman et al., 1998), to CD-MCBoost. The first experiment compared CD-MCBoost to OVA and AdaBoost.ECC, using decision stumps as weak learners. As shown in Table 7, CD-MCBoost had better accuracy than OVA in six and equal accuracy in four of the twelve datasets. Compared to AdaBoost.ECC, it had better performance in six and same performance in three of the twelve datasets. Comparing all methods, CD-MCBoost produced the most accurate classifier for six of twelve datasets, the next best method was AdaBoost-ECC with 4 wins and AdaBoost-OVA had the worst performance (1 win). These results are explained by the sub-optimal codewords of AdaBoost.ECC and the loss function of OVA, which encourages independent learning of coordinate predictors.

We next compared CD-MCBoost to multiclass LogitBoost. To guarantee a fair comparison, CD-MCBoost was implemented with regression weak learners and the logistic $\gamma - \phi$ loss of Table 3. As shown in Table 8, CD-MCBoost had better accuracy in four datasets, LogitBoost was better in three and the results were identical in five datasets. The similar performance of the two classifiers is not surprising since, as discussed in Section H.1.3, they are conceptually equivalent. The only difference between LogitBoost and this implementation of CD-MCBoost is in the use of different codewords. However, since LogitBoost uses the canonical basis in $\mathbb{R}^M$ and CD-MCBoost the vertexes of a simplex in $\mathbb{R}^{M-1}$, the theory predicts identical margin capacity for the two sets. These results confirm the lack of benefit in increasing d beyond $M-1$.

Table 9: Classification accuracy (%) of gradient descent boosting algorithms with depth 2 tree weak learners. GD-MCBoost implemented with the exponential $\gamma - \phi$ loss.

Data set id	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Ada.M1	72.9	–	–	–	–	96.4	–	59.1	**88.7**	–	73.1	60.0
SAMME	82.0	53.3	90.4	52.0	91.8	99.7	85.7	76.6	84.5	93.1	87.2	**80.0**
AdaBoost.MM	89.0	64.4	95.2	59.2	95.4	99.8	91.2	81.2	87.3	94.9	84.6	68.0
GD-MCBoost	**89.1**	**85.2**	**96.5**	**60.8**	**96.8**	**99.9**	**95.2**	**81.8**	87.3	**95.3**	**89.7**	76.0

9.4.2. GD-MCBOOST

The final set of experiments compared the performance of previous gradient descent methods to GD-MCBoost. In this experiment, we considered AdaBoost.M1 (Freund and Schapire, 1996), AdaBoost.MM (Mukherjee and Schapire, 2013) and AdaBoost.SAMME (Zhu et al., 2009), using decision trees of depth 2 as weak learners. These weak learners were designed with a greedy procedure, so as to 1) minimize the weighted error rate of (235) for AdaBoost.M1 (Freund and Schapire, 1996) and AdaBoost.SAMME (Zhu et al., 2009), 2) minimize the classification cost of (266) for AdaBoost.MM (Mukherjee and Schapire, 2013), and 3) maximize (32) for GD-MCBoost.

Several conclusions can be drawn from the results, presented in Table 9. First, AdaBoost.M1 was not able to boost the weak learners used in this experiment for half of the datasets. This is due to the "better than 50% error" boostability condition, which is too stringent for trees of depth 2. Second, compared to SAMME, GD-MCBoost achieved superior performance in eleven of the twelve datasets. The improvements were quite significant in some cases e.g. from 53% to 85% in dataset $\#2$ or from 85% to 95% in dataset $\#7$. The inferior performance of SAMME is explained by the use of the loss of (121), as discussed in Section 8.1. Third, compared to AdaBoost.MM, GD-MCBoost achieved higher accuracy in eleven datasets and the same performance in the remaining one. Again, the improvements were sometimes substantial, e.g. from 64% to 85% in dataset $\#2$. This is not surprising since, as discussed in Section H.2.5, that AdaBoost.MM is a sub-optimal special case of GD-MCBoost. Finally, when compared to all methods, GD-MCBoost achieved the highest accuracy in ten of the twelve datasets. Among the remaining methods, AdaBoost.MM had better performance, followed by AdaBoost-SAMME. AdaBoost.M1 had the worst results. It should be noted that the results of Tables 7, 8 and 9 are not directly comparable, since the classifiers are based on different types of weak learners and have different complexities.

10. Conclusion

In this work, we studied the problem of multiclass boosting with the goal of an integrated understanding of the roles of the optimization strategy, label codewords, weak learners, and multiclass risk. This motivated a new formulation of the problem based on multi-dimensional predictors, multi-dimensional real valued codewords, and proper multiclass margin loss functions. This formulation led to a number of interesting results, such as maximum capacity codeword sets, proper and margin enforcing $\gamma - \phi$ losses, and two new multiclass boosting algorithms, CD-MCBoost and GD-MCBoost, which differ in optimization strategy. CD-MCBoost implements a functional coordinate descent procedure and updates one predictor component at a time, GD-MCBoost a functional gradient descent in a space of multidimensional weak learners, updating all predictor components

simultaneously. Both MCBoost algorithms reduce to classical boosting algorithms (such as AdaBoost or LogitBoost) for binary problems, depending on the choice of $\gamma - \phi$ loss and have all the classical boosting properties, such as seeking the weak learner of maximum margin on a reweighted training sample at each iteration and well defined boostability conditions. Beyond the algorithms themselves, the proposed formulation enables a unified treatment of many previous multiclass boosting algorithms. This was used to show that all algorithms implement different combinations of optimization strategy, codewords, weak learners, and loss function, highlighting some of the deficiencies of these algorithms and explaining why they fail under some settings. In particular, it was shown that no previous method matches the support of MCBoost for real codewords of maximum capacity, a proper margin-enforcing loss function, and any family of multidimensional predictors and weak learners. Experimental results confirm the superiority of MCBoost, showing that the two proposed MCBoost algorithms outperform comparable prior methods on a number of datasets.

Beyond algorithms, a number of insights were shown both theoretically and experimentally. First, the dimension d of the predictor $f(x)$ seems to play a central role in large margin multiclass classification. For M classes, any dimension larger $M-1$ supports a codeword set of maximum capacity. This enables the decomposition of the classifier design problem into two sub-problems: the design of a maximal capacity codeword set, for which an exact algorithm is available, and the design of the best predictor for this codeword set. The latter can be accomplished with the MCBoost algorithms now introduced or some other risk minimization procedure. Second, a predictor of dimension lower than $M-1$ will, in principle, entail some loss of classification accuracy. It is still, nevertheless, possible to find the best codeword set in a sense related to the margin capacity. This can again be done with an exact algorithm. Interestingly, our experimental results have shown that it is possible to achieve performance identical to that of the margin capacity predictors for dimensions substantially smaller than $M-1$. The use of a low dimensional predictor can be seen as a form of dimensionality reduction. Unlike classical approaches, which operate on the space of observations x, this is done directly in the output space. Such an interpretation of low-dimensional predictors could be of interest for applications with complexity constraints, for example. Third, for a predictor of dimension $M-1$ or larger, it is fairly simple to guarantee that a $\gamma-\phi$ loss is margin enforcing and proper. It suffices to guarantee that γ is strictly increasing and ϕ is strictly positive and decreasing, and that the codeword set is of maximum margin capacity. In this case, the posterior class probabilities can be recovered from the outputs of the optimal predictor by a simple mapping, usually a softmax. Since these constraints are quite weak, it is not difficult to define new proper multiclass margin losses, tailored for particular applications. This was exemplified by the derivation of multiclass extensions of popular binary losses, such as the exponential, logistic, or Savage. The resulting MCBoost algorithms are multiclass generalizations of AdaBoost, LogitBoost, and SavageBoost, respectively. Finally, the discussion above highlights the fact that the design of a multiclass boosting algorithm requires the careful *joint* selection of optimization strategy, codewords, weak learners, and loss function. The combination of MCBoost, margin capacity codewords, a proper margin enforcing $\gamma - \phi$ loss, and a family of weak learners with better than random performance is a good solution to all these problems.

References

E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal Machine Learning Research*, pages 113–141, 2001.

O. Beijbom, M. Saberian, D. Kriegman, and N. Vasconcelos. Guess-averse loss functions for cost-sensitive multiclass boosting. In *Proceedings of International Conference on Machine Learning*, 2014.

V. Blanz, B. Scholkopf, H. Bultho, C. Burges, V. Vapnik, and T. Vetter. Comparison of view-based object recognition algorithms using realistic 3d models. In *Artificial Neural Networks ICANN*, pages 251–256. Springer, 1996.

A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. 2006.

H. Coxeter. *Regular Polytopes*. Dover Publications, 1973.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2002a.

K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47:201–233, May 2002b.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SOCG*, pages 253–262, 2004.

T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2. edition, 2001.

G. Eibl and R. Schapire. Multiclass boosting for weak classifiers. In *Journal of Machine Learning Research*, pages 6–189, 2005.

Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of International Conference on Machine Learning*, pages 148–156, 1996.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Comp. and Sys. Science*, 1997.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28, 1998.

B. Frigyik, S. Srivastava, and M. Gupta. An introduction to functional derivatives. *Technical Report(University of Washington)*, 2008.

T. Gao and D. Koller. Multiclass boosting with hinge loss based on output coding. In *Proceedings of International Conference on Machine Learning*, 2011.

Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1–15, 2012.

Y. Guermeur. Vc theory of large margin multi-category classifiers. *Journal of Machine Learning Research*, 8:2551–2594, December 2007.

V. Guruswami and A. Sahai. Multiclass learning, boosting, and error-correcting codes. In *Proceedings of Annual Conference on Computational Learning Theory*, pages 145–155, 1999.

T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *Neural Information Processing Systems*, 1998.

Roger A. Horn and Charles R. Johnson, editors. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 1986. ISBN 0-521-30586-1.

C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

J. Huang, S. Ertekin, Y. Song, H. Zha, and C. Giles. Efficient multiclass boosting classification with active learning. In *SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2007.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *Neural Information Processing Systems*, volume 22, pages 1042–1050. 2009.

Vitaly Kuznetsov, Mehryar Mohri, and Umar Syed. Multi-class deep boosting. In *Advances in Neural Information Processing Systems 27*, pages 2501–2509. 2014.

L. Li. Multiclass boosting with repartitioning. In *Proceedings of International Conference on Machine Learning*, pages 569–576.

Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2074–2081, 2012.

H. Masnadi-Shirazi and N. Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Neural Information Processing Systems*, pages 1049–1056, 2008.

H. Masnadi-Shirazi and N. Vasconcelos. A view of margin losses as regularizers of probability estimates. *Jounal of Machine Learning Research*, 16:2751–2795, December 2015.

L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Neural Information Processing Systems*, 2000.

D. Mease and A. Wyner. Evidence contrary to the statistical view of boosting. *Journal of Machine Learning Research*, 9:131–156, June 2008.

Mohammad Moghimi, Mohammad Saberian, Jian Yang, Li-Jia Li, Nuno Vasconcelos, and Serge Belongie. Boosted convolutional neural networks. In *British Machine Vision Conference*, 2016.

I. Mukherjee and R. Schapire. A theory of multiclass boosting. *Journal of Machine Learning Research*, 14, Feb 2013.

Nils Nilsson. *Learning machines*. McGraw-Hill, New York, 1965.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer Verlag, New York, 1999.

M.D. Reid and R.C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11, September 2010.

M. Saberian, H. Masnadi-Shirazi, and N. Vasconcelos. Taylorboost: First and second order boosting algorithms with explicit margin control. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

R. Schapire. Using output codes to boost multiclass learning problems. In *Proceedings of International Conference on Machine Learning*, pages 313–321, 1997.

R. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, December 1999.

B. Scholkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *Proceedings, First International Conference on Knowledge Discovery and Data Mining, Menlo Park*, pages 252–257. AAAI Press, 1995.

T. Sejnowski and C. Rosenberg. Parallel networks that learn to pronounce english text. *Journal of Complex Systems*, 1(1):145–168, 1987.

Y. Sun, S. Todorovic, J. Li, and D. Wu. Unifying the error-correcting and output-code adaboost within the margin framework. In *Proceedings of International Conference on Machine Learning*, pages 872–879, 2005.

R. Tammes. On the origin of number and arrangement of places of exit on the surface of pallen grains. *Rec. Trav. Bot. Neerl.*, 27:1–84, 1930.

Antonio Torralba, Robert Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

V. Vapnik. *Statistical Learning Theory*. John Wiley Sons Inc, 1998.

Paul A. Viola and Michael J. Jones. Robust real-time face detection. *International Journal Computer Vision*, 57:137–154, 2004.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of European Symposium On Artificial Neural Networks*, pages 219–224, 1999.

B. Zhang, G. Ye, Y. Wang, J. Xu, and G. Herman. Finding shareable informative patterns and optimal coding matrix for multiclass boosting. In *Proceedings of International Conference on Computer Vision*, pages 56 –63, 2009.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32:56–85, 2004.

J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class adaboost. *Statistics and Its Interface*, 2:349–3660, 2009.

H. Zou, J. Zhu, and T. Hastie. New multicategory boosting algorithms based on multicategory fisher-consistent losses. *Annals of Statistics*, 2, 2008.

Appendix A. Decision rule

A.1. Proof of Lemma 1

Consider a prediction $f(x)$. Defining

$$k^* = \arg\max_{k\in\{1,..,M\}} \left\langle y^k, f(x)\right\rangle, \tag{135}$$

it follows that

$$\mathcal{M}(y^{k^*}, f(x)) \geq 0 \geq \mathcal{M}(y^l, f(x)) \quad \forall\, l \neq k^*, \tag{136}$$

and thus

$$k^* = \arg\max_{k\in\{1,..,M\}} \mathcal{M}(y^k, f(x)). \tag{137}$$

To prove the converse we start by noting that, by definition of margin in (10), for any $f(x)$ there is at least one codeword with non-negative margin. Hence, if

$$k^* = \arg\max_{k\in\{1,..,M\}} \mathcal{M}(y^k, f(x)), \tag{138}$$

it follows from (10) that

$$\left\langle y^{k^*}, f(x)\right\rangle \geq \max_{l\neq k^*} \left\langle y^l, f(x)\right\rangle, \tag{139}$$

from which

$$k^* = \arg\max_{k\in\{1,..,M\}} \left\langle y^k, f(x)\right\rangle. \tag{140}$$

A.2. Proof of Corollary 1

If $F(x) = c$, it follows from Lemma 1 that

$$c = \arg\max_{k\in\{1,..,M\}} \left\langle y^k, f(x_i)\right\rangle, \tag{141}$$

and (18) follows from (10). If $\mathcal{M}(y^c, f(x)) > 0$, it follows from (10) that

$$\langle y^c, f(x)\rangle > \max_{k\neq c} \left\langle y^k, f(x)\right\rangle. \tag{142}$$

and, from (17), $F(x) = c$.

A.3. Proof of Corollary 2

If $\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y}) > 0$, it follows from (14) that

$$\mathcal{M}(y^{c_i}, f(x_i)) > 0 \quad \forall x_i \in \mathcal{D}, \tag{143}$$

and, from Corollary 1, all examples are classified correctly. Conversely, if all examples are classified correctly, then (143) follows from Corollary 1, and

$$\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y}) = \min_{(x_i, y^{c_i})\in\mathcal{D}} \mathcal{M}(y^{c_i}, f(x_i)) > 0. \tag{144}$$

Appendix B. Algorithms

B.1. Derivation of GD-MCBoost

For the problem of (25), the gradient expression follows from the application of (27) to (21).

$$\begin{aligned}
-\delta\overline{R}[f^t;g] &= -\frac{1}{n}\frac{\partial}{\partial\epsilon}\sum_{i=1}^{n} L_M^{\gamma-\phi}[y^{c_i}, f^t(x_i)+\epsilon g(x_i)]\Big|_{\epsilon=0} \\
&= -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial L_M^{\gamma-\phi}[y^{c_i}, f^t(x_i)+\epsilon g(x_i)]}{\partial\epsilon}\Bigg|_{\epsilon=0}
\end{aligned}$$

For the $\gamma-\phi$ loss of (23) this leads to

$$\begin{aligned}
-\delta\overline{R}[f^t;g] &= -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\epsilon}\gamma\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i)+\epsilon g(x_i), y^{c_i}-y^k\right\rangle\right]\right)\Bigg|_{\epsilon=0} \\
&= -\frac{1}{n}\sum_{i=1}^{n}\gamma'\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right)\times \\
&\times \left(\sum_{k\neq c_i}\frac{\partial}{\partial\epsilon}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle+\frac{\epsilon}{2}\left\langle g(x_i), y^{c_i}-y^k\right\rangle\right]\right)\Bigg|_{\epsilon=0} \\
&= -\frac{1}{n}\sum_{i=1}^{n}\gamma'\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right)\times \\
&\times \left(\sum_{k\neq c_i}\frac{1}{2}\left\langle g(x_i), y^{c_i}-y^k\right\rangle\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right) \\
&= -\frac{1}{n}\sum_{i=1}^{n}\gamma'\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right)\left(\sum_{k\neq c_i}\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right)\times \\
&\times \left(\sum_{k\neq c_i}\frac{1}{2}\left\langle g(x_i), y^{c_i}-y^k\right\rangle\frac{\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]}{\sum_{k\neq c_i}\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]}\right).
\end{aligned}$$

Using (29) and (30), this can finally be written as

$$\begin{aligned}
-\delta\overline{R}[f^t;g] &= \frac{1}{n}\sum_{i=1}^{n} w_i\left(\sum_{k\neq c_i}\frac{1}{2}\left\langle g(x_i), y^{c_i}-y^k\right\rangle\tau_k(x_i,c_i)\right) \\
&= \frac{1}{2n}\sum_{i=1}^{n} w_i\left\langle g(x_i), \sum_{k\neq c_i}(y^{c_i}-y^k)\tau_k(x_i,c_i)\right\rangle \\
&= \frac{1}{2n}\sum_{i=1}^{n} w_i\left\langle g(x_i), y^{c_i}-\sum_{k\neq c_i}y^k\tau_k(x_i,c_i)\right\rangle.
\end{aligned}$$

B.2. Derivation of CD-MCBoost

For the problem of (35) the gradient expression follows from the application of (27) to (21),

$$\begin{aligned}
-\delta\overline{R}[f^t;j,g] &= -\frac{1}{n}\frac{\partial}{\partial\epsilon}\sum_{i=1}^{n} L_M^{\gamma-\phi}[y^{c_i}, f^t(x_i)+\epsilon g(x_i)\mathbf{1}_j]\Big|_{\epsilon=0} \\
&= -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial L_M^{\gamma-\phi}[y^{c_i}, f^t(x_i)+\epsilon g(x_i)\mathbf{1}_j]}{\partial\epsilon}\Bigg|_{\epsilon=0}
\end{aligned}$$

For the $\gamma-\phi$ loss of (23) this leads to

$$\begin{aligned}
-\delta\overline{R}[f^t;j,g] &= -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\epsilon}\gamma\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i)+\epsilon g(x_i)\mathbf{1}_j, y^{c_i}-y^k\right\rangle\right]\right)\Bigg|_{\epsilon=0} \\
&= -\frac{1}{n}\sum_{i=1}^{n}\gamma'\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right)\times \\
&\quad \left(\sum_{k\neq c_i}\frac{\partial}{\partial\epsilon}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle+\frac{1}{2}\epsilon g(x_i)\left\langle \mathbf{1}_j, y^{c_i}-y^k\right\rangle\right]\right)\Bigg|_{\epsilon=0} \\
&= -\frac{1}{n}\sum_{i=1}^{n}\gamma'\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right)\times \\
&\quad \left(\sum_{k\neq c_i}\frac{1}{2}g(x_i)\left\langle \mathbf{1}_j, y^{c_i}-y^k\right\rangle\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right) \\
&= -\frac{1}{2n}\sum_{i=1}^{n}\gamma'\left(\sum_{k\neq c_i}\phi\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right)\left(\sum_{k\neq c_i}\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]\right) \\
&\times \left(\sum_{k\neq c_i}g(x_i)\left\langle \mathbf{1}_j, y^{c_i}-y^k\right\rangle\frac{\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]}{\sum_{k\neq c_i}\phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i}-y^k\right\rangle\right]}\right).
\end{aligned}$$

Using (29) and (30), this can finally be written as

$$\begin{aligned}
-\delta\overline{R}[f^t;j,g] &= \frac{1}{2n}\sum_{i=1}^{n} w_i\left(\sum_{k\neq c_i}g(x_i)\left\langle \mathbf{1}_j, y^{c_i}-y^k\right\rangle\tau_k(x_i,c_i)\right) \\
&= \frac{1}{2n}\sum_{i=1}^{n} w_i g(x_i)\left\langle \mathbf{1}_j, \sum_{k\neq c_i}(y^{c_i}-y^k)\tau_k(x_i,c_i)\right\rangle \\
&= \frac{1}{2n}\sum_{i=1}^{n} w_i g(x_i)\left\langle \mathbf{1}_j, y^{c_i}-\sum_{k\neq c_i}y^k\tau_k(x_i,c_i)\right\rangle.
\end{aligned}$$

Appendix C. Maximum capacity codewords

C.1. Proof of Theorem 1

Consider any direction v such that $||v|| = 1$. Using (9), (55) and the fact that the minimum cannot be larger than the average

$$\begin{aligned}
\mathcal{M}(y^k, v) = \min_{l\neq k} \frac{1}{2}\left\langle y^k - y^l, v\right\rangle &\leq \frac{1}{2(M-1)}\sum_{l\neq k}\left\langle y^k - y^l, v\right\rangle && (145)\\
&= \frac{1}{2(M-1)}\left\langle \sum_{l\neq k}(y^k - y^l), v\right\rangle \\
&= \frac{1}{2(M-1)}\left\langle (M-1)y^k - \sum_{l\neq k} y^l, v\right\rangle \\
&= \frac{1}{2(M-1)}\left\langle My^k - \sum_{l=1}^{M} y^l, v\right\rangle \\
&= \frac{M}{2(M-1)}\left\langle y^k, v\right\rangle \\
&\leq \frac{M}{2(M-1)}. && (146)
\end{aligned}$$

C.2. Proof of Theorem 2

By definition of margin capacity

$$\mathcal{C}[\mathcal{Y}] = \min_{k=1,\ldots,M} \mathcal{M}(y^k, \xi^k), \quad (147)$$

where ξ^k is the direction of largest capacity for class k. As shown in the proof of Theorem 1,

$$\begin{aligned}
\mathcal{M}(y^k, \xi^k) &= \min_{l\neq k} \frac{1}{2}\left\langle y^k - y^l, \xi^k\right\rangle && (148)\\
&\leq \frac{1}{2(M-1)}\sum_{l\neq k}\left\langle y^k - y^l, \xi^k\right\rangle && (149)\\
&= \frac{M}{2(M-1)}\left\langle y^k, \xi^k\right\rangle. && (150)
\end{aligned}$$

The margin component $\mathcal{M}(y^k, \xi^k)$ is maximum when equality holds. Since the minimum is equal to the average if and only if all elements are equal, this is the case if and only if

$$\frac{1}{2}\left\langle y^k - y^l, \xi^k\right\rangle = \frac{M}{2(M-1)}\left\langle y^k, \xi^k\right\rangle \quad \forall l \neq k. \quad (151)$$

Hence the maximum value of the margin component is

$$\mathcal{M}(y^k, \xi^k) = \frac{M}{2(M-1)}\left\langle y^k, \xi^k\right\rangle \leq \frac{M}{2(M-1)} \quad (152)$$

with equality if and only if

$$\left\langle y^k, \xi^k\right\rangle = 1. \quad (153)$$

Since both y^k and ξ^k are unit vectors, this holds if and only if

$$y^k = \xi^k. \quad (154)$$

It follows that the margin capacity of (147) meets the capacity bound of (60) if and only if, for all k, the directions of largest margin are $\xi^k = y^k$ and, from (151),

$$\begin{aligned}
\frac{1}{2}\left\langle y^k - y^l, y^k\right\rangle &= \frac{M}{2(M-1)} \quad \forall l \neq k && (155)\\
\frac{1}{2}\left(1 - \left\langle y^l, y^k\right\rangle\right) &= \frac{M}{2(M-1)} \quad \forall l \neq k, && (156)\\
\left\langle y^l, y^k\right\rangle &= -\frac{1}{M-1}, \forall l \neq k. && (157)
\end{aligned}$$

C.3. Proof of Theorem 3

Assume that $\mathcal{Y}^c(M, d)$ is a set of codewords that achieves the capacity bound of (60). Then, from Theorem 2 and Definition 6, $\mathcal{Y}^c(M, d)$ is a set of M centered, unit norm, d-dimensional vectors y^k such that

$$\left\langle y^k, y^l\right\rangle = -\frac{1}{M-1}, \ \forall k, l \neq k. \quad (158)$$

The proof is by construction and uses a known method for the design of regular simplexes (Coxeter, 1973). Let y^k be the codewords in $\mathcal{Y}^c(M, d)$. Without loss of generality we can set $y^1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^d$. From (158) it follows that

$$y_1^k = -\frac{1}{M-1} \quad \forall k > 1, \quad (159)$$

where y_i^k is the ith coordinate of vector y^k. Defining

$$\bar{y}^k = \frac{M-1}{\sqrt{M(M-2)}}[y_2^{k+1}, \ldots, y_d^{k+1}] \in \mathbb{R}^{d-1} \quad k = 1, \ldots, M-1, \quad (160)$$

it follows that $\sum_{k=1}^{M-1} \bar{y}^k = 0$,

$$\begin{aligned}
\|\bar{y}^k\|^2 &= \frac{(M-1)^2}{M(M-2)}[\|y^{k+1}\|^2 - [y_1^{k+1}]^2] \\
&= \frac{(M-1)^2}{M(M-2)}\left[1 - \frac{1}{(M-1)^2}\right] \\
&= 1, \quad \forall k && (161)
\end{aligned}$$

and

$$\begin{aligned}
\left\langle \bar{y}^k, \bar{y}^l\right\rangle &= \frac{(M-1)^2}{M(M-2)}\left[\left\langle y^{k+1}, y^{l+1}\right\rangle - y_1^{k+1}y_1^{l+1}\right] \\
&= \frac{(M-1)^2}{M(M-2)}\left[-\frac{1}{M-1} - \frac{1}{(M-1)^2}\right] \\
&= -\frac{(M-1)^2}{M(M-2)}\frac{M}{(M-1)^2} = -\frac{1}{M-2}, \quad \forall k, l \neq k. && (162)
\end{aligned}$$

Hence, the codewords $\bar{y}^k$ are the elements of $\mathcal{Y}^c(M-1, d-1)$, the codeword set that meets the capacity bound of $\mathcal{S}(M-1, d-1)$. In summary, the application of the simplex design procedure of (160) to a codeword set $\mathcal{Y}^c(M, d)$ produces a codeword set $\mathcal{Y}^c(M-1, d-1)$.

If $d \leq M-2$, the procedure can be applied $d-1$ times, to produce a codeword set $\mathcal{Y}^c(M-d+1, 1)$. This is a set of $M-d+1$ centered, unit norm, scalars that satisfy (158). It follows from the unit norm constraint that $\bar{y}^k \in \{+1, -1\}$ and thus, for $k \neq l$, $\left\langle \bar{y}^k, \bar{y}^l \right\rangle = -1$. This, however, contradicts (158), since $-\frac{1}{M-d+1-1} = -\frac{1}{M-d} \geq -\frac{1}{2}$. Hence, $\mathcal{S}(M, d)$ contains no set of codewords that meets the capacity bound, when $d \leq M-2$.

If $d \geq M-1$, the procedure can be applied $M-2$ times, to produce a codeword set $\mathcal{Y}^c(2, d-M+2)$. This is a set of 2 centered, unit norm, $(d-M+2)$-dimensional vectors that satisfy (158), i.e. $\left\langle \bar{y}^1, \bar{y}^2 \right\rangle = -1$. Since $\bar{y}^1 = [1, 0, \ldots, 0]$ and $\bar{y}^2 = [-1, 0, \ldots, 0]$ in $\mathbb{R}^{d-M+2}$ satisfy these conditions, there exists a sequence $\mathcal{Y}^c(2, d-M+2), \ldots, \mathcal{Y}^c(M, d)$ of codeword sets that meet the capacity bounds of $\mathcal{S}(2, d-M+2), \ldots, \mathcal{S}(M, d)$, respectively. Since the procedure used to design this sequence is the regular simplex design procedure of (160), the codewords in $\mathcal{Y}^c(2, d-M+2), \ldots, \mathcal{Y}^c(M, d)$ form a regular simplex in $\mathbb{R}^{d-M+2}, \ldots, \mathbb{R}^d$, respectively.

Appendix D. Low dimensional predictors

D.1. Proof of Lemma 2

We start with (68). The left inequality follows from (56), (57) and (9), since

$$\begin{aligned}
\mathcal{C}[\mathcal{Y}] &= \min_{k=1,\ldots,M} \max_{||v||=1} \mathcal{M}(y^k, v) \\
&\geq \min_{k=1,\ldots,M} \mathcal{M}(y^k, y^k) \\
&= \frac{1}{2} \min_{k=1,\ldots,M} \min_{l \neq k} \left[\|y^k\| - \left\langle y^k, y^l \right\rangle \right] \\
&= \frac{1}{4} \min_{k=1,\ldots,M} \min_{l \neq k} \left[2\|y^k\| - 2\left\langle y^k, y^l \right\rangle \right] \\
&= \frac{1}{4} \min_{k=1,\ldots,M} \min_{l \neq k} \left[\|y^k\| + \|y^l\| - 2\left\langle y^k, y^l \right\rangle \right] \\
&= \frac{1}{4} \min_{k,l \neq k} \|y^k - y^l\|^2. \qquad (163)
\end{aligned}$$

The right inequality follows from (9) since, for any v such that $||v|| = 1$,

$$\begin{aligned}
\mathcal{M}(y^k, v) &= \frac{1}{2} \min_{l \neq k} \left\langle y^k - y^l, v \right\rangle \\
&\leq \frac{1}{2} \min_{l \neq k} \|y^k - y^l\|^2, \qquad (164)
\end{aligned}$$

and thus

$$\begin{aligned}
\mathcal{C}[\mathcal{Y}] &= \min_{k=1,\ldots,M} \max_{||v||=1} \mathcal{M}(y^k, v) \\
&\leq \min_{k=1,\ldots,M} \max_{||v||=1} \frac{1}{2} \min_{l \neq k} \|y^k - y^l\|^2 \\
&= \frac{1}{2} \min_{k,l \neq k} \|y^k - y^l\|^2. \qquad (165)
\end{aligned}$$

Finally, (69) follows from the left inequality of (68) and (60).

D.2. Proof of Theorem 4

Let $\mathcal{Y}^*$ be a codeword set of maximum capacity. From (66) and (62)

$$\begin{aligned}
d_{min}[\mathcal{Y}^*] &= \min_{k,l \neq k} \|(y^*)^k - (y^*)^l\|^2 \\
&= \min_{k,l \neq k} \left[2 - 2\left\langle (y^*)^k, (y^*)^l \right\rangle \right] \\
&= \min_{k,l \neq k} \left[2 + \frac{2}{M-1} \right] \\
&= \frac{2M}{M-1} \qquad (166)
\end{aligned}$$

and $\mathcal{Y}^*$ meets the bound of (69). Hence, it is a max-min distance codeword set. Let $\mathcal{Y}^*$ be a codeword set of max-min distance, i.e.

$$d_{min}[\mathcal{Y}^*] = \frac{2M}{M-1}. \qquad (167)$$

From (68) it follows that

$$\mathcal{C}[\mathcal{Y}^*] \geq \frac{M}{2(M-1)}, \qquad (168)$$

and $\mathcal{Y}^*$ meets the capacity bound of (60). Hence, it is a codeword set of maximum capacity.

Appendix E. Properties

In this appendix, we derive several properties of MCBoost algorithms.

E.1. Classification weak learners

We start by deriving the specialization of the algorithm to classification weak learners. While we consider only GD-MCBoost, a similar derivation is possible for CD-MCBoost. We assume a weak learner space $\overline{\mathcal{H}}$ of multiclass classifiers, i.e. $h : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y} = \{y^1, \ldots, y^M\}$ is the codeword set of the classification problem.

E.1.1. MAXIMUM CAPACITY CODEWORDS

Assume that $\mathcal{Y}$ is a codeword set of maximum capacity in $\mathcal{S}(M, d)$. The predictor learned by MCBoost is then

$$\begin{aligned}
f(x) &= \sum_t \alpha_t g_t(x) \\
&= \sum_t \alpha_t \sum_k I(g_t(x) = y^k) y^k \\
&= \sum_k y^k \sum_{t | g_t(x) = y^k} \alpha_t \\
&= \sum_k \zeta_k y_k \qquad (169)
\end{aligned}$$

where

$$\zeta_k = \sum_{t|g_t(x)=y^k} \alpha_t. \tag{170}$$

Since

$$\begin{aligned}
\left\langle f(x), y^j \right\rangle &= \sum_k \zeta_k < y_k, y_j > \\
&= \left(\zeta_j < y_j, y_j > + \sum_{k\neq j} \zeta_k < y_k, y_j > \right) \\
&= \left(\zeta_j - \frac{1}{M-1} \sum_{k\neq j} \zeta_k \right) \\
&= \left(\sum_{t|g_t(x)=y^j} \alpha_t - \frac{1}{M-1} \sum_{k\neq j} \sum_{t|g_t(x)=y^k} \alpha_t \right),
\end{aligned} \tag{171}$$

where we have used the codeword properties of (55) and (61), and

$$\begin{aligned}
\left\langle f(x), y^j \right\rangle - \left\langle f(x), y^k \right\rangle &= \zeta_j - \frac{1}{M-1} \sum_{l\neq j} \zeta_l - \zeta_k + \frac{1}{M-1} \sum_{l\neq k} \zeta_l \\
&= \zeta_j - \frac{1}{M-1} \zeta_k - \zeta_k + \frac{1}{M-1} \zeta_j \\
&= \frac{M}{M-1} (\zeta_j - \zeta_k).
\end{aligned} \tag{172}$$

$f(x)$ has margin components

$$\begin{aligned}
\left\langle f(x), y^c - y^k \right\rangle &= \frac{M}{M-1} (\zeta_c - \zeta_k) \\
&= \frac{M}{M-1} \left(\sum_{t|g_t(x)=y^c} \alpha_t - \sum_{t|g_t(x)=y^k} \alpha_t \right).
\end{aligned} \tag{173}$$

With regards to learning, the steepest descent direction of (32) can be written as

$$g^*(x) = \arg\max_{g\in\overline{\mathcal{H}}} \sum_{j=1}^{M} \sum_{i|g(x_i)=y^j} w_i \left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle. \tag{174}$$

There are two possibilities for the dot-products $\left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle$:

1. $y^j = y^{c_i}$: in this case

$$\begin{aligned}
\left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle &= ||y^{c_i}||^2 - \sum_{k\neq c_i} \left\langle y^{c_i}, y^k \right\rangle \tau_k(x_i, c_i) \\
&= 1 + \frac{1}{M-1} \sum_{k\neq c_i} \tau_k(x_i, c_i) \\
&= 1 + \frac{1}{M-1} = \frac{M}{M-1}
\end{aligned} \tag{175}$$

where we have also used (30).

2. $y^j \neq y^{c_i}$: in this case

$$\begin{aligned}
\left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle &= \left\langle y^j, y^{c_i} \right\rangle - ||y^j||^2 \tau_j(x_i, c_i) - \sum_{k\neq c_i, j} \left\langle y^j, y^k \right\rangle \tau_k(x_i, c_i) \\
&= -\frac{1}{M-1} - \tau_j(x_i, c_i) + \frac{1}{M-1} \sum_{k\neq c_i, j} \tau_k(x_i, c_i) \\
&= -\frac{1}{M-1} - \tau_j(x_i, c_i) + \frac{1}{M-1} (1 - \tau_j(x_i, c_i)) \\
&= -\frac{M}{M-1} \tau_j(x_i, c_i),
\end{aligned} \tag{176}$$

where we have also used (30).

It follows that

$$\begin{aligned}
g^*(x) &= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{i|g(x_i)=y^{c_i}} w_i - \sum_{j\neq c_i} \sum_{i|g(x_i)=y^j} w_i \tau_j(x_i, c_i) \\
&= \arg\max_{g\in\overline{\mathcal{H}}} \sum_i w_i - \sum_{i|g(x_i)\neq y^{c_i}} w_i - \sum_{j\neq c_i} \sum_{i|g(x_i)=y^j} w_i \tau_j(x_i, c_i) \\
&= \arg\min_{g\in\overline{\mathcal{H}}} \sum_{j\neq c_i} \sum_{i|g(x_i)=y^j} w_i (1 + \tau_j(x_i, c_i)) \\
&= \arg\min_{g\in\overline{\mathcal{H}}} \sum_{i|g(x_i)\neq y^{c_i}} w_i \left(1 + \sum_{j\neq c_i} \tau_j(x_i, c_i) I(g(x_i) = y^j) \right).
\end{aligned} \tag{177}$$

E.1.2. CANONICAL CODEWORDS

We next consider the case where $\mathcal{Y}$ is the of set canonical codewords $y^j = \mathbf{1}_j$. In this case, the predictor learned by MCBoost is

$$\begin{aligned}
f(x) &= \sum_t \alpha_t g_t(x) \\
&= \sum_t \alpha_t \sum_k I(g_t(x) = \mathbf{1}_k) \mathbf{1}_k \\
&= \sum_k \mathbf{1}_k \sum_{t|g_t(x)=\mathbf{1}_k} \alpha_t.
\end{aligned} \tag{178}$$

which can be written as

$$f_k(x) = \sum_{t|g_t(x)=\mathbf{1}_k} \alpha_t. \tag{179}$$

It follows that

$$\left\langle f(x), y^j \right\rangle = \langle f(x), \mathbf{1}_j \rangle = f_j(x) = \sum_{t|g_t(x)=\mathbf{1}_j} \alpha_t. \tag{180}$$

and

$$\left\langle f(x), y^c - y^k \right\rangle = f_c(x) - f_k(x) = \sum_{t|g_t(x)=\mathbf{1}_c} \alpha_t - \sum_{t|g_t(x)=\mathbf{1}_k} \alpha_t. \tag{181}$$

With regards to learning, the steepest descent direction of (32) can be written as

$$g^*(x) = \arg\max_{g\in\overline{\mathcal{H}}} \sum_{j=1}^{M} \sum_{i|g(x_i)=y^j} w_i \left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle. \tag{182}$$

and there are two possibilities for the dot-products $\left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle$:

1. $y^j = y^{c_i}$: in this case

$$\left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle = ||y^{c_i}||^2 - \sum_{k\neq c_i} \left\langle y^{c_i}, y^k \right\rangle \tau_k(x_i, c_i) = 1. \tag{183}$$

2. $y^j \neq y^{c_i}$: in this case

$$\begin{aligned} \left\langle y^j, y^{c_i} - \sum_{k\neq c_i} y^k \tau_k(x_i, c_i) \right\rangle &= \left\langle y^j, y^{c_i} \right\rangle - ||y^j||^2 \tau_j(x_i, c_i) - \sum_{k\neq c_i, j} \left\langle y^j, y^k \right\rangle \tau_k(x_i, c_i) \\ &= -\tau_j(x_i, c_i). \end{aligned} \tag{184}$$

It follows that

$$\begin{aligned} g^*(x) &= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{i|g(x_i)=y^{c_i}} w_i - \sum_{j\neq c_i} \sum_{i|g(x_i)=y^j} w_i \tau_j(x_i, c_i) \\ &= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{i} w_i - \sum_{i|g(x_i)\neq y^{c_i}} w_i - \sum_{j\neq c_i} \sum_{i|g(x_i)=y^j} w_i \tau_j(x_i, c_i) \\ &= \arg\min_{g\in\overline{\mathcal{H}}} \sum_{j\neq c_i} \sum_{i|g(x_i)=y^j} w_i (1 + \tau_j(x_i, c_i)) \\ &= \arg\min_{g\in\overline{\mathcal{H}}} \sum_{i|g(x_i)\neq y^{c_i}} w_i \left(1 + \sum_{j\neq c_i} \tau_j(x_i, c_i) I(g(x_i) = y^j) \right) \end{aligned} \tag{185}$$

E.2. Exponential loss

We next consider the algorithm derived from the loss function

$$L_M[\bar{y}^{c_i}, f(x_i)] = e^{-\left\langle \bar{y}^{c_i}, \bar{f}(x_i) \right\rangle}. \tag{186}$$

E.2.1. GD-MCBOOST

Using this loss in (21), it follows that

$$\begin{aligned} -\delta\overline{R}[f^t; g] &= -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial L_M[y^{c_i}, f^t(x_i) + \epsilon g(x_i)]}{\partial \epsilon} \bigg|_{\epsilon=0} \\ &= -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \epsilon} e^{-\left\langle f^t(x_i) + \epsilon g(x_i), y^{c_i} \right\rangle} \bigg|_{\epsilon=0} \\ &= \frac{1}{n} \sum_{i=1}^{n} e^{-\left\langle f^t(x_i), y^{c_i} \right\rangle} \left\langle g(x_i), y^{c_i} \right\rangle \\ &= \frac{1}{n} \sum_{i=1}^{n} w_i \left\langle g(x_i), y^{c_i} \right\rangle, \end{aligned} \tag{187}$$

with

$$w_i = e^{-\left\langle f^t(x_i), y^{c_i} \right\rangle}. \tag{188}$$

Hence, the gradient descent direction is

$$g_t(x) = \arg\max_{g\in\overline{\mathcal{H}}}, \sum_{i=1}^{n} w_i \left\langle g(x_i), y^{c_i} \right\rangle, \tag{189}$$

and the descent update for the predictor

$$f^{t+1}(x) = f^t(x) + \alpha_t g_t(x), \tag{190}$$

for a suitably chosen step size α^t.

E.2.2. MAXIMUM CAPACITY CODEWORDS

Assume that $g_t(x) \in \mathcal{Y}$, where $\mathcal{Y}$ is a codeword set of maximum capacity in $\mathcal{S}(M, d)$. Then, as shown in Section E.1.1, the predictor learned by MCBoost is

$$f(x) = \sum_t \alpha_t g_t(x) = \sum_k y^k \sum_{t|g_t(x)=y^k} \alpha_t, \tag{191}$$

and has codeword projections

$$\left\langle f(x), y^j \right\rangle = \sum_{t|g_t(x)=y^j} \alpha_t - \frac{1}{M-1} \sum_{k\neq j} \sum_{t|g_t(x)=y^k} \alpha_t. \tag{192}$$

Hence, the weights of (188) are

$$w_i = e^{-\sum_t \alpha_t I(g_t(x)=y^{c_i}) - \frac{1}{M-1} \sum_{k\neq c_i} \sum_t \alpha_t I(g_t(x)=y^k)}. \tag{193}$$

(189) can be written as

$$\begin{aligned}
g_t(x) &= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{j=1}^{M} \sum_{i=1|g_t(x_i)=y^j}^{n} w_i \left\langle y^j, y^{c_i}\right\rangle \\
&= \arg\max_{g\in\overline{\mathcal{H}}} \left\{ \sum_{i|g_t(x_i)=y^{c_i}} w_i ||y^{c_i}||^2 + \sum_{j\neq c_i} \sum_{i|g_t(x_i)=y^j} w_i \left\langle y^j, y^{c_i}\right\rangle \right\} \\
&= \arg\max_{g\in\overline{\mathcal{H}}} \left\{ \sum_{i|g_t(x_i)=y^{c_i}} w_i - \frac{1}{M-1} \sum_{i|g_t(x_i)\neq y^{c_i}} w_i \right\} \\
&= \arg\max_{g\in\overline{\mathcal{H}}} \left\{ \sum_{i} w_i - \sum_{i|g_t(x_i)\neq y^{c_i}} w_i - \frac{1}{M-1} \sum_{i|g_t(x_i)\neq y^{c_i}} w_i \right\} \\
&= \arg\min_{g\in\overline{\mathcal{H}}} \sum_{i|g_t(x_i)\neq y^{c_i}} w_i \\
&= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{i|g_t(x_i)=y^{c_i}} w_i. \qquad (194)
\end{aligned}$$

E.2.3. CANONICAL CODEWORDS

Assume that $\mathcal{Y}$ is the of set canonical codewords $y^j = \mathbf{1}_j$. In this case, as shown in Section E.1.2, the predictor learned by boosting has components

$$f_j(x) = \sum_{t|g_t(x)=\mathbf{1}_j} \alpha_t = \sum_t \alpha_t I(g_t(x) = \mathbf{1}_j), \qquad (195)$$

and codeword projections

$$\left\langle f(x), y^j \right\rangle = f_j(x) = \sum_t \alpha_t I(g_t(x) = \mathbf{1}_j). \qquad (196)$$

Hence, the weights of (188) are

$$w_i = e^{-\sum_t \alpha_t I(g_t(x)=\mathbf{1}_{c_i})}. \qquad (197)$$

(189) can be written as

$$\begin{aligned}
g_t(x) &= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{j=1}^{M} \sum_{i=1|g_t(x_i)=y^j}^{n} w_i \left\langle \mathbf{1}_j, \mathbf{1}_{c_i}\right\rangle \\
&= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{i=1|g_t(x_i)=\mathbf{1}_{c_i}}^{n} w_i \\
&= \arg\max_{g\in\overline{\mathcal{H}}} \sum_{i=1}^{n} w_i I(g_t(x_i) = \mathbf{1}_{c_i}). \qquad (198)
\end{aligned}$$

Appendix F. Margin maximization

F.1. Proof of Theorem 5

Defining $l^* = \arg\min_{l\neq c} u^c - u^l$, (23) can be written as

$$L_M^{\gamma-\phi}[y^c, f(x)] = \gamma \left[\phi\left(u^c - u^{l^*}\right) + \sum_{l\neq c,l^*} \phi\left(u^c - u^l\right) \right], \qquad (199)$$

and it follows from (9) that

$$L_M^{\gamma-\phi}[y^c, f(x)] = \gamma \left[\phi[\mathcal{M}(y^c, f(x))] \left(1 + \sum_{l\neq c,l^*} \frac{\phi(u^c - u^l)}{\phi[\mathcal{M}(y^c, f(x))]} \right) \right]. \qquad (200)$$

Since γ is strictly increasing and $\phi(.)$ is strictly positive

$$\frac{\phi(u^c - u^l)}{\phi[\mathcal{M}(y^c, f(x))]} > 0, \forall c, l, x. \qquad (201)$$

and

$$\phi[\mathcal{M}(y^c, f(x))] \left(1 + \sum_{l\neq c,l^*} \frac{\phi(u^c - u^l)}{\phi[\mathcal{M}(y^c, f(x))]} \right) > \phi[\mathcal{M}(y^c, f(x))], \forall x \qquad (202)$$

from which

$$L_M^{\gamma-\phi}[y^c, f(x)] > \gamma\left(\phi[\mathcal{M}(y^c, f(x))]\right) \qquad \forall x. \qquad (203)$$

Combining (21), (14) and the fact that $\gamma(.)$ is strictly positive

$$\begin{aligned}
\overline{R}_{L_M^{\gamma-\phi}}(f) &= \frac{1}{n} \sum_{i=1}^{n} L_M^{\gamma-\phi}[y^{c_i}, f(x_i)] \\
&> \frac{1}{n} \sum_{i=1}^{n} \gamma\left(\phi[\mathcal{M}(y^{c_i}, f(x_i))]\right) \\
&= \frac{1}{n} \gamma\left(\phi[\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y})]\right) \left[1 + \sum_{i\neq i^*} \frac{\gamma\left(\phi[\mathcal{M}(y^{c_i}, f(x_i))]\right)}{\gamma\left(\phi[\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y})]\right)} \right] \\
&> \frac{1}{n} \gamma\left(\phi[\mathcal{M}_p(\mathcal{D}, f, \mathcal{Y})]\right) \qquad (204)
\end{aligned}$$

where $i^* = \arg\min_i \mathcal{M}(y^{c_i}, f(x_i))$.

Appendix G. Proper $\gamma - \phi$ losses

G.1. Proof of Theorem 6

From (95),

$$\begin{aligned} R_{L_M^{\gamma-\phi}}(f|x) &= \sum_{k=1}^{M} \eta_k(x) L_M^{\gamma-\phi}[y^k, f(x)] \\ &= \sum_{k=1}^{M} \eta_k(x) \gamma \left(\sum_{l=1, l\neq k}^{M} \phi(u^k(x) - u^l(x)) \right). \end{aligned} \tag{205}$$

The derivative of $R_{L_M^{\gamma-\phi}}(f|x)$ with respect to $f(x)$ is

$$\begin{aligned} \frac{\partial R_{L_M^{\gamma-\phi}}(f|x)}{\partial f(x)} &= \frac{\partial}{\partial f(x)} \sum_{k=1}^{M} \eta_k \gamma \left(\sum_{l=1, l\neq k}^{M} \phi(u^k - u^l) \right) \\ &= \frac{1}{2} \sum_{k=1}^{M} \left\{ \eta_k \gamma' \left(\sum_{l\neq k} \phi(u^k - u^l) \right) \sum_{l\neq k} \phi'(u^k - u^l)[y^k - y^l] \right\}. \end{aligned} \tag{206}$$

Defining

$$\pi_k = \gamma' \left(\sum_{l\neq k} \phi(u^k - u^l) \right), \tag{207}$$

results in

$$\begin{aligned} \frac{\partial R_{L_M^{\gamma-\phi}}(f|x)}{\partial f(x)} &= \frac{1}{2} \sum_{k=1}^{M} \left\{ \eta_k \pi_k \sum_{l\neq k} \phi'(u^k - u^l)[y^k - y^l] \right\} \\ &= \frac{1}{2} \sum_{l,k|k\neq l} \eta_k \pi_k \phi'(u^k - u^l)[y^k - y^l] \\ &= \frac{1}{2} \sum_{l,k|k\neq l} y^k \eta_k \pi_k \phi'(u^k - u^l) - \frac{1}{2} \sum_{l,k|k\neq l} y^l \eta_k \pi_k \phi'(u^k - u^l) \\ &= \frac{1}{2} \sum_{j,k|k\neq j} y^k \eta_k \pi_k \phi'(u^k - u^j) - \frac{1}{2} \sum_{l,k|k\neq l} y^k \eta_l \pi_l \phi'(u^l - u^k) \\ &= \frac{1}{2} \sum_{k=1}^{M} y^k \eta_k \pi_k \sum_{j\neq k} \phi'(u^k - u^j) - \frac{1}{2} \sum_{k=1}^{M} y^k \sum_{l\neq k} \eta_l \pi_l \phi'(u^l - u^k) \\ &= \frac{1}{2} \sum_{k=1}^{M} y^k \left[\eta_k \pi_k \sum_{j\neq k} \phi'(u^k - u^j) - \sum_{l\neq k} \eta_l \pi_l \phi'(u^l - u^k) \right] \\ &= \frac{1}{2} \mathbf{Y} \mathbf{Q}_f^{\phi} \mathbf{\Gamma}_f^{\gamma-\phi} \eta. \end{aligned} \tag{208}$$

where $\mathbf{Y}$ is the code matrix of (94), $\mathbf{Q}_f^{\phi}$ as defined in (97) and $\Gamma_f^{\gamma-\phi}$ as defined in (98). Hence, (96) holds for any minimizer of $R_{L_M^{\gamma-\phi}}(f|x)$.

G.2. Proof of Lemma 3

1. If γ is strictly monotonic then, from (98), $\mathbf{\Gamma}_f^{\gamma-\phi}$ is full rank and the statement follows.

2. Let $Rank(\mathbf{Y}) < M - 1$ and consider two possibilities.

   (a) The set $Null(\mathbf{Y}) \bigcap Range(\mathbf{Q}_f^{\phi})$ is empty. In this case, since $Null(\mathbf{Y}) \bigcup Range(\mathbf{Q}_f^{\phi}) \subset \mathbb{R}^M$ and

   $$\begin{aligned} M &\geq |Null(\mathbf{Y})| + |Range(\mathbf{Q}_f^{\phi})| \\ &= M - Rank(\mathbf{Y}) + |Range(\mathbf{Q}_f^{\phi})| \\ &\geq 2 + |Range(\mathbf{Q}_f^{\phi})|. \end{aligned} \tag{209}$$

   Since $\mathbf{Q}_f^{\phi} \in \mathbb{R}^{M\times M}$, it follows that $|Null(\mathbf{Q}_f^{\phi})| \geq 2$. Since $Null(\mathbf{Q}_f^{\phi}) \subset Null(\mathbf{Y}\mathbf{Q}_f^{\phi})$, it follows that $|Null(\mathbf{Y}\mathbf{Q}_f^{\phi})| \geq 2$.

   (b) The set $Null(\mathbf{Y}) \bigcap Range(\mathbf{Q}_f^{\phi})$ is non-empty. Hence, there is at least one vector $v_1 \in Null(\mathbf{Y}) \bigcap Range(\mathbf{Q}_f^{\phi})$. Since $v_1 \in Range(\mathbf{Q}_f^{\phi})$, there exists a vector v_2 such that $\mathbf{Q}_f^{\phi} v_2 = v_1$, i.e. $v_2 \notin Null(\mathbf{Q}_f^{\phi})$. Since $v_1 \in Null(\mathbf{Y})$, it follows that $\mathbf{Y}\mathbf{Q}_f^{\phi} v_2 = 0$ and $v_2 \in Null(\mathbf{Y}\mathbf{Q}_f^{\phi})$. On the other hand, it follows from (97) that the rows of $\mathbf{Q}_f^{\phi}$ sum to zero and

   $$|Null(\mathbf{Q}_f^{\phi})| \geq 1. \tag{210}$$

   Hence, there is at least a vector $v_0 \neq 0$ such that $\mathbf{Q}_f^{\phi} v_0 = 0$. It follows that $v_0 \in Null(\mathbf{Y}\mathbf{Q}_f^{\phi})$. In summary, there is a vector $v_0 \in Null(\mathbf{Q}_f^{\phi})$ and a vector $v_2 \notin Null(\mathbf{Q}_f^{\phi})$ such that $v_0, v_2 \in Null(\mathbf{Y}\mathbf{Q}_f^{\phi})$. Hence, $|Null(\mathbf{Y}\mathbf{Q}_f^{\phi})| \geq 2$.

   By combination of the two possibilities it follows that $|Null(\mathbf{Y}\mathbf{Q}_f^{\phi})| \geq 2$.

3. Since $\mathbf{Y} \in \mathbb{R}^{d\times M}$, $Rank(\mathbf{Y}) \leq M$. If $Rank(\mathbf{Y}) = M$, then the null space of $\mathbf{Y}$ contains only the origin. It follows that a vector is in the null space of $\mathbf{Y}\mathbf{Q}_f^{\phi}$ if and only if it is also on the null space of $\mathbf{Q}_f^{\phi}$. Assume that $Rank(\mathbf{Y}) = M - 1$. Since $\mathbf{Y} \in \mathbb{R}^{d\times M}$, $|Null(\mathbf{Y})| = 1$. Since the codewords are centered, as in (55), $\mathbf{Y}\mathbf{1} = 0$ and $Null(\mathbf{Y}) = Range(\mathbf{1})$, i.e. the null space of $\mathbf{Y}$ is spanned by the vector $\mathbf{1}$. Hence, for any $\eta \in Null(\mathbf{Y}\mathbf{Q}_f^{\phi})$ there is a scalar λ such that

   $$\mathbf{Q}_f^{\phi} \eta = \lambda \mathbf{1}. \tag{211}$$

   Denoting by r_Q^i the ith row of $\mathbf{Q}_f^{\phi}$, it follows that $< r_Q^i, \eta >= \lambda$ and $< \sum_{i=1}^{M} r_Q^i, \eta >= M\lambda$. Since, from (97), $\sum_{i=1}^{M} r_Q^i = 0$ it follows that $\lambda = 0$. Hence, $\eta \in Null(\mathbf{Q}_f^{\phi})$ and $Null(\mathbf{Y}\mathbf{Q}_f^{\phi}) \subset Null(\mathbf{Q}_f^{\phi})$. The statement follows from the fact that $Null(\mathbf{Q}_f^{\phi}) \subset Null(\mathbf{Y}\mathbf{Q}_f^{\phi})$ always holds.

4. Let $\overline{\mathbf{Q}}_f^\phi \in \mathbb{R}^{M-1\times M-1}$ be the matrix obtained by eliminating the first row and column of $\mathbf{Q}_f^\phi$. From (97) and strict monotonicity of ϕ

$$\begin{aligned} |\mathbf{Q}_f^\phi(k,k)| &= \left|\sum_{j\neq k}\phi'(u^k-u^j)\right| \\ &= \sum_{j\neq k}|\phi'(u^k-u^j)|, \end{aligned} \tag{212}$$

and

$$\begin{aligned} |\overline{\mathbf{Q}}_f^\phi(k,k)| &= \sum_{j\neq k+1}|\phi'(u^{k+1}-u^j)| \\ &= \sum_{j\neq k+1}|\mathbf{Q}_f^\phi(j,k+1)| \\ &= |\mathbf{Q}_f^\phi(1,k+1)| + \sum_{j\neq k}|\overline{\mathbf{Q}}_f^\phi(j,k)| \\ &> \sum_{j\neq k}|\overline{\mathbf{Q}}_f^\phi(j,k)|. \end{aligned} \tag{213}$$

Therefore $\overline{\mathbf{Q}}_f^\phi$ is strictly diagonally dominant and thus non-singular (Horn and Johnson, 1986). It follows that rows $r_Q^i, i=2,\ldots,M$ of $\mathbf{Q}_f^\phi$ are linearly independent and $Rank(\mathbf{Q}_f^\phi)\geq M-1$. Since, from (97),

$$r_Q^1 = -\sum_{i=2}^M r_Q^i, \tag{214}$$

it follows that $Rank(\mathbf{Q}_f^\phi)=M-1$. Hence, $|Null(\mathbf{Q}_f^\phi)|=1$.

G.3. Proof of Lemma 4

From (97), (98) and (100), $\mathbf{Q}_f^\phi\Gamma_f^{\gamma-\phi}\eta=0$ is equivalent to

$$\begin{aligned} 0 &= \eta_k\pi_k\sum_{j\neq k}\phi'(u^k-u^j) - \sum_{l\neq k}\eta_l\pi_l\phi'(u^l-u^k) \\ &= \eta_k\pi_k\sum_{j\neq k}\phi'(u^k-u^j) + \eta_k\pi_k\phi'(u^k-u^k) \\ &- \eta_k\pi_k\phi'(u^k-u^k) - \sum_{l\neq k}\eta_l\pi_l\phi'(u^l-u^k) \\ &= \eta_k\pi_k\sum_{j=1}^M\phi'(u^k-u^j) - \sum_{l=1}^M\eta_l\pi_l\phi'(u^l-u^k), \end{aligned} \tag{215}$$

and (99) follows. The proof that all η_k have the same sign is by contradiction. Assume the η_j have different signs and let $\mathcal{N},\mathcal{P}$ be sets of positive and negative indices, i.e. $\eta_j<0\ \forall j\in\mathcal{N}$, $\eta_j\geq 0\ \forall j\in\mathcal{P}$. Define

$$\zeta_{k,l} = \pi_k\phi'(u^k-u^l), \tag{216}$$

with π_k as in (100). Since ϕ,γ are strictly monotonic, all $\zeta_{k,l}$'s are non-zero and have the same sign. Without loss of generality, we assume they are positive. From (99),

$$\eta_k\sum_{l=1}^M\zeta_{k,l} - \sum_{l=1}^M\eta_l\zeta_{l,k} = 0. \tag{217}$$

Adding over all $k\in\mathcal{N}$,

$$\begin{aligned} 0 &= \sum_{k\in\mathcal{N}}\left\{\eta_k\sum_{l=1}^M\zeta_{k,l} - \sum_{l=1}^M\eta_l\zeta_{l,k}\right\} \\ &= \sum_{k\in\mathcal{N}}\eta_k\left\{\sum_{l\in\mathcal{N}}\zeta_{k,l}+\sum_{l\in\mathcal{P}}\zeta_{k,l}\right\} - \sum_{k\in\mathcal{N}}\left\{\sum_{l\in\mathcal{N}}\eta_l\zeta_{l,k}+\sum_{l\in\mathcal{P}}\eta_l\zeta_{l,k}\right\} \\ &= \sum_{k\in\mathcal{N},l\in\mathcal{N}}\eta_k\zeta_{k,l} + \sum_{k\in\mathcal{N},l\in\mathcal{P}}\eta_k\zeta_{k,l} - \sum_{k\in\mathcal{N},l\in\mathcal{N}}\eta_l\zeta_{l,k} - \sum_{k\in\mathcal{N},l\in\mathcal{P}}\eta_l\zeta_{l,k} \\ &= \sum_{k\in\mathcal{N},l\in\mathcal{P}}\eta_k\zeta_{k,l} - \sum_{k\in\mathcal{N},l\in\mathcal{P}}\eta_l\zeta_{l,k} \end{aligned} \tag{218}$$

and

$$\sum_{k\in\mathcal{N},l\in\mathcal{P}}\eta_k\zeta_{k,l} = \sum_{k\in\mathcal{P},l\in\mathcal{N}}\eta_k\zeta_{k,l}. \tag{219}$$

Since by definition of the sets $\mathcal{N},\mathcal{P}$, the two sides of this equation have opposite signs, we have a contradiction unless one of the sets $\mathcal{N},\mathcal{P}$ is empty.

G.4. Proof of theorem 7

From Lemmas 3 and 4, since $Rank(\mathbf{Y})\geq M-1$ and (ϕ,γ) strictly monotonic, the solution of (96) is identical to that of $\mathbf{Q}_f^\phi\mathbf{\Gamma}_f^{\gamma-\phi}\eta=0$ and, up to a normalization constant, a probability vector. Defining $\mathbf{\Psi}_f^{\gamma-\phi}=\mathbf{Q}_f^\phi\mathbf{\Gamma}_f^{\gamma-\phi}$, it follows that the system of equations

$$\left\{\begin{array}{l}\mathbf{\Psi}_f^{\gamma-\phi}\eta=0\\ \mathbf{1}^T\eta=1,\end{array}\right. \tag{220}$$

has a unique solution. Furthermore, it follows from the monotonicity of γ that $Rank(\mathbf{\Gamma}_f^{\gamma-\phi})=M$ and from (97) that $Rank(\mathbf{Q}_f^\phi)=M-1$. Hence, $Rank(\mathbf{\Psi}_f^{\gamma-\phi})=M-1$, i.e. the first row of $\mathbf{\Psi}_f^{\gamma-\phi}$ is a linear combination of the other rows and can be removed from (220). The system of equations of (220) can thus be written as

$$\mathbf{\Omega}_f^{\gamma-\phi}\eta=\mathbf{e}_1, \tag{221}$$

where

$$\mathbf{\Omega}_f^{\gamma-\phi}(k,l)=\left\{\begin{array}{ll}1 & k=1,\forall l\\ \mathbf{\Psi}_f^{\gamma-\phi}(k,l) & \text{otherwise.}\end{array}\right. \tag{222}$$

Since (220) has a unique solution, $\mathbf{\Omega}_f^{\gamma-\phi}$ is invertible and $\eta=\left(\mathbf{\Omega}_f^{\gamma-\phi}\right)^{-1}\mathbf{e}_1$.

Appendix H. MCBoost vs previous multiclass boosting algorithms

In this appendix we compare MCBoost to multiclass learning algorithms in the literature. We start by showing that many of the multiclass boosting algorithms are special instances of MCBoost using different, and sometimes sub-optimal, codewords, weak learners, loss functions, or optimization procedures.

H.1. Coordinate descent algorithms

Several previous learning algorithms can be seen as coordinate descent optimization procedures and are thus closely related to CD-MCBoost.

H.1.1. ONE-VS-ALL CLASSIFIERS

Multiclass classifiers are frequently implemented with the one-vs-all architecture (OVA), where M predictors $\bar{f}_k(x)$ are learned independently. Predictor $\bar{f}_k(x)$ discriminates between class k and all other classes. For predictors learned with AdaBoost, $\bar{f}_k(x)$ is learned by minimizing the risk

$$\overline{R}_k = \frac{1}{n}\sum_{i=1}^{n} e^{-\bar{z}_i^k \bar{f}_k(x_i)}, \tag{223}$$

where

$$\bar{z}_i^k = \begin{cases} 1 & \text{if } c_i = k \\ -1 & \text{otherwise.} \end{cases} \tag{224}$$

The M OVA predictors are combined into a multiclass predictor $\bar{f}(x) = [\bar{f}_1(x), \ldots, \bar{f}_M(x)] \in \mathbb{R}^M$, which is used to implement the decision rule $\arg\max_k \bar{f}_k(x)$.

Since this predictor is learned in a coordinate-wise manner, OVA is closest to CD-MCBoost. In fact, the individual risks of (223) can be combined into

$$\begin{aligned} \overline{R} &= \sum_k \overline{R}_k = \frac{1}{n}\sum_{i=1}^{n}\sum_k e^{-\bar{z}_i^k \bar{f}_k(x_i)} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left\{ e^{-\bar{f}_{c_i}(x_i)} + \sum_{k\neq c_i} e^{\bar{f}_k(x_i)} \right\}, \end{aligned}$$

which is the risk of (21) for the loss

$$L_{ova}[y^{c_i}, \bar{f}(x_i)] = e^{-\langle \bar{y}^{c_i}, \bar{f}(x_i)\rangle} + \sum_{k\neq c_i} e^{\langle \bar{y}^k, \bar{f}(x_i)\rangle}, \tag{225}$$

under the choice of canonical codewords $\bar{y}^j = \mathbf{1}_j \in \mathbb{R}^M$. Hence, the fundamental differences between OVA and CD-MCBoost are the use, by the former, of 1) canonical codewords and 2) the loss of (225). This combination is sub-optimal for two reasons. First, the minimization of the resulting risk decouples into M *independent* problems for learning M coordinate classifiers. This tends to produce weaker decision rules than the joint learning of all coordinate predictors. Second, the loss of (225) is not guess-averse (Beijbom et al., 2014). This can easily lead to sub-optimal classifiers.

H.1.2. ADABOOST.ECC

AdaBoost.ECC (Guruswami and Sahai, 1999) is a multiclass boosting algorithm based on the the error correcting output coding (ECOC) strategy of (Dietterich and Bakiri, 1995). It relies on binary-valued codewords and the $\gamma - \phi$ loss of (23) with $\phi(v) = e^{-2v}$ and $\gamma(v) = v$. It is initialized with an empty codeword set. At each iteration t, 1) each codeword is augmented with a new bit and 2) a corresponding predictor dimension $\bar{f}_t$ is learned. After T iterations, the algorithm learns a set of T-dimensional binary codewords $\bar{y}^j \in \{+1, -1\}^T$ and a T-dimensional predictor $\bar{f}(x) = [\bar{f}_1(x), \ldots, \bar{f}_T(x)] \in \mathbb{R}^T$. The decision rule is

$$\bar{F}(x) \equiv \arg\max_{j\in\{1,\ldots,M\}} \left\langle \bar{f}(x), \bar{y}^j \right\rangle. \tag{226}$$

Since the multi-dimensional predictor is learned in a coordinate-wise manner, AdaBoost.ECC is closest to CD-MCBoost. The main difference is in the definition of the codewords. While CD-MCBoost uses real-valued codewords selected a priori so as to maximize the margin capacity of (56), AdaBoost.ECC codewords are generated on-the-fly, by either random selection or the solution of a "max-cut" problem. This is sub-optimal for two reasons. First, the limitation to binary codewords can lead to codeword sets of margin capacity below the bound of (59). This limits the margin maximizing ability of the learning algorithm and degrades its generalization. Second, the codeword length increases at each iteration of the algorithm. Since this increases predictor dimensionality, it can easily lead to high computational costs and over-fitting. Finally, the procedures used to learn the codewords do not guarantee a codeword set optimal for classification. These limitations extend to a number of other methods based on the ECOC strategy (Schapire, 1997; Li; Zhang et al., 2009; Gao and Koller, 2011).

H.1.3. MULTICLASS LOGITBOOST

Multiclass LogitBoost (Friedman et al., 1998) is a multiclass boosting method that learns an M-dimensional predictor $\bar{f}(x) = [\bar{f}_1(x), \ldots, \bar{f}_M(x)] \in \mathbb{R}^M$ where each coordinate is a linear regressor

$$\bar{f}_j(x) = ax + b, \tag{227}$$

for some $a, b \in \mathbb{R}$. It is assumed that

$$\sum_{k=1}^{M} \bar{f}_k(x) = 0. \tag{228}$$

The predictor is learned by minimizing the negative log likelihood,

$$L_{Logit}[c_i, \bar{f}(x_i)] = -\log\left[\bar{P}_{C,X}(c_i|x_i)\right], \tag{229}$$

with probabilities defined as

$$\bar{P}_{C,X}(c_i|x_i) = \frac{e^{\bar{f}_{c_i}(x_i)}}{\sum_{j=1}^{M} e^{\bar{f}_j(x_i)}}. \tag{230}$$

Each iteration of the algorithm computes the best regression update for all coordinates, using Newton's method. These updates are then centered to satisfy (228) and added to the multi-dimensional predictor.

To compare this to MCBoost note that, from (229)-(230),

$$\begin{aligned} L_{Logit}[c_i, \bar{f}(x_i)] &= -\log\left[\frac{e^{\bar{f}_{c_i}(x_i)}}{\sum_{j=1}^M e^{\bar{f}_j(x_i)}}\right] \\ &= \log\left[\frac{\sum_{j=1}^M e^{\bar{f}_j(x_i)}}{e^{\bar{f}_{c_i}(x_i)}}\right] \\ &= \log\left[1+\sum_{j\neq c_i} e^{\bar{f}_j(x_i)-\bar{f}_{c_i}(x_i)}\right] \\ &= \log\left[1+\sum_{j\neq c_i} e^{-\left[\langle \bar{y}^{c_i},\bar{f}(x_i)\rangle - \langle \bar{y}^{j},\bar{f}(x_i)\rangle\right]}\right], \end{aligned} \tag{231}$$

with $\bar{y}^j = \mathbf{1}_j \in \mathbb{R}^M$. Since this is the special case of (23) with $\phi(v) = e^{-2v}$ and $\gamma(v) = \log(1+v)$, multiclass LogitBoost is an implementation of CD-MCBoost with the logistic $\gamma - \phi$ loss, canonical codewords, and regression weak learners. The only difference is that LogitBoost relies on the Newton method for finding the best update per iteration while CD-MCBoost uses gradient descent updates. Although it is possible to use Newton updates in MCBoost (Saberian et al., 2010), this is beyond the scope of the current manuscript. Finally, note that the analysis above holds for a number of algorithms inspired by multiclass LogitBoost (Huang et al., 2007; Zou et al., 2008).

H.2. Gradient descent algorithms

Several previous learning algorithms can be seen as gradient descent optimization procedures and are thus closely related to GD-MCBoost.

H.2.1. ADABOOST.M1

AdaBoost.M1 is the first multiclass Boosting algorithm introduced by (Freund and Schapire, 1996). It relies on multiclass weak learners $\hat{g}(x) : \mathcal{X} \to \{1, \ldots, M\}$ and the decision rule

$$\bar{F}(x) = \arg\max_{j\in\{1,\ldots,M\}} \hat{f}_j(x_i), \tag{232}$$

where

$$\hat{f}_j(x_i) = \sum_t \alpha_t I(\hat{g}_t(x_i) = j), \tag{233}$$

and $I(.)$ is the indicator function. AdaBoost.M1 is initialized with a uniform weight distribution over the training sample, e.g. $w_i = 1$. At iteration t, it selects the weak learner $\hat{g}_t$ of lowest weighted error rate

$$\begin{aligned} e_t^* &= \min_g \sum_i w_i[1 - I(g(x_i) = c_i)] && (234) \\ &= \max_g \sum_i w_i I(g(x_i) = c_i). && (235) \end{aligned}$$

The algorithm stops if $e_t^* > 50\%$, otherwise the weak learner is added to the ensemble with

$$\alpha_t = \log\left(\frac{1-e_t^*}{e_t^*}\right), \tag{236}$$

and the weights are updated according to

$$w_i = w_i \times e^{-\alpha_t I(\hat{g}_t(x_i)=c_i)} = e^{-\sum_t \alpha_t I(\hat{g}_t(x_i)=c_i)}. \tag{237}$$

In Appendix E.2.3 we show that (233), (235), and (237) are the equations of a GD-MCBoost algorithm for the minimization of the risk defined by the loss

$$L_{Ada.M1}[\bar{y}^{c_i}, \bar{f}(x_i)] = e^{-\langle \bar{y}^{c_i},\bar{f}(x_i)\rangle} \tag{238}$$

with $\bar{y}^j = \mathbf{1}_j \in \mathbb{R}^M$ and $\bar{f} = [\hat{f}_1, \ldots, \hat{f}_M]$ where $\hat{f}_j$ is given by (233). Hence, AdaBoost.M1 is a descent algorithm using the loss of (238), canonical codewords, and classification weak learners. While this has some similarity with GD-MCBoost, there are important differences. First, (238) is not a $\gamma - \phi$ loss. Second, the constraint $\alpha_t > 0$ requires the existence of a weak learner with error rate smaller than 50% for any weight distribution. As discussed above, this is usually too stringent for problems with large M. In practice, AdaBoost.M1 frequently terminates too early, due to the impossibility of finding such a weak learner (Zhu et al., 2009).

H.2.2. SAMME

SAMME (Zhu et al., 2009) is a multiclass Boosting algorithm explicitly introduced to address this problem. It learns an M-dimensional predictor $\bar{f} = [\bar{f}_1, \ldots, \bar{f}_M] \in \mathbb{R}^M$ such that $\sum_{k=1}^M \bar{f}_k(x) = 0$, using codewords

$$\bar{y}^j = \frac{M\mathbf{1}_j - \mathbf{1}}{M-1} = \left[\frac{-1}{M-1}, \frac{-1}{M-1}, \ldots, 1, \frac{-1}{M-1}, \frac{-1}{M-1}\right] \in \mathbb{R}^M, \tag{239}$$

and multiclass classification weak learners $\hat{g}(x) : \mathcal{X} \to \{\bar{y}^1 \ldots \bar{y}^M\}$. The boosting algorithm is derived explicitly to minimize the loss

$$L_{SAMME}[\bar{y}^{c_i}, f(x_i)] = e^{-\frac{1}{M}\langle \bar{y}^{c_i},\bar{f}(x_i)\rangle}. \tag{240}$$

using updates of the form of (33) and a decision rule

$$\begin{aligned} \bar{F}(x) &= \arg\max_{j\in\{1,\ldots,M\}} \bar{f}_j(x) && (241) \\ &\equiv \arg\max_{j\in\{1,\ldots,M\}} \langle \bar{y}^j, \bar{f}(x)\rangle && (242) \end{aligned}$$

with $\bar{y}^j = \mathbf{1}_j \in \mathbb{R}^M$. Appendix E.2.2 shows that this leads to the weak learner selection rule of (235), making the updates of SAMME nearly identical to those of AdaBoost.M1. The only difference is the step size α_t. (Zhu et al., 2009) have shown that the optimal step is

$$\alpha_t = \log\left(\frac{1-e_t^*}{e_t^*}\right) + \log(M-1). \tag{243}$$

When compared to (236), the addition of the term $\log(M-1)$ enables the algorithm to continue running when $e_t^* < 50\%$. This substantially loosens the boostability condition, enabling the use of much weaker weak learners. Comparing to GD-MCBoost, SAMME is a gradient descent procedure, uses a maximum capacity codeword set, and classification weak learners. There are, however, two main differences. The first is the use of the exponential loss of (240) instead of the $\phi - \gamma$ loss of (23). These losses are compared in Section 8.1. The second is the use of different codeword sets for learning the predictor (maximum capacity) and computing the decision rule (canonical). This inconsistency is likely to degrade classification accuracy.

H.2.3. ADABOOST.M2 AND ADABOOST.MR

AdaBoost.MR (Schapire and Singer, 1999) is a multiclass boosting algorithm that supports multiple labels per example. Its single-label version is AdaBoost.M2 (Freund and Schapire, 1996). Like SAMME, it was introduced to increase the boostability of AdaBoost.M1. The extension is, however, of a different nature. The idea behind AdaBoost.MR and AdaBoost.M2 is to define a distribution over mislabels and a pseudo-loss of weak learner h_t with respect to this distribution. The weak learners are defined on the product space of input and class labels

$$g_t : \mathcal{X} \times \{1, \ldots, M\} \to \mathbb{R} \tag{244}$$

and denoted *product weak learners*. The goal is to learn a decision rule

$$\bar{F}(x) = \arg\max_{j \in \{1,\ldots,M\}} \sum_t \alpha_t \hat{g}_t(x, j). \tag{245}$$

AdaBoost.MR defines a *weight per class* for each training example, $w(x_i, k) \quad k = 1, \ldots, M$. This can be seen as a weight distribution over classes and is initially uniform, i.e. $w(x_i, k) = 1, \forall k$. At each iteration, the weak learner $\hat{g}_t(.,.)$ of lowest pseudo loss

$$e_t = \frac{1}{2} \sum_{(i,y) \in B} w(x_i, y)(1 - g(x_i, y_i) + g(x_i, y)), \tag{246}$$

where $B = \{(i, y) | y \neq y_i\}$ is the set of mislabels, is selected and added to the current predictor with coefficient

$$\alpha_t = \log\left(\frac{1 - e_t}{e_t}\right). \tag{247}$$

The weights are updated according to

$$w(x_i, k) = w(x_i, k) \times e^{-\frac{\alpha_t}{2}[\hat{g}_t(x_i, y_i) - \hat{g}_t(x_i, k)]}. \tag{248}$$

To compare AdaBoost.M2 to MCBoost, we start by defining the class-specific predictors

$$\hat{f}_j = \sum_t \alpha_t \hat{g}_t(x, j), \tag{249}$$

the multi-dimensional predictor $\bar{f}(x) = [\hat{f}(x_i, 1), \ldots, \hat{f}(x_i, M)]$ and the multi-dimensional weak learner $g(x) = [g(x, 1), \ldots, g(x, M)]$. The decision rule of (245) is then equivalent to

$$\bar{F}(x) = \arg\max_{j \in \{1,\ldots,M\}} \left\langle \bar{y}^j, \bar{f}(x) \right\rangle, \tag{250}$$

with canonical codewords $\bar{y}^j = \mathbf{1}_j \in \mathbb{R}^M$ and the weights of (248) reduce to

$$w(x_i, j) = e^{-\frac{1}{2}\left[\left\langle \bar{y}^{c_i}, \bar{f}(x_i) \right\rangle - \left\langle \bar{y}^j, \bar{f}(x_i) \right\rangle\right]}. \tag{251}$$

The pseudo-loss of (246) can also be written as

$$\begin{aligned} e_t &= \frac{1}{2} \sum_i \sum_{j \neq c_i} w(x_i, j)(1 - \langle g(x_i), \bar{y}^{c_i} \rangle + \langle g(x_i), \bar{y}^j \rangle) \\ &= \frac{1}{2} \sum_i w_i \left(1 - \sum_{j \neq c_i} \frac{w(x_i, j)}{w_i} \left\langle g(x_i), \bar{y}^{c_i} - \bar{y}^j \right\rangle\right) \\ &= \frac{1}{2} \sum_i w_i \left(1 - \left\langle g(x_i), \bar{y}^{c_i} - \sum_{j \neq c_i} \bar{y}^j \frac{w(x_i, j)}{w_i} \right\rangle\right), \end{aligned} \tag{252}$$

with

$$w_i = \sum_{j \neq c_i} w(x_i, j). \tag{253}$$

Hence, the weak learner selected by AdaBoost.M2 is

$$\hat{g}_t \propto \arg\max_g \frac{1}{2} \sum_i w_i \left\langle g(x_i), \bar{y}^{c_i} - \sum_{j \neq c_i} \bar{y}^j \frac{w(x_i, j)}{w_i} \right\rangle. \tag{254}$$

Since the combination of (251), (253), and (254) is the special case of (29)-(32) with $\phi(v) = e^{-v}$ and $\gamma(v) = v$, AdaBoost.M2 is conceptually equivalent to GD-MCBoost with the exponential $\gamma - \phi$ loss of Table 3, canonical codewords, and the product weak learners of (244). In practice, however, the algorithms can behave very differently, due to the replacement of the MCBoost weak learners of (26) by the product learners of (244). Note that the latter augment training examples with their class identities, to form a new set of training examples $\bar{x} = [x, j] \; \forall j$. This is problematic because, besides increasing the size of the of training set M-fold, it creates many examples that differ by only one coordinate. These examples can be difficult to discriminate with the simple functions typically used to implement weak learners, e.g. decision stumps. These problems make AdaBoost.MR computationally intensive and very slow to converge. The same problem arises for AdaBoost.MH (Schapire and Singer, 1999) which also uses the weak learners of (244).

H.2.4. ADABOOST.MH

AdaBoost.MH is very similar to AdaBoost.MR. The only difference is the weight update stage, where (248) is replaced by

$$w(x_i, k) = w(x_i, k) \times e^{-\alpha_t z_{c_i,k} \hat{g}_t(x_i, k)}, \tag{255}$$

with $z_{c_i,k}$ as defined in (224). Using the procedure above, it can be shown that AdaBoost.MH minimizes the one-vs-all loss. This leads to a combination of the problems associated with the weak learners of (244) and the loss function of (225), discussed in Section H.1.1.

H.2.5. ADABOOST.MM

(Mukherjee and Schapire, 2013) proposed a framework for the characterization of multiclass boostability conditions and used it to motivate the AdaBoost.MM algorithm. This a is multiclass boosting method that uses classification weak learners $\hat{g}(x) : \mathcal{X} \to \{1, 2, \ldots, M\}$ to learn a predictor $\hat{f}(x) = \sum_t \alpha_t \hat{g}_t(x)$. The decision rule is

$$\bar{F}(x) = \arg \max_{j \in \{1,\ldots,M\}} \hat{f}(x_i, j), \tag{256}$$

where

$$\hat{f}(x_i, j) = \sum_t \alpha_t I(\hat{g}_t(x_i) = j), \tag{257}$$

and $I(.)$ is the indicator function. Each round of AdaBoost.MM computes the cost of assigning example i to class j according to

$$C_{i,j} = \begin{cases} e^{\hat{f}(x_i,j) - \hat{f}(x_i,c_i)} & \text{if } j \neq c_i \\ -\sum_{l \neq c_i} e^{\hat{f}(x_i,l) - \hat{f}(x_i,c_i)} & \text{if } j = c_i \end{cases}, \tag{258}$$

and selects the weak learner of lowest cost

$$g^* = \arg \min_{\hat{g}} \sum_i C_{i,\hat{g}(x_i)}. \tag{259}$$

AdaBoost.MM can be related to MCBoost by considering multidimensional weak learners of canonical output, i.e.

$$\bar{g}(x_i) = \mathbf{1}_{\hat{g}(x_i)}, \tag{260}$$

and a predictor

$$\bar{f}(x) = \sum_t \alpha_t \bar{g}_t(x). \tag{261}$$

Under these definitions, (257) can be written as

$$\hat{f}(x_i, j) = \left\langle \bar{f}(x_i), \bar{y}^j \right\rangle, \tag{262}$$

where

$$\bar{y}^j = \mathbf{1}_j \in \mathbb{R}^M, \tag{263}$$

is the canonical codeword set and the decision rule of (256) is equivalent to

$$\bar{F}(x) = \arg \max_{j \in \{1,\ldots,M\}} \left\langle \bar{y}^j, \bar{f}(x) \right\rangle. \tag{264}$$

Similarly, the costs of (258) can be written as

$$C_{i,j} = \begin{cases} e^{\left\langle \bar{f}(x_i), \bar{y}^j \right\rangle - \left\langle \bar{f}(x_i), \bar{y}^{c_i} \right\rangle} & \text{if } j \neq c_i \\ -\sum_{l \neq c_i} e^{\left\langle \bar{f}(x_i), \bar{y}^l \right\rangle - \left\langle \bar{f}(x_i), \bar{y}^{c_i} \right\rangle} & \text{if } j = c_i \end{cases}, \tag{265}$$

and the weak learner selection rule as

$$g^* = \arg \min_{\hat{g}} \sum_i \left\langle C_i, \bar{g}(x_i) \right\rangle, \tag{266}$$

where $C_i \in \mathbb{R}^M$ is the vector of components $C_{i,j}$. Using (263) and (265), this can be written as

$$\begin{aligned} C_i &= \sum_{j \neq c_i} \bar{y}^j e^{\left\langle \bar{y}^j, \bar{f}(x_i) \right\rangle - \left\langle \bar{y}^{c_i}, \bar{f}(x_i) \right\rangle} - \bar{y}^{c_i} \sum_{j \neq c_i} e^{\left\langle \bar{y}^j, \bar{f}(x_i) \right\rangle - \left\langle \bar{y}^{c_i}, \bar{f}(x_i) \right\rangle} \\ &= \sum_{j \neq c_i} (\bar{y}^j - \bar{y}^{c_i}) e^{-\left[\left\langle \bar{y}^{c_i}, \bar{f}(x_i) \right\rangle - \left\langle \bar{y}^j, \bar{f}(x_i) \right\rangle\right]}. \end{aligned} \tag{267}$$

It follows from (266) that (259) is equivalent to

$$\begin{aligned} g^* &= \arg \min_{\hat{g}} \sum_i \sum_{j \neq c_i} \left\langle \bar{g}(x_i), \bar{y}^j - \bar{y}^{c_i} \right\rangle e^{-\left[\left\langle \bar{y}^{c_i}, \bar{f}(x_i) \right\rangle - \left\langle \bar{y}^j, \bar{f}(x_i) \right\rangle\right]} \\ &= \arg \max_{\hat{g}} \sum_i w_i \left\langle \bar{g}(x_i), \bar{y}^{c_i} - \sum_{j \neq c_i} \bar{y}^j \frac{w(x_i, j)}{w_i} \right\rangle, \end{aligned} \tag{268}$$

with

$$\begin{aligned} w(x_i, j) &= e^{-\left[\left\langle \bar{y}^{c_i}, \bar{f}(x_i) \right\rangle - \left\langle \bar{y}^j, \bar{f}(x_i) \right\rangle\right]} \\ w_i &= \sum_{j \neq c_i} w(x_i, j), \end{aligned} \tag{269}$$

which is equivalent to the GD-MCBoost weak learner selection rule of (29) for $\phi(v) = e^{-2v}$ and $\gamma(v) = v$. Hence, AdaBoost.MM is an implementation of GD-MCBoost with the canonical codeword set $\mathbf{1}_j$, classification weak learners, and a $\gamma - \phi$ loss with $\phi(v) = e^{-2v}$ and $\gamma(v) = v$. The only difference is that each iteration of GD-MCBoost finds the optimal step size through a line search. On the other hand, AdaBoost.MM relies on heuristics to determine the step size. This can be sub-optimal.

Journal of Machine Learning Research 20 (2019) 1-29 Submitted 8/17; Revised 1/19; Published 9/19

Generalized maximum entropy estimation

Tobias Sutter TOBIAS.SUTTER@EPFL.CH
Risk Analytics and Optimization Chair
EPFL, Switzerland

David Sutter SUTTEDAV@PHYS.ETHZ.CH
Institute for Theoretical Physics
ETH Zurich, Switzerland

Peyman Mohajerin Esfahani P.MOHAJERINESFAHANI@TUDELFT.NL
Delft Center for Systems and Control
TU Delft, The Netherlands

John Lygeros LYGEROS@CONTROL.EE.ETHZ.CH
Department of Electrical Engineering and Information Technology
ETH Zurich, Switzerland

Editor: Benjamin Recht

Abstract

We consider the problem of estimating a probability distribution that maximizes the entropy while satisfying a finite number of moment constraints, possibly corrupted by noise. Based on duality of convex programming, we present a novel approximation scheme using a smoothed fast gradient method that is equipped with explicit bounds on the approximation error. We further demonstrate how the presented scheme can be used for approximating the chemical master equation through the zero-information moment closure method, and for an approximate dynamic programming approach in the context of constrained Markov decision processes with uncountable state and action spaces.

Keywords: Entropy maximization, convex optimization, relative entropy minimization, fast gradient method, approximate dynamic programming

1. Introduction

This article investigates the problem of estimating an unknown probability distribution given a finite number of observed moments that might be corrupted by noise. Given that the observed moments are consistent, i.e., there exists a probability distribution that satisfies all the moment constraints, the problem is underdetermined and has infinitely many solutions. This raises the question of which solution to choose. A natural choice would be to pick the one with the highest entropy, called the *MaxEnt distribution*. The main reason why the MaxEnt distribution is a natural choice is due to a concentration phenomenon described by Jaynes (2003):

> "*If the information incorporated into the maximum-entropy analysis includes all the constraints actually operating in the random experiment, then the distribution predicted by maximum entropy is overwhelmingly the most likely to be observed experimentally.*"

©2019 Tobias Sutter, David Sutter, Peyman Mohajerin Esfahani, John Lygeros.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-486.html.

See Jaynes (2003); Grünwald (2008) for a rigorous statement. This maximum entropy estimation problem subject to moment constraints, also known as the *principle of maximum entropy*, is applicable to large classes of problems in natural and social sciences — in particular in economics, see Golan (2008) for a comprehensive survey. Furthermore it has important applications in approximation methods to dynamical objects, such as in systems biology, where MaxEnt distributions are key objects in the so-called *moment closure method* to approximate the chemical master equation Smadbeck and Kaznessis (2013), or more recently in the context of approximating dynamic programming Mohajerin Esfahani et al. (2018) where MaxEnt distributions act as a regularizer, leading to computationally more efficient optimization programs.

Their operational significance motivates the study of numerical methods to compute MaxEnt distributions, which are the solutions of an infinite-dimensional convex optimization problem and as such computationally intractable in general. Since it was shown that the MaxEnt distribution subject to a finite number of moment constraints (if it exists) belongs to the exponential family of distributions Csiszár (1975), its computation can be reduced to solving a system of nonlinear equations, whose dimension is equal to the number of moment constraints Mead and Papanicolaou (1984). Furthermore, the system of nonlinear equations involves evaluating integrals over the support set $\mathbb{K}$ that are computationally difficult in general. Even if $\mathbb{K}$ is finite, finding the MaxEnt distribution is not straightforward, since solving a system of nonlinear equations can be computationally demanding.

In this article, we present a new approximation scheme to minimize the relative entropy subject to noisy moment constraints. This is a generalization of the introduced maximum entropy problem and extends the principle of maximum entropy to the so-called *principle of minimum discriminating information* Kullback (1959). We show that its dual problem exhibits a particular favorable structure that allows us to apply Nesterov's smoothing method Nesterov (2005) and hence tackle the presented problem using a fast gradient method obtaining process convergence properties, unlike Lasserre (2009).

Computing the MaxEnt distribution has applications in randomized rounding and the design of approximation algorithms. More precisely, it has been shown how to improve the approximation ratio for the symmetric and the asymmetric traveling salesman problem via MaxEnt distributions Asadpour et al. (2010); Gharan et al. (2011). Often, it is important to efficiently compute the MaxEnt distribution. For example, the zero-information moment closure method Smadbeck and Kaznessis (2013) (see Section 6), a recent approximate dynamic programming method for constrained Markov decision processes (see Section 7), as well as the approximation of the channel capacity of a large class of memoryless channels Sutter et al. (2015) deal with iterative algorithms that require the numerical computation of the MaxEnt distribution in each iteration step.

Related results. Before comparing the approach presented in this article with existing methods we provide a brief digression on the moment problem. Consider a one-dimensional moment problem formulated as follows: Given a set $\mathbb{K} \subset \mathbb{R}$ and a sequence $(y_i)_{i\in\mathbb{N}} \subset \mathbb{R}$ of moments, does there exist a measure μ supported on $\mathbb{K}$ such that

$$y_i = \int_{\mathbb{K}} x^i \mu(\mathrm{d}x) \quad \text{for all } i \in \mathbb{N} \ ? \tag{1}$$

For $\mathbb{K} = \mathbb{R}$ and $\mathbb{K} = [a, b]$ with $-\infty < a < b < \infty$ the above moment problem is known as the *Hamburger moment problem* and *Hausdorff moment problem*, respectively. If the moment sequence is finite, the problem is called a *truncated moment problem*. In both full and truncated cases, a measure μ that satisfies (1), is called a *representing measure* of the sequence $(y_i)_{i\in\mathbb{N}}$. If a representing measure is unique, it is said to be *determined by its moments*. From the Stone-Weierstrass theorem it followes directly that every non-truncated representing measure with compact support is determined by its moments. In the Hamburger moment problem, given a representing measure μ for a moment sequence $(y_i)_{i\in\mathbb{N}}$, a sufficient condition for μ being determined by its moments is the so-called *Carleman condition*, i.e., $\sum_{i=1}^{\infty} y_{2i}^{-1/2i} = \infty$. Roughly speaking this says that the moments should not grow too fast, see Akhiezer (1965) for further details. For the Hamburger and the Hausdorff moment problem, there are necessary and sufficient conditions for the existence of a representing measure for a given moment sequence $(y_i)_{i\in\mathbb{N}}$ in both the full as well as the truncated setting, that exploit the rich algebraic connection with Hankel matrices see (Lasserre, 2009, Theorems 3.2, 3.3, 3.4).

In (Lasserre, 2009, Section 12.3) it is shown that the maximum entropy subject to finite moment constraints can be approximated by using duality of convex programming. The problem can be reduced to an unconstrained finite-dimensional convex optimization problem and an approximation hierarchy of its gradient and Hessian in terms of two single semidefinite programs involving two linear matrix inequalities is presented. The desired accuracy is controlled by the size of the linear matrix inequalities. The method seems to be powerful in practice, however a rate of convergence has not been proven. Furthermore, it is not clear how the method extends to the case of uncertain moment constraints. In a finite dimensional setting, Dudik et al. (2007) presents a treatment of the maximum entropy principle with generalized regularization measures, that as a special case contains the setting presented here. However, convergence rates of algorithms presented are not known and again it is not clear how the method extends to the case of uncertain moment constraints. The discrete case, where the support set $\mathbb{K}$ is discrete, has been studied in more detail in the past. It has been shown the the maximum entropy problem in the discrete case has a succinct description that is polynomial-size in the input and can be efficiently computed Singh and Vishnoi (2014); Straszak and Vishnoi (2017). Furthermore it was shown that the maximum entropy problem is equivalent to the counting problem Singh and Vishnoi (2014).

Structure. The layout of this paper is as follows: In Section 2 we formally introduce the problem setting. Our results on an approximation scheme in a continuous setting are reported in Section 3. In Section 4, we show how these results simplify in the finite-dimensional case. Section 5 discusses the gradient approximation that is the dominant step of the proposed approximation method from a computational perspective. The theoretical results are applied in Section 6 to the zero-information moment closure method and in Section 7 to constrained Markov decision processes. We conclude in Section 8 with a summary of our work and comment on possible subjects of further research.

Notation. The logarithm with basis 2 and e is denoted by $\log(\cdot)$ and $\ln(\cdot)$, respectively. We define the standard $n-$simplex as $\Delta_n := \{x \in \mathbb{R}^n : x \geq 0, \sum_{i=1}^n x_i = 1\}$. For a probability mass function $p \in \Delta_n$ we denote its entropy by $H(p) := \sum_{i=1}^n -p_i \log p_i$. Let

$B(y, r) := \{x \in \mathbb{R}^n : \|x-y\|_2 \leq r\}$ denote the ball with radius r centered at y. Throughout this article, measurability always refers to Borel measurability. For a probability density p supported on a measurable set $B \subset \mathbb{R}$ we denote the differential entropy by $h(p) := -\int_B p(x) \log p(x) \mathrm{d}x$. For $\mathbb{A} \subset \mathbb{R}$ and $1 \leq p \leq \infty$, let $\mathrm{L}^p(\mathbb{A})$ denote the space of L^p-functions on the measure space $(\mathbb{A}, \mathcal{B}(\mathbb{A}), \mathrm{d}x)$, where $\mathcal{B}(\mathbb{A})$ denotes the Borel σ-algebra and $\mathrm{d}x$ the Lebesgue measure. Let $\mathbb{X}$ be a compact metric space, equipped with its Borel σ-field $\mathcal{B}(\cdot)$. The space of all probability measures on $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$ will be denoted by $\mathcal{P}(\mathbb{X})$. The *relative entropy* (or Kullback-Leibler divergence) between any two probability measures $\mu, \nu \in \mathcal{P}(\mathbb{X})$ is defined by

$$\mathsf{D}(\mu\|\nu) := \begin{cases} \int_{\mathbb{X}} \log\left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right) \mathrm{d}\mu, & \text{if } \mu \ll \nu \\ +\infty, & \text{otherwise}, \end{cases}$$

where $\ll$ denotes absolute continuity of measures, and $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}$ is the Radon-Nikodym derivative. The relative entropy is non-negative, and is equal to zero if and only if $\mu \equiv \nu$. Let $\mathbb{X}$ be restricted to a compact metric space and let us consider the pair of vector spaces $(\mathbb{M}(\mathbb{X}), \mathbb{B}(\mathbb{X}))$ where $\mathbb{M}(\mathbb{X})$ denotes the space of finite signed measures on $\mathcal{B}(\mathbb{X})$ and $\mathbb{B}(\mathbb{X})$ is the Banach space of bounded measurable functions on $\mathbb{X}$ with respect to the sup-norm and consider the bilinear form

$$\langle \mu, f \rangle := \int_{\mathbb{X}} f(x)\mu(\mathrm{d}x).$$

This induces the total variation norm as the dual norm on $\mathbb{M}(\mathbb{X})$, since by (Hernández-Lerma and Lasserre, 1999, p.2)

$$\|\mu\|_* = \sup_{\|f\|_\infty \leq 1} \langle \mu, f \rangle = \|\mu\|_{\mathsf{TV}},$$

making $\mathbb{M}(\mathbb{X})$ a Banach space. In the light of (Hernández-Lerma and Lasserre, 1999, p. 206) this is a dual pair of Banach spaces; we refer to (Anderson and Nash, 1987, Section 3) for the details of the definition of dual pairs. The Lipschitz norm is defined as $\|u\|_L := \sup_{x,x' \in \mathbb{X}}\{|u(x)|, \frac{|u(x)-u(x')|}{\|x-x'\|_\infty}\}$ and $\mathcal{L}(\mathbb{X})$ denotes the space of Lipschitz functions on $\mathbb{X}$.

2. Problem Statement

Let $\mathbb{K} \subset \mathbb{R}$ be compact and consider the scenario where a probability measure $\mu \in \mathcal{P}(\mathbb{K})$ is unknown and only observed via the following measurement model

$$y_i = \langle \mu, x^i \rangle + u_i, \quad u_i \in \mathcal{U}_i \quad \text{for } i = 1, \ldots, M\,, \tag{2}$$

where u_i represents the uncertainty of the obtained data point y_i and $\mathcal{U}_i \subset \mathbb{R}$ is compact, convex and $0 \in \mathcal{U}_i$ for all $i = 1, \ldots, M$. Given the data $(y_i)_{i=1}^M \subset \mathbb{R}$, the goal is to estimate a probability measure μ that is consistent with the measurement model (2). This problem (given that M is finite) is underdetermined and has infinitely many solutions. Among all possible solutions for (2), we aim to find the solution that maximizes the entropy. Define the set $T := \times_{i=1}^M \{y_i - u : u \in \mathcal{U}_i\} \subset \mathbb{R}^M$ and the linear operator $\mathcal{A} : \mathbb{M}(\mathbb{K}) \to \mathbb{R}^M$ by

$$(\mathcal{A}\mu)_i := \langle \mu, x^i \rangle = \int_{\mathbb{K}} x^i \mu(\mathrm{d}x) \quad \text{for all} \quad i = 1, \ldots, M\,.$$

The operator norm is defined as $\|\mathcal{A}\| := \sup_{\|\mu\|_{\mathsf{TV}}=1, \|y\|_2=1} \langle \mathcal{A}\mu, y\rangle$. Note that due to the compactness of $\mathbb{K}$ the operator norm is bounded, see Lemma 4 for a formal statement. The adjoint operator to $\mathcal{A}$ is given by $\mathcal{A}^* : \mathbb{R}^M \to \mathbb{B}(\mathbb{K})$, where $\mathcal{A}^* z(x) := \sum_{i=1}^M z_i x^i$; note that the domain and image spaces of the adjoint operator are well defined as $(\mathbb{B}(\mathbb{K}), \mathbb{M}(\mathbb{K}))$ is a topological dual pairs and the operator $\mathcal{A}$ is bounded (Hernández-Lerma and Lasserre, 1999, Proposition 12.2.5).

Given a reference measure $\nu \in \mathcal{P}(\mathbb{K})$, the problem of minimizing the relative entropy subject to moment constraints (2) can be formally described by

$$J^\star = \min_{\mu \in \mathcal{P}(\mathbb{K})} \left\{ \mathsf{D}\big(\mu \| \nu\big) \ : \ \mathcal{A}\mu \in T \right\}. \tag{3}$$

We note that the reference measure $\nu \in \mathcal{P}(\mathbb{K})$ is always fixed a priori. A typical choice for ν is the uniform measure over $\mathbb{K}$.

Proposition 1 (Existence & uniqueness of (3)**)** *The optimization problem* (3) *attains an optimal feasible solution that is unique.*

Proof The variational representation of the relative entropy (Boucheron et al., 2013, Corollary 4.15) implies that the mapping $\mu \mapsto \mathsf{D}\big(\mu \| \nu\big)$ is lower-semicontinuous Luenberger (1969). Note also that the space of probability measures on $\mathbb{K}$ is compact (Aliprantis and Border, 2007, Theorem 15.11). Moreover, since the linear operator $\mathcal{A}$ is bounded, it is continuous. As a result, the feasible set of problem (3) is compact and hence the optimization problem attains an optimal solution. Finally, the strict convexity of the relative entropy Csiszár (1975) ensures uniqueness of the optimizer. ■

Note that if $\mathcal{U}_i = \{0\}$ for all $i = 1, \ldots, M$, i.e., there is no uncertainty in the measurement model (2), Proposition 1 reduces to a known result Csiszár (1975). Consider the special case where the reference measure ν is the uniform measure on $\mathbb{K}$ and let p denote the Radon-Nikodym derivative $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}$ (whose existence can be assumed without loss of generality). Since $\mathcal{A}$ is weakly continuous and the differential entropy is known to be weakly lower semi-continuous Boucheron et al. (2013), we can restrict attention to a (weakly) dense subset of the feasible set and hence assume without loss of generality that $p \in \mathrm{L}^1(\mathbb{K})$. Problem (3) then reduces to

$$\max_{p \in \mathrm{L}^1(\mathbb{K})} \left\{ h(p) \ : \ \int_{\mathbb{K}} p(x)\mathrm{d}x = 1, \ \int_{\mathbb{K}} x^i p(x)\mathrm{d}x \in T_i, \ \forall i = 1, \ldots, M \right\}. \tag{4}$$

Problem (4) is a generalized maximum entropy estimation problem that, in case $\mathcal{U}_i = \{0\}$ for all $i = 1, \ldots, M$, simplifies to the standard entropy maximization problem subject to M moment constraints. In this article, we present a new approach to solve (3) that is based on its dual formulation. It turns out that the dual problem of (3) has a particular structure that allows us to apply Nesterov's smoothing method Nesterov (2005) to accelerate convergence. Furthermore, we will show how an ε-optimal solution to (3) can be reconstructed. This is done by solving the dual problem of (3). To achieve additional feasibility guarantees for the ε-optimal solution to (3), we introduce and a second smoothing step that is motivated by Devolder et al. (2012). The problem of entropy maximization subject to uncertain moment constraints (4) can be seen as a special case of (3).

3. Relative entropy minimization

We start by recalling that an unconstrained minimization of the relative entropy with an additional linear term in the cost admits a closed form solution. Let $c \in \mathbb{B}(\mathbb{K})$, $\nu \in \mathcal{P}(\mathbb{K})$ and consider the optimization problem

$$\min_{\mu \in \mathcal{P}(\mathbb{K})} \left\{ \mathsf{D}\big(\mu \| \nu\big) - \langle \mu, c \rangle \right\}. \tag{5}$$

Lemma 2 (Gibbs distribution) *The unique optimizer to problem* (5) *is given by the Gibbs distribution, i.e.,*

$$\mu^\star(\mathrm{d}x) = \frac{2^{c(x)}\nu(\mathrm{d}x)}{\int_{\mathbb{K}} 2^{c(x)}\nu(\mathrm{d}x)} \quad \textit{for } x \in \mathbb{K},$$

which leads to the optimal value of $-\log \int_{\mathbb{K}} 2^{c(x)}\nu(\mathrm{d}x)$.

Proof The result is standard and follows from Csiszár (1975) or alternatively by (Sutter et al., 2015, Lemma 3.10). ■

Let $\mathbb{R}^M \ni z \mapsto \sigma_T(z) := \max_{x \in T} \langle x, z \rangle \in \mathbb{R}$ denote the support function of T, which is continuous since T is compact (Rockafellar, 1997, Corollary 13.2.2). The primal-dual pair of problem (3) can be stated as

$$\text{(primal program)}: \quad J^\star = \min_{\mu \in \mathcal{P}(\mathbb{K})} \left\{ \mathsf{D}\big(\mu \| \nu\big) + \sup_{z \in \mathbb{R}^M} \left\{ \langle \mathcal{A}\mu, z \rangle - \sigma_T(z) \right\} \right\} \tag{6}$$

$$\text{(dual program)}: \quad J_{\mathsf{D}}^\star = \sup_{z \in \mathbb{R}^M} \left\{ -\sigma_T(z) + \min_{\mu \in \mathcal{P}(\mathbb{K})} \left\{ \mathsf{D}\big(\mu \| \nu\big) + \langle \mathcal{A}\mu, z \rangle \right\} \right\}, \tag{7}$$

where the dual function is given by

$$F(z) = -\sigma_T(z) + \min_{\mu \in \mathcal{P}(\mathbb{K})} \left\{ \mathsf{D}\big(\mu \| \nu\big) + \langle \mathcal{A}\mu, z \rangle \right\}. \tag{8}$$

Note that the primal program (6) is an infinite-dimensional convex optimization problem. The key idea of our analysis is driven by Lemma 2 indicating that the dual function, that involves a minimization running over an infinite-dimensional space, is analytically available. As such, the dual problem becomes an unconstrained finite-dimensional convex optimization problem, which is amenable to first-order methods.

Lemma 3 (Zero duality gap) *There is no duality gap between the primal program* (6) *and its dual* (7)*, i.e.,* $J^\star = J_{\mathsf{D}}^\star$*. Moreover, if there exists* $\bar{\mu} \in \mathcal{P}(\mathbb{K})$ *such that* $\mathcal{A}\bar{\mu} \in \mathsf{int}(T)$*, then the set of optimal dual variables in* (7) *is compact.*

Proof Recall that the relative entropy is known to be lower semicontinuous and convex in the first argument, which can be seen as a direct consequence of the duality relation for the relative entropy (Boucheron et al., 2013, Corollary 4.15). Hence, the desired zero duality gap follows by Sion's minimax theorem (Sion, 1958, Theorem 4.2). The compactness of the

set of dual optimizers is due to (Bertsekas, 2009, Proposition 5.3.1). ■

Because the dual function (8) turns out to be non-smooth, in the absence of any additional structure, the efficiency estimate of a black-box first-order method is of order $O(1/\varepsilon^2)$, where ε is the desired absolute additive accuracy of the approximate solution in function value Nesterov (2004). We show, however, that the generalized entropy maximization problem (6) has a certain structure that allows us to deploy the recent developments in Nesterov (2005) for approximating non-smooth problems by smooth ones, leading to an efficiency estimate of order $O(1/\varepsilon)$. This, together with the low complexity of each iteration step in the approximation scheme, offers a numerical method that has an attractive computational complexity. In the spirit of Nesterov (2005); Devolder et al. (2012), we introduce a smoothing parameter $\eta := (\eta_1, \eta_2) \in \mathbb{R}^2_{>0}$ and consider a smooth approximation of the dual function

$$F_\eta(z) := -\max_{x\in T}\left\{\langle x, z\rangle - \frac{\eta_1}{2}\|x\|_2^2\right\} + \min_{\mu\in\mathcal{P}(\mathbb{K})}\left\{\mathsf{D}(\mu\|\nu) + \langle \mathcal{A}\mu, z\rangle\right\} - \frac{\eta_2}{2}\|z\|_2^2\,, \tag{9}$$

with respective optimizers denoted by $x_z^\star$ and $\mu_z^\star$. Consider the projection operator $\pi_T : \mathbb{R}^m \to \mathbb{R}$, $\pi_T(y) = \arg\min_{x\in T}\|x-y\|_2^2$. It is straightforward to see that the optimizer $x_z^\star$ is given by

$$x_z^\star = \arg\min_{x\in T}\|x - \eta_1^{-1}z\|_2^2 = \pi_T\left(\eta_1^{-1}z\right).$$

Hence, the complexity of computing $x_z^\star$ is determined by the projection operator onto T; for simple enough cases (e.g., 2-norm balls, hybercubes) the solution is analytically available, while for more general cases (e.g., simplex, 1-norm balls) it can be computed at relatively low computational effort, see (Richter, 2012, Section 5.4) for a comprehensive survey. The optimizer $\mu_z^\star$ according to Lemma 2 is given by

$$\mu_z^\star(B) = \frac{\int_B 2^{-\mathcal{A}^* z(x)}\nu(\mathrm{d}x)}{\int_{\mathbb{K}} 2^{-\mathcal{A}^* z(x)}\nu(\mathrm{d}x)}, \quad \text{for all } B \in \mathcal{B}(\mathbb{K}).$$

Lemma 4 (Lipschitz gradient) *The dual function F_η defined in (9) is η_2-strongly concave and differentiable. Its gradient $\nabla F_\eta(z) = -x_z^\star + \mathcal{A}\mu_z^\star - \eta_2 z$ is Lipschitz continuous with Lipschitz constant $\frac{1}{\eta_1} + \left(\sum_{i=1}^M B^i\right)^2 + \eta_2$ and $B := \max\{|x| \;:\; x \in \mathbb{K}\}$.*

Proof The proof follows along the lines of (Nesterov, 2005, Theorem 1) and in particular by recalling that the relative entropy (in the first argument) is strongly convex with convexity parameter one and Pinsker's inequality, that says that for any $\mu \in \mathcal{P}(\mathbb{K})$ we have

$$\|\mu - \nu\|_{\mathsf{TV}} \le \sqrt{2\mathsf{D}(\mu\|\nu)}\,. \tag{10}$$

Moreover, we use the bound

$$\begin{aligned}
\|\mathcal{A}\| &= \sup_{\lambda\in\mathbb{R}^M,\,\mu\in\mathcal{P}(\mathbb{K})}\left\{\langle \mathcal{A}\mu, \lambda\rangle \;:\; \|\lambda\|_2 = 1,\ \|\mu\|_{\mathsf{TV}} = 1\right\}\\
&\le \sup_{\lambda\in\mathbb{R}^M,\,\mu\in\mathcal{P}(\mathbb{K})}\left\{\|\mathcal{A}\mu\|_2\,\|\lambda\|_2 \;:\; \|\lambda\|_2 = 1,\ \|\mu\|_{\mathsf{TV}} = 1\right\} \qquad (11)\\
&\le \sup_{\mu\in\mathcal{P}(\mathbb{K})}\left\{\|\mathcal{A}\mu\|_1 \;:\; \|\mu\|_{\mathsf{TV}} = 1\right\}\\
&= \sup_{\mu\in\mathcal{P}(\mathbb{K})}\left\{\sum_{i=1}^M\left|\int_{\mathbb{K}} x^i\mu(\mathrm{d}x)\right| \;:\; \|\mu\|_{\mathsf{TV}} = 1\right\}\\
&\le \sum_{i=1}^M B^i\,,
\end{aligned}$$

where (11) is due to the Cauchy-Schwarz inequality. ■

Note that F_η is η_2-strongly concave and according to Lemma 4 its gradient is Lipschitz continuous with constant $L(\eta) := \frac{1}{\eta_1} + \|\mathcal{A}\|^2 + \eta_2$. We finally consider the approximate dual program given by

$$\text{(smoothed dual program)}: \quad J_\eta^\star = \sup_{z\in\mathbb{R}^M} F_\eta(z)\,. \tag{12}$$

It turns out that (12) belongs to a favorable class of smooth and strongly convex optimization problems that can be solved by a fast gradient method given in Algorithm 1 (see Nesterov (2004)) with an efficiency estimate of the order $O(1/\sqrt{\varepsilon})$.

Algorithm 1: Optimal scheme for smooth & strongly convex optimization

Choose $w_0 = y_0 \in \mathbb{R}^M$ and $\eta \in \mathbb{R}^2_{>0}$

For $k \ge 0$ **do***
- **Step 1:** Set $y_{k+1} = w_k + \frac{1}{L(\eta)}\nabla F_\eta(w_k)$
- **Step 2:** Compute $w_{k+1} = y_{k+1} + \frac{\sqrt{L(\eta)}-\sqrt{\eta_2}}{\sqrt{L(\eta)}+\sqrt{\eta_2}}(y_{k+1} - y_k)$

[*The stopping criterion is explained in Remark 7]

Under an additional regularity assumption, solving the smoothed dual problem (12) provides an estimate of the primal and dual variables of the original non-smooth problems (6) and (7), respectively, as summarized in the next theorem (Theorem 5). The main computational difficulty of the presented method lies in the gradient evaluation ∇F_η. We refer to Section 5, for a detailed discussion on this subject.

Assumption 1 (Slater point) *There exits a strictly feasible solution to (3), i.e., $\mu_0 \in \mathcal{P}(\mathbb{K})$ such that $\mathcal{A}\mu_0 \in T$ and $\delta := \min_{y\in T^c}\|\mathcal{A}\mu_0 - y\|_2 > 0$.*

Note that finding a Slater point μ_0 such that Assumption 1 holds, in general can be difficult. In Remark 8 we present a constructive way of finding such an interior point. Given Assumption 1, for $\varepsilon > 0$ define

$$C := \mathsf{D}\big(\mu_0 \| \nu\big)\,, \qquad D := \frac{1}{2}\max_{x\in T}\|x\|_2, \qquad \eta_1(\varepsilon) := \frac{\varepsilon}{4D}, \qquad \eta_2(\varepsilon) := \frac{\varepsilon\delta^2}{2C^2}$$

$$N_1(\varepsilon) := 2\left(\sqrt{\frac{8DC^2}{\varepsilon^2\delta^2} + \frac{2\|\mathcal{A}\|^2C^2}{\varepsilon\delta^2} + 1}\right)\ln\left(\frac{10(\varepsilon+2C)}{\varepsilon}\right) \tag{13}$$

$$N_2(\varepsilon) := 2\left(\sqrt{\frac{8DC^2}{\varepsilon^2\delta^2} + \frac{2\|\mathcal{A}\|^2C^2}{\varepsilon\delta^2} + 1}\right)\ln\left(\frac{C}{\varepsilon\delta(2-\sqrt{3})}\sqrt{4\left(\frac{4D}{\varepsilon} + \|\mathcal{A}\|^2 + \frac{\varepsilon\delta^2}{2C^2}\right)\left(C + \frac{\varepsilon}{2}\right)}\right).$$

Theorem 5 (Convergence rate) *Given Assumption 1 and* (13)*, let $\varepsilon > 0$ and $N(\varepsilon) := \lceil \max\{N_1(\varepsilon), N_2(\varepsilon)\}\rceil$. Then, $N(\varepsilon)$ iterations of Algorithm 1 produce approximate solutions to the problems* (7) *and* (6) *given by*

$$\hat{z}_{k,\eta} := y_k \quad \textit{and} \quad \hat{\mu}_{k,\eta}(B) := \frac{\int_B 2^{-\mathcal{A}^*\hat{z}_{k,\eta}(x)}\nu(\mathrm{d}x)}{\int_{\mathbb{K}} 2^{-\mathcal{A}^*\hat{z}_{k,\eta}(x)}\nu(\mathrm{d}x)}, \quad \textit{for all } B \in \mathcal{B}(\mathbb{K})\,, \tag{14}$$

which satisfy

$$\textit{dual } \varepsilon\textit{-optimality:} \qquad 0 \le J^\star - F(\hat{z}_{k(\varepsilon)}) \le \varepsilon \tag{15a}$$

$$\textit{primal } \varepsilon\textit{-optimality:} \qquad |\mathsf{D}\big(\hat{\mu}_{k(\varepsilon)}\|\nu\big) - J^\star| \le 2(1+2\sqrt{3})\varepsilon \tag{15b}$$

$$\textit{primal } \varepsilon\textit{-feasibility:} \qquad d(\mathcal{A}\hat{\mu}_{k(\varepsilon)}, T) \le \frac{2\varepsilon\delta}{C}\,, \tag{15c}$$

where $d(\cdot, T)$ denotes the distance to the set T, i.e., $d(x,T) := \min_{y\in T}\|x-y\|_2$.

In some applications, Assumption 1 does not hold, as for example in the classical case where $\mathcal{U}_i = \{0\}$ for all $i = 1,\dots,M$, i.e., there is no uncertainty in the measurement model (2). Moreover, in other cases satisfying Assumption 1 using the construction described in Remark 8 might be computationally expensive. Interestingly, Algorithm 1 can be run irrespective of whether Assumption 1 holds or not, i.e. for any choice of C and δ. While explicit error bounds of Theorem 5 as well as the a-posteriori error bound discussed below do not hold anymore, the asymptotic convergence is not affected.

Proof Using Assumption 1, note that the constant defined as

$$\iota := \frac{\mathsf{D}\big(\mu_0\|\nu\big) - \min_{\mu\in\mathcal{P}(\mathbb{K})}\mathsf{D}\big(\mu\|\nu\big)}{\min_{y\in T^c}\|\mathcal{A}\mu_0 - y\|_2} = \frac{C}{\delta}$$

can be shown to be an upper bound for the optimal dual multiplier (Nedić and Ozdaglar, 2008, Lemma 1), i.e., $\|z^\star\|_2 \le \iota$. The dual function can be bounded from above by C, since weak duality ensures $F(z) \le J^\star \le \mathsf{D}\big(\mu_0\|\nu\big) = C$ for all $z \in \mathbb{R}^M$. Moreover, if we recall the preparatory Lemmas 3 and 4, we are finally in the setting such that the presented error bounds can be derived from Devolder et al. (2012), see Appendix A for a detailed derivation. ■

Theorem 5 directly implies that we need at most $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ iterations of Algorithm 1 to achieve ε-optimality of primal and dual solutions as well as ε-feasible primal variable. Note that Theorem 5 provides an explicit bound on the so-called *a-priori errors*, together with approximate optimizer of the primal (6) and dual (7) problem. The latter allows us to derive an *a-posteriori error* depending on the approximate optimizers, which is often significantly smaller than the a-priori error.

Corollary 6 (Posterior error estimation) *Given Assumption 1, the approximate primal and dual variables $\hat{\mu}$ and $\hat{z}$ given by* (14)*, satisfy the following a-posteriori error bound*

$$F(\hat{z}) \le J^\star \le \mathsf{D}\big(\hat{\mu}\|\nu\big) + \frac{C}{\delta}d(\mathcal{A}\hat{\mu}, T)\,,$$

where $d(\cdot, T)$ denotes the distance to the set T, i.e., $d(x,T) := \inf_{y\in T}\|x-y\|_2$.

Proof The two key ingredients of the proof are Theorem 5 and the Lipschitz continuity of the so-called perturbation function of convex programming. Let $z^\star$ denote the dual optimizer to (7). We introduce the perturbed program as

$$\begin{aligned}
J^\star(\varepsilon) &= \min_{\mu\in\mathcal{P}(\mathbb{K})}\{\mathsf{D}\big(\mu\|\nu\big) \,:\, d(\mathcal{A}\mu, T) \le \varepsilon\} \\
&= \min_{\mu\in\mathcal{P}(\mathbb{K})}\mathsf{D}\big(\mu\|\nu\big) + \sup_{\lambda\ge 0}\inf_{y\in T}\lambda\|\mathcal{A}\mu - y\| - \lambda\varepsilon \\
&= \sup_{\lambda\ge 0} -\lambda\,\varepsilon + \inf_{\substack{\mu\in\mathcal{P}(\mathbb{K})\\ y\in T}}\sup_{\|z\|_2\le\lambda}\big\langle \mathcal{A}\mu - y, z\big\rangle + \mathsf{D}\big(\mu\|\nu\big) && \text{(16a)} \\
&= \sup_{\substack{\lambda\ge 0\\ \|z\|_2\le\lambda}} -\lambda\,\varepsilon + \inf_{\substack{\mu\in\mathcal{P}(\mathbb{K})\\ y\in T}}\big\langle \mathcal{A}\mu - y, z\big\rangle + \mathsf{D}\big(\mu\|\nu\big) && \text{(16b)} \\
&\ge -\|z^\star\|_2\,\varepsilon + \inf_{\substack{\mu\in\mathcal{P}(\mathbb{K})\\ y\in T}}\big\langle \mathcal{A}\mu - y, z^\star\big\rangle + \mathsf{D}\big(\mu\|\nu\big) \\
&= -\|z^\star\|_2\,\varepsilon + J^\star.
\end{aligned}$$

Equation (16a) uses the strong duality property that follows by the existence of a Slater point that is due to the definition of the set T, see Section 2. Step (16b) follows by Sion's minimax theorem (Sion, 1958, Theorem 4.2). Hence, we have shown that the perturbation function is Lipschitz continuous with constant $\|z^\star\|_2$. Finally, recalling $\|z^\star\|_2 \le \frac{C}{\delta}$, established in the proof of Theorem 5 completes the proof. ■

Remark 7 (Stopping criterion of Algorithm 1) *There are two alternatives for defining a stopping criterion for Algorithm 1. Choose desired accuracy $\varepsilon > 0$.*

(i) a-priori stopping criterion*: Theorem 5 provides the required number of iterations $N(\varepsilon)$ to ensure an ε-close solution.*

(ii) a-posteriori stopping criterion*: Choose the smoothing parameter η as in* (13)*. Fix a (small) number of iterations ℓ that are run using Algorithm 1. Compute the a-posteriori error $\mathsf{D}\big(\hat{\mu}\|\nu\big) + \frac{C}{\delta}d(\mathcal{A}\hat{\mu}, T) - F(\hat{z})$ according to Corollary 6 and if it is smaller than If ε terminate the algorithm. Otherwise continue with another ℓ iterations.*

Remark 8 (Slater point computation) *To compute the respective constants in Assumption 1, we need to construct a strictly feasible point for* (3)*. For this purpose, we consider a polynomial density of degree r defined as $p_r(\alpha, x) := \sum_{i=0}^{r-1} \alpha_i x^i$. For notational simplicity we assume that the support set is the unit interval ($\mathbb{K} = [0,1]$), such that the moments induced by the polynomial density are given by*

$$\langle p_r(\alpha, x), x^i \rangle = \int_0^1 \sum_{j=0}^{r-1} \alpha_j x^{j+i} \mathrm{d}x = \sum_{j=0}^{r-1} \frac{\alpha_j}{j+i+1},$$

for $i = 0, \ldots, M$. Consider $\beta \in \mathbb{R}^{M+1}$, where $\beta_1 = 1$ and $\beta_i = y_{i-1}$ for $i = 2, \ldots, M+1$. Hence, the feasibility requirement of (3) *can be expressed as the linear constraint $A\alpha = \beta$, where $A \in \mathbb{R}^{(M+1)\times r}$, $\alpha \in \mathbb{R}^r$, $\beta \in \mathbb{R}^{M+1}$ and $A_{i,j} = \frac{1}{i+j-1}$ and finding a strictly feasible solution reduces to the following feasibility problem*

$$\begin{cases} \max\limits_{\alpha \in \mathbb{R}^r} & const \\ \text{s.t.} & A\alpha = \beta \\ & p_r(\alpha, x) \geq 0 \quad \forall x \in [0,1], \end{cases} \tag{17}$$

where p_r is a polynomial function in x of degree r with coefficients α. The second constraint of the program (17) *(i.e., $p_r(\alpha, x) \geq 0 \; \forall x \in [0,1]$)*[1] *can be equivalently reformulated as linear matrix inequalities of dimension $\lceil \frac{r}{2} \rceil$, using a standard result in polynomial optimization, see (Lasserre, 2009, Chapter 2) for details. We note that for small degree r, the set of feasible solutions to problem* (17) *may be empty, however, by choosing r large enough and assuming that the moments can be induced by a continuous density, problem* (17) *becomes feasible. Moreover, if $0 \in \mathsf{int}(T)$ the Slater point leads to a $\delta > 0$ in Assumption 1.*

Example 1 (Density estimation) *We are given the first* 3 *moments of an unknown probability measure defined on $\mathbb{K} = [0,1]$ as*[2]

$$y := \left(\frac{1 - \ln 2}{\ln 2}, \frac{\ln 4 - 1}{\ln 4}, \frac{5 - \ln 64}{\ln 64} \right) \approx (0.44,\ 0.28,\ 0.20).$$

The uncertainty set in the measurement model (2) *is assumed to be $\mathcal{U}_i = [-u, u]$ for all $i = 1, \ldots, 3$. A Slater point is constructed using the method described in Remark 8, where $r = 5$ is enough for the problem* (17) *to be feasible, leading to the constant $C = 0.0288$. The Slater point is depicted in Figure 1 and its differential entropy can be numerically computed as -0.0288.*

We consider two simulations for two different uncertainty sets (namely, $u = 0.01$ and $u = 0.005$). The underlying maximum entropy problem (4) *is solved using Algorithm 1. The respective features of the a-priori guarantees by Theorem 5 as well as the a-posteriori guarantees (upper and lower bounds) by Corollary 6 are reported in Table 1. Recall that $\hat{\mu}_{k(\varepsilon)}$ denotes the approximate primal variable after k-iterations of Algorithm 1 as defined in Theorem 5 and that $d(\mathcal{A}\hat{\mu}_{k(\varepsilon)}, T)$ (resp. $\frac{2\varepsilon\delta}{C}$) represent the a-posteriori (resp. a-priori) feasibility guarantees. It can be seen in Table 1 that increasing the uncertainty set $\mathcal{U}$ leads to a higher entropy, where the uniform density clearly has the highest entropy. This is also intuitively expected since enlarging the uncertainty set is equivalent to relaxing the moment constraints in the respective maximum entropy problem. The corresponding densities are graphically visualized in Figure 1.*

1. In a multi-dimensional setting one has to consider a tightening (i.e., $p_r(\alpha, x) > 0 \; \forall x \in [0,1]^n$).
2. The considered moments are actually induced by the probability density $f(x) := (\ln 2 \; (1+x))^{-1}$. We, however, do not use this information at any point of this example.

Table 1: Some specific simulation points of Example 1.

	$\mathcal{U} = [-0.01, 0.01]$				$\mathcal{U} = [-0.005, 0.005]$			
a-priori error ε	1	0.1	0.01	0.001	1	0.1	0.01	0.001
J_{UB}	-0.0174	-0.0189	-0.0194	-0.0194	-0.0223	-0.0236	-0.0237	-0.0238
J_{LB}	-0.0220	-0.0279	-0.0204	-0.0195	-0.0263	-0.0298	-0.0244	-0.0238
iterations $k(\varepsilon)$	99	551	5606	74423	232	1241	12170	157865
$d(\mathcal{A}\hat{\mu}_{k(\varepsilon)}, T)$	0.0008	0.0036	0.0005	0	0	0.001	0.0001	0
$\frac{2\varepsilon\delta}{C}$	0.69	0.069	0.0069	0.00069	0.35	0.035	0.0035	0.00035
runtime [s][3]	1.4	1.4	2.3	12.9	1.4	1.5	3.3	26.1

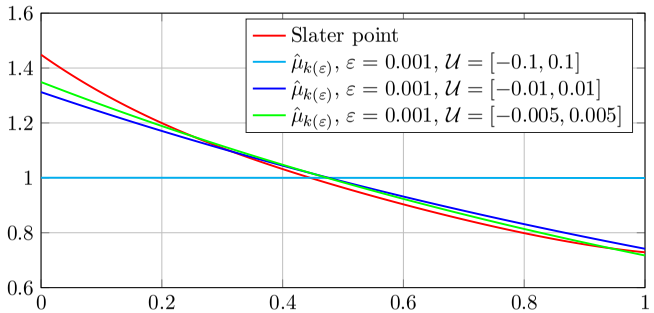


Figure 1: Maximum entropy densities obtained by Algorithm 1 for two different uncertainty sets. As a reference, the Slater point density, that was computed as described in Remark 8 is depicted in red.

4. Finite-dimensional case

We consider the finite-dimensional case where $\mathbb{K} = \{1, \ldots, N\}$ and hence we optimize in (3) over the probability simplex $\mathcal{P}(\mathbb{K}) = \Delta_N$. One substantial simplification, when restricting to the finite-dimensional setting, is that the Shannon entropy is non-negative and bounded from above (by $\log N$). Therefore, we can substantially weaken Assumption 1 to the following assumption.

3. Runtime includes Slater point computation. Simulations were run with Matlab on a laptop with a 2.2 GHz Intel Core i7 processor.

Assumption 2 (Regularity)

(i) There exists $\delta > 0$ such that $B(0,\delta) \subset \{x - \mathcal{A}\mu \ : \ \mu \in \Delta_N, x \in T\}$.

(ii) The reference measure $\nu \in \Delta_N$ has full support, i.e., $\min_{1\le i\le N} \nu_i > 0$.

Consider the the definitions given in (13) with $C := \max_{1\le i\le N} \log \frac{1}{\nu_i}$, then the following finite-dimensional equivalent to Theorem 5 holds.

Corollary 9 (A-priori error bound) *Given Assumption 2, $C := \max_{1\le i\le N} \log \frac{1}{\nu_i}$ and the definitions (13), let $\varepsilon > 0$ and $N(\varepsilon) := \lceil \{N_1(\varepsilon), N_2(\varepsilon)\} \rceil$. Then, $N(\varepsilon)$ iterations of Algorithm 1 produce the approximate solutions to the problems (7) and (6), given by*

$$\hat{z}_{k(\varepsilon)} := y_{k(\varepsilon)} \quad \text{and} \quad \hat{\mu}_{k(\varepsilon)}(B) := \frac{\sum_{i\in B} 2^{-\left(\mathcal{A}^*\hat{z}_{k(\varepsilon)}\right)_i}\nu_i}{\sum_{i=1}^N 2^{-\left(\mathcal{A}^*\hat{z}_{k(\varepsilon)}\right)_i}\nu_i} \quad \text{for all } B \subset \{1,2,\dots,N\}\,, \tag{18}$$

which satisfy

$$\text{dual } \varepsilon\text{-optimality:} \qquad 0 \le F(\hat{z}_{k(\varepsilon)}) - J^\star \le \varepsilon \tag{19a}$$

$$\text{primal } \varepsilon\text{-optimality:} \qquad |\mathsf{D}(\hat{\mu}_{k(\varepsilon)}\|\nu) - J^\star| \le 2(1+2\sqrt{3})\varepsilon \tag{19b}$$

$$\text{primal } \varepsilon\text{-feasibility:} \qquad d(\mathcal{A}\hat{\mu}_{k(\varepsilon)}, T) \le \frac{2\varepsilon\delta}{C}\,, \tag{19c}$$

where $d(\cdot,T)$ denotes the distance to the set T, i.e., $d(x,T) := \min_{y\in T}\|x-y\|_2$.

Proof Under Assumption 2 the dual optimal solutions in (7) are bounded by

$$\|z^\star\| \le \frac{1}{r}\max_{1\le i\le N} \log\frac{1}{\nu_i}\,. \tag{20}$$

This bound on the dual optimizer follows along the lines of (Devolder et al., 2012, Theorem 6.1). The presented error bounds can then be derived along the lines of Theorem 5. ■

In addition to the explicit error bound provided by Corollary 9, the a-posteriori upper and lower bounds presented in Corollary 6 directly apply to the finite-dimensional setting as well.

5. Gradient Approximation

The computationally demanding element for Algorithm 1 is the evaluation of the gradient $\nabla F_\eta(\cdot)$ given in Lemma 4. In particular, Theorem 5 and Corollary 9 assume that this gradient is known exactly. While this is not restrictive if, for example, $\mathbb{K}$ is a finite set, in general, $\nabla F_\eta(\cdot)$ involves an integration that can only be computed approximately. In particular if we consider a multi-dimensional setting (i.e., $\mathbb{K} \subset \mathbb{R}^d$), the evaluation of the gradient $\nabla F_\eta(\cdot)$ represents a multi-dimensional integration problem. This gives rise to the question of how the fast gradient method (and also Theorem 5) behaves in a case of inexact first-order information. Roughly speaking, the fast gradient method Algorithm 1,

while being more efficient than the classical gradient method (if applicable), is less robust when dealing with inexact gradients Devolder et al. (2014). Therefore, depending on the computational complexity of the gradient, one may consider the possibility of replacing Algorithm 1 with a classical gradient method. A detailed mathematical analysis of this tradeoff is a topic of further research, and we refer the interested readers to Devolder et al. (2014) for further details in this regard.

In this section we discuss two numerical methods to approximate this gradient. Note that in Lemma 4, given that T is simple enough the optimizer $x_z^\star$ is analytically available, so what remains is to compute $\mathcal{A}\mu_z^\star$, that according to Lemma 2 is given by

$$(\mathcal{A}\mu_z^\star)_i = \frac{\int_{\mathbb{K}} x^i 2^{-\mathcal{A}^* z(x)}\nu(\mathrm{d}x)}{\int_{\mathbb{K}} 2^{-\mathcal{A}^* z(x)}\nu(\mathrm{d}x)} \quad \text{for all } i = 1,\dots,M\,. \tag{21}$$

Semidefinite programming. Due to the specific structure of the considered problem, (21) represents an integration of exponentials of polynomials for which an efficient approximation in terms of two single semidefinite programs (SDPs) involving two linear matrix inequalities has been derived, where the desired accuracy is controlled by the size of the linear matrix inequalities constraints, see Bertsimas et al. (2008); Lasserre (2009) for a comprehensive study and for the construction of those SDPs. While the mentioned hierarchy of SDPs provides a certificate of optimality (hat is easy to evaluate and asymptotic convergence (in the size of the SDPs), a convergence rate that explicitly quantifies the size of the SDPs required for a desired accuracy is unknown. In practice, the hierarchy often converges in few iteration steps, which however, depends on the problem and is not known a priori.

Quasi-Monte Carlo. The most popular methods for integration problems of the from (21) are Monte Caro (MC) schemes, see Robert and Casella (2004) for a comprehensive summary. The main advantage of MC methods is that the root-mean-square error of the approximation converges to 0 with a rate of $O(N^{-1/2})$ that is independent of the dimension, where N are the number of samples used. In practise, this convergence often is too slow. Under mild assumptions on the integrand, the MC methods can be significantly improved with a more recent technique known as Quasi-Monte Carlo (QMC) methods. QMC methods can reach a convergence rate arbitrarily close to $O(N^{-1})$ with a constant not depending on the dimension of the problem. We would like to refer the reader to Dick et al. (2013); Sloan and Woźniakowski (1998); Kuo and Sloan (2005); Niederreiter (2010); Thély (2017) for a detailed discussion about the theory of QMC methods.

Remark 10 (Computational stability) *The evaluation of the gradient in Lemma 4 involves the term $\mathcal{A}\mu_z^\star$, where $\mu_z^\star$ is the optimizer of the second term in (9). By invoking Lemma 2 and the definition of the operator $\mathcal{A}$, the gradient evaluation reduces to*

$$(\mathcal{A}\mu_z^\star)_i = \frac{\int_{\mathbb{K}} x^i 2^{-\sum_{j=1}^M z_j x^j}\mathrm{d}x}{\int_{\mathbb{K}} 2^{-\sum_{j=1}^M z_j x^j}\mathrm{d}x} \quad \text{for } i = 1,\dots,M\,. \tag{22}$$

Note that a straightforward computation of the gradient via (22) is numerically difficult. To alleviate this difficulty, we follow the suggestion of (Nesterov, 2005, p. 148) which we briefly

elaborate here. Consider the functions $f(z,x) := -\sum_{j=1}^{M} z_j x^j$, $\bar{f}(z) := \max_{x\in\mathbb{K}} f(z,x)$ *and* $g(z,x) := f(z,x) - \bar{f}(z)$. *Note, that* $g(z,x) \geq 0$ *for all* $(z,x) \in \mathbb{R}^M \times \mathbb{R}$. *One can show that*

$$(\mathcal{A}\mu_z^\star)_i = \frac{\int_{\mathbb{K}} 2^{g(z,x)} \frac{\partial}{\partial z_i} g(z,x) \mathrm{d}x}{\int_{\mathbb{K}} 2^{g(z,x)} \mathrm{d}x} + \frac{\partial}{\partial z_i} \bar{f}(z) \quad \textit{for } i = 1, \dots, M\,,$$

which can be computed with significantly smaller numerical error than (22) *as the numerical exponent are always negative, leading to values always being smaller than* 1.

6. Zero-information moment closure method

In chemistry, physics, systems biology and related fields, stochastic chemical reactions are described by the *chemical master equation* (CME), that is a special case of the Chapman-Kolmogorov equation as applied to Markov processes van Kampen (1981); Wilkinson (2006). These equations are usually infinite-dimensional and analytical solutions are generally impossible. Hence, effort has been directed toward developing of a variety of numerical schemes for efficient approximation of the CME, such as stochastic simulation techniques (SSA) Gillespie (1976). In practical cases, one is often interested in the first few moments of the number of molecules involved in the chemical reactions. This motivated the development of approximation methods to those low-order moments without having to solve the underlying infinite-dimensional CME. One such approximation method is the so-called *moment closure method* Gillespie (2009), that briefly described works as follows: First the CME is recast in terms of moments as a linear ODE of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mu(t) = A\mu(t) + B\zeta(t)\,, \tag{23}$$

where $\mu(t)$ denotes the moments up to order M at time t and $\zeta(t)$ is an infinite vector describing the contains moments of order $M+1$ or higher. In general ζ can be an infinite vector, but for most of the standard chemical reactions considered in, e.g., systems biology it turns out that only a finite number of higher order moments affect the evolution of the first M moments. Indeed, if the chemical system involves only the so-called zeroth and first order reactions the vector ζ has dimension zero (reduces to a constant affine term), whereas if the system also involves second order reactions then ζ also contains some moments of order $M+1$ only. It is widely speculated that reactions up to second order are sufficient to realistically model most systems of interest in chemistry and biology Gillespie (2007); Gillespie et al. (2013). The matrix A and the linear operator B (that may potentially be infinite-dimensional) can be found analytically from the CME. The ODE (23), however, is intractable due to its higher order moments dependence. The approximation step is introduced by a so-called closure function

$$\zeta = \varphi(\mu)\,,$$

where the higher-order moments are approximated as a function of the lower-oder moments, see Singh and Hespanha (2006, 2007). A closure function that has recently attracted interest is known as the *zero-information* closure function (of order M) Smadbeck and Kaznessis (2013), and is given by

$$(\varphi(\mu))_i = \left\langle p_\mu^\star, x^{M+i} \right\rangle\,, \text{ for } i = 1, 2, \dots\,, \tag{24}$$

where $p_\mu^\star$ denotes the maximizer to the problem (4), where $T = \times_{i=1}^{M} \left\{ \left\langle \mu, x^i \right\rangle - u \;:\; u \in \mathcal{U}_i \right\}$, where $\mathcal{U}_i = [-\kappa, \kappa] \subset \mathbb{R}$ for all i and for a given $\kappa > 0$, that acts as a regularizer, in the sense of Assumption 1. This approximation reduces the infinite-dimensional ODE (23) to a finite-dimensional ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\mu(t) = A\mu(t) + B\varphi(\mu(t))\,. \tag{25}$$

To numerically solve (25) it is crucial to have an efficient evaluation of the closure function φ. In the zero-information closure scheme this is given by to the entropy maximization problem (4) and as such can be addressed using Algorithm 1.

To illustrate this point, we consider a reversible dimerisation reaction where two monomers ($\mathcal{M}$) combine in a second-order monomolecular reaction to form a dimer ($\mathcal{D}$); the reverse reaction is first order and involves the decomposition of the dimer into the two monomers. This gives rise to the chemical reaction system

$$\begin{aligned} 2\mathcal{M} &\xrightarrow{k_1} \mathcal{D} \\ \mathcal{D} &\xrightarrow{k_2} 2\mathcal{M}\,, \end{aligned} \tag{26}$$

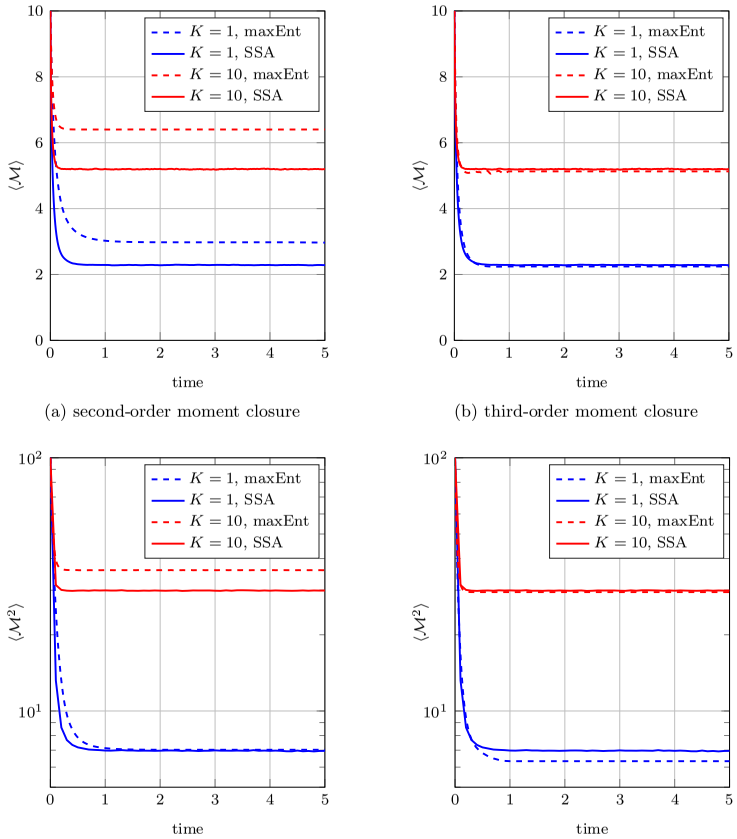
with reaction rate constants $k_1, k_2 > 0$. Note that the system as described has a single degree of freedom since $\mathcal{M} = 2\mathcal{D}_0 - 2\mathcal{D} + \mathcal{M}_0$, Where $\mathcal{M}$ denotes the count of the monomers, $\mathcal{D}$ the count of dimers, and $\mathcal{M}_0$ and $\mathcal{D}_0$ the corresponding initial conditions. Therefore, the matrices can be reduced to include only the moments of one component as a simplification and as such the zero-information closure function (24) consists of solving a one-dimensional entropy maximization problem such as given by (4), where the support are the natural numbers (upper bounded by $\mathcal{M}_0 + \mathcal{D}_0$ and hence compact). For illustrative purposes, let us look at a second order closure scheme, where the corresponding moment vectors are defined as $\mu = (1, \langle \mathcal{M} \rangle, \langle \mathcal{M}^2 \rangle)^\top \in \mathbb{R}^3$ and $\zeta = \langle \mathcal{M}^3 \rangle \in \mathbb{R}$ and the corresponding matrices are given by

$$A = \begin{pmatrix} 0 & 0 & 0 \\ k_2 S_0 & 2k_1 - k_2 & -2k_1 \\ 2k_2 S_0 & 2k_2(S_0 - 1) - 4k_1 & 8k_1 - 2k_2 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ -4k_1 \end{pmatrix},$$

where $S_0 = \mathcal{M}_0 + 2\mathcal{D}_0$. The simulation results, Figure 2, show the time trajectory for the average and the second moment of the number of $\mathcal{M}$ molecules in the reversible dimerization model (26), as calculated for the zero information closure (25) using Algorithm 1, for a second-order closure as well as a third-order closure. To solve the ODE (25) we use an explicit Runge-Kutta (4,5) formula (ode45) built into MATLAB. The results are compared to the average of 10^6 SSA Wilkinson (2006) trajectories. It can be seen how increasing the order of the closure method improves the approximation accuracy.

7. Approximate dynamic programming for constrained Markov decision processes

In this section, we show that problem (3) naturally appears in the context of approximate dynamic programming, which is at the heart of reinforcement learning (see Bertsekas and Tsitsiklis (1996); Bertsekas (1995); Recht (2018) and references therein). We consider



(a) second-order moment closure (b) third-order moment closure

(c) second-order moment closure (d) third-order moment closure

Figure 2: Reversible dimerization system (26) with reaction constants $K = {}^{k_2}/_{k_1}$: Comparison of the zero-information moment closure method (25), solved using Algorithm 1 and the average of 10^6 SSA trajectories. The initial conditions are $\mathcal{M}_0 = 10$ and $\mathcal{D}_0 = 0$ and the regularization term $\kappa = 0.01$.

constrained Markov decision processes (MDPs) that form an important class of stochastic control problems with applications in many areas; see Piunovskiy (1997); Altman (1999) and the comprehensive bibliography therein. A look at these references shows that most of the literature concerns constrained MDPs where the state and action spaces are either finite or countable. Inspired by the recent work Mohajerin Esfahani et al. (2018), we show here

that the entropy maximization problem (3) is a key element to approximate constrained MDPs on general, possibly uncountable, state and action spaces.

7.1. Constrained MDP problem formulation

Consider a discrete-time constrained MDP $\big(S, A, \{A(s) : s \in S\}, Q, c, d, \kappa\big)$, where S (resp. A) is a metric space called *state space* (resp. *action space*) and for each $s \in S$ the measurable set $A(s) \subseteq A$ denotes the set of *feasible actions* when the system is in state $s \in S$. The *transition law* is a stochastic kernel Q on S given the feasible state-action pairs in $\mathbb{K} := \{(s,a) : s \in S, a \in A(s)\}$. A stochastic kernel acts on real-valued measurable functions u from the left as $Qu(s,a) := \int_S u(s')Q(\mathrm{d}s'|s,a)$, for all $(s,a) \in \mathbb{K}$, and on probability measures μ on $\mathbb{K}$ from the right as $\mu Q(B) := \int_K Q(B|s,a)\mu\big(\mathrm{d}(s,a)\big)$, for all $B \in \mathcal{B}(S)$. The (measurable) function $c : \mathbb{K} \to \mathbb{R}_+$ denotes the so-called *one-stage cost function*, and the (measurable) function $d : \mathbb{K} \to \mathbb{R}^q$ the one-stage constraint cost along with the preassigned budget level $\kappa \in \mathbb{R}^q$.

In short, the MDP model described above may read as follows: When the system at the state $s \in S$ deploys the action $a \in A(s)$, it incurs the one-stage cost and constraint $c(s,a)$ and $d(s,a)$, respectively, and subsequently lands in the next state whose distribution is supported on S and described via $Q(\cdot|s,a)$. We consider the expected long-run average cost criteria[4], i.e., for any measurable function $\psi : \mathbb{K} \to \mathbb{R}^p$ with $p \in \{1, q\}$, we define

$$J_\psi(\pi,\nu) := \limsup_{n\to\infty} \frac{1}{n}\mathbb{E}_\nu^\pi\left(\sum_{t=0}^{n-1}\psi(s_t,a_t)\right).$$

Using the defintion above, the central object of this section is the optimization problem

$$J^\star := \begin{cases} \inf\limits_{\nu,\pi} & J_c(\pi,\nu) \\ \text{s.t.} & J_d(\pi,\nu) \le \kappa \\ & \nu \in \mathcal{P}(S),\ \pi \in \Pi, \end{cases} \tag{27}$$

where Π denotes the set of all control policies. We refer the reader to Hernández-Lerma and Lasserre (1996); Hernández-Lerma et al. (2003); Altman (1999) for a detailed mathematical treatment of this setting. It is well known that the MDP problem (27) can be stated equivalently as an infinite-dimensional linear program and its corresponding dual

$$J_P^\star := \begin{cases} \inf\limits_{\mu} & \langle \mu, c\rangle \\ \text{s.t.} & \mu(B\times A) = \mu Q(B) \quad \forall B \in \mathbb{B}(S) \\ & \langle \mu, d\rangle \le \kappa \\ & \mu \in \mathcal{P}(\mathbb{K}), \end{cases} \tag{28a}$$

$$J_D^\star := \begin{cases} \sup\limits_{u,\rho,\gamma} & \rho - \gamma^\top\kappa \\ \text{s.t.} & \rho + u(x) - Qu(x,a) \le c(x,a) + \gamma^\top d(x,a) \quad \forall (x,a) \in \mathbb{K} \\ & \langle \mu, d\rangle \le \kappa \\ & u \in, \rho \in \mathbb{R}, \gamma \in \mathbb{R}_+^q. \end{cases} \tag{28b}$$

4. We refer the interested reader to Mohajerin Esfahani et al. (2018) for extension to the discounted cost.

The following regularity assumption is required in order to ensure that the solutions are well posed and that equivalence between (27) and the LPs (28a), (28b) holds.

Assumption 3 (Control model) *We stipulate that*

(i) the set of feasible state-action pairs is the unit hypercube $\mathbb{K} = [0,1]^{\dim(S\times A)}$*;*

(ii) the transition law Q *is Lipschitz continuous, i.e., there exists* $L_Q > 0$ *such that for all* $k, k' \in \mathbb{K}$ *and all continuous functions* u

$$|Qu(k) - Qu(k')| \le L_Q \|u\|_\infty \|k - k'\|_{\ell_\infty};$$

(iii) the cost function c *is non-negative and Lipschitz continuous on* $\mathbb{K}$ *and* d *is continuous on* $\mathbb{K}$*.*

Under this assumption, strong duality between the linear programs (28a) and (28b) holds (i.e., the supremum and infimum are attained and $J_P^\star = J_D^\star$). Moreover, the LP formulation is equivalent to the original problem (27) in the sense that $J^\star = J_P^\star = J_D^\star$, see (Hernández-Lerma et al., 2003, Theorem 5.2).

Finding exact solutions to either (28a) or (28b) generally is impossible as the linear programs are infinite dimensional. This challenge has given rise to a wealth of approximation schemes in the literature under the names of approximate dynamic programming. Typically, one restricts decision space in (28b) to a finite dimensional subspace spanned by basis functions $\{u_i\}_{i=1}^n \subset \mathcal{L}(S)$ denoted by $\mathbb{U}_n := \{\sum_{i=1}^n \alpha_i u_i \ : \ \|\alpha\|_2 \le \theta\}$. Motivated by de Farias and Roy (2004); Mohajerin Esfahani et al. (2018) we then approximate the solution $J^\star$ by

$$J_{P,n}^\star := \begin{cases} \inf\limits_{\mu} & \langle \mu, c\rangle + \theta\|\mathcal{T}_n\mu - e\|_2 \\ \text{s.t.} & \langle \mu, d\rangle \le \kappa \\ & \mu \in \mathcal{P}(\mathbb{K}), \end{cases} \tag{29}$$

where the operator $\mathcal{T}_n : \mathcal{P}(\mathbb{K}) \to \mathbb{R}^{n+1}$ is defined as $(\mathcal{T}_n\mu)_1 = -1, (\mathcal{T}_n\mu)_{i+1} = \langle Qu_i - u_i, \mu\rangle$, $i = 1,\dots,n$ and $e := (-1, 0, \dots, 0) \in \mathbb{R}^{n+1}$. The optimization problem (29) can be solved with an accelerated first-order method provided by Algorithm 2, stated below, where each iteration step involves solving a problem an entropy maximization problem of the form (3). We define

$$c_{\alpha,\zeta} := \frac{1}{\zeta}(c - \alpha_0 + \sum_{i=1}^n \alpha_i(Qu_i - u_i)), \qquad \mathcal{Y} := \{\mu \in \mathcal{P}(\mathbb{K}) \ : \ \langle \mu, d\rangle \le \kappa\},$$
$$\mathbb{T}(q,\alpha) := (\alpha - q)\min\{1, \theta\|q-\alpha\|_2^{-1}\}, \quad \mathrm{y}_\zeta^\star(\alpha) := \arg\min_{y\in\mathcal{Y}} \left\{\mathsf{D}(y\|\lambda) + \langle y, c_{\alpha,\zeta}\rangle\right\}. \tag{30}$$

Steps 2 and 3 of Algorithm 2 are simple arithmetic operations. Step 1 relies on the solution to the optimization problem (30) that can be reduced to (3). To see this, note that the additional linear term in the objective function can be directly integrated into the analysis of Section 3. We provide the explicit construction in Section 7.2 for an inventory management system. We are interested in establishing bounds on the quality of the approximate solution $J_{n,\zeta}^{(k)}$ obtained by Algorithm 2. The results of (Mohajerin Esfahani et al., 2018, Theorem 3.3 and 5.3) allow us to obtain the following bound.

Algorithm 2: Approximate dynamic programming scheme

Input: $n, k \in \mathbb{N}$, $\zeta, \theta > 0$, and $w^{(0)} \in \mathbb{R}^{n+1}$ such that $\|w^{(0)}\|_2 \le \theta$

For $0 \le \ell \le k$ **do**
- **Step 1:** Define $r^{(\ell)} := \frac{\zeta}{4n}(e - \mathcal{T}_n \mathrm{y}_\zeta^\star(w^{(\ell)}))$
- **Step 2:** Let $z^{(\ell)} := \mathbb{T}\left(\sum_{j=0}^\ell \frac{j+1}{2} r^{(j)}, 0\right)$ and $\beta^{(\ell)} = \mathbb{T}\left(r^{(\ell)}, w^{(\ell)}\right)$
- **Step 3:** Set $w^{(\ell+1)} = \frac{2}{\ell+3} z^{(\ell)} + \frac{\ell+1}{\ell+3}\beta^{(\ell)}$

Output: $J_{n,\zeta}^{(k)} := \langle c, \hat{y}_\zeta\rangle + \theta\|\mathcal{T}_n\hat{y}_\zeta - e\|_2$ with $\hat{y}_\zeta := \sum_{j=0}^k \frac{2(j+1)}{(k+1)(k+2)} \mathrm{y}_\zeta^\star(w^{(j)})$

Theorem 11 (Approximation error) *Under Assumption 3, Algorithm 2 provides an approximation to* (27) *with the following error bound*

$$|J_{n,\zeta}^{(k)} - J^\star| \le (1 + \max\{L_Q, 1\})\|u^\star - \Pi_{\mathbb{U}_n}(u^\star)\|_L + \left(\frac{4n^2\theta}{k^2\zeta} + \dim(\mathbb{K})\zeta\max\{\log(\beta/\zeta), 1\}\right),$$

where $\beta := \frac{e}{\dim(\mathbb{K})}(\theta\sqrt{n}(\max\{L_Q, 1\} + 1) + \|c\|_L)$.

Note that the bound depends on the projection residual $\|u^\star - \Pi_{\mathbb{U}_n}(u^\star)\|_L$, where the projection mapping is defined as $\Pi_{\mathbb{U}_n}(u^\star) := \arg\min_{u\in\mathbb{U}_n} \|u^\star - u\|_2$, the parameters of the problem (notably the dimensions of the state and action spaces, the stage cost, and the Lipschitz constant of the kernel L_Q), and the design choices of the algorithm (the number of basis functions n, norm bound θ and ζ).

Remark 12 (Tuning parameters)

(i) The residual error $\|u^\star - \Pi_{\mathbb{U}_n}(u^\star)\|_L$ *can be approximated by leveraging results from the literature on universal function approximation. Prior information about the value function* $u^\star$ *may offer explicit quantitative bounds. For instance, for MDP under Assumption 3 we know that* $u^\star$ *is Lipschitz continuous. For appropriate choice of basis functions, we can therefore ensure a convergence rate of* $n^{-1/\dim(S)}$*, see for instance Farouki (2012) for polynomials and Olver (2009) for the Fourier basis functions.*

(ii) The regularization parameter θ *has to be chosen such that* $\theta > \|c\|_L$*. An optimal choice for* θ *and* ζ *is described in (Mohajerin Esfahani et al., 2018, Remark 4.6 and Theorem 5.3).*

7.2. Inventory management system

Consider an inventory model in which the state variable describes the stock level s_t at the beginning of period t. The control or action variable a_t at t is the quantity ordered and immediately supplied at the beginning of period t, and the "disturbance" or "exogenous" variable ξ_t is the demand during that period. We assume ξ_t to be i.i.d. random variables following an exponential distribution with parameter λ. The system equation Hernández-Lerma and Lasserre (1996) is

$$s_{t+1} = \max\{0, s_t + a_t - \xi_t\} =: (s_t + a_t - \xi_t)_+, \quad t = 0, 1, 2, \dots. \tag{31}$$

We assume that the system has a finite capacity C. Therefore $S = A = [0, C]$ and since the current stock plus the amount ordered cannot exceed the system's capacity, the set of feasible actions is $A(s) = [0, C - s]$ for every $s \in S$. Suppose we wish to maximize an expected profit for operating the system, we might take the net profit at stage t to be

$$r(s_t, a_t, \xi_t) := v \min\{s_t + a_t, \xi_t\} - pa_t - h(s_t + a_t), \tag{32}$$

which is of the form "profit = sales - production cost - holding cost". In (32), v, p and h are positive numbers denoting unit sale price, unit production cost, and unit holding cost, respectively. To write the cost (32) in the form of our control model (27), we define

$$\begin{aligned} c(s, a) :=\ & \mathbb{E}[-r(s_t, a_t, \xi_t) | s_t = s, a_t = a] \\ =\ & -v(s + a)e^{-\lambda(s+a)} - \frac{v}{\lambda}\left(1 - e^{-\lambda(s+a)}\left(\lambda(s + a) + 1\right)\right) + pa + h(s + a). \end{aligned} \tag{33}$$

Note that non-negativity of the cost can be ensured by subtracting the term $2vC$ from (32). We assume that there are regulatory constraints on the required stock level s_t, for example to avoid the risk of running into a shortage of a certain critical product. For simplicity, let us assume that the regulator enforces constraints on the long-term first and second moments of the stock s_t in the following sense

$$\limsup_{n\to\infty} \frac{1}{n}\mathbb{E}_\nu^\pi\left(\sum_{t=0}^{n-1} s_t\right) \geq \ell_1, \quad \text{and} \quad \limsup_{n\to\infty} \frac{1}{n}\mathbb{E}_\nu^\pi\left(\sum_{t=0}^{n-1} s_t^2\right) \leq \ell_2, \tag{34}$$

for given $\ell_1, \ell_2 \in \mathbb{R}_+$, where we assume that $\ell_1^2 < \ell_2$[5]. To express it in the form of our control model (27), we define $d_1(s, a) := -s$, $d_2(s, a) := s^2$, and $\kappa := (-\ell_1, \ell_2) \in \mathbb{R}^2$. From the described assumptions on the constants ℓ_1 and ℓ_2, it can be directly seen that the Slater point assumption described in Lemma 3 holds (consider the set $T := \{x \in \mathbb{R}^2 \ : \ x_1 \geq \ell_1, x_2 \leq \ell_2, x_1^2 \leq x_2\}$).

In the following, we will describe how Algorithm 2 and the respective error bounds given by Theorem 11 can be applied to the inventory management system. In particular we show how the computationally demanding part of Algorithm 2, given by (30), is directly addressed by the methodology presented in Section 3. To fulfill Assumption 3, we equivalently reformulate the above problem using the dynamics

$$s_{t+1} = (\min\{C, s_t + a_t\} - \xi_t)_+, \quad t = 0, 1, 2, \ldots, \tag{35}$$

where the admissible actions set is now the state-independent set $A = [0, C]$. Finally by normalizing the state and action variables through the definitions $\tilde{s}_t := \frac{s_t}{C}$ and $\tilde{a}_t := \frac{a_t}{C}$, we have

$$\tilde{s}_{t+1} = \left(\min\{1, \tilde{s}_t + \tilde{a}_t\} - \frac{\xi_t}{C}\right)_+, \quad t = 0, 1, 2, \ldots. \tag{36}$$

Furthermore, it can be seen directly using Leibnitz' rule that the transition law is Lipschitz continuous and $L_Q \leq \sqrt{2}C\lambda$. It remains to argue how (30) can be addressed by the

5. Note that the enforced constraints (34) imply the upper and lower bounds $\ell_1 \leq \limsup_{n\to\infty} \frac{1}{n}\mathbb{E}_\nu^\pi\left(\sum_{t=0}^{n-1} s_t\right) \leq \sqrt{\ell_2}$ and $\ell_1^2 \leq \limsup_{n\to\infty} \frac{1}{n}\mathbb{E}_\nu^\pi\left(\sum_{t=0}^{n-1} s_t^2\right) \leq \ell_2$.

methodology presented in Section 3. We introduce the linear operator $\mathcal{A} : \mathcal{P}(\mathbb{K}) \to \mathbb{R}^2$, defined by $(\mathcal{A}\mu)_i := \langle \mu, d_i \rangle$ for $i = 1, 2$. Then, the optimization problem (30) can be expressed as

$$\text{(primal program)}: \quad J^\star = \min_{\mu\in\mathcal{P}(\mathbb{K})} \left\{ \mathsf{D}\big(\mu\|\nu\big) - \langle \mu, c_{\alpha,\zeta} \rangle + \sup_{z\in\mathbb{R}^M} \left\{ \langle \mathcal{A}\mu, z \rangle - \sigma_T(z) \right\} \right\}, \tag{37}$$

$$\text{(dual program)}: \quad J_{\mathsf{D}}^\star = \sup_{z\in\mathbb{R}^M} \left\{ -\sigma_T(z) + \min_{\mu\in\mathcal{P}(\mathbb{K})} \left\{ \mathsf{D}\big(\mu\|\nu\big) + \langle \mu, \mathcal{A}^* z - c_{\alpha,\zeta} \rangle \right\} \right\}, \tag{38}$$

which apart from the additional linear term are in the form of (6) and (7). Due to our assumptions on ℓ_1 and ℓ_2 described above, there exists a strictly feasible solution to (30), i.e., $\mu_0 \in \mathcal{P}(\mathbb{K})$ such that $\mathcal{A}\mu_0 \in T$ and $\delta := \min_{y\in T^c} \|\mathcal{A}\mu_0 - y\|_2 > 0$. Hence, the method from Section 3 is applicable.

Numerical Simulation. For a given set of model parameters ($C = 1$, $\lambda = \frac{1}{2}$, $v = 1$, $p = \frac{1}{2}$, $h = \frac{1}{10}$) we consider four different scenarios:

(i) Unconstrained case (i.e., $\ell_1 = 0$ and $\ell_2 = 1$);

(ii) $\ell_1 = 0.5$ and $\ell_2 = 0.4$;

(iii) $\ell_1 = 0.5$ and $\ell_2 = 0.3$, which is strictly more constrained than scenario (ii);

(iv) $\ell_1 = 0.1$ and $\ell_2 = 0.1$.

For each scenario we run Algorithm 2 and plot the value of the resulting approximation $J_{n,\zeta}^{(k)}$ as a function of the number of iterations in Figure 3. In each iteration of Algorithm 2 the variable (30) is required, which is computed with Algorithm 1. The following simulation parameters were used:

Algorithm 1: $\eta_1 = \eta_2 = 10^{-3}$, 1500 iterations;

Algorithm 2: $\zeta = 10^{-1.5}$, $\theta = 3$, $n = 10$, Fourier basis $u_{2i-1}(s) = \frac{C}{2i\pi}\cos\left(\frac{2i\pi s}{C}\right)$ and $u_{2i}(s) = \frac{C}{2i\pi}\sin\left(\frac{2i\pi s}{C}\right)$ for $i = 1, \ldots, \frac{n}{2}$.

As shown in Figure 3, the value of $J_{n,\zeta}^{(k)}$ converges at around 1000 iterations of Algorithm 2.[6] The fact that scenario (ii) is a (strict) relaxation in terms of the constraints compared to scenario (iii) is visualized by the numerical simulation as the expected profit of scenario (ii) is continuously higher compared to scenario (iii). Figure 3 also indicates that in Scenario (iv) the constraints are the most restrictive.

8. Conclusion and future work

We presented an approximation scheme to a generalization of the classical problem of estimating a density via a maximum entropy criterion, given some moment constraints. The key idea used is to apply smoothing techniques to the non-smooth dual function of the entropy maximization problem, that enables us to solve the dual problem efficiently with fast

6. 1000 iterations of Algorithm 2 took around 5.5 hours with Matlab on a laptop with a 2.2 GHz Intel Core i7 processor.

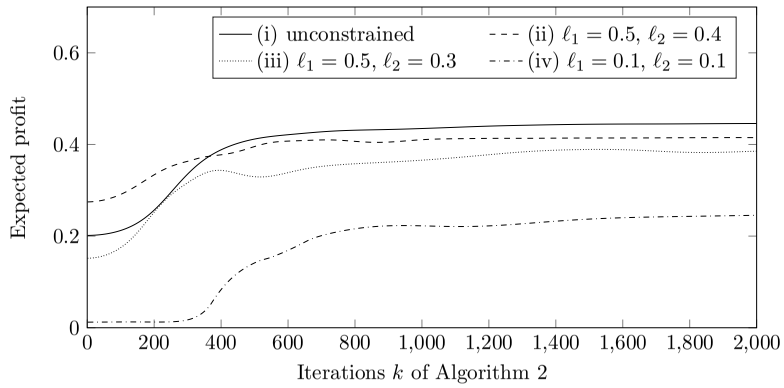


Figure 3: The expected profit of the inventory system, approximated by $-J_{n,\zeta}^{(k)}$, resulting from Algorithm 2 is displayed for four different scenarios (i)-(iv) representing four different constraints on the inventory system.

gradient methods. Due to the favorable structure of the considered entropy maximization problem, we provide explicit error bounds on the approximation error as well as a-posteriori error estimates.

The proposed method requires one to evaluate the gradient (21) in every iteration step, which, as highlighted in Section 5, in the infinite-dimensional setting involves an integral. As such the method used to compute those integrals has to be included to the complexity of the proposed algorithm and, in higher dimensions, may become is the dominant factor. Therefore, it would be interesting to investigate this integration step in more detail. Two approaches, one based on semidefinite programming and another invoking Quasi-Monte Carlo integration techniques, are briefly sketched. What remains open is to quantify the accuracy required in the gradient approximations, which could be done along the lines of Devolder et al. (2014). Another potential direction, would be to test the proposed numerical method in the context of approximating the channel capacity of a large class of memoryless channels Sutter et al. (2015), as mentioned in the introduction.

Finally it should be mentioned that the approximation scheme proposed in this article can be further generalized to quantum mechanical entropies. In this setup probability mass functions are replaced by density matrices (i.e., positive semidefinite matrices, whose trace is equal to one). The von Neumann entropy of such a density matrix ρ is defined by $H(\rho) := -\mathrm{tr}(\rho \log \rho)$, which reduces to the (Shannon) entropy in case the density matrix ρ is diagonal. Also the relative entropy can be generalized to the quantum setup Umegaki (1962) and general treatment of our approximation scheme, its analysis can be lifted to the this (strictly) more general framework. As demonstrated in Sutter et al. (2016), (quantum) entropy maximization problems can be used to efficiently approximate the classical capacity of quantum channels.

Appendix A. Detailed proof of Theorem 5

Given the bounds $\|z^\star\|_2 \leq \frac{C}{\delta}$ and $F(z) \leq J^\star \leq \mathsf{D}\big(\mu_0 \| \nu\big) = C$ for all $z \in \mathbb{R}^M$ as argued above, we show how the error bounds of Theorem 5 follow from Devolder et al. (2012). The dual ε-optimality (15a) is derived from (Devolder et al., 2012, Equation (7.8)), that in our setting states

$$0 \leq J^\star - F(\hat{z}_k) \leq \frac{3\varepsilon}{4} + 5\left(F(z^\star) - F(0) + \frac{\varepsilon}{2}\right) e^{-\frac{k}{2}\sqrt{\frac{\eta_2}{L(\eta)}}}. \tag{39}$$

Using the parameters as defined in Theorem 5, i.e.,

$$C := \mathsf{D}\big(\mu_0 \| \nu\big), \qquad D := \frac{1}{2}\max_{x \in T} \|x\|_2, \qquad \eta_1(\varepsilon) := \frac{\varepsilon}{4D}, \qquad \eta_2(\varepsilon) := \frac{\varepsilon\delta^2}{2C^2}, \tag{40}$$

the fact that $F(z^\star) - F(0) = F(z^\star) \leq C$, and the Lipschitz constant $L(\eta) = \frac{1}{\eta_1} + \|\mathcal{A}\|^2 + \eta_2$ derived in Lemma 4, inequality (39) ensures $0 \leq J^\star - F(\hat{z}_k) \leq \varepsilon$ if

$$k \geq N_1(\varepsilon) := 2\left(\sqrt{\frac{8DC^2}{\varepsilon^2\delta^2} + \frac{2\|\mathcal{A}\|^2 C^2}{\varepsilon\delta^2} + 1}\right) \ln\left(\frac{10(\varepsilon + 2C)}{\varepsilon}\right).$$

The primal ε-optimality bound (15b) following the derivations in Devolder et al. (2012) is implied if k is chosen large enough such that $\|\nabla F_\eta(z_k)\|_2 \leq \frac{2\varepsilon\delta}{C}$. According to (Devolder et al., 2012, Equation (7.11)) the gradient can be bounded by

$$\|\nabla F_\eta(z_k)\|_2 \leq \sqrt{4L(\eta)\left(F(z^\star) - F(0) + \frac{\varepsilon}{2}\right)} e^{-\frac{k}{2}\sqrt{\frac{\eta_2}{L(\eta)}}} + 2\sqrt{3}\eta_2 \frac{C}{\delta}. \tag{41}$$

Again using the parameters and constants as we did for the dual ε-optimality bound we find that $\|\nabla F_\eta(z_k)\|_2 \leq \frac{2\varepsilon\delta}{C}$ if

$$k \geq N_2(\varepsilon) := 2\left(\sqrt{\frac{8DC^2}{\varepsilon^2\delta^2} + \frac{2\|\mathcal{A}\|^2 C^2}{\varepsilon\delta^2} + 1}\right) \ln\left(\frac{C}{\varepsilon\delta(2-\sqrt{3})}\sqrt{4\left(\frac{4D}{\varepsilon} + \|\mathcal{A}\|^2 + \frac{\varepsilon\delta^2}{2C^2}\right)\left(C + \frac{\varepsilon}{2}\right)}\right).$$

The primal ε-feasiblility (15c) finally directly follows from (Devolder et al., 2012, Equation (7.14)) and the bound $\|z^\star\|_2 \leq \frac{C}{\delta}$.

Acknowledgments

The authors thank Andreas Milias-Argeitis for helpful discussions regarding the moment closure example. TS and JL acknowledge "SystemsX" under the grant "SignalX". DS acknowledges support by the Swiss National Science Foundation (SNSF) via the National Centre of Competence in Research "QSIT" and by the Air Force Office of Scientific Research (AFOSR) via grant FA9550-16-1-0245. PME was supported by the Swiss National Science Foundation under grant "P2EZP2 165264".

References

N. I. Akhiezer. *The classical moment problem and some related questions in analysis*. Translated by N. Kemmer. Hafner Publishing Co., New York, 1965.

Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Springer, 2007. ISBN 9783540326960. doi: 10.1007/3-540-29587-9.

Eitan Altman. *Constrained Markov decision processes*. Stochastic Modeling. Chapman & Hall/CRC, Boca Raton, FL, 1999. ISBN 0-8493-0382-6.

Edward J. Anderson and Peter Nash. *Linear programming in infinite-dimensional spaces: theory and applications*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, 1987. ISBN 9780471912507.

Arash Asadpour, Michel X. Goemans, Aleksander M?dry, Shayan Oveis Gharan, and Amin Saberi. *An $O(\log n/\log\log n)$-approximation Algorithm for the Asymmetric Traveling Salesman Problem*, pages 379–389. 2010. doi: 10.1137/1.9781611973075.32. URL `https://epubs.siam.org/doi/abs/10.1137/1.9781611973075.32`.

D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Dimitri P. Bertsekas. *Convex Optimization Theory*. Athena Scientific optimization and computation series. Athena Scientific, 2009. ISBN 9781886529311.

Dimitris Bertsimas, Xuan Vinh Doan, and Jean Lasserre. Approximating integrals of multivariate exponentials: a moment approach. *Oper. Res. Lett.*, 36(2):205–210, 2008. ISSN 0167-6377. doi: 10.1016/j.orl.2007.07.002. URL `http://dx.doi.org/10.1016/j.orl.2007.07.002`.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013. ISBN 978-0-19-953525-5. doi: 10.1093/acprof:oso/9780199535255.001.0001. URL `http://dx.doi.org/10.1093/acprof:oso/9780199535255.001.0001`.

I. Csiszár. *I*-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3:146–158, 1975. doi: 10.1214/aop/1176996454.

D.P. de Farias and B. Van Roy. On Constraint Sampling in the Linear Programming Approach to Approximate Dynamic Programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

Olivier Devolder, Francois Glineur, and Yurii Nesterov. Double smoothing technique for large-scale linearly constrained convex optimization. *SIAM Journal on Optimization*, 22 (2):702–727, 2012. doi: 10.1137/110826102.

Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 146(1-2, Ser. A):37–75, 2014.

ISSN 0025-5610. doi: 10.1007/s10107-013-0677-5. URL `https://doi.org/10.1007/s10107-013-0677-5`.

Josef Dick, Frances Y. Kuo, and Ian H. Sloan. High-dimensional integration: The Quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 5 2013. ISSN 1474-0508. doi: 10.1017/S0962492913000044. URL `http://journals.cambridge.org/article_S0962492913000044`.

Miroslav Dudik, Steven J. Phillips, and Robert E. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, 8:1217–1260, 2007. ISSN 1532-4435.

Rida T. Farouki. The Bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29(6):379 – 419, 2012.

Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS 11, pages 550–559, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4571-4. doi: 10.1109/FOCS.2011.80. URL `http://dx.doi.org/10.1109/FOCS.2011.80`.

C. S. Gillespie. Moment-closure approximations for mass-action models. *IET Systems Biology*, 3(1):52–58, January 2009. ISSN 1751-8849. doi: 10.1049/iet-syb:20070031.

Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Computational Phys.*, 22(4):403–434, 1976. ISSN 0021-9991. doi: 10.1016/0021-9991(76)90041-3.

Daniel T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55, 2007. doi: 10.1146/annurev.physchem.58.032806.104637. URL `https://doi.org/10.1146/annurev.physchem.58.032806.104637`. PMID: 17037977.

Daniel T Gillespie, Andreas Hellander, and Linda R Petzold. Perspective: Stochastic algorithms for chemical kinetics. *The Journal of Chemical Physics*, 138(17):170901, 2013. doi: 10.1063/1.4801941.

Amos Golan. Information and entropy econometrics a review and synthesis. *Foundations and Trends in Econometrics*, 2:1–145, 2008. ISSN 1551-3076. doi: 10.1561/0800000004. URL `http://dx.doi.org/10.1561/0800000004`.

Peter Grünwald. Entropy concentration and the empirical coding game. *Statistica Neerlandica*, 62(3):374–392, 2008. ISSN 1467-9574. doi: 10.1111/j.1467-9574.2008.00401.x. URL `http://dx.doi.org/10.1111/j.1467-9574.2008.00401.x`.

O. Hernández-Lerma and J.B. Lasserre. *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. Applications of Mathematics Series. Springer, 1996. ISBN 9780387945798.

O. Hernández-Lerma and J.B. Lasserre. *Further topics on discrete-time Markov control processes*. Applications of Mathematics Series. Springer, 1999. ISBN 9780387986944. doi: 10.1007/978-1-4612-0561-6.

Onésimo Hernández-Lerma, Juan González-Hernández, and Raquiel R. López-Martínez. Constrained average cost Markov control processes in Borel spaces. *SIAM J. Control Optim.*, 42(2):442–468, 2003. ISSN 0363-0129. doi: 10.1137/S0363012999361627. URL `https://doi.org/10.1137/S0363012999361627`.

E. T. Jaynes. *Probability theory: The logic of science*. Cambridge University Press, Cambridge, 2003. ISBN 0-521-59271-2. doi: 10.1017/CBO9780511790423. URL `http://dx.doi.org/10.1017/CBO9780511790423`.

Solomon Kullback. *Information theory and statistics*. John Wiley and Sons, Inc., New York; Chapman and Hall, Ltd., London, 1959.

Frances Y Kuo and Ian H Sloan. Lifting the curse of dimensionality. *Notices of the AMS*, 52(11):1320–1328, 2005.

Jean B. Lasserre. *Moments, Positive Polynomials and Their Applications*. Imperial College Press optimization series. Imperial College Press, 2009. ISBN 9781848164468. doi: 10.1142/9781848164468_0001.

David G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., New York-London-Sydney, 1969.

Lawrence R. Mead and N. Papanicolaou. Maximum entropy in the problem of moments. *J. Math. Phys.*, 25(8):2404–2417, 1984. ISSN 0022-2488. doi: 10.1063/1.526446. URL `http://dx.doi.org/10.1063/1.526446`.

Peyman Mohajerin Esfahani, Tobias Sutter, Daniel Kuhn, and John Lygeros. From infinite to finite programs: explicit error bounds with applications to approximate dynamic programming. *SIAM J. Optim.*, 28(3):1968–1998, 2018. ISSN 1052-6234. doi: 10.1137/17M1133087. URL `https://doi.org/10.1137/17M1133087`.

Angelia Nedić and Asuman Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2008.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer, 2004. ISBN 9781402075537. doi: 10.1007/978-1-4419-8853-9.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005. ISSN 0025-5610. doi: 10.1007/s10107-004-0552-5. URL `http://dx.doi.org/10.1007/s10107-004-0552-5`.

Harald Niederreiter. Quasi-Monte Carlo methods. *Encyclopedia of quantitative finance*, 2010.

Sheehan Olver. On the convergence rate of a modified Fourier series. *Mathematics of Computation*, 78(267):1629–1645, 2009.

A. B. Piunovskiy. *Optimal control of random sequences in problems with constraints*, volume 410 of *Mathematics and its Applications*. Kluwer Academic Publishers, Dordrecht, 1997. ISBN 0-7923-4571-1. doi: 10.1007/978-94-011-5508-3. URL `https://doi.org/10.1007/978-94-011-5508-3`. With a preface by V. B. Kolmanovskii and A. N. Shiryaev.

Benjamin Recht. A Tour of Reinforcement Learning: The View from Continuous Control. *arXiv e-prints*, art. arXiv:1806.09460, June 2018.

Stefan Richter. Computational complexity certification of gradient methods for real-time model predictive control. *PhD thesis, ETH Zurich*, 2012. doi: 10.3929/ethz-a-007587480.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2 edition, 2004.

R. Tyrrell Rockafellar. *Convex analysis*. Princeton Landmarks in Mathematics and Physics Series. Princeton University Press, 1997. ISBN 9780691015866.

A. Singh and J. P. Hespanha. Lognormal moment closures for biochemical reactions. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 2063–2068, Dec 2006. doi: 10.1109/CDC.2006.376994.

Abhyudai Singh and João Pedro Hespanha. A derivative matching approach to moment closure for the stochastic logistic model. *Bull. Math. Biol.*, 69(6):1909–1925, 2007. ISSN 0092-8240. doi: 10.1007/s11538-007-9198-9. URL `http://dx.doi.org/10.1007/s11538-007-9198-9`.

Mohit Singh and Nisheeth K. Vishnoi. Entropy, optimization and counting. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 50–59, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2710-7. doi: 10.1145/2591796.2591803. URL `http://doi.acm.org/10.1145/2591796.2591803`.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8:171–176, 1958. ISSN 0030-8730.

Ian H Sloan and Henryk Woźniakowski. When are Quasi-Monte Carlo algorithms efficient for high dimensional integrals? *Journal of Complexity*, 14(1):1–33, 1998.

Patrick Smadbeck and Yiannis N. Kaznessis. A closure scheme for chemical master equations. *Proceedings of the National Academy of Sciences*, 110(35):14261–14265, 2013. doi: 10.1073/pnas.1306481110. URL `http://www.pnas.org/content/110/35/14261.abstract`.

Damian Straszak and Nisheeth K. Vishnoi. Computing maximum entropy distributions everywhere, 2017. available at `arXiv:1711.02036`.

D. Sutter, T. Sutter, P. Mohajerin Esfahani, and R. Renner. Efficient approximation of quantum channel capacities. *IEEE Transactions on Information Theory*, 62(1):578–598, Jan 2016. ISSN 0018-9448. doi: 10.1109/TIT.2015.2503755.

T. Sutter, D. Sutter, P. Mohajerin Esfahani, and J. Lygeros. Efficient approximation of channel capacities. *IEEE Transactions on Information Theory*, 61(4):1649–1666, April 2015. ISSN 0018-9448. doi: 10.1109/TIT.2015.2401002.

Maxime Thély. Stochastic programming with Quasi-Monte Carlo methods. Master's thesis, ETH Zurich, Switzerland, 2017.

H. Umegaki. Conditional expectation in an operator algebra. *Kodai Mathematical Seminar Reports*, 14:59–85, 1962. doi: 10.2996/kmj/1138844604.

N. G. van Kampen. *Stochastic processes in physics and chemistry*, volume 888 of *Lecture Notes in Mathematics*. North-Holland Publishing Co., Amsterdam-New York, 1981. ISBN 0-444-86200-5.

Darren James Wilkinson. *Stochastic modelling for systems biology*. Chapman & Hall/CRC Mathematical and Computational Biology Series. Chapman & Hall/CRC, Boca Raton, FL, 2006. ISBN 978-1-58488-540-5; 1-58488-540-8.

Journal of Machine Learning Research 20 (2019) 1-62 Submitted 9/17; Revised 8/19; Published 9/19

Decentralized Dictionary Learning Over Time-Varying Digraphs

Amir Daneshmand ADANESHM@PURDUE.EDU
Ying Sun SUN578@PURDUE.EDU
Gesualdo Scutari GSCUTARI@PURDUE.EDU
School of Industrial Engineering
Purdue University
West-Lafayette, IN, USA

Francisco Facchinei FACCHINEI@DIAG.UNIROMA1.IT
Department of Computer, Control, and Management Engineering
University of Rome "La Sapienza"
Rome, Italy

Brian M. Sadler BRIAN.M.SADLER6.CIV@MAIL.MIL
U.S. Army Research Laboratory
Adelphi, MD, USA

Editor: Sathiya Keerthi

Abstract

This paper studies Dictionary Learning problems wherein the learning task is distributed over a multi-agent network, modeled as a time-varying directed graph. This formulation is relevant, for instance, in Big Data scenarios where massive amounts of data are collected/stored in different locations (e.g., sensors, clouds) and aggregating and/or processing all data in a fusion center might be inefficient or unfeasible, due to resource limitations, communication overheads or privacy issues. We develop a unified decentralized algorithmic framework for this class of *nonconvex* problems, which is proved to converge to stationary solutions at a sublinear rate. The new method hinges on Successive Convex Approximation techniques, coupled with a decentralized tracking mechanism aiming at locally estimating the gradient of the smooth part of the sum-utility. To the best of our knowledge, this is the first provably convergent decentralized algorithm for Dictionary Learning and, more generally, bi-convex problems over (time-varying) (di)graphs.

Keywords: Decentralized algorithms, dictionary learning, directed graph, non-convex optimization, time-varying network

1. Introduction and Motivation

This paper introduces, analyzes, and tests numerically the first *provably convergent distributed method* for a fairly general class of Dictionary Learning (DL) problems. More specifically, we study the problem of finding a matrix $\mathbf{D} \in \mathbb{R}^{M\times K}$ (a.k.a. the dictionary), by which the data matrix $\mathbf{S} \in \mathbb{R}^{M\times N}$ can be represented through a matrix $\mathbf{X} \in \mathbb{R}^{K\times N}$, with a favorable structure on $\mathbf{D}$ and $\mathbf{X}$ (e.g., sparsity). We target scenarios where computational resources and data are not centrally available, but distributed over a group of I agents, which can communicate through a (possibly) *time-varying, directed* network; see Fig. 1.

©2019 Amir Daneshmand, Ying Sun, Gesualdo Scutari, Francisco Facchinei, Brian M. Sadler.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-534.html.

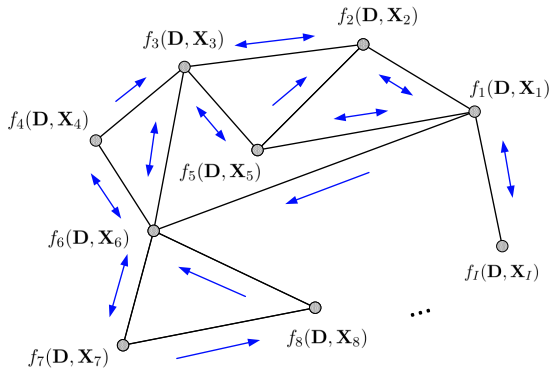


Figure 1: Directed network topology

Each agent $i \in \{1,2,\dots,I\}$ owns one block $\mathbf{S}_i \in \mathbb{R}^{M\times n_i}$ of the data $\mathbf{S} \triangleq [\mathbf{S}_1,\dots,\mathbf{S}_I]$, with $\sum_{i=1}^I n_i = N$. Partitioning the representation matrix $\mathbf{X} \triangleq [\mathbf{X}_1,\dots,\mathbf{X}_I]$ according to $\mathbf{S}$, with $\mathbf{X}_i \in \mathbb{R}^{K\times n_i}$, the class of *distributed* DL problems we aim at studying reads

$$\begin{aligned} \min_{\mathbf{D},\{\mathbf{X}_i\}_{i=1}^I} \quad & U(\mathbf{D},\mathbf{U}) \triangleq \underbrace{\sum_{i=1}^I f_i(\mathbf{D},\mathbf{X}_i)}_{\triangleq F(\mathbf{D},\mathbf{X})} + \sum_{i=1}^I g_i(\mathbf{X}_i) + G(\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{D}\in\mathcal{D},\ \mathbf{X}_i \in \mathcal{X}_i, \quad i=1,\dots,I, \end{aligned} \tag{P}$$

where $f_i : \mathcal{D}\times\mathcal{X}_i \to \mathbb{R}$ is the fidelity function of agent i, which measures the mismatch between the data $\mathbf{S}_i$ and the (local) model; this function is assumed to be smooth and *biconvex* (i.e., convex in $\mathbf{D}$ for fixed $\mathbf{X}_i$, and vice versa); $G : \mathcal{D} \to \mathbb{R}$ and $g_i : \mathcal{X}_i \to \mathbb{R}$ are (possibly non-smooth) convex functions, which are generally used to impose extra structure on the solution (e.g., low-rank or sparsity); and $\mathcal{D} \subseteq \mathbb{R}^{M\times K}$ and $\mathcal{X}_i \subseteq \mathbb{R}^{K\times n_i}$ are some closed convex sets. To avoid scaling ambiguity in the model, $\mathcal{D}$ is assumed to be bounded, without loss of generality. Since all f_i's share the common variable $\mathbf{D}$, we call it a *shared* variable and, by the same token, $\mathbf{X}_i$'s are termed *private* variables. Note that, in this distributed setting, agent i knows only its own functions f_i (and g_i) but not $\sum_{j\neq i} f_j$. Hence, agents aim to cooperatively solve Problem P leveraging local communications with their neighbors.

Problem P encompasses several DL-based formulations of practical interest, corresponding to different choices of the fidelity functions, regularizers, and feasible sets; examples include the *elastic net* (Zou and Hastie, 2005) sparse DL, sparse PCA (Shen and Huang, 2008), non-negative matrix factorization and low-rank approximation (Hastie et al., 2015), supervised DL (Mairal et al., 2008), sparse singular value decomposition (Lee et al., 2010), non-negative sparse coding (Hoyer, 2004), principal component pursuit (Candès et al., 2011), robust non-negative sparse matrix factorization, and discriminative label consistent learning

(Jiang et al., 2011). More details on explicit customizations of the general model P can be found in Sec. 2.

Our distributed setting is motivated by several data-intensive applications in several fields, including signal processing and machine learning, and network systems (such as clouds, cluster computers, networks of sensor vehicles, or autonomous robots) wherein the sheer volume and spatial/temporal disparity of scattered data, energy constraints, and/or privacy issues, render centralized processing and storage infeasible or inefficient. Also, time-varying communications arise, for instance, in mobile wireless networks (e.g., ad-hoc networks), wherein nodes are mobile and/or communicate through fading channels. Moreover, since nodes generally transmit at different power and/or communication channels are not symmetric, directed links are a natural assumption.

Our goal is to design a provably convergent *decentralized* method for Problem P, over *time-varying* and *directed* graphs. To the best of our knowledge this is an open problem, as documented next.

1.1. Challenges and related works

The design of distributed algorithms for P faces the following challenges:

(i) Problem P is *non-convex* and *non-separable* in the optimization variables;

(ii) Each agent i owns exclusively $\mathbf{S}_i$ and thus can only compute its own function f_i;

(iii) Each f_i depends on a common set of variables−the dictionary $\mathbf{D}$−shared among all the agents, as well as the private variables $\mathbf{X}_i$. Shared and private variables need to be treated differently. In fact, in several applications, the size of private variables is much larger than that of the shared ones; hence, broadcasting agents' private variables over the network would result in an unaffordable communication overhead;

(iv) The gradient of each f_i is in general *neither bounded nor globally Lipschitz* on the feasible region. This represents a challenge in the design of provably convergent distributed algorithms, as boundedness and Lipschitzianity of the gradient are standard assumptions in the analysis of most distributed schemes for nonconvex problems;

(v) G and g_i's are *nonsmooth*;

(vi) The graph is *directed*, *time-varying*; no other structure is assumed (such as star or ring topology, etc.), but some long term connectivity properties (cf. Assumption B).

Centralized methods for the solution of Problem P (or some closely related variants) have been extensively studied and prominent examples are (Aharon et al., 2006; Mairal et al., 2010; Razaviyayn et al., 2014b). However, we are not aware of any *distributed* algorithm that can address challenges i)-vi) (even some subsets of them), as documented next.

Ad-hoc heuristics: Several attempts have been made to extend centralized approaches for DL problems to a distributed setting (*undirected, static* graphs), under more or less restrictive assumptions; examples include primal methods (Raja and Bajwa, 2013; Chainais and Richard, 2013; Wai et al., 2015) and (primal/)dual-based ones (Chen et al., 2015; Liang et al., 2014; Chouvardas et al., 2015). While these schemes represent good heuristics, their

theoretical convergence remains an open question, and numerical results are contradictory. For instance, some schemes are shown not to converge while some others fail to reach asymptotic agreement among the local copies of the dictionary; see, e.g. (Chainais and Richard, 2013).

Recently and independently from our conference work (Daneshmand et al., 2016), Zhao et al. (2016) proposed a distributed primal-dual-based method for a class of dictionary learning problems related, but different from Problem P. Specifically, they considered: quadratic loss functions f_i, with a quadratic regularization on the dictionary (i.e., $G = 0$), and norm ball constraints on the private variables. The network is modeled as a fixed undirected graph. Asymptotic convergence of the scheme to stationary solutions is proved, but no rate analysis is reported. We remark that the scheme in (Zhao et al., 2016), in order to establish convergence, requires some penalty parameters to go to infinity, which makes the method numerically not attractive.

Distributed nonconvex optimization: Since the DL problem P is an instance of nonconvex optimization problems, we briefly discuss here the few works in the literature on distributed methods for non-convex optimization (Bianchi and Jakubowicz, 2013; Tatarenko and Touri, 2017; Wai et al., 2017; Di Lorenzo and Scutari, 2016; Sun et al., 2016; Hong et al., 2017; Scutari and Sun, 2019); we group these papers as follows. The schemes in (Bianchi and Jakubowicz, 2013; Tatarenko and Touri, 2017; Wai et al., 2017; Hong et al., 2017), while substantially different, are all applicable to *smooth, unconstrained* optimization, with (Bianchi and Jakubowicz, 2013; Wai et al., 2017) handling also *compact* constraints and (Tatarenko and Touri, 2017) implementable on (time-varying) digraphs. The distributed algorithms in (Di Lorenzo and Scutari, 2016; Sun et al., 2016; Scutari and Sun, 2019) can handle objectives with additive *nonsmooth* convex functions, with (Sun et al., 2016; Scutari and Sun, 2019) applicable to (time-varying) digraphs.

All the above schemes *cannot* adequately deal with *private* (i.e., $\mathbf{X}_i$'s) *and* shared variables (i.e., $\mathbf{D}$), which are a key feature of Problem P. Furthermore, convergence therein is proved under the assumption that the gradient of (the smooth part of) the objective function is *globally Lipschitz continuous*, a property that we do not assume and that is not satisfied in many of the applications we consider. The design of provably convergent distributed algorithms for P remains an open problem, let alone rate guarantees.

1.2. Major contributions

In this paper, we propose the *first provably convergent* distributed algorithm for the general class of DL problems P, addressing all challenges i)-vi). The proposed approach uses a general convexification-decomposition technique that hinges on recent (centralized) Successive Convex Approximation methods (Scutari et al., 2014; Facchinei et al., 2015). This technique is coupled with a perturbed push-sum consensus scheme preserving the *feasibility* of the iterates and a tracking mechanism aiming at estimating locally the gradient of $\sum_i f_i$. Both communication and tracking protocols are implementable on *time-varying* undirected or *directed* graphs (B-strongly connected). The scheme is proved to converge to stationary solutions of Problem P, under mild assumptions on the step-size employed by the algorithm; a sublinear convergence rate is also established. On the technical side, we contribute to the literature of distributed algorithms for bi-convex (nonsmooth) constrained optimization by

putting forth a new non-trivial convergence analysis that, for the first time, i) avoids the assumption that the gradients ∇f_i are globally Lipschitz; and ii) deals with *private* and shared optimization variables. Numerical experiments show that the proposed schemes compare favorably with ad-hoc algorithms, proposed for special instances of Problem P.

1.3. Paper Organization

The rest of the paper is organized as follows. The problem and network setting are introduced in Sec. 2, along with some motivating applications. Sec. 3 presents the algorithm and its convergence properties; the proofs of our results are given in the Appendix, Sec. A. Extensive numerical experiments showing the effectiveness of the proposed scheme are discussed in Sec. 5 whereas Sec. 6 draws some conclusions.

1.4. Notation

Throughout the paper we use the following notation. We denote by $\mathbb{R}^n_+$ and $\mathbb{N}_+$ the non-negative orthant and the set of non-negative integers, respectively. Given $x \in \mathbb{R}$, $\lceil x \rceil$ (resp. $\lfloor x \rfloor$) denotes the smallest (resp. the largest) integer greater (resp. smaller) than or equal to x. Vectors are denoted by bold lower-case letters (e.g., $\mathbf{x}$) whereas matrices are denoted by bold capital letters (e.g., $\mathbf{A}$). The k-th canonical vector is denoted by $\mathbf{e}_k$. The inner product between two real matrices, $\mathbf{A}$ and $\mathbf{B}$, is denoted by $\langle \mathbf{A}, \mathbf{B} \rangle \triangleq \mathrm{tr}(\mathbf{A}^\intercal \mathbf{B})$, where $\mathrm{tr}(\bullet)$ is the trace operator; $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product. Given the real matrix $\mathbf{A}$, with ij-entries denoted by A_{ij}, we will use the following matrix norms: the Frobenius norm $||\mathbf{A}||_F \triangleq \sqrt{\sum_{i,j} |A_{ij}|^2}$; the $L_{1,1}$ norm $||\mathbf{A}||_{1,1} \triangleq \sum_{i,j} |A_{ij}|$; the $L_{2,\infty}$ norm $||\mathbf{A}||_{2,\infty} \triangleq \max_i \sqrt{\sum_j A_{ij}^2}$; the $L_{\infty,\infty}$ norm $||\mathbf{A}||_{\infty,\infty} = \max_{i,j} |A_{ij}|$; and the spectral norm $||\mathbf{A}||_2 \triangleq \sigma_{\max}(\mathbf{A})$, where $\sigma_{\max}(\mathbf{A})$ denotes the maximum singular value of $\mathbf{A}$. The matrix quantities $\nabla_D f_i(\mathbf{D}, \mathbf{X}_i)$ and $\nabla_{X_i} f_i(\mathbf{D}, \mathbf{X}_i)$ are the gradients of f_i with respect to $\mathbf{D}$ and $\mathbf{X}_i$, evaluated at $(\mathbf{D}, \mathbf{X}_i)$, respectively, with the partial derivatives arranged according to the patterns of $\mathbf{D}$ and $\mathbf{X}_i$, respectively. The same convention is adopted for subgradients of g_i and G, that are therefore written as matrices of the same dimensions of $\mathbf{X}_i$ and $\mathbf{D}$, respectively. Table 1 summarizes the main notation and symbols used in the paper.

Because of the nonconvexity of Problem P, we aim at computing stationary solutions of P, defined as follows: a tuple $(\mathbf{D}^*, \mathbf{X}^*)$, with $\mathbf{X}^* \triangleq [\mathbf{X}_1^*, \ldots, \mathbf{X}_I^*]$ is a stationarity solution of P if the following holds: $\mathbf{D}^* \in \mathcal{D}$, $\mathbf{X}_i^* \in \mathcal{X}_i$, $i = 1, \ldots, I$, and

$$\begin{aligned} &\left\langle \nabla_D F(\mathbf{D}^*, \mathbf{X}^*), \mathbf{D} - \mathbf{D}^* \right\rangle + G(\mathbf{D}) - G(\mathbf{D}^*) \geq 0, && \forall\, \mathbf{D} \in \mathcal{D}, \\ &\langle \nabla_{X_i} f_i(\mathbf{D}^*, \mathbf{X}_i^*), \mathbf{X}_i - \mathbf{X}_i^* \rangle + g_i(\mathbf{X}_i) - g_i(\mathbf{X}_i^*) \geq 0, && \forall \mathbf{X}_i \in \mathcal{X}_i,\ i = 1, \ldots, I. \end{aligned} \tag{2}$$

2. Problem Setup and Motivating Examples

In this section, we first discuss the assumptions underlying our model and then provide several examples of possible applications.

Symbol	Definition	Member of	Reference
$F(\mathbf{D}, \mathbf{X})$	$\sum_{i=1}^I f_i(\mathbf{D}, \mathbf{X}_i)$	$\mathbb{R}^{M\times K} \times \mathbb{R}^{K\times N} \to \mathbb{R}$	(P)
$U(\mathbf{D}, \mathbf{X})$	$F(\mathbf{D}, \mathbf{X}) + \sum_{i=1}^I g_i(\mathbf{X}_i) + G(\mathbf{D})$	$\mathbb{R}^{M\times K} \times \mathbb{R}^{K\times N} \to \mathbb{R}$	(P)
$\mathbf{S}_i$	Local data matrix	$\mathbb{R}^{M\times n_i}$	
$\mathbf{S}$	$[\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_I]$	$\mathbb{R}^{M\times N}$	
$\mathbf{D}$	Dictionary matrix variable	$\mathcal{D} \subseteq \mathbb{R}^{M\times K}$	
$\mathbf{D}_{(i)}$	Local copy of $\mathbf{D}$ of agent i	$\mathcal{D} \subseteq \mathbb{R}^{M\times K}$	
$\mathbf{D}_{(i)}^\nu$	$\mathbf{D}_{(i)}$ at iteration ν	$\mathcal{D} \subseteq \mathbb{R}^{M\times K}$	
$\mathbf{D}^\nu$	$[\mathbf{D}_{(1)}^{\nu\intercal}, \mathbf{D}_{(2)}^{\nu\intercal}, \ldots, \mathbf{D}_{(I)}^{\nu\intercal}]^\intercal$	$\mathbb{R}^{M\times KI}$	(25)
$\widetilde{\mathbf{D}}_{(i)}^\nu$	Solution of subproblem (8)	$\mathcal{D} \subseteq \mathbb{R}^{M\times K}$	(8)
$\overline{\mathbf{D}}^\nu$	$(1/I)\sum_{i=1}^I \mathbf{D}_{(i)}^\nu$	$\mathcal{D} \subseteq \mathbb{R}^{M\times K}$	(25)
$\mathbf{U}_{(i)}^\nu$	Local update of dictionary variable	$\mathcal{D} \subseteq \mathbb{R}^{M\times K}$	(10)
$\mathbf{X}_i$	Local matrix variable	$\mathcal{X}_i \subseteq \mathbb{R}^{K\times n_i}$	
$\mathbf{X}$	$[\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_I]$	$\mathcal{X} \subseteq \mathbb{R}^{K\times N}$	
$\mathbf{X}_i^\nu$	$\mathbf{X}_i$ at iteration ν	$\mathcal{X}_i \subseteq \mathbb{R}^{K\times n_i}$	
$\mathbf{X}^\nu$	$\mathbf{X}$ at iteration ν : $[\mathbf{X}_1^\nu, \mathbf{X}_2^\nu, \ldots, \mathbf{X}_I^\nu]$	$\mathcal{X} \subseteq \mathbb{R}^{K\times N}$	(25)
$\mathbf{\Theta}_{(i)}^\nu$	Gradient-tracking variable	$\mathcal{D} \subseteq \mathbb{R}^{M\times K}$	(13)
$\mathbf{A}^\nu$	$(a_{ij}^\nu)_{i,j=1}^I -$ consensus weights at time ν	$\mathbb{R}^{I\times I}$	Assumption F

Table 1: Table of notation

2.1. Problem Assumptions

We consider Problem P under the following assumptions.

Assumption A (On Problem P)

(A1) *Each $f_i : \mathcal{O} \times \mathcal{O}_i \to \mathbb{R}$ is $\mathcal{C}^2$, lower bounded, and biconvex, where $\mathcal{O} \supseteq \mathcal{D}$ and $\mathcal{O}_i \supseteq \mathcal{X}_i$ are convex open sets;*

(A2) *Given $\mathbf{D} \in \mathcal{D}$, each $\nabla_{X_i} f_i(\mathbf{D}, \bullet)$ is Lipschitz continuous on $\mathcal{X}_i$, with Lipschitz constant $L_{\nabla X_i}(\mathbf{D})$. Furthermore, each $L_{\nabla X_i} : \mathcal{D} \to \mathbb{R}_+$ is continuous;*

(A3) *$\mathcal{D}$ is compact and convex; and each $\mathcal{X}_i$ is closed and convex (not necessarily bounded);*

(A4) *$G : \mathcal{O} \to \mathbb{R}$ is convex (possibly non-smooth);*

(A5) *For all $i = 1, \ldots, I$, either i) $\mathcal{X}_i$ is compact and $g_i : \mathcal{O}_i \to \mathbb{R}$ is convex; or ii) g_i is μ_i-strongly convex.*

The above assumptions are quite mild and are satisfied by several problems of practical interest; see Sec. 2.2 for several concrete examples.

Network topology

We study Problem P in the following network setting. Time is slotted and in each time-slot ν the network of the I agents is modeled as a digraph $\mathcal{G}^\nu = (\mathcal{V}, \mathcal{E}^\nu)$, where $\mathcal{V} = \{1, \ldots, I\}$ is the set of agents and $\mathcal{E}^\nu$ is the set of edges (communication links); we use $(i, j) \in \mathcal{E}^\nu$ to indicate that there is a directed link from node i to node j. The *in-neighborhood* of agent

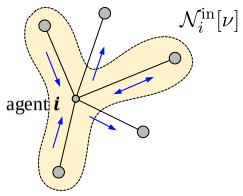


Figure 2: Illustration of in-neighborhood set of agent i at time ν.

$i \in \mathcal{V}$ at time ν is defined as $\mathcal{N}_i^{\text{in}}[\nu] = \{j \in \mathcal{V} | (j,i) \in \mathcal{E}^\nu\} \cup \{i\}$ (see Fig. 2) whereas its *out-neighborhood* is $\mathcal{N}_i^{\text{out}}[\nu] = \{j \in \mathcal{V} | (i,j) \in \mathcal{E}^\nu\} \cup \{i\}$. In words, agent i can receive information from its in-neighborhood members, and send information to its out-neighbors. The *out-degree* of agent i is defined as $d_i^\nu \triangleq \left|\mathcal{N}_i^{\text{out}}[\nu]\right|$, where $|\bullet|$ denotes the cardinality of a set. If the graph is undirected, the set of in-neighbors and out-neighbors coincide; in such a case we just write $\mathcal{N}_i$ to denote the set of neighbors of agent i. When the network is static, all the above quantities do not depend on the iteration index ν; hence, we will drop the superscript "ν". To let information propagate over the network, we assume that the sequence $\{\mathcal{G}^\nu\}_\nu$ possesses some "long-term" connectivity property, as stated next.

Assumption B (B-strong connectivity) *The graph sequence $\{\mathcal{G}^\nu\}_\nu$ is B-strongly connected, i.e., there exists an (arbitrarily large) integer $B > 0$ (unknown to the agents) such that the graph with edge set $\cup_{t=kB}^{(k+1)B-1} \mathcal{E}^t$ is strongly connected, for all $k \geq 0$.*

Notice that this condition is quite mild and widely used in the literature to analyze convergence of distributed algorithms over time-varying networks. Generally speaking, it permits strong connectivity to occur over time windows of length B, so that information can propagate from every node to every other node in the network. Assumption B is satisfied in several practical scenarios. For instance, commonly used settings in cloud computing infrastructures are star, ring, tree, hypercube, or n-dimensional mesh (Torus) topologies, which all satisfy Assumption B. It is worth mentioning that the multi-hop network topologies of these structures are migrating towards *high-radix mesh and Torus*, since they are scalable, low-energy consuming, and much cheaper than other topologies, like *fat-tree* topologies (Kim, 2008). These type of connected networks are generally time-invariant and undirected, and clearly they satisfy Assumption B.

2.2. Motivating examples

We conclude this section discussing some practical instances of Problem P, all satisfying Assumption A, which show the generality of the proposed model.

Elastic net sparse DL (Tosic and Frossard, 2011; Zou and Hastie, 2005)

Sparse approximation of a signal with an adaptive dictionary is one of the most studied DL problems (Tosic and Frossard, 2011). When an elastic net sparsity-inducing regularizer is

used (Zou and Hastie, 2005), the problem can be written as

$$\begin{aligned} \min_{\mathbf{D},\{\mathbf{X}_i\}_{i=1}^I} \quad & \sum_{i=1}^I \left\{ \frac{1}{2} \left\|\mathbf{S}_i - \mathbf{D}\mathbf{X}_i\right\|_F^2 + \lambda \left\|\mathbf{X}_i\right\|_{1,1} + \frac{\mu}{2} \left\|\mathbf{X}_i\right\|_F^2 \right\} \\ \text{s.t.} \quad & \mathbf{D} \in \mathcal{D},\ \mathbf{X}_i \in \mathbb{R}^{K \times n_i}, \quad i = 1, 2, \ldots, I, \end{aligned} \tag{3}$$

where $\mathcal{D} \triangleq \{\mathbf{D} : ||\mathbf{D}\mathbf{e}_k||_2 \leq \alpha,\ k = 1, 2, \ldots, K\}$, and α, λ, $\mu > 0$ are the tuning parameters. Problem (3) is an instance of P, with $f_i(\mathbf{D}, \mathbf{X}_i) = (1/2) \cdot \|\mathbf{S}_i - \mathbf{D}\mathbf{X}_i\|_F^2$, $g_i(\mathbf{X}_i) = \lambda \|\mathbf{X}_i\|_{1,1} + \frac{\mu}{2} \|\mathbf{X}_i\|_F^2$, $G(\mathbf{D}) = 0$, and $\mathcal{X}_i = \mathbb{R}^{K \times n_i}$. It is not difficult to check that (3) satisfies Assumption A, and the Lipschitz constant in A2 is given by $L_{\nabla X_i}(\mathbf{D}) = (\sigma_{\max}(\mathbf{D}))^2$.

Supervised DL (Mairal et al., 2008)

Consider a classification problem with training set $\{\mathbf{s}_n, y_n\}_{n=1}^N$, where $\mathbf{s}_n$ is the feature vector with associated binary label y_n. The discriminative DL problem aims at simultaneously learning a dictionary $\mathbf{D}^{(1)} \in \mathbb{R}^{M \times K}$ such that $\mathbf{s}_n = \mathbf{D}^{(1)}\mathbf{x}_n$, for some sparse $\mathbf{x}_n \in \mathbb{R}^K$, and finding a bilinear classifier $\zeta_n(\mathbf{D}^{(2)}, \mathbf{x}_n, \mathbf{s}_n) \triangleq \mathbf{s}_n^\intercal \mathbf{D}^{(2)} \mathbf{x}_n$ that best separates the coded data with distinct labels (Mairal et al., 2008). Assume that each agent i owns $\{(\mathbf{s}_n, y_n) : n \in \mathcal{S}_i\}$, with $\{\mathcal{S}_i\}_{i=1}^I$ being a partition of $\{1, \ldots, N\}$, then the discriminative DL reads

$$\begin{aligned} \min_{\substack{\mathbf{D}^{(1)},\mathbf{D}^{(2)} \\ \{\mathbf{x}_n\}_{n=1}^N}} \quad & \sum_{i=1}^I \sum_{n \in \mathcal{S}_i} \left[\ell\left(y_n \zeta_n\left(\mathbf{D}^{(2)}, \mathbf{x}_n, \mathbf{s}_n\right)\right) + \frac{1}{2} \left\|\mathbf{s}_n - \mathbf{D}^{(1)}\mathbf{x}_n\right\|_2^2 + g_n(\mathbf{x}_n) \right] \\ \text{s.t.} \quad & \mathbf{D}^{(1)} \in \mathcal{D}^{(1)},\ \mathbf{D}^{(2)} \in \mathcal{D}^{(2)},\ \mathbf{x}_n \in \mathbb{R}^K, \quad n = 1, 2, \ldots, N, \end{aligned} \tag{4}$$

where $\ell(x) \triangleq \log(1 + e^{-x})$ is the *logistic* loss function; and $g_n(\mathbf{x}_n) \triangleq \lambda \|\mathbf{x}_n\|_1 + (\mu/2) \cdot \|\mathbf{x}_n\|_2^2$ is the elastic net regularizer. The dictionary $\mathbf{D}^{(1)}$ and classifier parameter $\mathbf{D}^{(2)}$ are constrained to belong to the convex compact sets $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$, respectively. Problem (4) is an instance of Problem P, with $\mathbf{D} \triangleq \left[\mathbf{D}^{(1)}, \mathbf{D}^{(2)}\right]$, $\mathbf{S}_i \triangleq [\mathbf{s}_n]_{n \in \mathcal{S}_i}$ and $\mathbf{X}_i \triangleq [\mathbf{x}_n]_{n \in \mathcal{S}_i}$. Note that Assumption A is satisfied, and the Lipschitz constant in A2 is given by $L_{\nabla X_i}(\mathbf{D}) = (1/4) \cdot \|y_i \cdot \mathbf{s}_i^\intercal \mathbf{D}^{(2)}\|_2^2 + (\sigma_{\max}(\mathbf{D}^{(1)}))^2$.

DL for low-rank plus sparse representation (Bouwmans et al., 2017)

The low-rank plus sparse decomposition problems cover many applications in signal processing and machine learning (Bouwmans et al., 2017), including matrix completion, image denoising, deblurring, superresolution, and Principal Component Pursuit (PCP) (Candès et al., 2011). Consider the bi-linear model $\mathbf{S} \approx \mathbf{L} + \mathbf{H}\mathbf{Q}\mathbf{U}$: the data matrix $\mathbf{S}$ is decomposed as the superposition of a low-rank matrix $\mathbf{L}$ (capturing the correlations among data) and $\mathbf{H}\mathbf{Q}\mathbf{U}$, where $\mathbf{Q} \in \mathbb{R}^{M \times \tilde{K}}$ is an over-complete dictionary (capturing the representative modes of the data), $\mathbf{U} \in \mathbb{R}^{\tilde{K} \times N}$ is a sparse matrix (representing the data parsimoniously), and $\mathbf{H}$ is a given *degradation* matrix, which accounts for tasks such as denoising, superresolution, and deblurring. To enforce $\mathbf{L}$ to be low-rank, we employ the nuclear norm $\|\mathbf{L}\|_*$ regularizer, which can be equivalently rewritten as $\|\mathbf{L}\|_* = \inf\{\frac{1}{2}\|\mathbf{P}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 : \mathbf{L} = \mathbf{P}\mathbf{V}\}$, where $\mathbf{P} \in \mathbb{R}^{M \times L}$, $\mathbf{V} \in \mathbb{R}^{L \times N}$, and $L \ll \min(M, N)$ (Srebro and Shraibman, 2005; Recht

et al., 2010). Partitioning $\mathbf{V}$ and $\mathbf{U}$ according to $\mathbf{S}$, i.e., $\mathbf{S}_i = \mathbf{P}\mathbf{V}_i + \mathbf{H}\mathbf{Q}\mathbf{U}_i$, the problem reads

$$\begin{aligned} \min_{\mathbf{P},\mathbf{Q},(\mathbf{V}_i,\mathbf{U}_i)_{i=1}^I} \quad & \sum_{i=1}^I \left[\frac{1}{2} \left\| \mathbf{S}_i - [\mathbf{P}\ \mathbf{H}\mathbf{Q}] \begin{bmatrix} \mathbf{V}_i \\ \mathbf{U}_i \end{bmatrix} \right\|_F^2 \right. \\ & \left. + \frac{\zeta}{2I} \left(\|\mathbf{P}\|_F^2 + I \cdot \|\mathbf{V}_i\|_F^2 \right) + \lambda \|\mathbf{X}_i\|_{1,1} + \frac{\mu}{2} \|\mathbf{X}_i\|_F^2 \right] \\ \text{s.t.} \quad & \mathbf{D} \in \mathcal{D}, \quad \mathbf{X}_i \in \mathbb{R}^{(L+\tilde{K}) \times n_i}, \quad i = 1, 2, \ldots, I, \end{aligned} \tag{5}$$

where $\mathcal{D}$ is some compact set; $\zeta > 0$ is a constant used to promote the low-rank structure on $\mathbf{L}$ while sparsity on $\mathbf{X}$ is enforced by the elastic net regularization, with constants λ, $\mu > 0$. Problem (5) is clearly an instance of Problem P wherein f_i is the quadratic loss, and $[\mathbf{P}, \mathbf{Q}]$ and $[\mathbf{V}_i^\intercal, \mathbf{U}_i^\intercal]^\intercal$ are the shared and private variables ($K = L + \tilde{K}$), respectively. Assumption A is satisfied, and the Lipschitz constant in A2 is given by $L_{\nabla X_i}(\mathbf{D}) = (\sigma_{\max}(\mathbf{P}\mathbf{H}\mathbf{Q}))^2$.

A variant of this problem, which still is a particular case of Problem (5), is obtained by replacing the quadratic loss function with the smoothed Huber function to achieve robustness against outliers (Aravkin et al., 2014).

Sparse SVD/PCA (Lee et al., 2010; Udell et al., 2016; Mairal et al., 2010)

Computing the SVD of a set of data with sparse singular vectors (Sparse SVD) is the foundation of many applications in multivariate analysis, e.g., biclustering (Lee et al., 2010). As proposed in (Mairal et al., 2010), Problem P can be used to accomplish this task by imposing sparsity on the factors $\mathbf{D}$ and $\mathbf{X}$ of $\mathbf{S}$. More specifically, we have

$$\begin{aligned} \min_{\mathbf{D},(\mathbf{X}_i)_{i=1}^I} \quad & \sum_{i=1}^I \left\{ \frac{1}{2} \|\mathbf{S}_i - \mathbf{D}\mathbf{X}_i\|_F^2 + \lambda_X \|\mathbf{X}_i\|_{1,1} + \frac{\mu_X}{2} \|\mathbf{X}_i\|_F^2 \right\} + \lambda_D \|\mathbf{D}\|_{1,1} + \frac{\mu_D}{2} \|\mathbf{D}\|_F^2 \\ \text{s.t.} \quad & \mathbf{D} \in \mathcal{D} \triangleq \{\mathbf{D} \in \mathbb{R}^{M \times K} : ||\mathbf{D}||_{2,\infty} \le \alpha\}, \quad \mathbf{X}_i \in \mathbb{R}^{K \times n_i}, \quad i = 1, 2, \ldots, I, \end{aligned} \tag{6}$$

where λ_D, λ_X, μ_D, μ_X, $\alpha > 0$ are given constants. Problem (6) is an instance of P, with $f_i(\mathbf{D}, \mathbf{X}_i) = (1/2) \cdot ||\mathbf{S}_i - \mathbf{D}\mathbf{X}_i||_F^2$; $G(\mathbf{D}) = \lambda_D ||\mathbf{D}||_{1,1} + (\mu_D/2) \cdot ||\mathbf{D}||_F^2$, and $g_i(\mathbf{X}_i) = \lambda_X ||\mathbf{X}_i||_{1,1} + (\mu_X/2) \cdot ||\mathbf{X}_i||_F^2$. Note that orthonormality of factors are relaxed for sake of simplicity. A related formulation, termed Sparse PCA, has also been used in (Udell et al., 2016). It is not difficult to show that Assumption A is satisfied, and the Lipschitz constant in A2 is given by $L_{\nabla X_i}(\mathbf{D}) = (\sigma_{\max}(\mathbf{D}))^2$.

Non-negative Sparse Coding (NNSC) (Hoyer, 2004)

Non-negative Matrix Factorization (NMF) was primarily proposed by (Lee and Seung, 1999) as a better alternative to the classic SVD in learning localized features of image datasets, such as face images. The formulation enforces non-negativity of the entries of $\mathbf{D}$ and $\mathbf{X}$. This has been shown to empirically lead to sparse solutions; however no explicit control on sparsity is employed in the model. To overcome this shortcoming, (Hoyer, 2004) proposed a non-negative sparse coding (NNSC) formulation which extends NMF by adding a sparsity-

inducing penalty function of $\mathbf{X}$. The problem reads

$$\begin{aligned} \min_{\mathbf{D},(\mathbf{X}_i)_{i=1}^I} \quad & \sum_{i=1}^I \left\{ \frac{1}{2} \|\mathbf{S}_i - \mathbf{D}\mathbf{X}_i\|_F^2 + \lambda \|\mathbf{X}_i\|_{1,1} + \frac{\mu}{2} \|\mathbf{X}_i\|_F^2 \right\} \\ \text{s.t.} \quad & \mathbf{D} \in \mathcal{D} \triangleq \{\mathbf{D} \in \mathbb{R}_+^{M \times K} \mid ||\mathbf{D}||_{2,\infty} \le \alpha\}, \ \mathbf{X}_i \in \mathbb{R}_+^{K \times n_i}, \quad i = 1, 2, \ldots, I, \end{aligned} \tag{7}$$

for some λ, μ, $\alpha > 0$. Problem (7) is another instance of P, with $f_i(\mathbf{D}, \mathbf{X}_i) = (1/2) \cdot ||\mathbf{S}_i - \mathbf{D}\mathbf{X}_i||_F^2$, $g_i(\mathbf{X}_i) = \lambda \|\mathbf{X}_i\|_{1,1} + (\mu/2) \cdot \|\mathbf{X}_i\|_F^2$, $G(\mathbf{D}) = 0$, $\mathcal{D} = \{\mathbf{D} \in \mathbb{R}_+^{M \times K} \mid ||\mathbf{D}||_{2,\infty} \le \alpha\}$, and $\mathcal{X}_i = \mathbb{R}_+^{K \times n_i}$. Assumption A is satisfied, and the Lipschitz constant in A2 is given by $L_{\nabla X_i}(\mathbf{D}) = (\sigma_{\max}(\mathbf{D}))^2$.

3. Algorithmic Design

We introduce now our algorithmic framework. To shed light on the core idea behind the proposed scheme, we begin introducing an informal and constructive description of the algorithm, followed by its formal description along with its convergence properties.

Each agent i controls its private variable $\mathbf{X}_i$ and maintains a local copy of the shared variables $\mathbf{D}$, denoted by $\mathbf{D}_{(i)}$, along with an auxiliary variable $\mathbf{\Theta}_{(i)}$; we anticipate that $\mathbf{\Theta}_{(i)}$ aims at *locally* estimating the gradient sum $\sum_j \nabla_D f_j(\mathbf{D}_{(i)}, \mathbf{X}_j)$, an information that is not available at agent i's side. The value of these variables at iteration ν is denoted by $\mathbf{X}_i^\nu$, $\mathbf{D}_{(i)}^\nu$, and $\mathbf{\Theta}_{(i)}^\nu$, respectively. Roughly speaking, the update of these variables is designed so that asymptotically i) all the $\mathbf{D}_{(i)}$ will be consensual, i.e., $\mathbf{D}_{(i)} = \mathbf{D}_{(j)}$, $\forall i \neq j$; and ii) the tuples $(\mathbf{D}_{(i)}, (\mathbf{X}_j)_{j=1}^I)$ will be a stationary solutions of Problem P. This is accomplished throughout the following two steps, which are performed iteratively and in parallel across the agents.

Step 1: Local Optimization

The nonconvexity of f_i together with the lack of knowledge of $\sum_{j \neq i} f_j$ in F prevents agent i to solve directly Problem P with respect to $(\mathbf{D}_{(i)}, \mathbf{X}_i)$. Since f_i is *bi-convex* in $(\mathbf{D}_{(i)}, \mathbf{X}_i)$, a natural approach is then to update $\mathbf{D}_{(i)}$ and $\mathbf{X}_i$ in an *alternating* fashion by solving a *local* approximation of P. Specifically, at iteration ν, given the iterates $\mathbf{X}_i^\nu$, $\mathbf{D}_{(i)}^\nu$, and $\mathbf{\Theta}_{(i)}^\nu$, agent i fixes $\mathbf{X}_i = \mathbf{X}_i^\nu$ and solves the following strongly convex problem in $\mathbf{D}_{(i)}$:

$$\tilde{\mathbf{D}}_{(i)}^\nu \triangleq \operatorname*{argmin}_{\mathbf{D}_{(i)} \in \mathcal{D}} \tilde{f}_i(\mathbf{D}_{(i)}; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) + \left\langle I \cdot \mathbf{\Theta}_{(i)}^\nu - \nabla_D f_i(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu), \mathbf{D}_{(i)} - \mathbf{D}_{(i)}^\nu \right\rangle + G\left(\mathbf{D}_{(i)}\right), \tag{8}$$

where $\tilde{f}_i(\bullet; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)$ is a suitably chosen strongly convex approximation of $f_i(\bullet, \mathbf{X}_i^\nu)$ at $(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)$ (cf. Assumption C, Sec. 3.1); and $\mathbf{\Theta}_{(i)}^\nu$, as anticipated, is used to track the gradient of F, with $\lim_{\nu \to \infty} \|I \cdot \mathbf{\Theta}_{(i)}^\nu - \sum_{j=1}^I \nabla_D f_j(\mathbf{D}_{(i)}^\nu, \mathbf{X}_j^\nu)\| = 0$; which would lead to

$$\lim_{\nu \to \infty} \left\| \left(I \cdot \mathbf{\Theta}_{(i)}^\nu - \nabla_D f_i(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) \right) - \sum_{j \neq i} \nabla_D f_j(\mathbf{D}_{(i)}^\nu, \mathbf{X}_j^\nu) \right\| = 0. \tag{9}$$

This sheds light on the role of the linear term in (8): it can be regarded as a proxy of the sum-gradient $\sum_{j \neq i} \nabla_D f_j(\mathbf{D}_{(i)}^\nu, \mathbf{X}_j^\nu)$, which is not available at agent i's side. In Step 2 below we show how to update $\mathbf{\Theta}_{(i)}^\nu$ using only local information, so that (9) holds.

Given $\widetilde{\mathbf{D}}_{(i)}^{\nu}$, a step-size is employed in the update of $\mathbf{D}_{(i)}$, generating the iterate $\mathbf{U}_{(i)}^{\nu}$:

$$\mathbf{U}_{(i)}^{\nu} = \mathbf{D}_{(i)}^{\nu} + \gamma^{\nu}(\widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu}), \tag{10}$$

where γ^{ν} is the step-size, to be properly chosen (see Assumption E, Sec. 3.1).

Let us now consider the update of the private variables $\mathbf{X}_i$. Fixing $\mathbf{D}_{(i)} = \mathbf{U}_{(i)}^{\nu}$, agent i computes the new update $\mathbf{X}_i^{\nu+1}$ by solving the following strongly convex optimization problem:

$$\mathbf{X}_i^{\nu+1} \triangleq \underset{\mathbf{X}_i \in \mathcal{X}_i}{\operatorname{argmin}} \quad \tilde{h}_i(\mathbf{X}_i; \mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) + g_i(\mathbf{X}_i), \tag{11}$$

where $\tilde{h}_i(\bullet; \mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$ is a strongly convex function of $\mathbf{X}_i$, approximating $f_i(\mathbf{U}_{(i)}^{\nu}, \bullet)$ at $(\mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$; see Assumption C (cf. Sec. 3.1) for specific instances of $\tilde{h}_i$.

Step 2: Local Communications

Let us design now a local communication mechanism ensuring asymptotic consensus over the local copies $\mathbf{D}_{(i)}$'s and property (9). To do so, we build on the (perturbed) push-sum protocol proposed in (Sun et al., 2016) (see also Kempe et al. (2003)). Specifically, an extra scalar variable ϕ_i is introduced at each agent's side to deal with the directed nature of the graph; given ϕ_i^{ν} and $\mathbf{U}_{(j)}^{\nu}$ from its in-neighbors $j \in \mathcal{N}_i$, each agent i updates its own local estimate $\mathbf{D}_{(i)}^{\nu}$ and ϕ_i^{ν} according to:

$$\phi_i^{\nu+1} = \sum_{j \in \mathcal{N}_i^{\text{in}}[\nu]} a_{ij}^{\nu}\, \phi_j^{\nu} \quad \text{and} \quad \mathbf{D}_{(i)}^{\nu+1} = \frac{1}{\phi_i^{\nu+1}} \sum_{j \in \mathcal{N}_i^{\text{in}}[\nu]} a_{ij}^{\nu}\, \phi_j^{\nu} \mathbf{U}_{(j)}^{\nu}, \tag{12}$$

where a_{ij}^{ν}'s are some weights (to be properly chosen, see Assumption F, Sec. 3.1); and $\phi_i^0 = 1$, for all $i = 1, \ldots, I$.

Note that the updates in (12) can be implemented locally: all agents only need to (i) send their local variable $\mathbf{U}_{(j)}^{\nu}$ and the scalar weight $a_{ij}^{\nu}\, \phi_j^{\nu}$ to their neighbors; and (ii) collect locally the information coming from the neighbors.

To update the $\mathbf{\Theta}_{(i)}^{\nu}$ variables we leverage the gradient tracking mechanism first introduced in (Di Lorenzo and Scutari, 2016), coupled with the push-sum consensus scheme (Sun et al., 2016), resulting in the following perturbed push-sum scheme:

$$\mathbf{\Theta}_{(i)}^{\nu+1} = \frac{1}{\phi_i^{\nu+1}} \sum_{j \in \mathcal{N}_i^{\text{in}}[\nu]} a_{ij}^{\nu} \phi_j^{\nu} \mathbf{\Theta}_{(j)}^{\nu} + \frac{1}{\phi_i^{\nu+1}} \left(\nabla_D f_i(\mathbf{D}_{(i)}^{\nu+1}, \mathbf{X}_i^{\nu+1}) - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) \right), \tag{13}$$

with $\mathbf{\Theta}_{(i)}^0 \triangleq \nabla_D f_i(\mathbf{D}_{(i)}^0, \mathbf{X}_i^0)$, for all $i = 1, \ldots, I$. The update (13) follows similar logic as that of $\mathbf{D}_{(i)}^{\nu}$ in (12), with the difference that (13) contains a perturbation [the second term in the RHS of (13)], which employs $\mathbf{\Theta}_{(i)}^{\nu}$ and ensures the desired tracking properties (otherwise $\mathbf{\Theta}_{(i)}^{\nu}$ would converge to the average of their initial values). Note that (13) can be performed locally by agent i, following the same procedure as described for (12).

Combining the above steps, we can now formally introduce the proposed distributed algorithm for the DL problems P, as described in Algorithm 1, and termed D^4L (Decentralized Dictionary Learning over Dynamic Digraphs) *Algorithm*.

Algorithm 1 : Decentralized Dictionary Learning over Dynamic Digraphs (D^4L)

Initialization : set $\nu = 0$ and $\phi_i^0 = 1$, $\mathbf{D}_{(i)}^0 \in \mathcal{D}$, $\mathbf{X}_i^0 \in \mathcal{X}_i$, $\mathbf{\Theta}_{(i)}^0 = \nabla_D f_i(\mathbf{D}_{(i)}^0, \mathbf{X}_i^0)$, for all $i = 1, 2, \ldots, I$.

`S1.` If $(\mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$ satisfies a suitable stopping criterion: `STOP`;

`S2.` **Local Optimization:** Each agent i computes:

(a) $\widetilde{\mathbf{D}}_{(i)}^{\nu}$ and $\mathbf{U}_{(i)}^{\nu}$ according to (8) and (10);

(b) $\mathbf{X}_i^{\nu+1}$ according to (11);

`S3.` **Local Communications:** Each agent i collects data from its current neighbors and updates:

(a) $\phi_i^{\nu+1}$ and $\mathbf{D}_{(i)}^{\nu+1}$ according to (12);

(b) $\mathbf{\Theta}_{(i)}^{\nu+1}$ according to (13);

`S4.` Set $\nu + 1 \to \nu$, and go to `S1`.

3.1. Algorithmic Assumptions

Before stating the main convergence result for the D^4L Algorithm, we discuss the main assumptions governing the choices of the free parameters of the algorithm, namely: the surrogate functions $\tilde{f}_i$ and $\tilde{h}_i$, the step-size γ^{ν}, and the consensus weights $(a_{ij}^{\nu})_{i,j=1}^{I}$.

3.1.1. On the choice of $\tilde{f}_i$ and $\tilde{h}_i$.

The surrogate functions are chosen to satisfy the following assumption.

Assumption C (On $\tilde{f}_i$ and $\tilde{h}_i$) *Given $\mathbf{D}_{(i)}^{\nu}$ and $\mathbf{X}_i^{\nu}$, $\tilde{f}_i(\bullet; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$ in (8) is either*

$$\tilde{f}_i(\mathbf{D}_{(i)}; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) = f_i(\mathbf{D}_{(i)}, \mathbf{X}_i^{\nu}) + \frac{\tau_{D,i}^{\nu}}{2} \|\mathbf{D}_{(i)} - \mathbf{D}_{(i)}^{\nu}\|_F^2, \tag{14}$$

or

$$\tilde{f}_i(\mathbf{D}_{(i)}; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) = \left\langle \nabla_D f_i(\mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}), \mathbf{D}_{(i)} - \mathbf{D}_{(i)}^{\nu} \right\rangle + \frac{\tau_{D,i}^{\nu}}{2} \left\|\mathbf{D}_{(i)} - \mathbf{D}_{(i)}^{\nu}\right\|_F^2, \tag{15}$$

where $\tau_{D,i}^{\nu}$ is a positive scalar satisfying Assumption D.

Given $\mathbf{U}_{(i)}^{\nu}$ and $\mathbf{X}_i^{\nu}$, $\tilde{h}_i(\bullet; \mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$ in (11) is either

$$\tilde{h}_i(\mathbf{X}_i; \mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) \triangleq f_i(\mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i) + \frac{\tau_{X,i}^{\nu}}{2} \|\mathbf{X}_i - \mathbf{X}_i^{\nu}\|_F^2, \tag{16}$$

or

$$\tilde{h}_i(\mathbf{X}_i; \mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) = \left\langle \nabla_{X_i} f_i(\mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}), \mathbf{X}_i - \mathbf{X}_i^{\nu} \right\rangle + \frac{\tau_{X,i}^{\nu}}{2} \left\|\mathbf{X}_i - \mathbf{X}_i^{\nu}\right\|_F^2, \tag{17}$$

where $\tau_{X,i}^{\nu}$ is a positive scalar satisfying Assumption D.

Assumption D (On $\tau^\nu_{X,i}$ and $\tau^\nu_{D,i}$) *The parameters $(\tau^\nu_{X,i})_{i=1}^I$ and $(\tau^\nu_{D,i})_{i=1}^I$ are chosen such that*

(D1) *$\{\tau^\nu_{D,i}\}_\nu$ and $\{\tau^\nu_{X,i}\}_\nu$ satisfy*

$$0 < \inf_\nu \tau^\nu_{D,i} \le \sup_\nu \tau^\nu_{D,i} < +\infty, \tag{18}$$

and

$$\sup_\nu \tau^\nu_{X,i} < +\infty,$$
$$\tau^\nu_{X,i} \ge \tfrac{1}{2} L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}) + \epsilon, \quad \forall \nu \ge 1, \tag{19}$$

for all $i = 1, 2, \ldots, I$, where $\epsilon > 0$ is an arbitrarily small constant, and $L_{\nabla X_i}$ is defined in Assumption A2.

(D2) *Stronger convergence results [cf. Theorem 2] can be obtained if, under Assumption A5(ii), the sequences $\{\tau^\nu_{D,i}\}_\nu$ and $\{\tau^\nu_{X,i}\}_\nu$, in addition to D1, also satisfy*

$$\sum_{t=0}^{\infty} \left| \tau^{t+1}_{D,i} - \tau^t_{D,i} \right| < \infty, \tag{20}$$

and

$$\limsup_\nu \left| \tau^\nu_{X,i} - \tau^{\nu-1}_{X,i} \right| < \mu, \tag{21}$$

where $\mu \triangleq \min_i \mu_i$ and μ_i is the strongly convexity constant of f_i.

Discussion. Several comments are in order.

• *On the choice of $\tilde{f}_i$ and $\tilde{h}_i$.* Since f_i (resp. h_i) is convex in $\mathbf{D}_{(i)}$ (resp. $\mathbf{X}_i$), (14) [resp. (16)] is a natural choice for the surrogate $\tilde{f}_i$ (resp. $\tilde{h}_i$): the structure of f_i (resp. h_i) is preserved while a quadratic term is added to make the overall surrogate strongly convex. The non-smooth strongly convex subproblems (8) and (11) resulting from (14) and (16) can be solved using standard solvers, e.g., projected subgradient methods. When dealing with large-scale instances, effective methods are also (Facchinei et al., 2015; Daneshmand et al., 2015).

The alternative surrogates $\tilde{f}_i$ and $\tilde{h}_i$ as given in (15) and (17), respectively, are based on the the linearization of the original f_i and h_i. This option is motivated by the fact that, for specific instances of f_i and h_i, (15) and (17) lead to subproblems (8) and (11) whose solution can be computed in closed form. For instance, consider the *elastic net sparse DL* problem (3) in Sec. 2.2, where $f_i(\mathbf{D}, \mathbf{X}_i) = \frac{1}{2}||\mathbf{S}_i - \mathbf{D}\mathbf{X}_i||_F^2$; $G(\mathbf{D}) = 0$; and $g_i(\mathbf{X}_i) = \lambda\,||\mathbf{X}_i||_1 + \frac{\mu}{2}\,||\mathbf{X}_i||_F^2$, with $\lambda, \mu > 0$. By using (15), the resulting subproblem (8) admits the following closed form solution:

$$\widetilde{\mathbf{D}}^\nu_{(i)} = P_{\mathcal{D}}\left[\mathbf{D}^\nu_{(i)} - \frac{I}{\tau^\nu_{D,i}} \mathbf{\Theta}^\nu_{(i)} \right]. \tag{22}$$

Referring to the sparse coding subproblem (11), if $\tilde{h}_i$ is chosen according to (16), computing the update $\mathbf{X}_i^{\nu+1}$ results in solving a LASSO problem. If instead one uses the surrogate in (17), the solution of (11) can be computed in closed form as

$$\mathbf{X}_i^{\nu+1} = \frac{\tau^\nu_{X,i}}{\mu + \tau^\nu_{X,i}} \mathcal{T}_{\frac{\lambda}{\tau^\nu_{X,i}}} \left(\mathbf{X}_i^\nu - \frac{1}{\tau^\nu_{X,i}} \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu) \right), \tag{23}$$

where $\mathcal{T}$ is the soft-thresholding operator $\mathcal{T}_\theta(x) \triangleq \max(|x| - \theta, 0) \cdot \mathrm{sign}(x)$ [with $\mathrm{sign}(\cdot)$ denoting the sign function], applied to the matrix argument component-wise.

• *On the choice of $\tau^\nu_{X,i}$ and $\tau^\nu_{D,i}$.* These coefficients must satisfy Assumption D. Roughly speaking, D1 ensures that $(\tau^\nu_{X,i})_{i=1}^I$ and $(\tau^\nu_{D,i})_{i=1}^I$ are bounded (both from below and above) while D2 guarantees that these parameters are asymptotically "stable". A trivial choice for $\tau^\nu_{D,i}$ satisfying both (18) and (20) is $\tau^\nu_{D,i} = c$, for some $c > 0$; some practical rules for $\tau^\nu_{X,i}$ satisfying both (19) and (21) are the following:

(a) Use a constant $\tau^\nu_{X,i}$, that is,

$$\tau^\nu_{X,i} = \max_{\mathbf{D} \in \mathcal{D}} \left[\max\left(\sigma_{\max}(\nabla^2_{X_i} f_i(\mathbf{D}, \mathbf{X}_i^\nu)), \tilde{\epsilon} \right) \right],$$

for some $\tilde{\epsilon} > 0$. The above value can be, however, much larger than any $\sigma_{\max}(\nabla^2_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu))$, which can slow down the practical convergence of the algorithm;

(b) A less conservative choice is to satisfy (19) iteratively, while guaranteeing that $\tau^\nu_{X,i}$ is uniformly positive:

$$\tau^\nu_{X,i} = \max(L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}), \tilde{\epsilon}), \tag{24}$$

where $\tilde{\epsilon}$ is any positive (possibly small) constant;

(c) A generalization of (b) is

$$\tau^\nu_{X,i} \in \left[\max(L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}), \tilde{\epsilon}),\ L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}) + \tilde{\mu} \right],$$

for some $\tilde{\epsilon}$ and $\tilde{\mu}$ such that $0 < \tilde{\epsilon} \le \tilde{\mu} < \mu$.

Remark 1 *While the choices (a)-(c) above clearly satisfy* (19)*, it can be shown that* (21) *also holds, as a consequence of the continuity of $L_{\nabla X_i}(\cdot)$ and Proposition 5 (cf. Appendix A.4).*

Note that all the above rules do not require any coordination among the agents, but are implementable in a fully distributed manner, using only local information.

3.1.2. On the choice of γ^ν

The step-size can be chosen according to the following assumption.

Assumption E (On γ^ν) *$\{\gamma^\nu\}_\nu$ satisfies: $\gamma^\nu \in (0, 1]$, for all ν; $\sum_{\nu=0}^\infty \gamma^\nu = \infty$; and $\sum_{\nu=0}^\infty (\gamma^\nu)^2 < \infty$.*

The above assumption is the standard diminishing-rule; see, e.g., (Bertsekas and Tsitsiklis, 1997). Here, we only recall one rule, satisfying Assumption E, that we found very effective in our experiments, namely (Facchinei et al., 2015): $\gamma^\nu = \gamma^{\nu-1}(1 - \epsilon_0 \gamma^{\nu-1})$ with $\gamma^0 \in (0, 1]$ and $\epsilon_0 \in (0, 1/\gamma^0)$.

3.1.3. On the choice of the weigh coefficients $\{a_{ij}^\nu\}$.

We denote by $\mathbf{A}^\nu$ the matrix whose entries are the weights a_{ij}^ν's, i.e., $[\mathbf{A}^\nu]_{i,j} = a_{ij}^\nu$. This matrix is chosen so that the following conditions are satisfied.

Assumption F (On the weighting matrix) *Given the digraph $\mathcal{G}^\nu = (\mathcal{V}, \mathcal{E}^\nu)$, each matrix $\mathbf{A}^\nu$, with $[\mathbf{A}^\nu]_{ij} = a_{ij}^\nu$, satisfies*

(F1) $a_{ii}^\nu \geq \kappa > 0$ *for all* $i = 1, \ldots, I$;

(F2) $a_{ij}^\nu \geq \kappa > 0$, *if* $(j,i) \in \mathcal{E}^\nu$; *and* $a_{ij}^\nu = 0$ *otherwise;*

(F3) $\mathbf{A}^\nu$ *is column stochastic, i.e.,* $\mathbf{1}^\intercal \mathbf{A}^\nu = \mathbf{1}^\intercal$.

When the graph $\mathcal{G}^\nu$ is directed, a valid choice of $\mathbf{A}^\nu$ is (Kempe et al., 2003): $a_{ij}^\nu = 1/d_j^\nu$ if $j \in \mathcal{N}_i^{\text{in}}[\nu]$, and $a_{ij}^\nu = 0$ otherwise, where d_j^ν is the out-degree of agent j at time ν. The resulting communication protocols (12)–(13) can be easily implemented in a distributed fashion:each agent i) broadcasts its local variable normalized by its current out-degree; and ii) collects locally the information coming from its neighbors. When the graph is undirected, several options are available in the literature, including: the Laplacian, Metropolis-Hastings, and maximum-degree weights; see, e.g., (Xiao et al., 2005).

4. Convergence of D^{4}L

In this section, we provide the main convergence results for the D^{4}L Algorithm. We begin introducing some definitions, instrumental to state our results. Let

$$\mathbf{D}^\nu \triangleq [\mathbf{D}_{(1)}^{\nu\intercal}, \mathbf{D}_{(2)}^{\nu\intercal}, \ldots, \mathbf{D}_{(I)}^{\nu\intercal}]^\intercal, \quad \mathbf{X}^\nu \triangleq [\mathbf{X}_1^\nu, \mathbf{X}_2^\nu, \ldots, \mathbf{X}_I^\nu], \quad \text{and} \quad \overline{\mathbf{D}}^\nu \triangleq \frac{1}{I}\sum_{i=1}^{I} \mathbf{D}_{(i)}^\nu. \tag{25}$$

Given the sequence $\{(\mathbf{D}^\nu, \mathbf{X}^\nu)\}_\nu$ generated by the D^{4}L Algorithm, convergence is stated measuring the distance of the sequence $\{(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu\}_\nu$ from optimality as well as the consensus disagreement among the local variables $\mathbf{D}_{(i)}^\nu$'s. Distance from stationarity is measured by

$$\Delta^\nu \triangleq \max(\Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu), \Delta_X(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)) \tag{26}$$

where

$$\Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \triangleq ||\widehat{\mathbf{D}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) - \overline{\mathbf{D}}^\nu||_{\infty,\infty}, \quad \Delta_X(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \triangleq ||\widehat{\mathbf{X}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) - \mathbf{X}^\nu||_{\infty,\infty}, \tag{27}$$

with the functions $\widehat{\mathbf{D}}(\bullet,\bullet)$ and $\widehat{\mathbf{X}}(\bullet,\bullet)$ defined as:

$$\widehat{\mathbf{D}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \triangleq \operatorname*{argmin}_{\mathbf{D}' \in \mathcal{D}} \left\langle \nabla_D F(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu), \mathbf{D}' - \overline{\mathbf{D}}^\nu \right\rangle + \frac{\hat{\tau}_D}{2} \left\|\mathbf{D}' - \overline{\mathbf{D}}^\nu\right\|^2 + G(\mathbf{D}'), \tag{28}$$

$$\widehat{\mathbf{X}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \triangleq [\widehat{\mathbf{X}}_1(\overline{\mathbf{D}}^\nu, \mathbf{X}_1^\nu), \ldots, \widehat{\mathbf{X}}_I(\overline{\mathbf{D}}^\nu, \mathbf{X}_I^\nu)],$$

$$\text{with } \widehat{\mathbf{X}}_i(\overline{\mathbf{D}}^\nu, \mathbf{X}_i^\nu) \triangleq \operatorname*{argmin}_{\mathbf{X}_i' \in \mathcal{X}_i} \left\langle \nabla_{X_i} f_i(\overline{\mathbf{D}}^\nu, \mathbf{X}_i^\nu), \mathbf{X}_i' - \mathbf{X}_i^\nu \right\rangle + \frac{\hat{\tau}_X}{2} \left\|\mathbf{X}_i' - \mathbf{X}_i^\nu\right\|^2 + g_i(\mathbf{X}_i'), \tag{29}$$

for some given constants $\hat{\tau}_D > 0$ and $\hat{\tau}_X > 0$. Note that $\Delta_D(\bullet,\bullet)$ and $\Delta_X(\bullet,\bullet)$ are valid merit functions, in the sense that they are continuous and $\Delta_D(\overline{\mathbf{D}}^\infty, \mathbf{X}^\infty) = \Delta_X(\overline{\mathbf{D}}^\infty, \mathbf{X}^\infty) = 0$ if and only if $(\overline{\mathbf{D}}^\infty, \mathbf{X}^\infty)$ is a stationary solution of Problem (P) (Facchinei et al., 2015).

The consensus error at iteration ν is measured by the function

$$e^\nu \triangleq ||\mathbf{D}^\nu - \mathbf{1} \otimes \overline{\mathbf{D}}^\nu||_{\infty,\infty}. \tag{30}$$

Asymptotic convergence of D^{4}L to stationary solutions of (P) is stated in Theorem 2 below while the convergence rate is studied in Theorem 3.

Theorem 2 *Given Problem P under Assumption A, let $\left\{(\mathbf{D}^\nu, \mathbf{X}^\nu)\right\}_\nu$ be the sequence generated by the D^{4}L Algorithm for a given initial point $(\mathbf{D}^0, \mathbf{X}^0)$ and under Assumptions B, C, D1, E, F. Then,*

(a) `[Consensus]`*: All $\mathbf{D}_{(i)}^\nu$'s are asymptotically consensual, i.e., $\lim_{\nu\to\infty} e^\nu = 0$;*

(b) `[Convergence]`*: i) $\{(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ is bounded; ii) $\{U(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ converges to a finite value; iii) $\lim_{\nu\to\infty} \Delta_X(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$; and iv) $\liminf_{\nu\to\infty} \Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$. Therefore, $\{(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ has at least one limit point which is a stationary solution of P.*

If, in particular, Assumption A5(ii) holds and D1 is reinforced by D2, then convergence in (b) can be strengthened as follows:

(b') *Case (b) holds and $\lim_{\nu\to\infty} \Delta^\nu = 0$, implying that all the limit points of $\{(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ are stationary solutions of P.*

Proof The proof is quite involved and is given in Appendix A.2. ■

The above theorem states two main convergence results under Assumptions B, C, D1, E, F: i) existence of at least a subsequence of $(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ converging to a stationary solution of Problem P; and ii) asymptotic consensus of all $\mathbf{D}_{(i)}^\nu$ to a common value $\overline{\mathbf{D}}^\nu$. If Assumption A5(ii) is also assumed and D1 is reinforced by D2 the stronger results in (b') can be proven, showing that every limit point is a stationary solution. Note that from a practical point of view the weaker result guaranteeing existence of at least a subsequence converging to a stationary solution is perfectly satisfying, since it guarantees that the algorithm can be terminated after a finite number of iterations with an approximate solution.

Theorem 3 *Consider either settings of Theorem 2, with the additional assumption that the step-size sequence $\{\gamma^\nu\}_\nu$ is non-increasing. For any given $\epsilon > 0$, let $T_{D,\epsilon} \triangleq \min\{\nu \in \mathbb{N}_+ : \Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \leq \epsilon\}$ and $T_{X,\epsilon} \triangleq \min\{\nu \in \mathbb{N}_+ : \Delta_X(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \leq \epsilon\}$. Then,*

(a) `[Rate of consensus error]`*:*

for every $\theta \in (0,1)$; $$e^\nu = \mathcal{O}\left(\gamma^{\lceil \theta\nu \rceil}\right), \tag{31}$$

(b) `[Rate of optimization errors]`*:*

$$T_{X,\epsilon} = \mathcal{O}\left(\frac{1}{\epsilon^2}\right). \tag{32}$$

Let $\gamma^\nu = K/\nu^p$ with some constant $K > 0$ and $p \in (1/2, 1)$. Then,

$$T_{D,\epsilon} = \mathcal{O}\left(\frac{1}{\epsilon^{2/(1-p)}}\right). \tag{33}$$

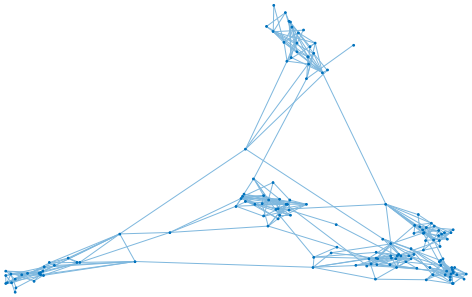


Figure 3: Topology of a simulated (sparse) network

Proof See Appendix A.3 ■

We remark that, while a convergence rate has been established in the literature (see, e.g., Razaviyayn et al. (2014a)) for certain *centralized* algorithms applied to special classes of DL problems, Theorem 3 represents the first rate result for a *distributed* algorithm tackling the class of DL problems P.

5. Numerical Experiments

In this section, we test numerically our algorithmic framework on several classes of problems, namely: (i) Image denoising, (ii) Biclustering, (iii) Sparse PCA, and (iv) Non-negative sparse coding. We recall that D^{4}L is the first provably convergent distributed algorithm for Problem P; comparisons are thus not simple. To give the sense of the performance of D^{4}L, in our experiments,

(i) when available, we implemented, *centralized* algorithms tailored to the specific problems under consideration and used the results as *benchmarks*;

(ii) for undirected graphs, we extended the (distributed) Prox-PDA-IP (Zhao et al., 2016) algorithm to the simulated instances of Problem P (generalizations of this method to directed graphs seem not possible);

(iii) for both undirected and directed graphs, we implemented a suitable version of the Adapt-Then-Combine (ATC) Algorithm (Chainais and Richard, 2013). Note that ATC has no formal convergence proof, and is originally designed to handle only undirected graphs, but we managed to make a sensible extension of this method to directed graphs too, by using some of the ideas developed in this paper.

All codes are written in MATLAB 2016b, and implemented on a computer with Intel Xeon (E5-1607 v3) quad-core 3.10GHz processor and 16.0 GB of DDR4 main memory.

5.1. Image Denoising

Problem formulations: We consider denoising a 512×512 pixels image of a fishing boat (USC, 1997)−see Fig. 5(a). We simulate a cluster computer network composed of 150 nodes (computers). Denoting by $\mathbf{F}_0$ and $\mathbf{F}$ the noise-free and corrupted image, respectively, the SNR (in dB) is defined as SNR $\triangleq 20 \cdot \log(||\text{vec}(\mathbf{F}_0)||_2/\sqrt{\text{MSE}})$ while the Peak SNR (in dB) is

defined as PSNR $\triangleq 20 \cdot \log(\max_i(\text{vec}(\mathbf{F}_0))_i/\sqrt{\text{MSE}})$, where MSE is the Mean-Squared-Error between $\mathbf{F}_0$ and $\mathbf{F}$. The fishing boat image is corrupted by additive white Gaussian noise, so that SNR = 15 dB and PSNR = 20.34 dB.

To perform the denoising task, we consider the elastic net sparse DL formulation (3). We extract 255,150 square sliding $s \times s$ pixel patches ($s = 8$) and aggregate the vectorized extracted patches in a single data matrix $\mathbf{S}$ of size $64 \times 255,150$. The size of the dictionary is $s^2 \times s^2 = 64 \times 64$; the data matrix is equally distributed across the 150 nodes, resulting in sparse representation matrices $\mathbf{X}_i$ of size 64×1701 ($K = 64$ and $n_i = 1701$). The total number of optimization variables is then $16,333,696$. The free parameters λ, μ and α in (3) are set to $\lambda = 1/s$, $\mu = \lambda$ and $\alpha = 1$, respectively.

Algorithms and tuning: We tested: i) two instances of the D^{4}L Algorithm, corresponding to two alternative choices of the surrogate functions; ii) the Prox-PDA-IP algorithm (Zhao et al., 2016), adapted to problem (3) (only on undirected networks); iii) the ATC algorithm (Chainais and Richard, 2013); and iv) the *centralized* K-SVD algorithm (Elad and Aharon, 2006) (`KSVD-Box v13` package), used as a benchmark. More specifically, the two instances of the D^{4}L Algorithm are:

- `Plain` D^{4}L: $\tilde{h}_i$ is chosen as in (16) (the original function) and $\tilde{f}_i$ as in (15);
- `Linearized` D^{4}L: $\tilde{h}_i$ is given by (17) (first-order approximation) and $\tilde{f}_i$ is given by (15).

The rest of the parameters in both instances of D^{4}L is set as: $\gamma^\nu = \gamma^{\nu-1}(1 - \epsilon\gamma^{\nu-1})$, with $\gamma^0 = 0.5$ and $\epsilon = 10^{-2}$; $\tau^\nu_{D,i} = 10$; and $\tau^\nu_{X,i} = \max(L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}), 1)$ [cf. (24)].

Our adaptation of the Prox-PDA-IP algorithm to Problem (3) is summarized in Algorithm 2. The difference with the original version in (Zhao et al., 2016) are: i) the elastic net penalty is used in the objective function for the $\mathbf{X}_i$'s variables, instead of the ℓ_1-norm and ℓ_2-norm ball constraints; and ii) the variables $\mathbf{D}_{(i)}$'s are constrained in $\mathcal{D} \triangleq \{\mathbf{D} : ||\mathbf{D}\mathbf{e}_k||_2 \leq \alpha, k = 1, 2, \ldots, K\}$ rather than using the ℓ_2-norm regularization in the objective function. The other symbols used in Algorithm 2 are: i) the incidence matrix of $\mathcal{G}$, denoted by $\mathbf{M} = (M_{ei})_{e,i} \in \mathbb{R}^{E \times I}$, with $E \triangleq |\mathcal{E}|$; ii) the matrices $\mathbf{\Omega}^\nu_e \in \mathbb{R}^{M \times K}$, $e = 1, ..., E$, which are the ν-th iterate of the dual matrix variables $\mathbf{\Omega}_e \in \mathbb{R}^{M \times K}$, as introduced in the original Prox-PDA-IP; and iii) $\{\beta^\nu\}_{\nu \in \mathbb{N}_+}$ is the increasing penalty parameter, set to $\beta^\nu = 0.002\nu$.

All the algorithms are initialized to the same value: $\mathbf{D}^0_{(i)}$'s coincide with randomly (uniformly) chosen columns of $\mathbf{S}_{(i)}$'s whereas all $\mathbf{X}^0_i$'s are set to zero.

While the subproblems solved at each iteration ν in Linearized D^{4}L admit a closed-form−see (23) and (22)−in both Plain D^{4}L and ATC, the update of the dictionary has the closed form expression (22), but the update of the private variables calls for the solution of a LASSO problem (cf. Sec. 3.1). For both Plain D^{4}L and ATC, the LASSO subproblems at iteration ν are solved using the (sub)gradient algorithm, with the following tuning. A diminishing step-size is used, set to $\gamma^r = \gamma^{r-1}(1 - \epsilon\gamma^{r-1})$, where $\gamma^0 = 0.9$, $\epsilon = 10^{-3}$, and r denotes the inner iteration index. A warm start is used for the subgradient algorithm: the initial points are set to $\mathbf{X}^\nu_i$, where ν is the iteration index of the outer loop. We terminate

Algorithm 2 : Prox-PDA-IP algorithm (Zhao et al., 2016)

Initialization : $\mathbf{D}_{(i)}^0 \in \mathcal{D}$, $\mathbf{X}_i^0 \in \mathcal{X}_i$, $\mathbf{\Omega}^0 = \mathbf{0}$;

`S1`. If $(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)_i$ satisfies stopping criterion: `STOP`;

`S2`. Each agent i computes $\theta_i^\nu = ||\mathbf{D}_{(i)}^\nu \mathbf{X}_i^\nu - \mathbf{S}_i||_F^2$ and:

(a) $\mathbf{X}_i^{\nu+1} = \underset{\mathbf{X}_i \in \mathbb{R}^{K \times n_i}}{\operatorname{argmin}} \; f_i(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i) + g_i(\mathbf{X}_i) + \frac{\beta^{\nu+1}\theta_i^\nu}{2}||\mathbf{X}_i - \mathbf{X}_i^\nu||_F^2 + \frac{\beta^{\nu+1}}{2}||\mathbf{D}_{(i)}^\nu(\mathbf{X}_i - \mathbf{X}_i^\nu)||_F^2$;

(b) $\mathbf{D}_{(i)}^{\nu+1} = \underset{\mathbf{D}_{(i)} \in \mathcal{D}}{\operatorname{argmin}} \; f_i(\mathbf{D}_{(i)}, \mathbf{X}_i^{\nu+1}) + \sum_{e=1}^E M_{ei} \left\langle \mathbf{\Omega}_e^\nu, \mathbf{D}_{(i)} \right\rangle$
$+\beta^{\nu+1}\left(d_i||\mathbf{D}_{(i)}||_F^2 - \left\langle \mathbf{D}_{(i)}, (d_i - 1)\mathbf{D}_{(i)}^\nu + \sum_{j \in \mathcal{N}_i} \mathbf{D}_{(j)}^\nu \right\rangle\right)$;

(c) $\mathbf{\Omega}_e^{\nu+1} = \mathbf{\Omega}_e^\nu + \beta^{\nu+1} \sum_{i=1}^I M_{ei} \mathbf{D}_{(i)}^{\nu+1}, \quad \forall e = (i,j) \in \mathcal{E}$;

`S3`. Set $\nu + 1 \to \nu$, and go to `S1`.

the subgradient algorithm in the inner loop when $J_i^r \leq 10^{-6}$, with

$$J_i^r \triangleq \left\| \mathbf{X}_i^{\nu,r} - \frac{s}{1+s} \mathcal{T}_{\frac{1}{s}} \left(\mathbf{X}_i^{\nu,r} - \left(\nabla_{X_i} f_i(\mathbf{U}_{(i)}^\nu, \mathbf{X}_i^{\nu,r}) + \tau_{X,i}^\nu (\mathbf{X}_i^{\nu,r} - \mathbf{X}_i^\nu) \right) \right) \right\|_{\infty,\infty},$$

where $\mathbf{X}_i^{\nu,r}$ denotes the value of $\mathbf{X}_i$ at the r-th inner iteration and outer iteration ν; and $\mathcal{T}_\theta(x) \triangleq \max(|x| - \theta, 0) \cdot \operatorname{sign}(x)$ is the soft-thresholding operator, applied to the matrix argument componentwise. In all our simulations, we observed that the above accuracy was reached within 30 (inner) iterations of the subgradient algorithm.

In the Prox-PDA-IP scheme, Step `S2` (cf. Algorithm 2) calls for the solution of two subproblems, including a LASSO problem. As for Plain D^{4}L and ATC, we used the (projected) (sub)-gradient algorithm (with the same diminishing step-size rule) to solve the subproblems; we terminated the inner loop when the length between two consecutive iterates of the (projected) (sub)-gradient algorithm goes for the first time below 10^{-6}.

We simulated both undirected and directed static graphs. In the former case, there is no need of the ϕ-variables and, in the second equation of (12) [and (13)], the terms $(\phi_j^\nu a_{ij}^\nu)/\phi_i^{\nu+1}$ reduce to a_{ij}. The weights a_{ij} are chosen according to the Metropolis-Hasting rule (Xiao et al., 2007); the resulting matrix $\mathbf{A}^\nu = [a_{ij}]_{ij}$ is thus time-invariant and doubly stochastic. When the graph is directed, we use the update of the ϕ_i^ν's as in (12), with the weights a_{ij}^ν chosen according to the push-sum protocol (Kempe et al., 2003) (cf. Sec. 3.1.3).

Convergence speed and quality of the reconstruction: In the first set of simulations, we considered an undirected graph composed of 150 nodes, clustered in 6 groups of 25 (see Fig. 3). Starting from this topology, we kept adding random edges till a connected graph was obtained. Specifically, an arc is added between two nodes in the same cluster (resp. different clusters) with probability $p_1 = 0.2$ (resp. $p_2 = 2 \times 10^{-3}$).

In Fig. 4 we plot the objective function value [subplot on the left], the consensus disagreement e^ν as in (30) [subplot in the center], and the distance from stationarity Δ^ν as in

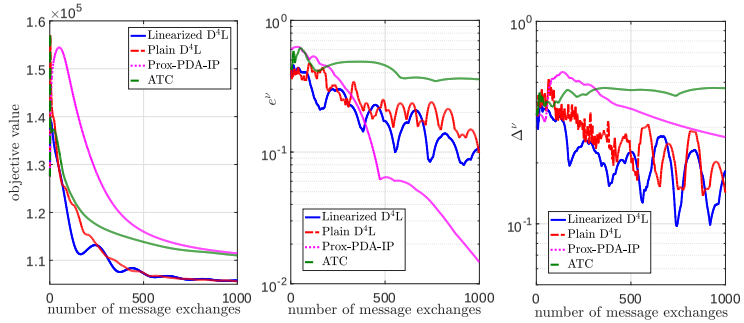


Figure 4: Denoising problem – D^{4}L, Prox-PDA-IP and ATC algorithms: objective value [subplot on the left], consensus disagreement [subplot in the center], and distance from stationarity Δ^ν [cf. (26)] [subplot on the right] vs. number of message exchanges.

(28) [subplot on the right] versus the *number of message exchanges*, achieved by Plain D^{4}L, Linearized D^{4}L, Prox-PDA-IP, and ATC. Note that the number of messages exchanged in the ATC algorithm at iteration ν coincides with ν whereas for Prox-PDA-IP and the D^{4}L schemes is 2ν (recall that the latter schemes employ two steps of communications per iteration). The figures clearly show that both versions of D^{4}L are much faster than Prox-PDA-IP and ATC (or, equivalently, they require fewer information exchanges). Moreover, ATC does not seem to reach a consensus on the local copies of the dictionary, while Prox-PDA-IP and D^{4}L schemes reach an agreement quite soon. In Fig. 5, we plot the reconstructed images along with their PSNR and MSE, obtained by the algorithms, when terminated after 1000 message exchanges. The figures clearly show superior performance of D^{4}L over its competitors. Also, the values of PSNR and MSE achieved by D^{4}L are comparable with those obtained by (centralized) K-SVD (`KSVD-Box v13` package).

A closer look at Fig. 4 shows that a significant decay on the objective function occurs in the first 200 message exchanges. It is then interesting to check the quality of the reconstructed images, achieved by the algorithms if terminated then. In Fig. 6, we report the images and values of PSNR and MSE obtained by terminating the schemes after 200 message exchanges (we also plot the benchmark obtained by K-SVD, run till optimality). The figure shows that both versions of D^{4}L attain high quality solutions even if terminated after few message exchanges while ATC and Prox-PDA-IP lag behind. This means that, in practice, there is no need to run D^{4}L till very low values of e^ν and Δ^ν are achieved.

Since the algorithms do not have the same cost-per-iteration, to get further insights into the performance of these schemes, we also compare them in terms of running time. In Table 2, we report the *averaged elapsed time* to execute one iteration of all algorithms. We considered the same setting as in the previous figures, but we terminated all algorithms after 273 seconds, which corresponds to the time for the fastest algorithm (i.e. Linearized D^{4}L) to perform 200 message exchanges [cf. Fig. 6]. The associated reconstructed images are shown in Fig. 7. Once again, these results clearly show that the linearized D^{4}L scheme significantly outperforms Prox-PDA-IP and ATC. Also, Linearized D^{4}L performs considerably better

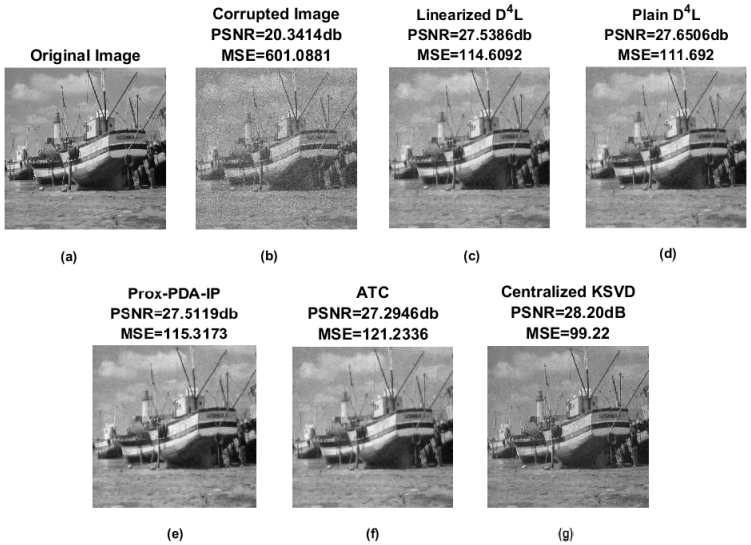


Figure 5: Denoising outcome. (a): original image; (b): corrupted image; (c)-(f): denoising achieved by D^{4}L, Prox-PDA-IP and ATC terminated after 1000 message exchanges; and (g): denoising achieved by centralized K-SVD (`KSVD-Box v13`).

than Plain D^{4}L, when terminated early; the explanation is in Table 2 which shows that the time of one iteration of the former algorithm is much shorter than that of Plain D^{4}L.

Algorithm	Average Time per Iteration (sec)
Linearized D^{4}L	2.862
Plain D^{4}L	11.328
Prox-PDA-IP	30.98
ATC	9.838

Table 2: D^{4}L vs. Prox-PDA-IP and ATC: Average computation time per iteration

Impact of the graph topology and connectivity: We study now the influence of the topology and graph connectivity on the performance of the algorithms. We consider *directed, static* graphs. We generated 5 instances of digraphs, with different connectivity, according to the following procedure. There are 500 nodes ($I = 500$), which are clustered in $n_c = 50$ clusters, each of them containing $10 = I/n_c$ nodes. Each node has an outgoing arc to another node in the same cluster with probability p_1 while p_2 is the probability of an outgoing arc to a node in a different cluster. We chose the values of p_1 and p_2, as in Table 3; we simulated three scenarios, namely: N1 corresponds to a "highly" connected network, N3 describes a "poorly" connected scenario, and N2 is an intermediate case. For

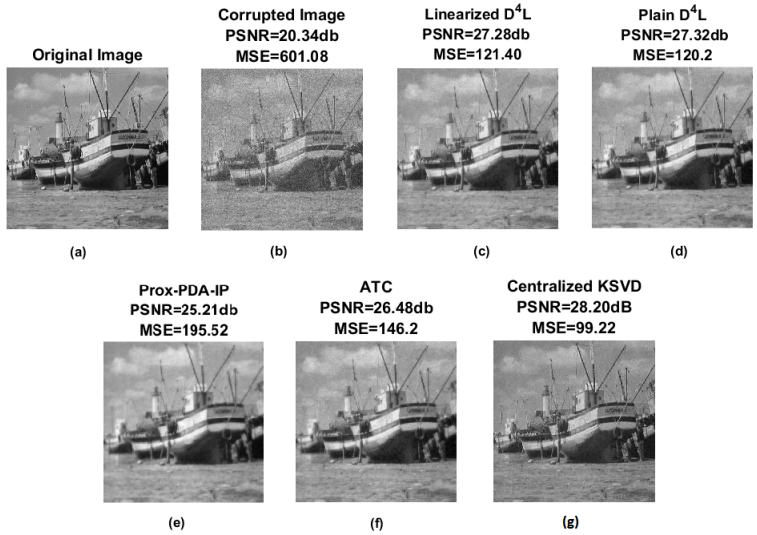


Figure 6: Denoising outcome. (a): original image; (b): corrupted image; (c)-(f): denoising achieved by D^{4}L, Prox-PDA-IP and ATC terminated after 200 message exchanges; and (g): denoising achieved by centralized K-SVD (`KSVD-Box v13`).

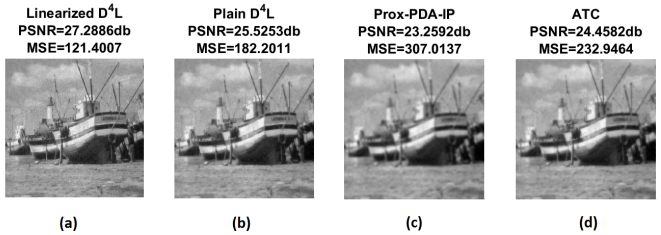


Figure 7: Denoising outcome. (a): Linearized D^{4}L; (b): Plain D^{4}L; (c): Prox-PDA-IP; (d) ATC; all terminated after after 273 seconds run-time (corresponding to 200 message exchanges of the Linearized D^{4}L).

each scenario, we generated 5 random instances (if a generated graph was not strongly connected we discarded it and generated a new one) and then ran Plain and Linearized D^{4}L and ATC on the resulting 15 graphs. Recall that ATC was not designed to work on directed networks. We thus modified it by using our new consensus protocol (but not the gradient tracking mechanism); we term it *Modified* ATC.

In Fig. 8 we plot the *average* value of the objective function [subplot on the left], the consensus disagreement e^ν [subplot in the center], and the distance from stationarity Δ^ν [subplot on the right], achieved by Plain D^{4}L, Linearized D^{4}L, and Modified ATC, versus the number of message exchanges, for the three scenarios N1 [subplot (a)], N2 [subplot (b)]

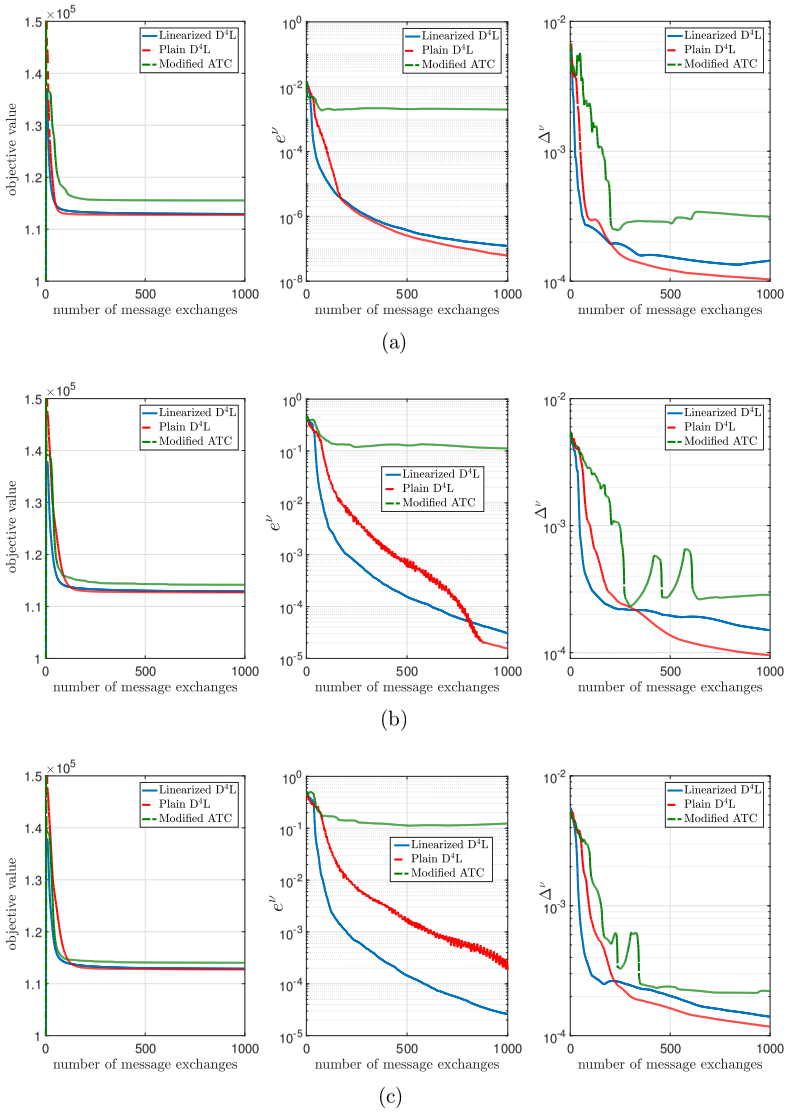


Figure 8: Denoising problem–D^{4}L and Modified ATC algorithms: objective value [subplots on the left], consensus disagreement [subplots in the center], and distance from stationarity Δ^ν [cf. (26)] [subplots on the right] vs. number of message exchanges. Comparison over three network settings [cf. Table 3]: N1 [subplots (a)], N2 [subplots (b)], and N3 [subplots (c)].

Network #	I	n_c	p_1	p_2
N1	500	50	0.9	0.9
N2	500	50	0.1	0.01
N3	500	50	0.05	0.01

Table 3: Network setting

and N3 [subplot (c)]. The average is taken over the aforementioned 5 digraph realizations. While also this batch of tests confirms the better behavior of D^{4}L schemes over ATC, it is interesting to observe that there seems to be little influence of the degree of connectivity on the behavior of Linearized and Plain D^{4}L. The only aspect for which a reasonable influence can be seen is on consensus. In fact, with respect to consensus, Linearized D^{4}L seems to improve over Plain D^{4}L, when connectivity decreases. This has a natural interpretation. Plain D^{4}L solves much more accurate subproblems at each iteration and this is, in some sense useless, especially in early iterations, when information has not spread across the network. It seems clear that the less connected the graph, the more time information needs to spread. Therefore, in scenario N1, the two methods are almost equivalent and, looking at consensus error, we see that initially Linearized D^{4}L is better than Plain D^{4}L, but soon, as information spreads, Plain D^{4}L becomes, even if slightly, better than Linearized D^{4}L. The same behavior can be observed for scenario N2, but this time the initial advantage of Linearized D^{4}L is larger and the switching point is reached much later. This is consistent with the fact that information needs more time to spread and therefore solving the accurate subproblem is not advantageous. If one passes to N3, where connectivity is very loose, there is no switching point within the first 1000 message exchanges.

5.2. Biclustering

Biclustering has been shown to be useful in several applications, including biology, information retrieval, and data mining; see, e.g., (Madeira and Oliveira, 2004).

Problem Formulation: We consider a Biclustering problem in the form (6), applied to genetic information. We solved the problem simulating a networked computer cluster composed of 500 nodes (see Table 3). The genetic data is borrowed from (Lee et al., 2010) (centered and normalized): the data matrix $\mathbf{S}$ of size $56 \times 12,625$ ($M = 56$ and $N = 12,625$) contains microarray gene expressions of 56 patients (rows); each patient is either identified to be normal (Normal) or belonging to one of the following three types of lung cancer: pulmonary carcinoid tumors (Carcinoid), colon metastases (Colon), and small cell carcinoma (SmallCell). We considered the *unsupervised* instance of the problem, meaning that none of the a-priori information about the type of patients' cancer is used to perform biclustering. Following the numerical experiments of (Lee et al., 2010), we seek rank-3 sparse matrices $\mathbf{X}_i$, and the data matrix $\mathbf{S}$ is equally distributed across the 500 nodes, resulting thus in $K = 3$ and $n_i = 26$. The total number of variables is then $39,168$. The other parameters are set as follows: $\alpha = 1$, $\lambda_X = \mu_X = 0.1$, and $\lambda_D = \mu_D = 0.1$.

Algorithms and tuning: We tested the instance of D^{4}L where $\tilde{f}_i$ and $\tilde{h}_i$ are chosen according to (14) and (16), respectively. The rationale behind this choice is to exploit the

extra structure of the original function f_i, plus, in case of $\tilde{f}_i$, there is no certain benefit in using the linear approximation (15) as it does *not* lead to any closed form solution of the subproblem (8). Note that the Prox-PDA-IP scheme is not applicable here since the network is directed.

The other parameters of the algorithm are set to: $\gamma^\nu = \gamma^{\nu-1}(1-\epsilon\gamma^{\nu-1})$, with $\gamma^0 = 0.2$ and $\epsilon = 10^{-2}$; and $\tau^\nu_{D,i} = 100$ and $\tau^\nu_{X,i} = \max(L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}), 100)$. We term such an instance of D^{4}L *Plain D^{4}L*. We compared Plain D^{4}L with the following algorithms: i) (a modified version of) the distributed ATC algorithm (Chainais and Richard, 2013), where the optimization of $\mathbf{D}$ is adjusted to solve (6) (the elastic-net penalty is added), and the consensus mechanism is modified with our new consensus protocol to handle directed network topologies; we termed this instance *Modified ATC*; and ii) the centralized SSVD algorithm proposed in Lee et al. (2010) (implemented using the MATLAB code provided by the authors), to benchmark the results obtained by the distributed algorithms. All the distributed algorithm are initialized setting each $\mathbf{X}^0_i = \mathbf{0}$, and each $\mathbf{D}^0_{(i)}$ equal to some randomly chosen columns of $\mathbf{S}_i$.

In D^{4}L, the subproblems (8) and (11) at iteration ν do not have a closed form solution; they are solved using the projected (sub)gradient algorithm, with diminishing step-size $\gamma^r = \gamma^{r-1}(1-\epsilon\gamma^{r-1})$, where $\gamma^0 = 0.9$, $\epsilon = 10^{-3}$, and r denotes the inner iteration index. A warm start is used for the projected subgradient algorithm; the initial points are set to $\mathbf{D}^\nu_{(i)}$ and $\mathbf{X}^\nu_i$ in problems (8) and (11), respectively, where ν is the iteration index of the outer loop. We terminate the projected subgradient algorithm solving (8) and (11) when $J^r_{D,i} \triangleq \|\widehat{\mathbf{D}}^{\nu,r}_{(i)} - \mathbf{D}^{\nu,r}_{(i)}\|_{\infty,\infty} \leq 10^{-6}$ and $J^r_{X,i} \triangleq \|\widehat{\mathbf{X}}^{\nu,r}_i - \mathbf{X}^{\nu,r}_i\|_{\infty,\infty}, \leq 10^{-6}$, respectively, where

$$\widehat{\mathbf{D}}^{\nu,r}_{(i)} \triangleq \underset{\mathbf{D}_{(i)}\in\mathcal{D}}{\operatorname{argmin}} \left\langle \nabla_D f_i(\mathbf{D}^{\nu,r}_{(i)}, \mathbf{X}^\nu_i) + I\mathbf{\Theta}^\nu_{(i)} - \nabla_D f_i(\mathbf{D}^\nu_{(i)}, \mathbf{X}^\nu_i) + \tau^\nu_{D,i}(\mathbf{D}^{\nu,r}_{(i)} - \mathbf{D}^\nu_{(i)}), \mathbf{D}_{(i)} - \mathbf{D}^{\nu,r}_{(i)} \right\rangle + \frac{100}{2}\left\|\mathbf{D}_{(i)} - \mathbf{D}^{\nu,r}_{(i)}\right\|^2 + G\left(\mathbf{D}_{(i)}\right),$$

$$\widehat{\mathbf{X}}^{\nu,r}_i \triangleq \underset{\mathbf{X}_i\in\mathbb{R}^{K\times n_i}}{\operatorname{argmin}} \left\langle \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}^{\nu,r}_i) + \tau^\nu_{X,i}\left(\mathbf{X}^{\nu,r}_i - \mathbf{X}^\nu_i\right), \mathbf{X}_i - \mathbf{X}^{\nu,r}_i \right\rangle + \frac{100}{2}\|\mathbf{X}_i - \mathbf{X}^{\nu,r}_i\|^2 + g_i\left(\mathbf{X}_i\right),$$

with $\mathbf{D}^{\nu,r}_{(i)}$ and $\mathbf{X}^{\nu,r}_i$ denoting the value of $\mathbf{D}_{(i)}$ and $\mathbf{X}_i$ at the ν-th outer and r-th inner iteration, respectively. In all our simulations, the above accuracy was reached within 50 (inner) iterations of the projected subgradient algorithm.

Convergence speed and quality of the reconstruction: We simulated 3 *directed static* network topologies, namely: N1-N3, as given in Table 3. In Fig. 9 we plot he *average* value of the objective function [subplot on the left], the consensus disagreement e^ν [subplot in the center], and the distance from stationarity Δ^ν [subplot on the right], achieved by Plain D^{4}L and Modified ATC, versus the number of message exchanges, for the three scenarios N1 [subplot (a)], N2 [subplot (b)] and N3 [subplot (c)]. The average is taken over 5 digraph realizations. Fig. 9 shows that Plain D^{4}L algorithm attains satisfactory merit values in all network scenarios, while Modified ATC fails to reach consensus/convergence, even in highly connected networks. The poor performance of Modified ATC seem mainly due to the incapability of locking the consensus.

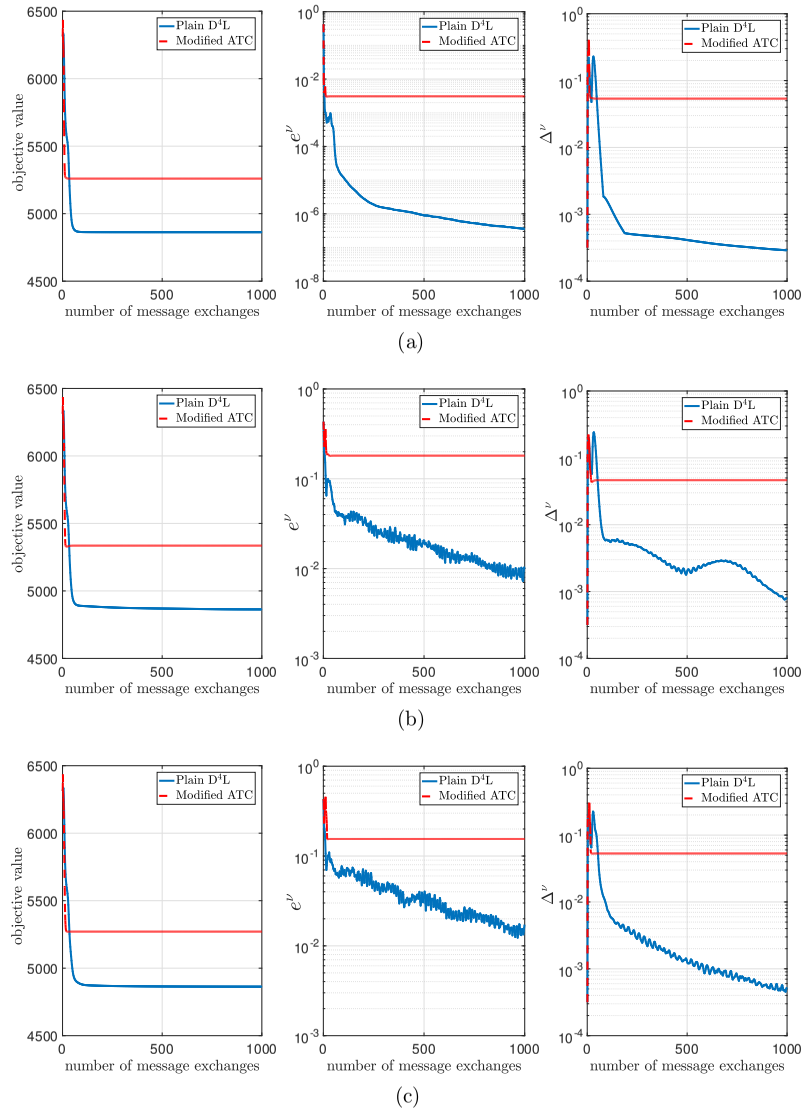


Figure 9: Biclustering problem–Plain D^{4}L and Modified ATC algorithms: objective value [subplots on the left], consensus disagreement [subplots in the center], and distance from stationarity Δ^ν [cf. (26)] [subplots on the right] vs. number of message exchanges. Comparison over three network settings [cf. Table 3]: N1 [subplots (a)], N2 [subplots (b)], and N3 [subplots (c)].

In order to assess the quality of the solutions achieved by the three algorithms, we employ the following procedure. Given the limit point (up to the fixed accuracy) $\mathbf{D}^\infty$ of the algorithm under consideration, patients' information is in form of (unlabeled clusters of) data points $\{\mathbf{D}^\infty_{m,:}\}_{m=1}^{56}$, where $\mathbf{D}^\infty_{m,:}$ denotes the m-th row of $\mathbf{D}^\infty$ and represents an individual patient. In order to compare $\mathbf{D}^\infty$ with the labeled ground truth, we need to tag labels to the clustered points of $\mathbf{D}^\infty$. To do so, we run the K-means clustering algorithm on $\{\mathbf{D}^\infty_{m,:}\}_{m=1}^{56}$. Specifically, we first run K-means 100 times and, in each run, we perform a preliminary clustering using 10% of the points (randomly chosen). Then, among the 100 obtained clustering configurations, we picked the one with the smallest "within-cluster sum of point-to-centroid distances".[1] Finally, we assign to each cluster the label associated with the most populated type of cancer in the cluster. Denoting the ground truth classes by $\{\mathcal{C}_i\}_{i=1}^4$ (recall that there are 4 classes/types of cancer), where each $\mathcal{C}_i$ consists of the group of patients with the same type of cancer, and by $\{\widetilde{\mathcal{C}}_i\}_{i=1}^4$ the clustering obtained by the procedure described above applied to the outcome $\mathbf{D}^\infty$ of the simulated algorithms, we measure the quality of the clustering by the *Jaccard index*, defined as

$$J = \frac{\left|\bigcup_i \left(\mathcal{C}_i \cap \widetilde{\mathcal{C}}_i\right)\right|}{\sum_i \left|\mathcal{C}_i \cup \widetilde{\mathcal{C}}_i\right|}.$$

Clearly $0 \leq J \leq 1$, and the higher the index value, the better the quality of the clustering.

In Table 4, we report the average and Maximum Absolute Deviation (MAD) of the Jaccard indices from their average, computed over the aforementioned 5 realizations of the three graph topologies, as in Table 3 (see also Fig. 9). The values in the table clearly show that Plain D^{4}L achieves better results than those produced by Modified ATC or centralized methods. Moreover, the value of the Jaccard index from Plain D^{4}L does not depend on the specific network topology. which is not the case for Modified ATC.

Network #	Plain D^{4}L	Modified ATC	Lee et al. (2010)
N1	0.8983/0	0.7778/0	–
N2	0.8983/0	0.7045/0.3218	–
N3	0.8983/0	0.7892/0.0172	–
Centralized	–	–	0.7231/–

Table 4: Biclustering problem−Average/MAD of Jaccard indices over 5 realizations of digraphs.

5.3. Non-negative Sparse Coding (NNSC) and Sparse PCA (SPCA)

Problem Formulation: We consider the Non-negative Sparse Coding (NSC) formulation (7) (Hoyer, 2004) and the Sparse PCA problem (6) (Mairal et al., 2010). For both formulations, we run experiments using the following two datasets:

- MIT-CBCL face database #1 (Sung, 1996): a pool of $N = 2,429$ vectorized face images of size 19×19 pixels each (i.e. $M = 361$);

1. Given a clustering partition $\{\mathcal{C}_i\}_{i=1}^4$, the "within-cluster sum of point-to-centroid distances" measures the quality of the k-means clustering, and is defined as $\sum_{i=1}^4 \sum_{j\in\mathcal{C}_i} ||\mathbf{D}^\infty_{j,:} - \overline{\mathbf{D}}^\infty_{\mathcal{C}_i}||^2$, where $\overline{\mathbf{D}}_{\mathcal{C}_i} \triangleq \frac{1}{|\mathcal{C}_i|}\sum_{j\in\mathcal{C}_i}\mathbf{D}^\infty_{j,:}$ and $|\mathcal{C}_i|$ denotes the cardinality of the set $\mathcal{C}_i$.

- The VOC 2006 database (Everingham et al., 2010): a pool of $N = 10,000$ vectorized natural image patches of size 16×16 pixels each (i.e. $M = 256$).

Consistently with (Mairal et al., 2010), the free parameters are set as:

- NNSC (7): $K = 49$, $\lambda = \mu = 1/\sqrt{M}$, and $\alpha = 1$;
- Sparse PCA (6): $K = 49$, $\lambda_X = \mu_X = 1/\sqrt{M}$, $\lambda_D = \mu_D = 1/\sqrt{M}$, and $\alpha = 1$.

The total number of variables for the above optimization problems are 136,710 for the MIT-CBCL dataset, and 502,544 for the VOC 2006 dataset.

We simulated the communication network as *static directed* graphs of size I, clustered in n_c groups, where each node has an outgoing arc to another node in the same cluster with probability p_1, while p_2 is the probability of an outgoing arc to a node in a different cluster. We run our tests over 6 different network scenarios, with various size I and probability pair (p_1, p_2), as given in Table 5. Note that if N/I is not an integer, we pad zero columns to the data matrix $\mathbf{S}$ so that all the agents own equal-size partitions $\mathbf{S}_i$'s, thus $n_i = \lceil N/I \rceil$ in both problems (6) and (7).

Network #	I	n_c	p_1	p_2
N4	10	2	0.9	0.3
N5	10	2	0.2	0.1
N6	50	5	0.9	0.3
N7	50	5	0.2	0.1
N8	250	10	0.9	0.3
N9	250	10	0.2	0.1

Table 5: Network setting for the NNSC and Sparse PCA problems.

5.3.1. Non-negative Sparse Coding

Algorithms and tuning: We test the Plain D^{4}L, with $\tilde{f}_i$ and $\tilde{h}_i$ chosen according to (15) and (16), respectively. The other parameters of the algorithm are set to: $\gamma^\nu = \gamma^{\nu-1}(1 - \epsilon\gamma^{\nu-1})$, with $\gamma^0 = 0.2$ and $\epsilon = 10^{-2}$; and $\tau^\nu_{D,i} = 10$ and $\tau^\nu_{X,i} = \max(L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}), 10)$. We compare the proposed scheme with a modified version of ATC, equipped with our new consensus protocol, implementable on directed networks. All the distributed algorithm are initialized setting $\mathbf{X}^0_i = \mathbf{0}$ and $\mathbf{D}^0_{(i)}$ equal to some randomly chosen columns of $\mathbf{S}_i$. Both Plain D^{4}L and Modified ATC call for solving a LASSO problem in updating the private variables (cf. Sec. 3.1); the update of the dictionary has instead a closed form expression, see (22). For both Plain D^{4}L and Modified ATC, the LASSO subproblems at iteration ν are solved using the projected (sub)gradient algorithm with diminishing step-size $\gamma^r = \gamma^{r-1}(1 - \epsilon\gamma^{r-1})$, where $\gamma^0 = 0.9$, $\epsilon = 10^{-3}$, and r denoting the inner iteration index. We terminate the projected subgradient algorithm in the inner loop when $J^r_{X,i} \triangleq \|\widehat{\mathbf{X}}^{\nu,r}_i - \mathbf{X}^{\nu,r}_i\|_{\infty,\infty} \leq 10^{-4}$, where

$$\widehat{\mathbf{X}}^{\nu,r}_i \triangleq \underset{\mathbf{X}_i\in\mathcal{X}_i}{\operatorname{argmin}} \left\langle \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}^{\nu,r}_i) + \tau^\nu_{X,i}\left(\mathbf{X}^{\nu,r}_i - \mathbf{X}^\nu_i\right), \mathbf{X}_i - \mathbf{X}^{\nu,r}_i \right\rangle + \frac{1}{2}\|\mathbf{X}_i - \mathbf{X}^{\nu,r}_i\|^2 + g_i(\mathbf{X}_i),$$

and $\mathbf{X}_i^{\nu,r}$ denotes the value of $\mathbf{X}_i$ at the ν-th outer and r-th inner iteration. In all our simulations, the above accuracy was reached within 30 (inner) iterations of the projected subgradient algorithm.

Convergence speed and quality of the reconstruction: We run the Plain D^{4}L and Modified ATC algorithms over different network settings, as listed in Table 5, and we terminated them after 1500 message exchanges. We replicated the tests for 5 independent realizations and we reported the *average* of the final values of the objective function, the consensus disagreement, and the distance from stationarity in Table 6 and Table 7, for the MIT-CBCL and VOC 2006 datasets, respectively. In Fig. 10 and Fig. 11 (for MIT-CBCL and VOC 2006 datasets, respectively), we plot the *average* value (over the aforementioned 5 graph realizations) of the objective function [subplot on the left], the consensus disagreement e^ν [subplot in the center], and the distance from stationarity Δ^ν [subplot on the right], versus number of message exchanges, for the two extreme network scenarios N4 [subplot (a)] and N9 [subplot (b)]. These results show that the proposed Plain D^{4}L significantly outperforms the Modified ATC algorithm. They also show a remarkable stability of Plain D^{4}L with respect to the simulated network graphs, which is not observed for the Modified ATC, whose performance deteriorates significantly going from N4 to N9.

Network #	objective value	consensus disagreement	distance from stationarity
N4	169.8/171.9	9.17e-7/2.9e-4	4.7e-4/8.8e-2
N5	169.9/172.2	4.5e-6/6.7e-4	5.3e-4/9.8e-2
N6	169.8/177.2	2.4e-7/1.9e-4	5.1e-4/6.3e-2
N7	169.9/177.3	1.1e-6/6.3e-4	5.5e-4/6.1e-2
N8	169.8/191.0	2.1e-7/1.2e-4	5.1e-4/2.2e-2
N9	169.8/190.9	5.8e-7/3.1e-4	6.6e-4/1.1e-2

Table 6: NNSC problem (MIT-CBCL dataset)–Plain D^{4}L/Modified ATC algorithms: objective value, consensus disagreement, and distance from stationarity obtained after 1500 message exchanges.

Network #	objective value	consensus disagreement	distance from stationarity
N4	845.2/848.8	2.9e-6/1.3e-4	1.1e-3/4.1e-3
N5	845.9/850.1	1.5e-5/5.3e-4	1.5e-3/6.1e-3
N6	844.8/879.5	5.7e-7/2.2e-4	1.3e-3/4.4e-2
N7	844.5/879.3	1.9e-6/1.2e-3	1.3e-3/3.9e-2
N8	844.8/941.2	5.6e-7/1.5e-4	1.6e-3/4.8e-2
N9	845.0/941.8	1.5e-6/3.6e-4	1.1e-3/5.1e-2

Table 7: NNSC problem (VOC 2006 dataset)–Plain D^{4}L/Modified ATC algorithms: objective value, consensus disagreement, and distance from stationarity obtained after 1500 message exchanges.

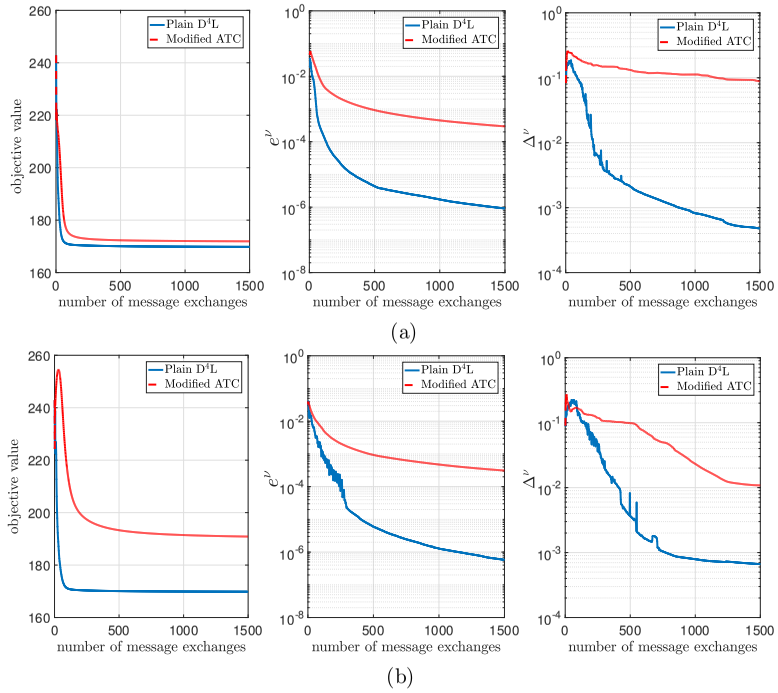


Figure 10: NNSC problem (MIT-CBCL dataset)–Plain D^{4}L and Modified ATC algorithms: objective value [subplots on the left], consensus disagreement [subplots in the center], and distance from stationarity Δ^ν [cf. (26)] [subplots on the right] vs. number of message exchanges. Comparison over the network settings N4 [subplots (a)] and N9 [subplots (b)] (cf. Table 5).

5.3.2. Sparse Principal Component Analysis (SPCA)

Algorithms and tuning: We test the same D^{4}L version as used in the Biclustering experiments (cf. Subsec. 5.2), i.e., $\tilde{f}_i$ and $\tilde{h}_i$ are chosen according to (14) and (16), respectively; we set $\gamma^\nu = \gamma^{\nu-1}(1-\epsilon\gamma^{\nu-1})$, with $\gamma^0 = 0.2$ and $\epsilon = 10^{-2}$; and $\tau_{D,i}^\nu = 10$ and $\tau_{X,i}^\nu = \max(L_{\nabla X_i}(\mathbf{U}_{(i)}^\nu), 10)$. We term it Plain D^{4}L. We compare Plain D^{4}L with a modified version of the ATC algorithm, which has been adapted to solve (6) and equipped with our new consensus protocol to handle directed network topologies. All the distributed algorithm are initialized, setting $\mathbf{X}_i^0 = \mathbf{0}$ and $\mathbf{D}_{(i)}^0$ equal to some randomly chosen columns of $\mathbf{S}_i$. The subproblems (8) and (11) at iteration ν are solved using the projected (sub)gradient algorithm; the setting is the same as that used in the Biclustering problem (cf. Subsec. 5.2). We terminate the projected subgradient algorithm in the inner loop when $J_{D,i}^r \leq 10^{-4}$ (in

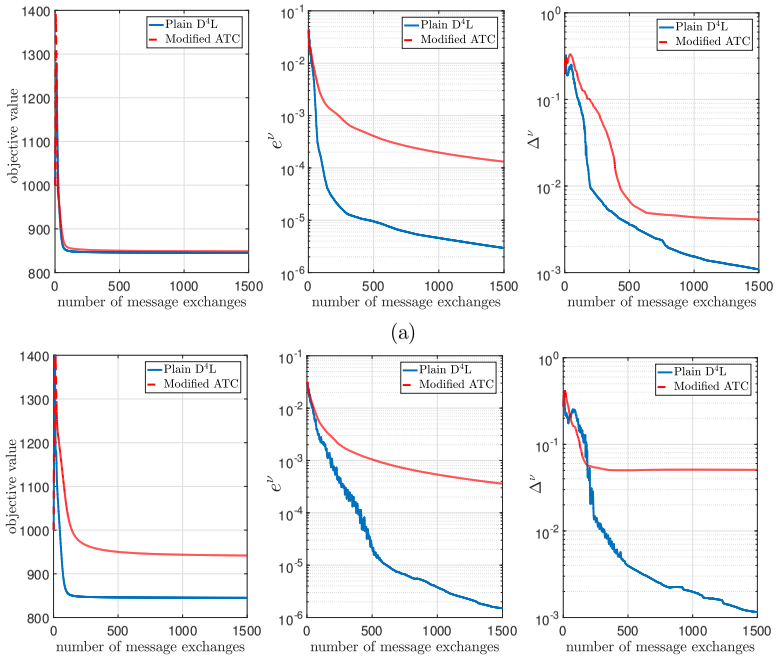


Figure 11: NNSC problem (VOC 2006 dataset)–Plain D^{4}L and Modified ATC algorithms: objective value [subplots on the left], consensus disagreement [subplots in the center], and distance from stationarity Δ^ν [cf. (26)] [subplots on the right] vs. number of message exchanges. Comparison over the network settings N4 [subplots (a)] and N9 [subplots (b)] (cf. Table 5).

solving subproblem (8)) and $J^r_{X,i} \leq 10^{-4}$ (in solving subproblem (11)), where $J^r_{D,i}$ and $J^r_{X,i}$ are defined as those in Subsec. 5.2. In all our simulations, the above accuracy was reached within 30 (inner) iterations of the projected subgradient algorithm.

Convergence speed and quality of the reconstruction: We test the Plain D^{4}L and the Modified ATC in different network settings, as listed in Table 5. The setting of the experiments and the averaging procedure of the reported values is the same of those used for the NNSC problem. The results of our experiments are reported in Table 8 and Figure 12 for the MIT-CBCL dataset; and in Table 9 and Figure 13 for the VOC 2006 dataset. The behaviors or the algorithms are very similar to those described in the NNSC case and confirm all previous observations.

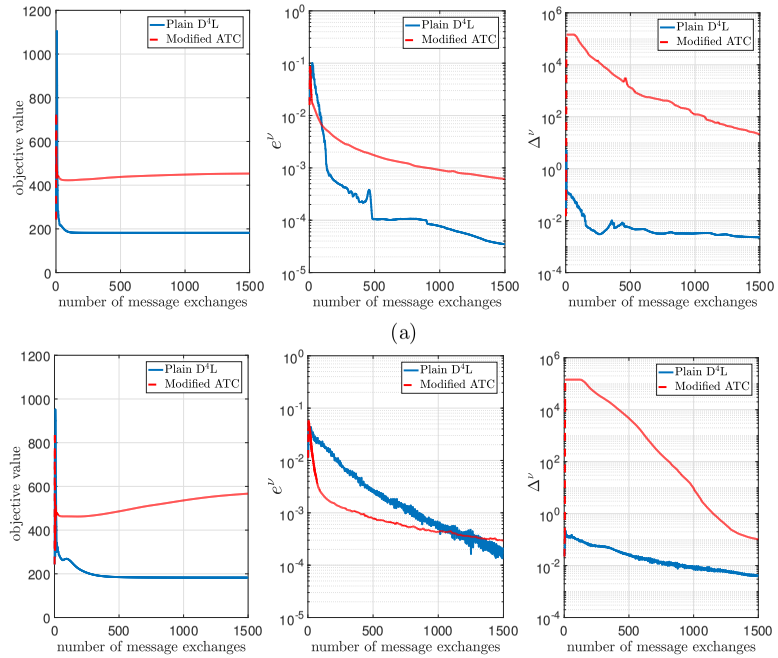


Figure 12: SPCA problem (MIT-CBCL dataset)–Plain D^{4}L and Modified ATC algorithms: objective value [subplots on the left], consensus disagreement [subplots in the center], and distance from stationarity Δ^ν [cf. (26)] [subplots on the right] vs. number of message exchanges. Comparison over the network settings N4 [subplots (a)] and N9 [subplots (b)] (cf. Table 5).

Network #	objective value	consensus disagreement	distance from stationarity
N4	181.9/453.1	3.5e-5/6.1e-4	2.2e-3/21.39
N5	185.1/446.4	2.6e-5/1.5e-3	1.3e-3/136.2
N6	182.7/517.3	5.2e-6/4.3e-4	1.3e-3/1.2e-1
N7	186.3/512.0	2.3e-5/9.2e-4	1.4e-3/1.18
N8	181.9/566.0	4.0e-5/1.7e-4	2.6e-3/1.5e-1
N9	182.6/566.9	1.7e-4/2.9e-4	4.2e-3/1.0e-1

Table 8: SPCA problem (MIT-CBCL dataset)–Plain D^{4}L/Modified ATC algorithms: objective value, consensus disagreement, and distance from stationarity obtained after 1500 message exchanges.

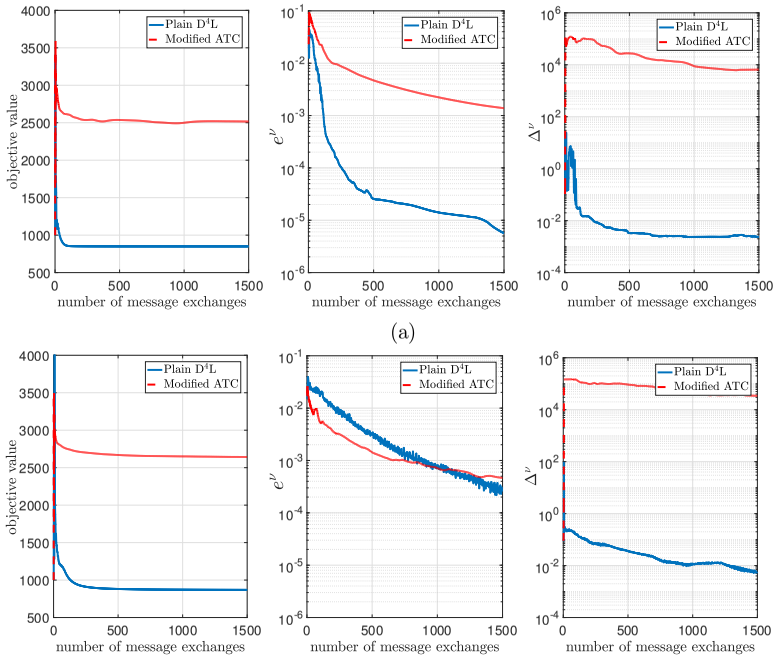


Figure 13: SPCA problem (VOC 2006 dataset)–Plain D^{4}L and Modified ATC algorithms: objective value [subplots on the left], consensus disagreement [subplots in the center], and distance from stationarity Δ^ν [cf. (26)] [subplots on the right] vs. number of message exchanges. Comparison over the network settings N4 [subplots (a)] and N9 [subplots (b)] (cf. Table 5).

Network #	objective value	consensus disagreement	distance from stationarity
N4	849.3/2516.0	5.7e-6/1.3e-3	2.3e-3/6532
N5	866.0/2533.6	1.8e-5/3.5e-3	2.0e-3/1.3e+4
N6	864.8/2621.0	7.2e-6/4.3e-4	3.4e-3/1.7e+4
N7	859.1/2624.1	5.3e-5/1.1e-3	2.4e-3/1.6e+4
N8	872.7/2643.2	6.6e-5/2.3e-4	1.1e-2/2.9e+4
N9	869.1/2642.0	2.3e-4/4.6e-4	5.6e-3/3.4e+4

Table 9: SPCA problem (VOC 2006 dataset)–Plain D^{4}L/Modified ATC algorithms: objective value, consensus disagreement, and distance from stationarity obtained after 1500 message exchanges.

6. Conclusions

This paper studied a fairly general class of distributed dictionary learning problems over time-varying multi-agent networks, with arbitrary topologies. We proposed the first decentralized algorithmic framework−the D^{4}L Algorithm−with provable convergence for this class of problems. Numerical experiments showed promising performance of our scheme with respect to state-of-the-art distributed methods.

Acknowledgments

The work of Daneshmand, Sun, and Scutari has been supported by the the National Science Foundation (NSF grants CIF 1564044 and 1632599) and the Army Research Office (Grant W911NF1810238). The work of Facchinei has been partially supported by the Italian Ministry of Education, Research and University, under the PLATINO (PLATform for INnOvative services in future internet) PON project (Grant Agreement no. PON01_01007).

Appendix A. Appendix

In this section we prove the major results of the paper, Theorems 2 and 3. We begin rewriting the D^{4}L Algorithm in an equivalent vector-matrix form (cf. Sec. A.1), which is more suitable for the analysis. Theorem 2 and Theorem 3 are then proved in Sec. A.2 and Sec. A.3, respectively. Some miscellanea results supporting the main proofs are collected in Appendix A.4. Table 10 below summarizes the symbols appearing in the proofs.

Symbol	Definition	Member of	Reference
$\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}$	$(1/I)\sum_{i=1}^I \phi_i^\nu \mathbf{D}_{(i)}^\nu$	$\mathcal{D}\subseteq\mathbb{R}^{M\times K}$	(41)
$\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu}$	$(1/I)\sum_{i=1}^I \phi_i^\nu \mathbf{U}_{(i)}^\nu$	$\mathcal{D}\subseteq\mathbb{R}^{M\times K}$	(41)
$\widetilde{\mathbf{D}}^\nu$	$[\widetilde{\mathbf{D}}_{(1)}^{\nu\intercal},\widetilde{\mathbf{D}}_{(2)}^{\nu\intercal},\dots,\widetilde{\mathbf{D}}_{(I)}^{\nu\intercal}]^\intercal$	$\mathbb{R}^{M\times KI}$	(34)
$\mathbf{U}^\nu$	$[\mathbf{U}_{(1)}^{\nu\intercal},\mathbf{U}_{(2)}^{\nu\intercal},\dots,\mathbf{U}_{(I)}^{\nu\intercal}]^\intercal$	$\mathbb{R}^{M\times KI}$	(34)
$\boldsymbol{\Theta}^\nu$	$[\boldsymbol{\Theta}_{(1)}^{\nu\intercal},\boldsymbol{\Theta}_{(2)}^{\nu\intercal},\dots,\boldsymbol{\Theta}_{(I)}^{\nu\intercal}]^\intercal$	$\mathbb{R}^{M\times KI}$	(34)
$\mathbf{G}^\nu$	$[\nabla_D f_1(\mathbf{D}_{(1)}^\nu,\mathbf{X}_1^\nu)^\intercal,\ \dots\ ,\ \nabla_D f_I(\mathbf{D}_{(I)}^\nu,\mathbf{X}_I^\nu)^\intercal]^\intercal$	$\mathbb{R}^{M\times KI}$	(34)
$\mathbf{W}^\nu$	$(w_{ij}^\nu)_{i,j=1}^I=(a_{ij}^\nu\ \phi_j^\nu/\phi_i^{\nu+1})_{i,j=1}^I=(\boldsymbol{\Phi}^{\nu+1})^{-1}\mathbf{A}^\nu\boldsymbol{\Phi}^\nu$	$\mathbb{R}^{I\times I}$	(35)
$\mathbf{W}^{\nu:l}$	$\mathbf{W}^\nu\cdot\mathbf{W}^{\nu-1}\cdots\mathbf{W}^l,\quad \nu>l$	$\mathbb{R}^{I\times I}$	(44)
$\widehat{\mathbf{W}}^\nu$	$\mathbf{W}^\nu\otimes\mathbf{I}_M$	$\mathbb{R}^{MI\times MI}$	(36)
$\widehat{\mathbf{W}}^{\nu:l}$	$\mathbf{W}^{\nu:l}\otimes\mathbf{I}_M,\quad \nu>l$	$\mathbb{R}^{MI\times MI}$	(44)
$\boldsymbol{\phi}^\nu$	$[\phi_1^\nu,\phi_2^\nu,\dots,\phi_I^\nu]^\intercal$	$\mathbb{R}^I$	(34)
$\boldsymbol{\Phi}^\nu$	$\mathrm{diag}\,(\boldsymbol{\phi}^\nu)$	$\mathbb{R}^{I\times I}$	(34)
$\widehat{\boldsymbol{\Phi}}^\nu$	$\boldsymbol{\Phi}^\nu\otimes\mathbf{I}_M$	$\mathbb{R}^{MI\times MI}$	(36)
$\mathbf{J}_{\boldsymbol{\phi}^\nu}$	$(1/I)\mathbf{1}\boldsymbol{\phi}^{\nu\intercal}$	$\mathbb{R}^{I\times I}$	(45)
$\widehat{\mathbf{J}}_{\boldsymbol{\phi}^\nu}$	$\mathbf{J}_{\boldsymbol{\phi}^\nu}\otimes\mathbf{I}_M$	$\mathbb{R}^{MI\times MI}$	(45)

Table 10: Table of notation (appendix)

A.1. D^{4}L in vector-matrix form

We rewrite Algorithm 1 in a more convenient vector-matrix form. To do so, we introduce the following notation. Recalling the definitions of $\widetilde{\mathbf{D}}^{\nu}_{(i)}$ [cf. (8)], $\mathbf{U}^{\nu}_{(i)}$ [cf. (10)], ϕ^{ν}_i [cf. (12)], and $\mathbf{\Theta}^{\nu}_{(i)}$ [cf. (13)], define the corresponding aggregate quantities

$$\begin{aligned}
&\widetilde{\mathbf{D}}^{\nu} \triangleq [\widetilde{\mathbf{D}}^{\nu\intercal}_{(1)}, \widetilde{\mathbf{D}}^{\nu\intercal}_{(2)}, \dots, \widetilde{\mathbf{D}}^{\nu\intercal}_{(I)}]^{\intercal},\\
&\mathbf{U}^{\nu} \triangleq [\mathbf{U}^{\nu\intercal}_{(1)}, \mathbf{U}^{\nu\intercal}_{(2)}, \dots, \mathbf{U}^{\nu\intercal}_{(I)}]^{\intercal},\\
&\boldsymbol{\phi}^{\nu} \triangleq [\phi^{\nu}_1, \phi^{\nu}_2, \dots, \phi^{\nu}_I]^{\intercal}, \quad \mathbf{\Phi}^{\nu} \triangleq \operatorname{diag}(\boldsymbol{\phi}^{\nu}),\\
&\mathbf{\Theta}^{\nu} \triangleq [\mathbf{\Theta}^{\nu\intercal}_{(1)}, \mathbf{\Theta}^{\nu\intercal}_{(2)}, \dots, \mathbf{\Theta}^{\nu\intercal}_{(I)}]^{\intercal},\\
&\mathbf{G}^{\nu} \triangleq \left[\nabla_D f_1(\mathbf{D}^{\nu}_{(1)}, \mathbf{X}^{\nu}_1)^{\intercal},\ \nabla_D f_2(\mathbf{D}^{\nu}_{(2)}, \mathbf{X}^{\nu}_2)^{\intercal},\ \dots\ ,\ \nabla_D f_I(\mathbf{D}^{\nu}_{(I)}, \mathbf{X}^{\nu}_I)^{\intercal}\right]^{\intercal},
\end{aligned} \tag{34}$$

where $\operatorname{diag}(\mathbf{x})$ is a diagonal matrix whose diagonal entries are the components of the vector $\mathbf{x}$. Combining the weights a^{ν}_{ij} and the variables ϕ^{ν}_i in the update of $\mathbf{D}^{\nu}_{(i)}$ [cf. (12)] in the single coefficient

$$w^{\nu}_{ij} \triangleq \frac{a^{\nu}_{ij}\phi^{\nu}_j}{\sum_k a^{\nu}_{ik}\phi^{\nu}_k},$$

we define the weight matrix $\mathbf{W}^{\nu}$, whose entries are $[\mathbf{W}^{\nu}]_{i,j} = w^{\nu}_{ij}$. Note that $\mathbf{W}^{\nu}$ has the same zero-pattern of $\mathbf{A}^{\nu}$, and the following properties hold (the latter under Assumption F)

$$\mathbf{W}^{\nu} = \left(\mathbf{\Phi}^{\nu+1}\right)^{-1}\mathbf{A}^{\nu}\mathbf{\Phi}^{\nu}, \quad \text{and} \quad \mathbf{W}^{\nu}\mathbf{1} = \mathbf{1}. \tag{35}$$

Finally, we define the following "augmented" matrices

$$\widehat{\mathbf{W}}^{\nu} \triangleq \mathbf{W}^{\nu} \otimes \mathbf{I}_M \quad \text{and} \quad \widehat{\mathbf{\Phi}}^{\nu} \triangleq \mathbf{\Phi}^{\nu} \otimes \mathbf{I}_M, \tag{36}$$

where $\mathbf{I}_M$ is the M-by-M identity matrix.

Using the above notation, the main iterates of the D^{4}L Algorithm, i.e., (10), (12), and (13), can be rewritten in compact form as

$$\mathbf{U}^{\nu} = \mathbf{D}^{\nu} + \gamma^{\nu}(\widetilde{\mathbf{D}}^{\nu} - \mathbf{D}^{\nu}), \tag{37}$$

$$\boldsymbol{\phi}^{\nu+1} = \mathbf{A}^{\nu}\boldsymbol{\phi}^{\nu}, \tag{38}$$

$$\mathbf{D}^{\nu+1} = \widehat{\mathbf{W}}^{\nu}\mathbf{U}^{\nu}, \tag{39}$$

$$\mathbf{\Theta}^{\nu+1} = \widehat{\mathbf{W}}^{\nu}\mathbf{\Theta}^{\nu} + \left(\widehat{\mathbf{\Phi}}^{\nu+1}\right)^{-1}\left(\mathbf{G}^{\nu+1} - \mathbf{G}^{\nu}\right). \tag{40}$$

Instrumental to the analysis of the consensus disagreement are the following weighted average quantities:

$$\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}} \triangleq \frac{1}{I}\sum_{i=1}^{I}\phi^{\nu}_i\,\mathbf{D}^{\nu}_{(i)}, \quad \overline{\mathbf{U}}_{\boldsymbol{\phi}^{\nu}} \triangleq \frac{1}{I}\sum_{i=1}^{I}\phi^{\nu}_i\,\mathbf{U}^{\nu}_{(i)}. \tag{41}$$

Using (12), (41) and the column stochasticity of $\mathbf{A}^{\nu}$ (cf. Assumption F3), it is not difficult to check that

$$\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}} = \overline{\mathbf{U}}_{\boldsymbol{\phi}^{\nu}}, \tag{42}$$

which, together with (10), leads to the following dynamics for $\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}$:

$$\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}} = \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}} + \frac{\gamma^{\nu}}{I}\sum_{i=1}^{I}\phi^{\nu}_i\left(\widetilde{\mathbf{D}}^{\nu}_{(i)} - \mathbf{D}^{\nu}_{(i)}\right). \tag{43}$$

A.2. Proof of Theorem 2

We prove Theorem 2 in the following order (consistent with the statements in the theorem):

Step 1–Asymptotic consensus & related properties: We prove that consensus is asymptotically achieved, that is, $\lim_{\nu\to\infty} e^{\nu} = 0$ [statement (a)], along with some properties on related quantities, which will be used in the other steps–see Sec. A.2.1;

Step 2–Boundedness of the iterates and Lipschitz continuity of ∇f_i: We prove that the sequence $\left\{\left(\mathbf{D}^{\nu}, \mathbf{X}^{\nu}\right)\right\}_{\nu}$ generated by the algorithm is bounded [statement (b-i)], and, as a consequence, ∇f_i, $\nabla_D \tilde{f}_i$, and $\nabla_{X_i}\tilde{h}_i$ are Lipschitz continuous on a (suitably defined) compact set containing $\left\{\left(\mathbf{D}^{\nu}, \mathbf{X}^{\nu}\right)\right\}_{\nu}$ [Remarks 8 and 9]–see Sec. A.2.2;

Step 3–Decrease of $\{U(\overline{\mathbf{D}}^{\nu}, \mathbf{X}^{\nu})\}_{\nu}$: By leveraging the Lipschitz continuity of the gradients as in Step 2, we study the decrease properties of $\{U(\overline{\mathbf{D}}^{\nu}, \mathbf{X}^{\nu})\}_{\nu}$ [statement (b-ii)]–see Sec. A.2.3;

Step 4–Vanishing X-stationarity: We prove that $\lim_{\nu\to\infty}\Delta_X(\overline{\mathbf{D}}^{\nu}, \mathbf{X}^{\nu}) = 0$ [statement (b-iii)]–see Sec. A.2.4;

Step 5–Vanishing liminf D-stationarity: We prove $\liminf_{\nu\to\infty}\Delta_D(\overline{\mathbf{D}}^{\nu}, \mathbf{X}^{\nu}) = 0$ [statement (b-iv)]–see Sec. A.2.5;

Step 6–Vanishing D-stationarity: Finally, we prove $\lim_{\nu\to\infty}\Delta_D(\overline{\mathbf{D}}^{\nu}, \mathbf{X}^{\nu}) = 0$ [statement (b')]–see Sec. A.2.6.

A.2.1. Step 1–Asymptotic consensus and related properties

1) Preliminaries: To analyze the dynamics of the consensus disagreement, we first introduce the following product matrices and their augmented counterparts: Given $\mathbf{W}^{\nu}$ [cf. (35)], and $\nu, l \in \mathbb{N}_+$, let

$$\mathbf{W}^{\nu:l} \triangleq \begin{cases} \mathbf{W}^{\nu}\cdot\mathbf{W}^{\nu-1}\cdots\mathbf{W}^{l}, & \nu > l,\\ \mathbf{W}^{\nu}, & \nu = l,\\ \mathbf{0}_I, & \nu < l, \end{cases} \quad \text{and} \quad \widehat{\mathbf{W}}^{\nu:l} \triangleq \mathbf{W}^{\nu:l}\otimes\mathbf{I}_M. \tag{44}$$

Define also the weight-average matrices

$$\mathbf{J}_{\boldsymbol{\phi}^{\nu}} \triangleq \frac{1}{I}\mathbf{1}\boldsymbol{\phi}^{\nu\intercal}, \quad \widehat{\mathbf{J}}_{\boldsymbol{\phi}^{\nu}} \triangleq \mathbf{J}_{\boldsymbol{\phi}^{\nu}}\otimes\mathbf{I}_M. \tag{45}$$

For notational simplicity, when $\boldsymbol{\phi}^{\nu} = \mathbf{1}$, we use $\mathbf{J}$ instead of $\mathbf{J}_{\boldsymbol{\phi}^{\nu}}$ and $\widehat{\mathbf{J}} \triangleq \mathbf{J}\otimes\mathbf{I}_M$. Using (35) and (41), it is not difficult to check that the following hold:

$$\widehat{\mathbf{J}}_{\boldsymbol{\phi}^{\nu}}\mathbf{D}^{\nu} = \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \tag{46}$$

$$\widehat{\mathbf{J}}_{\boldsymbol{\phi}^{\nu+1}}\widehat{\mathbf{W}}^{\nu:l} = \widehat{\mathbf{J}}\,\widehat{\mathbf{\Phi}}^{l} = \widehat{\mathbf{J}}_{\boldsymbol{\phi}^{l}}. \tag{47}$$

The dynamics of the consensus disagreement e^ν boils down to studying the decay of $\|\mathbf{W}^{\nu:l} - \mathbf{J}_{\boldsymbol{\phi}^l}\|_2$ (this will be clear in the proof of Proposition 5 below). The following lemma shows that $\mathbf{W}^{\nu:l}$ converges geometrically to $\mathbf{J}_{\boldsymbol{\phi}^l}$, as $\nu \to \infty$.

Lemma 4 (Scutari and Sun (2018)-Lemma 4.13, Ch. 3.4.2.5) *Let $\{\mathcal{G}^\nu\}_\nu$ be a sequence of digraphs satisfying Assumption B; let $\{\mathbf{A}^\nu\}_\nu$ be a sequence of matrices satisfying Assumption F; and let $\{\mathbf{W}^\nu\}_\nu$ be the sequence of matrices defined in* (35)*. Then, there holds*

$$\left\|\mathbf{W}^{\nu:l} - \mathbf{J}_{\boldsymbol{\phi}^l}\right\|_2 \le c_W\,(\rho)^{\nu-l+1}, \quad \forall \nu \ge l,\ \nu, l \in \mathbb{N}_+,$$

where $c_W > 0$ is a (proper) constant, and $\rho \in (0,1)$ is defined as

$$\rho = \left(1 - \tilde{\kappa}^{-(I-1)B}\right)^{\frac{1}{B(I-1)}} < 1, \tag{48}$$

with $\tilde{\kappa} = \kappa^{IB+1}/I$; and κ is defined in Assumption F.

Furthermore, the sequence $\{\phi_i^\nu\}_\nu$ satisfies

$$\underline{\epsilon}_\phi \triangleq \inf_{\nu\in\mathbb{N}_+}\left(\min_{1\le i\le I}\phi_i^\nu\right) \ge \kappa^{IB}, \quad \text{and} \quad \bar{\epsilon}_\phi \triangleq \sup_{\nu\in\mathbb{N}_+}\left(\max_{1\le i\le I}\phi_i^\nu\right) \le I-(I-1)\kappa^{IB}. \tag{49}$$

If all the matrices $\mathbf{A}^\nu$ are doubly-stochastic, then $\underline{\epsilon}_\phi = \bar{\epsilon}_\phi = 1$.

2) Proof of $\lim_{\nu\to\infty} e^\nu = 0$. Using (30), we can write

$$\begin{aligned} e^\nu &\overset{(a)}{\le} K\left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}^\nu\right\|_F \overset{(b)}{\le} K\left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F + K\sqrt{I}\left\|\overline{\mathbf{D}}^\nu - \overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F \\ &\overset{(c)}{\le} 2K\left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F, \end{aligned} \tag{50}$$

where (a) follows from the equivalence of norms; (b) is due to the triangle inequality; and in (c) we used $\sum_{i=1}^I a_i \le \sqrt{I}||\mathbf{a}||$, with $\mathbf{a} = (a_i)_{i=1}^I \in \mathbb{R}^I$.

The following proposition concludes the proof of statement (a), proving that $\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\|_F$ is square summable, along with some additional properties on related quantities.

Proposition 5 *In the above setting, there hold:*

$$\lim_{\nu\to\infty} ||\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}||_F = 0; \tag{51}$$

$$\lim_{\nu\to\infty}\sum_{t=1}^{\nu} ||\mathbf{D}^t - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^t}||_F^2 < \infty; \tag{52}$$

$$\lim_{\nu\to\infty}\sum_{t=1}^{\nu} ||\mathbf{U}^t - \mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^t}||_F^2 < \infty. \tag{53}$$

Proof To prove (51), let us first expand $\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}$ as follows: for any $\nu \ge 1$,

$$\begin{aligned} \mathbf{D}^\nu &\overset{(39)}{=} \widehat{\mathbf{W}}^{\nu-1}\mathbf{U}^{\nu-1} = \widehat{\mathbf{W}}^{\nu-1}\mathbf{D}^{\nu-1} + \widehat{\mathbf{W}}^{\nu-1}\left(\mathbf{U}^{\nu-1} - \mathbf{D}^{\nu-1}\right) \\ &= \widehat{\mathbf{W}}^{\nu-1:0}\mathbf{D}^0 + \sum_{t=0}^{\nu-1}\widehat{\mathbf{W}}^{\nu-1:t}\left(\mathbf{U}^t - \mathbf{D}^t\right), \end{aligned} \tag{54}$$

where the last equality follows from induction and the definition of $\widehat{\mathbf{W}}^{\nu:l}$ [cf. (44)]. Similarly, we expand the subtrahend as

$$\begin{aligned} \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu} &\overset{(46)}{=} \widehat{\mathbf{J}}_{\boldsymbol{\phi}^\nu}\mathbf{D}^\nu \overset{(54)}{=} \widehat{\mathbf{J}}_{\boldsymbol{\phi}^\nu}\left(\widehat{\mathbf{W}}^{\nu-1:0}\mathbf{D}^0 + \sum_{t=0}^{\nu-1}\widehat{\mathbf{W}}^{\nu-1:t}\left(\mathbf{U}^t - \mathbf{D}^t\right)\right) \\ &\overset{(47)}{=} \widehat{\mathbf{J}}\left(\mathbf{D}^0 + \sum_{t=0}^{\nu-1}\widehat{\boldsymbol{\Phi}}^t\left(\mathbf{U}^t - \mathbf{D}^t\right)\right). \end{aligned} \tag{55}$$

Subtracting (55) from (54) and using (37), yields

$$\begin{aligned} \left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F &\le \left\|\widehat{\mathbf{W}}^{\nu-1:0} - \widehat{\mathbf{J}}\right\|_2\left\|\mathbf{D}^0\right\|_F + \sum_{t=0}^{\nu-1}\gamma^t\left\|\widehat{\mathbf{W}}^{\nu-1:t} - \widehat{\mathbf{J}}_{\boldsymbol{\phi}^t}\right\|_2\left\|\widetilde{\mathbf{D}}^t - \mathbf{D}^t\right\|_F \\ &\overset{(a)}{\le} c_1(\rho)^\nu + c_2\sum_{t=0}^{\nu-1}\gamma^t\,(\rho)^{\nu-t} \xrightarrow[\nu\to\infty]{(b)} 0, \end{aligned} \tag{56}$$

for some finite constants $c_1, c_2 > 0$, where (a) is due to Lemma 4 and the boundedness of $\{||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||\}_\nu$; and (b) follows from Lemma 13(a) in Appendix A.4.

Let us now proceed to prove (52). Using (56), we have

$$\begin{aligned} &\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\left\|\mathbf{D}^t - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^t}\right\|_F^2 \le \lim_{\nu\to\infty}\sum_{t=1}^{\nu}\left(c_1(\rho)^t + c_2\sum_{l=0}^{t-1}\gamma^l\,(\rho)^{t-l}\right)^2 \\ &\overset{(a)}{\le} \frac{2c_1^2}{1-(\rho)^2} + 2\,c_2^2\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\sum_{l=0}^{t-1}\sum_{k=0}^{t-1}\gamma^l\gamma^k(\rho)^{t-k}(\rho)^{t-l} \\ &\overset{(b)}{\le} \frac{2c_1^2}{1-(\rho)^2} + c_2^2\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\sum_{l=0}^{t-1}(\gamma^l)^2(\rho)^{t-l}\sum_{k=0}^{t-1}(\rho)^{t-k} + c_2^2\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\sum_{k=0}^{t-1}(\gamma^k)^2(\rho)^{t-k}\sum_{l=0}^{t-1}(\rho)^{t-l} \\ &\le \frac{2c_1^2}{1-(\rho)^2} + \frac{2c_2^2}{1-\rho}\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\sum_{l=0}^{t-1}(\gamma^l)^2(\rho)^{t-l} \overset{(c)}{<} \infty, \end{aligned}$$

where in (a) and (b) we used $(a+b)^2 \le 2(a^2+b^2)$ and $ab \le (a^2+b^2)/2$, respectively, and (c) is due to Lemma 13(b) (cf. Appendix A.4).

We prove now (53). Using (41) and (37), we get

$$\begin{aligned} \left\|\mathbf{U}^t - \mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^t}\right\|_F^2 &= \left\|\gamma^t\left(\widetilde{\mathbf{D}}^t - \mathbf{1}\otimes\frac{1}{I}\sum_{i=1}^I\phi_i^t\widetilde{\mathbf{D}}_{(i)}^t\right) + \left(1-\gamma^t\right)\left(\mathbf{D}^t - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^t}\right)\right\|_F^2 \\ &\le 2\left(\gamma^t\right)^2\left\|\widetilde{\mathbf{D}}^t - \mathbf{1}\otimes\frac{1}{I}\sum_{i=1}^I\phi_i^t\widetilde{\mathbf{D}}_{(i)}^t\right\|_F^2 + 2\left\|\mathbf{D}^t - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^t}\right\|_F^2, \end{aligned} \tag{57}$$

where in the last inequality we used Jensen's inequality and $(1-\gamma^t) \le 1$. Therefore,

$$\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\left\|\mathbf{U}^t - \mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^t}\right\|_F^2 \le \lim_{\nu\to\infty} 2\,c_3\sum_{t=1}^{\nu}\left(\gamma^t\right)^2 + \lim_{\nu\to\infty} 2\sum_{t=1}^{\nu}\left\|\mathbf{D}^t - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^t}\right\|_F^2 \overset{(a)}{<} \infty,$$

where c_3 is a positive finite constant, and (a) follows from Assumption E and (52). ■

Remark 6 (On $L_{\nabla X_i}(\overline{\mathbf{U}}_{\phi^\nu})$ and $L_{\nabla X_i}(\mathbf{U}^\nu_{(i)})$) *Recall that $\nabla_{X_i} f_i(\overline{\mathbf{U}}_{\phi^\nu}, \bullet)$ is Lipschitz continuous on $\mathcal{X}_i$, with constant $L_{\nabla X_i}(\overline{\mathbf{U}}_{\phi^\nu})$ (cf. Assumption A2). Since $||\mathbf{U}^\nu_{(i)} - \overline{\mathbf{U}}_{\phi^\nu}||_F \underset{\nu\to\infty}{\longrightarrow} 0$ [cf. (53), Proposition 5] and $L_{\nabla X_i}(\mathbf{D})$ is continuous, we have*

$$\left| L_{\nabla X_i}(\overline{\mathbf{U}}_{\phi^\nu}) - L_{\nabla X_i}(\mathbf{U}^\nu_{(i)}) \right| \underset{\nu\to\infty}{\longrightarrow} 0, \quad i = 1, 2, \dots I. \tag{58}$$

A.2.2. Step 2–Boundedness of the iterates

We show that the sequence $\left\{\left(\mathbf{D}^\nu, \mathbf{X}^\nu\right)\right\}_\nu$ generated by the D^4L Algorithm is bounded [statement (b-i)]. We prove the result only for $\tilde{h}_i$ given by (17); the proof can be easily tailored to the other choice of $\tilde{h}_i$. If the sets $\mathcal{X}_i$ are bounded [Assumption A5(i)], the result follows readily. Therefore, we consider next the setting under A5(ii).

Throughout the proof, we will use the following properties of $\tilde{f}_i$ and $\tilde{h}_i$.

Remark 7 *The surrogate functions $\tilde{f}_i$ and $\tilde{h}_i$ as in Assumption C have the following properties: for all $i = 1, 2, \dots, I$,*

(a) $\tilde{f}_i(\bullet; \mathbf{D}, \mathbf{X}_i)$ is strongly convex on $\mathcal{D}$, uniformly with respect to $(\mathbf{D}, \mathbf{X}_i) \in \mathcal{D} \times \mathcal{X}_i$, with constant $\tau^\nu_{D,i} > 0$; and $\nabla_D \tilde{f}_i(\mathbf{D}; \mathbf{D}, \mathbf{X}_i) = \nabla_D f_i(\mathbf{D}, \mathbf{X}_i)$, for all $(\mathbf{D}, \mathbf{X}_i) \in \mathcal{D} \times \mathcal{X}_i$.

(b) $\tilde{h}_i(\bullet; \mathbf{D}, \mathbf{X}_i)$ is strongly convex on $\mathcal{X}_i$, uniformly with respect to $(\mathbf{D}, \mathbf{X}_i) \in \mathcal{D} \times \mathcal{X}_i$, with constant $\tau^\nu_{X,i} > 0$; and $\nabla_{X_i} \tilde{h}_i(\mathbf{X}_i; \mathbf{D}, \mathbf{X}_i) = \nabla_{X_i} f_i(\mathbf{D}, \mathbf{X}_i)$, for all $(\mathbf{D}, \mathbf{X}_i) \in \mathcal{D} \times \mathcal{X}_i$.

By the optimality of $\mathbf{X}_i^{\nu+1}$ in (11), there exist $\mathbf{\Xi}_i^0 \in \partial_{X_i} g_i(\mathbf{X}_i^0)$ and $\mathbf{\Xi}_i^{\nu+1} \in \partial_{X_i} g_i(\mathbf{X}_i^{\nu+1})$ such that

$$\begin{aligned}
0 &\le \left\langle \nabla_{X_i} \tilde{h}_i(\mathbf{X}_i^{\nu+1}; \mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu) + \mathbf{\Xi}_i^{\nu+1}, \mathbf{X}_i^0 - \mathbf{X}_i^{\nu+1} \right\rangle \\
&= \left\langle \mathbf{\Xi}_i^{\nu+1} - \mathbf{\Xi}_i^0 + \tau^\nu_{X,i}(\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0), \mathbf{X}_i^0 - \mathbf{X}_i^{\nu+1} \right\rangle \\
&\quad + \left\langle \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu) - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0), \mathbf{X}_i^0 - \mathbf{X}_i^{\nu+1} \right\rangle \\
&\quad - \left\langle \tau^\nu_{X,i}(\mathbf{X}_i^\nu - \mathbf{X}_i^0), \mathbf{X}_i^0 - \mathbf{X}_i^{\nu+1} \right\rangle \\
&\quad + \left\langle \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0) + \mathbf{\Xi}_i^0, \mathbf{X}_i^0 - \mathbf{X}_i^{\nu+1} \right\rangle.
\end{aligned}$$

Using Remark 7(b) and the μ_i-strongly convexity of g_i's, we obtain

$$\begin{aligned}
(\tau^\nu_{X,i} + \mu_i) \left\| \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0 \right\|_F^2 &\le \left\langle \tau^\nu_{X,i} \mathbf{X}_i^\nu - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu), \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0 \right\rangle \\
&\quad - \left\langle \tau^\nu_{X,i} \mathbf{X}_i^0 - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0), \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0 \right\rangle \\
&\quad - \Bigg\langle \underbrace{\nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0) + \mathbf{\Xi}_i^0}_{\triangleq \mathbf{Z}_i^\nu \in \partial_{X_i} U(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0)}, \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0 \Bigg\rangle.
\end{aligned} \tag{59}$$

Define $\Upsilon_i^\nu(\mathbf{X}_i) \triangleq \tau^\nu_{X,i} \mathbf{X}_i - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i)$ and rewrite (59) as

$$(\tau^\nu_{X,i} + \mu_i) \left|| \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0 \right||_F \le \left\| \Upsilon_i^\nu(\mathbf{X}_i^\nu) - \Upsilon_i^\nu(\mathbf{X}_i^0) \right\|_F + \left\| \mathbf{Z}_i^\nu \right\|_F. \tag{60}$$

Since $\mathcal{D}$ is compact and $\mathbf{X}_i^0$ is given, we have $||\mathbf{Z}_i^\nu||_F \le B_Z$, for all i, $\nu \ge 1$, and some finite $B_Z > 0$. Let us bound next $||\Upsilon_i^\nu(\mathbf{X}_i^\nu) - \Upsilon_i^\nu(\mathbf{X}_i^0)||_F$. We write

$$\begin{aligned}
\left\| \Upsilon_i^\nu(\mathbf{X}_i^\nu) - \Upsilon_i^\nu(\mathbf{X}_i^0) \right\|_F^2 &= (\tau^\nu_{X,i})^2 \left\| \mathbf{X}_i^\nu - \mathbf{X}_i^0 \right\|_F^2 + \left\| \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu) - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0) \right\|_F^2 \\
&\quad - 2\tau^\nu_{X,i} \left\langle \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu) - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0), \mathbf{X}_i^\nu - \mathbf{X}_i^0 \right\rangle \\
&\overset{(a)}{\le} (\tau^\nu_{X,i})^2 \left\| \mathbf{X}_i^\nu - \mathbf{X}_i^0 \right\|_F^2 \\
&\quad + \left(1 - \frac{2\tau^\nu_{X,i}}{L_{\nabla X_i}(\mathbf{U}^\nu_{(i)})} \right) \left\| \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^\nu) - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i^0) \right\|_F^2,
\end{aligned} \tag{61}$$

where in (a) we used the $1/L_{\nabla X_i}(\mathbf{U}^\nu_{(i)})$-co-coercivity of $\nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \bullet)$ [due to the convexity of $f_i(\mathbf{U}^\nu_{(i)}, \bullet)$ and the $L_{\nabla X_i}(\mathbf{U}^\nu_{(i)})$-Lipschitianity of $\nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \bullet)$ (Rockafellar and Wets, 1998, Prop.12.60)], i.e.,

$$\begin{aligned}
&\left\langle \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i) - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{Y}_i), \mathbf{X}_i - \mathbf{Y}_i \right\rangle \ge \\
&\qquad \frac{1}{L_{\nabla X_i}(\mathbf{U}^\nu_{(i)})} \cdot \left\| \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{X}_i) - \nabla_{X_i} f_i(\mathbf{U}^\nu_{(i)}, \mathbf{Y}_i) \right\|_F^2, \qquad \forall \mathbf{X}_i, \mathbf{Y}_i \in \mathcal{X}_i.
\end{aligned}$$

Note that $||\Upsilon_i^\nu(\mathbf{X}_i^\nu) - \Upsilon_i^\nu(\mathbf{X}_i^0)||_F \le \tau^\nu_{X,i} ||\mathbf{X}_i^\nu - \mathbf{X}_i^0||_F$ as long as $\tau^\nu_{X,i} \ge \frac{1}{2} L_{\nabla X_i}(\mathbf{U}^\nu_{(i)})$, which is satisfied under D1. Therefore, we can bound (60) as

$$(\tau^\nu_{X,i} + \mu_i) ||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0||_F \le \tau^\nu_{X,i} \left\| \mathbf{X}_i^\nu - \mathbf{X}_i^0 \right\|_F + B_Z. \tag{62}$$

We can now prove that, starting from $\mathbf{X}_i^0$, the iterates $\mathbf{X}_i^\nu$ stays in the ball $\mathcal{B}_i(R_i, \mathbf{X}_i^0) \triangleq \{\mathbf{X}_i \in \mathbb{R}^{K \times n_i} : \|\mathbf{X}_i - \mathbf{X}_i^0\|_F \le R_i\}$, for all $\nu \ge 1$, where $R_i \ge B_Z/\mu_i$. Let us prove it by induction. Evidently $\mathbf{X}_i^0 \in \mathcal{B}_i(R_i, \mathbf{X}_i^0)$. Let $\mathbf{X}_i^\nu \in \mathcal{B}_i(R_i, \mathbf{X}_i^0)$; by (62), we get

$$||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^0||_F \le \frac{\tau^\nu_{X,i}}{\tau^\nu_{X,i} + \mu_i} ||\mathbf{X}_i^\nu - \mathbf{X}_i^0||_F + \frac{B_Z}{\tau^\nu_{X,i} + \mu_i} \le R_i,$$

where the second inequality is due to $\mathbf{X}_i^\nu \in \mathcal{B}_i(R, \mathbf{X}_i^0)$ and $R \ge B_Z/\mu$. Hence $\mathbf{X}_i^{\nu+1} \in \mathcal{B}_i(R_i, \mathbf{X}_i^0)$. Therefore, $\mathbf{X}_i^\nu \in \mathcal{B}_i(R_i, \mathbf{X}_i^0)$, for all $\nu \ge 0$. Since $\mathcal{D}$ is bounded (cf. Assumption A3), it follows that $(\mathbf{D}^\nu_{(i)}, \mathbf{X}_i^\nu) \in \mathcal{D} \times \mathcal{B}_i(R_i, \mathbf{X}_i^0)$, for all $\nu \ge 0$. □

Remark 8 (On the Lipschitz continuity of ∇f_i's) *Since f_i is $\mathcal{C}^2$, a direct consequence of the boundedness of $\left\{\left(\mathbf{D}^\nu, \mathbf{X}^\nu\right)\right\}_\nu$ is that ∇f_i [the gradient of f_i with respect to $(\mathbf{D}, \mathbf{X}_i)$] is Lipschitz continuous on $\mathcal{D} \times \mathcal{B}_i(R_i, \mathbf{X}_i^0)$, that is, there exists some positive finite constant $L_{\nabla,i}$ such that*

$$||\nabla f_i(\mathbf{D}, \mathbf{X}_i) - \nabla f_i(\mathbf{D}', \mathbf{X}_i')||_F \le L_{\nabla,i} ||(\mathbf{D}, \mathbf{X}_i) - (\mathbf{D}', \mathbf{X}_i')||_F, \tag{63}$$

for all $(\mathbf{D}, \mathbf{X}_i), (\mathbf{D}', \mathbf{X}_i') \in \mathcal{D} \times \mathcal{B}_i(R_i, \mathbf{X}_i^0)$, and $i = 1, 2, \dots, I$. We define $L_\nabla \triangleq \max_i L_{\nabla,i}$.

The above result also implies that $\nabla_D F : \mathcal{D} \times (\mathcal{X}_1 \times \cdots \times \mathcal{X}_I) \to \mathcal{D}$ [cf. (P)] is Lipschitz continuous on $\mathcal{D} \times (\mathcal{B}_1(R_1, \mathbf{X}_1^0) \times \cdots \times \mathcal{B}_I(R_I, \mathbf{X}_I^0))$, with constant $L_{\nabla_D} \triangleq I \cdot L_\nabla$.

Remark 9 (On the Lipschitz continuity of $\nabla_D \tilde{f}_i$ and $\nabla_{X_i}\tilde{h}_i$) *$\nabla_D \tilde{f}_i : \mathcal{D} \times \mathbb{R}^{K\times n_i} \times \mathbb{R}^{K\times n_i} \to \mathbb{R}^{M\times K}$ and $\nabla_{X_i}\tilde{h}_i : \mathbb{R}^{K\times n_i} \times \mathcal{D} \times \mathbb{R}^{K\times n_i} \to \mathbb{R}^{K\times n_i}$ are Lipschitz continuous on $\mathcal{D}\times\mathcal{D}\times\mathcal{B}_i(R_i,\mathbf{X}_i^0)$ and $\mathcal{B}_i(R_i,\mathbf{X}_i^0)\times\mathcal{D}\times\mathcal{B}_i(R_i,\mathbf{X}_i^0)$, respectively, with constants $\tilde{L}^D_{\nabla,i}$ and $\tilde{L}^X_{\nabla,i}$. Let us denote $\tilde{L}^D_{\nabla} \triangleq \max_i \ \tilde{L}^D_{\nabla,i}$ and $\tilde{L}^X_{\nabla} \triangleq \max_i \ \tilde{L}^X_{\nabla,i}$.*

A.2.3. Step 3–Decrease of $\{U(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$

We study here the properties of $\{U(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ showing, in particular, that it is convergent [statement (b-ii)].

We begin with the following intermediate result.

Proposition 10 *Consider the setting of Theorem 2(b); there exist positive constants s_X, $\bar{\tau}_D$, c_7, c_8, and a sufficiently large $\bar{\nu} \in \mathbb{N}_+$ such that the following holds: for all $\nu \geq \bar{\nu}$,*

$$U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}, \mathbf{X}^{\nu+1}) \leq U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\bar{\nu}}}, \mathbf{X}^{\bar{\nu}}) - \sum_{l=\bar{\nu}}^{\nu} Y^l + \sum_{l=\bar{\nu}}^{\nu} W^l + E^{\nu,\bar{\nu}}, \tag{64}$$

where

$$Y^l \triangleq s_X \left(||\mathbf{X}^{l+1} - \mathbf{X}^l||_F - \frac{Z^l}{2s_X} \right)^2 + \frac{\bar{\tau}_D}{I}\gamma^l \left(||\widetilde{\mathbf{D}}^l - \mathbf{D}^l||_F - \frac{I\bar{\epsilon}_\phi}{2\,\bar{\tau}_D} T^l \right)^2, \tag{65}$$

$$Z_l \triangleq c_7 \sum_{t=l}^{\infty} \gamma^t (\rho)^{t-l} + L_X \|\mathbf{U}^l - \mathbf{1}\otimes \overline{\mathbf{U}}_{\boldsymbol{\phi}^l}\|_F, \tag{66}$$

$$W^l \triangleq \frac{I\,\bar{\epsilon}_\phi^2}{4\bar{\tau}_D}\gamma^l \left(T^l\right)^2 + \frac{1}{4\,s_X}\left(Z^l\right)^2 + L_G \gamma^l \left\| \mathbf{D}^l - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^l} \right\|_F, \tag{67}$$

$$E^{\nu,\bar{\nu}} \triangleq \left[\frac{c_6}{1-\rho}((\rho)^{\bar{\nu}} - (\rho)^{\nu+1}) + \frac{c_7 B_X}{(1-\rho)^2} \right] \cdot \left(\max_{t\geq\bar{\nu}} \gamma^t \right), \tag{68}$$

$$\{T^\nu\}_\nu \ \text{ is such that } \ \sum_{\nu=1}^{\infty} (T^\nu)^2 < \infty, \tag{69}$$

with $\rho \in (0,1)$ and $\bar{\epsilon}_\phi$ defined in (48) and (49), respectively.

Proof See Appendix A.4.3 ■

Note that the sequences $\{Z^l\}_l$, $\{W^l\}_l$, and $\{E^{\nu,\bar{\nu}}\}_{\nu,\bar{\nu}}$ above satisfy

$$\lim_{\nu\to\infty} \sum_{l=\bar{\nu}}^{\nu} (Z^l)^2 < \infty; \tag{70}$$

$$\lim_{\nu\to\infty} \sum_{l=\bar{\nu}}^{\nu} W^l < \infty; \tag{71}$$

$$\lim_{\bar{\nu}\to\infty} \underbrace{\left(\lim_{\nu\to\infty} E^{\nu,\bar{\nu}} \right)}_{\triangleq E^{\infty,\bar{\nu}} < \infty} = 0, \tag{72}$$

where (70) follows from (53) [cf. Prop. 5], Assumption E, and Lemma 13(b) [cf. (107)] in Appendix A.4.1; (71) is a consequence of $\lim_{\nu\to\infty}\sum_{l=\nu_1}^{\nu}\gamma^l(T^l)^2 < \infty$ [due to (69)], (52) [cf. Prop. 5], and (70); and eq. (72) is proved by inspection.

It follows from (64), (70)-(72) and the lower-boundedness of U (due to Assumption A1) that $\{U(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ is convergent. Indeed, taking the limsup of the LHS of (64) and using (71) and (72), we get

$$-\infty < \limsup_{\nu\to\infty} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}, \mathbf{X}^{\nu+1}) \leq U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\bar{\nu}}}, \mathbf{X}^{\bar{\nu}}) + \sum_{l=\bar{\nu}}^{\infty} W^l + E^{\infty,\bar{\nu}} < \infty.$$

Taking now the liminf of the RHS of the above inequality with respect to $\bar{\nu}$ while using (72) and $\lim_{\bar{\nu}\to\infty}\sum_{l=\bar{\nu}}^{\infty} W^l = 0$, yields

$$-\infty < \limsup_{\nu\to\infty} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}, \mathbf{X}^{\nu+1}) \leq \liminf_{\bar{\nu}\to\infty} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\bar{\nu}}}, \mathbf{X}^{\bar{\nu}}) < \infty,$$

which implies the convergence of $\{U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu)\}_\nu$ to a finite value, and

$$\lim_{\nu\to\infty} \sum_{l=\bar{\nu}}^{\nu} \left(||\mathbf{X}^{l+1} - \mathbf{X}^l||_F - \frac{Z^l}{2s_X} \right)^2 < \infty, \tag{73}$$

$$\lim_{\nu\to\infty} \sum_{l=\bar{\nu}}^{\nu} \gamma^l \left(||\widetilde{\mathbf{D}}^l - \mathbf{D}^l||_F - \frac{I\bar{\epsilon}_\phi}{2\bar{\tau}_D} T^l \right)^2 < \infty. \tag{74}$$

Finally, we deduce that $\{U(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ converges to the same limit point of $\{U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu)\}_\nu$, due to i) $\sqrt{I}\left\|\overline{\mathbf{D}}^\nu - \overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F \leq \left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F \xrightarrow[\nu\to\infty]{(51)} 0$; ii) the continuity of U; and iii) the boundedness of $\left\{\left(\mathbf{D}^\nu, \mathbf{X}^\nu\right)\right\}_\nu$ [cf. Sec. A.2.2]. This concludes the proof. □

A.2.4. Step 4–Vanishing X-stationarity

Building on the results in the previous step, we prove here $\lim_{\nu\to\infty}\Delta_X(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$ [statement (b-iii)]. For notational simplicity, we will use the shorthand $\widehat{\mathbf{X}}_i^\nu \triangleq \widehat{\mathbf{X}}_i(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ and $\widehat{\mathbf{X}}^\nu \triangleq \widehat{\mathbf{X}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$, with $\widehat{\mathbf{X}}_i(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ and $\widehat{\mathbf{X}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ defined in (29).

Using the equivalence of norms and the triangle inequality, we have

$$\Delta_X(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \leq K_X\|\widehat{\mathbf{X}}^\nu - \mathbf{X}^\nu\|_F \leq K_X \underbrace{\|\mathbf{X}^{\nu+1} - \mathbf{X}^\nu\|_F}_{\texttt{term I}} + K_X \underbrace{\|\mathbf{X}^{\nu+1} - \widehat{\mathbf{X}}^\nu\|_F}_{\texttt{term II}},$$

for some $K_X > 0$. The rest of the proof consists in showing that `term I` and `term II` above are asymptotically vanishing.

• **On `term I`:** `Term I` can be bounded as

$$\frac{1}{2}||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F^2 \leq \left(||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F - \frac{Z^\nu}{2s_X} \right)^2 + \left(\frac{Z^\nu}{2s_X} \right)^2, \tag{75}$$

for all $\nu \in \mathbb{N}_+$, where the inequality follows from $\frac{1}{2}a^2 \leq (a-b)^2 + b^2$, with $a, b \in \mathbb{R}$. This, together with (70) and (73), yields

$$\sum_{\nu=0}^{\infty} ||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F^2 < \infty \quad \Longrightarrow \quad \lim_{\nu\to\infty} ||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F = 0. \tag{76}$$

• **On term II:** Invoking the optimality of $\widehat{\mathbf{X}}_i^\nu$ [cf. (29)] and $\mathbf{X}_i^{\nu+1}$ [cf. (11)], yields

$$\left\langle \nabla_{X_i} f_i(\overline{\mathbf{D}}^\nu, \mathbf{X}_i^\nu) + \hat{\tau}_X(\widehat{\mathbf{X}}_i^\nu - \mathbf{X}_i^\nu), \mathbf{X}_i^{\nu+1} - \widehat{\mathbf{X}}_i^\nu \right\rangle + g_i(\mathbf{X}_i^{\nu+1}) - g_i(\widehat{\mathbf{X}}_i^\nu) \geq 0,$$
$$\left\langle \nabla_{X_i} \tilde{h}_i(\mathbf{X}_i^{\nu+1}; \mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu), \widehat{\mathbf{X}}_i^\nu - \mathbf{X}_i^{\nu+1} \right\rangle + g_i(\widehat{\mathbf{X}}_i^\nu) - g_i(\mathbf{X}_i^{\nu+1}) \geq 0.$$

Summing the two inequalities above and using Remark 7(b), lead to

$$\begin{aligned}\hat{\tau}_X ||\mathbf{X}_i^{\nu+1} - \widehat{\mathbf{X}}_i^\nu||_F^2 \leq & \left\langle \hat{\tau}_X \left(\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right) + \nabla_{X_i} f_i(\overline{\mathbf{D}}^\nu, \mathbf{X}_i^\nu) - \nabla_{X_i} f_i(\mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu), \mathbf{X}_i^{\nu+1} - \widehat{\mathbf{X}}_i^\nu \right\rangle \\ & + \left\langle \nabla_{X_i} \tilde{h}_i(\mathbf{X}_i^\nu; \mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu) - \nabla_{X_i} \tilde{h}_i(\mathbf{X}_i^{\nu+1}; \mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu), \mathbf{X}_i^{\nu+1} - \widehat{\mathbf{X}}_i^\nu \right\rangle.\end{aligned} \tag{77}$$

Using the $\tilde{L}_{\nabla,i}^X$-Lipschitz continuity of $\nabla_{X_i}\tilde{h}_i$ [cf. Remark 9] and the $L_{\nabla,i}$-Lipschitz continuity of ∇f_i, and (10), it is not difficult to show that (77) implies

$$\begin{aligned}&||\mathbf{X}_i^{\nu+1} - \widehat{\mathbf{X}}_i^\nu||_F \\ &\leq \frac{\tilde{L}_{\nabla,i}^X + \hat{\tau}_X}{\hat{\tau}_X} \left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\|_F + \frac{L_{\nabla,i}}{\hat{\tau}_X} \left\|\overline{\mathbf{D}}^\nu - \mathbf{D}_{(i)}^\nu\right\|_F + \frac{L_{\nabla,i}}{\hat{\tau}_X}\gamma^\nu \left\|\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\right\|_F, \\ &\leq \frac{\tilde{L}_{\nabla,i}^X + \hat{\tau}_X}{\hat{\tau}_X} \left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\|_F + \frac{L_{\nabla,i}}{\hat{\tau}_X} \left\|\overline{\mathbf{D}}^\nu - \mathbf{D}_{(i)}^\nu\right\|_F + \frac{L_{\nabla,i}}{\hat{\tau}_X}\gamma^\nu B_D,\end{aligned} \tag{78}$$

for some $B_D > 0$, where in the last inequality we used the boundedness of $\{||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F\}_\nu$. Eq. (78) together with (76), Theorem 2(a), and $\gamma^\nu \overset{\nu\to\infty}{\longrightarrow} 0$ (cf. Assumption E), yield

$$\lim_{\nu\to\infty} \|\mathbf{X}_i^{\nu+1} - \widehat{\mathbf{X}}_i^\nu\|_F = 0. \tag{79}$$

This concludes the proof of statement (b-iiii). □

A.2.5. Step 5–Vanishing liminf D-stationarity

We prove $\liminf_{\nu\to\infty} \Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$ [statement (b-iv)] and $\{(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ has at least one limit point which is a stationary solution of P. For notational simplicity, we will use the shorthand $\widehat{\mathbf{D}}^\nu \triangleq \widehat{\mathbf{D}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$, with $\widehat{\mathbf{D}}(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ defined in (28).

We begin bounding $\Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ as

$$\Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) \leq K_D\|\widehat{\mathbf{D}}^\nu - \overline{\mathbf{D}}^\nu\|_F \leq K_D \underbrace{\|\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\|_F}_{\texttt{term I}} + K_D \underbrace{\|\widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu\|_F}_{\texttt{term II}} + K_D\|\mathbf{D}_{(i)}^\nu - \overline{\mathbf{D}}^\nu\|_F,$$

for some $K_D > 0$. Note that $\|\mathbf{D}_{(i)}^\nu - \overline{\mathbf{D}}^\nu\|_F \to 0$ [Theorem 2(a)]. We show next that $\liminf_{\nu\to\infty} \|\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\|_F = 0$ and $\|\widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu\|_F \to 0$, which proves $\liminf_{\nu\to\infty} \Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$.

• **On term I:** Similarly to the derivations of (75), there holds

$$\frac{\gamma^\nu}{2}||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F^2 \leq \gamma^\nu \left(||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F - \frac{I\bar{\epsilon}_\phi}{2\,\bar{\tau}_D}T^\nu\right)^2 + \left(\frac{I\bar{\epsilon}_\phi}{2\,\bar{\tau}_D}\right)^2 \gamma^\nu (T^\nu)^2, \tag{80}$$

for all $\nu \in \mathbb{N}_+$. By eq. (74) and $\sum_{\nu=0}^\infty (T^\nu)^2 < \infty$ [cf. (69)], we have

$$\sum_{\nu=0}^\infty \gamma^\nu ||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F^2 < \infty \quad \overset{\text{Assumption E}}{\Longrightarrow} \quad \liminf_{\nu\to\infty} ||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F = 0. \tag{81}$$

• **On term II:** Using the optimality of $\widehat{\mathbf{D}}^\nu$ [cf. (28)] and $\widetilde{\mathbf{D}}_{(i)}^\nu$ [cf. (8)], yields

$$\left\langle \nabla_D F(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) + \hat{\tau}_D(\widehat{\mathbf{D}}^\nu - \overline{\mathbf{D}}^\nu), \widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu \right\rangle + G(\widetilde{\mathbf{D}}_{(i)}^\nu) - G(\widehat{\mathbf{D}}^\nu) \geq 0,$$
$$\left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^\nu; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) + I \cdot \mathbf{\Theta}_i^\nu - \nabla_D f_i(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu), \widehat{\mathbf{D}}^\nu - \widetilde{\mathbf{D}}_{(i)}^\nu \right\rangle + G(\widehat{\mathbf{D}}^\nu) - G(\widetilde{\mathbf{D}}_{(i)}^\nu) \geq 0.$$

Summing the two inequalities above and using Remark 7(a), yields

$$\begin{aligned}&\hat{\tau}_D||\widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu||_F^2 \\ &\leq \left\langle \hat{\tau}_D(\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu) + \hat{\tau}_D(\mathbf{D}_{(i)}^\nu - \overline{\mathbf{D}}^\nu) + \nabla_D F(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) - \nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu), \widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu \right\rangle \\ &\quad + \left\langle \nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu) - I \cdot \mathbf{\Theta}_i^\nu, \widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu \right\rangle \\ &\quad - \left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^\nu; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) - \nabla_D \tilde{f}_i(\mathbf{D}_{(i)}^\nu; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu), \widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu \right\rangle.\end{aligned} \tag{82}$$

Using the $\tilde{L}_{\nabla,i}^D$-Lipschitz continuity of $\nabla_D \tilde{f}_i$ [cf. Remark 9] and the L_{∇_D}-Lipschitz continuity of $\nabla_D F$ [cf. Remark 8], it is not difficult to check that (82) implies

$$\begin{aligned}\left\|\widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu\right\|_F \leq & \frac{\tilde{L}_{\nabla,i}^D + \hat{\tau}_D}{\hat{\tau}_D} \left\|\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\right\|_F + \left\|\overline{\mathbf{D}}^\nu - \mathbf{D}_{(i)}^\nu\right\|_F + \frac{L_{\nabla_D}}{\hat{\tau}_D\sqrt{I}} \left\|\mathbf{D}^\nu - \mathbf{1} \otimes \overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\| \\ & + \frac{I}{\hat{\tau}_D} \left\|\mathbf{\Theta}_i^\nu - \frac{1}{I}\nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu)\right\|_F,\end{aligned} \tag{83}$$

for all $i = 1, \ldots, I$. Since $\liminf_{\nu\to\infty} \|\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\|_F = 0$ [cf. (81)], $\|\overline{\mathbf{D}}^\nu - \mathbf{D}_{(i)}^\nu\|_F \to 0$ [Theorem 2(a)], and $\left\|\mathbf{D}^\nu - \mathbf{1} \otimes \overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\| \to 0$ [cf. (51)], to prove $\liminf_{\nu\to\infty} \|\widetilde{\mathbf{D}}_{(i)}^\nu - \widehat{\mathbf{D}}^\nu\|_F = 0$, it is sufficient to show that the last term on the RHS of the above inequality is asymptotically vanishing, which is done in the lemma below.

Lemma 11 (Vanishing gradient-tracking error) *In the setting above, there holds:*

$$\sum_{\nu=0}^\infty \left\|\mathbf{\Theta}^\nu - \mathbf{1} \otimes \frac{1}{I}\nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu)\right\|_F^2 < \infty. \tag{84}$$

Proof See Sec. A.4.4. ■

By $\liminf_{\nu\to\infty} \Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$, it follows that there exists an infinte subset $\mathcal{N} \subseteq \mathbb{N}_+$ such that $\lim_{\mathcal{N}\ni\nu\to\infty} \Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$. Since $\{(\mathbf{D}^\nu, \mathbf{X}^\nu)\}_\nu$ is bounded [cf. Sec. A.2.2], it has a convergent subsequence $\{(\mathbf{D}^\nu, \mathbf{X}^\nu)\}_{\nu\in\mathbb{N}'}$, with $\mathbb{N}' \subseteq \mathbb{N}$; let $(\overline{\mathbf{D}}^\infty, \mathbf{X}^\infty)$ denote its limit point. Then, it must be $\lim_{\mathcal{N}'\ni\nu\to\infty} \Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$. Combining this result with $\lim_{\nu\to\infty} \Delta_X(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu) = 0$ (cf. Sec. A.2.4), one can conclude $\lim_{\mathcal{N}'\ni\nu\to\infty} \Delta^\nu = 0$; hence, $(\overline{\mathbf{D}}^\infty, \mathbf{X}^\infty)$ is a stationary solution of Problem P.

A.2.6. Step 6–Vanishing D-stationarity

Finally, we prove $\lim_{\nu\to\infty}\Delta_D(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)=0$ [statement (b')]. In view of the results already proved in Step 5, it is sufficient to show that $\limsup_{\nu\to\infty}||\widetilde{\mathbf{D}}^\nu-\mathbf{D}^\nu||_F=0$.

1) Preliminaries: We begin introducing the following preliminary results.

Proposition 12 *In the setting of Theorem 2(a), the following hold for $\widetilde{\mathbf{D}}^\nu$ [cf. (8)] and $\mathbf{X}^\nu$ [cf. (11)]:*

(a) There exists some constant $L_D>0$ and sequence $\{\tilde{T}^\nu\}_\nu$, with $\lim_{\nu\to\infty}\tilde{T}^\nu=0$, such that, for any $\nu_1,\nu_2\in\mathbb{N}_+$,

$$||\widetilde{\mathbf{D}}^{\nu_2}-\widetilde{\mathbf{D}}^{\nu_1}||_F\le L_D\Big(||\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_2}}-\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_1}}||_F+||\mathbf{X}^{\nu_2}-\mathbf{X}^{\nu_1}||_F\Big)+\tilde{T}^{\nu_1}+\tilde{T}^{\nu_2};\tag{85}$$

(b) There exist some constants $0<p_X<1$ and $q_X>0$, and a sufficiently large $\nu_X\in\mathbb{N}_+$ such that, for all $\nu\ge\nu_X$,

$$||\mathbf{X}^{\nu+1}-\mathbf{X}^\nu||_F\le p_X||\mathbf{X}^\nu-\mathbf{X}^{\nu-1}||_F+q_X||\mathbf{U}^\nu-\mathbf{U}^{\nu-1}||_F.\tag{86}$$

Proof See Appendix A.4.5 ■

2) Proof of $\limsup_{\nu\to\infty}||\widetilde{\mathbf{D}}^\nu-\mathbf{D}^\nu||_F=0$. For notational simplicity, let us define $\Delta\widetilde{\mathbf{D}}^\nu\triangleq\widetilde{\mathbf{D}}^\nu-\mathbf{D}^\nu$. Suppose by contradiction that $\limsup_{\nu\to\infty}||\Delta\widetilde{\mathbf{D}}^\nu||_F>0$; since $\liminf_{\nu\to\infty}||\Delta\widetilde{\mathbf{D}}^\nu||_F=0$ [cf. (81)], there exists $\delta>0$ such that $||\Delta\widetilde{\mathbf{D}}^\nu||_F>2\delta$ and $||\Delta\widetilde{\mathbf{D}}^{\nu'}||_F<\delta$ for infinitely many $\nu,\nu'\in\mathbb{N}_+$. Therefore, one can find an infinite subset of indices, denoted by $\mathcal{K}$, having the following properties: for any $\nu\in\mathcal{K}$, there exists an index $i_\nu>\nu$ such that

$$||\Delta\widetilde{\mathbf{D}}^\nu||_F<\delta,\qquad ||\Delta\widetilde{\mathbf{D}}^{i_\nu}||_F>2\delta,\tag{87}$$

$$\delta\le||\Delta\widetilde{\mathbf{D}}^j||_F\le2\delta,\qquad \nu<j<i_\nu.\tag{88}$$

Let ν_2 be a sufficiently large integer such that (64) holds and $T^\nu<\frac{2\bar{\tau}_D\delta}{I\bar{\epsilon}_\phi}$, for all $\nu\ge\nu_2$ [such ν_2 exists, due to (69)]. Note that there exists a $\bar{\delta}>0$ such that $\delta-\frac{I\bar{\epsilon}_\phi}{2\bar{\tau}_D}T^\nu\ge\bar{\delta}$, for all $\nu\ge\nu_2$. Choose $\mathcal{K}\ni\nu\ge\nu_2$; using (64), with $\nu=i_\nu$ and $\bar{\nu}=\nu+1$, yields

$$U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{i_\nu+1}},\mathbf{X}^{i_\nu+1})\le U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}^{\nu+1})-c_8\,(\bar{\delta})^2\sum_{l=\nu+1}^{i_\nu}\gamma^l+\sum_{l=\nu+1}^{i_\nu}W^l+E^{i_\nu,\nu+1},\tag{89}$$

for some finite constant $c_8>0$. Using the convergence of $\{U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu},\mathbf{X}^\nu)\}_\nu$, $\sum_{l=1}^\infty W^l<\infty$ [cf. (71)], and $\lim_{\mathcal{K}\ni\nu\to\infty}E^{i_\nu,\nu+1}=0$ [cf. (72)], inequality (89) implies

$$\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{l=\nu+1}^{i_\nu}\gamma^l=0.\tag{90}$$

We show next that (90) leads to a contradiction.

It follows from (87) and (88) that, for all $\mathcal{K}\ni\nu\ge\nu_2$,

$$\begin{aligned}\delta<&||\Delta\widetilde{\mathbf{D}}^{i_\nu}||_F-||\Delta\widetilde{\mathbf{D}}^\nu||_F\\ \overset{(a)}{\le}&||\Delta\widetilde{\mathbf{D}}^{i_\nu}-\Delta\widetilde{\mathbf{D}}^\nu||_F=||\widetilde{\mathbf{D}}^{i_\nu}-\mathbf{D}^{i_\nu}-\widetilde{\mathbf{D}}^\nu+\mathbf{D}^\nu||_F\\ \overset{(b)}{\le}&||\widetilde{\mathbf{D}}^{i_\nu}-\widetilde{\mathbf{D}}^\nu||+||\mathbf{D}^\nu\pm\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\pm\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^{i_\nu}}-\mathbf{D}^{i_\nu}||_F\\ \overset{(85)}{\le}&\Big(L_D+\sqrt{I}\Big)\,||\overline{\mathbf{D}}_{\boldsymbol{\phi}^{i_\nu}}-\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}||_F+\widetilde{E}^{i_\nu,\nu},\end{aligned}\tag{91}$$

with

$$\widetilde{E}^{i_\nu,\nu}\triangleq L_D||\mathbf{X}^{i_\nu}-\mathbf{X}^\nu||_F+||\mathbf{D}^\nu-\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}||_F+||\mathbf{D}^{i_\nu}-\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^{i_\nu}}||_F+\tilde{T}^\nu+\tilde{T}^{i_\nu},$$

where in (a) we used the reverse triangle inequality (i.e. $||\mathbf{A}||_F-||\mathbf{B}||_F\le||\mathbf{A}-\mathbf{B}||_F,\forall\mathbf{A},\mathbf{B}\in\mathbb{R}^{MI\times K}$); and in (b) we add/subtracted some dummy terms and used the triangle inequality.

We prove next that $\lim_{\nu\to\infty}\widetilde{E}^{i_\nu,\nu}=0$. Clearly, if $\lim_{\nu\to\infty}||\mathbf{X}^{i_\nu}-\mathbf{X}^\nu||_F=0$, then (51) [cf. Proposition 5] and $\tilde{T}^\nu\overset{\nu\to\infty}{\longrightarrow}0$ [cf. Proposition 12(a)] imply $\lim_{\nu\to\infty}\widetilde{E}^{i_\nu,\nu}=0$. It is then sufficient to show $\lim_{\nu\to\infty}||\mathbf{X}^{i_\nu}-\mathbf{X}^\nu||_F=0$.

First, we bound $||\mathbf{X}^{i_\nu}-\mathbf{X}^\nu||_F$ properly.

Summing (86) from $\nu>\nu_2$ to $i_\nu-1$, yields

$$\sum_{t=\nu}^{i_\nu-1}||\mathbf{X}^{t+1}-\mathbf{X}^t||_F\le p_X\sum_{t=\nu}^{i_\nu-1}||\mathbf{X}^t-\mathbf{X}^{t-1}||_F+q_X\sum_{t=\nu}^{i_\nu-1}||\mathbf{U}^t-\mathbf{U}^{t-1}||_F,\tag{92}$$

implying

$$||\mathbf{X}^{i_\nu}-\mathbf{X}^\nu||_F\le\sum_{t=\nu}^{i_\nu-1}||\mathbf{X}^{t+1}-\mathbf{X}^t||_F\le\frac{p_X}{1-p_X}||\mathbf{X}^\nu-\mathbf{X}^{\nu-1}||_F+\frac{q_X}{1-p_X}\sum_{t=\nu}^{i_\nu-1}||\mathbf{U}^t-\mathbf{U}^{t-1}||_F.$$

Since $\lim_{\nu\to\infty}||\mathbf{X}^\nu-\mathbf{X}^{\nu-1}||_F=0$ [cf. (76)], it follows from the above inequality that $\lim_{\mathcal{K}\ni\nu\to\infty}||\mathbf{X}^{i_\nu}-\mathbf{X}^\nu||_F=0$, if $\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu-1}||\mathbf{U}^t-\mathbf{U}^{t-1}||_F=0$, which is proved next. Rewrite first $||\mathbf{U}^t-\mathbf{U}^{t-1}||_F$ as

$$\begin{aligned}||\mathbf{U}^t-\mathbf{U}^{t-1}||_F&\le||\mathbf{U}^t-\mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^t}||_F+||\mathbf{U}^{t-1}-\mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^{t-1}}||_F+\sqrt{I}||\overline{\mathbf{U}}_{\boldsymbol{\phi}^t}-\overline{\mathbf{U}}_{\boldsymbol{\phi}^{t-1}}||_F\\ &\overset{(42)-(43)}{=}||\mathbf{U}^t-\mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^t}||_F+||\mathbf{U}^{t-1}-\mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^{t-1}}||_F+\frac{\gamma^t}{\sqrt{I}}\left|\left|\sum_{i=1}^I\phi_i^t\left(\widetilde{\mathbf{D}}_{(i)}^t-\mathbf{D}_{(i)}^t\right)\right|\right|_F.\end{aligned}$$

Since $||\sum_{i=1}^I\phi_i^t(\widetilde{\mathbf{D}}_{(i)}^t-\mathbf{D}_{(i)}^t)||_F$ is bounded (due to $\phi_i^t\le\bar{\epsilon}_\phi$ [cf. (49)] and compactness of $\mathcal{D}$) and $\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu}\gamma^t=0$ [cf. (90)], there holds $\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu}\gamma^t||\sum_{i=1}^I\widetilde{\mathbf{D}}_{(i)}^t-\mathbf{D}_{(i)}^t||_F=0$. Therefore, to prove $\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu-1}||\mathbf{U}^t-\mathbf{U}^{t-1}||_F=0$, it is sufficient to show that $\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu}||\mathbf{U}^t-\mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^t}||_F=0$ [which implies also $\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu}||\mathbf{U}^{t-1}-\mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^{t-1}}||_F=0$, due to $\lim_{\nu\to\infty}||\mathbf{U}^\nu-\mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu}||_F=0$, see (53)]. By (57), the boundedness

of $\{\widetilde{\mathbf{D}}^\nu\}_\nu$, and (90), it is sufficient to show that $\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu}||\mathbf{D}^t-\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^t}||_F=0$. We have

$$\begin{aligned}
&\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{l=\nu}^{i_\nu}||\mathbf{D}^l-\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^l}||_F \overset{(56)}{\leq} \lim_{\mathcal{K}\ni\nu\to\infty}\sum_{l=\nu}^{i_\nu}\left(c_1(\rho)^l+c_2\sum_{t=0}^{l-1}\gamma^t\,(\rho)^{l-t}\right)\\
&=c_2\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{l=\nu}^{i_\nu}\sum_{t=0}^{l-1}\gamma^t\,(\rho)^{l-t}=c_2\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=0}^{i_\nu-1}\sum_{l=\max(t+1,\nu)}^{i_\nu}\gamma^t\,(\rho)^{l-t}\\
&=c_2\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=0}^{\nu-1}\sum_{l=\nu}^{i_\nu}\gamma^t\,(\rho)^{l-t}+c_2\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu-1}\sum_{l=t+1}^{i_\nu}\gamma^t\,(\rho)^{l-t}\\
&\leq c_2\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=0}^{\nu-1}\sum_{l=\nu}^{i_\nu}\gamma^t\,(\rho)^{l-t}+\frac{c_2}{1-\rho}\underbrace{\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=\nu}^{i_\nu-1}\gamma^t}_{=0\quad\text{by (90)}}\\
&\leq\frac{c_2}{1-\rho}\lim_{\mathcal{K}\ni\nu\to\infty}\sum_{t=0}^{\nu-1}\gamma^t\,(\rho)^{\nu-t}\overset{(104)}{=}0.
\end{aligned}$$

This proves $\lim_{\nu\to\infty}||\mathbf{X}^{i_\nu}-\mathbf{X}^\nu||_F=0$ and thus $\lim_{\nu\to\infty}\ \widetilde{E}^{i_\nu,\nu}=0$.

We can now prove that (90) leads to a contradiction. Since $\widetilde{E}^{i_{\nu_\nu},\nu}\overset{\nu\to\infty}{\longrightarrow}0$, there exists a sufficiently large integer $\nu_3\in\mathcal{K}$, such that $\nu_3>\nu_2$ and $\widetilde{E}^{i_{\nu_\nu},\nu}<\delta$, for all $\nu>\nu_3$. Define δ' such that $0<\delta'\leq\delta-\widetilde{E}^{i_{\nu_\nu},\nu}$. Using (43) and $\mathbf{1}^\intercal\boldsymbol{\phi}^\nu=I$, (91) implies

$$\frac{\delta'}{(L_D+1)\sqrt{I}}<\sum_{t=\nu}^{i_\nu-1}\gamma^t\,||\Delta\widetilde{\mathbf{D}}^t||_F\overset{(87)-(88)}{\leq}2\delta\sum_{t=\nu}^{i_\nu-1}\gamma^t, \tag{93}$$

for all $\mathcal{K}\ni\nu>\nu_3$. Equation (93) contradicts (90). Hence, it must be $\limsup_{\nu\to\infty}\ ||\Delta\widetilde{\mathbf{D}}^\nu||_F=0$, and thus $\lim_{\nu\to\infty}\ ||\widetilde{\mathbf{D}}^\nu-\mathbf{D}^\nu||_F=0$. □

A.3. Proof of Theorem 3

(a) Rate of consensus error. Fix $\theta\in(0,1)$. Combining (50) and (56), we obtain

$$\begin{aligned}
e^\nu\leq&c_8\sum_{l=1}^{\nu}\gamma^{\nu-l}\,(\rho)^l=c_8\sum_{l=1}^{\lfloor(1-\theta)\nu\rfloor}\gamma^{\nu-l}\,(\rho)^l+c_8\sum_{l=\lfloor(1-\theta)\nu\rfloor+1}^{\nu}\gamma^{\nu-l}\,(\rho)^l\\
\leq&c_8\,\gamma^{\nu-\lfloor(1-\theta)\nu\rfloor}\sum_{l=0}^{\lfloor(1-\theta)\nu\rfloor}(\rho)^l+c_8\,\gamma^0\sum_{l=\lfloor(1-\theta)\nu\rfloor+1}^{\nu}(\rho)^l,\\
\overset{(a)}{=}&c_8\,\gamma^{\lceil\theta\nu\rceil}\,\frac{1-(\rho)^{\lfloor(1-\theta)\nu\rfloor+1}}{1-\rho}+c_8\,\gamma^0\,(\rho)^{\lfloor(1-\theta)\nu\rfloor+1}\cdot\frac{1-(\rho)^{\lceil\theta\nu\rceil}}{1-\rho}\\
\leq&c_9\left(\gamma^{\lceil\theta\nu\rceil}+(\rho)^{(1-\theta)\nu}\right)\overset{(b)}{\leq}c_9\left(\gamma^{\lceil\theta\nu\rceil}+\left((\rho)^{\frac{1-\theta}{\theta}}\right)^{\lceil\theta\nu\rceil-1}\right)\overset{(c)}{=}\mathcal{O}\left(\gamma^{\lceil\theta\nu\rceil}\right),
\end{aligned}\tag{94}$$

for some positive constants c_8 and c_9, where in (a) we used $\nu-\lfloor(1-\theta)\nu\rfloor=\lceil\theta\nu\rceil$; (b) follows from $x\geq\lceil x\rceil-1$, $x\in\mathbb{R}$; and in (c) we used $(\tilde{\rho})^\nu=o(\gamma^\nu)$, for any $\tilde{\rho}\in(0,1)$. This proves statement (a).

(b) Rate of optimization errors. In the following we will use the shorthand: $\widehat{\mathbf{X}}_i^\nu\triangleq\widehat{\mathbf{X}}_i(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$ and $\widehat{\mathbf{X}}^\nu\triangleq\widehat{\mathbf{X}}(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$, with $\widehat{\mathbf{X}}_i(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$ and $\widehat{\mathbf{X}}(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$ defined in (29); and $\widehat{\mathbf{D}}^\nu\triangleq\widehat{\mathbf{D}}(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$, with $\widehat{\mathbf{D}}(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$ defined in (28).

1) Proof of (32)**:** We begin bounding $\Delta_X(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$ as

$$\frac{1}{2}(\Delta_X(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu))^2\overset{(a)}{\leq}\frac{K_1}{2}||\widehat{\mathbf{X}}^\nu-\mathbf{X}^\nu||_F^2\leq K_1||\mathbf{X}^{\nu+1}-\widehat{\mathbf{X}}^\nu||_F^2+K_1||\mathbf{X}^{\nu+1}-\mathbf{X}^\nu||_F^2, \tag{95}$$

where (a) holds by equivalence of the norms with K_1 being a proper positive constant.

By squaring both sides of (78) and using $\frac{1}{n}(\sum_{i=1}^n a_i)^2\leq\sum_{i=1}^n a_i^2,\forall a_i\in\mathbb{R}$ (by Jensen inequality), the first term on the RHS of (95) can be bounded as

$$\frac{1}{3K_2}||\mathbf{X}_i^{\nu+1}-\widehat{\mathbf{X}}_i^\nu||_F^2\leq\left\|\mathbf{X}_i^{\nu+1}-\mathbf{X}_i^\nu\right\|_F^2+\left\|\overline{\mathbf{D}}^\nu-\mathbf{D}_{(i)}^\nu\right\|_F^2+(\gamma^\nu)^2, \tag{96}$$

for some positive constant $K_2>0$. Summing (96) over $i=1,\ldots,I$, yields

$$\begin{aligned}
\frac{1}{3K_2}||\mathbf{X}^{\nu+1}-\widehat{\mathbf{X}}^\nu||_F^2&\leq\left\|\mathbf{X}^{\nu+1}-\mathbf{X}^\nu\right\|_F^2+\left\|\mathbf{D}^\nu-\mathbf{1}\otimes\overline{\mathbf{D}}^\nu\right\|_F^2+I(\gamma^\nu)^2\\
&\leq\left\|\mathbf{X}^{\nu+1}-\mathbf{X}^\nu\right\|_F^2+4\left\|\mathbf{D}^\nu-\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F^2+I(\gamma^\nu)^2.
\end{aligned}\tag{97}$$

Finally, combining (95) and (97), yields

$$\frac{1}{2K_1}(\Delta_X(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu))^2\leq(3K_2+1)\left\|\mathbf{X}^{\nu+1}-\mathbf{X}^\nu\right\|_F^2+12K_1\left\|\mathbf{D}^\nu-\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F^2+3K_2I(\gamma^\nu)^2. \tag{98}$$

It follows from (98) together with (76), (52) and Assumption E

$$\sum_{\nu=0}^{\infty}(\Delta_X(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu))^2<\infty.$$

By the definition of $T_{X,\epsilon}$, there holds

$$T_{X,\epsilon}\epsilon^2\leq\sum_{\nu=0}^{T_{X,\epsilon}}(\Delta_X(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu))^2<\infty,$$

which proves (32).

2) Proof of (33)**:** Following the same approach as above, we can bound $\Delta_D(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu)$ as

$$\frac{1}{3}(\Delta_D(\overline{\mathbf{D}}^\nu,\mathbf{X}^\nu))^2\leq\frac{K_3}{3}||\widehat{\mathbf{D}}^\nu-\overline{\mathbf{D}}^\nu||_F^2\leq K_3||\widehat{\mathbf{D}}^\nu-\widetilde{\mathbf{D}}_{(i)}^\nu||_F^2+K_3||\widetilde{\mathbf{D}}_{(i)}^\nu-\mathbf{D}_{(i)}^\nu||_F^2+K_3||\mathbf{D}_{(i)}^\nu-\overline{\mathbf{D}}^\nu||_F^2, \tag{99}$$

for some $K_3>0$. Using (83), the first term on the RHS of (99) can be bounded as

$$\begin{aligned}
\frac{1}{4K_4}||\widetilde{\mathbf{D}}_{(i)}^\nu-\widehat{\mathbf{D}}^\nu||_F^2\leq&||\widetilde{\mathbf{D}}_{(i)}^\nu-\mathbf{D}_{(i)}^\nu||_F^2+\left\|\boldsymbol{\Theta}_{(i)}^\nu-\frac{1}{I}\nabla_DF(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu},\mathbf{X}^\nu)\right\|_F^2+||\overline{\mathbf{D}}^\nu-\mathbf{D}_{(i)}^\nu||_F^2\\
&+\left\|\mathbf{D}^\nu-\mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|^2
\end{aligned}\tag{100}$$

with some constant $K_4 > 0$. Summing (100) over $i = 1, \dots, I$, we get

$$\begin{aligned}
&\frac{1}{4K_4}\|\widetilde{\mathbf{D}}^\nu - \mathbf{1}\otimes\widehat{\mathbf{D}}^\nu\|_F^2 \\
\le&\|\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu\|_F^2 + \left\|\mathbf{\Theta}^\nu - \mathbf{1}\otimes\frac{1}{I}\nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu)\right\|_F^2 + \left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}^\nu\right\|_F^2 + I\left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F^2 \\
\le&\|\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu\|_F^2 + \left\|\mathbf{\Theta}^\nu - \mathbf{1}\otimes\frac{1}{I}\nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu)\right\|_F^2 + (4+I)\left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F^2.
\end{aligned} \tag{101}$$

Summing (99) over $i = 1, \dots, I$ and using (101), yields

$$\begin{aligned}
&\frac{I}{3K_3}(\Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu))^2 \\
\le&(4K_4+1)\|\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu\|_F^2 + 4K_4\left\|\mathbf{\Theta}^\nu - \mathbf{1}\otimes\frac{1}{I}\nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu)\right\|_F^2 \\
&+ 4((4+I)K_4+1)\left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F^2.
\end{aligned} \tag{102}$$

It follows from (102) together with (81), (84), and (52)

$$\sum_{\nu=0}^{\infty}\gamma^\nu(\Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu))^2 < \infty.$$

By definition of $T_{D,\epsilon}$ and non-increasing property of $\{\gamma^\nu\}_\nu$, we get

$$\gamma^{T_{D,\epsilon}}\, T_{D,\epsilon}\,\epsilon^2 \le \sum_{\nu=0}^{T_{D,\epsilon}}\gamma^\nu(\Delta_D(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu))^2 < \infty. \tag{103}$$

Using $\gamma^\nu = K/\nu^p$, with some constant $K > 0$ and $p \in (1/2, 1)$, (103) provides the desired result as in (33). □

A.4. Miscellaneous results

This section contains some miscellaneous results used in the proofs of Theorems 2 and 3.

A.4.1. Sequence properties

The following lemma summarizes some summability properties of suitably chosen sequences, which appear in some of the proofs.

Lemma 13 *Given the sequences $\{a^\nu\}_\nu$ and $\{b^\nu\}_\nu$, and a scalar $\lambda \in [0, 1)$, the following hold:*

(a) If $\lim_{\nu\to\infty} a^\nu = 0$, then,

$$\lim_{\nu\to\infty}\sum_{t=1}^{\nu} a^t(\lambda)^{\nu-t} = 0. \tag{104}$$

(b) If $\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\left(a^t\right)^2 < \infty$ and $\lim_{\nu\to\infty}\sum_{t=1}^{\nu}\left(b^t\right)^2 < \infty$, then

$$\lim_{\nu\to\infty}\sum_{l=1}^{\nu}\sum_{t=1}^{l} a^t b^l(\lambda)^{l-t} < \infty, \tag{105}$$

$$\lim_{\nu\to\infty}\sum_{l=1}^{\nu}\sum_{t=1}^{l}\left(a^t\right)^2(\lambda)^{l-t} < \infty, \tag{106}$$

$$\lim_{\nu,\nu'\to\infty}\sum_{l=1}^{\nu}\left(\sum_{t=l}^{\nu'} a^t(\lambda)^{t-l}\right)^2 < \infty. \tag{107}$$

Proof For the proof of (a) and (105)-(106) in (b), see (Nedić et al., 2010, Lemma 7). We prove next (107). Expand the LHS of (107) as:

$$\begin{aligned}
\lim_{\nu,\nu'\to\infty}\sum_{l=1}^{\nu}\left(\sum_{t=l}^{\nu'} a^t(\lambda)^{t-l}\right)^2 &= \lim_{\nu,\nu'\to\infty}\sum_{l=1}^{\nu}\sum_{t=l}^{\nu'}\sum_{k=l}^{\nu'} a^t a^k(\lambda)^{t-l}(\lambda)^{k-l} \\
&\le \lim_{\nu,\nu'\to\infty}\sum_{l=1}^{\nu}\sum_{t=l}^{\nu'}\sum_{k=l}^{\nu'}\frac{(a^t)^2+(a^k)^2}{2}(\lambda)^{t-l}(\lambda)^{k-l} = \lim_{\nu,\nu'\to\infty}\sum_{l=1}^{\nu}\sum_{t=l}^{\nu'}(a^t)^2(\lambda)^{t-l}\sum_{k=l}^{\nu'}(\lambda)^{k-l},
\end{aligned}$$

where the inequality is due to $a \cdot b \le (a^2 + b^2)/2$. Using the bound on the sum of the geometric series, the above inequality yields

$$\begin{aligned}
\lim_{\nu,\nu'\to\infty}\sum_{l=1}^{\nu}\left(\sum_{t=l}^{\nu'} a^t(\lambda)^{t-l}\right)^2 &\le \frac{1}{1-\lambda}\lim_{\nu,\nu'\to\infty}\sum_{l=1}^{\nu}\sum_{t=l}^{\nu'}(a^t)^2(\lambda)^{t-l} = \frac{1}{1-\lambda}\lim_{\nu'\to\infty}\sum_{t=1}^{\nu'}\lim_{\nu\to\infty}\sum_{l=1}^{\min(\nu,t)}(a^t)^2(\lambda)^{t-l} \\
&\le \frac{1}{1-\lambda}\lim_{\nu'\to\infty}\sum_{t=1}^{\nu'}(a^t)^2\sum_{l=1}^{t}(\lambda)^{t-l} \le \frac{1}{(1-\lambda)^2}\lim_{\nu'\to\infty}\sum_{t=1}^{\nu'}(a^t)^2 < \infty.
\end{aligned}$$

■

A.4.2. On the properties of the best-response map

Some key properties of the best-response maps defined in (8) and (11) are summarized and proved next.

Proposition 14 *Let $\left\{\left(\mathbf{D}^\nu, \mathbf{X}^\nu\right)\right\}_\nu$ be the sequence generated by the D^4L Algorithm, in the setting of Theorem 2(a). Given the solution maps defined in (8) and (11), the following hold:*

(a) There exist some constants $s_D > 0$ and $\eta > 0$, and a sequence $\{T^\nu\}_\nu$, with $\sum_{\nu=1}^{\infty}(T^\nu)^2 < \infty$, such that: for all $\nu \ge 1$,

$$\begin{aligned}
&\left\langle \nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}, \mathbf{X}^\nu), \sum_{i=1}^{I}\phi_i^\nu\left(\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\right)\right\rangle + \sum_{i=1}^{I}\phi_i^\nu\left(G(\widetilde{\mathbf{D}}_{(i)}^\nu) - G(\mathbf{D}_{(i)}^\nu)\right) \\
&\le -s_D\left(||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F - \frac{I\bar{\epsilon}_\phi}{2\,s_D}T^\nu\right)^2 + \eta\,||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F\sum_{t=1}^{\nu}(\rho)^{\nu-t}||\mathbf{X}^t - \mathbf{X}^{t-1}||_F + \frac{I^2\,\bar{\epsilon}_\phi^2}{4\,s_D}(T^\nu)^2,
\end{aligned} \tag{108}$$

where $\rho \in (0,1)$ and $\bar{\epsilon}_\phi$ are defined in (48) and (49), respectively;
(b) There exist finite constants $s_X > 0$ and $L_X > 0$, such that: for all $\nu \geq 1$,

$$\sum_{i=1}^{I} \left\langle \nabla_{X_i} f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}, \mathbf{X}_i^{\nu}), \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu} \right\rangle + \sum_{i=1}^{I} \left(g_i(\mathbf{X}_i^{\nu+1}) - g_i(\mathbf{X}_i^{\nu}) \right)$$
$$\leq - \sum_{i=1}^{I} \tau_{X,i}^{\nu} ||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}||_F^2 + L_X ||\mathbf{U}^{\nu} - \mathbf{1} \otimes \overline{\mathbf{U}}_{\boldsymbol{\phi}^{\nu}}||_F \; ||\mathbf{X}^{\nu+1} - \mathbf{X}^{\nu}||_F. \tag{109}$$

Proof **(a)** It follows from the optimality of $\widetilde{\mathbf{D}}_{(i)}^{\nu}$ [cf. (8)] and convexity of G that

$$\left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu}; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) + I \boldsymbol{\Theta}_{(i)}^{\nu} - \nabla_D f_i(\mathbf{D}_{(i)}, \mathbf{X}_i), \mathbf{D}_{(i)}^{\nu} - \widetilde{\mathbf{D}}_{(i)}^{\nu} \right\rangle + G(\mathbf{D}_{(i)}^{\nu}) - G(\widetilde{\mathbf{D}}_{(i)}^{\nu}) \geq 0. \tag{110}$$

Adding and subtracting inside the first term $\sum_j \nabla_D f_j(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_j^{\nu})$ and using $\nabla_D \tilde{f}_i(\mathbf{D}_{(i)}^{\nu}; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) = \nabla_D f_i(\mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$ [cf. Remark 7], inequality (110) becomes

$$\left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu}; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) - \nabla_D \tilde{f}_i(\mathbf{D}_{(i)}^{\nu}; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}), \widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu} \right\rangle$$
$$+ \left\langle I \cdot \boldsymbol{\Theta}_{(i)}^{\nu} - \sum_{j=1}^{I} \nabla_D f_j(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_j^{\nu}), \widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu} \right\rangle$$
$$+ \left\langle \sum_{j=1}^{I} \nabla_D f_j(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_j^{\nu}), \widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu} \right\rangle + G(\widetilde{\mathbf{D}}_{(i)}^{\nu}) - G(\mathbf{D}_{(i)}^{\nu}) \leq 0.$$

Invoking the uniform strongly convexity of $\tilde{f}_i(\bullet; \mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$, the definition of $\boldsymbol{\Theta}_{(i)}^{\nu}$ in (13), and recalling that $\nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}^{\nu}) = \sum_j \nabla_D f_j(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_j^{\nu})$, we get

$$\left\langle \nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}^{\nu}), \widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu} \right\rangle + G(\widetilde{\mathbf{D}}_{(i)}^{\nu}) - G(\mathbf{D}_{(i)}^{\nu})$$
$$\leq -\tau_{D,i}^{\nu} \left\| \widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu} \right\|^2 + I \left\| \boldsymbol{\Theta}_{(i)}^{\nu} - \frac{1}{I} \sum_{j=1}^{I} \nabla_D f_j(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_j^{\nu}) \right\|_F \left\| \widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu} \right\|_F.$$

Multiplying both side of the above inequality by the positive quantities ϕ_i^{ν} and summing over $i = 1, 2, \ldots, I$ while using $\phi_i^{\nu} \leq \bar{\epsilon}_\phi$ [cf. (49)], yields

$$\left\langle \nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}^{\nu}), \sum_{i=1}^{I} \phi_i^{\nu} \left(\widetilde{\mathbf{D}}_{(i)}^{\nu} - \mathbf{D}_{(i)}^{\nu} \right) \right\rangle + \sum_{i=1}^{I} \phi_i^{\nu} \left(G(\widetilde{\mathbf{D}}_{(i)}^{\nu}) - G(\mathbf{D}_{(i)}^{\nu}) \right)$$
$$\leq -s_D ||\widetilde{\mathbf{D}}^{\nu} - \mathbf{D}^{\nu}||^2 + I \bar{\epsilon}_\phi \underbrace{\left\| \boldsymbol{\Theta}^{\nu} - \mathbf{1} \otimes \frac{1}{I} \sum_{i=1}^{I} \nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu}) \right\|_F}_{\text{gradient tracking error}} \left\| \widetilde{\mathbf{D}}^{\nu} - \mathbf{D}^{\nu} \right\|_F, \tag{111}$$

where s_D is any positive constant such that $s_D \leq \min_{i,\nu} \phi_i^{\nu} \tau_{D,i}^{\nu}$ [note that such a constant exists because $\phi_i^{\nu} \geq \underline{\epsilon}_\phi$, with $\underline{\epsilon}_\phi > 0$ defined in (49), and all $\tau_{D,i}^{\nu}$ are uniformly bounded away from zero–see Assumption D1].

Now let us bound the *gradient tracking error* term in (111). Using (40) recursively, $\boldsymbol{\Theta}^{\nu}$ can be rewritten as

$$\boldsymbol{\Theta}^{\nu} = \widehat{\mathbf{W}}^{\nu-1:0} \boldsymbol{\Theta}^0 + \sum_{t=1}^{\nu-1} \widehat{\mathbf{W}}^{\nu-1:t} \left(\widehat{\boldsymbol{\Phi}}^t \right)^{-1} \left(\mathbf{G}^t - \mathbf{G}^{t-1} \right) + \left(\widehat{\boldsymbol{\Phi}}^{\nu} \right)^{-1} \left(\mathbf{G}^{\nu} - \mathbf{G}^{\nu-1} \right). \tag{112}$$

Using the definition of $\mathbf{G}^{\nu}$ [cf. (34)] and $\widehat{\mathbf{J}}$ [cf. (45)], write

$$\mathbf{1} \otimes \frac{1}{I} \sum_{i=1}^{I} \nabla_D f_i(\mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) = \widehat{\mathbf{J}} \mathbf{G}^{\nu} = \widehat{\mathbf{J}} \mathbf{G}^0 + \sum_{t=1}^{\nu} \widehat{\mathbf{J}} \left(\mathbf{G}^t - \mathbf{G}^{t-1} \right),$$

which, using $\boldsymbol{\Theta}^0 = \mathbf{G}^0$, leads to the following expansion for $\mathbf{1} \otimes \frac{1}{I} \sum_{i=1}^{I} \nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu})$:

$$\mathbf{1} \otimes \frac{1}{I} \sum_{i=1}^{I} \nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu}) = \widehat{\mathbf{J}} \widetilde{\boldsymbol{\Theta}}^0 + \sum_{t=1}^{\nu} \widehat{\mathbf{J}} \left(\mathbf{G}^t - \mathbf{G}^{t-1} \right)$$
$$+ \mathbf{1} \otimes \frac{1}{I} \sum_{i=1}^{I} \left(\nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu}) - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) \right). \tag{113}$$

Using (112) and (113), the *gradient tracking error* term in (111) can be upper bounded as

$$\begin{aligned}
&\left\| \boldsymbol{\Theta}^{\nu} - \mathbf{1} \otimes \frac{1}{I} \sum_{i=1}^{I} \nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu}) \right\|_F \\
&\overset{(a)}{\leq} \left\| \widehat{\mathbf{W}}^{\nu-1:0} - \widehat{\mathbf{J}} \right\|_2 \left\| \boldsymbol{\Theta}^0 \right\|_F + \frac{1}{\underline{\epsilon}_\phi} \sum_{t=1}^{\nu-1} \left\| \widehat{\mathbf{W}}^{\nu-1:t} - \widehat{\mathbf{J}}_{\phi^t} \right\|_2 \left\| \mathbf{G}^t - \mathbf{G}^{t-1} \right\|_F \\
&\quad + \left\| \left(\widehat{\boldsymbol{\Phi}}^{\nu} \right)^{-1} - \widehat{\mathbf{J}} \right\|_2 \left\| \mathbf{G}^{\nu} - \mathbf{G}^{\nu-1} \right\|_F + \frac{1}{\sqrt{I}} \sum_{i=1}^{I} \left\| \nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu}) - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) \right\|_F \\
&\overset{(b)}{\leq} c_4 (\rho)^{\nu} + c_5 L_{\nabla} \sum_{t=1}^{\nu} (\rho)^{\nu-t} \left(\left\| \mathbf{D}^t - \mathbf{D}^{t-1} \right\|_F + \left\| \mathbf{X}^t - \mathbf{X}^{t-1} \right\|_F \right) + L_{\nabla} \left\| \mathbf{D}^{\nu} - \mathbf{1} \otimes \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}} \right\|_F \\
&\overset{(c)}{=} T^{\nu} + c_5 L_{\nabla} \sum_{t=1}^{\nu} (\rho)^{\nu-t} \left\| \mathbf{X}^t - \mathbf{X}^{t-1} \right\|_F,
\end{aligned} \tag{114}$$

for some positive finite constants c_4 and c_5, where in (a) we used the lower bound $\phi_i^{\nu} \geq \underline{\epsilon}_\phi$ [cf. (49)] and $\widehat{\mathbf{J}}_{\phi^t} = \widehat{\mathbf{J}} \widehat{\boldsymbol{\Phi}}^t$ [cf. (47)]; and in (b) we used (34), (63) (cf. Remark 8), and Lemma 4; and in (c) we defined T^{ν} as

$$T^{\nu} \triangleq c_4 (\rho)^{\nu} + c_5 L_{\nabla} \sum_{t=1}^{\nu} (\rho)^{\nu-t} \left\| \mathbf{D}^t - \mathbf{D}^{t-1} \right\|_F + L_{\nabla} \left\| \mathbf{D}^{\nu} - \mathbf{1} \otimes \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}} \right\|_F. \tag{115}$$

Substituting (114) into (111) yields

$$\begin{aligned}
&\Big\langle \nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu},\mathbf{X}^\nu), \sum_{i=1}^I \phi_i^\nu \left(\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\right)\Big\rangle + \sum_{i=1}^I \phi_i^\nu \left(G(\widetilde{\mathbf{D}}_{(i)}^\nu) - G(\mathbf{D}_{(i)}^\nu)\right)\\
&\quad\le -s_D||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F^2 + I\bar{\epsilon}_\phi T^\nu ||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F\\
&\qquad + I\,\bar{\epsilon}_\phi\, c_5\, L_\nabla\, ||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F \sum_{t=1}^\nu (\rho)^{\nu-t} \left\|\mathbf{X}^t - \mathbf{X}^{t-1}\right\|_F\\
&\quad= -s_D\left(||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F - \frac{I\,\bar{\epsilon}_\phi}{2\,s_D}T^\nu\right)^2 + \frac{I^2\,\bar{\epsilon}_\phi^2}{4\,s_D}(T^\nu)^2\\
&\qquad + \eta\, ||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F \sum_{t=1}^\nu (\rho)^{\nu-t}\left\|\mathbf{X}^t - \mathbf{X}^{t-1}\right\|_F,
\end{aligned}$$

with $\eta = I\bar{\epsilon}_\phi c_5 L_\nabla$.

To complete the proof, we need to show that $\sum_{\nu=1}^\infty (T^\nu)^2 < \infty$. Note that the first term on the RHS of (115) is square summable, and so is the third one, due to Proposition 5 [cf. (52)]. Invoking Jensen's inequality, it is sufficient to show that the second term on RHS of (115) is square summable. Following the same approach used to prove (52), we have

$$\begin{aligned}
&\lim_{\nu\to\infty}\sum_{t=1}^\nu \left(\sum_{l=1}^t (\rho)^{t-l}\left\|\mathbf{D}^l - \mathbf{D}^{l-1}\right\|_F\right)^2\\
&\le \lim_{\nu\to\infty}\sum_{t=1}^\nu\sum_{l=1}^t\sum_{k=1}^t (\rho)^{t-l}(\rho)^{t-k}\left\|\mathbf{D}^l - \mathbf{D}^{l-1}\right\|_F\left\|\mathbf{D}^k - \mathbf{D}^{k-1}\right\|_F\\
&\overset{(a)}{\le} \frac{1}{1-\rho}\lim_{\nu\to\infty}\sum_{t=1}^\nu\sum_{l=1}^t (\rho)^{t-l}(\gamma^l)^2\left\|\widetilde{\mathbf{D}}^{l-1} - \mathbf{D}^{l-1}\right\|_F^2 \overset{(b)}{<} \infty,
\end{aligned}$$

where (a) follows from $ab \le (a^2+b^2)/2$, and (b) is due to Lemma 13 and Assumption A3. Hence $\sum_{\nu=1}^\infty (T^\nu)^2 < \infty$.

(b) We prove this statement using the definition (15) of $\tilde{f}_i$; the same conclusion holds also using the alternative choice (14) of $\tilde{f}_i$; the proof is thus omitted. Invoking the optimality of $\mathbf{X}^{\nu+1}$ [cf. (11)] together with the convexity of g_i, yield

$$\begin{aligned}
&\left\langle \nabla_{X_i}\tilde{h}_i(\mathbf{X}_i^{\nu+1};\mathbf{U}_{(i)}^\nu,\mathbf{X}_i^\nu) - \nabla_{X_i}\tilde{h}_i(\mathbf{X}_i^\nu;\mathbf{U}_{(i)}^\nu,\mathbf{X}_i^\nu), \mathbf{X}_i^\nu - \mathbf{X}_i^{\nu+1}\right\rangle\\
&+\left\langle \nabla_{X_i}\tilde{h}_i(\mathbf{X}_i^\nu;\mathbf{U}_{(i)}^\nu,\mathbf{X}_i^\nu) - \nabla_{X_i}\tilde{h}_i(\mathbf{X}_i^\nu;\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu},\mathbf{X}_i^\nu), \mathbf{X}_i^\nu - \mathbf{X}_i^{\nu+1}\right\rangle\\
&+\left\langle \nabla_{X_i}\tilde{h}_i(\mathbf{X}_i^\nu;\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu},\mathbf{X}_i^\nu), \mathbf{X}_i^\nu - \mathbf{X}_i^{\nu+1}\right\rangle + g_i(\mathbf{X}_i^\nu) - g_i(\mathbf{X}_i^{\nu+1}) \ge 0.
\end{aligned}$$

Using Remark 7 and $\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu} = \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}$ [cf. (42)], we obtain

$$\begin{aligned}
&\left\langle \nabla_{X_i} f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}_i^\nu), \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\rangle + g_i(\mathbf{X}_i^{\nu+1}) - g_i(\mathbf{X}_i^\nu)\\
&\quad\le -\tau_{X,i}^\nu\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\|_F^2 + \left\langle \nabla_{X_i} f_i(\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu},\mathbf{X}_i^\nu) - \nabla_{X_i} f_i(\mathbf{U}_{(i)}^\nu,\mathbf{X}_i^\nu), \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\rangle,
\end{aligned}$$

which, together with (63), leads to the desired result (109), with $L_X = \sqrt{I}L_\nabla$. ■

A.4.3. Proof of Proposition 10

We begin observing that

$$G(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}) \le G(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}) + \frac{\gamma^\nu}{I}\sum_{i=1}^I \phi_i^\nu\left(G(\widetilde{\mathbf{D}}_{(i)}^\nu) - G(\mathbf{D}_{(i)}^\nu)\right) + \frac{\gamma^\nu}{I}\sum_{i=1}^I \phi_i^\nu\left(G(\mathbf{D}_{(i)}^\nu) - G(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu})\right), \tag{116}$$

due to (43) and the convexity of G [together with $\mathbf{1}^\intercal\boldsymbol{\phi}^\nu = 1$].

Invoking the descent lemma for $f_i(\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu},\bullet)$ and using $\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu} = \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}$ [cf. (42)], we get: for sufficiently large ν, say $\nu \ge \nu_0$,

$$\begin{aligned}
&U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}^{\nu+1})\\
&\le \sum_{i=1}^I\left\{f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}_i^\nu) + g_i(\mathbf{X}_i^\nu)\right\} + G(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}) + \sum_{i=1}^I\left\{g_i(\mathbf{X}_i^{\nu+1}) - g_i(\mathbf{X}_i^\nu)\right\}\\
&\quad+\sum_{i=1}^I\left\langle \nabla_{X_i} f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}_i^\nu), \mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\rangle + \frac{1}{2}\sum_{i=1}^I L_{\nabla X_i}(\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu})\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\|_F^2\\
&\overset{(a)}{\le} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}^\nu) - \sum_{i=1}^I\left\{\left(\tau_{X,i}^\nu - \frac{1}{2}L_{\nabla X_i}(\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu})\right)\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^\nu\right\|_F^2\right\}\\
&\quad+L_X||\mathbf{U}^\nu - \mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu}||_F\,||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F\\
&\overset{(b)}{\le} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}^\nu) - s_X||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F^2 + L_X||\mathbf{U}^\nu - \mathbf{1}\otimes\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu}||_F\,||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F,
\end{aligned} \tag{117}$$

where in (a) we used Proposition 14(b); and in (b) $s_X > 0$ is a constant such that $\inf_{\nu\ge\nu_0}(\tau_{X,i}^\nu - \frac{1}{2}L_{\nabla X_i}(\overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu})) \ge s_X$, for all $i = 1,\dots,I$. Note that such a constant exists because of (58) and Assumption D1.

To upper bound $U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}^\nu)$, we apply the descent lemma to $F(\bullet,\mathbf{X}^\nu)$. Recalling that $\nabla_D F(\bullet,\mathbf{X}^\nu)$ is Lipschitz continuous with constant L_{∇_D} and using (43), we get

$$\begin{aligned}
U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}},\mathbf{X}^\nu) &\le F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu},\mathbf{X}^\nu) + \frac{\gamma^\nu}{I}\Big\langle \nabla_D F(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu},\mathbf{X}^\nu), \sum_{i=1}^I \phi_i^\nu\left(\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\right)\Big\rangle\\
&\quad+\frac{L_{\nabla_D}}{2}\left(\frac{\gamma^\nu}{I}\right)^2\left\|\sum_{i=1}^I \phi_i^\nu\left(\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu\right)\right\|_F^2 + \sum_{i=1}^I g_i(\mathbf{X}_i^\nu) + G(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}),\\
&\overset{(a)}{\le} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu},\mathbf{X}^\nu) - \frac{s_D\cdot\gamma^\nu}{I}\left(||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F - \frac{I\bar{\epsilon}_\phi}{2s_D}T^\nu\right)^2\\
&\quad+\frac{\eta\cdot\gamma^\nu}{I}||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F\sum_{t=1}^\nu(\rho)^{\nu-t}||\mathbf{X}^t - \mathbf{X}^{t-1}||_F + \frac{I\bar{\epsilon}_\phi^2\gamma^\nu}{4s_D}(T^\nu)^2\\
&\quad+\frac{L_{\nabla_D}}{2}(\gamma^\nu)^2\,||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F^2 + \underbrace{\frac{\gamma^\nu}{I}\sum_{i=1}^I \phi_i^\nu\left(G(\mathbf{D}_{(i)}^\nu) - G(\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu})\right)}_{\le L_G\gamma^\nu\left\|\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu}\right\|_F},
\end{aligned} \tag{118}$$

where in (a) we used (116) and Proposition 14(a), and the Lipschitz continuity of G, due to the convexity of G (G is thus locally Lipschitz continuous) and the compactness of $\mathcal{D}$; we denoted by $L_G > 0$ the Lipschitz constant.

Combining (117) with (118) and defining $\bar{\tau}_D^\nu \triangleq s_D - \gamma^\nu \frac{IL_{\nabla_D}}{2}$, we get: for $\nu \geq \nu_0$,

$$\begin{aligned} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}, \mathbf{X}^{\nu+1}) \leq\, & U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}^{\nu}) - s_X ||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F^2 + L_X ||\mathbf{U}^\nu - \mathbf{1} \otimes \overline{\mathbf{U}}_{\boldsymbol{\phi}^\nu}||_F \, ||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F \\ & - \frac{\bar{\tau}_D^\nu \gamma^\nu}{I} \left(||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F - \frac{I\,\bar{\epsilon}_\phi}{2\,\bar{\tau}_D^\nu} T^\nu \right)^2 + \frac{I\,\bar{\epsilon}_\phi^2\,\gamma^\nu}{4\,\bar{\tau}_D^\nu} (T^\nu)^2 \\ & + \frac{\eta\,\gamma^\nu}{I} ||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F \underbrace{\sum_{t=1}^{\nu} (\rho)^{\nu-t} ||\mathbf{X}^t - \mathbf{X}^{t-1}||_F}_{\leq c_5\,(\rho)^\nu + \rho^{-1} \sum_{t=1}^{\nu} (\rho)^{\nu-t} ||\mathbf{X}^{t+1} - \mathbf{X}^t||_F} + L_G\,\gamma^\nu \left\| \mathbf{D}^\nu - \mathbf{1} \otimes \overline{\mathbf{D}}_{\boldsymbol{\phi}^\nu} \right\|_F, \end{aligned} \tag{119}$$

where $c_5 \triangleq (\rho)^{-1} ||\mathbf{X}^1 - \mathbf{X}^0||_F$. Since $\gamma^\nu \downarrow 0$, there exists an integer $\nu_1 \geq \nu_0$ and some $\bar{\tau}_D$ such that $\bar{\tau}_D^\nu \geq \bar{\tau}_D > 0$, for all $\nu \geq \nu_1$. Let $\bar{\nu}$ be any integer $\bar{\nu} \geq \nu_1$. Then, applying (119) recursively on $\nu, \nu-1, \ldots, \bar{\nu}+1, \bar{\nu}$, and using the boundedness of $\{||\widetilde{\mathbf{D}}^\nu - \mathbf{D}^\nu||_F\}_\nu$, we obtain

$$\begin{aligned} & U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}, \mathbf{X}^{\nu+1}) \\ & \leq U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\bar{\nu}}}, \mathbf{X}^{\bar{\nu}}) - s_X \sum_{l=\bar{\nu}}^{\nu} ||\mathbf{X}^{l+1} - \mathbf{X}^l||_F^2 + L_X \sum_{l=\bar{\nu}}^{\nu} ||\mathbf{U}^l - \mathbf{1} \otimes \overline{\mathbf{U}}_{\boldsymbol{\phi}^l}||_F \, ||\mathbf{X}^{l+1} - \mathbf{X}^l||_F \\ & - \frac{\bar{\tau}_D}{I} \sum_{l=\bar{\nu}}^{\nu} \gamma^l \left(||\widetilde{\mathbf{D}}^l - \mathbf{D}^l||_F - \frac{I\bar{\epsilon}_\phi}{2\,\bar{\tau}_D} T^l \right)^2 + \frac{I\bar{\epsilon}_\phi^2}{4\,\bar{\tau}_D} \sum_{l=\bar{\nu}}^{\nu} \gamma^l \left(T^l \right)^2 \\ & + c_6 \sum_{l=\bar{\nu}}^{\nu} \gamma^l (\rho)^l + c_7 \sum_{l=\bar{\nu}}^{\nu} \sum_{t=1}^{l} \gamma^l (\rho)^{l-t} \left\| \mathbf{X}^{t+1} - \mathbf{X}^t \right\|_F + L_G \sum_{l=\bar{\nu}}^{\nu} \gamma^l \left\| \mathbf{D}^l - \mathbf{1} \otimes \overline{\mathbf{D}}_{\boldsymbol{\phi}^l} \right\|_F, \end{aligned} \tag{120}$$

for some finite constants $c_6, c_7 > 0$. Using the boundedness of $\{||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F\}_\nu$ (cf. Step 2), i.e., $||\mathbf{X}^{\nu+1} - \mathbf{X}^\nu||_F \leq B_X$, for all ν and some $B_X > 0$, we can bound the double-sum term on the RHS of (120) as

$$\begin{aligned} & \sum_{l=\bar{\nu}}^{\nu} \sum_{t=1}^{l} \gamma^l (\rho)^{l-t} \left\| \mathbf{X}^{t+1} - \mathbf{X}^t \right\|_F = \sum_{l=1}^{\nu} \sum_{t=\max(\bar{\nu},l)}^{\nu} \gamma^t (\rho)^{t-l} \left\| \mathbf{X}^{l+1} - \mathbf{X}^l \right\|_F \\ & = \sum_{l=1}^{\bar{\nu}-1} \left\| \mathbf{X}^{l+1} - \mathbf{X}^l \right\|_F \sum_{t=\bar{\nu}}^{\nu} \gamma^t (\rho)^{t-l} + \sum_{l=\bar{\nu}}^{\nu} \left\| \mathbf{X}^{l+1} - \mathbf{X}^l \right\|_F \sum_{t=l}^{\nu} \gamma^t (\rho)^{t-l} \\ & \overset{(a)}{\leq} B_X \cdot \left(\max_{\bar{\nu} \leq t \leq \nu} \gamma^t \right) \cdot \frac{(1-(\rho)^{\bar{\nu}})\,(1-(\rho)^{\nu-\bar{\nu}+1})}{(1-\rho)^2} + \sum_{l=\bar{\nu}}^{\nu} \left\| \mathbf{X}^{l+1} - \mathbf{X}^l \right\|_F \sum_{t=l}^{\infty} \gamma^t (\rho)^{t-l} \\ & \leq \frac{B_X}{(1-\rho)^2} \left(\max_{t \geq \bar{\nu}} \gamma^t \right) + \sum_{l=\bar{\nu}}^{\nu} \left\| \mathbf{X}^{l+1} - \mathbf{X}^l \right\|_F \sum_{t=l}^{\infty} \gamma^t (\rho)^{t-l}, \end{aligned} \tag{121}$$

where in (a) we used the summability of $\sum_{t=l}^{\nu} \gamma^t (\rho)^{t-l}$, and the following bound

$$\sum_{l=1}^{\bar{\nu}-1} \sum_{t=\bar{\nu}}^{\nu} \gamma^t (\rho)^{t-l} \leq \left(\max_{t \geq \bar{\nu}} \gamma^t \right) \frac{(1-(\rho)^{\bar{\nu}})\,(1-(\rho)^{\nu-\bar{\nu}+1})}{(1-\rho)^2}.$$

Substituting (121) in (120), yields

$$\begin{aligned} U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu+1}}, \mathbf{X}^{\nu+1}) \leq\, & U(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\bar{\nu}}}, \mathbf{X}^{\bar{\nu}}) - s_X \sum_{l=\bar{\nu}}^{\nu} \left\| \mathbf{X}^{l+1} - \mathbf{X}^l \right\|_F^2 \\ & + \sum_{l=\bar{\nu}}^{\nu} \underbrace{\left(c_7 \sum_{t=l}^{\infty} \gamma^t (\rho)^{t-l} + L_X ||\mathbf{U}^l - \mathbf{1} \otimes \overline{\mathbf{U}}_{\boldsymbol{\phi}^l}||_F \right)}_{\triangleq Z^l} \left\| \mathbf{X}^{l+1} - \mathbf{X}^l \right\|_F \\ & - \frac{\bar{\tau}_D}{I} \sum_{l=\bar{\nu}}^{\nu} \gamma^l \left(||\widetilde{\mathbf{D}}^l - \mathbf{D}^l||_F - \frac{I\bar{\epsilon}_\phi}{2\,\bar{\tau}_D} T^l \right)^2 + \frac{I\bar{\epsilon}_\phi^2}{4\,\bar{\tau}_D} \sum_{l=\bar{\nu}}^{\nu} \gamma^l \left(T^l \right)^2 \\ & + \underbrace{\left[\frac{c_6}{1-\rho} \left((\rho)^{\bar{\nu}} - (\rho)^{\nu+1} \right) + \frac{c_7 B_X}{(1-\rho)^2} \right] \cdot \left(\max_{t \geq \bar{\nu}} \gamma^t \right)}_{\triangleq E^{\nu,\bar{\nu}}} + L_G \sum_{l=\bar{\nu}}^{\nu} \gamma^l \left\| \mathbf{D}^l - \mathbf{1} \otimes \overline{\mathbf{D}}_{\boldsymbol{\phi}^l} \right\|_F \end{aligned}$$

which complete the proof. □

A.4.4. Proof of Lemma 11

The result follows by squaring both sides of eq. (114) [in the proof of Proposition 14 (a)], using $\sum_{\nu=1}^{\infty} (T^\nu)^2 < \infty$ [cf. Proposition 14 (a)], and

$$\begin{aligned} & \lim_{\nu \to \infty} \sum_{t=1}^{\nu} \left(\sum_{l=1}^{t} (\rho)^{t-l} \left\| \mathbf{X}^l - \mathbf{X}^{l-1} \right\|_F \right)^2 \\ & \leq \lim_{\nu \to \infty} \sum_{t=1}^{\nu} \sum_{l=1}^{t} \sum_{k=1}^{t} (\rho)^{t-l} (\rho)^{t-k} \left\| \mathbf{X}^l - \mathbf{X}^{l-1} \right\|_F \left\| \mathbf{X}^k - \mathbf{X}^{k-1} \right\|_F \\ & \overset{(a)}{\leq} \frac{1}{1-\rho} \lim_{\nu \to \infty} \sum_{t=1}^{\nu} \sum_{l=1}^{t} (\rho)^{t-l} \left\| \mathbf{X}^l - \mathbf{X}^{l-1} \right\|_F^2 \overset{(b)}{<} \infty, \end{aligned}$$

where (a) follows from $ab \leq (a^2+b^2)/2, \forall a, b \in \mathbb{R}$, and (b) is due to (76) and Lemma 13. □

A.4.5. Proof of Proposition 12

(a) Using the optimality of $\widetilde{\mathbf{D}}_{(i)}^{\nu}$ defined in (8) together with convexity of G, yields

$$\begin{aligned} & \left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_1}; \mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}) + I \cdot \boldsymbol{\Theta}_{(i)}^{\nu_1} - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}), \widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \widetilde{\mathbf{D}}_{(i)}^{\nu_1} \right\rangle + G(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}) - G(\widetilde{\mathbf{D}}_{(i)}^{\nu_1}) \geq 0, \\ & \left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}; \mathbf{D}_{(i)}^{\nu_2}, \mathbf{X}_i^{\nu_2}) + I \cdot \boldsymbol{\Theta}_{(i)}^{\nu_2} - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu_2}, \mathbf{X}_i^{\nu_2}), \widetilde{\mathbf{D}}_{(i)}^{\nu_1} - \widetilde{\mathbf{D}}_{(i)}^{\nu_2} \right\rangle + G(\widetilde{\mathbf{D}}_{(i)}^{\nu_1}) - G(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}) \geq 0. \end{aligned}$$

Summing the two inequalities above while adding/subtracting inside the inner product $\nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}; \mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1})$ and using (13), yield

$$\begin{aligned} & \left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}; \mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}) - \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}; \mathbf{D}_{(i)}^{\nu_2}, \mathbf{X}_i^{\nu_2}), \widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \widetilde{\mathbf{D}}_{(i)}^{\nu_1} \right\rangle \\ & - \left\langle \nabla_D f_i(\mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}) - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu_2}, \mathbf{X}_i^{\nu_2}), \widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \widetilde{\mathbf{D}}_{(i)}^{\nu_1} \right\rangle + I \left\langle \boldsymbol{\Theta}_{(i)}^{\nu_1} - \boldsymbol{\Theta}_{(i)}^{\nu_2}, \widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \widetilde{\mathbf{D}}_{(i)}^{\nu_1} \right\rangle \\ & \geq \left\langle \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_1}; \mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}) - \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}; \mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}), \widetilde{\mathbf{D}}_{(i)}^{\nu_1} - \widetilde{\mathbf{D}}_{(i)}^{\nu_2} \right\rangle \geq \tau_{D,i}^{\nu_1} \, ||\widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \widetilde{\mathbf{D}}_{(i)}^{\nu_1}||_F^2, \end{aligned} \tag{122}$$

where the second inequality follows from the $\tau_{D,i}^{\nu_1}$-strong convexity of $\tilde{f}_i(\bullet; \mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1})$ [cf. Remark 7]. To bound the first term on the LHS of the above inequality, let us use the expression (15) of $\tilde{f}_i$, and write

$$\begin{aligned}&\nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}; \mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}) - \nabla_D \tilde{f}_i(\widetilde{\mathbf{D}}_{(i)}^{\nu_2}; \mathbf{D}_{(i)}^{\nu_2}, \mathbf{X}_i^{\nu_2}) \\ &\quad = \nabla_D f_i(\mathbf{D}_{(i)}^{\nu_1}, \mathbf{X}_i^{\nu_1}) - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu_2}, \mathbf{X}_i^{\nu_2}) + \tau_{D,i}^{\nu_1}\left(\mathbf{D}_{(i)}^{\nu_2} - \mathbf{D}_{(i)}^{\nu_1}\right) + (\tau_{D,i}^{\nu_1} - \tau_{D,i}^{\nu_2})\left(\widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \mathbf{D}_{(i)}^{\nu_2}\right).\end{aligned} \tag{123}$$

Substituting (123) in (122), we get

$$\begin{aligned}&\left\|\widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \widetilde{\mathbf{D}}_{(i)}^{\nu_1}\right\|_F \\ &\le \frac{|\tau_{D,i}^{\nu_1} - \tau_{D,i}^{\nu_2}|}{\tau_{D,i}^{\nu_1}}\left\|\widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \mathbf{D}_{(i)}^{\nu_2}\right\|_F + \left\|\mathbf{D}_{(i)}^{\nu_1} - \mathbf{D}_{(i)}^{\nu_2}\right\|_F + \frac{I}{\tau_{D,i}^{\nu_1}}\left\|\mathbf{\Theta}_{(i)}^{\nu_1} - \mathbf{\Theta}_{(i)}^{\nu_2}\right\|_F \\ &\overset{(a)}{\le} \frac{|\tau_{D,i}^{\nu_1} - \tau_{D,i}^{\nu_2}|}{\tau_{D,i}^{\nu_1}}\left\|\widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \mathbf{D}_{(i)}^{\nu_2}\right\|_F + \|\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_2}} - \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_1}}\|_F + \left\|\mathbf{D}_{(i)}^{\nu_1} - \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_1}}\right\|_F + \left\|\mathbf{D}_{(i)}^{\nu_2} - \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_2}}\right\|_F \\ &\quad + \frac{I}{\tau_{D,i}^{\nu_1}}\left\|\mathbf{\Theta}_{(i)}^{\nu_1} - \frac{1}{I}\sum_{i=1}^{I}\nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_1}}, \mathbf{X}_i^{\nu_1})\right\|_F \\ &\quad + \frac{I}{\tau_{D,i}^{\nu_1}}\left\|\mathbf{\Theta}_{(i)}^{\nu_2} - \frac{1}{I}\sum_{i=1}^{I}\nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_2}}, \mathbf{X}_i^{\nu_2})\right\|_F \\ &\quad + \frac{L_{\nabla_D}}{\tau_{D,i}^{\nu_1}}\left\|(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_1}}, \mathbf{X}^{\nu_1}) - (\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_2}}, \mathbf{X}^{\nu_2})\right\|,\end{aligned} \tag{124}$$

where (a) holds by add/subtracting average quantities $\frac{1}{I}\sum_{i=1}^{I}\nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu})$ and $\overline{\mathbf{D}}_{(i)}^{\nu}$, triangle inequality, and invoking the Lipschitz continuity bound (63). By the compactness of $\mathcal{D}$, we have $||\widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \mathbf{D}_{(i)}^{\nu_2}||_F \le B_D$, for some finite $B_D > 0$. Furthermore, by Assumption D, $\tau_{D,i}^{\nu}$ is convergent to some $\tau_{D,i}^{\infty} > 0$ and there exists a sufficiently small $\tilde{s}_D > 0$ such that $\tilde{s}_D \le \tau_{D,i}^{\nu} \le \tilde{s}_D^{-1}$, for all ν and i. Thus (124) gives

$$\left\|\widetilde{\mathbf{D}}_{(i)}^{\nu_2} - \widetilde{\mathbf{D}}_{(i)}^{\nu_1}\right\|_F \le \left(\frac{L_{\nabla_D}}{\tilde{s}_D} + 1\right)\left\|(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_1}}, \mathbf{X}^{\nu_1}) - (\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu_2}}, \mathbf{X}^{\nu_2})\right\| + \tilde{T}_i^{\nu_1} + \tilde{T}_i^{\nu_2} \tag{125}$$

with $\tilde{T}_i^{\nu} \triangleq \frac{I}{\tilde{s}_D}||\widetilde{\mathbf{\Theta}}_i^{\nu} - \frac{1}{I}\sum_{i=1}^{I}\nabla_D f_i(\overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}, \mathbf{X}_i^{\nu})||_F + ||\mathbf{D}_{(i)}^{\nu} - \overline{\mathbf{D}}_{\boldsymbol{\phi}^{\nu}}||_F + \frac{B_D}{\tilde{s}_D}|\tau_{D,i}^{\nu} - \tau_{D,i}^{\infty}|$. Convergence of $\tau_{D,i}^{\nu}$ (Assumption D2), Lemma 11 and Proposition 5 yield $\tilde{T}_i^{\nu} \to 0$ as $\nu \to \infty$. Summing (125) leads to the desired result, with $L_D = I(\frac{L_{\nabla_D}}{\tilde{s}_D} + 1)$ and $\tilde{T}^{\nu} \triangleq \sum_{i=1}^{I}\tilde{T}_i^{\nu}$.

It is clear that the claim also holds when (14) is chosen for $\tilde{f}_i$ [specifically, trivial extension is to modify the RHS of (123) and following a similar steps as in the rest of the proof]; we omit further details.

(b) We prove (86) when $\tilde{h}_i$ is given by (17); we leave the proof under (16) to the reader, since it is almost identical to that under (17).

Invoking optimality of each $\mathbf{X}_i^{\nu}$ and $\mathbf{X}_i^{\nu+1}$ defined in (11) while using the strong convexity of $\tilde{h}_i(\bullet; \mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu})$ and g_i's, it is not difficult to show that the following holds:

$$\begin{aligned}&(\tau_{X,i}^{\nu} + \mu_i)\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}\right\|_F^2 \le \\ &\qquad \left\langle \nabla_{X_i}\tilde{h}_i(\mathbf{X}_i^{\nu}; \mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu}) - \nabla_{X_i}\tilde{h}_i(\mathbf{X}_i^{\nu}; \mathbf{U}_{(i)}^{\nu-1}, \mathbf{X}_i^{\nu-1}), \mathbf{X}_i^{\nu} - \mathbf{X}_i^{\nu+1}\right\rangle.\end{aligned}$$

Using (17), the definition $\Upsilon_i^{\nu}(\mathbf{X}_i) \triangleq \tau_{X,i}^{\nu}\mathbf{X}_i - \nabla_{X_i} f_i(\mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i)$, and the Cauchy-Schwarz inequality, the above inequality yields

$$\begin{aligned}(\tau_{X,i}^{\nu} + \mu_i)\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}\right\|_F^2 &\le \left\|\Upsilon_i^{\nu}(\mathbf{X}_i^{\nu}) - \Upsilon_i^{\nu}(\mathbf{X}_i^{\nu-1})\right\|_F\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}\right\|_F \\ &\quad + \left\|\nabla_{X_i} f_i(\mathbf{U}_{(i)}^{\nu}, \mathbf{X}_i^{\nu-1}) - \nabla_{X_i} f_i(\mathbf{U}_{(i)}^{\nu-1}, \mathbf{X}_i^{\nu-1})\right\|_F\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}\right\|_F \\ &\quad + |\tau_{X,i}^{\nu} - \tau_{X,i}^{\nu-1}|\left\|\mathbf{X}_i^{\nu} - \mathbf{X}_i^{\nu-1}\right\|_F\left\|\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}\right\|_F.\end{aligned} \tag{126}$$

Following the same steps used to prove (61), it is not difficult to check that, under Assumption D1, $||\Upsilon_i^{\nu}(\mathbf{X}_i^{\nu}) - \Upsilon_i^{\nu}(\mathbf{X}_i^{0})||_F \le \tau_{X,i}^{\nu}||\mathbf{X}_i^{\nu} - \mathbf{X}_i^{0}||_F$. Using in (126) this bound together with the Lipschitz continuity of $\nabla_{X_i} f_i(\bullet, \mathbf{X}_i^{\nu-1})$ [cf. Remark 8] and summing over i, yield

$$\begin{aligned}||\mathbf{X}^{\nu+1} - \mathbf{X}^{\nu}||_F^2 &= \sum_{i=1}^{I}||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}||_F^2 \\ &\le \sum_{i=1}^{I}\frac{\tau_{X,i}^{\nu} + |\tau_{X,i}^{\nu} - \tau_{X,i}^{\nu-1}|}{\tau_{X,i}^{\nu} + \mu_i}||\mathbf{X}_i^{\nu} - \mathbf{X}_i^{\nu-1}||_F\,||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}||_F \\ &\quad + \sum_{i=1}^{I}\frac{L_{\nabla}}{\tau_{X,i}^{\nu} + \mu_i}||\mathbf{U}_{(i)}^{\nu} - \mathbf{U}_{(i)}^{\nu-1}||_F\,||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}||_F.\end{aligned} \tag{127}$$

Define

$$p_X^{\nu} \triangleq \max_i \frac{\tau_{X,i}^{\nu} + |\tau_{X,i}^{\nu} - \tau_{X,i}^{\nu-1}|}{\tau_{X,i}^{\nu} + \mu_i}, \quad \text{and} \quad q_X^{\nu} \triangleq \max_i \frac{L_{\nabla}}{\tau_{X,i}^{\nu} + \mu_i}. \tag{128}$$

Note that $0 < p_X^{\nu}, q_X^{\nu} < \infty$. Then, (127) becomes

$$\begin{aligned}||\mathbf{X}^{\nu+1} - \mathbf{X}^{\nu}||_F^2 &\le p_X^{\nu}\sum_{i=1}^{I}||\mathbf{X}_i^{\nu} - \mathbf{X}_i^{\nu-1}||_F\,||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}||_F + q_X^{\nu}\sum_{i=1}^{I}||\mathbf{U}_{(i)}^{\nu} - \mathbf{U}_{(i)}^{\nu-1}||_F\,||\mathbf{X}_i^{\nu+1} - \mathbf{X}_i^{\nu}||_F \\ &\le p_X^{\nu}||\mathbf{X}^{\nu} - \mathbf{X}^{\nu-1}||_F\,||\mathbf{X}^{\nu+1} - \mathbf{X}^{\nu}||_F + q_X^{\nu}||\mathbf{U}^{\nu} - \mathbf{U}^{\nu-1}||_F\,||\mathbf{X}^{\nu+1} - \mathbf{X}^{\nu}||_F,\end{aligned}$$

where in the last inequality we used $\sum_i a_i b_i \le ||\mathbf{a}|| \cdot ||\mathbf{b}||$. Therefore,

$$||\mathbf{X}^{\nu+1} - \mathbf{X}^{\nu}||_F \le p_X^{\nu}||\mathbf{X}^{\nu} - \mathbf{X}^{\nu-1}||_F + q_X^{\nu}||\mathbf{U}^{\nu} - \mathbf{U}^{\nu-1}||_F. \tag{129}$$

If, in addition, Assumption D2 holds, then it follows from (128) that there exists a sufficiently large $\nu_X > 0$ such that $p_X^{\nu} \in (0, \delta_0)$, for all $\nu \ge \nu_X$ and some $\delta_0 \in (0, 1)$. □

References

University of Southern California, Signal and image processing institute. Volume 3: Miscellaneous image database, 1997. Available online: `http://sipi.usc.edu/database/database.php?volume=misc`.

M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11): 4311–4322, November 2006.

A. Aravkin, S. Becker, V. Cevher, and P. Olsen. A variational approach to stable principal component pursuit. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pages 32–41, Arlington, Virginia, United States, 2014. AUAI Press. ISBN 978-0-9749039-1-0.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 1997.

P. Bianchi and J. Jakubowicz. Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58(2): 391–405, February 2013.

T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E. Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 23:1–71, February 2017.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, June 2011.

P. Chainais and C. Richard. Learning a common dictionary over a sensor network. In *Proceedings of the 2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 133–136, Saint Martin, French West Indies, France, December 2013.

J. Chen, Z. J. Towfic, and A. H. Sayed. Dictionary learning over distributed models. *IEEE Transactions on Signal Processing*, 63(4):1001–1016, February 2015.

S. Chouvardas, Y. Kopsinis, and S. Theodoridis. An online algorithm for distributed dictionary learning. In *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3292–3296, Brisbane, Queensland, Australia, April 2015.

A. Daneshmand, F. Facchinei, V. Kungurtsev, and G. Scutari. Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(13):3914–3929, August 2015.

A. Daneshmand, G. Scutari, and F. Facchinei. Distributed dictionary learning. In *Proceedings of the 50th Asilomar Conference on Signals, Systems and Computers*, pages 1001–1005, November 2016.

P. Di Lorenzo and G. Scutari. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, June 2016.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, December 2006.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, June 2010.

F. Facchinei, G. Scutari, and Simone Sagratella. Parallel selective algorithms for nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(7):1874–1889, April 2015.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity*. CRC Press, Taylor & Francis Group, 2015.

M. Hong, D. Hajinezhad, and M. M. Zhao. Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1529–1538, Sydney, Australia, August 2017.

P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, December 2004.

Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1697–1704, Colorado Springs, CO, USA, June 2011.

D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedins of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 482–491, Cambridge, MA, USA, October 2003.

J. Kim. *High-radix Interconnection Networks*. PhD thesis, Stanford, CA, USA, 2008.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, December 2010.

J. Liang, M. Zhang, X. Zeng, and G. Yu. Distributed dictionary learning for sparse representation in sensor networks. *IEEE Transactions on Image Processing*, 23(6):2528–2541, June 2014.

S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1): 24–45, January 2004.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1033–1040, Lake Tahoe, NV, December 2008.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, January 2010.

A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, April 2010.

H. Raja and W. U. Bajwa. Cloud K-SVD: Computing data-adaptive representations in the cloud. In *Proceedings of 51st Annual Allerton Conference*, pages 1474–1481, Allerton House, UIUC, Illinois, USA, October 2013.

M. Razaviyayn, M. Hong, Z. Luo, and J. Pang. Parallel successive convex approximation for nonsmooth nonconvex optimization. In *NIPS*, 2014a.

M. Razaviyayn, H. W. Tseng, and Z. Q. Luo. Dictionary learning for sparse representation: Complexity and algorithms. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5247–5251, Florence, Italy, May 2014b.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, August 2010.

R.T. Rockafellar and J.B. Wets. *Variational Analysis*. Springer, 1998.

G. Scutari and Y. Sun. Parallel and distributed successive convex approximation methods for big-data optimization. In F. Facchinei and J.-S. Pang, editors, *Multi-agent Optimization*, chapter 3, pages 141–308. Lecture Notes in Mathematics 2224, Springer, 2018.

G. Scutari and Y. Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, to appear 2019.

G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang. Decomposition by partial linearization: Parallel optimization of multiuser systems. *IEEE Transaction on Signal Processing*, 62:641–656, February 2014.

H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 6(99):1015–1034, July 2008.

N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *Proceedings of the Learning Theory: 18th Annual Conference on Learning Theory (COLT)*, pages 545–560, Bertinoro, Italy, June 2005.

Y. Sun, G. Scutari, and D. Palomar. Distributed nonconvex multiagent optimization over time-varying networks. In *Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 788–794, November 2016.

K. K. Sung. *Learning and Example Selection for Object and Pattern Recognition*. PhD thesis, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, MIT, Cambridge, MA, 1996.

T. Tatarenko and B. Touri. Non-convex distributed optimization. *IEEE Trans. on Automatic Control*, 62:3744–3757, August 2017.

I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2): 27–38, March 2011.

M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, June 2016.

H. T. Wai, T. H. Chang, and A. Scaglione. A consensus-based decentralized algorithm for non-convex optimization with application to dictionary learning. In *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3546–3550, Brisbane, Queensland, Australia, April 2015.

H. T. Wai, J. Lafond, A. Scaglione, and E. Moulines. Decentralized frankwolfe algorithm for convex and nonconvex problems. *IEEE Transaction on Automatic Control*, 62:5522–5537, November 2017.

L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 63–70, Los Angeles, CA, April 2005.

L. Xiao, S. Boyd, and S. J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, January 2007.

M. Zhao, Q. Shi, and M. Hong. A distributed algorithm for dictionary learning over networks. In *Proceedings of 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 505–509, December 2016.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, March 2005.

Journal of Machine Learning Research 20 (2019) 1-81 Submitted 10/17; Revised 4/19; Published 9/19

Nonparametric Bayesian Aggregation for Massive Data

Zuofeng Shang ZUOFENGSHANG@GMAIL.COM
Department of Mathematical Sciences
New Jersey Institute of Technology
Newark, NJ 07102, USA

Botao Hao HAO22@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47906, USA

Guang Cheng CHENGG@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47906, USA

Editor: Francis Bach

Abstract

We develop a set of scalable Bayesian inference procedures for a general class of nonparametric regression models. Specifically, nonparametric Bayesian inferences are separately performed on each subset randomly split from a massive dataset, and then the obtained local results are aggregated into global counterparts. This aggregation step is explicit without involving any additional computation cost. By a careful partition, we show that our aggregated inference results obtain an oracle rule in the sense that they are equivalent to those obtained directly from the entire data (which are computationally prohibitive). For example, an aggregated credible ball achieves desirable credibility level and also frequentist coverage while possessing the same radius as the oracle ball.

Keywords: Credible region, divide-and-conquer, Gaussian process prior, linear functional, nonparametric Bayesian inference

1. Introduction

With rapid development in modern technology, massive data sets are becoming more and more common. An important feature of massive data is their large volume which hinders applications of traditional statistical methods. For example, due to huge data amount and limited CPU memory, it is often impossible to process the entire data in a single machine. In the parallel computing environment, a common practice is to distribute massive data to multiple processors, and then aggregate local results in an efficient way. A series of frequentist methods such as Kleiner et al. (2011); McDonald et al. (2010); Zhang et al. (2015a); Zhao et al. (2016) have been proposed in this Divide-and-Conquer (D&C) framework.

In Bayesian community, there are quite a few computational or methodological works developed for massive data such as scalable algorithms for Bayesian variable selection Boom et al. (2015); Wang et al. (2014) and scalable posterior sampling in parametric models Wang and Dunson (2013); Wang et al. (2015). Theoretical guarantees of D&C methods have been

©2019 Zuofeng Shang, Botao Hao, Guang Cheng.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-641.html.

recently obtained in robust estimation Minsker et al. (2017), posterior interval estimation Srivastava et al. (2018), credible sets of signal in Gaussian white noise Szabó and van Zanten (2019); Szabo and van Zanten (2018). Rather, the present paper puts focus on uncertainty quantification of the model parameter in general nonparametric regression, primarily in theoretical aspects. For instance, how to aggregate individual posterior means into a global one that maintains frequentist optimality? How to aggregate individual credible balls into a global one with a minimal possible radius? And how many divisions and what kind of priors should be chosen to guarantee Bayesian and frequentist validity of the aggregated ball? We attempt to address these questions in a univariate nonparametric regression setup.

Specifically, we develop a set of aggregation procedures in Bayesian nonparametric regression. As a first step, nonparametric Bayesian regression is separately fitted based on each subsample randomly split from a massive dataset. A variety of finite sample valid credible balls (credible intervals) for regression functions (their linear functionals Rivoirard et al. (2012), e.g., local values) are then constructed from each individual posterior distribution based on MCMC. In the second step, we aggregate these credible balls (credible intervals) into global counterparts analytically without involving any additional computation. For example, the center of an aggregated ball is obtained by weighted averaging Fourier coefficients of all individual (approximate) posterior modes, while the radius is given through an explicit formula on individual radii. A notable advantage of this distributed strategy is its dramatically faster computational speed, and this computational advantage becomes more obvious as data size grows.

Our aggregation procedures are proven to obtain an oracle rule in the sense that they are equivalent to those obtained directly from the entire data, i.e., called as oracle results which are computationally prohibitive in practice. For example, our aggregated posterior means are proven to achieve optimal estimation rate, and our aggregated credible ball achieves desirable credibility level and also frequentist coverage while possessing asymptotically the same radius as the oracle ball. These oracle results hold when the assigned Gaussian process priors in each subset are properly chosen and the number of subsets does not grow too fast. A fundamental theory underlying Bayesian aggregation is a *uniform* version of nonparametric Gaussian approximation theorem, also called as Bernstein-von Mises theorem. Developed based on our recent work Shang and Cheng (2017), this theory states that a sequence of individual posterior distributions converge to Gaussian processes uniformly over the number of subsets.

The rest of this paper is organized as follows. Section 3 describes our Bayesian nonparametric model with a Gaussian process prior, based on which our main results are developed in Section 4. Specifically, a uniform nonparametric Gaussian approximation theorem is established in Section 4.1, and all the Bayesian aggregation procedures together with their theoretical guarantee are provided in Sections 4.2–4.6. Section 5 provides a simulation study to justify our methods. Section 6 applies the proposed procedures to a real dataset of large size. Main proofs are provided in Appendix. Other results and additional plots are given in a supplementary document Shang and Cheng.

2. Nonparametric Bayesian Aggregation: An Illustration

In this section, we provide a concrete example to demonstrate the intuition of our nonparametric Bayesian aggregation procedure. Our example is based on the special uniform design and periodic Sobolev space which makes our aggregation procedure explicit and easy to understand. Section 2.1 describes our nonparametric Bayesian model, and Section 2.2 demonstrates our algorithm and its numeric performance. General aggregation procedures will be proposed in Sections 3 and 4 with asymptotic properties investigated as well.

2.1. Nonparametric Bayesian model

Suppose that we observe the data $Z_i = (Y_i, X_i)$, $i = 1, \ldots, N$, generated from the following Gaussian regression model with uniform design

$$Y_i|f, X_i \sim N(f(X_i), 1), \quad X_1, \ldots, X_N \overset{iid}{\sim} Unif[0,1]. \tag{1}$$

Randomly split $\{1, 2, \ldots, N\}$ into s subsets $I_1, I_2, \ldots, I_s$ with $|I_1| = \cdots = |I_s| = n$ (so $N = ns$). Denote $\mathbf{D}_j = \{Z_i | i \in I_j\}$ the j-th subsample for $j = 1, \ldots, s$ and $\mathbf{D} = \cup_{j=1}^s \mathbf{D}_j$ the entire sample.

Suppose that f belongs to an m-order periodic Sobolev space $S_0^m[0,1]$ where $S_0^m[0,1]$ is the collection of all functions on $[0,1]$ of the form

$$f(x) = \sqrt{2}\sum_{k=1}^{\infty} f_k \cos(2\pi kx) + \sqrt{2}\sum_{k=1}^{\infty} g_k \sin(2\pi kx) \tag{2}$$

with real coefficients f_k, g_k satisfying

$$\sum_{k=1}^{\infty} (f_k^2 + g_k^2)(2\pi k)^{2m} < \infty. \tag{3}$$

Here, $m > 1/2$ is a constant describing the smoothness of the functions. Wahba (1990) Wahba (1990) introduced a Gaussian process (GP) prior on f which has an interesting smoothing spline interpretation. Specifically, she assumed that the coefficients f_k, g_k in (2) are independent and normally distributed as follows:

$$f_k, g_k \sim N\left(0, [(2\pi k)^{2m+\beta} + n\lambda(2\pi k)^{2m}]^{-1}\right), \quad k = 1, 2, \ldots, \tag{4}$$

where $\beta > 1$ and $\lambda \geq 0$ are predecided constants. In particular, β represents the "relative smoothness" of the prior to the parameter space and λ represents the amount of rescaling. Rescaling priors are also considered by Szabó and van Zanten (2019); Szabo and van Zanten (2018) for constructing credible sets of signals in Gaussian white noise. It can be examined that if f satisfies (2) and (4), then f is a Gaussian process with mean zero and isotropic covariance function

$$K_0(x, x') = 2\sum_{k=1}^{\infty} \frac{\cos(2\pi k(x - x'))}{(2\pi k)^{2m+\beta} + n\lambda(2\pi k)^{2m}}, \quad x, x' \in [0,1]. \tag{5}$$

Wahba Wahba (1990) showed that the above GP prior (4) generates a posterior distribution corresponding to a penalized likelihood function (with λ the penalty parameter). This

provides a Bayesian interpretation for smoothing splines. Below we provide some details to justify this argument.

Let Π_λ denote the probability distribution of f under (4). To derive the posterior distribution, we need to find the "prior density" of f. Unlike the parametric settings where the prior densities are Radon-Nikodym (RN) derivatives w.r.t. Lebesgue measure, in the current infinite-dimensional setting it is impossible to do so since there is no Lebesgue measure on $S_0^m[0,1]$ (see Hunt et al. (1992)). Instead, we need to characterize the prior density of f as an RN derivative w.r.t. other kinds of measures such as Gaussian measure. Following Wahba Wahba (1990), Π_λ and $\Pi \equiv \Pi_0$ (corresponding to $\lambda = 0$) are equivalent probability measures, and the RN derivative of Π_λ w.r.t. Π is

$$\begin{aligned}
\frac{d\Pi_\lambda}{d\Pi}(f) &= \prod_{k=1}^{\infty}\left(1 + n\lambda(2\pi k)^{-\beta}\right)^{-1} \times \exp\left(-\frac{n\lambda}{2}\sum_{k=1}^{\infty}(f_k^2 + g_k^2)(2\pi k)^{2m}\right) \\
&= \prod_{k=1}^{\infty}\left(1 + n\lambda(2\pi k)^{-\beta}\right)^{-1} \times \exp\left(-\frac{n\lambda}{2}\int_0^1 f^{(m)}(x)^2 dx\right) \\
&= \prod_{k=1}^{\infty}\left(1 + n\lambda(2\pi k)^{-\beta}\right)^{-1} \times \exp\left(-\frac{n\lambda}{2}J(f)\right),
\end{aligned} \tag{6}$$

where $J(f) = \int_0^1 f^{(m)}(x)^2 dx$. Note that $\prod_{k=1}^{\infty}\left(1 + n\lambda(2\pi k)^{-\beta}\right)^{-1}$ converges thanks to $\beta > 1$ so that (6) is a valid expression. (6) provides an expression for the prior density of f, which induces the following posterior distribution for f given subsample j:

$$\begin{aligned}
dP(f|\mathbf{D}_j) &\propto P(\mathbf{D}_j|f)d\Pi_\lambda(f) \\
&\propto \exp\left(-\frac{1}{2}\sum_{i\in I_j}(Y_i - f(X_i))^2 - \frac{n\lambda}{2}J(f)\right)d\Pi(f), \; j = 1, \ldots, s.
\end{aligned} \tag{7}$$

Recall that I_j indexes the j-th subsample. The right hand-side of (7) corresponds to penalized likelihood function $\ell_j(f) = -\frac{1}{2n}\sum_{i\in I_j}(Y_i - f(X_i))^2 - \frac{\lambda}{2}J(f)$ which has been well studied in smoothing spline literature (Wahba (1990)). Theoretically, we recommend to choose $\lambda \asymp N^{-\frac{2m}{2m+\beta}}$ which will be proven to yield optimal Bayesian inference; see Sections 3 and 4. The duality between the posterior and smoothing spline, i.e., (7), enables us to easily choose λ for practical use, e.g., GCV considered by Wahba (1990).

2.2. Nonparamtric Bayesian Aggregation

First of all, we calculate $\breve{f}_{j,n} = E\{f|\mathbf{D}_j\}$, $j = 1, \ldots, s$, the posterior means based on individual posterior distributions (7). Then we construct a $(1-\alpha)$-th credible ball centering at $\breve{f}_{j,n}$ with radius $r_{j,n}(\alpha)$. That is, $r_{j,n}(\alpha) > 0$ such that $P(f \in S_0^m[0,1] : \|f - \breve{f}_{j,n}\|_{L^2} \leq r_{j,n}(\alpha)|\mathbf{D}_j) = 1 - \alpha$, where $\|\cdot\|_{L^2}$ is the usual L^2-norm, i.e., $\|f\|_{L^2} = \sqrt{\int_0^1 f(x)^2 dx}$. In practice, $\breve{f}_{j,n}$ and $r_{j,n}(\alpha)$ can be both estimated by the posterior samples. For instance, generate M independent samples $f_{j1}, \ldots, f_{jM}$ from (7); estimate $\breve{f}_{j,n}$ by their average and estimate $r_{j,n}(\alpha)$ by the $(1-\alpha)$-th percentile of $\|f_{jl} - \breve{f}_{j,n}\|_{L^2}$ for $1 \leq l \leq M$. We postpone the computational details of the sampling procedure to Section 8.5.

We next present a concrete aggregation scheme (procedures (1)–(3) below) to construct a credible ball based on these individual results $\{\breve{f}_{j,n}, r_{j,n}(\alpha)\}_{j=1}^s$. Specifically, an aggregated

credible ball for f, denoted $R_N(\alpha)$, is constructed with its center/radius obtained through *weighted* averaging the individual centers/radii. Unlike simple averaging commonly used in frequentist setting (see Zhang et al. (2015a)), our procedures for posterior mean aggregation and radius aggregation are weighted averaging with weights $w_{s,N,\lambda,k}$ defined in (10). These weights are used to calibrate the prior effect such that the aggregation procedure can have satisfactory asymptotic property. The details of our procedure are demonstrated as follows:

1. *Posterior mean aggregation.* For j-th subsample and $k \geq 1$, find

$$\breve{f}_{j,n,k} = \sqrt{2}\int_0^1 \breve{f}_{j,n}(x)\cos(2\pi kx)dx,\ \breve{g}_{j,n,k} = \sqrt{2}\int_0^1 \breve{f}_{j,n}(x)\sin(2\pi kx)dx, \tag{8}$$

where $\breve{f}_{j,n}$ is the posterior mean based on subsample j. Then we aggregate these quantities through the following formulas:

$$\breve{f}_{N,\lambda,k} = \sum_{j=1}^{s}\breve{f}_{j,n,k}/s,\ \breve{g}_{N,\lambda,k} = \sum_{j=1}^{s}\breve{g}_{j,n,k}/s. \tag{9}$$

In the end, we let

$$\breve{f}_{N,\lambda}(x) = \sum_{k=1}^{\infty} w_{s,N,\lambda,k}\left\{\breve{f}_{N,\lambda,k}\sqrt{2}\cos(2\pi kx) + \breve{g}_{N,\lambda,k}\sqrt{2}\sin(2\pi kx)\right\}, \tag{10}$$

where $w_{s,N,\lambda,k} = \frac{s(2\pi k)^{2m+\beta}+N(1+\lambda(2\pi k)^{2m})}{(2\pi k)^{2m+\beta}+N(1+\lambda(2\pi k)^{2m})}$ for $k \geq 1$.

2. *Posterior radius aggregation.* Aggregate the radii $r_{j,n}(\alpha)$ through the following formula:

$$r_N(\alpha) = \sqrt{A_{N,s}\left(\frac{1}{s}\sum_{j=1}^{s} r_{j,n}(\alpha)^2\right) + B_{N,s}}, \tag{11}$$

where

$$\begin{aligned} A_{N,s} &= \sqrt{C_2/D_2}\,s^{-\frac{4m+2\beta-1}{2(2m+\beta)}}, \\ B_{N,s} &= \left(2C_1 - 2D_1\sqrt{C_2/D_2}\,s^{-\frac{1}{2(2m+\beta)}}\right)N^{-\frac{2m+\beta-1}{2m+\beta}}, \\ C_k &= \int_0^\infty (1+(2\pi x)^{2m}+(2\pi x)^{2m+\beta})^{-k}dx, \quad k=1,2, \\ D_k &= \int_0^\infty (1+(2\pi x)^{2m})^{-k}dx, \quad k=1,2. \end{aligned} \tag{12}$$

3. *Aggregated credible ball*:

$$R_N(\alpha) = \{f \in S_0^m[0,1] : \|f - \breve{f}_{N,\lambda}\|_{L^2} \leq r_N(\alpha)\}. \tag{13}$$

Algorithms based on weighted averaging have been proposed in numerous computational aspects. For instance, Huang and Gelman (2005); Neiswanger et al. (2013); Scott et al. (2016) proposed computational procedures for efficiently aggregating local MCMC samples in which

the aggregation steps involve proper weight averaging. Such algorithms are particularly useful to produce MCMC samples from the oracle posterior which can be used for various inferential purposes, e.g., estimation and testing. The present paper focuses on inferences, e.g., construction of credible balls, in a special class of nonparametric regression models, and has more extensive theoretical guarantees.

In practice, one can approximate the integral (8) through discretization; see Section 8.5. Theorem 3 will show that $R_N(\alpha)$ given in (13) asymptotically covers $1-\alpha$ mass of the posterior based on the full data set and includes the true function with probability approaching one. More theoretical study on $R_N(\alpha)$ such as its center and radius can be found in Sections 4.2 and 4.3. Note that these sections present an aggregation procedure in a more general context, which covers (13) as a special case.

A toy simulation study was carried out to examine the proposed procedures (1)–(3). Specifically, we examine the computing time and coverage probability (CP) of $R_N(\alpha)$ for various choices of s. The CP is defined as the relative frequency of the sets that cover the truth. We choose $m = \beta = 2$ in our GP prior (4). Results are summarized in Figure 1. Plot (a) displays the true function f_0 under which data were generated. Plot (b) displays how the CP varies as $\gamma := \log(s)/\log(N)$. Plot (c) displays that the computing time decreases when γ increases. There seems to be a transition for CP vs. γ, i.e., CP is uniformly close to one when $0 \leq \gamma < 0.3$ and approaches zero when $\gamma > 0.4$. In conclusion, $R_N(\alpha)$ possesses both satisfactory frequentist coverage and computational efficiency when $\gamma \approx 0.2$. Other choices of γ either lower CP or slow down the computing. Thus, under a proper choice of s, our aggregation procedure can maintain good statistical properties and reduce computing burden at the same time. Careful readers may have noticed that the CP approaches one rather than the credibility level $(1-\alpha)$. This issue can be addressed by a modified aggregated set proposed in Section 4.4. More comprehensive simulation results are provided in Section 5 to examine various aggregation procedures such as the pointwise credible intervals.

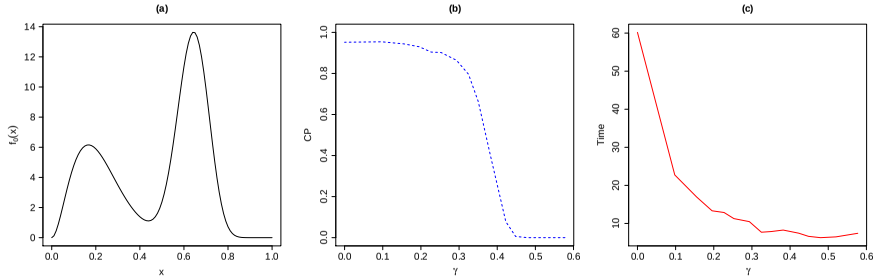


Figure 1. *Examination of our aggregation procedures (1)–(3). Results are based on $N = 1200$ observations generated from (1) and a GP prior (4) with $m = \beta = 2$ and $\lambda = N^{-2/3}$. (a) True regression function $f_0(x) = 2.4\beta_{30,17}(x) + 1.6\beta_{3,11}(x)$, where $\beta_{a,b}$ is the probability density function for Beta(a,b). (b) Coverage probability (CP) of $R_N(0.95)$ vs. γ. (c) Computing time (in seconds) of $R_N(0.95)$ vs. γ.*

3. A Nonparametric Bayesian Framework Based on General Design and Space

In this section, we introduce a more general Bayesian nonparametric framework based on general design and function space under which the aggregation results will be obtained. Suppose that the data $\{Y_i, X_i\}_{i=1}^N$ follow a nonparametric regression model:

$$Y_i|f, X_i \overset{ind.}{\sim} N(f(X_i), \sigma^2), \quad X_1, \ldots, X_N \overset{iid}{\sim} \pi(x), \tag{14}$$

where $\pi(\cdot)$ is a probability density on $\mathbb{I} = (0,1)$, and f belongs to an m-order Sobolev space $S^m(\mathbb{I})$:

$$S^m(\mathbb{I}) = \{f \in L^2(\mathbb{I}) | f^{(0)}, f^{(1)}, \ldots, f^{(m-1)} \text{are abs. cont. and } f^{(m)} \in L^2(\mathbb{I})\}. \tag{15}$$

In particular, $S_0^m[0,1]$ is a proper subset of $S^m(\mathbb{I})$. Throughout, we let $m > 1/2$ such that $S^m(\mathbb{I})$ is a reproducing kernel Hilbert space (RKHS). For technical convenience, assume $\sigma^2 = 1$ and $0 < \inf_{x\in\mathbb{I}} \pi(x) \le \sup_{x\in\mathbb{I}} \pi(x) < \infty$. When σ^2 is unknown, our approach can still be applied with σ^2 replaced by its consistent estimate.

For any $f, g \in S^m(\mathbb{I})$, define $V(f,g) = E\{f(X)g(X)\}$ and $J(f,g) = \int_0^1 f^{(m)}(x)g^{(m)}(x)dx$. Following Shang et al. (2013), there exists a sequence of eigenfunctions $\varphi_1, \varphi_2, \ldots \in S^m(\mathbb{I})$ and a sequence of eigenvalues $0 = \rho_1 = \rho_2 = \cdots = \rho_m < \rho_{m+1} \le \rho_{m+2} \le \cdots$ such that $\rho_\nu \asymp \nu^{2m}$ and

$$V(\varphi_\nu, \varphi_\mu) = \delta_{\nu\mu}, \quad J(\varphi_\nu, \varphi_\mu) = \rho_\nu \delta_{\nu\mu}, \quad \nu, \mu \ge 1, \tag{16}$$

where $\delta_{\nu\mu}$ is the Kronecker's delta.

We next place a prior distribution Π_λ on f, where Π_λ is a probability measure on $S^m(\mathbb{I})$ and $\lambda \ge 0$ is a hyperparameter. Similar to Section 2, we will characterize Π_λ through its Radon-Nikodym (RN) derivative w.r.t. Π, with Π a pre-given probability measure Π on $S^m(\mathbb{I})$. Specifically, assume that the RN derivative of Π_λ w.r.t. Π satisfies

$$\frac{d\Pi_\lambda}{d\Pi}(f) \propto \exp\left(-\frac{n\lambda}{2}J(f)\right), \tag{17}$$

where $J(f)$ is defined in (3). Interestingly, it is possible to explicitly construct Π_λ and Π such that (17) holds. To see this, let

$$G_\lambda(\cdot) = \sum_{\nu=m+1}^{\infty} w_\nu \varphi_\nu(\cdot), \tag{18}$$

where w_ν's are independent of the observations satisfying $w_\nu \sim N(0, 1/(\rho_\nu^{1+\beta/(2m)} + n\lambda\rho_\nu)), \nu > m$. Let $G(\cdot) = G_{\lambda=0}(\cdot)$. Suppose Π_λ and Π are probability measures induced by G_λ and G, i.e., $\Pi_\lambda(S) = P(G_\lambda \in S)$ and $\Pi(S) = P(G \in S)$ for any measurable $S \subseteq S^m(\mathbb{I})$. It follows by Hájek's lemma (see Shang and Cheng (2017)) that (17) holds. In (18), $\lambda \ge 0$ and $\beta > 1$ are both hyper-parameters characterizing the smoothness of the prior. It is easy to check that the sample path of G_λ belongs to $S^m(\mathbb{I})$ for any $\beta > 1$ almost surely. As demonstrated in a simulation study, the GCV-selected λ is sufficient to provide satisfactory results.

4. Main Results

In this section, we present a series of main results that are built upon a uniform Gaussian approximation theorem (Section 4.1). Three classes of aggregation procedures are then proposed: aggregated credible balls in both strong and weak topology, and aggregated credible intervals for linear functionals. These results can be classified into two types: *finite sample* construction (Sections 4.3, 4.4 and 4.5) and *asymptotic* construction (Section 4.6). The former construction is often time-consuming since its radius (interval length) is obtained through s posterior sampling, while the latter employs a large-sample limit of the radius given by an explicit formula. The computational gain will be illustrated by the simulations in Section 5. Similar to Section 2, let $I_1, I_2, \ldots, I_s$ be a random partition of $\{1, 2, \ldots, N\}$ such that $\cup_{j=1}^s I_j = \{1, 2, \ldots, N\}$ with $|I_j| = n$ for $j = 1, \ldots, s$ and $N = ns$.

4.1. A Uniform Gaussian Approximation Theorem

A fundamental theory underlying Bayesian aggregation is developed in this section. It is a *uniform* version of Gaussian approximation theorem that characterizes the limit shapes of a sequence of individual posterior distributions. This uniform validity holds if the number of posterior distributions does not grow too fast. Also, Bayesian aggregation procedures possess frequentist validity if λ is chosen properly.

Similar to (7), we note that each sub-posterior distribution can be written as

$$dP(f|\mathbf{D}_j) \propto \exp(n\ell_{jn}(f))d\Pi(f),$$

where $\ell_{jn}(f) = n^{-1}\sum_{i\in I_j}(Y_i - f(X_i))^2 - (\lambda/2)J(f)$. Define

$$\widehat{f}_{j,n} = \arg\max_{f\in S^m(\mathbb{I})} \ell_{jn}(f), \; j = 1, \ldots, s. \tag{19}$$

Suppose that $\widehat{f}_{j,n}$ admits the following Fourier expansion:

$$\widehat{f}_{j,n}(\cdot) = \sum_{\nu=1}^{\infty} \widehat{f}_\nu^{(j)} \varphi_\nu(\cdot), \; 1 \le j \le s. \tag{20}$$

Define $h = \lambda^{1/(2m)}$ with $h^* := N^{-\frac{1}{2m+\beta}}$. We remark that h^* is an optimal choice for our aggregation procedure as will be shown later.

Theorem 1 *(Uniform Gaussian Approximation) Suppose that f_0 admits a Fourier expansion $f_0(\cdot) = \sum_{\nu=1}^\infty f_\nu^0 \varphi_\nu(\cdot)$ which further satisfies*

$$\textit{Condition } (\boldsymbol{S}): \qquad \sum_{\nu=1}^{\infty} |f_\nu^0|^2 \rho_\nu^{1+\frac{\beta-1}{2m}} < \infty$$

If the following holds

$$m > 1 + \frac{\sqrt{3}}{2} \approx 1.866, 1 < \beta < 2m + \frac{1}{2m} - 1, s = o(N^{\frac{\beta-1}{2m+\beta}}) \;\; \textit{and} \;\; h \asymp h^*, \tag{21}$$

then we have as $N \to \infty$,

$$\sup_{S\in\mathcal{S}} \max_{1\le j\le s} |P(S|\boldsymbol{D}_j) - P_{0j}(S)| = O_{P_{f_0}}\left(\sqrt{s}N^{-\frac{4m^2+2m\beta-10m+1}{4m(2m+\beta)}}(\log N)^{\frac{5}{2}}\right), \tag{22}$$

where $\mathcal{S}$ is the Borel σ-algebra on $S^m(\mathbb{I})$ with respect to Π, and P_{0j}'s are GPs defined by

$$P_{0j}(S) = \frac{\int_S \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f)}{\int_{S^m(\mathbb{I})} \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f)}, \quad S \in \mathcal{S}. \tag{23}$$

Proof of Theorem 1 is rooted in Shang and Cheng (2017) who essentially considered $s = 1$. Substantial efforts have been made here to quantify a range of partition size s such that local posteriors can be uniformly approximated by GPs. The explicit structure of the GPs provides a guideline for our aggregation procedures which will be introduced in subsequent sections. It should be emphasized that our aggregation of GPs is weighted-averaging which is different from product-based ones such as Cao and Fleet (2014).

Condition (**S**) amounts to requiring known regularity of the truth $f_0 \in S^{m+\frac{\beta-1}{2}}(\mathbb{I})$. This can be seen from the inequality $\sum_{\nu=1}^{\infty} |f_\nu^0|^2 \nu^{2m+\beta-1} < \infty$ since $\rho_\nu \asymp \nu^{2m}$. This condition essentially means that f_0 has derivatives up to order $m + \frac{\beta-1}{2}$ (when this order is integer-valued). Combined with (21) this means that the regularity of f_0 belongs to $(m, 2m + \frac{1}{4m} - 1)$, i.e., the truth function is jointly confined by both functional space and the prior. The $\|\cdot\|$-norm used in (23) is defined as follows. For any $g, \widetilde{g} \in S^m(\mathbb{I})$, define

$$\langle g, \widetilde{g} \rangle = V(g, \widetilde{g}) + \lambda J(g, \widetilde{g}) \tag{24}$$

and its squared norm $\|g\|^2 = \langle g, g \rangle$. Clearly, $\langle \cdot, \cdot \rangle$ is a valid inner product on $S^m(\mathbb{I})$.

Remark 1 *We remark that (21) can be replaced by a more general rate condition:*

$$nh^{2m+1} \geq 1,\ a_n = O(\widetilde{r}_n),\ b_n \leq 1,\ r_n^2 b_n \leq \widetilde{r}_n^2,\ n\widetilde{r}_n^2 b_n = o(1),$$

where $r_n = (nh)^{-1/2} + h^m, \widetilde{r}_n = (nh/\log 2s)^{-1/2} + h^{m+\frac{\beta-1}{2}}, a_n = n^{-1/2} h^{-\frac{6m-1}{4m}} r_n \log N, b_n = n^{-1/2} h^{-\frac{6m-1}{4m}} (\log N)^{3/2}$. Here, we provide a technical explanation for the terms $r_n, \widetilde{r}_n, a_n, b_n$. Specifically, r_n can be viewed as the rate of convergence of local ordinary penalized MLE (19), $\widetilde{r}_n$ can be viewed as the posterior contraction rate of the local Bayesian mode, a_n, b_n are error bounds of the higher-order remainders in the Taylor expansions of the individual penalized likelihood functions. Uniform Gaussian approximation for general h (not necessarily $h \asymp h^*$) can be established under such condition.

Theorem 3.5 in Shang and Cheng (2017) shows that P_{0j} (conditional on $\mathbf{D}_j$) is induced by a Gaussian process, denoted as W^j, in the sense that $P_{0j}(S) = P(W^j \in S|\mathbf{D}_j)$ for any $S \in \mathcal{S}$. Define

$$\tau_\nu^2 = \rho_\nu^{1+\frac{\beta}{2m}},\ \nu \geq 1. \tag{25}$$

Then we have

$$W^j(\cdot) \quad = \quad \sum_{\nu=1}^{\infty} (a_{n,\nu} \widetilde{f}_\nu^{(j)} + b_{n,\nu} \tau_\nu v_\nu) \varphi_\nu(\cdot),\ j = 1, 2, \ldots, s,$$

where $a_{n,\nu} = n(1 + \lambda\rho_\nu)(\tau_\nu^2 + n(1 + \lambda\rho_\nu))^{-1}$, $b_{n,\nu} = (\tau_\nu^2 + n(1 + \lambda\rho_\nu))^{-1/2}$ and $v_\nu \sim N(0, \tau_\nu^{-2})$. For convenience, define the mean functions of W^j as

$$\widetilde{f}_{j,n}(\cdot) := \sum_{\nu=1}^{\infty} a_{n,\nu} \widetilde{f}_\nu^{(j)} \varphi_\nu(\cdot),\ j = 1, \ldots, s, \tag{26}$$

such that we can re-express W^j as

$$W^j = \widetilde{f}_{j,n} + W_n,\ j = 1, \ldots, s,$$

where $W_n(\cdot) := \sum_{\nu=1}^{\infty} b_{n,\nu} \tau_\nu v_\nu \varphi_\nu(\cdot)$ is a zero-mean GP. Note that the posterior mode $\widetilde{f}_{j,n}$ is very close to $\widehat{f}_{j,n}$ since $\|\widetilde{f}_{j,n} - \widehat{f}_{j,n}\| = o_{P_{f_0}}(1)$ uniformly for $1 \leq j \leq s$; see the proof of Theorem 3. The above characterization of W^j is useful for the subsequent Bayesian aggregation procedures.

4.2. Aggregated posterior means

In this section, we propose a method to aggregate the posterior means $\breve{f}_{j,n} := E\{f|\mathbf{D}_j\}$, for $j = 1, \ldots, s$. The aggregated mean function, denoted as $\breve{f}_{N,\lambda}(\cdot)$, can be viewed as a nonparametric Bayesian estimate of f, and will be used to construct aggregated credible balls/intervals to be introduced later.

Our aggregation procedure is

$$\breve{f}_{N,\lambda}(\cdot) = \sum_{\nu=1}^{\infty} \frac{a_{N,\nu}}{a_{n,\nu}} V\left(\frac{1}{s}\sum_{j=1}^{s} \breve{f}_{j,n}, \varphi_\nu\right) \varphi_\nu(\cdot). \tag{27}$$

Note that when the model is Gaussian and $f \in S_0^m(0,1)$, (27) becomes (10). Next we will show that the aggregation procedure (27) yields minimax optimality in the following theorem.

Theorem 2 *Under conditions of Theorem 1, the following result holds:*

$$\max_{1 \leq j \leq s} \|\breve{f}_{j,n} - \widetilde{f}_{j,n}\| = O_{P_{f_0}}\left(\widetilde{r}_n \sqrt{s} N^{-\frac{4m^2+2m\beta-10m+1}{4m(2m+\beta)}} (\log N)^{\frac{5}{2}}\right), \tag{28}$$

If, in addition, $3/2 < \beta < 2m + 1/(2m) - 3/2$ and s satisfies

$$s = o\left(N^{\frac{4m^2+2m\beta-11m+1}{8m(2m+\beta)}} (\log N)^{-\frac{3}{2}}\right), \tag{29}$$

then it holds that

$$\|\breve{f}_{N,\lambda} - f_0\|_2 = O_{P_{f_0}}\left(N^{-\frac{2m+\beta-1}{2(2m+\beta)}}\right), \tag{30}$$

where $\|f\|_2 = \sqrt{V(f)}$ denotes the V-norm.

According to van der Vaart et al. (2008b), the rate in (30) is minimax optimal given Condition (**S**).

4.3. Aggregated credible region in strong topology

In this section, we construct an aggregated credible region based on s individual credible regions (w.r.t. a weighted ℓ^2-norm). Specifically, s radii are combined in an explicit manner. This aggregated region possesses nominal posterior mass asymptotically, and is further proven to cover the true function with probability tending to one. This nice frequentist

property is achieved as long as s is not diverging fast and the assigned GP prior in each subset is chosen by setting $h \asymp h^*$, i.e., $\lambda \asymp N^{-2m/(2m+\beta)}$. The conservative frequentist coverage can be improved to the nominal level if we use a weaker norm in defining credible region; see Section 4.4.

Based on each subset $\mathbf{D}_j$, the individual credible ball is constructed as follows:

$$R_{j,n}(\alpha) = \{f \in S^m(\mathbb{I}) : \|f - \breve{f}_{j,n}\|_2 \le r_{j,n}(\alpha)\}.$$

The credible ball centers around the posterior mean $\breve{f}_{j,n}$, while its radius $r_{j,n}(\alpha)$ is directly sampled from MCMC such that $P(R_{j,n}(\alpha)|\mathbf{D}_j) = 1-\alpha$ for any $\alpha \in (0,1)$. We will construct an "aggregated" region centering at $\breve{f}_{N,\lambda}$ with radius explicitly constructed as follows:

$$r_N(\alpha) = \sqrt{\frac{1}{N}\left[\zeta_{1,N} + \sqrt{\frac{\zeta_{2,N}}{\zeta_{2,n}}}\left(\frac{n}{s}\sum_{j=1}^{s} r_{j,n}^2(\alpha) - \zeta_{1,n}\right)\right]}, \tag{31}$$

where

$$\zeta_{k,n} = \sum_{\nu=1}^{\infty}\left(\frac{n}{\tau_\nu^2 + n(1+\lambda\rho_\nu)}\right)^k \text{ for } k = 1, 2.$$

The final aggregated credible region is obtained as

$$R_N(\alpha) := \{f \in S^m(\mathbb{I}) : \|f - \breve{f}_{N,\lambda}\|_2 \le r_N(\alpha)\}. \tag{32}$$

Our theorem below confirms that $R_N(\alpha)$ indeed possesses (asymptotic) posterior mass $(1-\alpha)$, and more importantly, proves that it covers the true function f_0 with probability tending to one.

Theorem 3 *Suppose that f_0 satisfies Condition (**S**), $m > 1 + \frac{\sqrt{3}}{2}$, $3/2 < \beta < 2m + 1/(2m) - 3/2$, $s = o(N^{\frac{\beta-1}{2m+\beta}})$, (29) and $h \asymp h^*$. Then for any $\alpha \in (0,1)$, $P(R_N(\alpha)|\boldsymbol{D}) = 1 - \alpha + o_{P_{f_0}}(1)$ and $\lim_{n\to\infty} P_{f_0}(f_0 \in R_N(\alpha)) = 1$.*

From the proof of Theorem 3, we point out that when $s = 1$, the posterior mass of the aggregated credible region is exactly $1-\alpha$, consistent with Shang and Cheng (2017). This remark also applies to other aggregated procedures to be presented later.

Remark 2 *When $h \asymp h^*$, the radius of the aggregated ball $r_N(\alpha) \asymp N^{-\frac{2m+\beta-1}{2(2m+\beta)}}$ according to the discussions in Section 4.6. This is the optimal rate at which a posterior ball contracts based on the entire sample; see van der Vaart et al. (2008b).*

4.4. Aggregated credible region in weak topology

In this section, we invoke a weaker norm (than that used in Section 4.3) to construct an aggregated credible region. Under this new norm (inspired by Castillo et al. (2013, 2014)), it is proven that the frequentist coverage *exactly* matches with the asymptotic credibility level. The requirement on s and h in this section remains the same as Section 4.3.

We define a weaker norm than $\|\cdot\|_2$, denoted $\|\cdot\|_\omega$. For any $f \in S^m(\mathbb{I})$ with $f = \sum_\nu f_\nu \varphi_\nu$, define $\|f\|_\omega^2 = \sum_{\nu=1}^{\infty} \omega_\nu f_\nu^2$, where $\omega_\nu = (\nu(\log 2\nu))^{-\tau}$ for some constant $\tau > 1$. Since $\omega_\nu < 1$ for

all $\nu \ge 1$, we have $\|f\|_\omega \le \|f\|_2$. Under the new $\|\cdot\|_\omega$-norm, each individual $(1-\alpha)$ credible region is constructed as

$$R_{j,n}^{\omega}(\alpha) = \{f \in S^m(\mathbb{I}) : \|f - \breve{f}_{j,n}\|_\omega \le r_{\omega,j,n}(\alpha)\},$$

where $r_{\omega,j,n}(\alpha)$ is directly obtained from posterior sampling such that $P(R_{j,n}^{\omega}(\alpha)|\mathbf{D}_j) = 1-\alpha$.

Under $\|\cdot\|_\omega$-norm, the aggregated credible region is constructed as:

$$R_N^{\omega}(\alpha) := \{f \in S^m(\mathbb{I}) : \|f - \breve{f}_{N,\lambda}\|_\omega \le r_{\omega,N}(\alpha)\}, \tag{33}$$

where the radius is given as

$$r_{\omega,N}(\alpha) = \sqrt{\frac{1}{s^2}\sum_{j=1}^{s} r_{\omega,j,n}^2(\alpha)}. \tag{34}$$

Interestingly, Section 4.6 illustrates that the aggregated radius $r_{\omega,N}(\alpha)$ contracts at root-N rate.

Our theorem below shows that the frequentist covergage of $R_N^{\omega}(\alpha)$ exactly matches with the asymptotic posterior mass, both of which achieve the nominal level $(1-\alpha)$.

Theorem 4 *Suppose that f_0 satisfies Condition (**S**), $m > 1 + \sqrt{3}/2$, $2 \le \beta < \frac{(2m-1)^2}{2m}$, $s = o(N^{\frac{\beta-1}{2m+\beta}})$, $s = o(N^{\frac{4m^2+2m\beta-12m+1}{8m(2m+\beta)}}(\log N)^{-\frac{3}{2}})$, and $h \asymp h^*$. Then for any $\alpha \in (0,1)$, $P(R_N^{\omega}(\alpha)|\boldsymbol{D}) = 1 - \alpha + o_{P_{f_0}}(1)$ and $\lim_{n\to\infty} P_{f_0}(f_0 \in R_N^{\omega}(\alpha)) = 1 - \alpha$.*

4.5. Aggregated credible interval for linear functional

In this section, we construct aggregated credible intervals for a class of linear functionals of f, denoted as $F(f)$. Examples include the evaluation functional, i.e., $F(f) = f(x)$, and integral functional, i.e., $F(f) = \int_0^1 f(x)dx$. Specifically, the interval is centered at $F(\breve{f}_{N,\lambda})$ with an length aggregated through s lengths obtained from posterior sampling. Posterior and frequentist coverage properties of this aggregated interval depends on the functional form $F(\cdot)$. Again, our theory holds when s is mildly diverging and $h \asymp h^*$.

Let $F : S^m(\mathbb{I}) \mapsto \mathbb{R}$ be a linear Π-measurable functional satisfying the following Condition (**F**): $\sup_{\nu\ge1} |F(\varphi_\nu)| < \infty$, and there exist constants $\kappa > 0$ and $r \in [0,1]$ such that for any $f \in S^m(\mathbb{I})$,

$$|F(f)| \le \kappa h^{-r/2}\|f\|. \tag{35}$$

It follows by Shang and Cheng (2017) that the evaluation functional satisfies Condition (**F**) with $r = 1$ and the integral functional satisfies Condition (**F**) with $r = 0$.

Based on each $\mathbf{D}_j$, we obtain from posterior samples the following $(1-\alpha)$ credible interval:

$$CI_{j,n}^{F}(\alpha) := \{f \in S^m(\mathbb{I}) : |F(f) - F(\breve{f}_{j,n})| \le r_{F,j,n}(\alpha)\},$$

where $r_{F,j,n}(\alpha)$ is a radius such that $P(CI_{j,n}^{F}(\alpha)|\mathbf{D}_j) = 1-\alpha$. The aggregated credible interval is constructed as

$$CI_N^{F}(\alpha) := \{f \in S^m(\mathbb{I}) : |F(f) - F(\breve{f}_{N,\lambda})| \le \bar{r}_{F,N}(\alpha)\} \tag{36}$$

where

$$r_{F,N}(\alpha) = \frac{\theta_{1,N}}{\theta_{1,n}} \sqrt{\frac{1}{s}\sum_{j=1}^{s} r_{F,j,n}(\alpha)^2} \quad \text{and} \quad \theta_{k,n}^2 = \sum_{\nu=1}^{\infty} \frac{F(\varphi_\nu)^2}{(\tau_\nu^2 + n(1+\lambda\rho_\nu))^k} \text{ for } k = 1, 2. \tag{37}$$

The shrinking rate of $\bar{r}_{F,N}(\alpha)$ depends on the functional form F; see Section 4.6.

Our theorem below investigates the asymptotic properties of $CI_F^N(\alpha)$ in terms of both posterior and frequentist coverage.

Theorem 5 *Suppose that $f_0 = \sum_{\nu=1}^{\infty} f_\nu^0 \varphi_\nu$ satisfies Condition ($\boldsymbol{S'}$): $\sum_{\nu=1}^{\infty} |f_\nu^0|^2 \nu^{2m+\beta} < \infty$, $E_{f_0}\{\epsilon^4|X\} \le M_4$ a.s. for some constant $M_4 > 0$, $N^k \theta_{k,N}^2 \gtrsim h^{-r}$ for $k = 1, 2$, $m > 1 + \frac{\sqrt{3}}{2}$, $2 \le \beta < \frac{(2m-1)^2}{2m}$, $s = o(N^{\frac{\beta-1}{2m+\beta}})$, $s = o(N^{\frac{4m^2+2m\beta-12m+1}{8m(2m+\beta)}} (\log N)^{-\frac{3}{2}})$, (29) and $h \asymp h^*$. Then for any $\alpha \in (0,1)$, $P(CI_N^F(\alpha)|\boldsymbol{D}) = 1 - \alpha + o_{P_{f_0}}(1)$, and $\liminf_{N\to\infty} P_{f_0}(f_0 \in CI_N^F(\alpha)) \ge 1 - \alpha$ given that Condition ($\boldsymbol{F}$) holds. Moreover, if $0 < \sum_{\nu=1}^{\infty} F(\varphi_\nu)^2 < \infty$, then $\lim_{N\to\infty} P_{f_0}(f_0 \in CI_N^F(\alpha)) = 1 - \alpha$.*

Note that Condition ($\mathbf{S'}$) is slightly stronger than Condition ($\mathbf{S}$) required in Theorem 1. Indeed, this condition essentially means that f_0 has derivatives up to order $m + \frac{\beta}{2}$ (when this order is integer-valued). Hence, Theorem 5 requires a more smooth true function f_0.

It was shown in Shang and Cheng (2017) that the integral functional $F_x(f) := \int_0^x f(z)dz$ for any $x \in [0,1]$ satisfies (35) with $r = 0$ and $0 < \sum_{\nu=1}^{\infty} F_x(\varphi_\nu)^2 < \infty$. Therefore, the $(1-\alpha)$-th credible interval of $F_x(f)$ achieves exactly $(1-\alpha)$ frequentist coverage, while that for the evaluation functional is more conservative. These theoretical findings will be empirically verified in Section 5 .

4.6. Asymptotic aggregated inference

In practice, the centers $\breve{f}_{N,\lambda}$, $F(\breve{f}_{N,\lambda})$ and the radii $r_{j,n}(\alpha)$, $r_{\omega,j,n}(\alpha)$, $r_{F,j,n}(\alpha)$ in Sections 4.3 – 4.5 are directly obtained from posterior samples. Sometimes posterior sampling is time consuming and inefficient, particularly as $s \to \infty$. This computational consideration motivates us to propose an *asymptotic* approach in which one replaces the above centers/radii by their large sample limits. Our new asymptotic inference procedures dramatically improve the computing speed, as displayed in simulations; see Section 5.

Define

$$\widetilde{f}_{N,\lambda}(\cdot) = \sum_{\nu=1}^{\infty} \frac{a_{N,\nu}}{a_{n,\nu}} V\left(\frac{1}{s}\sum_{j=1}^{s} \widetilde{f}_{j,n}, \varphi_\nu\right) \varphi_\nu(\cdot). \tag{38}$$

Clearly, $\widetilde{f}_{N,\lambda}$ is a counterpart of $\breve{f}_{N,\lambda}$ (27) with $\breve{f}_{j,n}$ therein replaced by $\widetilde{f}_{j,n}$. By a careful examination of the proofs of Theorems 3 – 5, it can be shown that the following limits hold:

$$\begin{aligned}
\|\breve{f}_{N,\lambda} - \widetilde{f}_{N,\lambda}\| &= o_{P_{f_0}}(N^{-1/2}h^{-1/4}), \\
\max_{1\le j\le s}\left|\frac{nr_{j,n}^2(\alpha) - \zeta_{1,n}}{\sqrt{2\zeta_{2,n}}} - z_\alpha\right| &= o_{P_{f_0}}(1), \\
\max_{1\le j\le s}|\sqrt{n}r_{\omega,j,n}(\alpha) - \sqrt{c_\alpha}| &= o_{P_{f_0}}(1), \\
\max_{1\le j\le s}|r_{F,j,n}(\alpha)/\theta_{1,n} - z_{\alpha/2}| &= o_{P_{f_0}}(1),
\end{aligned} \tag{39}$$

where $z_\alpha = \Phi^{-1}(1-\alpha)$ with $\Phi(\cdot)$ being the c.d.f. of standard normal random variable, and $c_\alpha > 0$ satisfies $P(\sum_{\nu=1}^{\infty} d_\nu \eta_\nu^2 \le c_\alpha) = 1 - \alpha$ with η_ν being independent standard normal random variables.

It yields from (39) that the following approximation relationships hold uniformly for $1 \le j \le s$:

$$r_{j,n}(\alpha) \approx \sqrt{\frac{\zeta_{1,n} + \sqrt{2\zeta_{2,n}}z_\alpha}{n}}, \quad r_{\omega,j,n}(\alpha) \approx \sqrt{\frac{c_\alpha}{n}} \quad \text{and} \quad r_{F,j,n}(\alpha) \approx \theta_{1,n} z_{\alpha/2},$$

which further implies (by the aggregation formulae (31), (34) and (37))

$$\begin{aligned}
r_N(\alpha) \approx r_N^\dagger(\alpha) &:= \sqrt{\frac{\zeta_{1,N} + \sqrt{2\zeta_{2,N}}z_\alpha}{N}}, \\
r_{\omega,N}(\alpha) \approx r_{\omega,N}^\dagger(\alpha) &:= \sqrt{\frac{c_\alpha}{N}}, \\
r_{F,N}(\alpha) \approx r_{F,N}^\dagger(\alpha) &:= \theta_{1,N} z_{\alpha/2}.
\end{aligned} \tag{40}$$

Thus, we have the following *asymptotic* counterparts of $R_N(\alpha)$, $R_N^\omega(\alpha)$ and $CI_N^F(\alpha)$:

$$R_N^\dagger(\alpha) := \{f \in S^m(\mathbb{I}) : \|f - \widetilde{f}_{N,\lambda}\|_2 \le r_N^\dagger(\alpha)\}, \tag{41}$$

$$R_N^{\dagger\omega}(\alpha) := \{f \in S^m(\mathbb{I}) : \|f - \widetilde{f}_{N,\lambda}\|_\omega \le r_{\omega,N}^\dagger(\alpha)\}, \tag{42}$$

$$CI_N^{\dagger F}(\alpha) := \{f \in S^m(\mathbb{I}) : |F(f) - F(\widetilde{f}_{N,\lambda})| \le r_{F,N}^\dagger(\alpha)\}. \tag{43}$$

Our theorem below shows that the posterior coverage and frequentist coverage of the above computationally efficient alternatives remain the same as those for $R_N(\alpha)$, $R_N^\omega(\alpha)$ and $CI_N^F(\alpha)$ under the same set of conditions.

Theorem 6 *Suppose that all assumptions in Theorems 3 – 5 hold. Then for any $\alpha \in (0,1)$, $R_N^\dagger(\alpha)$, $R_N^{\dagger\omega}(\alpha)$ and $CI_N^{\dagger F}(\alpha)$ possess exactly the same posterior and frequentist properties as $R_N(\alpha)$, $R_N^\omega(\alpha)$ and $CI_N^F(\alpha)$, respectively.*

As a byproduct, (40) implies the contraction rate of each aggregated credible ball/interval in Sections 4.3 – 4.6. It is easy to see that $r_{\omega,N}(\alpha) \asymp N^{-1/2}$. As for $r_{F,N}(\alpha)$, it depends on the functional form F. For example, when F is an evaluation functional, it holds that $\theta_{1,N}^2 \asymp (Nh)^{-1}$, leading to $N^{-\frac{2m+\beta-1}{2(2m+\beta)}}$ when $h \asymp h^*$; when F is an integral functional, we have $r_{F,N}(\alpha) \asymp N^{-1/2}$ since $\theta_{1,N}^2 \asymp N^{-1}$. As for $r_N(\alpha)$, it can be shown by a simple fact $\zeta_{1,N}, \zeta_{2,N} \asymp h^{-1}$ that $r_N(\alpha) \asymp (Nh)^{-1/2} \asymp N^{-\frac{2m+\beta-1}{2(2m+\beta)}}$ when $h \asymp h^*$. This contraction rate turns out to be optimal based on the entire sample; see van der Vaart et al. (2008b). However, if we choose h in the scale of subsample size n, e.g., $h \asymp n^{-\frac{1}{2m+\beta}}$, similar arguments show that $r_N(\alpha) \asymp N^{-\frac{2m+\beta-1}{2(2m+\beta)}} s^{-\frac{1}{2(2m+\beta)}}$. Hence, such a region contracts faster than the optimal rate, which results in unsatisfactory frequentist coverage.

Table 1 summarizes six aggregated credible regions/intervals from Sections 4.3 – 4.5 in terms of their centers and radii.

Table 1. *Summary of Aggregated* $(1-\alpha)$ *Credible Regions/Intervals*

Type	Name	Notation	Center	Radius
Finite-sample	strong CR for f	$R_N(\alpha)$	$\breve{f}_{N,\lambda}$	$r_N(\alpha)$
	weak CR for f	$R_N^{\omega}(\alpha)$	$\breve{f}_{N,\lambda}$	$r_{\omega,N}(\alpha)$
	CI for $F(f)$	$CI_N^F(\alpha)$	$F(\breve{f}_{N,\lambda})$	$r_{F,N}(\alpha)$
Asymptotic	strong CR for f	$R_N^{\dagger}(\alpha)$	$\widetilde{f}_{N,\lambda}$	$r_N^{\dagger}(\alpha)$
	weak CR for f	$R_N^{\dagger\omega}(\alpha)$	$\widetilde{f}_{N,\lambda}$	$r_{\omega,N}^{\dagger}(\alpha)$
	CI for $F(f)$	$CI_N^{\dagger F}(\alpha)$	$F(\widetilde{f}_{N,\lambda})$	$r_{F,N}^{\dagger}(\alpha)$

5. Simulation Study

In this section, statistical properties of the proposed aggregated procedures are examined using a simulation study. We generated samples from the following model

$$Y_{ij} = f_0(X_{ij}) + \epsilon_{ij},\ i = 1, 2, \ldots, n, j = 1, 2, \ldots, s, \tag{44}$$

where $X_{ij} \overset{iid}{\sim} Unif[0,1]$, $\epsilon_{ij} \overset{iid}{\sim} N(0,1)$, and ϵ_{ij} are independent of X_{ij}. The true regression function was chosen to be $f_0(x) = 2.4\beta_{30,17}(x) + 1.6\beta_{3,11}(x)$, where $\beta_{a,b}$ is the probability density function for $Beta(a,b)$.

Consider GP prior $f \sim \sum_{\nu=1}^{n} w_\nu \varphi_\nu$, where w_ν are defined in (18). The proposed Bayesian procedures were examined. Specifically, we computed the frequentist coverage proportions (CP) of the credible regions (32), (33), (41), (42), and credible intervals (36), (43). In particular, (32), (33) and (36) were constructed based on posterior samples, as described in Sections 4.2–4.5; whereas (41), (42) and (43) were constructed based on asymptotic theory developed in Section 4.6. To ease presentation, we call (32) and (33) as finite-sample credible regions (FCR), and call (41) and (42) as asymptotic credible regions (ACR).

The calculation of CP was based on 500 independent experiments. Specifically, the CP is the proportion of the credible regions/intervals containing $f_0/F(f_0)$ (for a linear functional F). Two types of F were considered: (1) the evaluation functional $F_x(f) = f(x)$ for any $x \in [0,1]$, and (2) the integral functional $F_x(f) = \int_0^x f(z)dz$ for any $x \in [0,1]$. In both cases, we consider F_x with x being 15 evenly spaced points in [0.05,0.95]. To make the study more complete, a set of credibility levels were examined, i.e., $1-\alpha = 0.1, 0.3, 0.5, 0.7, 0.9, 0.95$. In each experiment, $N = 1200$ independent samples were generated from the model (44). For ACR and FCR, we chose the number of divisions $s = 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60$. Define $\gamma = \log s / \log N$. Note that $s = 1$ (equivalently, $\gamma = 0$) means "no division."

Figure 2 demonstrates the results for FCR and ACR based on strong topology, i.e., (32) and (41). The red dotted line indicates the $(1-\alpha)$ credibility level. It can be seen that the CP of both FCR and ACR is above the credibility levels when γ is small, while it suddenly drops to zero as γ is beyond some threshold, say 0.3. This observation supports our theory that s should not grow too fast, and that the credible regions based on strong

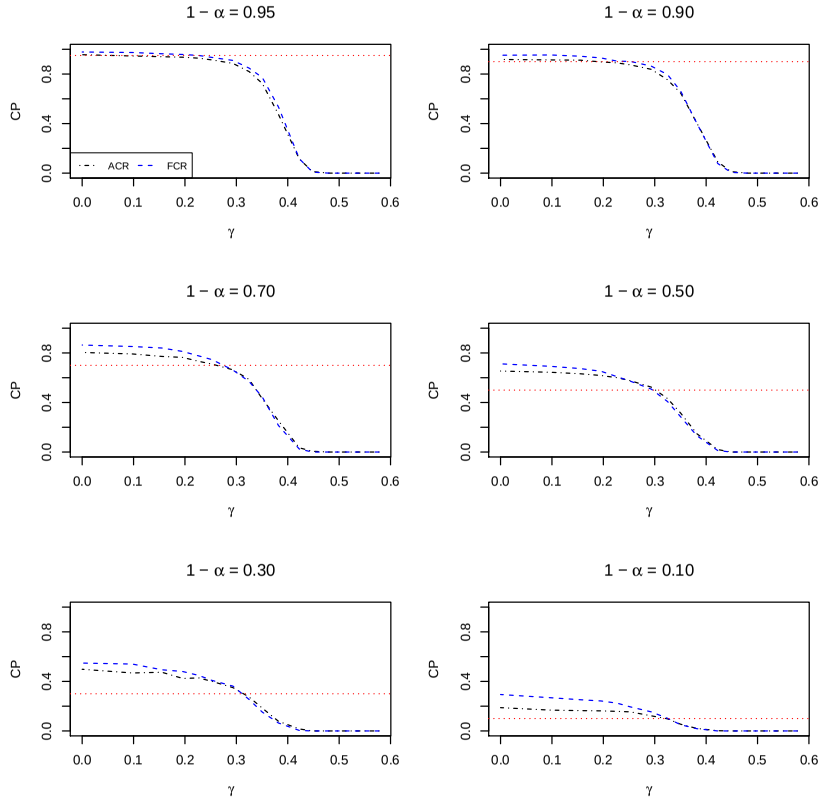


Figure 2. *CP of ACR and FCR based on strong topology. Dotted red lines indicate credibility levels.*

topology tends to be more "conservative." Figure 3 demonstrates the results for FCR and ACR based on weak topology, i.e., (33) and (42). We observe that the CP of both ACR and FCR approaches the desired credibility levels when $\gamma \le 0.3$, but quickly drops to zero when γ becomes large. This observation also supports our theory that the use of weak topology leads to a more satisfactory frequentist coverage.

For credible intervals of linear functionals, we chose the number of divisions $s = 1, 6, 15, 60$. Figures 4 and 5 display the results for evaluation functional and integral functional, respectively, based on posterior samples. It can be seen that when $s = 60$, the CP of the credible intervals for the evaluation functional drops to zero at most of the x points, indicating the failure in covering the true values of the function. However, when $s = 1, 6, 15$, the CP is

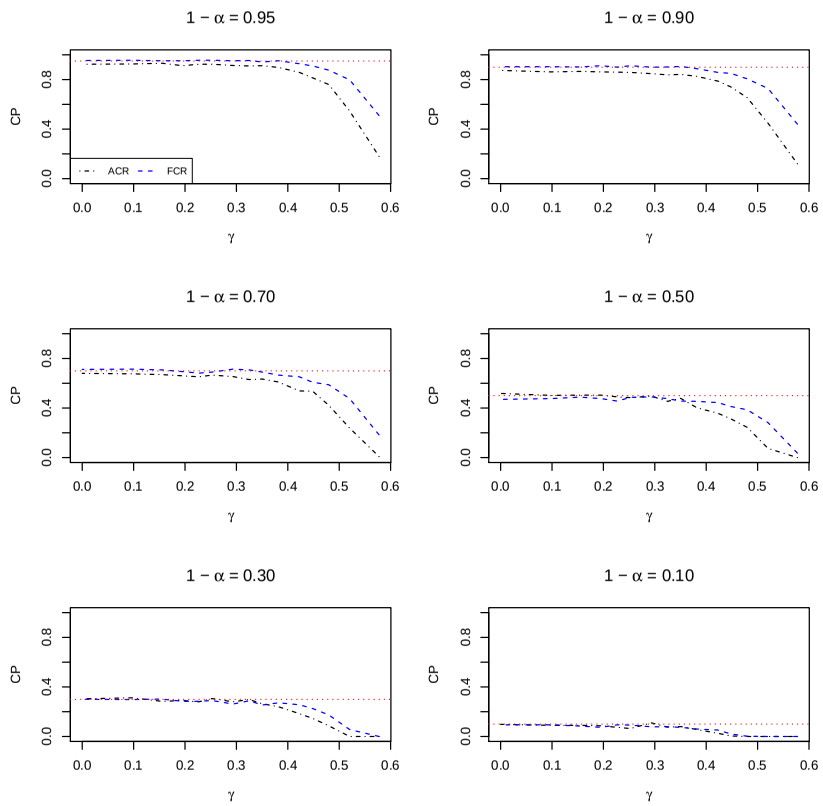


Figure 3. *CP of ACR and FCR based on weak topology. Dotted red lines indicate credibility levels.*

above the credibility levels except for the points where the true function f_0 has peaks; see (a) of Figure 1. The observation that the CP stays above $(1-\alpha)$ coincides with our theory that the credible interval of the evaluation functional is conservative. On the other hand, it can be seen that when $s = 60$, the CP of the credible intervals for the integral functional becomes far below the credibility levels at most x. However, when $s = 1, 6, 15$, the CP is close to the credibility levels at all x. This finding coincides with our theory that the the credible interval of the integral functional achieves exactly $(1-\alpha)$ frequentist coverage. The above results also support our claim that s cannot grow too fast for guaranteeing frequency validity. Credible intervals based on asymptotic theory, i.e., (43), were summarized in Figures 11 and 12 of the supplement document Shang and Cheng. Interpretations of these results are similar to those based on finite posterior samples.

The supplement document Shang and Cheng also includes Figures 13 – 16 which demonstrate how the radii/lengths of the aggregated credible regions/intervals change along with γ, the size of the subsample. It can be observed that when $\gamma \le 0.3$, indicating that the full sample is divided into at most twelve subsamples, the radii of the aggregated regions/intervals are almost identical to the radii of the regions/intervals directly constructed from the full sample, i.e., $\gamma = 0$. This means that our aggregated procedures, based on a suitable amount of divisions, indeed mimic the oracle procedures. However, when γ increases to 0.6, the distinctions between the the aggregated and oracle procedures quickly become obvious.

We also repeated the above study for $N = 1800$ and 2400. The plots corresponding to these studies are given in supplement document; see Section S.8.6 of Shang and Cheng. The interpretations of these additional results are similar as above.

To the end of this section, computing efficiency is investigated. Figure 6 displays the results based on a single experiment for various choices of N. Specifically, we look at the value of the quantity $\rho = 1 - (T/T_0)$ versus a collection of γ's for FCR and ACR, where T_0 (T) is the computing time without using D&C (based on D&C). We observe that T is substantially smaller than T_0, and this computation efficiency (as reflected by the value of ρ) becomes more obvious as γ grows for each fixed N. This can also be seen as N grows for each fixed γ. However, this reduction in computing time does not affect the performances of the aggregated credible regions when $0 \le \gamma \le 0.3$, as demonstrated in Figures 2, 3, 13–16.

6. Real Data Analysis

In this section, we apply our methods to Million Song Data (MSD) and Flight Delay Data (FDD).

6.1. Million Song Data

As a real application, we apply our aggregation procedure to analyze MSD. The MSD is a perfect example of large dataset, a freely-available collection of audio features and metadata for a million contemporary popular music tracks. Each observation is a song track released between the year 1922 and 2011. The response variable Y_i is the year when the song was released and the covariate X_i is the timbre average of the song. The main purpose is to explore a relationship, denoted as f, between song features and years in a nonparametric regression model, i.e., year = f(timbre)+error. The above model is useful to

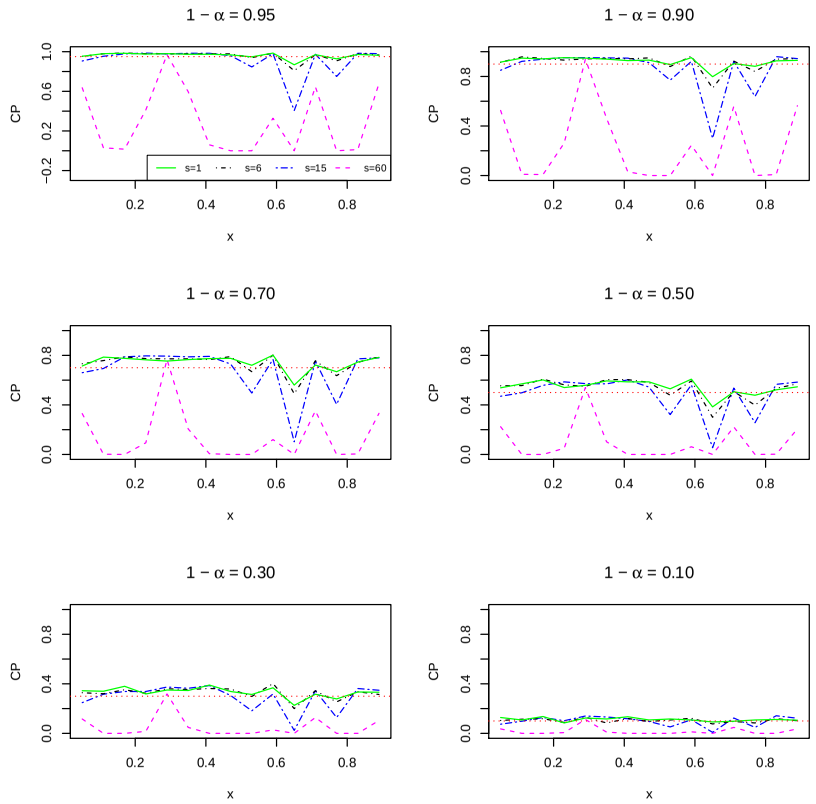


Figure 4. *CP of $F_x(f) = f(x)$ against x based on posterior samples of f. Dotted red lines indicate credibility levels.*

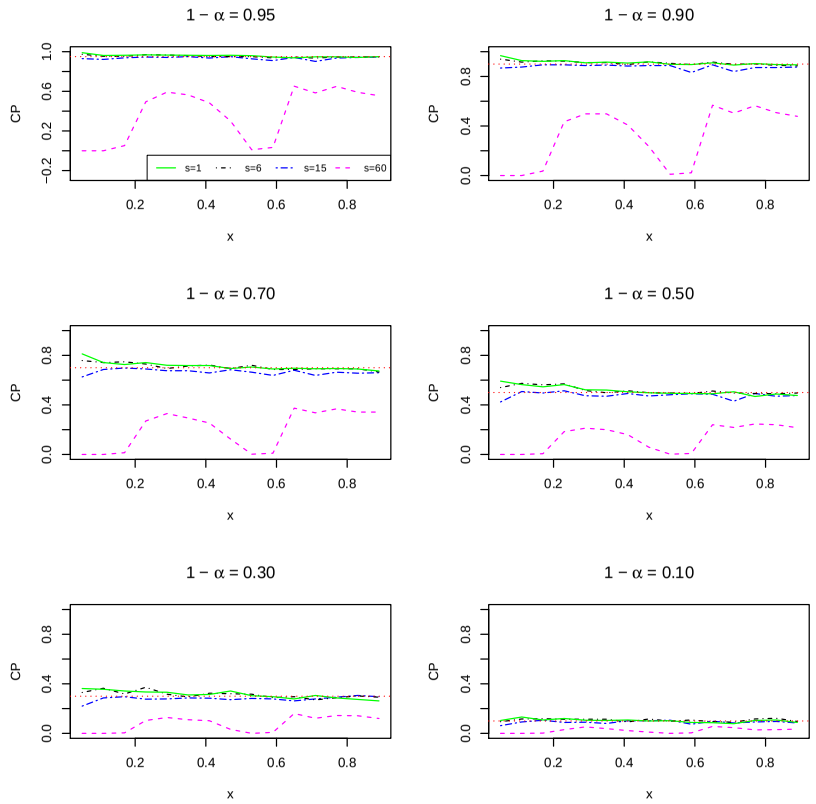


Figure 5. *CP of $F_x(f) = \int_0^x f(z)dz$ against x based on posterior samples of f. Dotted red lines indicate credibility levels.*

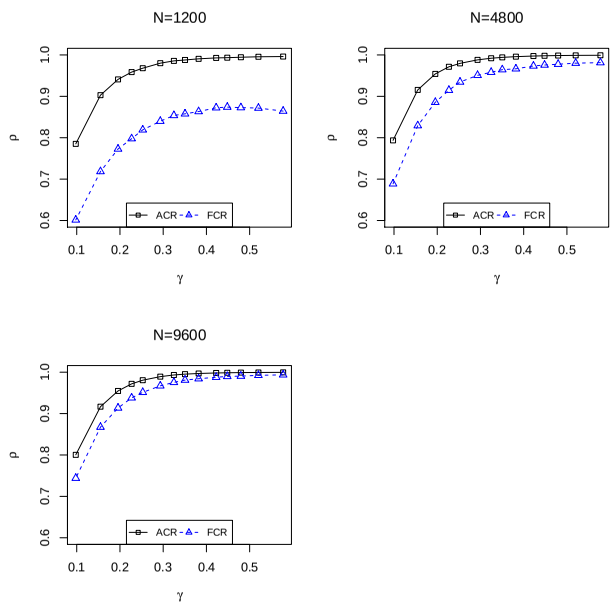


Figure 6. *ρ versus γ based on FCR and ACR for single experiment.*

predict production year based on song timbre. Due to enormous sample size, processing the entire data is infeasible. In frequentist setting, a distributed kernel ridge regression method was proposed by Zhang et al. (2015a,b) for estimation purposes (without quantifying uncertainty).

In the Bayesian setup, we applied our aggregation procedure to construct 95% credible sets for f based on a subset of $N = 10,000$ songs released from the year 1996 to 2010. We randomly split the observations to $s = 5, 10, 20$ subsets. We also compared our results with the baseline method in which all ten thousand observations were used. Credible sets are displayed as gray areas in Figure 7. We find that the shapes of all credible sets are overall the same when the timbre ranges from -4 to 4, e.g., all display a W-shape, although the results are a bit sensitive near the endpoints. Therefore, the overall pattern of the sets appears to be insensitive to the above selections of s.

6.2. Flight Delay Data

We applied our aggregation procedure to one more real data set, the FDD. The data consists of flight arrival and departure information for all commercial flights within the United States, from October 1987 to April 2008. The main purpose is to find the key factors that have an

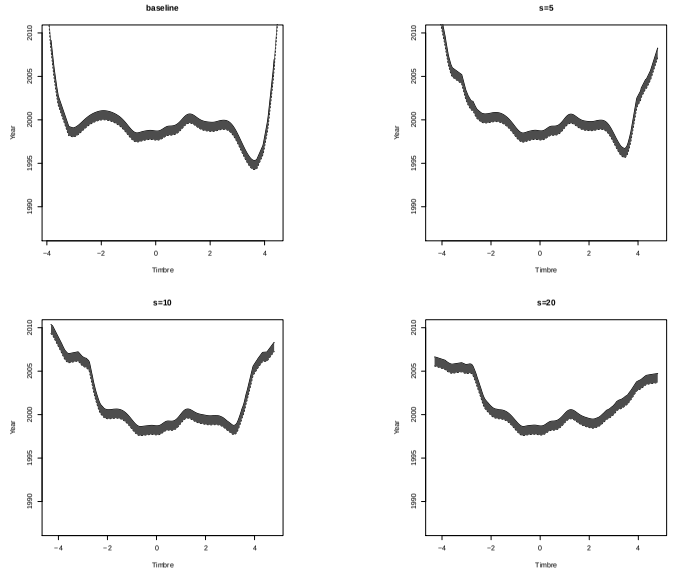


Figure 7. *95% Credible sets (grey areas) for f based on a subset of 10,000 samples in Million Song Data. The first plot refers to the baseline method where the whole samples were used. The rest three plots refer to the aggregation procedure which was applied to 5, 10, 20 random splits.*

impact on the flight delay. We considered the relationship (denoted f) between month and the length of the flight delay, i.e., length of flight delay = f(month)+error. Negative length of delay implies that the flight arrived earlier. We applied the same Bayesian aggregation procedure as described in MSD to a randomly selected subset of $N = 10,000$ flight information in the year 2007. We randomly split the observations to $s = 10, 100, 500$ subsamples, based on which the aggregated credible sets for f were constructed. We also compared the results with the baseline where all the ten thousand samples were used. Credible sets are displayed as gray areas in Figure 8. Again, the shapes of the four credible sets appear to be almost the same for all s.

6.3. Computation Efficiency

We compare the overall execute computation time of both MSD and FDD on different numbers of splits, e.g. computational time per machine × number of machines in Figure 9-10. It can be seen that the computing time dramatically decreases as the number of splits increases, which reflects the scalability of our proposed algorithm.

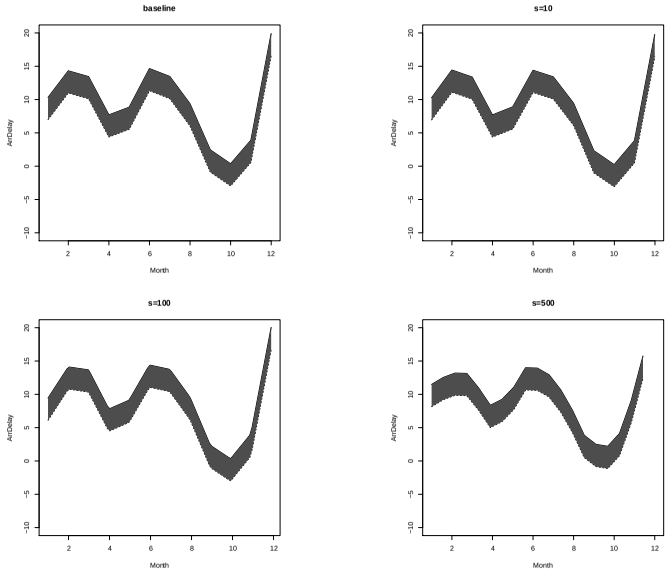


Figure 8. *95% Credible sets (grey areas) for f based on a subset of 10,000 samples in Flight Delay Data. The first plot refers to the baseline method where the whole samples were used. The rest three plots refer to the aggregation procedure which was applied to 10, 100, 500 random splits.*

7. Conclusions

This paper proposes algorithms for aggregating individual posterior results such as modes, balls, intervals, into their global counterparts. The algorithms are easy-to-implement which are particularly useful in big data scenarios. We also experimented the proposed algorithms through simulated and real data sets. A notable contribution of this article is to provide rigorously justified theoretical guarantees. The major tool for proving our theoretical results is a uniform Gaussian approximation theorem which shows that the individual posterior distributions converge uniformly to Gaussian processes provided that the number of subsets is not too large.

Acknowledgments

Shang's research is sponsored by NSF grants DMS 1821157 and DMS 1764280. Guang Cheng would like to acknowledge support by NSF DMS-1712907, DMS-1811812, DMS-1821183, and Office of Naval Research (ONR N00014-18-2759). In addition, Guang Cheng is a member of Institute for Advanced Study, Princeton and visiting Fellow of SAMSI for the Deep Learning Program in the Fall of 2019; he would like to thank both Institutes for their hospitality.

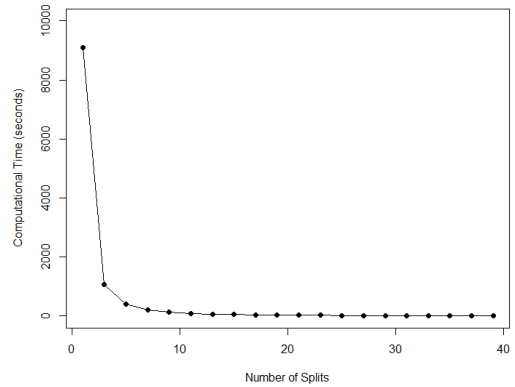


Figure 9. *Computational time of aggregation procedures for MSD.*

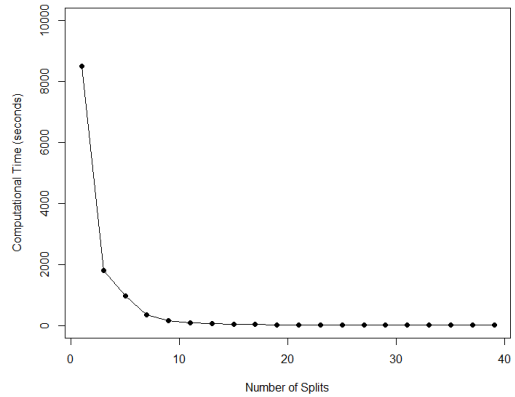


Figure 10. *Computational time of aggregation procedures for FDD.*

8. APPENDIX

This appendix section contains the proofs of the main results. Section 8.1 contains proof of Theorem 1 and relevant preliminary results. Section 8.2 includes the proof of Theorem 2. Sections 8.3 and 4.4 includes the proofs of Theorems 3 and 4, i.e., coverage properties of the credible sets based on strong and weak topology respectively.

All proofs crucially depend on an eigensystem designed for simultaneous diagonalization of the two bilinear functionals U, V induced from likelihood and prior, respectively. In fact, (φ_ν, ρ_ν) is a solution of the following ordinary differential system (whose existence and uniqueness is guaranteed by Birkhoff (1908)):

$$\begin{aligned} &(-1)^m \varphi_\nu^{(2m)}(\cdot) = \rho_\nu \pi(\cdot)\varphi_\nu(\cdot), \\ &\varphi_\nu^{(j)}(0) = \varphi_\nu^{(j)}(1) = 0, \quad j = m, m+1, \ldots, 2m-1, \end{aligned} \tag{1}$$

Properties of this eigen-system are summarized in Proposition 1, whose proof can be found in (Shang et al., 2013, Proposition 2.2).

Proposition 1 *It holds that* $\sup_{\nu\in\mathbb{N}} \|\varphi_\nu\|_\infty < \infty$, *and that the sequence* ρ_ν *is nondecreasing with* $\rho_1 = \cdots = \rho_m = 0$, *and* $\rho_\nu > 0$ *for* $\mu > m$. *Moreover,* $\rho_\nu \asymp \nu^{2m}$ *and*

$$V(\varphi_\mu, \varphi_\nu) = \delta_{\mu\nu}, \quad J(\varphi_\mu, \varphi_\nu) = \rho_\mu \delta_{\mu\nu}, \ \mu, \nu \in \mathbb{N}, \tag{2}$$

where $\delta_{\mu\nu}$ *is the Kronecker's delta. In particular, any* $f \in S^m(\mathbb{I})$ *admits a Fourier expansion* $f = \sum_\nu V(f, \varphi_\nu)\varphi_\nu$ *with convergence held in the* $\|\cdot\|$*-norm.*

8.1. Proofs in Section 4.1

The proof of Theorem 1 requires the following technical result which derives a local contraction rate $\widetilde{r}_n$ uniformly over s: $\widetilde{r}_n = (nh/\log 2s)^{-1/2} + h^{m+\frac{\beta-1}{2}}$. The proof can be found in (Shang and Cheng).

Proposition 1 *If* f_0 *satisfies Condition (**S**) and the following Rate Condition (**R**) holds:*

$$nh^{2m+1} \geq 1, \ a_n = O(\widetilde{r}_n), \ b_n \leq 1, \ r_n^2 b_n \leq \widetilde{r}_n^2.$$

Let $a \geq 0$ *be a fixed constant. Then for any* $\varepsilon \in (0,1)$, *there exist positive constants* M', N' *s.t. for any* $n \geq N'$,

$$P_{f_0}\left(\max_{1\leq j\leq s}\{E\{\|f - f_0\|^a I(\|f - f_0\| \geq M'\widetilde{r}_n)|\boldsymbol{D}_j\} \geq M' s^2 \exp(-n\widetilde{r}_n^2/\log(2s))\right) \leq \varepsilon \tag{3}$$

We remark that Proposition 1 significantly generalizes the classical results in Ghosal et al. (2000); van der Vaart et al. (2008a).

Proof [Proof of Theorem 1] Let M_1, M_2 be large positive constants. For any fixed constant $a \geq 0$, consider three events:

$$\begin{aligned} \mathcal{E}_n' &= \{\max_{1\leq j\leq s} \|\widehat{f}_{j,n} - f_0\| \leq M_1 \widetilde{r}_n\} \\ \mathcal{E}_n'' &= \{\max_{1\leq j\leq s} E\{\|f - f_0\|^a I(\|f - f_0\| \geq M_2\widetilde{r}_n)|\mathbf{D}_j\} \leq M_2 s^2 \exp(-n\widetilde{r}_n^2/\log(2s))\} \\ \mathcal{E}_n''' &= \{\max_{1\leq j\leq s} E_{0j}\{\|f - f_0\|^a I(\|f - f_0\| \geq M_2\widetilde{r}_n)\} \leq M_2 \exp(-n\widetilde{r}_n^2)\} \end{aligned}$$

where E_{0j} means expectation taken under P_{0j}. It follows from Shang and Cheng and Proposition 1 that we can choose $M_1 > M_2$ (both large enough) s.t. $P_{f_0}(\mathcal{E}_n' \cap \mathcal{E}_n'') \geq 1 - \varepsilon_1/2$ where $\varepsilon_1 > 0$ is an arbitrary constant. Meanwhile, by (Shang and Cheng) we have, on $\mathcal{E}_n'$, for any $1 \leq j \leq s$,

$$\begin{aligned} & E_{0j}\{\|f - f_0\|^a I(\|f - f_0\| \geq M_2\widetilde{r}_n)\} \\ = & \frac{\int_{\|f-f_0\|\geq M_2\widetilde{r}_n} \|f - f_0\|^a \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f)}{\int_{S^m(\mathbb{I})} \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f)} \\ \leq & \frac{\int_{\|f-f_0\|\geq M_2\widetilde{r}_n} \|f - f_0\|^a \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f)}{\int_{\|f-f_0\|\leq\widetilde{r}_n} \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f)} \\ \leq & \exp\left(-\left((M_2 - M_1)^2/2 - (M_1+1)^2/2 - c_3/4\right) n\widetilde{r}_n^2\right) C(a, \Pi), \end{aligned} \tag{4}$$

where $c_3 > 0$ is a universal constant and $C(a, \Pi) = \int_{S^m(\mathbb{I})} \|f - f_0\|^a d\Pi(f)$. We can choose $M_2 > C(a, \Pi)$ so that the quantity (4) is less than $M_2 \exp(-n\widetilde{r}_n^2)$. So $\mathcal{E}_n'$ implies $\mathcal{E}_n'''$, so that $P_{f_0}(\mathcal{E}_n''') \geq P_{f_0}(\mathcal{E}_n' \cap \mathcal{E}_n'') \geq 1 - \varepsilon_1/2$. Define $\mathcal{E}_n = \mathcal{E}_n' \cap \mathcal{E}_n'' \cap \mathcal{E}_n'''$, then it can be seen that $P_{f_0}(\mathcal{E}_n) \geq 1 - \varepsilon_1$.

Let T_j be defined as

$$T_{j2}(f) \quad = \quad -\frac{1}{2n}\sum_{i\in I_j}[(\Delta f)(X_i)^2 - E_X\{(\Delta f)(X)^2\}]. \tag{5}$$

Following Lemma 9, for any $1 \leq j \leq s$,

$$\ell_{jn}(f) - \ell_{jn}(\widehat{f}_{j,n}) + \frac{1}{2}\|f - \widehat{f}_{j,n}\|^2 = T_j(f). \tag{6}$$

It follows from the proof of Proposition 1 that on $\mathcal{E}_n$, for any $f \in S^m(\mathbb{I})$ satisfying $\|f - f_0\| \leq M_2\widetilde{r}_n$ and $1 \leq j \leq s$,

$$|T_j(f)| \leq D \times \widetilde{r}_n^2 b_n, \tag{7}$$

where $D = D(M_1, M_2)$ is a positive constant depending only on M_1, M_2. Recall that our assumption says that $\varepsilon_2 \equiv nD\widetilde{r}_n^2 b_n = o(1)$.

For $1 \leq j \leq s$, define

$$\begin{aligned} J_{nj1} &= \int_{S^m(\mathbb{I})} \exp\left(n(\ell_{jn}(f) - \ell_{jn}(\widehat{f}_{j,n}))\right) d\Pi(f), \\ J_{nj2} &= \int_{S^m(\mathbb{I})} \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f), \end{aligned}$$

$$\begin{aligned} \bar{J}_{nj1} &= \int_{\|f-f_0\|\leq M_2\widetilde{r}_n} \exp\left(n(\ell_{jn}(f) - \ell_{jn}(\widehat{f}_{j,n}))\right) d\Pi(f), \\ \bar{J}_{nj2} &= \int_{\|f-f_0\|\leq M_2\widetilde{r}_n} \exp\left(-\frac{n}{2}\|f - \widehat{f}_{j,n}\|^2\right) d\Pi(f). \end{aligned}$$

For simplicity, let $\varepsilon_3 = M_2 s^2 \exp(-n\widetilde{r}_n^2/\log(2s))$. On $\mathcal{E}_n$ (with $a = 0$) and for any $1 \leq j \leq s$,

$$0 \leq \frac{J_{nj1} - \bar{J}_{nj1}}{J_{nj1}} \leq M_2 s^2 \exp(-n\widetilde{r}_n^2/\log(2s)) = \varepsilon_3, \quad 0 \leq \frac{J_{nj2} - \bar{J}_{nj2}}{J_{nj2}} \leq \exp(-n\widetilde{r}_n^2) \leq \varepsilon_3.$$

By some algebra, it can be shown that the above inequalities lead to

$$(1-\varepsilon_3)\cdot\frac{\bar{J}_{nj2}}{\bar{J}_{nj1}} \le \frac{J_{nj2}}{J_{nj1}} \le \frac{1}{1-\varepsilon_3}\cdot\frac{\bar{J}_{nj2}}{\bar{J}_{nj1}}. \tag{8}$$

Meanwhile, on $\mathcal{E}_n$ and for any $1 \le j \le s$, using (7) and the elementary inequality $|\exp(x)-1| \le 2|x|$ for $|x| \le \log 2$, we get that

$$\begin{aligned} |\bar{J}_{nj2} - \bar{J}_{nj1}| &\le \int_{\|f-f_0\|\le M_2\widetilde{r}_n} \exp\left(-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)\times|\exp(nT_j(f))-1|d\Pi(f) \\ &\le 2\varepsilon_2\bar{J}_{nj2}, \end{aligned}$$

leading to that

$$\frac{1}{1+2\varepsilon_2} \le \frac{\bar{J}_{nj2}}{\bar{J}_{nj1}} \le \frac{1}{1-2\varepsilon_2}. \tag{9}$$

Combining (8) and (9), on $\mathcal{E}_n$ and for any $1 \le j \le s$, $\frac{1-\varepsilon_3}{1+2\varepsilon_2} \le \frac{J_{nj2}}{J_{nj1}} \le \frac{1}{(1-2\varepsilon_2)(1-\varepsilon_3)}$. When n is large, $\varepsilon_3 \le \varepsilon_2$ and both quantities are small, the above inequalities lead to

$$-4\varepsilon_2 \le \frac{1-\varepsilon_3}{1+2\varepsilon_2} - 1 \le \frac{J_{nj2}}{J_{nj1}} - 1 \le \frac{1}{(1-2\varepsilon_2)(1-\varepsilon_3)} - 1 \le 4\varepsilon_2 \tag{10}$$

For simplicity, denote $R_{nj}(f) = nT_j(f)$. For any $S \in \mathcal{S}$, let $S' = S \cap \{f \in S^m(\mathbb{I}) : \|f - f_0\| \le M_2\widetilde{r}_n\}$. Then on $\mathcal{E}_n$, we get that $\max_{1\le j\le s}|P(S|\mathbf{D}_j) - P_{0j}(S)| \le \max_{1\le j\le s}|P(S'|\mathbf{D}_j) - P_{0j}(S')| + 2\varepsilon_3$. Moreover, it follows from (10) that on $\mathcal{E}_n$ and for any $1 \le j \le s$,

$$\begin{aligned} &|P(S'|\mathbf{D}_j) - P_{0j}(S')| \\ =\ & \left|\int_{S'}\left(\frac{\exp(n(\ell_{jn}(f)-\ell_{jn}(\widehat{f}_{j,n})))}{J_{nj1}} - \frac{\exp\left(-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)}{J_{nj2}}\right)d\Pi(f)\right| \\ \le\ & \int_{S'}\exp\left(-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)\times\left|\frac{\exp(R_{nj}(f))}{J_{nj1}} - \frac{1}{J_{nj2}}\right|d\Pi(f) \\ \le\ & \int_{S'}\exp\left(-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)\times\frac{|\exp(R_{nj}(f))-1|}{J_{nj2}}d\Pi(f) \\ &+\int_{S'}\exp\left(-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)\times\exp(R_{nj}(f))\times\left|\frac{1}{J_{nj1}} - \frac{1}{J_{nj2}}\right|d\Pi(f) \\ \le\ & 2\varepsilon_2\frac{\int_{S'}\exp\left(-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)d\Pi(f)}{J_{nj2}} \\ &+\exp(\varepsilon_2)\times\left|\frac{1}{J_{nj1}} - \frac{1}{J_{nj2}}\right|\times\int_{S'}\exp\left(-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)d\Pi(f) \\ \le\ & 2\varepsilon_2 + \exp(\varepsilon_2)\times\left|\frac{J_{nj2}}{J_{nj1}} - 1\right| \le 2\varepsilon_2 + 4\varepsilon_2\exp(\varepsilon_2) \le 14\varepsilon_2. \end{aligned}$$

Note that the right hand side is free of S. Then we get that on $\mathcal{E}_n$, $\sup_{S\in\mathcal{S}}\max_{1\le j\le s}|P(S|\mathbf{D}_j) - P_{0j}(S)| \le 14\varepsilon_2 + 2\varepsilon_3 \le 16\varepsilon_2$. This implies that for sufficiently large n,

$$\begin{aligned} & P_{f_0}\left(\sup_{S\in\mathcal{S}}\max_{1\le j\le s}|P(S|\mathbf{D}_j) - P_{0j}(S)| > 16\varepsilon_2\right) \\ \le\ & P_{f_0}(\mathcal{E}_n^c) + P_{f_0}\left(\mathcal{E}_n, \sup_{S\in\mathcal{S}}\max_{1\le j\le s}|P(S|\mathbf{D}_j) - P_{0j}(S)| > 16\varepsilon_2\right) = P_{f_0}(\mathcal{E}_n^c) \le \varepsilon_1. \end{aligned}$$

The desirable result follows by the simple fact $\varepsilon_2 \lesssim \sqrt{s}N^{-\frac{4m^2+2m\beta-10m+1}{4m(2m+\beta)}}(\log N)^{\frac{5}{2}}$ when $h \asymp h^*$. ■

8.2. Proofs in Section 4.2

Proof [Proof of Theorem 2] We first show (28). Let $A_n = \{f \in S^m(\mathbb{I}) : \|f - f_0\| \ge M\widetilde{r}_n\}$ and $B_j = \{f \in S^m(\mathbb{I}) : dP(f|\mathbf{D}_j) \ge dP_{0j}(f)\}$ for $1 \le j \le s$. By Proposition 1, Theorem 1 and (4) with $a = 1$ therein, we can choose $M > 0$ sufficiently large such that

$$\begin{aligned} &\max_{1\le j\le s}\|E(f|\mathbf{D}_j) - E_{0j}(f)\| \\ =\ & \max_{1\le j\le s}\left\|\int(f-f_0)dP(f|\mathbf{D}_j) - \int(f-f_0)dP_{0j}(f)\right\| \\ \le\ & \max_{1\le j\le s}\left\|\int_{A_n}(f-f_0)dP(f|\mathbf{D}_j)\right\| + \max_{1\le j\le s}\left\|\int_{A_n}(f-f_0)dP_{0j}(f)\right\| \\ &+\max_{1\le j\le s}\left\|\int_{A_n^c}(f-f_0)(dP(f|\mathbf{D}_j) - dP_{0j}(f))\right\| \\ \le\ & \max_{1\le j\le s}E\{\|f-f_0\|I(f\in A_n)|\mathbf{D}_j\} + \max_{1\le j\le s}E_{0j}\{\|f-f_0\|I(f\in A_n)\} \\ &+M\widetilde{r}_n\max_{1\le j\le s}\int_{A_n^c}|dP(f|\mathbf{D}_j) - dP_{0j}(f)| \\ =\ & O_{P_{f_0}}\left(s^2\exp(-n\widetilde{r}_n^2/\log(2s)) + \exp(-n\widetilde{r}_n^2) + \widetilde{r}_n\sqrt{s}N^{-\frac{4m^2+2m\beta-10m+1}{4m(2m+\beta)}}(\log N)^{\frac{5}{2}}\right) \\ =\ & O_{P_{f_0}}\left(\widetilde{r}_n\sqrt{s}N^{-\frac{4m^2+2m\beta-10m+1}{4m(2m+\beta)}}(\log N)^{\frac{5}{2}}\right) \equiv O_{P_{f_0}}(L_N), \end{aligned}$$

where the second last equality uses Theorem 1 and the fact that, uniformly for j,

$$\begin{aligned} &\int_{A_n^c}|dP(f|\mathbf{D}_j) - dP_{0j}(f)| \\ =\ & |P(A_n^c\cap B_j|\mathbf{D}_j) - P_{0j}(A_n^c\cap B_j)| + |P(A_n^c\cap B_j^c|\mathbf{D}_j) - P_{0j}(A_n^c\cap B_j^c)|. \end{aligned}$$

Then (28) follows from the trivial fact that $E_{0j}\{f\} = E(W^j|\mathbf{D}_j) = \widetilde{f}_{j,n}$.

Next we show (30). By direct examinations we can verify the following Rate Conditions ($\mathbf{R}$):

$$n\widetilde{r}_n^2 b_n = o(1), N\widetilde{r}_N^2 b_N = o(1), Nh^{1/2}a_N^2 = o(1), Nh^{1/2}a_n^2 = o(1).$$

Define $Rem_{j,n} = \widehat{f}_{j,n} - f_0 - S_{j,n}(f_0)$ for $j = 1, 2, \ldots, s$. It follows by Lemma 6 of Shang and Cheng that $\max_{1\le j\le s}\|Rem_{j,n}\| = O_{P_{f_0}}(a_n)$.

It is easy to see that $a_{N,\nu}/a_{n,\nu} \le s$ for all $\nu \ge 1$. Then it holds from (38) that

$$\begin{aligned}\|\breve{f}_{N,\lambda} - \widetilde{f}_{N,\lambda}\|^2 &= \sum_{\nu\ge1}\left(\frac{a_{N,\nu}}{a_{n,\nu}}\right)^2 V\left(\frac{1}{s}\sum_{j=1}^{s}(\breve{f}_{j,n} - \widetilde{f}_{j,n}), \varphi_\nu\right)^2 (1+\lambda\rho_\nu)\\ &\le s^2\|\frac{1}{s}\sum_{j=1}^{s}(\breve{f}_{j,n} - \widetilde{f}_{j,b})\|^2 = O_{P_{f_0}}\left(s^2 L_N^2\right) = o_{P_{f_0}}(N^{-1}h^{-1/2}). \qquad (11)\end{aligned}$$

The last equality owes to the condition $s^4\log(2s) = o\left(N^{\frac{4m^2+2m\beta-11m+1}{2m(2m+\beta)}}(\log N)^{-5}\right)$ and $\beta > 3/2$.

By direct examinations, we have

$$\begin{aligned}\widetilde{f}_{N,\lambda} - f_0 &= \sum_{\nu=1}^{\infty}\left(a_{N,\nu}\left(\frac{1}{s}\sum_{j=1}^{s}V(\widehat{f}_{j,n},\varphi_\nu)\right) - f_\nu^0\right)\varphi_\nu\\ &= \sum_{\nu=1}^{\infty}\left(a_{N,\nu}\left(\frac{1}{s}\sum_{j=1}^{s}V(Rem_{j,n} + f_0 + S_{j,n}(f_0),\varphi_\nu)\right) - f_\nu^0\right)\varphi_\nu\\ &= \sum_{\nu=1}^{\infty}a_{N,\nu}V(\frac{1}{s}\sum_{j=1}^{s}Rem_{j,n},\varphi_\nu)\varphi_\nu + \sum_{\nu=1}^{\infty}(a_{N,\nu}-1)f_\nu^0\varphi_\nu\\ &\quad + \sum_{\nu=1}^{\infty}a_{N,\nu}V(\frac{1}{N}\sum_{i=1}^{N}\epsilon_i K_{X_i},\varphi_\nu)\varphi_\nu - \sum_{\nu=1}^{\infty}a_{N,\nu}V(\mathcal{P}_\lambda f_0,\varphi_\nu)\varphi_\nu. \qquad (12)\end{aligned}$$

Denote the four terms in the above equation by T_1, T_2, T_3, T_4.

Since $a_{N,\nu} \le 1$, it is easy to see that

$$\begin{aligned}\|T_1\|_2^2 &= \sum_{\nu=1}^{\infty}a_{N,\nu}^2|V(\frac{1}{s}\sum_{j=1}^{s}Rem_{j,n},\varphi_\nu)|^2\\ &\le \sum_{\nu=1}^{\infty}|V(\frac{1}{s}\sum_{j=1}^{s}Rem_{j,n},\varphi_\nu)|^2 = \|\frac{1}{s}\sum_{j=1}^{s}Rem_{j,n}\|_2^2 \le (\max_{1\le j\le s}\|Rem_{j,n}\|)^2 = O_{P_{f_0}}(a_n^2).\end{aligned} \qquad (13)$$

Using $h \asymp N^{-1/(2m+\beta)}$ and a direct algebra we get that

$$\|T_2\|_2^2 = \sum_{\nu=1}^{\infty}(a_{N,\nu}-1)^2|f_\nu^0|^2 \asymp \sum_{\nu=1}^{\infty}\left(\frac{\nu^{2m+\beta}}{\nu^{2m+\beta}+N(1+\lambda\nu^{2m})}\right)^2|f_\nu^0|^2 = o(N^{-\frac{2m+\beta-1}{2m+\beta}}) = o(N^{-1}h^{-1}).$$

Meanwhile, it follows by Proposition Shang and Cheng that

$$\begin{aligned}\|T_4\|_2^2 &= \sum_{\nu=1}^{\infty}a_{N,\nu}^2|f_\nu^0|^2\left(\frac{\lambda\rho_\nu}{1+\lambda\rho_\nu}\right)^2 \le \sum_{\nu=1}^{\infty}|f_\nu^0|^2\left(\frac{\lambda\rho_\nu}{1+\lambda\rho_\nu}\right)^2\\ &\lesssim \sum_{\nu=1}^{\infty}|f_\nu^0|^2(h\nu)^{2m+\beta-1}\frac{(h\nu)^{2m-\beta+1}}{(1+(h\nu)^{2m})^2} = o(N^{-\frac{2m+\beta-1}{2m+\beta}}) = o(N^{-1}h^{-1}).\end{aligned}$$

Define $R(x,x') = \sum_{\nu=1}^{\infty}a_{N,\nu}\frac{\varphi_\nu(x)\varphi_\nu(x')}{1+\lambda\rho_\nu}$ for any $x, x' \in \mathbb{I}$. Also define $R_x(\cdot) = R(x,\cdot)$. It is easy to see that $R_x \in S^m(\mathbb{I})$ for any $x \in \mathbb{I}$. Then it can be shown that $T_3 = \frac{1}{N}\sum_{i=1}^{N}\epsilon_i R_{X_i}$,

leading to

$$\|T_3\|_2^2 = V(T_3,T_3) = \frac{1}{N^2}\sum_{i=1}^{N}\epsilon_i^2 V(R_{X_i},R_{X_i}) + \frac{2}{N^2}\sum_{i<k}\epsilon_i\epsilon_k V(R_{X_i},R_{X_k}).$$

Since $E_{f_0}\{\epsilon^2 V(R_X,R_X)\} = O(h^{-1})$, we have $E_{f_0}\{\|T_3\|_2^2\} = O(N^{-1}h^{-1})$. Therefore, $\|\widetilde{f}_{N,\lambda} - f_0\|_2^2 = O_{P_{f_0}}(N^{-1}h^{-1}) = O_{P_{f_0}}\left(N^{-\frac{2m+\beta-1}{2m+\beta}}\right)$. This together with (11) leads to (30). ■

8.3. Proofs in Section 4.3

Before proving Theorem 3, we give some preliminary notation and results. Define an "oracle" penalized likelihood $\ell_{N,\lambda}(f) = -\frac{1}{2N}\sum_{i=1}^{N}(Y_i - f(X_i))^2 - \frac{\lambda}{2}J(f)$. Applying Theorem 1 to $s = 1$, we have

$$\sup_{S\in\mathcal{S}}|P(S|\mathbf{D}) - P_0(S)| = o_{P_{f_0}}(1), \qquad (14)$$

where $P_0(S) = \frac{\int_S \exp\left(-\frac{N}{2}\|f-\widehat{f}_{N,\lambda}^{or}\|^2\right)d\Pi(f)}{\int_{S^m(\mathbb{I})}\exp\left(-\frac{N}{2}\|f-\widehat{f}_{N,\lambda}^{or}\|^2\right)d\Pi(f)}$ and $\widehat{f}_{N,\lambda}^{or} = \arg\max_{f\in S^m(\mathbb{I})}\ell_{N,\lambda}(f)$ is the "oracle" smoothing spline estimator based on full data. Consider a generalized Fourier expansion of $\widehat{f}_{N,\lambda}^{or}$: $\widehat{f}_{N,\lambda}^{or}(\cdot) = \sum_{\nu=1}^{\infty}V(\widehat{f}_{N,\lambda}^{or},\varphi_\nu)\varphi_\nu(\cdot)$. By Theorem 5.2 in Shang and Cheng (2017), we have $P_0(S) = P(W^{or} \in S|\mathbf{D})$ for any $S \in \mathcal{S}$, where $W^{or}(\cdot) = \sum_{\nu=1}^{\infty}(a_{N,\nu}V(\widehat{f}_{N,\lambda}^{or},\varphi_\nu) + b_{N,\nu}\tau_\nu v_\nu)\varphi_\nu(\cdot)$. Here, $a_{n,\nu}$ $b_{n,\nu}$ are analogous to ones in the definition of $W^j(\cdot)$ in Section 4.1, and $v_\nu \sim N(0,\tau_\nu^{-2})$ and τ_ν^2 are given in (25). Define the mean functions of W^{or} as $\widetilde{f}_{N,\lambda}^{or}(\cdot) := \sum_{\nu=1}^{\infty}a_{N,\nu}V(\widehat{f}_{N,\lambda}^{or},\varphi_\nu)\varphi_\nu(\cdot)$. So we can re-express W^{or} as $W^{or} = \widetilde{f}_{N,\lambda}^{or} + W_N$, where $W_N(\cdot) := \sum_{\nu=1}^{\infty}b_{N,\nu}\tau_\nu v_\nu\varphi_\nu(\cdot)$ is a zero-mean GP.

The following result describes the distribution of W_n and W_N.

Lemma 2 *As $N \to \infty$, $\frac{n\|W_n\|_2^2-\zeta_{1,n}}{\sqrt{2\zeta_{2,n}}} \xrightarrow{d} N(0,1)$, and $\frac{N\|W_N\|_2^2-\zeta_{1,N}}{\sqrt{2\zeta_{2,N}}} \xrightarrow{d} N(0,1)$.*

Proof [Proof of Theorem 3] We can show that Rate Conditions (**R**) hold by direct calculations.

It is sufficient to investigate the P_{f_0}-probability of the event $\{\|f_0 - \breve{f}_{N,\lambda}\|_2 \le r_N(\alpha)\}$. To achieve this goal, we first prove the following fact:

$$\max_{1\le j\le s}|z_{j,n}(\alpha) - z_\alpha| = o_{P_{f_0}}(1), \qquad (15)$$

where $z_\alpha = \Phi^{-1}(1-\alpha)$ and Φ is the c.d.f. of $N(0,1)$, and $z_{j,n}(\alpha) = (nr_{j,n}(\alpha)^2 - \zeta_{1,n})/\sqrt{2\zeta_{2,n}}$. The proof of the theorem follows by (15) and a careful analysis of $f_0 - \breve{f}_{N,\lambda}$.

We first show (15). It follows by Theorem 1 that for any $j = 1, 2, \ldots, s$,

$$\begin{aligned}|P(R_{j,n}(\alpha)|\mathbf{D}_j) - P_{0j}(R_{j,n}(\alpha))| &\le \max_{1\le k\le s}|P(R_{j,n}(\alpha)|\mathbf{D}_k) - P_{0k}(R_{j,n}(\alpha))|\\ &\le \sup_{S\in\mathcal{S}}\max_{1\le k\le s}|P(S|\mathbf{D}_k) - P_{0k}(S)| = o_{P_{f_0}}(1).\end{aligned}$$

Together with $P(R_{j,n}(\alpha)|\mathbf{D}_j) = 1-\alpha$, we have $\max_{1\le j\le s}|P_{0j}(R_{j,n}(\alpha)) - (1-\alpha)| = o_{P_{f_0}}(1)$. Let $\Delta_j = \breve{f}_{j,n} - \widetilde{f}_{j,n}$ for $1 \le j \le s$. It is clear that

$$\begin{aligned} P_{0j}(R_{j,n}(\alpha)) &= P(W^j \in R_{j,n}(\alpha)|\mathbf{D}_j) = P(\|W_n + \Delta_j\|_2 \le r_{j,n}(\alpha)|\mathbf{D}_j) \\ &= P(\|W_n\|_2^2 + 2\langle W_n, \Delta_j\rangle_2 + \|\Delta_j\|_2^2 \le r_{j,n}(\alpha)^2|\mathbf{D}_j), \end{aligned} \tag{16}$$

and, for any $\varepsilon \in (0,1)$,

$$\begin{aligned} & P(|\langle W_n, \Delta_j\rangle_2|^2 \ge \|\Delta_j\|_2^2/(n\varepsilon)|\mathbf{D}_j) \le n\varepsilon E\{|\langle W_n, \Delta_j\rangle_2|^2|\mathbf{D}_j\}/\|\Delta_j\|_2^2 \\ = & \frac{n\varepsilon}{\|\Delta_j\|_2^2}\sum_{\nu\ge1} b_{n,\nu}^2|V(\Delta_j, \varphi_\nu)|^2 \le \frac{n\varepsilon}{\|\Delta_j\|_2^2} \times \frac{\|\Delta_j\|_2^2}{n} = \varepsilon, \end{aligned} \tag{17}$$

and by Theorem 2, $\max_{1\le j\le s}\|\Delta_j\|_2^2 = O_{P_{f_0}}(L_N^2)$, where $L_N = \widetilde{r}_n\sqrt{s}N^{-\frac{4m^2+2m\beta-10m+1}{4m(2m+\beta)}}(\log N)^{\frac{5}{2}}$. By (29), $\zeta_{k,n} \asymp n^{1/(2m+\beta)}$ (Lemma 2), and direct examinations it holds that

$$\max_{1\le j\le s}\frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} = o_{P_{f_0}}(1). \tag{18}$$

Combining (16) and (17) we get that

$$\begin{aligned} P_{0j}(R_{j,n}(\alpha)) &\ge \Phi_n\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} - \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right) - \varepsilon, \\ P_{0j}(R_{j,n}(\alpha)) &\le \Phi_n\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} + \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right) + \varepsilon, \end{aligned}$$

where Φ_n is the c.d.f. of U_n. It follows by Lemma 2 and Polya's theorem (Chow and Teicher (2012)) that Φ_n uniformly converges to $\Phi(\cdot)$, the c.d.f. of standard normal variable. Therefore, when n becomes large enough,

$$\left|\Phi_n\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} - \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right) - \Phi\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} - \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right)\right| \le \varepsilon,$$

$$\left|\Phi_n\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} + \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right) - \Phi\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} + \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right)\right| \le \varepsilon,$$

where implies that

$$\Phi\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} - \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right) \le P_{0j}(R_{j,n}(\alpha)) + 2\varepsilon = \Phi(z_\alpha) + 2\varepsilon + o_{P_{f_0}}(1),$$

$$\Phi\left(z_{j,n}(\alpha) - \frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}} + \frac{2n\|\Delta_j\|_2}{\sqrt{n\varepsilon\zeta_{2,n}}}\right) \ge P_{0j}(R_{j,n}(\alpha)) - 2\varepsilon = \Phi(z_\alpha) - 2\varepsilon + o_{P_{f_0}}(1).$$

Since (18) implies that $\frac{n\|\Delta_j\|_2^2}{\sqrt{\zeta_{2,n}}}$ and $\frac{2\sqrt{n}\|\Delta_j\|_2}{\sqrt{\zeta_{2,n}}}$ are both $o_{P_{f_0}}(1)$ uniformly for j, so (15) holds.

Next we prove the theorem. Consider expansion (12). Only focus on T_3. Define $W(N) = 2\sum_{1\le i<k\le N}\epsilon_i\epsilon_k V(R_{X_i}, R_{X_k})$. Let $W_{ik} = 2\epsilon_i\epsilon_k V(R_{X_i}, R_{X_k})$, then $W(N) = \sum_{1\le i<k\le N} W_{ik}$. Note that $W(N)$ is clean in the sense of de Jong (1987). Let $\sigma^2(N) = E_{f_0}\{W(N)^2\}$ and G_I, G_{II}, G_{IV} be defined as $G_I = \sum_{i<j}E_{f_0}\{W_{ij}^4\}$, $G_{II} = \sum_{i<j<k}(E_{f_0}\{W_{ij}^2W_{ik}^2\} + E_{f_0}\{W_{ji}^2W_{jk}^2\} + E_{f_0}\{W_{ki}^2W_{kj}^2\})$, and

$$G_{IV} = \sum_{i<j<k<l}(E_{f_0}\{W_{ij}W_{ik}W_{lj}W_{lk}\} + E_{f_0}\{W_{ij}W_{il}W_{kj}W_{kl}\} + E_{f_0}\{W_{ik}W_{il}W_{jk}W_{jl}\}).$$

Since φ_ν are uniformly bounded, we get that $\|R_x\|_2^2 = \sum_{\nu=1}^\infty \frac{|\varphi_\nu(x)|^2}{(1+N^{-1}\tau_\nu^2+\lambda\rho_\nu)^2} \lesssim h^{-1}$, where "$\lesssim$" is free of x. This implies that $G_I = O(N^2h^{-4})$ and $G_{II} = O(N^3h^{-4})$.

It can also be shown that for pairwise distinct i,k,t,l,

$$\begin{aligned} E_{f_0}\{W_{ik}W_{il}W_{tk}W_{tl}\} &= 2^4E_{f_0}\{\epsilon_i^2\epsilon_k^2\epsilon_t^2\epsilon_l^2V(R_{X_i}, R_{X_k})V(R_{X_i}, R_{X_l})V(R_{X_t}, R_{X_k})V(R_{X_t}, R_{X_l})\} \\ &= 2^4\sum_{\nu=1}^\infty\frac{a_{N,\nu}^8}{(1+\lambda\rho_\nu)^8} = O(h^{-1}), \end{aligned}$$

which implies that $G_{IV} = O(N^4h^{-1})$. In the mean time, a straight algebra leads to that

$$\sigma^2(N) = 4\binom{N}{2}\sum_{\nu=1}^\infty\frac{a_{N,\nu}^4}{(1+\lambda\rho_\nu)^4} = 4\binom{N}{2}\sum_{\nu=1}^\infty\left(\frac{N}{\tau_\nu^2+N(1+\lambda\rho_\nu)}\right)^4 = 2N(N-1)\zeta_{4,N} \asymp N^2h^{-1}.$$

Since $Nh^2 \asymp N^{1-2/(2m+\beta)} \to \infty$, we get that G_I, G_{II} and G_{IV} are all of order $o(\sigma^4(N))$. Then it follows by de Jong (1987) that as $N\to\infty$, $\frac{W(N)}{N\sqrt{2\zeta_{4,N}}} \xrightarrow{d} N(0,1)$. Since $\zeta_{4,N} \asymp h^{-1}$, the above equation leads to that $W(N)/N = O_{P_{f_0}}(h^{-1/2})$.

It follows by direct examination that $Var_{f_0}\{\sum_{i=1}^N\epsilon_i^2V(R_{X_i}, R_{X_i})\} \le NE_{f_0}\{\epsilon_i^4\|R_{X_i}\|_2^4\} = O(Nh^{-2})$, leading to that $\sum_{i=1}^N\epsilon_i^2V(R_{X_i}, R_{X_i}) = E_{f_0}\{\sum_{i=1}^N\epsilon_i^2V(R_{X_i}, R_{X_i})\} + O_{P_{f_0}}(N^{1/2}h^{-1}) = N\zeta_{2,N} + O_{P_{f_0}}(N^{1/2}h^{-1})$. Therefore, it follows by Rate Condition ($\mathbf{R}$), i.e., $Nha_n^2 = o(1)$, and the analysis on T_1, T_2, T_3, T_4 in (12) that

$$Nh\|\widetilde{f}_{N,\lambda} - f_0\|_2^2 = Nh\|T_3\|_2^2 + O_{P_{f_0}}(Nha_n^2) + o_{P_{f_0}}(1) = h\zeta_{2,N} + o_{P_{f_0}}(1). \tag{19}$$

In the end, note from (15) and $\zeta_{k,n} \asymp n^{\alpha_1}$ for $\alpha_1 = 1/(2m+\beta)$ (see proof of Lemma 2) that $\frac{n}{s}\sum_{j=1}^s r_{j,n}(\alpha)^2 = \zeta_{1,n} + \sqrt{2\zeta_{2,n}}z_\alpha + o_{P_{f_0}}(\sqrt{\zeta_{2,n}})$, which leads to that

$$Nr_N(\alpha)^2 = \zeta_{1,N} + \sqrt{2\zeta_{2,N}}z_\alpha + o_{P_{f_0}}(h^{-1/2}). \tag{20}$$

Therefore, $Nhr_N(\alpha)^2 = h\zeta_{1,N}(1 + o_{P_{f_0}}(1))$. Since $\liminf_{N\to\infty}(h\zeta_{1,N} - h\zeta_{2,N}) > 0$, we get by (11) that, with P_{f_0}-probability approaching one, $\|\breve{f}_{N,\lambda} - f_0\|_2 \le r_N(\alpha)$. Meanwhile, it follows by Shang and Cheng that $\|\widehat{f}_{N,\lambda}^{or} - f_0 - S_{N,\lambda}(f_0)\|_2 = O_{P_{f_0}}(a_N)$ and $\|\frac{1}{s}\sum_{j=1}^s\widehat{f}_{j,n} - f_0 - \frac{1}{s}\sum_{j=1}^s S_{j,n}(f_0)\|_2 = O_{P_{f_0}}(a_n)$, where $S_{N,\lambda}(f_0) = \frac{1}{N}\sum_{i=1}^N\epsilon_iK_{X_i} - \mathcal{P}_\lambda f_0$. Note that $S_{N,\lambda}(f_0) = \frac{1}{s}\sum_{j=1}^s S_{j,n}(f_0)$, which leads to $\|\widehat{f}_{N,\lambda}^{or} - \frac{1}{s}\sum_{j=1}^s\widehat{f}_{j,n}\|_2 = O_{P_{f_0}}(a_n + a_N)$. Since $a_{N,\nu} \le 1$, we get

that

$$\begin{aligned} N\|\widetilde{f}^{or}_{N,\lambda}-\widetilde{f}_{N,\lambda}\|^2 &= N\sum_{\nu=1}^{\infty}a^2_{N,\nu}V\left(\widehat{f}^{or}_{N,\lambda}-\frac{1}{s}\sum_{j=1}^{s}\widehat{f}_{j,n},\varphi_\nu\right)^2(1+\lambda\rho_\nu)\\ &\le N\sum_{\nu=1}^{\infty}V\left(\widehat{f}^{or}_{N,\lambda}-\frac{1}{s}\sum_{j=1}^{s}\widehat{f}_{j,n},\varphi_\nu\right)^2(1+\lambda\rho_\nu)\\ &= N\|\widehat{f}^{or}_{N,\lambda}-\frac{1}{s}\sum_{j=1}^{s}\widehat{f}_{j,n}\|^2=O_{P_{f_0}}(Na_n^2+Na_N^2) \qquad (21)\\ &= o_{P_{f_0}}(h^{-1/2}),\ \ (\text{by condition } Nh^{1/2}a_n^2+Nh^{1/2}a_N^2=o(1)) \end{aligned}$$

Using (11) we get that $N\|\widetilde{f}^{or}_{N,\lambda}-\breve{f}_{N,\lambda}\|_2^2=o_{P_{f_0}}(h^{-1/2})$. Since $E\{|\langle W_N,\widetilde{f}^{or}_{N,\lambda}-\breve{f}_{N,\lambda}\rangle_2|^2|\mathbf{D}\}=\sum_{\nu\ge1}b^2_{N,\nu}V(\widetilde{f}^{or}_{N,\lambda}-\breve{f}_{N,\lambda},\varphi_\nu)^2\le\|\widetilde{f}^{or}_{N,\lambda}-\breve{f}_{N,\lambda}\|_2^2/N=o_{P_{f_0}}(N^{-2}h^{-1/2})$, we have that $N\|W^{or}-\breve{f}_{N,\lambda}\|_2^2=N\|W_N\|_2^2+o_{P_{f_0}}(h^{-1/2})$. It follows by $P\left(\frac{N\|W_N\|_2^2-\zeta_{1,N}}{\sqrt{2\zeta_{2,N}}}\le z_\alpha\right)\to1-\alpha$, (14) and (20) that $P(R_N(\alpha)|\mathbf{D})=1-\alpha+o_{P_{f_0}}(1)$. This completes the proof. ■

8.4. Proofs in Section 4.4

Before proving Theorem 4, let us present a preliminary lemma.

Lemma 3 *As $N\to\infty$, $N\|W_N\|_\omega^2\xrightarrow{d}\sum_{\nu=1}^{\infty}d_\nu\eta_\nu^2$, and $n\|W_n\|_\omega^2\xrightarrow{d}\sum_{\nu=1}^{\infty}d_\nu\eta_\nu^2$, where η_ν are independent standard normal random variables.*

Proof [Proof of Theorem 4] By direct examinations, one can show that Rate Conditions ($\mathbf{R}'$): $n\tilde{r}_n^2b_n=o(1)$, $N\tilde{r}_N^2b_N=o(1)$, $Na_N^2=o(1)$ and $Na_n^2=o(1)$ are all satisfied.

We first have the following fact:

$$\max_{1\le j\le s}|\sqrt{n}r_{\omega,j,n}(\alpha)-\sqrt{c_\alpha}|=o_{P_{f_0}}(1), \qquad (22)$$

where $c_\alpha>0$ satisfies $P(\sum_{\nu=1}^{\infty}d_\nu\eta_\nu^2\le c_\alpha)=1-\alpha$ with η_ν being independent standard normal random variables. It follows from (22) that

$$Nr_{\omega,N}(\alpha)^2=c_\alpha+o_{P_{f_0}}(1). \qquad (23)$$

By Theorem 2 and the condition $s=o(N^{\frac{4m^2+2m\beta-12m+1}{8m(2m+\beta)}}(\log N)^{-\frac{3}{2}})$ we have the following $\max_{1\le j\le s}n\|\Delta_j\|_\omega^2=\max_{1\le j\le s}n\|\Delta_j\|_2^2=O_{P_{f_0}}(nL_N^2)=o_{P_{f_0}}(1)$. Also, for arbitrarily small $\varepsilon\in(0,1)$, $P(|\langle W_n,\Delta_j\rangle_\omega|^2\ge\|\Delta_j\|_\omega^2/(n\varepsilon)|\mathbf{D}_j)\le\varepsilon$. The proof of (22) is then similar to the proof of (15) and details are omitted.

Let T_1,T_2,T_3,T_4 be defined in (12). It follows from the proof of Theorem 3 that $\|T_1\|_\omega^2\le\|T_1\|_2^2=O_{P_{f_0}}(a_n^2)$, so $N\|T_1\|_\omega^2=O_{P_{f_0}}(Na_n^2)=o_{P_{f_0}}(1)$ due to the condition $Na_n^2=o(1)$. It follows by condition $h\asymp N^{-1/(2m+\beta)}$, dominated convergence theorem and direct

examinations,

$$\begin{aligned} \|T_2\|_\omega^2 &= \sum_{\nu=1}^{\infty}d_\nu(a_{N,\nu}-1)^2|f_\nu^0|^2\asymp N^{-2}\sum_{\nu=1}^{\infty}d_\nu\frac{\nu^{2m+\beta+1}}{(1+(h\nu)^{2m}+(h\nu)^{2m+\beta})^2}\times\nu^{2m+\beta-1}|f_\nu^0|^2\\ &\lesssim N^{-1}\sum_{\nu=1}^{\infty}\frac{(h\nu)^{2m+\beta+1}}{(1+(h\nu)^{2m}+(h\nu)^{2m+\beta})^2}\times\nu^{2m+\beta-1}|f_\nu^0|^2=o(N^{-1}), \end{aligned}$$

and

$$\begin{aligned} \|T_4\|_\omega^2 &= \sum_{\nu=1}^{\infty}d_\nu a^2_{N,\nu}\left(\frac{\lambda\rho_\nu}{1+\lambda\rho_\nu}\right)^2|f_\nu^0|^2\lesssim\sum_{\nu=1}^{\infty}d_\nu\frac{(h\nu)^{2m-\beta+1}}{(1+(h\nu)^{2m}+(h\nu)^{2m+\beta})^2}\times|f_\nu^0|^2(h\nu)^{2m+\beta-1}\\ &\lesssim h^{2m+\beta}\sum_{\nu=1}^{\infty}\frac{(h\nu)^{2m-\beta}}{(1+(h\nu)^{2m}+(h\nu)^{2m+\beta})^2}\times|f_\nu^0|^2\nu^{2m+\beta-1}=o(N^{-1}). \end{aligned}$$

By direct examination it can be shown that $T_3=\frac{1}{N}\sum_{i=1}^{N}\epsilon_i\sum_{\nu=1}^{\infty}\frac{\varphi_\nu(X_i)\varphi_\nu}{1+\lambda\rho_\nu+N^{-1}\tau_\nu^2}$. It follows by Shang and Cheng (2017) that, as $N\to\infty$, $N\|T_3\|_\omega^2\xrightarrow{d}\sum_{\nu=1}^{\infty}d_\nu\eta_\nu^2$. By the above analysis on T_1 through T_4, and $N\|\breve{f}_{N,\lambda}-\widetilde{f}_{N,\lambda}\|_\omega^2=O_{P_{f_0}}(Ns^2L_N^2)=o_{P_{f_0}}(1)$, we get that $N\|\breve{f}_{N,\lambda}-f_0\|_\omega^2\xrightarrow{d}\sum_{\nu=1}^{\infty}d_\nu\eta_\nu^2$. It follows by (23) that $\lim_{N\to\infty}P_{f_0}(f_0\in R_N^\omega(\alpha))=1-\alpha$.

It follows by $N\|\widetilde{f}^{or}_{N,\lambda}-\widetilde{f}_{N,\lambda}\|_2^2=O_{P_{f_0}}(Na_N^2+Na_n^2)=o_{P_{f_0}}(1)$ (see (21)), $P(N\|W_N\|_2^2\le c_\alpha)\to1-\alpha$, (23) and (14) that $P(R_N^\omega(\alpha)|\mathbf{D})=1-\alpha+o_{P_{f_0}}(1)$. Proof is completed. ■

8.5. Computational Details

In this subsection, we provide some computational details relating to Section 2.2. For convenience, we rewrite model (2.1) as following:

$$Y_{ji}=f(X_{ji})+\epsilon_{ji},\ j=1,\dots,s,\ i=1,\dots,n.$$

Calculation of posterior means. In order to calculate the posterior mean $\breve{f}_{j,n}$, we have to generate samples of f from its posterior distribution $P(f|\{Y_{ji},X_{ji}\}_{i=1}^n)$. In practice, directly sampling the function f from $P(f|\{Y_{ji},X_{ji}\}_{i=1}^n)$ is impossible. Instead, we generate some samples from $(f(X_{j1}),\dots,f(X_{jn}))^\top$. As n is large, $(f(X_{j1}),\dots,f(X_{jn}))^\top$ can represent the whole curve of f. Firstly, let us derive the posterior distribution for $(f(X_{j1}),\dots,f(X_{jn}))^\top$. For the j-th subsample, the likelihood function is written by

$$Y_{j1},\dots,Y_{jn}|X_{j1},\dots,X_{jn}\sim N((f(X_{j1}),\dots,f(X_{jn}))^\top,I_n).$$

Since f follows a GP prior with mean zero and covariance function K_0, where K_0 is given in (5), the prior of $(f(X_{j1}),\dots,f(X_{jn}))^\top$ is multivariate Gaussian:

$$(f(X_{j1}),\dots,f(X_{jn}))^\top\sim N(0,K_j),$$

where K_j is the covariance matrix satisfying

$$K_j=\begin{bmatrix}K_0(X_{j1},X_{j1}), & \cdots & K_0(X_{j1},X_{jn})\\ \vdots & \ddots & \vdots\\ K_0(X_{jn},X_{j1}) & \cdots & K_0(X_{jn},X_{jn})\end{bmatrix}.$$

$K_0(x,x')$ involves an infinite summation which is practically infeasible. Instead, the infinite sum is approximated by a finite one, i.e.,

$$K_0(x,x') \approx 2\sum_{k=1}^{M} \frac{\cos(2\pi k(x-x'))}{(2\pi k)^{2m+\beta} + n\lambda(2\pi k)^{2m}}.$$

In our numerical study, we found that $M = 100$ can already provide a good approximation. Due to the conjugacy, the posterior distribution of $(f(X_{j1}),\ldots,f(X_{jn}))^\top$ also follows a multivariate Gaussian distribution

$$(f(X_{j1}),\ldots,f(X_{jn}))^\top \Big| \{Y_{ji},X_{ji}\}_{i=1}^n \sim N\Big(K_j(K_j+\frac{1}{n}I_n)^{-1}(Y_{j1},\ldots,Y_{jn})^\top, K_j(K_j+\frac{1}{n}I_n)^{-1}\frac{1}{n}\Big).$$

Next we generate M independent samples, denoted $(f^{(l)}(X_{j1}),\ldots,f^{(l)}(X_{jn}))^\top, l = 1,\ldots,M$, from above multivariate Gaussian distribution. Therefore, the posterior mean can be approximated by

$$\Big(\breve{f}_{jn}(X_{j1}),\ldots,\breve{f}_{jn}(X_{jn})\Big)^\top = \Big(\frac{1}{M}\sum_{l=1}^{M} f^{(l)}(X_{j1}),\ldots,\frac{1}{M}\sum_{l=1}^{M} f^{(l)}(X_{jn})\Big)^\top.$$

Calculation of posterior radius. Once we have M independent samples $\{(f^{(l)}(X_{j1}),\ldots,f^{(l)}(X_{jn}))\}_{l=1}^M$, we are able to approximate $\|f^{(l)} - \breve{f}_{j,n}\|_{L^2}$ by

$$L_l = \Big(\frac{1}{n}\sum_{i=1}^{n}(f^{(l)}(X_{ji}) - \breve{f}_{jn}(X_{ji}))^2\Big)^{\frac{1}{2}}, \text{ for } l = 1,\ldots,M.$$

Finally, the radius $r_{j,n}(\alpha)$ is approximated by the upper α-th percentile of $\{L_1,\ldots,L_M\}$.

Calculation of the integral. We approximate (8) by

$$\breve{f}_{j,n,k} \approx \frac{\sqrt{2}}{n}\sum_{i=1}^{n}\breve{f}_{j,n}(X_{ji})\cos(2\pi k X_{ji})dx,\ \breve{g}_{j,n,k} \approx \frac{\sqrt{2}}{n}\sum_{i=1}^{n}\breve{f}_{j,n}(X_{ji})\sin(2\pi k X_{ji})dx.$$

In (12), C_k and D_k also involve two integrals. Since they are independent of samples, any numerical method for integral calculation is applicable. We also approximate $\breve{f}_{N,\lambda}(x)$ in (10) by

$$\breve{f}_{N,\lambda}(x) \approx \sum_{k=1}^{M} w_{s,N,\lambda,k}\left\{\breve{f}_{N,\lambda,k}\sqrt{2}\cos(2\pi kx) + \breve{g}_{N,\lambda,k}\sqrt{2}\sin(2\pi kx)\right\}.$$

References

George D Birkhoff. Boundary value and expansion problems of ordinary linear differential equations. *Transactions of the American Mathematical Society*, 9(4):373–395, 1908.

Willem van den Boom, Galen Reeves, and David B Dunson. Scalable approximations of marginal posteriors in variable selection. *arXiv preprint arXiv:1506.06629*, 2015.

Robert H Cameron and William T Martin. Transformations of weiner integrals under translations. *Annals of Mathematics*, pages 386–396, 1944.

Yanshuai Cao and David J Fleet. Generalized product of experts for automatic and principled fusion of gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

Ismaël Castillo, Richard Nickl, et al. Nonparametric bernstein–von mises theorems in gaussian white noise. *The Annals of Statistics*, 41(4):1999–2028, 2013.

Ismaël Castillo, Richard Nickl, et al. On the bernstein–von mises phenomenon for nonparametric bayes procedures. *The Annals of Statistics*, 42(5):1941–1969, 2014.

Yuan Shih Chow and Henry Teicher. *Probability theory: independence, interchangeability, martingales*. Springer Science & Business Media, 2012.

Peter de Jong. A central limit theorem for generalized quadratic forms. *Probability Theory and Related Fields*, 75(2):261–277, 1987.

Subhashis Ghosal, Jayanta K Ghosh, Aad W Van Der Vaart, et al. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531, 2000.

Jorgen Hoffmann-Jorgensen, Lawrence A Shepp, and Richard M Dudley. On the lower tail of gaussian seminorms. *The Annals of Probability*, pages 319–342, 1979.

Zaijing Huang and Andrew Gelman. Sampling for bayesian computation with large datasets. *Available at SSRN 1010107*, 2005.

Brian R Hunt, Tim Sauer, and James A Yorke. Prevalence: a translation-invariant almost every on infinite-dimensional spaces. *Bulletin of the American mathematical society*, 27 (2):217–238, 1992.

Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I Jordan. Bootstrapping big data. In *Advances in neural information processing systems, workshop: Big learning: Algorithms, systems, and tools for learning at scale*, 2011.

Michael R Kosorok. *Introduction to empirical processes and semiparametric inference*. Springer, 2008.

James Kuelbs, Wenbo V Li, and Werner Linde. The gaussian measure of shifted balls. *Probability Theory and Related Fields*, 98(2):143–162, 1994.

Cheng Li, Sanvesh Srivastava, and David B Dunson. Simple, scalable and accurate posterior interval estimation. *Biometrika*, 104(3):665–680, 2017.

Wenbo Li et al. A gaussian correlation inequality and its applications to small ball probabilities. *Electronic Communications in Probability*, 4:111–118, 1999.

Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 456–464. Association for Computational Linguistics, 2010.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David B Dunson. Robust and scalable bayes via a median of subset posterior measures. *The Journal of Machine Learning Research*, 18(1):4488–4527, 2017.

Carl N Morris et al. Natural exponential families with quadratic variance functions: statistical theory. *The Annals of Statistics*, 11(2):515–529, 1983.

Willie Neiswanger, Chong Wang, and Eric Xing. Asymptotically exact, embarrassingly parallel mcmc. *arXiv preprint arXiv:1311.4780*, 2013.

Iosif Pinelis et al. Optimum bounds for the distributions of martingales in banach spaces. *The Annals of Probability*, 22(4):1679–1706, 1994.

Vincent Rivoirard, Judith Rousseau, et al. Bernstein–von mises theorem for linear functionals of the density. *The Annals of Statistics*, 40(3):1489–1523, 2012.

Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

Steven L Scott, Alexander W Blocker, Fernando V Bonassi, Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016.

Hao Botao Shang, Zuofeng and Guang Cheng. Supplementary document to "nonparametric bayesian aggregation for massive data.".

Zuofeng Shang and Guang Cheng. Gaussian approximation of general non-parametric posterior distributions. *Information and Inference: A Journal of the IMA*, 7(3):509–529, 2017.

Zuofeng Shang, Guang Cheng, et al. Local and global asymptotic inference in smoothing spline models. *The Annals of Statistics*, 41(5):2608–2638, 2013.

Sanvesh Srivastava, Cheng Li, and David B Dunson. Scalable bayes via barycenter in wasserstein space. *The Journal of Machine Learning Research*, 19(1):312–346, 2018.

Botond Szabo and Harry van Zanten. Adaptive distributed methods under communication constraints. *arXiv preprint arXiv:1804.00864*, 2018.

Botond Szabó and Harry van Zanten. An asymptotic analysis of distributed nonparametric methods. *Journal of Machine Learning Research*, 20(87):1–30, 2019.

Sara Van De Geer and SA Van De Geer. *Empirical Processes in M-estimation*. Cambridge UP, 2006.

Aad W van der Vaart, J Harry van Zanten, et al. Reproducing kernel hilbert spaces of gaussian priors. In *Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh*, pages 200–222. Institute of Mathematical Statistics, 2008a.

Aad W van der Vaart, J Harry van Zanten, et al. Rates of contraction of posterior distributions based on gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463, 2008b.

Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

Xiangyu Wang and David B Dunson. Parallelizing mcmc via weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

Xiangyu Wang, Peichao Peng, and David B Dunson. Median selection subset aggregation for parallel inference. In *Advances in neural information processing systems*, pages 2195–2203, 2014.

Xiangyu Wang, Fangjian Guo, Katherine A Heller, and David B Dunson. Parallelizing mcmc with random partition trees. In *Advances in neural information processing systems*, pages 451–459, 2015.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015a.

Yuchen Zhang, Martin Wainwright, and Michael Jordan. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. In *International Conference on Machine Learning*, pages 457–465, 2015b.

Tianqi Zhao, Guang Cheng, and Han Liu. A partially linear framework for massive heterogeneous data. *Annals of statistics*, 44(4):1400, 2016.

Supplementary document to Nonparametric Bayesian Aggregation for Massive Data

This supplementary document is structured as follows.

- Section S.8.1 contains the proofs of Lemmas 2 and 3.
- Section S.8.2 contains the proofs of the main results in Section 4.5 and 4.6 that were not included in the main paper.
- Section S.8.3 proves Proposition 1, i.e., a uniform contraction rate result. Preliminary results relevant to the proof of Proposition 1 are provided in Section S.8.4.
- Section S.8.5 includes a result that characterizes the posterior tail moments of $\|f - f_0\|^a$ for any $a \geq 0$.
- Section S.8.6 includes additional simulation results supplementary to Section 5.

S.8.1. Proofs of Lemmas 2 and 3

Proof [Proof of Lemma 2] We only show the first limit distribution since the proof of the second one is similar.

Let $\eta_\nu = \tau_\nu v_\nu$. Then η_ν is a sequence of *iid* standard normals. Note that

$$\|W_n\|_2^2 = \sum_{\nu=1}^{\infty} \frac{\eta_\nu^2}{\tau_\nu^2 + n(1+\lambda\rho_\nu)}.$$

Let $U_n = (n\|W_n\|_2^2 - \zeta_{1,n})/\sqrt{2\zeta_{2,n}}$, then we have

$$U_n = \frac{1}{\sqrt{2\zeta_{2,n}}} \sum_{\nu=1}^{\infty} \frac{n(\eta_\nu^2 - 1)}{\tau_\nu^2 + n(1+\lambda\rho_\nu)}.$$

By straightforward calculations and Taylor's expansion of $\log(1-x)$, it can be shown that the logarithm of the moment generating function of U_n equals

$$\log E\{\exp(tU_n)\} = t^2/2 + O\left(t^3 \zeta_{2,n}^{-3/2} \zeta_{3,n}\right). \tag{S.1}$$

Without loss of generality, assume that $N = n^a$ for some $a \geq 1$. Then $\alpha_1 := \min\{1/(2m+\beta), a/(2m+\beta)\} = 1/(2m+\beta)$. It follows by (Shang and Cheng, 2017, Lemma S.1) that $\zeta_{2,n} \asymp n^{\alpha_1}$ and $\zeta_{3,n} \asymp n^{\alpha_1}$, so the remainder term in (S.1) is $O(n^{-\alpha_1/2}) = o(1)$. So $\lim_{n\to\infty} E\{\exp(tU_n)\} = \exp(t^2/2)$. Proof is completed. ■

Proof [Proof of Lemma 3] The proof follows by moment generating function approach and direct calculations. ■

S.8.2. Proofs in Sections 4.5 and 4.6

This section contains the proofs in Sections 4.5 and 4.6.

Proofs in Section 4.5

Proof [Proof of Theorem 5] Recall in the proof of Theorem 4 we showed that Rate Conditions ($\mathbf{R}'$) are satisfied.

It is easy to see that

$$F(W_n) \overset{d}{=} N(0, \theta_{1,n}^2), \quad \text{and} \quad F(W_N) \overset{d}{=} N(0, \theta_{1,N}^2). \tag{S.2}$$

For $1 \leq j \leq s$, define $R_{j,n}^F(\alpha) = \{f \in S^m(\mathbb{I}) : |F(f) - F(\breve{f}_{j,n})| \leq r_{F,j,n}(\alpha)\}$. It follows by Theorem 1 that $\max_{1\leq j\leq s} |1 - \alpha - P_{0j}(R_{j,n}^F(\alpha))| = o_{P_{f_0}}(1)$. Since $s = o(N^{\frac{4m^2+2m\beta-12m+1}{8m(2m+\beta)}} (\log N)^{-\frac{3}{2}})$, it can be examined that $NL_N^2 = o(1)$. Together with the condition $h^{-r} \lesssim N\theta_{1,N}^2$ and the fact $\theta_{k,N} \leq \theta_{k,n}$, one can verify that $h^{-r} \lesssim N\theta_{1,N}^2 \leq N\theta_{1,n}^2 = o(L_N^{-2}\theta_{1,n}^2)$. So we have by (35) and Theorem 2 that

$$\max_{1\leq j\leq s} |F(\Delta_j)| = O_{P_{f_0}}(h^{-r/2} L_N) = o_{P_{f_0}}(\theta_{1,n}).$$

Combined with (S.2) we get that

$$\begin{aligned} P_{0j}(R_{j,n}^F(\alpha)) &= P(|F(W_n) - F(\Delta_j)| \leq r_{F,j,n}(\alpha) | \mathbf{D}_j) \\ &= \Phi\left(\frac{r_{F,j,n}(\alpha) + F(\Delta_j)}{\theta_{1,n}}\right) + \Phi\left(\frac{r_{F,j,n}(\alpha) - F(\Delta_j)}{\theta_{1,n}}\right) - 1 \\ &= 2\Phi\left(\frac{r_{F,j,n}(\alpha)}{\theta_{1,n}}\right) - 1 + o_{P_{f_0}}(1), \text{ uniformly for } 1 \leq j \leq s. \end{aligned}$$

The above argument leads to $\Phi(r_{F,j,n}(\alpha)/\theta_{1,n}) = 1 - \alpha/2 + o_{P_{f_0}}(1)$ uniformly for $1 \leq j \leq s$, which further leads to the following

$$\max_{1\leq j\leq s} |r_{F,j,n}(\alpha)/\theta_{1,n} - z_{\alpha/2}| = o_{P_{f_0}}(1). \tag{S.3}$$

Consider the decomposition (12) with T_1, T_2, T_3, T_4 being defined therein. It follows by (13) and rate condition $Na_n^2 = o(1)$ that $N\|T_1\|^2 = O_{P_{f_0}}(Na_n^2) = o_{P_{f_0}}(1)$. Meanwhile, it follows by Condition ($\mathbf{S}'$), $N^{-1} \asymp h^{2m+\beta}$ and $\lambda = h^{2m}$ and direct examinations that

$$\begin{aligned} N\|T_2\|^2 &= N\sum_{\nu=1}^{\infty} (a_{N,\nu} - 1)^2 |f_\nu^0|^2 (1+\lambda\rho_\nu) \\ &\asymp N\sum_{\nu=1}^{\infty} \left(\frac{\nu^{2m+\beta}}{\nu^{2m+\beta} + N(1+\lambda\nu^{2m})}\right)^2 |f_\nu^0|^2 (1+\lambda\nu^{2m}) \\ &\asymp \sum_{\nu=1}^{\infty} \frac{(h\nu)^{2m+\beta} + (h\nu)^{4m+\beta}}{(1+(h\nu)^{2m} + (h\nu)^{2m+\beta})^2} \times |f_\nu^0|^2 \nu^{2m+\beta} = o(1), \end{aligned}$$

and

$$\begin{aligned} N\|T_4\|^2 &= N\sum_{\nu=1}^{\infty} a_{N,\nu}^2 \left(\frac{\lambda\rho_\nu}{1+\lambda\rho_\nu}\right)^2 |f_\nu^0|^2 (1+\lambda\rho_\nu) \\ &\asymp \sum_{\nu=1}^{\infty} \frac{(h\nu)^{2m-\beta}}{1+(h\nu)^{2m}} \times |f_\nu^0|^2 \nu^{2m+\beta} = o(1). \end{aligned}$$

By (11) and $Ns^2L_N^2 = o(1)$ we get $\|\breve{f}_{N,\lambda} - \widetilde{f}_{N,\lambda}\| = o_{P_{f_0}}(N^{-1/2})$. Therefore, $\|\breve{f}_{N,\lambda} - f_0 - T_3\| \le \|\breve{f}_{N,\lambda} - \widetilde{f}_{N,\lambda}\| + \|T_1 + T_2 + T_4\| = o_{P_{f_0}}(N^{-1/2})$. If follows from (35) that $|F(\breve{f}_{N,\lambda} - f_0) - F(T_3)| = o_{P_{f_0}}(h^{-r/2}N^{-1/2})$.

Note that $F(T_3) = \frac{1}{N}\sum_{i=1}^N \epsilon_i F(R_{X_i})$, where the kernel R is defined in the proof of Theorem 3. We will derive asymptotic distribution for $F(T_3)$. Let $s_N^2 = Var_{f_0}(\sum_{i=1}^N \epsilon_i F(R_{X_i}))$. It is easy to show that

$$s_N^2 = N^3 \sum_{\nu=1}^{\infty} \frac{F(\varphi_\nu)^2}{(\tau_\nu^2 + N(1+\lambda\rho_\nu))^2} = N^3\theta_{2,N}^2.$$

Clearly, by uniform boundedness of φ_ν and $F(\varphi_\nu)$, we get

$$|F(R_x)| = |\sum_{\nu=1}^{\infty} a_{N,\nu} \frac{\varphi_\nu(x)F(\varphi_\nu)}{1+\lambda\rho_\nu}| \lesssim h^{-1},$$

where the "$\lesssim$" is free of $x \in \mathbb{I}$, and

$$E_{f_0}\{\epsilon^2 F(R_X)^2\} = N^2 \sum_{\nu=1}^{\infty} \frac{F(\varphi_\nu)^2}{(\tau_\nu^2 + N(1+\lambda\rho_\nu))^2} = N^2\theta_{2,N}^2. \tag{S.4}$$

Then for any $\delta > 0$, by condition $E_{f_0}\{\epsilon^4|X\} \le M_4$ a.s.,

$$\begin{aligned} & \frac{1}{s_N^2}\sum_{i=1}^N E_{f_0}\{\epsilon_i^2 F(R_{X_i})^2 I(|\epsilon_i F(R_{X_i})| \ge \delta s_N)\} \\ \le\ & \frac{N}{s_N^2}(\delta s_N)^{-2} E_{f_0}\{\epsilon^4 F(R_X)^4\} \\ \lesssim\ & \frac{N}{s_N^2}(\delta s_N)^{-2} h^{-2} E_{f_0}\{\epsilon^2 F(R_X)^2\} \lesssim \delta^{-2}N^{-1}h^{-2+r} = o(1), \end{aligned}$$

where the last $o(1)$-term follows by $h \asymp h^*$ and $2 - r < 2m + \beta$. By Lindeberg's central limit theorem, as $N \to \infty$,

$$\frac{F(T_3)}{\sqrt{N}\theta_{2,N}} = \frac{1}{s_N}\sum_{i=1}^N \epsilon_i F(R_{X_i}) \xrightarrow{d} N(0,1). \tag{S.5}$$

By condition $N^2\theta_{2,N}^2 \gtrsim h^{-r}$, we have

$$\left|\frac{F(\breve{f}_{N,\lambda} - f_0 - T_3)}{\sqrt{N}\theta_{2,N}}\right| = o_{P_{f_0}}\left(\frac{h^{-r/2}N^{-1/2}}{\sqrt{N}\theta_{2,N}}\right) = o_{P_{f_0}}(1).$$

It follows by (S.3) that

$$r_{F,N}(\alpha) = \theta_{1,N}\sqrt{\frac{1}{s}\sum_{j=1}^s r_{F,j,n}(\alpha)^2/\theta_{1,n}^2} = \theta_{1,N} z_{\alpha/2}(1 + o_{P_{f_0}}(1)), \tag{S.6}$$

leading to that

$$\frac{r_{F,N}(\alpha)}{\sqrt{N}\theta_{2,N}} = \frac{\theta_{1,N}}{\sqrt{N}\theta_{2,N}} \times z_{\alpha/2}(1 + o_{P_{f_0}}(1)).$$

It can be shown that

$$\frac{\theta_{1,N}^2}{N\theta_{2,N}^2} = \frac{\sum_{\nu=1}^{\infty}\frac{F(\varphi_\nu)^2}{1+\lambda\rho_\nu+N^{-1}\tau_\nu^2}}{\sum_{\nu=1}^{\infty}\frac{F(\varphi_\nu)^2}{(1+\lambda\rho_\nu+N^{-1}\tau_\nu^2)^2}} \ge 1,$$

together with (S.5) we get that

$$\begin{aligned} & P_{f_0}(|F(f_0) - F(\breve{f}_{N,\lambda})| \le r_{F,N}(\alpha)) \\ =\ & P_{f_0}\left(\left|\frac{F(\breve{f}_{N,\lambda} - f_0 - T_3)}{\sqrt{N}\theta_{2,N}} + \frac{F(T_3)}{\sqrt{N}\theta_{2,N}}\right| \le \frac{r_{F,N}(\alpha)}{\sqrt{N}\theta_{2,N}}\right) \\ \ge\ & P_{f_0}\left(\left|\frac{F(\breve{f}_{N,\lambda} - f_0 - T_3)}{\sqrt{N}\theta_{2,N}} + \frac{F(T_3)}{\sqrt{N}\theta_{2,N}}\right| \le z_{\alpha/2}(1 + o_{P_{f_0}}(1))\right) \\ \to\ & 1 - \alpha. \end{aligned} \tag{S.7}$$

Notice that when $0 < \sum_{\nu=1}^{\infty} F(\varphi_\nu)^2 < \infty$, $\frac{\theta_{1,N}^2}{N\theta_{2,N}^2} \to 1$, leading to that the probability in (S.7) approaches exactly $1 - \alpha$.

In the end, we show that $P(R_N^F(\alpha)|\mathbf{D}) = 1 - \alpha + o_{P_{f_0}}(1)$, where $R_N^F(\alpha) = \{f \in S^m(\mathbb{I}) : |F(f) - F(\breve{f}_{N,\lambda})| \le r_{F,N}(\alpha)\}$. By rate condition $N(a_N^2 + a_n^2) = o(1)$, proof of (21) leading to $\|\widetilde{f}_{N,\lambda}^{or} - \widetilde{f}_{N,\lambda}\| = O_{P_{f_0}}(a_N + a_n)$, and (35) we have

$$\frac{F(\widetilde{f}_{N,\lambda}^{or} - \widetilde{f}_{N,\lambda})}{\theta_{1,N}} = O_{P_{f_0}}\left(\frac{h^{-r/2}(a_N + a_n)}{\theta_{1,N}}\right) = o_{P_{f_0}}(1),$$

where the last $o(1)$-term follows by condition $N\theta_{1,N}^2 \gtrsim h^{-r}$ and Rate Condition ($\mathbf{R}'$). From (S.6) we get that

$$\begin{aligned} P_0(R_N^F(\alpha)) &= P(W^{or} \in R_N^F(\alpha)|\mathbf{D}) \\ &= P(|F(W^{or}) - F(\breve{f}_{N,\lambda})| \le r_{F,N}(\alpha)|\mathbf{D}) \\ &= P\left(\left|\frac{F(\widetilde{f}_{N,\lambda}^{or} - \widetilde{f}_{N,\lambda})}{\theta_{1,N}} + \frac{F(W_N)}{\theta_{1,N}}\right| \le \frac{r_{F,N}(\alpha)}{\theta_{1,N}}\middle|\mathbf{D}\right) \\ &= 1 - \alpha + o_{P_{f_0}}(1). \end{aligned} \tag{S.8}$$

So it follows from (14) that $P(R_N^F(\alpha)|\mathbf{D}) = 1 - \alpha + o_{P_{f_0}}(1)$. Proof is completed. ■

Proofs in Section 4.6

Proof [Proof of Theorem 6] It follows from (20) that $r_N(\alpha) - r_N^\dagger(\alpha) = o_{P_{f_0}}(N^{-1}h^{-1/2})$, which together with (19) leads to that $\lim_{n\to\infty} P_{f_0}(f_0 \in R_N^\dagger(\alpha)) = 1$. It follow from Lemma 2, (14) and the proof of Theorem 3 that $P(R_N^\dagger(\alpha)|\mathbf{D}) = 1 - \alpha + o_{P_{f_0}}(1)$.

It follows from (23) that $r_{\omega,N}(\alpha)^2 - r_{\omega,N}^\dagger(\alpha)^2 = o_{P_{f_0}}(N^{-1})$. Then the desired results on $R_N^{\dagger\omega}(\alpha)$ directly follow from the proof of Theorem 4.

It follows by (S.6) that $r^{\dagger}_{F,N}(\alpha) = r_{F,N}(\alpha)(1 + o_{P_{f_0}}(1))$. Then the desired results on $CI^{\dagger F}_N(\alpha)$ follow from (S.7) and (S.8). ■

S.8.3. Proofs of Proposition 1 and relevant results

The goal of this section is to prove Proposition 1 and relevant results. Before proofs, we exactly describe the Fréchet derivatives of the likelihood function that will be technically useful. Suppose that (Y, X) follows model (14) based on f. Let $g, g_k \in S^m(\mathbb{I})$ for $k = 1, 2$. For $j = 1, 2, \ldots, s$, the Fréchet derivative of ℓ_{jn} can be identified as

$$D\ell_{jn}(g)g_1 = \frac{1}{n}\sum_{i\in I_j}(Y_i - g(X_i))\langle K_{X_i}, g_1\rangle - \langle \mathcal{P}_\lambda g, g_1\rangle := \langle S_{j,n}(g), g_1\rangle.$$

Define $S_\lambda(g) = E\{S_{j,n}(g)\}$. We also use DS_λ and D^2S_λ to represent the second- and third-order Fréchet derivatives of S_λ. Note that $S_{j,n}(\widehat{f}_{j,n}) = 0$, and $S_{j,n}(f)$ can be expressed as

$$S_{j,n}(f) = \frac{1}{n}\sum_{i\in I_j}(Y_i - f(X_i))K_{X_i} - \mathcal{P}_\lambda f. \tag{S.9}$$

The Fréchet derivative of $S_{j,n}$ is denoted $DS_{j,n}(g)g_1g_2$. These derivatives can be explicitly written as

$$D^2\ell_{jn}(g)g_1g_2 := DS_{j,n}(g)g_1g_2 = -\frac{1}{n}\sum_{i\in I_j} g_1(X_i)g_2(X_i) - \langle \mathcal{P}_\lambda g_1, g_2\rangle,$$

The proof of Proposition 1 requires a series of preliminary lemmas. Define $H^m(b) = \{f \in S^m(\mathbb{I}) : J(f) \le b^2\}$. We first state a basic lemma about a concentration phenomenon of smoothing spline estimates in the distributed setup.

Lemma 4 *If b, r, h, M are positives satisfying the following Rate Condition (**H**):*

1. $h^{1/2}r \le 1$,
2. $c_K^2 M^{1/2} r h^{-1/2} B(h) \le 1/2$, *where $B(h) = A(h, 2)$ with $A(h, \varepsilon)$ given in (S.19),*

then, for any $1 \le j \le s$, the following two results hold:

1. $\sup_{f\in H^m(b)} P_f\left(\|\widehat{f}_{j,n} - f\| \ge \delta_n\right) \le 2\exp(-Mnhr^2)$, *where $\delta_n = bh^m + 2c_K(C_\epsilon + M)r$ with $C_\epsilon = E\{(|\epsilon| + 1)^2\exp(|\epsilon| + 1)\}$ an absolute constant;*
2. $\sup_{f\in H^m(b)} P_f\left(\|\widehat{f}_{j,n} - f - S_{j,n}(f)\| > a_n\right) \le 2\exp(-Mnhr^2)$, *where $a_n = c_K^2 M^{1/2} h^{-1/2} r B(h)\delta_n$. Here, $S_{j,n}(f)$ is the Fréchet derivative of the likelihood function $\ell_{jn}(f)$; see (S.9) for its exact expression.*

Lemma 5 *For any fixed constants $M > 1$ and $b > 0$, let*

$$r = (nh/\log 2s)^{-1/2}, \delta_n = bh^m + 2c_K(C_\epsilon + M)r, \tag{S.10}$$

$$a_n = c_K^2 M^{1/2} h^{-1/2} r B(h)\delta_n. \tag{S.11}$$

Then as $n \to \infty$,

$$P_{f_0}\left(\max_{1\le j\le s}\|\widehat{f}_{j,n} - f_0\| \ge \delta_n\right) \le 6sN^{-M} \to 0,$$

and

$$P_{f_0}\left(\max_{1\le j\le s}\|\widehat{f}_{j,n} - f_0 - S_{j,n}(f_0)\| > a_n\right) \le 8sN^{-M} \to 0.$$

Proof [Proof of Lemma 5] The result is a straightforward consequence of Lemma 4. ■

Lemma 6 *It holds that*

$$\max_{1\le j\le s}\|\widehat{f}_{j,n} - f_0 - S_{j,n}(f_0)\| = O_{P_{f_0}}(a_n). \tag{S.12}$$

Proof [Proof of Lemma 6] The proof follows by Lemma 5, and simple fact that $B(h) \lesssim h^{-\frac{2m-1}{4m}}$. ■

Lemma 7 *Under Condition (**S**), we get $\max_{1\le j\le s}\|\widehat{f}_{j,n} - f_0\| = O_{P_{f_0}}(\widetilde{r}_n)$.*

Proof [Proof of Lemma 7] Recall that

$$S_{j,n}(f_0) = -\frac{1}{n}\sum_{i\in I_j}(Y_i - f_0(X_i))K_{X_i} - \mathcal{P}_\lambda f_0.$$

It was shown by Shang et al. (2013) that $\mathcal{P}_\lambda\varphi_\nu = \frac{\lambda\varphi_\nu}{1+\lambda\varphi_\nu}\varphi_\nu$. Since f_0 satisfies Condition (**S**),

$$\begin{aligned}
\|\mathcal{P}_\lambda f_0\|^2 &= \langle\sum_{\nu=1}^{\infty} f_\nu^0\frac{\lambda\rho_\nu}{1+\lambda\rho_\nu}\varphi_\nu, \sum_{\nu=1}^{\infty} f_\nu^0\frac{\lambda\rho_\nu}{1+\lambda\rho_\nu}\varphi_\nu\rangle \\
&= \sum_{\nu=1}^{\infty}|f_\nu^0|^2\frac{\lambda^2\rho_\nu^2}{1+\lambda\rho_\nu} \\
&= \lambda^{1+\frac{\beta-1}{2m}}\sum_{\nu=1}^{\infty}|f_\nu^0|^2\rho_\nu^{1+\frac{\beta-1}{2m}}\frac{(\lambda\rho_\nu)^{1-\frac{\beta-1}{2m}}}{1+\lambda\rho_\nu} = O(h^{2m+\beta-1}),
\end{aligned}$$

where the last equation follows by $\lambda = h^{2m}$, $\sup_{x\ge 0}\frac{x^{1-\frac{\beta-1}{2m}}}{1+x} < \infty$, and Condition (**S**). On the other side, it follows by the proof of (S.22) that

$$\begin{aligned}
&P_{f_0}\left(\max_{1\le j\le s}\|\sum_{i\in I_j}(Y_i - f_0(X_i))K_{X_i}\| \ge L(M)n(nh/\log 2s)^{-1/2}\right) \\
\le\ & 2s\exp\left(-Mnh(nh/\log 2s)^{-1}\right) = (2s)^{1-M} \to 0, \text{ as } M \to \infty,
\end{aligned}$$

where $L(M) := c_K(C_\epsilon + M)$. This implies that

$$\max_{1\le j\le s}\|\sum_{i\in I_j}(Y_i - f_0(X_i))K_{X_i}\| = O_{P_{f_0}}(n(nh/\log 2s)^{-1/2}),$$

and hence,

$$\max_{1\le j\le s}\|S_{j,n}(f_0)\| = O_{P_{f_0}}((nh/\log 2s)^{-1/2} + h^{m+\frac{\beta-1}{2}}) = O_{P_{f_0}}(\widetilde{r}_n).$$

Together with (S.12) of Lemma 6 and the rate condition $a_n \lesssim \widetilde{r}_n$, we get that $\max_{1\le j\le s}\|\widehat{f}_{j,n} - f_0\| = O_{P_{f_0}}(\widetilde{r}_n)$. ■

Consider a function class

$$\mathcal{G} = \{g \in S^m(\mathbb{I}) : \|g\|_\infty \le 1, J(g,g) \le c_K^{-2} h^{-2m+1}\}. \tag{S.13}$$

Lemma 8 *For any fixed constant $M > 1$, as $n \to \infty$,*

$$P_{f_0}\left(\max_{1\le j\le s}\sup_{g\in\mathcal{G}}\|Z_{j,n}(g)\| \le B(h)\sqrt{M\log N}\right) \to 1,$$

where $Z_{j,n}(g) = \frac{1}{\sqrt{n}}\sum_{i\in I_j}[\psi_{j,n}(Z_i;g)K_{X_i} - E\{\psi_{j,n}(Z_i;g)K_{X_i}\}]$, $\psi_{j,n}(Z_i;g) = c_K^{-1}h^{1/2}g(X_i)$.

Proof [Proof of Lemma 8] It is easy to see that $\psi_{j,n}(Z_i;g)$ satisfies the Lipschitz continuity condition (S.20). Then the result directly follows by Lemma 12 (see appendix). ■

Lemma 9 *For $j = 1,\dots,s$,*

1. *$\ell_{jn}(f) - \ell_{jn}(\widehat{f}_{j,n}) = I_{j,n}(f)$, where $I_{j,n}(f) = \int_0^1\int_0^1 sDS_{j,n}(\widehat{f}_{j,n} + ss'(f - \widehat{f}_{j,n}))(f - \widehat{f}_{j,n})(f - \widehat{f}_{j,n})dsds'$ for any $f \in S^m(\mathbb{I})$;*
2. *$I_{j,n}(f) = T_j(f) - \frac{1}{2}\|f - \widehat{f}_{j,n}\|^2$, where recall that (see 5)*

$$T_j(f) = -\frac{1}{2n}\sum_{i\in I_j}[(f-\widehat{f}_{j,n})(X_i)^2 - E_X\{(f-\widehat{f}_{j,n})(X)^2\}]. \tag{S.14}$$

Proof [Proof of Lemma 9] Let $\Delta f = f - \widehat{f}_{j,n}$. Therefore,

$$\begin{aligned} I_{j,n}(f) &= -\frac{1}{n}\int_0^1\int_0^1 s\sum_{i\in I_j}(\Delta f)(X_i)^2 dsds' - \lambda J(\Delta f,\Delta f)/2 \\ &= -\frac{1}{2n}\sum_{i\in I_j}[(\Delta f)(X_i)^2 - E_X\{(\Delta f)(X)^2\}] - \frac{1}{2}\|\Delta f\|^2 \\ &= T_j(f) - \frac{1}{2}\|\Delta f\|^2. \end{aligned}$$

By Taylor's expansion in terms of Fréchet derivatives, $\ell_{jn}(f) - \ell_{jn}(\widehat{f}_{j,n}) = S_{j,n}(\widehat{f}_{j,n})(f - \widehat{f}_{j,n}) + I_{j,n}(f) = I_{j,n}(f)$. ■

Lemma 10 *There exists a universal constant $c_3 > 0$ s.t.*

$$\Pi(\|f - f_0\| \le \widetilde{r}_n) \ge \exp(-c_3\widetilde{r}_n^{-\frac{2}{2m+\beta-1}}),$$

where recall that Π is the probability measure induced by G.

Proof [Proof of Lemma 10] Note that $\lambda \le \widetilde{r}_n^{\frac{4m}{2m+\beta-1}}$. Then it follows by Lemma 13 (with d_n therein replaced by $\widetilde{r}_n$) and the proof of Theorem 7 that

$$\begin{aligned} \Pi(\|f-f_0\|\le\widetilde{r}_n) &= P(\|G-f_0\|\le\widetilde{r}_n) \\ &\ge P(V(G-f_0)\le\widetilde{r}_n^2/2, \lambda J(G-f_0)\le\widetilde{r}_n^2/2) \\ &\ge P(V(G-f_0)\le\widetilde{r}_n^2/2, J(G-f_0)\le\widetilde{r}_n^{\frac{2(\beta-1)}{2m+\beta-1}}/2) \\ &= P(\widetilde{V}(\widetilde{G}-\widetilde{f}_0)\le\widetilde{r}_n^2/2, \widetilde{J}(\widetilde{G}-\widetilde{f}_0)\le\widetilde{r}_n^{\frac{2(\beta-1)}{2m+\beta-1}}/2) \\ &\ge P(\widetilde{V}(\widetilde{G}-\omega)\le(1/\sqrt{2}-1/2)^2\widetilde{r}_n^2, \widetilde{J}(\widetilde{G}-\omega)\le(1/\sqrt{2}-1/2)^2\widetilde{r}_n^{\frac{2(\beta-1)}{2m+\beta-1}}) \\ &\ge \exp(-\|\omega\|_\beta^2/2) \\ &\quad\times P(\widetilde{V}(\widetilde{G})\le(1/\sqrt{2}-1/2)^2\widetilde{r}_n^2, \widetilde{J}(\widetilde{G})\le(1/\sqrt{2}-1/2)^2\widetilde{r}_n^{\frac{2(\beta-1)}{2m+\beta-1}}) \\ &\ge \exp(-\|\omega\|_\beta^2/2)P(\widetilde{V}(\widetilde{G})\le(1/\sqrt{2}-1/2)^2\widetilde{r}_n^2/2) \\ &\quad\times P(\widetilde{J}(\widetilde{G})\le(1/\sqrt{2}-1/2)^2\widetilde{r}_n^{\frac{2(\beta-1)}{2m+\beta-1}}/2) \\ &\ge \exp(-c_3\widetilde{r}_n^{-\frac{2}{2m+\beta-1}}), \end{aligned}$$

where $c_3 > 0$ is a universal constant. ■

Proof [Proof of Proposition 1] Fix any $\varepsilon \in (0,1)$. Let M_1 be a large constant so that (thanks to Lemma 7) the event

$$\mathcal{E}_n' = \{\max_{1\le j\le s}\|\widehat{f}_{j,n} - f_0\| \le M_1\widetilde{r}_n\} \tag{S.15}$$

has probability approaching one. Meanwhile, for a fixed constant $M > 1$, define

$$\mathcal{E}_n'' = \left\{\max_{1\le j\le s}\sup_{g\in\mathcal{G}}\|Z_{j,n}(g)\| \le B(h)\sqrt{M\log N}\right\}. \tag{S.16}$$

By Lemma 8 we have that $\mathcal{E}_n''$ has P_{f_0}-probability approaching one. Thus, it holds that, when n becomes large, $P_{f_0}(\mathcal{E}_n) \ge 1 - \varepsilon/2$, where $\mathcal{E}_n := \mathcal{E}_n' \cap \mathcal{E}_n''$. In the rest of the proof we simply assume that $\mathcal{E}_n$ holds.

For some positive constant M_0, it follows by Theorem 7 that

$$\max_{1\le j\le s} E\{\|f-f_0\|^a I(\|f-f_0\|\ge M_0 r_n)|\mathbf{D}_j\} = O_{P_{f_0}}(s^2\exp(-nr_n^2)).$$

Let $C' > M_1$ be a constant to be further determined later, then we have that

$$\begin{aligned} &\max_{1\le j\le s} E\{\|f-f_0\|^a I(\|f-f_0\|\ge 2C'\widetilde{r}_n)|\mathbf{D}_j\} \\ \le\ &\max_{1\le j\le s} E\{\|f-f_0\|^a I(\|f-f_0\|\ge M_0 r_n)|\mathbf{D}_j\} \\ &+\max_{1\le j\le s} E\{\|f-f_0\|^a I(2C'\widetilde{r}_n\le\|f-f_0\|\le M_0 r_n)|\mathbf{D}_j\}. \end{aligned}$$

The first term is $O_{P_{f_0}}(s^2\exp(-nr_n^2))$. Thus, when n is sufficiently large,

$$P_{f_0}\left(\max_{1\le j\le s} E\{\|f-f_0\|^a I(\|f-f_0\|\ge M_0 r_n)|\mathbf{D}_j\}\ge M's^2\exp(-nr_n^2)/2\right)\le \varepsilon/2$$

for a large constant $M'>0$.

Next we only need to handle the second term. Let $\Delta f = f-\widehat{f}_{j,n}$. It follows by Lemma 9 that $I_{j,n}(f)=T_j(f)-\frac{1}{2}\|\Delta f\|^2$, and $\ell_{jn}(f)-\ell_{jn}(\widehat{f}_{j,n})=I_{j,n}(f)$. Therefore,

$$\begin{aligned}&E\{\|f-f_0\|^a I(f\in A_n)|\mathbf{D}_j\}\\ =\ &\frac{\int_{A_n}\|f-f_0\|^a\exp(n(\ell_{jn}(f)-\ell_{jn}(\widehat{f}_{j,n})))d\Pi(f)}{\int_{S^m(\mathbb{I})}\exp(n(\ell_{jn}(f)-\ell_{jn}(\widehat{f}_{j,n})))d\Pi(f)}=\frac{\int_{A_n}\|f-f_0\|^a\exp(nI_{j,n}(f))d\Pi(f)}{\int_{S^m(\mathbb{I})}\exp(nI_{j,n}(f))d\Pi(f)},\end{aligned}$$

where $A_n=\{f\in S^m(\mathbb{I}): 2C'\widetilde{r}_n\le\|f-f_0\|\le M_0r_n\}$.

Let

$$J_{j1}=\int_{S^m(\mathbb{I})}\exp(nI_{j,n}(f))d\Pi(f),\ J_{j2}=\int_{A_n}\|f-f_0\|^a\exp(nI_{j,n}(f))d\Pi(f).$$

Then on $\mathcal{E}_n$ and for $\|f-f_0\|\le\widetilde{r}_n$, we have $\|f-\widehat{f}_{j,n}\|\le\|f-f_0\|+\|\widehat{f}_{j,n}-f_0\|\le(M_1+1)\widetilde{r}_n$.

Let $d_n=c_K(M_1+1)h^{-1/2}\widetilde{r}_n$. It follows by similar arguments as above (S.23) that $d_n^{-1}\Delta f\in\mathcal{G}$. Note that on $\mathcal{E}_n$ and for $\|f-f_0\|\le\widetilde{r}_n$, for all $1\le j\le s$,

$$\begin{aligned}|T_j(f)| &= \frac{1}{2n}\left|\sum_{i\in I_j}[(\Delta f)(X_i)^2-E_X\{(\Delta f)(X)^2\}]\right|\\ &= \frac{1}{2n}\left|\langle\sum_{i\in I_j}[(\Delta f)(X_i)K_{X_i}-E_X\{(\Delta f)(X)K_X\}],\Delta f\rangle\right|\\ &\le \frac{1}{2n}\|\Delta f\|\times\|\sum_{i\in I_j}[(\Delta f)(X_i)K_{X_i}-E_X\{(\Delta f)(X)K_X\}]\|\\ &= \frac{c_Kh^{-1/2}d_n\|\Delta f\|}{2\sqrt{n}}\times\|Z_{j,n}(d_n^{-1}\Delta f)\|\\ &\le \frac{c_Kh^{-1/2}d_n\|\Delta f\|}{2\sqrt{n}}B(h)\sqrt{M\log N}\\ &\le D(c_K,M,M_1)\times n^{-1/2}h^{-\frac{6m-1}{4m}}\widetilde{r}_n^2\sqrt{\log N}\le D(c_K,M,M_1)\times\widetilde{r}_n^2b_n,\end{aligned}\tag{S.17}$$

where $D(c_K,M,M_1)$ is constant depending only on c_K,M_1,M.

It follows that on $\mathcal{E}_n$ and for all $1\le j\le s$,

$$\begin{aligned}J_{j1} &\ge \int_{\|f-f_0\|\le\widetilde{r}_n}\exp(nI_{j,n}(f))d\Pi(f)\\ &= \int_{\|f-f_0\|\le\widetilde{r}_n}\exp\left(nT_j(f)-\frac{n}{2}\|f-\widehat{f}_{j,n}\|^2\right)d\Pi(f)\\ &\ge \exp\left(-[D(c_K,M,M_1)b_n+(M_1+1)^2/2]n\widetilde{r}_n^2\right)\Pi(\|f-f_0\|\le\widetilde{r}_n).\end{aligned}$$

Since $\Pi(\|f-f_0\|\le\widetilde{r}_n)\ge\exp(-c_3\widetilde{r}_n^{-\frac{2}{2m+\beta-1}})$ (Lemma 10), together with $\widetilde{r}_n\ge(nh)^{-1/2}+h^{m+\frac{\beta-1}{2}}\ge 2n^{-\frac{2m+\beta-1}{2(2m+\beta)}}$, we get that $n\widetilde{r}_n^{2+\frac{2}{2m+\beta-1}}\ge n(4n^{-\frac{2m+\beta-1}{2m+\beta}})^{1+\frac{1}{2m+\beta-1}}=4$. Therefore, $\widetilde{r}_n^{-\frac{2}{2m+\beta-1}}\le n\widetilde{r}_n^2/4$, leading to

$$\Pi(\|f-f_0\|\le\widetilde{r}_n)\ge\exp\left(-\frac{c_3}{4}n\widetilde{r}_n^2\right).\tag{S.18}$$

This implies by rate conditions $b_n\le 1$ that, on $\mathcal{E}_n$ and for any $1\le j\le s$,

$$\begin{aligned}J_{j1} &\ge \exp\left(-[D(c_K,M,M_1)b_n+(M_1+1)^2/2+c_3/4]n\widetilde{r}_n^2\right)\\ &\ge \exp\left(-[D(c_K,M,M_1)+(M_1+1)^2/2+c_3/4]n\widetilde{r}_n^2\right).\end{aligned}$$

Next we handle J_{j2}. The idea is similar to how we handle J_{j1} but with technical difference. Let $\Delta f=f-\widehat{f}_{j,n}$. Note that $\widetilde{r}_n^2\le r_n^2\log(2s)$, and hence, on $\mathcal{E}_n$, for any $f\in A_n$, i.e., $\|f-f_0\|\le M_0r_n$, we get that $\|\Delta f\|=\|\widehat{f}_{j,n}-f\|\le\|\widehat{f}_{j,n}-f_0\|+\|f-f_0\|\le M_1\widetilde{r}_n+M_0r_n\le(M_0+M_1)r_n\sqrt{\log(2s)}$. Let $d_{*n}=c_K(M_0+M_1)h^{-1/2}r_n\sqrt{\log(2s)}$. Then $d_{*n}^{-1}\Delta f\in\mathcal{G}$. Using previous similar arguments handling $T_j(f)$, we have that on $\mathcal{E}_n$, for any $f\in A_n$ and $1\le j\le s$,

$$\begin{aligned}|T_j(f)| &\le \frac{\|\Delta f\|}{2\sqrt{n}}c_Kh^{-1/2}d_{*n}\cdot B(h)\sqrt{M\log N}\\ &\le \frac{1}{2}c_K^2(M_0+M_1)^2M^{1/2}n^{-1/2}h^{-1}r_n^2B(h)(\log N)^{3/2}\\ &\le D(c_K,M,M_0,M_1)\times n^{-1/2}r_n^2h^{-\frac{6m-1}{4m}}(\log N)^{3/2}\\ &= D(c_K,M,M_0,M_1)\times r_n^2b_n\le D(c_K,M,M_0,M_1)\times\widetilde{r}_n^2,\end{aligned}$$

where $D(c_K,M,M_0,M_1)$ is constant only depending on c_K,M,M_0,M_1 and the last inequality follows by rate condition $r_n^2b_n\le\widetilde{r}_n^2$. It is easy to see that on $\mathcal{E}_n$ and for any $f\in A_n$ and $1\le j\le s$, $\|\widehat{f}_{j,n}-f\|\ge\|f-f_0\|-\|\widehat{f}_{j,n}-f_0\|\ge(2C'-M_1)\widetilde{r}_n$, leading to that

$$J_{j2}\le\exp\left(-\left(\frac{(2C'-M_1)^2}{2}-D(c_K,M,M_0,M_1)\right)n\widetilde{r}_n^2\right)C(a,\Pi),$$

where $C(a,\Pi)=\int_{S^m(\mathbb{I})}\|f-f_0\|^ad\Pi(f)$ is the ath prior moment of $\|f-f_0\|$ which is finite. Choose $C'>M_1$ to be large such that

$$\frac{(2C'-M_1)^2}{2}\ge 1+D(c_K,M,M_1)+D(c_K,M,M_0,M_1)+(M_1+1)^2/2+c_3/4.$$

Therefore, on $\mathcal{E}_n$,

$$\max_{1\le j\le s}E\{\|f-f_0\|^aI(f\in A_n)|\mathbf{D}_j\}\le\frac{\max_{1\le j\le s}J_{j2}}{\min_{1\le j\le s}J_{j1}}\le\exp(-n\widetilde{r}_n^2)C(a,\Pi).$$

So we get that

$$P_{f_0}\left(\max_{1\le j\le s}E\{\|f-f_0\|^aI(f\in A_n)|\mathbf{D}_j\}\ge\exp(-n\widetilde{r}_n^2)C(a,\Pi)\right)\le P_{f_0}(\mathcal{E}_n^c)\le\varepsilon/2.$$

By $\widetilde{r}_n^2 \le r_n^2 \log(2s)$, the above leads to that

$$P_{f_0}\Big(\max_{1\le j\le s} E\{\|f-f_0\|^a I(\|f-f_0\| \ge 2C'\widetilde{r}_n)|\mathbf{D}_j\} \\ \ge (M' + C(a,\Pi))s^2\exp(-n\widetilde{r}_n^2/\log(2s))\Big) \le \varepsilon.$$

Proof is completed. ■

S.8.4. Proofs of other results in Section S.8.3

Let $N(\varepsilon, \mathcal{G}, \|\cdot\|_\infty)$ be the ε-packing number in terms of supremum norm, where recall that the space $\mathcal{G}$ is defined in (S.13). The following result can be found in Van De Geer and Van De Geer (2006).

Lemma 11 *There exists a universal constant $c_0 > 0$ s.t. for any $\varepsilon > 0$,*

$$\log N(\varepsilon, \mathcal{G}, \|\cdot\|_\infty) \le c_0(\sqrt{2}c_K^{-1})^{1/m} h^{-\frac{2m-1}{2m}} \varepsilon^{-1/m}.$$

For $r \ge 0$, define $\Psi(r) = \int_0^r \sqrt{\log(1+\exp(x^{-1/m}))}dx$. For arbitrary $\varepsilon > 0$, define

$$A(h,\varepsilon) = \frac{32\sqrt{6}}{\tau}\sqrt{2}c_K^{-1}c_0^m h^{-(2m-1)/2}\Psi\left(\frac{1}{2\sqrt{2}}c_K c_0^{-m} h^{(2m-1)/2}\varepsilon\right) \\ + \frac{10\sqrt{24}\varepsilon}{\tau}\sqrt{\log\left(1+\exp\left(2c_0((\sqrt{2})^{-1}c_K h^{(2m-1)/2}\varepsilon)^{-1/m}\right)\right)}, \quad \text{(S.19)}$$

where $\tau = \sqrt{\log 1.5} \approx 0.6368$.

We have the following useful lemma.

Lemma 12 *For any $1 \le j \le s$ and $f \in S^m(\mathbb{I})$, suppose that $\psi_{j,n,f}(z;g)$ is a measurable function defined upon $z = (y,x) \in \mathcal{Y}\times\mathbb{I}$ and $g \in \mathcal{G}$ satisfying $\psi_{j,n,f}(z;0) = 0$ and the following Lipschitz continuity condition: for any $i \in I_j$ and $g_1, g_2 \in \mathcal{G}$,*

$$|\psi_{j,n,f}(Z_i;g_1) - \psi_{j,n,f}(Z_i;g_2)| \le c_K^{-1}h^{1/2}\|g_1-g_2\|_\infty. \quad \text{(S.20)}$$

Then for any constant $t \ge 0$ and $n \ge 1$,

$$\sup_{f\in S^m(\mathbb{I})} P_f\left(\sup_{g\in\mathcal{G}}\|Z_{j,n,f}(g)\|_f > t\right) \le 2\exp\left(-\frac{t^2}{B(h)^2}\right),$$

where $B(h) = A(h,2)$ and

$$Z_{j,n,f}(g) = \frac{1}{\sqrt{n}}\sum_{i\in I_j}[\psi_{j,n,f}(Z_i;g)K_{X_i} - E_f\{\psi_{j,n,f}(Z_i;g)K_{X_i}\}].$$

Proof [Proof of Lemma 12] For any $f \in S^m(\mathbb{I})$ and $n \ge 1$, and any $g_1, g_2 \in \mathcal{G}$, we get that

$$\|(\psi_{j,n,f}(Z_i;g_1) - \psi_{j,n,f}(Z_i;g_2))K_{X_i}\| \le c_K^{-1}h^{1/2}\|g_1-g_2\|_\infty c_K h^{-1/2} = \|g_1-g_2\|_\infty.$$

By Theorem 3.5 of Pinelis et al. (1994), for any $t > 0$, $P_f\left(\|Z_{j,n,f}(g_1) - Z_{j,n,f}(g_2)\| \ge t\right) \le 2\exp\left(-\frac{t^2}{8\|g_1-g_2\|_\infty^2}\right)$. Then by Lemma 8.1 in Kosorok (2008), we have

$$\big\|\|Z_{j,n,f}(g_1) - Z_{j,n,f}(g_2)\|\big\|_{\psi_2} \le \sqrt{24}\|g_1-g_2\|_\infty,$$

where $\|\cdot\|_{\psi_2}$ denotes the Orlicz norm associated with $\psi_2(s) := \exp(s^2) - 1$. Recall $\tau = \sqrt{\log 1.5} \approx 0.6368$. Define $\phi(x) = \psi_2(\tau x)$. Then it can be shown by elementary calculus that $\phi(1) \le 1/2$, and for any $x, y \ge 1$, $\phi(x)\phi(y) \le \phi(xy)$. By a careful examination of the proof of Lemma 8.2, it can be shown that for any random variables $\xi_1, \ldots, \xi_l$,

$$\|\max_{1\le i\le l}\xi_i\|_{\psi_2} \le \frac{2}{\tau}\psi_2^{-1}(l)\max_{1\le i\le l}\|\xi_i\|_{\psi_2}. \quad \text{(S.21)}$$

Next we use a "chaining" argument. Let $T_0 \subset T_1 \subset T_2 \subset \cdots \subset T_\infty := \mathcal{G}$ be a sequence of finite nested sets satisfying the following properties:

- for any T_q and any $s, t \in T_q$, $\|s-t\|_\infty \ge \varepsilon 2^{-q}$; each T_q is "maximal" in the sense that if one adds any point in T_q, then the inequality will fail;
- the cardinality of T_q is upper bounded by

  $$\log|T_q| \le \log N(\varepsilon 2^{-q}, \mathcal{G}, \|\cdot\|_\infty) \le c_0(\sqrt{2}c_K^{-1})^{1/m}h^{-(2m-1)/(2m)}(\varepsilon 2^{-q})^{-1/m},$$

  where $c_0 > 0$ is absolute constant;
- each element $t_{q+1} \in T_{q+1}$ is uniquely linked to an element $t_q \in T_q$ which satisfies $\|t_q - t_{q+1}\|_\infty \le \varepsilon 2^{-q}$.

For arbitrary $s_{k+1}, t_{k+1} \in T_{k+1}$ with $\|s_{k+1} - t_{k+1}\|_\infty \le \varepsilon$, choose two chains (both being of length $k+2$) t_q and s_q with $t_q, s_q \in T_q$ for $0 \le q \le k+1$. The ending points s_0 and t_0 satisfy

$$\|s_0 - t_0\|_\infty \le \sum_{q=0}^{k}[\|s_q - s_{q+1}\|_\infty + \|t_q - t_{q+1}\|_\infty] + \|s_{k+1} - t_{k+1}\|_\infty \\ \le 2\sum_{q=0}^{k}\varepsilon 2^{-q} + \varepsilon \le 5\varepsilon,$$

and hence, $\left\|\|Z_{j,n,f}(s_0)-Z_{j,n,f}(t_0)\|_f\right\|_{\psi_2}\le 5\sqrt{24}\varepsilon$. It follows by the proof of Theorem 8.4 of Kosorok (2008) and (S.21) that

$$\begin{aligned}
&\left\|\max_{s_{k+1},t_{k+1}\in T_{k+1}}\|Z_{j,n,f}(s_{k+1})-Z_{j,n,f}(t_{k+1})-(Z_{j,n,f}(s_0)-Z_{j,n,f}(t_0))\|\right\|_{\psi_2}\\
\le\ & 2\sum_{q=0}^{k}\left\|\max_{\substack{u\in T_{q+1},v\in T_q\\ u,v\text{ link each other}}}\|Z_{j,n,f}(u)-Z_{j,n,f}(v)\|\right\|_{\psi_2}\\
\le\ & \frac{4}{\tau}\sum_{q=0}^{k}\psi_2^{-1}(N(2^{-q-1}\varepsilon,\mathcal{G},\|\cdot\|_\infty))\\
&\times\max_{\substack{u\in T_{q+1},v\in T_q\\ u,v\text{ link each other}}}\left\|\|Z_{j,n,f}(u)-Z_{j,n,f}(v)\|\right\|_{\psi_2}\\
\le\ & \frac{4\sqrt{24}}{\tau}\sum_{q=0}^{k}\sqrt{\log\left(1+N(\varepsilon 2^{-q-1},\mathcal{G},\|\cdot\|_\infty)\right)}\varepsilon 2^{-q}\\
\le\ & \frac{8\sqrt{24}}{\tau}\sum_{q=1}^{k+1}\sqrt{\log\left(1+\exp\left(c_0c_K^{-1/m}h^{-(2m-1)/(2m)}(\varepsilon 2^{-q})^{-1/m}\right)\right)}\varepsilon 2^{-q}\\
\le\ & \frac{32\sqrt{6}}{\tau}\int_0^{\varepsilon/2}\sqrt{\log\left(1+\exp\left(c_0c_K^{-1/m}h^{-(2m-1)/(2m)}x^{-1/m}\right)\right)}dx\\
=\ & \frac{32\sqrt{6}}{\tau}c_K^{-1}c_0^{m}h^{-(2m-1)/2}\Psi\left(\frac{1}{2}c_Kc_0^{-m}h^{(2m-1)/2}\varepsilon\right).
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\left\|\max_{\substack{u,v\in T_0\\ \|u-v\|_\infty\le 5\varepsilon}}\|Z_{j,n,f}(u)-Z_{j,n,f}(v)\|_f\right\|_{\psi_2} &\le \frac{2}{\tau}\psi_2(|T_0|^2)\max_{\substack{u,v\in T_0\\ \|u-v\|_\infty\le 5\varepsilon}}\left\|\|Z_{j,n,f}(u)-Z_{j,n,f}(v)\|_f\right\|_{\psi_2}\\
&\le \frac{2}{\tau}\psi_2^{-1}(N(\varepsilon,\mathcal{G},\|\cdot\|_\infty)^2)(5\sqrt{24}\varepsilon).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\left\|\max_{\substack{s,t\in T_{k+1}\\ \|s-t\|_\infty\le\varepsilon}}\|Z_{j,n,f}(s)-Z_{j,n,f}(t)\|\right\|_{\psi_2} &\le \frac{32\sqrt{6}}{\tau}c_K^{-1}c_0^{m}h^{-(2m-1)/2}\Psi\left(\frac{1}{2}c_Kc_0^{-m}h^{(2m-1)/2}\varepsilon\right)\\
&\quad+\frac{2}{\tau}\psi_2^{-1}(N(\varepsilon,\mathcal{G},\|\cdot\|_\infty)^2)(5\sqrt{24}\varepsilon)\\
&\le \frac{32\sqrt{6}}{\tau}c_K^{-1}c_0^{m}h^{-(2m-1)/2}\Psi\left(\frac{1}{2}c_Kc_0^{-m}h^{(2m-1)/2}\varepsilon\right)\\
&\quad+\frac{10\sqrt{24}\varepsilon}{\tau}\sqrt{\log\left(1+\exp\left(2c_0(c_Kh^{(2m-1)/2}\varepsilon)^{-1/m}\right)\right)}\\
&= A(h,\varepsilon).
\end{aligned}$$

Now for any $g_1,g_2\in\mathcal{G}$ with $\|g_1-g_2\|_\infty\le\varepsilon/2$. Let $k\ge 2$, hence, $2^{1-k}\le 1-\|g_1-g_2\|_\infty/\varepsilon$. Since T_k is "maximal", there exist $s_k,t_k\in T_k$ s.t. $\max\{\|g_1-s_k\|_\infty,\|g_2-t_k\|_\infty\}\le\varepsilon 2^{-k}$. It is

easy to see that $\|s_k-t_k\|_\infty\le\varepsilon$. So

$$\begin{aligned}
\|Z_{j,n,f}(g_1)-Z_{j,n,f}(g_2)\| &\le \|Z_{j,n,f}(g_1)-Z_{j,n,f}(s_k)\|+\|Z_{j,n,f}(g_2)-Z_{j,n,f}(t_k)\|\\
&\quad+\|Z_{j,n,f}(s_k)-Z_{j,n,f}(t_k)\|\\
&\le 4\sqrt{n}\varepsilon 2^{-k}+\max_{\substack{u,v\in T_k\\ \|u-v\|_\infty\le\varepsilon}}\|Z_{j,n,f}(u)-Z_{j,n,f}(v)\|.
\end{aligned}$$

Therefore, letting $k\to\infty$ we get that

$$\begin{aligned}
&\left\|\sup_{\substack{g_1,g_2\in\mathcal{G}\\ \|g_1-g_2\|_\infty\le\varepsilon/2}}\|Z_{j,n,f}(g_1)-Z_{j,n,f}(g_2)\|\right\|_{\psi_2}\\
\le\ & 4\sqrt{n}\varepsilon 2^{-k}/\sqrt{\log 2}+\left\|\max_{\substack{u,v\in T_k\\ \|u-v\|_\infty\le\varepsilon}}\|Z_{j,n,f}(u)-Z_{j,n,f}(v)\|\right\|_{\psi_2}\\
\le\ & 4\sqrt{n}\varepsilon 2^{-k}/\sqrt{\log 2}+A(h,\varepsilon)\to A(h,\varepsilon).
\end{aligned}$$

Taking $\varepsilon=2$ in the above inequality, we get that

$$\left\|\sup_{\substack{g_1,g_2\in\mathcal{G}\\ \|g_1-g_2\|_\infty\le 1}}\|Z_{j,n,f}(g_1)-Z_{j,n,f}(g_2)\|\right\|_{\psi_2}\le A(h,2)=B(h).$$

By Lemma 8.1 in Kosorok (2008), we have

$$P_f\left(\sup_{g\in\mathcal{G}}\|Z_{j,n,f}(g)\|\ge t\right)\le 2\exp\left(-\frac{t^2}{B(h)^2}\right).$$

Note that the right hand side in the above does not depend on f. This completes the proof. ■

Proof [Proof of Lemma 4] Let $f\in H^m(b)$ be the parameter based on which the data are drawn. It is easy to see that $DS_\lambda(f)g=-E\{g(X)K_X\}-\mathcal{P}_\lambda g,\ \ \forall g\in S^m(\mathbb{I})$. Therefore, for any $g,\widetilde{g}\in S^m(\mathbb{I})$, $\langle DS_\lambda(f)g,\widetilde{g}\rangle=-\langle g,\widetilde{g}\rangle$, implying $DS_\lambda(f)=-id$.

The proof of (1) is finished in two parts.

Part I: For any $f\in S^m(\mathbb{I})$, define an operator mapping $S^m(\mathbb{I})$ to $S^m(\mathbb{I})$:

$$T_{1f}(g)=g+S_\lambda(f+g),\ g\in S^m(\mathbb{I}).$$

First observe that, under P_f with $f\in H^m(b)$,

$$\|S_\lambda(f)\|=\|\mathcal{P}_\lambda f\|=\sup_{\|g\|=1}|\langle\mathcal{P}_\lambda f,g\rangle|\le\sqrt{\lambda J(f)}\le h^m b.$$

Let $r_{1n}=bh^m$. Let $\mathbb{B}(r_{1n})=\{g\in S^m(\mathbb{I}):\|g\|\le r_{1n}\}$ be the r_{1n}-ball. For any $g\in\mathbb{B}(r_{1n})$, using $DS_\lambda(f)=-id$, it is easy to see that $\|T_{1f}(g)\|=\|S_\lambda(f)\|\le bh^m=r_{1n}$. Therefore, T_{1f}

maps $\mathbb{B}(r_{1n})$ to itself. For any $g_1, g_2 \in \mathbb{B}(r_{1n})$, by Taylor's expansion we have

$$\begin{aligned}\|T_{1f}(g_1) - T_{1f}(g_2)\| &= \|g_1 - g_2 + S_\lambda(f+g_1) - S_\lambda(f+g_2)\| \\ &= \|g_1 - g_2 + \int_0^1 DS_\lambda(f+g_2+sg)g ds\| = 0.\end{aligned}$$

This shows that T_{1f} is a contraction mapping which maps $\mathbb{B}(r_{1n})$ into $\mathbb{B}(r_{1n})$. By contraction mapping theorem (see Rudin et al. (1964)), T_{1f} has a unique fixed point $g' \in \mathbb{B}(r_{1n})$ satisfying $T_{1f}(g') = g'$. Let $f_\lambda = f + g'$. Then $S_\lambda(f_\lambda) = 0$ and $\|f_\lambda - f\| \le r_{1n}$.

Part II: For any $f \in H^m(b)$, under (14) with f being the truth, let f_λ be the function obtained in **Part I** s.t. $\|f_\lambda - f\| \le r_{1n}$. Define an operator

$$T_{2f}(g) = g + S_{j,n}(f_\lambda + g),\ g \in S^m(\mathbb{I}).$$

Rewrite T_{2f} as

$$T_{2f}(g) = [DS_{j,n}(f_\lambda)g - DS_\lambda(f_\lambda)g] + S_{j,n}(f_\lambda).$$

Denote the above two terms by I_{1f}, I_{2f}, respectively.

For any $i \in I_j$, let $R_i = (Y_i - f_\lambda(X_i))K_{X_i} - E_f\{(Y - f_\lambda(X))K_X\}$. Obviously,

$$\begin{aligned}\|E_f\{(Y - f_\lambda(X))K_X\}\| &= \sup_{\|g\|=1} |\langle E_f\{(Y - f_\lambda(X))K_X\}, g\rangle| \\ &= \sup_{\|g\|=1} |E_f\{(Y - f_\lambda(X))g(X)\}| \le \|f - f_\lambda\| \le r_{1n}.\end{aligned}$$

Therefore, $\|R_i\| \le c_K h^{-1/2}|Y_i - f_\lambda(X_i)| + r_{1n}$ which leads to that

$$E\left\{\exp\left(\frac{\|R_i\|}{c_K h^{-1/2}}\right)\right\} \le E\left(\exp(|\epsilon_i| + 1)\right) \le C_\epsilon,$$

where $C_\epsilon = E\{(|\epsilon| + 1)^2 \exp(|\epsilon| + 1)\}$. Let $\delta = hr/c_K$. By condition $rh^{1/2} \le 1$, we have

$$E\{\exp(\delta\|R_i\|) - 1 - \delta\|R_i\|\} \le E\{(\delta\|R_i\|)^2 \exp(\delta\|R_i\|)\} \le c_K^2 C_\epsilon \delta^2 h^{-1}.$$

It follows by Theorem 3.2 of Pinelis et al. (1994) that, for $L(M) := c_K(C_\epsilon + M)$,

$$\begin{aligned}P_f\left(\|\sum_{i\in I_j} R_i\|_f \ge L(M)nr\right) &\le 2\exp\left(-L(M)\delta nr + c_K^2 C_\epsilon n h^{-1}\delta^2\right) \\ &= 2\exp(-Mnhr^2). \qquad \text{(S.22)}\end{aligned}$$

We note that the right hand side in (S.22) does not depend on f. Moreover, it is easy to see that $S_{j,n}(f_\lambda) = S_{j,n}(f_\lambda) - S_\lambda(f_\lambda) = \frac{1}{n}\sum_{i\in I_j} R_i$. Let

$$\mathcal{E}_{n,1} = \{\|S_{j,n}(f_\lambda)\| \le L(M)r\},$$

then $\sup_{f\in H^m(C)} P_f(\mathcal{E}_{n,1}^c) \le 2\exp(-Mnhr^2)$. Define $\psi_{j,n}(X_i; g) = c_K^{-1} h^{1/2} g(X_i),\ i \in I_j$, and $Z_{j,n}(g) = \frac{1}{\sqrt{n}}\sum_{i\in I_j}[\psi_{j,n}(X_i; g)K_{X_i} - E_f\{\psi_{j,n}(X_i; g)K_{X_i}\}]$. By Lemma 12, $\sup_{f\in H^m(b)} P_f(\mathcal{E}_{n,2}^c) \le 2\exp(-Mnhr^2)$, where $\mathcal{E}_{n,2} = \{\sup_{g\in\mathcal{G}} \|Z_{j,n}(g)\| \le \sqrt{Mnhr^2}B(h)\}$.

For any $g \in S^m(\mathbb{I})\backslash\{0\}$, let $\bar{g} = g/d_n'$, where $d_n' = c_K h^{-1/2}\|g\|$. It follows that

$$\|\bar{g}\|_\infty \le c_K h^{-1/2}\|\bar{g}\| = c_K h^{-1/2}\|g\|/d_n' = 1, \text{ and}$$

$$J(\bar{g}, \bar{g}) = d_n'^{-2} J(g, g) = h^{-2m}\frac{\lambda J(g,g)}{c_K^2 h^{-1}\|g\|^2} \le h^{-2m}\frac{\|g\|^2}{c_K^2 h^{-1}\|g\|^2} \le c_K^{-2} h^{-2m+1}.$$

Therefore, $\bar{g} \in \mathcal{G}$. Consequently, on $\mathcal{E}_{n,2}$, for any $g \in S^m(\mathbb{I})\backslash\{0\}$, we get $\|Z_{j,n}(\bar{g})\| \le \sqrt{Mnhr^2}B(h)$, which leads to that

$$\begin{aligned}\|DS_{j,n}(f_\lambda)g - DS_\lambda(f_\lambda)g\| &= \frac{1}{n}\|\sum_{i\in I_j}[g(X_i)K_{X_i} - E\{g(X_i)K_{X_i}\}]\|_f \\ &\le c_K^2 M^{1/2} r h^{-1/2} B(h)\|g\| \le \|g\|/2, \qquad \text{(S.23)}\end{aligned}$$

where the last inequality follows by condition $c_K^2 M^{1/2} r h^{-1/2} B(h) \le 1/2$. Note that the above inequality also holds for $g = 0$.

Let $r_{2n} = 2L(M)r$. Therefore, it follows by (S.23) that, for any $f \in H^m(b)$, on $\mathcal{E}_n := \mathcal{E}_{n,1} \cap \mathcal{E}_{n,2}$ and for any $g \in \mathbb{B}(r_{2n})$, $\|T_{2f}(g)\| \le \|g\|/2 + r_{2n}/2 \le r_{2n}$. Meanwhile, for any $g_1, g_2 \in \mathbb{B}(r_{2n})$, replacing g by $g_1 - g_2$ in (S.23), we get that $\|T_{2f}(g_1) - T_{2f}(g_2)\| \le \|g_1 - g_2\|/2$. Therefore, for any $f \in H^m(b)$, on $\mathcal{E}_n$, T_{2f} is a contraction mapping from $\mathbb{B}(r_{2n})$ to itself. By contraction mapping theorem, there exists uniquely an element $g'' \in \mathbb{B}(r_{2n})$ s.t. $T_{2f}(g'') = g''$. Let $\widehat{f}_{j,n} = f_\lambda + g''$. Clearly, $S_{j,n}(\widehat{f}_{j,n}) = 0$, and hence, $\widehat{f}_{j,n}$ is the maximizer of ℓ_{jn}; see (19). So we get that, on $\mathcal{E}_n$, $\|\widehat{f}_{j,n} - f\|_f \le \|f_\lambda - f\| + \|\widehat{f}_{j,n} - f_\lambda\| \le r_{1n} + r_{2n} = bh^m + 2L(M)r$. The desired conclusion follows by the trivial fact: $\sup_{f\in H^m(b)} P_f(\mathcal{E}_n^c) \le 4\exp(-Mnhr^2)$. Proof of (1) is completed.

Next we show (2).

For any $f \in H^m(b)$, let $\widehat{f}_{j,n}$ be the penalized MLE of f obtained by (19). Let $g_n = \widehat{f}_{j,n} - f$, $\delta_n = bh^m + 2L(M)r$, $d_n' = c_K h^{-1/2}\delta_n$.

On $\mathcal{E}_n$, we have $\|g_n\|_f \le \delta_n$. Let $\bar{g} = g_n/d_n'$. Clearly, $\bar{g} \in \mathcal{G}$. Then we get that

$$\begin{aligned}&\|S_{j,n}(f+g_n) - S_{j,n}(f) - (S_\lambda(f+g_n) - S_\lambda(f))\| \\ &= \frac{1}{n}\|\sum_{i\in I_j}[g_n(X_i)K_{X_i} - E_X\{g_n(X)K_X\}]\| \\ &= \frac{c_K d_n'}{\sqrt{nh}}\|Z_{j,n}(\bar{g})\| \le c_K^2 M^{1/2} h^{-1/2} r B(h)\delta_n = a_n. \qquad \text{(S.24)}\end{aligned}$$

Since $S_{j,n}(f+g_n) = 0$ and $DS_\lambda(f) = -id$, from (S.24) we have on $\mathcal{E}_n$,

$$a_n \ge \|S_{j,n}(f) + DS_\lambda(f)g_n + \int_0^1\int_0^1 sD^2S_\lambda(f + ss'g_n)g_n g_n ds ds'\| = \|S_{j,n}(f) - g_n\|$$

which implies that $\|\widehat{f}_{j,n} - f - S_{n,\lambda}(f)\| \le a_n$. Since $\sup_{f\in H^m(bC)} P_f(\mathcal{E}_n^c) \le 4\exp(-Mnhr^2)$, proof of (2) is completed. ■

S.8.5. An initial contraction rate

Theorem 7 below states that the s posterior measures uniformly contract at rate $r_n = (nh)^{-1/2} + h^m$, where recall that $h = \lambda^{1/(2m)}$. This is an initial rate result that holds irrespective the diverging rate of s.

Theorem 7 *(An Initial Contraction Rate) Suppose $f_0 = \sum_{\nu=1}^{\infty} f_\nu^0 \varphi_\nu$ satisfies Condition (**S**). Let $a \geq 0$ be a fixed constant. If $r_n = o(h^{3/2})$, $h^{1/2}\log N = o(1)$, $nh^{2m+1} \geq 1$, then there exists a universal constant $M > 0$ s.t.*

$$\max_{1\leq j\leq s} E\{\|f - f_0\|^a I(\|f - f_0\| \geq Mr_n)|\boldsymbol{D}_j\} = O_{P_{f_0}}(s^2 \exp(-nr_n^2))$$

as $n \to \infty$, no matter s is fixed or diverges at any rate.

Before proving Theorem 7, we present a preliminary lemma.

Let $\{\widetilde{\varphi}_\nu : \nu \geq 1\}$ be a bounded orthonormal basis of $L^2(\mathbb{I})$ under usual L^2 inner product. For any $b \in [0, \beta]$, define

$$\widetilde{H}_b = \{\sum_{\nu=1}^{\infty} f_\nu \widetilde{\varphi}_\nu : \sum_{\nu=1}^{\infty} f_\nu^2 \rho_\nu^{1+b/(2m)} < \infty\}.$$

Then $\widetilde{H}_b$ can be viewed as a version of Sobolev space with regularity $m + b/2$. Define $\widetilde{G} = \sum_{\nu=1}^{\infty} v_\nu \widetilde{\varphi}_\nu$, a centered GP, and $\widetilde{f}_0 = \sum_{\nu=1}^{\infty} f_\nu^0 \widetilde{\varphi}_\nu$. Define $\widetilde{V}(f, g) = \langle f, g\rangle_{L^2} = \int_0^1 f(x)g(x)dx$, the usual L^2 inner product, $\widetilde{J}(f) = \sum_{\nu=1}^{\infty} |\widetilde{V}(f, \widetilde{\varphi}_\nu)|^2 \rho_\nu$, a functional on $\widetilde{H}_0$. For simplicity, denote $\widetilde{V}(f) = \widetilde{V}(f, f)$. Clearly, $\widetilde{f}_0 \in \widetilde{H}_\beta$. Since $\widetilde{G}$ is a Gaussian process with covariance function

$$\widetilde{r}(s, t) = E\{\widetilde{G}(s)\widetilde{G}(t)\} = \sum_{\nu=1}^{m} \sigma_\nu^2 \widetilde{\varphi}_\nu(s)\widetilde{\varphi}_\nu(t) + \sum_{\nu>m} \rho_\nu^{-(1+\frac{\beta}{2m})} \widetilde{\varphi}_\nu(s)\widetilde{\varphi}_\nu(t),$$

it follows by van der Vaart et al. (2008a) that $\widetilde{H}_\beta$ is the RKHS of $\widetilde{G}$. For any $\widetilde{H}_b$ with $0 \leq b \leq \beta$, define inner product

$$\langle \sum_{\nu=1}^{\infty} f_\nu \widetilde{\varphi}_\nu, \sum_{\nu=1}^{\infty} g_\nu \widetilde{\varphi}_\nu\rangle_b = \sum_{\nu=1}^{m} \sigma_\nu^{-2} f_\nu g_\nu + \sum_{\nu>m} f_\nu g_\nu \rho_\nu^{1+\frac{b}{2m}}.$$

Let $\|\cdot\|_b$ be the norm corresponding to the above inner product. The following lemma is used in the proof of Theorem 7. Its proof can be found in Shang and Cheng (2017).

Lemma 13 *Let d_n be any positive sequence. If Condition (**S**) holds, then there exists $\omega \in \widetilde{H}_\beta$ such that*

1. $\widetilde{V}(\omega - \widetilde{f}_0) \leq \frac{1}{4} d_n^2$,
2. $\widetilde{J}(\omega - \widetilde{f}_0) \leq \frac{1}{4} d_n^{\frac{2(\beta-1)}{2m+\beta-1}}$,
3. $\|\omega\|_\beta^2 = O(d_n^{-\frac{2}{2m+\beta-1}})$.

To ease reading, we sketch the proof of Theorem 7. We first show the following result: for any $\varepsilon > 0$, as $n \to \infty$,

$$\max_{1\leq j\leq s} \int_{\|f-f_0\|_\infty \geq \varepsilon} \|f - f_0\|^a dP(f|\mathbf{D}_j) = O_{P_{f_0}}(s^2 \exp(-nr_n^2)) \tag{S.25}$$

To show (S.25), we can rewrite the posterior density of f by

$$p(f|\mathbf{D}_j) = \frac{\prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-n\lambda J(f)/2)d\Pi(f)}{\int_{S^m(\mathbb{I})}\prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-n\lambda J(f)/2)d\Pi(f)},\ 1 \leq j \leq s,$$

where recall that $p_f(z)$ is the probability density of $Z = (Y, X)$ under f. For $1 \leq j \leq s$, define

$$I_{j1} = \int_{S^m(\mathbb{I})} \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-n\lambda J(f)/2)d\Pi(f), \tag{S.26}$$

$$I_{j2} = \int_{A_n} \|f - f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-\frac{n\lambda}{2}J(f))d\Pi(f), \tag{S.27}$$

$$I'_{j2} = \int_{A'_n} \|f - f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-\frac{n\lambda}{2}J(f))d\Pi(f), \tag{S.28}$$

where $A_n = \{f \in S^m(\mathbb{I}) : \|f - f_0\| \geq 2\delta_n\}$ and $A'_n = \{f \in S^m(\mathbb{I}) : \|f - f_0\| \geq \sqrt{2}Mr_n\}$, with the quantities δ_n, M specified later. Using LeCam's uniformly consistent test Ghosal et al. (2000), we will show that $\max_{1\leq j\leq s} I_{j2}/I_{j1}$ is of an exponential order (in the sense of P_{f_0}). Then (S.25) holds by taking $a = 0$ in I_{j2}. The proof of Theorem 7 will be completed by decomposing I'_{j2}/I_{j1} into three terms based on an auxiliary event $\{f \in S^m(\mathbb{I}) : \|f - f_0\|_\infty \leq \varepsilon\}$ with each term of an exponential order.

Proof [Proof of Theorem 7] Note that there exists a universal constant $c' > 0$ such that $\Psi(x) \leq c'x^{1-1/(2m)}$ for any $0 < x < 1$. Therefore, there exists a universal constant $c'' > 0$ s.t. $B(h) \leq c''h^{-(2m-1)/(4m)}$.

Define $B_n = \{f \in S^m(\mathbb{I}) : V(f - f_0) \leq r_n^2, J(f - f_0) \leq r_n^{\frac{2(\beta-1)}{2m+\beta-1}}\}$. Then

$$\begin{aligned} I_{j1} &\geq \int_{B_n} \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-n\lambda J(f)/2)d\Pi(f) \\ &= \int_{B_n} \exp(\sum_{i\in I_j} R_i(f, f_0))\exp(-n\lambda J(f)/2)d\Pi(f), \end{aligned}$$

where $R_i(f, f_0) = \log\big(p_f(Z_i)/p_{f_0}(Z_i)\big) = Y_i(f(X_i) - f_0(X_i)) - f(X_i)^2/2 + f_0(X_i)^2/2$ for any $i \in I_j$. Define $d\Pi^*(f) = d\Pi(f)/\Pi(B_n)$, a reduced probability measure on B_n. By Jensen's

inequality,

$$\begin{aligned}
& \log \int_{B_n} \exp(\sum_{i\in I_j} R_i(f,f_0)) \exp(-n\lambda J(f)/2) d\Pi^*(f) \\
\geq & \int_{B_n} \left(\sum_{i\in I_j} R_i(f,f_0) - n\lambda J(f)/2 \right) d\Pi^*(f) \\
= & \int_{B_n} \sum_{i\in I_j} [R_i(f,f_0) - E_{f_0}\{R_i(f,f_0)\}] d\Pi^*(f) \\
& + n \int_{B_n} E_{f_0}\{R_i(f,f_0)\} d\Pi^*(f) - \int_{B_n} \frac{n\lambda J(f)}{2} d\Pi^*(f) \\
:= & J_{j1} + J_{j2} + J_{j3}.
\end{aligned}$$

For any $f \in B_n$, $\|f - f_0\|^2 = V(f - f_0) + \lambda J(f - f_0) \leq r_n^2 + \lambda r_n^{\frac{2(\beta-1)}{2m+\beta-1}}$. By (Shang and Cheng, 2017, Lemma A.9) and the condition $h^{-3/2} r_n = o(1)$, we can choose n to be sufficiently large so that $\|f - f_0\|_\infty \leq c h^{-1/2} \|f - f_0\| \leq c\sqrt{h^{-1} r_n^2 + h^{2m-1}} \leq 1$.

It follows by Taylor's expansion and $E_{f_0}\{Y_i - f_0(X_i)|X_i\} = 0$, that for any $f \in B_n$,

$$|E_{f_0}\{R_i(f,f_0)\}| = E_{f_0}\{(f(X) - f_0(X))^2\}/2 \leq r_n^2/2.$$

Therefore, $J_{j2} \geq -n r_n^2/2$ for any $1 \leq j \leq s$.

Since $r_n^2 = o(1)$, we can choose n to be large so that $|E_{f_0}\{R_i(f,f_0)\}| \leq 1$. Meanwhile, for any $f \in B_n$, for some $s \in [0,1]$, we have

$$\begin{aligned}
|R_i(f,f_0)| &= |Y_i(f(X_i) - f_0(X_i)) - f(X_i)^2/2 + f_0(X_i)^2/2| \\
&= |Y_i - f_0(X_i) - \frac{1}{2}(f - f_0)(X_i)| \times |(f - f_0)(X_i)| \\
&\leq |Y_i - f_0(X_i)| + 1/2 = |\epsilon_i| + 1/2.
\end{aligned}$$

We have used $\|f - f_0\|_\infty \leq 1$ in the above inequalities.

For any $1 \leq i \leq N$, define $A_i = \{|\epsilon_i| \leq 2\log N\}$. It is easy to check that $P_{f_0}(\cup_{i=1}^N A_i^c) \to 0$, as $N \to \infty$. Define $\xi_i = \int_{B_n} R_i(f,f_0) d\Pi^*(f) \times I_{A_i}$, we get that $|\xi_i| \leq 2\log N + 1/2$, a.s. It can also be shown by $r_n^2 \gg 1/n \geq 1/N$ that, as $n, N \to \infty$,

$$\begin{aligned}
|E_{f_0}\{\int_{B_n} R_i(f,f_0) d\Pi^*(f) \times I_{A_i^c}\}| &\leq E_{f_0}\{(|\epsilon_i| + 1/2) \times I_{A_i^c}\} \\
&\leq C_\epsilon(1/N + 1/N^2) \leq r_n^2,
\end{aligned}$$

where C_ϵ is an absolute constant.

Let $\delta = 1/(\sqrt{n} r_n)$. Note that by the condition $h^{1/2}\log N = o(1)$ we have $\delta \log N = (\log N)/(\sqrt{n} r_n) \leq h^{1/2} \log N = o(1)$, we can let n be large so that $\delta(2\log N + 1) \leq 1$. Let $d_i = \xi_i - E_{f_0}\{\xi_i\}$ for $i \in I_j$, then it is easy to see that

$$|d_i| \leq |\xi_i| + |E_{f_0}\{\xi_i\}| \leq 2\log N + 1, \ a.s.$$

Let $e_i = E_{f_0}\{\exp(\delta|d_i|) - 1 - \delta|d_i|\}$. It can be shown using inequality $\exp(x) - 1 - x \leq x^2 \exp(x)$ for $x \geq 0$ and Cauchy-Schwartz inequality that

$$\begin{aligned}
|e_i| &\leq E_{f_0}\{\delta^2 d_i^2 \exp(\delta|d_i|)\} \\
&\leq e\delta^2 E_{f_0}\{d_i^2\} \\
&\leq e\delta^2 E_{f_0}\{\xi_i^2\} \\
&\leq e\delta^2 \int_{B_n} E_{f_0}\{R_i(f,f_0)^2\} d\Pi^*(f) \\
&\leq e\delta^2 \int_{B_n} E_{f_0}\{(|\epsilon_i| + 1/2)^2 (f - f_0)(X_i)^2\} d\Pi^*(f) \\
&\leq e C_\epsilon \delta^2 r_n^2,
\end{aligned}$$

where the last step follows from $V(f - f_0) \leq r_n^2$ for any $f \in B_n$. Therefore, it follows by (Pinelis et al., 1994, Theorem 3.2) that

$$\begin{aligned}
& P_{f_0}\left(\max_{1\leq j\leq s} |\sum_{i\in I_j} [\xi_i - E_{f_0}\{\xi_i\}]| \geq 4\sqrt{n} r_n \log N \right) \\
\leq & \ s P_{f_0}\left(|\sum_{i\in I_j} [\xi_i - E_{f_0}\{\xi_i\}]| \geq 4\sqrt{n} r_n \log N \right) \\
\leq & \ 2s \exp(-4\sqrt{n} r_n (\log N)\delta + e C_\epsilon \delta^2 n r_n^2) \\
\leq & \ 2s/N^2 \to 0, \text{ as } N \to \infty.
\end{aligned} \tag{S.29}$$

Since $\sqrt{n} r_n \gg \log N$, we can let n be large so that $4\sqrt{n} r_n \log N \leq n r_n^2$. Since on $\cap_{i=1}^N A_i$,

$$J_{j1} = \sum_{i\in I_j} [\xi_i - E_{f_0}\{\xi_i\}] - n E_{f_0}\{\int_{B_n} R_i(f,f_0) d\Pi^*(f) \times I_{A_i^c}\},$$

we get from (S.29) that with P_{f_0}-probability approaching one, for any $1 \leq j \leq s$,

$$J_{j1} \geq -4\sqrt{n} r_n \log N - n r_n^2 \geq -2n r_n^2.$$

Meanwhile, for any $f \in B_n$, $J(f) \leq (1 + J(f_0)^{1/2})^2$. Therefore, $J_{j3} \geq -\frac{(1+J(f_0)^{1/2})^2}{2} n\lambda$. So, with probability approaching one, for any $1 \leq j \leq s$,

$$I_{j1} \geq \exp\left(-5n r_n^2/2 - \frac{(1 + J(f_0)^{1/2})^2}{2} n\lambda \right) \Pi(B_n).$$

To proceed, we need a lower bound for $\Pi(B_n)$. It follows by Lemma 13 by replacing d_n therein by r_n, by Gaussian correlation inequality (see Theorem 1.1 of Li et al. (1999)), by Cameron-Martin theorem (see Cameron and Martin (1944) or (Kuelbs et al., 1994, eqn

(4.18))) and (Hoffmann-Jorgensen et al., 1979, Example 4.5) that

$$\begin{aligned}
\Pi(B_n) &= P(V(G-f_0)\le r_n^2, J(G-f_0)\le r_n^{\frac{2(\beta-1)}{2m+\beta-1}}) \\
&= P(\widetilde{V}(\widetilde{G}-\widetilde{f}_0)\le r_n^2, \widetilde{J}(\widetilde{G}-\widetilde{f}_0)\le r_n^{\frac{2(\beta-1)}{2m+\beta-1}}) \\
&\ge P(\widetilde{V}(\widetilde{G}-\omega)\le r_n^2/4, \widetilde{J}(\widetilde{G}-\omega)\le r_n^{\frac{2(\beta-1)}{2m+\beta-1}}/4) \\
&\ge \exp(-\frac{1}{2}\|\omega\|_\beta^2)P(\widetilde{V}(\widetilde{G})\le r_n^2/4, \widetilde{J}(\widetilde{G})\le r_n^{\frac{2(\beta-1)}{2m+\beta-1}}/4) \\
&\ge \exp(-\frac{1}{2}\|\omega\|_\beta^2)P(\widetilde{V}(\widetilde{G})\le r_n^2/8)P(\widetilde{J}(\widetilde{G})\le r_n^{\frac{2(\beta-1)}{2m+\beta-1}}/8) \\
&\ge \exp(-c_1 r_n^{-2/(2m+\beta-1)}),
\end{aligned} \tag{S.30}$$

where $c_1>0$ is a universal constant.

Since $\beta>1$ and $r_n^2=(nh)^{-1}+\lambda\ge n^{-2m/(2m+1)}$, we get $r_n^2\ge\lambda$ and $nr_n^{\frac{2(2m+\beta)}{2m+\beta-1}}\ge n^{1-\frac{2m(2m+\beta)}{(2m+1)(2m+\beta-1)}}>1$, so $nr_n^2>r_n^{-\frac{2}{2m+\beta-1}}$. Consequently, with P_{f_0}-probability approaching one

$$\min_{1\le j\le s} I_{j1}\ge\exp(-c_2 nr_n^2), \tag{S.31}$$

where $c_2=5/2+(1+J(f_0)^{1/2})^2/2+c_1$.

Let $b=2\sqrt{c_2+1}$ and $C\ge b^2/4$. Next we examine I_{j2} defined in (S.27) with $A_n=\{f\in S^m(\mathbb{I}):\|f-f_0\|\ge 2\delta_n\}$, for $\delta_n=bh^m+2c_K(C_\epsilon+C)r$, $r=r_nh^{-1/2}$. By the condition $h^{-3/2}r_n=o(1)$ and $B(h)\lesssim h^{-(2m-1)/(4m)}$ it can be easily checked that the Rate Condition (**H**): is satisfied (when n becomes large) with M therein replaced by C. For $1\le j\le s$, define test $\phi_{j,n}=I(\|\widehat{f}_{j,n}-f_0\|\ge\delta_n)$. It follows by part (1) of Theorem 4 that for any $1\le j\le s$,

$$E_{f_0}\{\phi_{j,n}\}=P_{f_0}(\|\widehat{f}_{j,n}-f_0\|\ge\delta_n)\le 2\exp(-Cnr_n^2),$$

and

$$\begin{aligned}
\sup_{\substack{f\in H^m(b)\\ \|f-f_0\|\ge 2\delta_n}} E_f\{1-\phi_{j,n}\} &= \sup_{\substack{f\in H^m(b)\\ \|f-f_0\|\ge 2\delta_n}} P_f(\|\widehat{f}_{j,n}-f_0\|<\delta_n) \\
&\le \sup_{\substack{f\in H^m(b)\\ \|f-f_0\|\ge 2\delta_n}} P_f(\|\widehat{f}_{j,n}-f\|\ge\delta_n)\le 2\exp(-Cnr_n^2).
\end{aligned}$$

An immediate consequence is $E_{f_0}\{\max_{1\le j\le s}\phi_{j,n}\}\le 2s\exp(-Cnr_n^2)$, which implies $\max_{1\le j\le s}\phi_{j,n}=O_{P_{f_0}}(s\exp(-Cnr_n^2))$.

Note that for any $f\in A_n\backslash H^m(b)$, $J(f)>b^2$. Since $nh^{2m+1}\ge 1$ leads to $r_n^2=(nh)^{-1}+\lambda\le 2\lambda$, it then holds that, for any $1\le j\le s$,

$$\begin{aligned}
& E_{f_0}\{I_{j2}(1-\phi_{j,n})\} \\
=& \int_{A_n}\|f-f_0\|^a E_f\{1-\phi_{j,n}\}\exp(-n\lambda J(f)/2)d\Pi(f) \\
=& \int_{A_n\backslash H^m(b)}\|f-f_0\|^a E_f\{1-\phi_{j,n}\}\exp(-n\lambda J(f)/2)d\Pi(f) \\
&+\int_{A_n\cap H^m(b)}\|f-f_0\|^a E_f\{1-\phi_{j,n}\}\exp(-n\lambda J(f)/2)d\Pi(f) \\
\le& \left(\exp(-b^2n\lambda/2)+2\exp(-Cnr_n^2)\right)C(a,\Pi) \\
\le& 3\exp(-b^2nr_n^2/4)C(a,\Pi),
\end{aligned}$$

where the last inequality follows by $C\ge b^2/4$ and $\lambda\ge r_n^2/2$. So

$$E_{f_0}\{\max_{1\le j\le s} I_{j2}(1-\phi_{j,n})\}\le\sum_{j=1}^s E_{f_0}\{I_{j2}(1-\phi_{j,n})\}\le 3s\exp(-b^2nr_n^2/4)C(a,\Pi),$$

which implies $\max_{1\le j\le s} I_{j2}(1-\phi_{j,n})=O_{P_{f_0}}(s\exp(-b^2nr_n^2/4))$. On the other hand, as $n\to\infty$,

$$E_{f_0}\{\max_{1\le j\le s} I_{j2}\}\le s\int_{S^m(\mathbb{I})}\|f-f_0\|^2 d\Pi(f)$$

which implies that $\max_{1\le j\le s} I_{j2}=o_{P_{f_0}}(s)$. Therefore,

$$\max_{1\le j\le s}\frac{I_{j2}}{I_{j1}}\phi_{j,n}\le\frac{\max_{1\le j\le s} I_{j2}\times\max_{1\le j\le s}\phi_{j,n}}{\min_{1\le j\le s} I_{j1}}=O_{P_{f_0}}(s^2\exp(-nr_n^2)). \tag{S.32}$$

By the above arguments and (S.31), we have

$$\begin{aligned}
\max_{1\le j\le s}\int_{A_n}\|f-f_0\|^a dP(f|\mathbf{D}_j) &= \max_{1\le j\le s}\frac{I_{j2}}{I_{j1}} \\
&\le \max_{1\le j\le s}\frac{I_{j2}}{I_{j1}}\phi_{j,n}+\max_{1\le j\le s}\frac{I_{j2}(1-\phi_{j,n})}{I_{j1}} \\
&= O_{P_{f_0}}(s^2\exp(-nr_n^2))+O_{P_{f_0}}(s\exp(-b^2nr_n^2/4)\exp(c_2nr_n^2)) \\
&= O_{P_{f_0}}(s^2\exp(-nr_n^2)).
\end{aligned}$$

By condition $r_nh^{-3/2}=o(1)$ and the trivial fact $\delta_n\asymp r_nh^{-1/2}$, we have that $h^{-1/2}\delta_n=o(1)$. Therefore, eventually $\int_{\|f-f_0\|_\infty\ge\varepsilon}\|f-f_0\|^a dP(f|\mathbf{D}_j)\le\int_{A_n}\|f-f_0\|^a dP(f|\mathbf{D}_j)$ for all $1\le j\le s$, which implies that (S.25) holds.

Now we will prove the theorem. Let I'_{j2} be defined as in (S.28) with $A'_n=\{f\in S^m(\mathbb{I}):\|f-f_0\|\ge\sqrt{2}Mr_n\}$ for a fixed number satisfying $M>\max\{2,J(f_0)^{1/2}+\sqrt{2(c_2+1)},1+\|f_0\|_\infty\}$ (M will be further described). Let $A'_{n1}=\{f\in S^m(\mathbb{I}):V(f-f_0)\ge M^2r_n^2,\lambda J(f-f_0)\le M^2r_n^2\}$ and $A'_{n2}=\{f\in S^m(\mathbb{I}):\lambda J(f-f_0)\ge M^2r_n^2\}$. For any $f\in A'_{n2}$, it can be shown that

$$Mr_n\le\sqrt{\lambda J(f-f_0)}\le\sqrt{\lambda}(J(f)^{1/2}+J(f_0)^{1/2})\le(\lambda J(f))^{1/2}+J(f_0)^{1/2}r_n,$$

which leads to $\lambda J(f) \geq (M - J(f_0)^{1/2})^2 r_n^2$. So we have

$$\begin{aligned}
& E_{f_0}\{\max_{1\leq j\leq s}\int_{A'_{n2}} \|f-f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-\frac{n\lambda}{2}J(f))d\Pi(f)\} \\
\leq\ & \sum_{j=1}^{s} E_{f_0}\{\int_{A'_{n2}} \|f-f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-\frac{n\lambda}{2}J(f))d\Pi(f)\} \\
=\ & s\int_{A'_{n2}} \|f-f_0\|^a \exp(-\frac{n\lambda}{2}J(f))d\Pi(f)\} \\
\leq\ & s\exp(-(M-J(f_0)^{1/2})^2 nr_n^2/2)C(a,\Pi),
\end{aligned}$$

which leads to that

$$\begin{aligned}
& \max_{1\leq j\leq s}\int_{A'_{n2}} \|f-f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-\frac{n\lambda}{2}J(f))d\Pi(f) \\
=\ & O_{P_{f_0}}(s\exp(-(M-J(f_0)^{1/2})^2 nr_n^2/2)). \qquad \text{(S.33)}
\end{aligned}$$

It follows from (S.31) and (S.33) that

$$\begin{aligned}
& \max_{1\leq j\leq s}\frac{1}{I_{j1}}\int_{A'_{n2}} \|f-f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-\frac{n\lambda}{2}J(f))d\Pi(f) \\
=\ & O_{P_{f_0}}\left(s\exp(-(M-J(f_0)^{1/2})^2 nr_n^2/2 + c_2 nr_n^2)\right) = O_{P_{f_0}}(s\exp(-nr_n^2)), \qquad \text{(S.34)}
\end{aligned}$$

where the last inequality follows by $(M - J(f_0)^{1/2})^2 > 2(c_2+1)$.

To continue, we need to build uniformly consistent test. Let $d_H^2(P_f, P_g) = \frac{1}{2}\int(\sqrt{dP_f} - \sqrt{dP_g})^2$ be the squared Hellinger distance between the two probability measures $P_f(z)$ and $P_g(z)$. Recall that their corresponding probability density functions are p_f and p_g, respectively. Nextwe present a lemma showing the local equivalence of V and d_H^2.

Lemma 14 *Let $\varepsilon \in (0,1)$ satisfy $\varepsilon^2 + 32\varepsilon\exp(1/2)C_\epsilon \leq 2$, where $C_\epsilon = E\{\exp(|\epsilon|)\}$. Then for any $f, g \in S^m(\mathbb{I})$ satisfying $\|f-g\|_\infty \leq \varepsilon$, $V(f-g)/16 \leq d_H^2(P_f, P_g) \leq 3V(f-g)/16$.*

Let ε satisfy the conditions in Lemma 14. Define $\mathcal{F}_n = \{f \in S^m(\mathbb{I}) : \|f-f_0\|_\infty \leq \varepsilon/2, J(f) \leq (M + J(f_0)^{1/2})^2 r_n^2 \lambda^{-1}\}$. Let $\mathcal{P}_n = \{P_f : f \in \mathcal{F}_n\}$ and $D(\delta, \mathcal{P}_n, d_H)$ be the δ-packing number in terms of d_H. Since $r_n^2 \geq \lambda$ which leads to $(M + J(f_0)^{1/2})r_n h^{-m} > M + J(f_0)^{1/2} > \varepsilon + \|f_0\|_\infty$, it can be easily checked that $\mathcal{F}_n \subset (M + J(f_0)^{1/2})r_n h^{-m}\mathcal{T}$, where $\mathcal{T} = \{f \in S^m(\mathbb{I}) : \|f\|_\infty \leq 1, J(f) \leq 1\}$.

For any $f, g \in \mathcal{F}_n$ with $\|f-g\|_\infty \leq \varepsilon$, it follows by Lemma 14 that $D(\delta, \mathcal{P}_n, d_H) \leq D(4\delta/\sqrt{3}, \mathcal{F}_n, d_V)$, where d_V is the distance induced by V, i.e., $d_V(f,g) = V^{1/2}(f-g)$. And hence, it follows by (Kosorok, 2008, Theorem 9.21) that

$$\begin{aligned}
\log D(\delta, \mathcal{P}_n, d_H) &\leq \log D(4\delta/\sqrt{3}, \mathcal{F}_n, d_V) \\
&\leq \log D(4\delta/\sqrt{3}, (M + J(f_0)^{1/2})r_n h^{-m}\mathcal{T}, d_V) \\
&\leq c_V\left(\frac{\delta}{(M + J(f_0)^{1/2})r_n h^{-m}}\right)^{-1/m},
\end{aligned}$$

where c_V is a universal constant only depending on the regularity level m. This implies that for any $\delta > 2r_n$,

$$\begin{aligned}
\log D(\delta/2, \mathcal{P}_n, d_H) &\leq \log D(r_n, \mathcal{P}_n, d_H) \\
&\leq c_V(M + J(f_0)^{1/2})^{1/m} h^{-1} \\
&\leq c_V(M + J(f_0)^{1/2})^{1/m} nr_n^2,
\end{aligned}$$

where the last inequality follows by the fact $r_n^2 \geq (nh)^{-1}$. Thus, the right side of the above inequality is constant in δ. By (Ghosal et al., 2000, Theorem 7.1), with $\delta = Mr_n/4$, there exists test $\widetilde{\phi}_{j,n}$ and a universal constant $k_0 > 0$ satisfying

$$\begin{aligned}
E_{f_0}\{\widetilde{\phi}_{j,n}\} &= P_{f_0}\widetilde{\phi}_{j,n} \\
&\leq \frac{\exp(c_V(M + J(f_0)^{1/2})^{1/m} nr_n^2)\exp(-k_0 n\delta^2)}{1 - \exp(-k_0 n\delta^2)} \\
&= \frac{\exp(c_V(M + J(f_0)^{1/2}) nr_n^2 - k_0 M^2 nr_n^2/16)}{1 - \exp(-k_0 M^2 nr_n^2/16)},
\end{aligned}$$

and, combined with Lemma 14,

$$\begin{aligned}
\sup_{\substack{f\in\mathcal{F}_n \\ d_V(f,f_0)\geq 4\delta}} E_f\{1-\widetilde{\phi}_{j,n}\} &= \sup_{\substack{f\in\mathcal{F}_n \\ d_V(f,f_0)\geq 4\delta}} P_f\{1-\widetilde{\phi}_{j,n}\} \\
&\leq \sup_{\substack{f\in\mathcal{F}_n \\ d_H(P_f,P_{f_0})\geq \delta}} P_f\{1-\widetilde{\phi}_{j,n}\} \\
&\leq \exp(-k_0 n\delta^2) = \exp(-k_0 M^2 nr_n^2/16).
\end{aligned}$$

This implies that

$$\begin{aligned}
& E_{f_0}\{\max_{1\leq j\leq s}\int_{\substack{f\in\mathcal{F}_n \\ d_V(f,f_0)\geq 4\delta}} \|f-f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-n\lambda J(f)/2)d\Pi(f)(1-\widetilde{\phi}_{j,n})\} \\
\leq\ & \sum_{j=1}^{s}\int_{\substack{f\in\mathcal{F}_n \\ d_V(f,f_0)\geq 4\delta}} \|f-f_0\|^a E_{f_0}\{\prod_{i\in I_j}(p_f/p_{f_0})(Z_i)(1-\widetilde{\phi}_{j,n})\}d\Pi(f) \\
=\ & \sum_{j=1}^{s}\int_{\substack{f\in\mathcal{F}_n \\ d_V(f,f_0)\geq 4\delta}} \|f-f_0\|^a E_f\{1-\widetilde{\phi}_{j,n}\}d\Pi(f) \\
\leq\ & s\exp(-k_0 M^2 nr_n^2/16)C(a,\Pi).
\end{aligned}$$

Therefore,

$$\begin{aligned}
& \max_{1\leq j\leq s}\int_{\substack{f\in\mathcal{F}_n \\ d_V(f,f_0)\geq 4\delta}} \|f-f_0\|^a \prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-n\lambda J(f)/2)d\Pi(f)(1-\widetilde{\phi}_{j,n}) \\
=\ & O_{P_{f_0}}\left(s\exp(-k_0 M^2 nr_n^2/16)\right). \qquad \text{(S.35)}
\end{aligned}$$

Meanwhile, it follows by (S.31) and (S.35) that

$$\begin{aligned}
& \max_{1\le j\le s}\int_{A'_{n1},\|f-f_0\|_\infty\le\varepsilon/2}\|f-f_0\|^a dP(f|\mathbf{D}_j)(1-\widetilde{\phi}_{j,n}) \\
\le\ & \max_{1\le j\le s}\int_{\mathcal{F}_n,d_V(f,f_0)\ge 4\delta}\|f-f_0\|^a dP(f|\mathbf{D}_j)(1-\widetilde{\phi}_{j,n}) \\
\le\ & \frac{\max_{1\le j\le s}\int_{d_V(f,f_0)\ge 4\delta}^{f\in\mathcal{F}_n}\|f-f_0\|^a\prod_{i\in I_j}(p_f/p_{f_0})(Z_i)\exp(-n\lambda J(f)/2)d\Pi(f)(1-\widetilde{\phi}_{j,n})}{\min_{1\le j\le s} I_{j1}} \\
=\ & O_{P_{f_0}}\left(s\exp(-k_0M^2nr_n^2/16+c_2nr_n^2)\right)=O_{P_{f_0}}\left(s\exp(-nr_n^2)\right).
\end{aligned}$$

Choose the constant M to be even bigger so that $c_V(M+J(f_0)^{1/2})+1+c_2<k_0M^2/16$. Similar to (S.32) we get

$$\max_{1\le j\le s}\int_{A'_{n1},\|f-f_0\|_\infty\le\varepsilon/2}\|f-f_0\|^a dP(f|\mathbf{D}_j)\widetilde{\phi}_{j,n}=O_{P_{f_0}}(s^2\exp(-nr_n^2)).$$

Therefore,

$$\max_{1\le j\le s}\int_{A'_{n1},\|f-f_0\|_\infty\le\varepsilon/2}\|f-f_0\|^a dP(f|\mathbf{D}_j)=O_{P_{f_0}}(s^2\exp(-nr_n^2)). \tag{S.36}$$

Together with (S.25), (S.32) and (S.36), we get

$$\begin{aligned}
& \max_{1\le j\le s}\int_{A'_n}\|f-f_0\|^a dP(f|\mathbf{D}_j) \\
\le\ & \max_{1\le j\le s}\int_{A'_{n1}}\|f-f_0\|^a dP(f|\mathbf{D}_j)+\max_{1\le j\le s}\int_{A'_{n2}}\|f-f_0\|^a dP(f|\mathbf{D}_j) \\
\le\ & \max_{1\le j\le s}\int_{A'_{n1},\|f-f_0\|_\infty\le\varepsilon/2}\|f-f_0\|^a dP(f|\mathbf{D}_j)+\max_{1\le j\le s}\int_{\|f-f_0\|_\infty>\varepsilon/2}\|f-f_0\|^a dP(f|\mathbf{D}_j) \\
& +\max_{1\le j\le s}\int_{A'_{n2}}\|f-f_0\|^a dP(f|\mathbf{D}_j) \\
=\ & O_{P_{f_0}}(s^2\exp(-nr_n^2)).
\end{aligned}$$

This completes the proof. ■

Proof [Proof of Lemma 14] For any $f,g\in S^m(\mathbb{I})$ with $\|f-g\|_\infty\le\varepsilon$, define $\Delta_Z(f,g)=\frac{1}{2}[Y(f(X)-g(X))-f(X)^2/2+g(X)^2/2]$, where recall and $Z=(Y,X)$. It is easy to see by direct calculations that $d_H^2(P_f,P_g)=1-E_g\{\exp(\Delta_Z(f,g))\}$. By Taylor's expansion, for some random $t\in[0,1]$,

$$\begin{aligned}
& 1-E_g\{\exp(\Delta_Z(f,g))\} \\
=\ & -E_g\{\Delta_Z(f,g)\}-\frac{1}{2}E_g\{\Delta_Z(f,g)^2\}-\frac{1}{6}E_g\{\exp(t\Delta_Z(f,g))\Delta_Z(f,g)^3\}.
\end{aligned}$$

We will analyze the terms on the right side of the equation.

Define $\xi=Y-\dot{A}(g(X))$. By Morris et al. (1983) we get $E_g\{\xi|X\}=0$ and $E_g\{\xi^2|X\}=1$. By Taylor's expansion, $\Delta_Z(f,g)=\frac{1}{2}[\xi(f(X)-g(X))-\frac{1}{2}(f(X)-g(X))^2$. Then we get that $-E_g\{\Delta_Z(f,g)\}=\frac{1}{4}V(f-g)$ and

$$\begin{aligned}
E_g\{\Delta_Z(f,g)^2\} &= E_g\{(\frac{1}{2}\xi(f(X)-g(X))-\frac{1}{4}(f(X)-g(X))^2)\} \\
&= \frac{1}{4}E_g\{\xi^2(f(X)-g(X))^2\}-\frac{1}{4}E_g\{\xi(f(X)-g(X))^3\}+\frac{1}{16}E_g\{(f(X)-g(X))^4\} \\
&= \frac{1}{4}V(f-g)+\frac{1}{16}E_g\{(f(X)-g(X))^4\}.
\end{aligned}$$

Since $\|f-g\|_\infty\le\varepsilon<1$ and $|\Delta_Z(f,g)|\le\frac{1}{2}(|\xi|+1/2)|f(X)-g(X)|$, we get

$$\begin{aligned}
& |E_g\{\exp(t\Delta_Z(f,g))\Delta_Z(f,g)^3\}| \\
\le\ & E_g\{\exp(|\Delta_Z(f,g)|)|\Delta_Z(f,g)|^3\} \\
\le\ & E_g\{\exp(\varepsilon|\xi|/2+\varepsilon/4)(|\xi|/2+1/4)^3|f(X)-g(X)|^3\} \\
=\ & 6E_g\left\{\exp(\varepsilon|\xi|/2+\varepsilon/4)\times\frac{1}{3!}\left(|\xi|/2+1/4\right)^3|f(X)-g(X)|^3\right\} \\
\le\ & 6E_g\{\exp(\varepsilon|\xi|/2+\varepsilon/4)\exp(|\xi|/2+1/4)|f(X)-g(X)|^3\} \\
\le\ & 6\exp(\varepsilon/4+/4)E_g\{\exp(|\xi|)|f(X)-g(X)|^3\} \\
\le\ & 6\varepsilon\exp(1/2)C_\epsilon V(f-g).
\end{aligned}$$

It also holds that $|E_g\{(f(X)-g(X))^4\}|\le\varepsilon^2V(f-g)$. Therefore, for any $f,g\in S^m(\mathbb{I})$ with $\|f-g\|_\infty\le\varepsilon$,

$$\begin{aligned}
& |d_H^2(P_f,P_g)-V(f-g)/8| \\
=\ & |\frac{1}{32}E_g\{(f(X)-g(X))^4\}+\frac{1}{6}E_g\{\exp(t\Delta_Z(f,g))\Delta_Z(f,g)^3\}| \\
\le\ & \left(\varepsilon C_\epsilon\exp(1/2)+\varepsilon^2/32\right)V(f-g)<V(f-g)/16,
\end{aligned}$$

which implies $V(f-g)/16\le d_H^2(P_f,P_g)\le 3V(f-g)/16$. This proves Lemma 14. ■

S.8.6. Additional Plots in Section 5

Radius of the credible sets/intervals

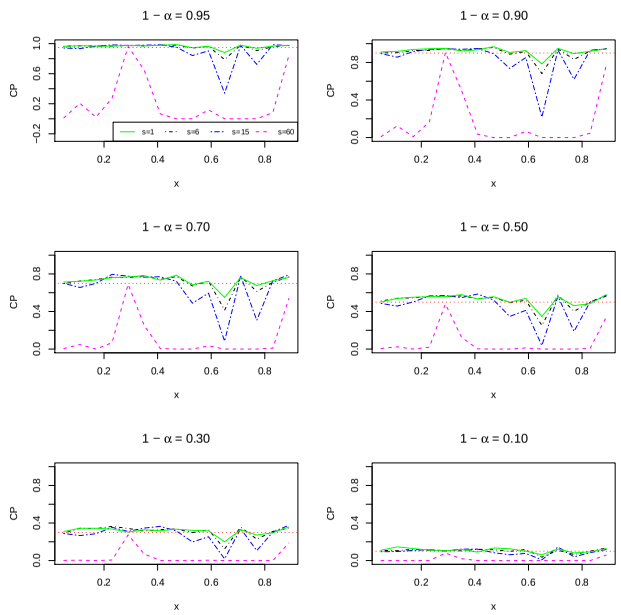


Figure 11. *CP of $F_x(f) = f(x)$ against x based on asymptotic theory.*

Results on larger N

Simulation results about credible regions/intervals in Section 5 are based on $N = 1200$. This section repeated the same study for $N = 1800, 2400$. Results are summarized in following plots.

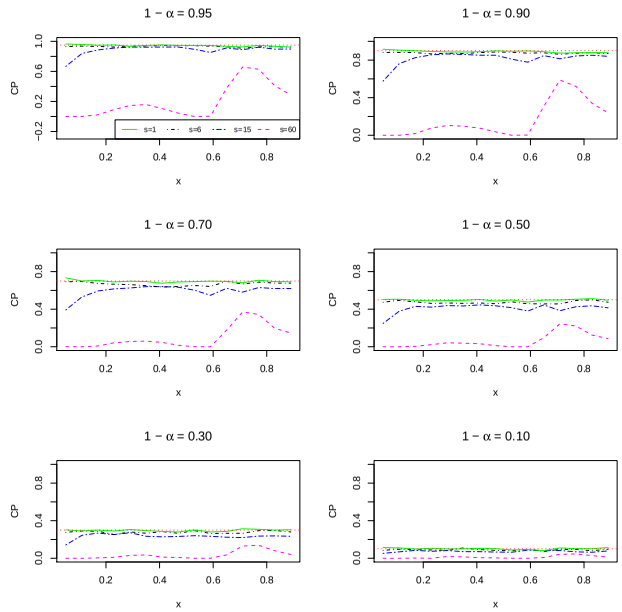


Figure 12. *CP of $F_x(f) = \int_0^x f(z)dz$ against x based on asymptotic theory.*

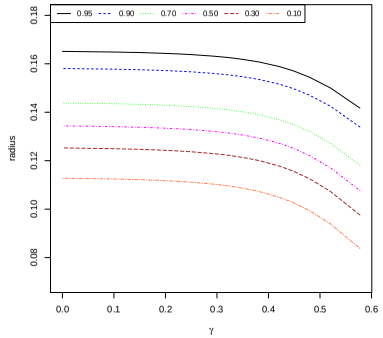


Figure 13. *Radius of credible region (32) against γ. Legend indicates the credibility levels $1 - \alpha$.*

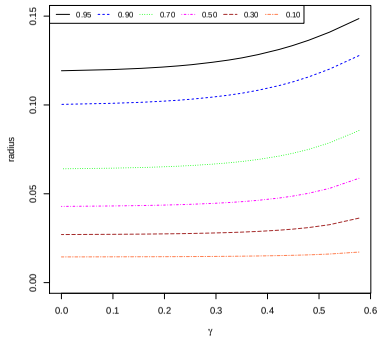


Figure 14. *Radius of credible region (33) against γ. Legend indicates the credibility levels $1-\alpha$.*

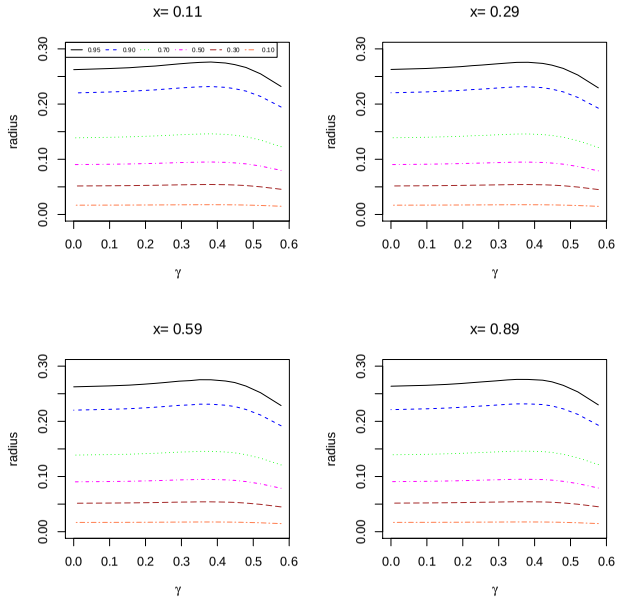


Figure 15. *Radius of credible interval (36) for pointwise functional $F_x(f) = f(x)$ against γ. Legend indicates the credibility levels $1-\alpha$. Four values of x are considered.*

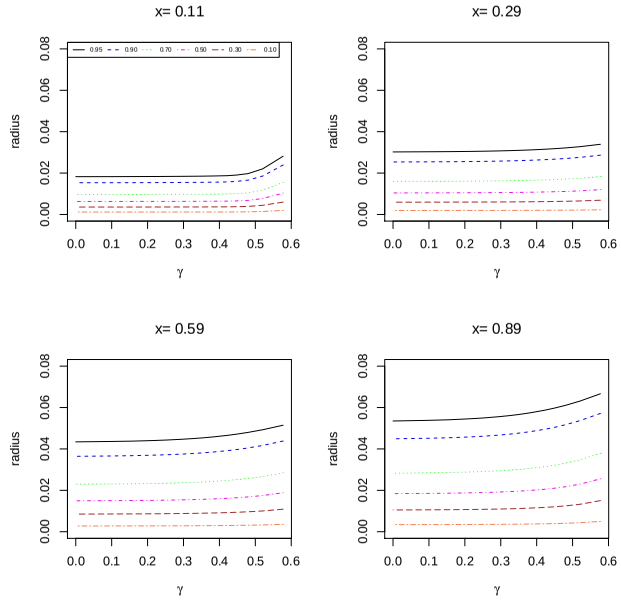


Figure 16. *Radius of credible interval (36) for integral functional $F_x(f) = \int_0^x f(z)dz$ against γ. Legend indicates the credibility levels $1-\alpha$. Four values of x are considered.*

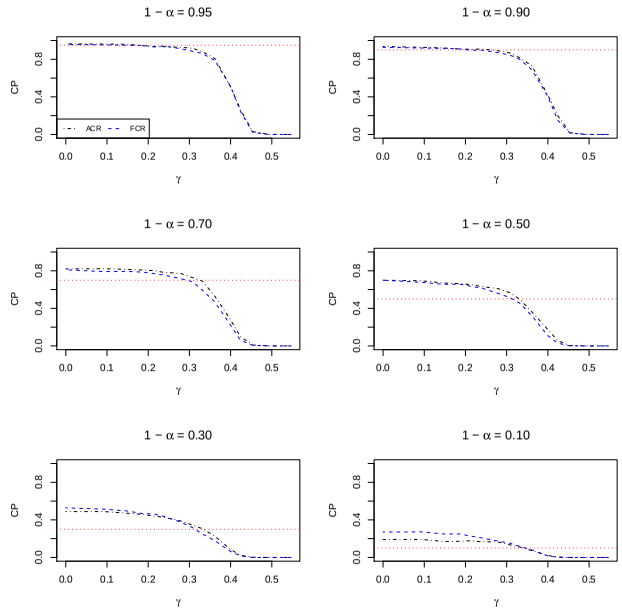


Figure 17. *$N = 1800$: CP of ACR and FCR based on strong topology.*

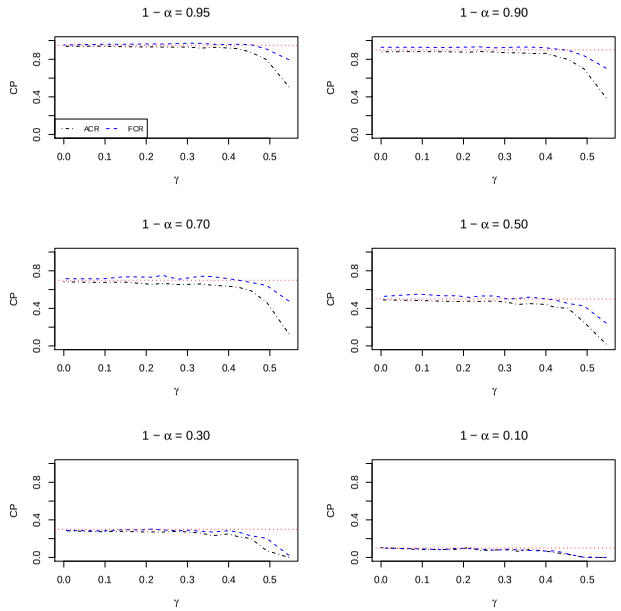


Figure 18. $N = 1800$: *CP of ACR and FCR based on weak topology.*

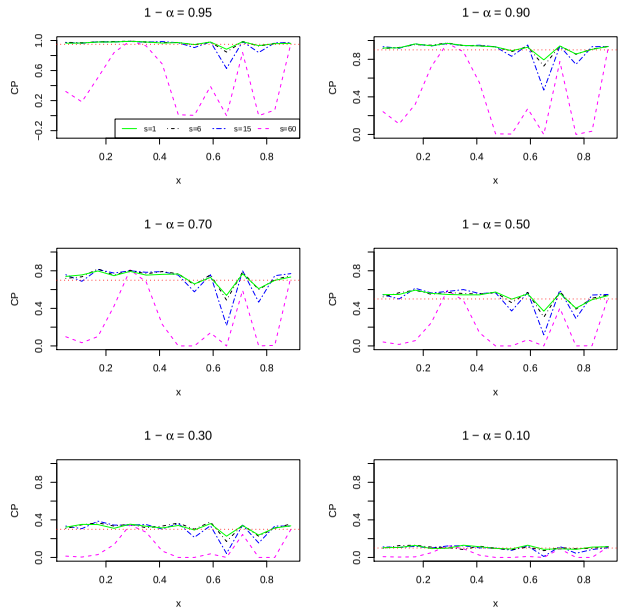


Figure 19. $N = 1800$: *CP of* $F_x(f) = f(x)$ *against* x *based on posterior samples of* f.

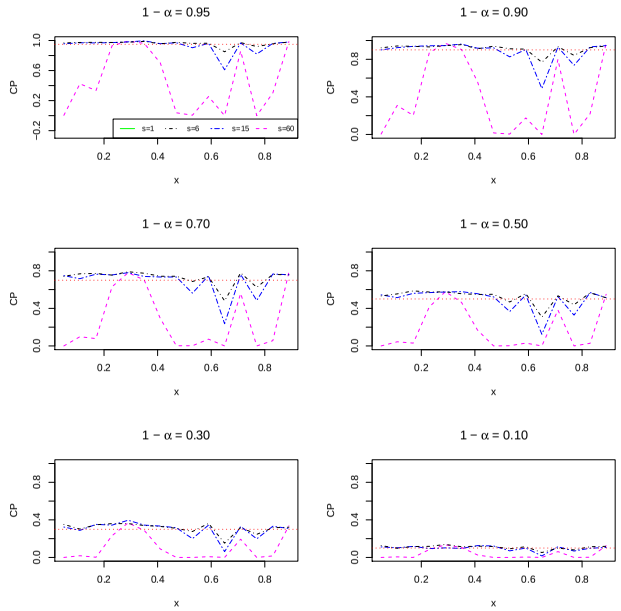


Figure 20. *$N = 1800$: CP of $F_x(f) = f(x)$ against x based on asymptotic theory.*

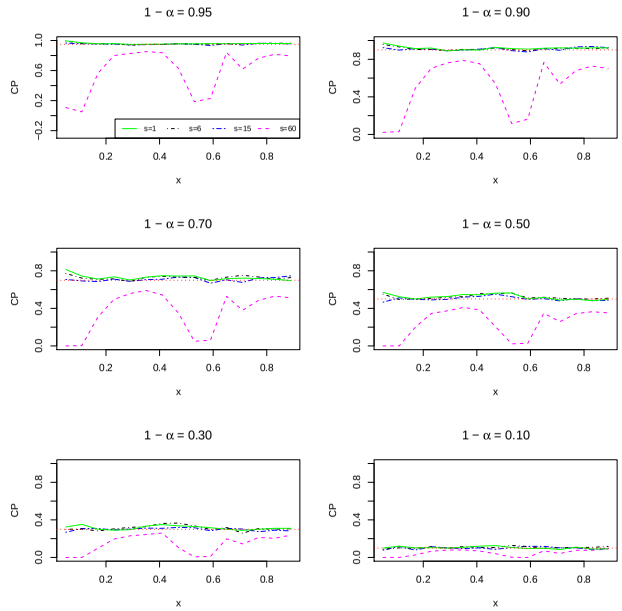


Figure 21. *$N = 1800$: CP of $F_x(f) = \int_0^x f(z)dz$ against x based on posterior samples of f.*

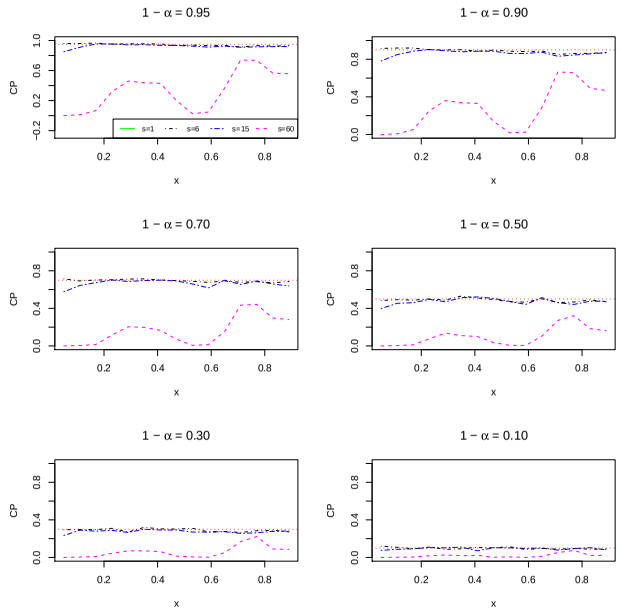


Figure 22. $N = 1800$*: CP of* $F_x(f) = \int_0^x f(z)dz$ *against* x *based on asymptotic theory.*

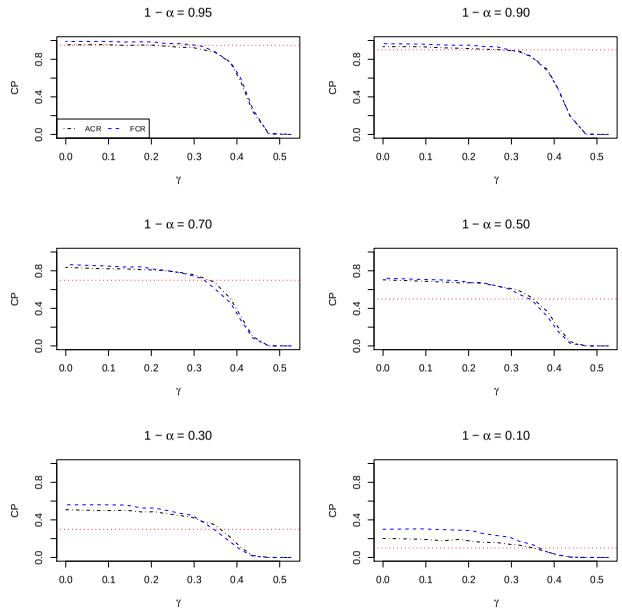


Figure 23. $N = 2400$*: CP of ACR and FCR based on strong topology.*

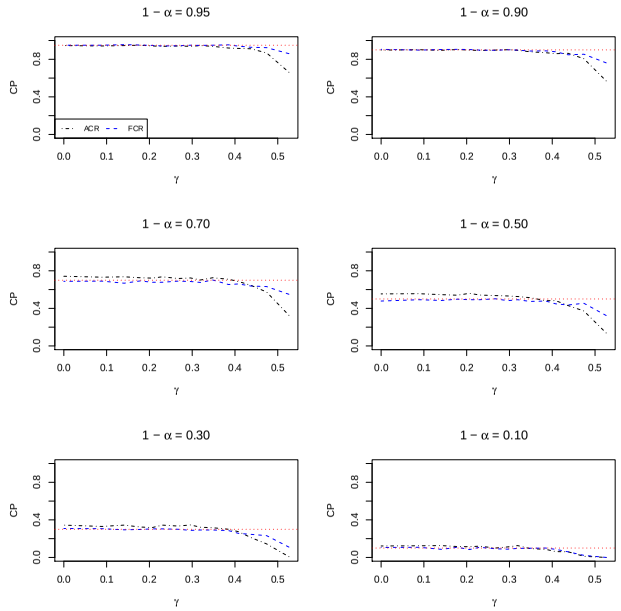


Figure 24. $N = 2400$: *CP of ACR and FCR based on weak topology.*

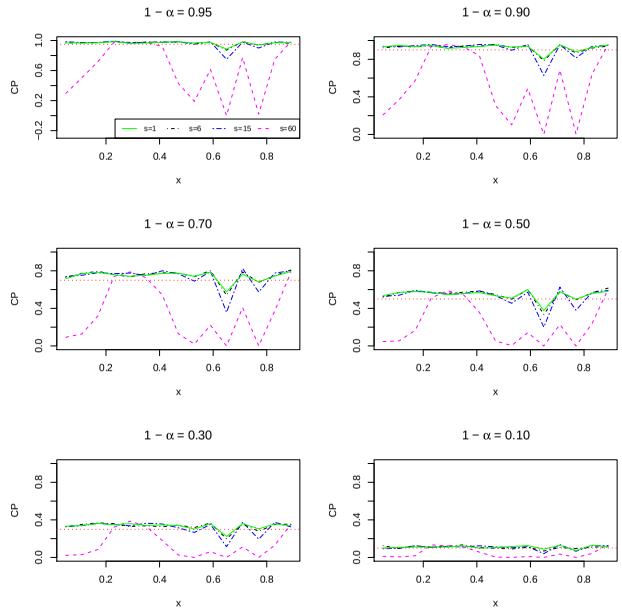


Figure 25. $N = 2400$: *CP of* $F_x(f) = f(x)$ *against* x *based on posterior samples of* f.

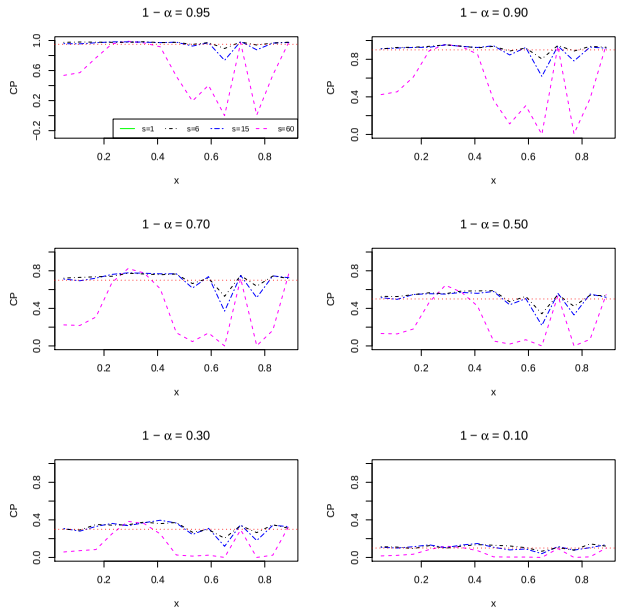


Figure 26. $N = 2400$: *CP of* $F_x(f) = f(x)$ *against* x *based on asymptotic theory.*

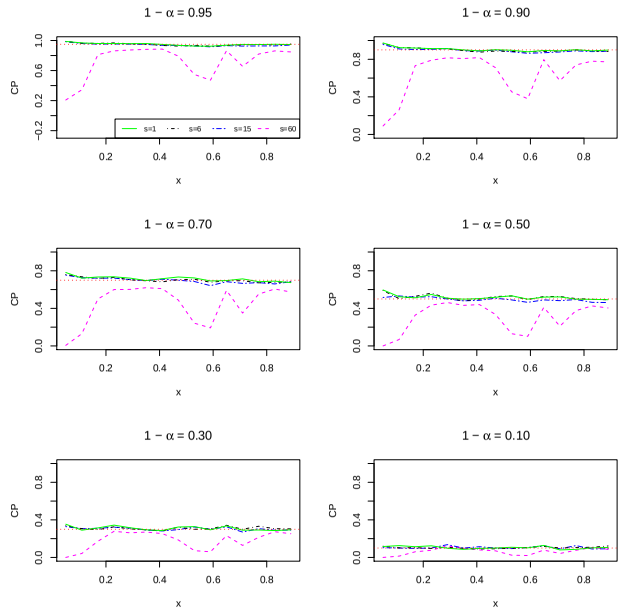


Figure 27. $N = 2400$: *CP of* $F_x(f) = \int_0^x f(z)dz$ *against* x *based on posterior samples of* f.

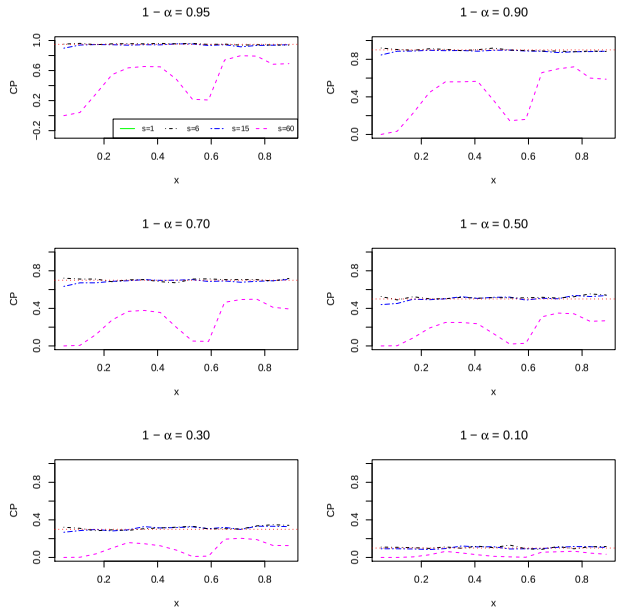


Figure 28. $N = 2400$*: CP of* $F_x(f) = \int_0^x f(z)dz$ *against* x *based on asymptotic theory.*

Journal of Machine Learning Research 20 (2019) 1-43 Submitted 12/17; Revised 5/19; Published 9/19

Provably Accurate Double-Sparse Coding

Thanh V. Nguyen THANHNG@IASTATE.EDU
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011, USA

Raymond K. W. Wong RAYWONG@STAT.TAMU.EDU
Department of Statistics
Texas A&M University
College Station, TX 77843, USA

Chinmay Hegde CHINMAY@IASTATE.EDU
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011, USA

Editor: Animashree Anandkumar

Abstract

Sparse coding is a crucial subroutine in algorithms for various signal processing, deep learning, and other machine learning applications. The central goal is to learn an overcomplete dictionary that can sparsely represent a given input dataset. However, a key challenge is that storage, transmission, and processing of the learned dictionary can be untenably high if the data dimension is high. In this paper, we consider the double-sparsity model introduced by Rubinstein et al. (2010b) where the dictionary itself is the product of a fixed, known basis and a data-adaptive sparse component. First, we introduce a simple algorithm for double-sparse coding that can be amenable to efficient implementation via neural architectures. Second, we theoretically analyze its performance and demonstrate asymptotic sample complexity and running time benefits over existing (provable) approaches for sparse coding. To our knowledge, our work introduces the first computationally efficient algorithm for double-sparse coding that enjoys rigorous statistical guarantees. Finally, we corroborate our theory with several numerical experiments on simulated data, suggesting that our method may be useful for problem sizes encountered in practice.

Keywords: Sparse coding, provable algorithms, unsupervised learning

1. Introduction

1.1. Motivation

Representing signals as sparse linear combinations of atoms from a dictionary is a popular approach in many domains. In this paper, we study the problem of *dictionary learning* (also known as sparse coding), where the goal is to learn an efficient basis (dictionary) that represents the underlying class of signals well. In the typical sparse coding setup, the dictionary is *overcomplete* (i.e., the cardinality of the dictionary exceeds the ambient signal dimension) while the representation is *sparse* (i.e., each signal is encoded by a combination of only a few dictionary atoms.)

©2019 Thanh V. Nguyen, Raymond K. W. Wong, and Chinmay Hegde.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-728.html.

Sparse coding has a rich history in diverse fields such as signal processing, machine learning, and computational neuroscience. Discovering optimal basis representations of data is a central focus of image analysis (Krim et al., 1999; Elad and Aharon, 2006; Rubinstein et al., 2010a), and dictionary learning has proven widely successful in imaging problems such as denoising, deconvolution, inpainting, and compressive sensing (Elad and Aharon, 2006; Candes and Tao, 2005; Rubinstein et al., 2010a). Sparse coding approaches have also been used as a core building block of deep learning systems for prediction (Gregor and LeCun, 2010; Boureau et al., 2010) and associative memory (Mazumdar and Rawat, 2017). Interestingly, the seminal work by Olshausen and Field (1997) has shown intimate connections between sparse coding and neuroscience: the dictionaries learned from image patches of natural scenes bear remarkable resemblance to spatial receptive fields observed in the mammalian primary visual cortex.

From a mathematical standpoint, the sparse coding problem is formulated as follows. Given p data samples $Y = [y^{(1)}, y^{(2)}, \ldots, y^{(p)}] \in \mathbb{R}^{n\times p}$, the goal is to find a dictionary $D \in \mathbb{R}^{n\times m}$ $(m > n)$ and corresponding sparse code vectors $X = [x^{(1)}, x^{(2)}, \ldots, x^{(p)}] \in \mathbb{R}^{m\times p}$ such that the representation DX fits the data samples as well as possible. Typically, one obtains the dictionary and the code vectors as the solution to the following optimization problem:

$$\begin{aligned} \min_{D,X} \mathcal{L}(D,X) &= \frac{1}{2}\sum_{j=1}^{p} \|y^{(j)} - Dx^{(j)}\|_2^2, \\ \text{s.t. } \sum_{j=1}^{p} \mathcal{S}(x^{(j)}) &\le S \end{aligned} \tag{1}$$

where $\mathcal{S}(\cdot)$ is some sparsity-inducing penalty function on the code vectors, such as the ℓ_1-norm. The objective function $\mathcal{L}$ controls the reconstruction error while the constraint enforces the sparsity of the representation. However, even a cursory attempt at solving the optimization problem (1) reveals the following obstacles:

The constrained optimization problem (1) involves a non-convex (in fact, bilinear) objective function, as well as potentially non-convex constraints depending on the choice of the sparsity-promoting function $\mathcal{S}$ (for example, the ℓ_0 function.) Hence, obtaining *provably* correct algorithms for this problem can be challenging. Indeed, the vast majority of practical approaches for sparse coding have been heuristics (Engan et al., 1999; Aharon et al., 2006; Mairal et al., 2009). Recent works in the theoretical machine learning community have bucked this trend, providing provably accurate algorithms if certain assumptions are satisfied (Spielman et al., 2012; Agarwal et al., 2014; Arora et al., 2014a, 2015; Sun et al., 2015; Błasiok and Nelson, 2016; law Adamczak, 2016; Chatterji and Bartlett, 2017). However, relatively few of these newer methods have been shown to provide good empirical performance in actual sparse coding problems.

Even if theoretical correctness issues were to be set aside, and we are somehow able to efficiently learn sparse codes of the input data, we often find that applications using such learned sparse codes encounter *memory* and *running-time* issues. Indeed, in the overcomplete case, the storage of the learned dictionary D incurs $mn = \Omega(n^2)$ memory cost, which is prohibitive when n is large. Therefore, in practical applications (such as image analysis)

one typically resorts to chop the data into smaller blocks (e.g., partitioning image data into patches) to make the problem manageable.

A related line of research has been devoted to learning dictionaries that obey some type of *structure*. Such structural information can be leveraged to incorporate prior knowledge of underlying signals as well as to resolve computational challenges due to the data dimension. For instance, the dictionary is assumed to be separable, or obey a convolutional structure. One such variant is the *double-sparse* coding problem (Rubinstein et al., 2010b; Sulam et al., 2016) where the dictionary D *itself* exhibits a sparse structure. To be specific, the dictionary is expressed as:

$$D = \Phi A,$$

i.e., it is composed of a known "base dictionary" $\Phi \in \mathbb{R}^{n\times n}$, and a learned "synthesis" matrix $A \in \mathbb{R}^{n\times m}$ whose columns are sparse. The base dictionary Φ is typically any fixed basis chosen according to domain knowledge, while the synthesis matrix A is column-wise sparse and is to be learned from the data. The basis Φ is typically orthonormal (such as the canonical or wavelet basis); however, there are cases where the base dictionary Φ is overcomplete (Rubinstein et al., 2010b; Sulam et al., 2016).

There are several reasons why such the double-sparsity model can be useful. First, the double-sparsity assumption is rather appealing from a conceptual standpoint, since it lets us combine the knowledge of decades of modeling efforts in harmonic analysis with the flexibility of learning new representations tailored to specific data families. Moreover, such a double-sparsity model has computational benefits. If the columns of A are (say) r-sparse (i.e., each column contains no more than $r \ll n$ non-zeroes) then the overall burden of storing, transmitting, and computing with A is much lower than that for general unstructured dictionaries. Finally, such a model lends itself well to *interpretable* learned features if the atoms of the base dictionary are semantically meaningful.

All the above reasons have spurred researchers to develop a series of algorithms to learn doubly-sparse codes (Rubinstein et al., 2010b; Sulam et al., 2016). However, despite their empirical promise, no theoretical analysis of their performance have been reported in the literature and to date, we are unaware of a provably accurate, polynomial-time algorithm for the double-sparse coding problem. Our goal in this paper is precisely to fill this gap.

1.2. Our Contributions

In this paper, we provide a new framework for double-sparse coding. To the best of our knowledge, our approach is the first method that enjoys *provable* statistical and algorithmic guarantees for this problem. In addition, our approach enjoys two benefits: we demonstrate that the method is *neurally plausible* (i.e., its execution can plausibly be achieved using a neural network architecture) and *robust* to noise.

Inspired by the aforementioned recent theoretical advances in sparse coding, we assume a learning-theoretic setup where the data samples arise from a ground-truth generative model. Informally, suppose there exists a true (but unknown) synthesis matrix A^* that is column-wise r-sparse, and the ith data sample is generated as:

$$y^{(i)} = \Phi A^* x^{*(i)} + \text{ noise}, \quad i = 1, 2, \ldots, p$$

where the code vector $x^{*(i)}$ is independently drawn from a distribution supported on the set of k-sparse vectors. We desire to learn the underlying matrix A^*. Informally, suppose that the synthesis matrix A^* is *incoherent* (the columns of A^* are sufficiently close to orthogonal) and has bounded spectral norm. Finally, suppose that the number of dictionary elements, m, is at most a constant multiple of n. All of these assumptions are standard[1].

We will demonstrate that the true synthesis matrix A^* can be recovered (with small error) in a tractable manner as sufficiently many samples are provided. Specifically, we make the following novel contributions:

1. We propose a new algorithm that produces a coarse estimate of the synthesis matrix that is sufficiently close to the ground truth A^*. Our algorithm builds upon spectral initialization-based ideas that have recently gained popularity in non-convex machine learning (Zhang et al., 2016; Wang et al., 2016).

2. Given the above coarse estimate of the synthesis matrix A^*, we propose a descent-style algorithm to refine the above estimate of A^*. This algorithm is simpler than previously studied double-sparse coding algorithms (such as the Trainlets approach of Sulam et al. (2016)), while still giving good statistical performance. Moreover, this algorithm can be realized in a manner amenable to neural implementations.

3. We provide a rigorous analysis of both algorithms. Put together, our analysis produces the first provably polynomial-time algorithm for double-sparse coding. We show that the algorithm provably returns a good estimate of the ground-truth; in particular, in the absence of noise we prove that $\Omega(mr \text{ polylog } n)$ samples are sufficient for a good enough initialization in the first algorithm, as well as guaranteed linear convergence of the descent phase up to a precise error parameter that can be interpreted as the radius of convergence.

   Indeed, our analysis shows that employing the double-sparsity model helps in this context, and leads to a strict improvement in sample complexity, as well as running time over previous rigorous methods for (regular) sparse coding such as Arora et al. (2015).

4. We also analyze our approach in a more realistic setting with the presence of additive noise and demonstrate its stability. We prove that $\Omega(mr \text{ polylog } n)$ samples are sufficient to obtain a good enough estimate in the initialization, and also to obtain guaranteed linear convergence during descent to provably recover A^*.

5. We underline the benefit of the double-sparse structure over the regular model by analyzing the algorithms in Arora et al. (2015) under the noisy setting. As a result, we obtain the sample complexity $O\big((mk + \sigma_\varepsilon^2 \frac{mn^2}{k})\text{polylog } n\big)$, which demonstrates a negative effect of noise on this approach.

6. We rigorously develop a hard thresholding intialization that extends the spectral scheme in Arora et al. (2015). Additionally, we provide more results for the case where A is orthonormal, sparse dictionary to relax the condition on r, which may be of independent interest.

1. We clarify both the data and the noise model more concretely in Section 2 below.

Setting	Reference	Sample (w/o noise)	Sample (w/ noise)	Time	Expt
Regular	MOD (Engan et al., 1999)	✗	✗	✗	✓
	K-SVD (Aharon et al., 2006)	✗	✗	✗	✓
	Spielman et al. (2012)	$O(n^2 \log n)$	✗	$\widetilde{\Omega}(n^4)$	✓
	Arora et al. (2014b)	$\widetilde{O}(m^2/k^2)$	✗	$\widetilde{O}(np^2)$	✗
	Gribonval et al. (2015a)	$O(nm^3)$	$O(nm^3)$	✗	✗
	Arora et al. (2015)	$\widetilde{O}(mk)$	✗	$\widetilde{O}(mn^2p)$	✗
Double Sparse	Double Sparsity (Rubinstein et al., 2010b)	✗	✗	✗	✓
	Gribonval et al. (2015b)	$\widetilde{O}(mr)$	$\widetilde{O}(mr)$	✗	✗
	Trainlets (Sulam et al., 2016)	✗	✗	✗	✓
	This paper	$\widetilde{O}(mr)$	$\widetilde{O}(mr + \sigma_\varepsilon^2 \frac{mnr}{k})$	$\widetilde{O}(mnp)$	✓

Table 1: *Comparison of various sparse coding techniques. Expt: whether numerical experiments have been conducted. ✗ in all other columns indicates no provable guarantees. Here, n is the signal dimension, and m is the number of atoms. The sparsity levels for A and x are r and k respectively, and p is the sample size.*

7. While our analysis mainly consists of sufficiency results and involves unknown constants hidden in big-O notation, we demonstrate our findings by reporting a suite of numerical experiments on synthetic test datasets.

Overall, our approach results in strict improvement in sample complexity, as well as running time, over previous rigorously analyzed methods for (regular) sparse coding, such as Arora et al. (2015). See Table 1 for a detailed comparison.

1.3. Techniques

At a high level, our method is an adaptation of the seminal approach of Arora et al. (2015). As is common in the statistical learning literature, we assume a "ground-truth" generative model for the observed data samples, and attempt to estimate the parameters of the generative model given a sufficient number of samples. In our case, the parameters correspond to the synthesis matrix A^*, which is column-wise r-sparse. The natural approach is to formulate a loss function in terms of A such as Equation (1), and perform gradient descent with respect to the surface of the loss function to learn A^*.

The key challenge in sparse coding is that the gradient is inherently coupled with the *codes* of the training samples (i.e., the columns of X^*), which are unknown *a priori*. However, the main insight of Arora et al. (2015) is that within a small enough neighborhood of A^*, a noisy version of X^* can be estimated, and therefore the overall method is similar to performing *approximate gradient descent*. Formulating the actual algorithm as a noisy variant of approximate gradient descent allows us to overcome the finite-sample variability

of the loss, and obtain a descent property directly related to (the population parameter) A^*.

The second stage of our approach (i.e., our descent-style algorithm) leverages this intuition. However, instead of standard gradient descent, we perform approximate *projected* gradient descent, such that the column-wise r-sparsity property is enforced in each new estimate of A^*. Indeed, such an extra projection step is critical in showing a sample complexity improvement over the existing approach of Arora et al. (2015).The key novelty is in figuring out how to perform the projection in each gradient iteration. For this purpose, we develop a novel initialization algorithm that identifies the locations of the non-zeroes in A^* even before commencing the descent phase. This is nontrivially different from initialization schemes used in previous rigorous methods for sparse coding, and the analysis is somewhat more involved.

In Arora et al. (2015), (the principal eigenvector of) a weighted covariance matrix of y (estimated by the weighted average of outer products $y_i y_i^T$) is shown to provide a coarse estimate of a dictionary atom. We extend this idea and rigoriously show that the diagonal of the weighted covariance matrix serves as a good indicator of the support of a column in A^*. The success relies on the concentration of the diagonal vector with dimension n, instead of the covariance matrix with dimensions $n \times n$. With the support selected, our scheme only utilizes a reduced weighted covariance matrix with dimensions at most $r \times r$. This initialization scheme enables us to effectively reduce the dimension of the problem, and therefore leads to significant improvement in sample complexity and running time over previous (provable) sparse coding methods when the data representation sparsity k is much smaller than m.

Further, we rigorously analyze the proposed algorithms in the presence of noise with a bounded expected norm. Our analysis shows that our method is stable, and in the case of i.i.d. Gaussian noise with bounded expected ℓ_2-norms, is at least a polynomial factor better than previous polynomial time algorithms for sparse coding.

The empirical performance of our proposed method is demonstrated by a suite of numerical experiments on synthetic datasets. In particular, we show that our proposed methods are simple and practical, and improve upon previous provable algorithms for sparse coding.

1.4. Paper Organization

The remainder of this paper is organized as follows. Section 2 introduces notation, key model assumptions, and informal statements of our main theoretical results. Section 3 outlines our initialization algorithm (along with supporting theoretical results) while Section 4 presents our descent algorithm (along with supporting theoretical results). Section 5 provides a numerical study of the efficiency of our proposed algorithms, and compares it with previously proposed methods. Finally, Section 6 concludes with a short discussion. All technical proofs are relegated to the appendix.

2. Setup and Main Results

2.1. Notation

We define $[m] \triangleq \{1, \ldots, m\}$ for any integer $m > 1$. For any vector $x = [x_1, x_2, \ldots, x_m]^T \in \mathbb{R}^m$, we write $\text{supp}(x) \triangleq \{i \in [m] : x_i \neq 0\}$ as the support set of x. Given any subset $S \subseteq [m]$, x_S corresponds to the sub-vector of x indexed by the elements of S. For any matrix $A \in \mathbb{R}^{n \times m}$, we use $A_{\bullet i}$ and $A_{j\bullet}^T$ to represent the i-th column and the j-th row respectively. For some appropriate sets R and S, let $A_{R\bullet}$ (respectively, $A_{\bullet S}$) be the submatrix of A with rows (respectively columns) indexed by the elements in R (respectively S). In addition, for the i-th column $A_{\bullet i}$, we use $A_{R,i}$ to denote the sub-vector indexed by the elements of R. For notational simplicity, we use $A_{R\bullet}^T$ to indicate $(A_{R\bullet})^T$, the tranpose of A after a row selection. Besides, we use $\circ$ and $\text{sgn}(\cdot)$ to represent the element-wise Hadamard operator and the element-wise sign function respectively. Further, $\text{threshold}_K(x)$ is a thresholding operator that replaces any elements of x with magnitude less than K by zero.

The ℓ_2-norm $\|x\|$ for a vector x and the spectral norm $\|A\|$ for a matrix A appear several times. In some cases, we also utilize the Frobenius norm $\|A\|_F$ and the operator norm $\|A\|_{1,2} \triangleq \max_{\|x\|_1 \leq 1} \|Ax\|$. The norm $\|A\|_{1,2}$ is essentially the maximal Euclidean norm of any column of A.

For clarity, we adopt asymptotic notations extensively. We write $f(n) = O(g(n))$ (or $f(n) = \Omega(g(n))$) if $f(n)$ is upper bounded (respectively, lower bounded) by $g(n)$ up to some positive constant. Next, $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. Also $\widetilde{\Omega}$ and $\widetilde{O}$ represent Ω and O up to a multiplicative poly-logarithmic factor respectively. Finally $f(n) = o(g(n))$ (or $f(n) = \omega(g(n))$) if $\lim_{n\to\infty} |f(n)/g(n)| = 0$ ($\lim_{n\to\infty} |f(n)/g(n)| = \infty$).

Throughout the paper, we use the phrase "with high probability" (abbreviated to w.h.p.) to describe an event with failure probability of order at most $n^{-\omega(1)}$. In addition, $g(n) = O^*(f(n))$ means $g(n) \leq Kf(n)$ for some small enough constant K.

2.2. Generative Model of Data

Suppose that the observed samples are given by

$$y^{(i)} = Dx^{*(i)} + \varepsilon, \quad i = 1, \ldots, p,$$

i.e., we are given p samples of y generated from a fixed (but unknown) dictionary D where the sparse code x^* and the error ε are drawn from a joint distribution $\mathcal{D}$ specified below. In the double-sparse setting, the dictionary is assumed to follow a decomposition $D = \Phi A^*$, where $\Phi \in \mathbb{R}^{n \times n}$ is a known *orthonormal* basis matrix and A^* is an unknown, ground truth synthesis matrix. An alternative (and interesting) setting is an overcomplete Φ with a square A^*, which our analysis below does not cover; we defer this to future work. Our approach relies upon the following assumptions on the synthesis dictionary A^*:

A1 A^* is overcomplete (i.e., $m \geq n$) with $m = O(n)$.

A2 A^* is μ-incoherent, i.e., for all $i \neq j$, $|\langle A_{\bullet i}^*, A_{\bullet j}^* \rangle| \leq \mu/\sqrt{n}$.

A3 $A_{\bullet i}^*$ has at most r non-zero elements, and is normalized such that $\|A_{\bullet i}^*\| = 1$ for all i. Moreover, $|A_{ij}^*| \geq \tau$ for $A_{ij}^* \neq 0$ and $\tau = \Omega(1/\sqrt{r})$.

A4 A^* has bounded spectral norm such that $\|A^*\| \leq O(\sqrt{m/n})$.

All these assumptions are standard. In Assumption **A2**, the incoherence μ is typically of order $O(\log n)$ with high probability for a normal random matrix (Arora et al., 2014b). Assumption **A3** is a common assumption in sparse signal recovery. The bounded spectral norm assumption is also standard (Arora et al., 2015). In addition to Assumptions **A1-A4**, we make the following distributional assumptions on $\mathcal{D}$:

B1 Support $S = \text{supp}(x^*)$ is of size at most k and uniformly drawn without replacement from $[m]$ such that $\mathbb{P}[i \in S] = \Theta(k/m)$ and $\mathbb{P}[i, j \in S] = \Theta(k^2/m^2)$ for some $i, j \in [m]$ and $i \neq j$.

B2 The nonzero entries x_S^* are pairwise independent and sub-Gaussian given the support S with $\mathbb{E}[x_i^* | i \in S] = 0$ and $\mathbb{E}[x_i^{*2} | i \in S] = 1$.

B3 For $i \in S$, $|x_i^*| \geq C$ where $0 < C \leq 1$.

B4 The additive noise ε has i.i.d. Gaussian entries with variance σ_ε^2 with $\sigma_\varepsilon = O(1/\sqrt{n})$.

For the rest of the paper, we set $\Phi = I_n$, the identity matrix of size n. This only simplifies the arguments but does not change the problem because one can study an equivalent model:

$$y' = Ax^* + \varepsilon',$$

where $y' = \Phi^T y$ and $\varepsilon' = \Phi^T \varepsilon$, as $\Phi^T \Phi = I_n$. Due to the Gaussianity of ε, ε' also has independent entries. Although this property is specific to Gaussian noise, all the analysis carried out below can be extended to sub-Gaussian noise with minor (but rather tedious) changes in concentration arguments.

Our goal is to devise an algorithm that produces a provably "good" estimate of A^*. For this, we need to define a suitable measure of "goodness". We use the following notion of distance that measures the maximal column-wise difference in ℓ_2-norm under some suitable transformation.

Definition 1 ((δ, κ)-nearness) *A is said to be δ-close to A^* if there is a permutation $\pi : [m] \to [m]$ and a sign flip $\sigma : [m] : \{\pm 1\}$ such that $\|\sigma(i) A_{\bullet \pi(i)} - A_{\bullet i}^*\| \leq \delta$ for every i. In addition, A is said to be (δ, κ)-near to A^* if $\|A_{\bullet \pi} - A^*\| \leq \kappa \|A^*\|$ also holds.*

For notational simplicity, in our theorems we simply replace π and σ in Definition 1 with the identity permutation $\pi(i) = i$ and the positive sign $\sigma(\cdot) = +1$ while keeping in mind that in reality we are referring to one element of the equivalence class of all permutations and sign flip transforms of A^*.

We will also need some technical tools from Arora et al. (2015) to analyze our gradient descent-style method. Consider any iterative algorithm that looks for a desired solution $z^* \in \mathbb{R}^n$ to optimize some function $f(z)$. Suppose that the algorithm produces a sequence of estimates $z^1, \ldots, z^s$ via the update rule:

$$z^{s+1} = z^s - \eta g^s,$$

for some vector g^s and scalar step size η. The goal is to characterize "good" directions g^s such that the sequence converges to z^* under the Euclidean distance. The following gives one such sufficient condition for g^s.

Definition 2 *A vector g^s at the sth iteration is $(\alpha, \beta, \gamma_s)$-correlated with a desired solution z^* if*

$$\langle g^s, z^s - z^* \rangle \geq \alpha \|z^s - z^*\|^2 + \beta \|g^s\|^2 - \gamma_s.$$

We know from convex optimization that if f is 2α-strongly convex and $1/2\beta$-smooth, and g^s is chosen as the gradient $\nabla_z f(z)$, then g^s is $(\alpha, \beta, 0)$-correlated with z^*. In our setting, the desired solution corresponds to A^*, the ground-truth synthesis matrix. In Arora et al. (2015), it is shown that $g^s = \mathbb{E}_y[(A^s x - y)\mathrm{sgn}(x)^T]$, where $x = \mathrm{threshold}_{C/2}((A^s)^T y)$ indeed satisfies Definition 2. This g^s is a population quantity and not explicitly available, but one can estimate such g^s using an empirical average. The corresponding estimator $\widehat{g}^s$ is a random variable, so we also need a related *correlated-with-high-probability* condition:

Definition 3 *A direction $\widehat{g}^s$ at the sth iteration is $(\alpha, \beta, \gamma_s)$-correlated-w.h.p. with a desired solution z^* if, w.h.p.,*

$$\langle \widehat{g}^s, z^s - z^* \rangle \geq \alpha \|z^s - z^*\|^2 + \beta \|\widehat{g}^s\|^2 - \gamma_s.$$

From Definition 2, one can establish a form of descent property in each update step, as shown in Theorem 1.

Theorem 1 *Suppose that g^s satisfies the condition described in Definition 2 for $s = 1, 2, \ldots, T$. Moreover, $0 < \eta \leq 2\beta$ and $\gamma = \max_{s=1}^T \gamma_s$. Then, the following holds for all s:*

$$\|z^{s+1} - z^*\|^2 \leq (1 - 2\alpha\eta)\|z^s - z^*\|^2 + 2\eta\gamma_s.$$

In particular, the above update converges geometrically to z^* with an error γ/α. That is,

$$\|z^{s+1} - z^*\|^2 \leq (1 - 2\alpha\eta)^s \|z^0 - z^*\|^2 + 2\gamma/\alpha.$$

We can obtain a similar result for Definition 3 except that $\|z^{s+1} - z^*\|^2$ is replaced with its expectation.

Armed with the above tools, we now state some informal versions of our main results:

Theorem 2 (Provably correct initialization, informal) *There exists a neurally plausible algorithm to produce an initial estimate A^0 that has the correct support and is $(\delta, 2)$-near to A^* with high probability. Its running time and sample complexity are $\widetilde{O}(mnp)$ and $\widetilde{O}(mr)$ respectively. This algorithm works when the sparsity level satisfies $r = O^*(\log n)$.*

Our algorithm can be regarded as an extension of Arora et al. (2015) to the double-sparse setting. It reconstructs the support of one single column and then estimates its direction in the subspace defined by the support. Our proposed algorithm enjoys neural plausibility by implementing a thresholding non-linearity and Oja's update rule. We provide a neural implementation of our algorithm in Appendix G. The adaption to the sparse structure results in a strict improvement upon the original algorithm both in running time and sample complexity. However, our algorithm is limited to the sparsity level $r = O^*(\log n)$, which is rather small but plausible from the modeling standpoint. For comparison, we analyze a natural extension of the algorithm of Arora et al. (2015) with an extra hard-thresholding

step for every learned atom. We obtain the same order restriction on r, but somewhat worse bounds on sample complexity and running time. The details are found in Appendix F.

We hypothesize that a stronger incoherence assumption can lead to provably correct initialization for a much wider range of r. For purposes of theoretical analysis, we consider the special case of a *perfectly incoherent* synthesis matrix A^* such that $\mu = 0$ and $m = n$. In this case, we can indeed improve the sparsity parameter to $r = O^*\big(\min(\frac{\sqrt{n}}{\log^2 n}, \frac{n}{k^2 \log^2 n})\big)$, which is an exponential improvement. This analysis is given in Appendix E.

The next theorem summarizes our result for the descent algorithm:

Theorem 3 (Provably correct descent, informal) *There exists a neurally plausible algorithm for double-sparse coding that converges to A^* with geometric rate when the initial estimate A^0 has the correct support and $(\delta, 2)$-near to A^*. The running time per iteration is $O(mkp + mrp)$ and the sample complexity is $\widetilde{O}(m + \sigma_\varepsilon^2 \frac{mnr}{k})$.*

Similar to Arora et al. (2015), our proposed algorithm enjoys neural plausibility. Moreover, we can achieve a better running time and sample complexity per iteration than previous methods, particularly in the noisy case. We show in Appendix F that in this regime the sample complexity of Arora et al. (2015) is $\widetilde{O}(m + \sigma_\varepsilon^2 \frac{mn^2}{k})$. For instance, when $\sigma_\varepsilon \asymp n^{-1/2}$, the sample complexity bound is significantly worse than $\widetilde{O}(m)$ in the noiseless case. In contrast, our proposed method leverages the sparse structure to overcome this problem and obtain improved results.

We are now ready to introduce our methods in detail. As discussed above, our approach consists of two stages: an initialization algorithm that produces a coarse estimate of A^*, and a descent-style algorithm that refines this estimate to accurately recover A^*.

3. Stage 1: Initialization

In this section, we present a neurally plausible algorithm that can produce a coarse initial estimate of the ground truth A^*. We give a neural implementation of the algorithm in Appendix G.

Our algorithm is an adaptation from the algorithm in Arora et al. (2015). The idea is to estimate dictionary atoms in a greedy fashion by iteratively re-weighting the given samples. The samples are re-scaled in a way that the weighted (sample) covariance matrix has the dominant first singular value, and its corresponding eigenvector is close to one particular atom with high probability. However, while this algorithm is conceptually very appealing, it incurs severe computational costs in practice. More precisely, the overall running time is $\widetilde{O}(mn^2p)$ in expectation, which is unrealistic for large-scale problems.

To overcome this burden, we leverage the double-sparsity assumption in our generative model to obtain a more efficient approach. The high-level idea is to first estimate the support of each column in the synthesis matrix A^*, and then obtain a coarse estimate of the nonzero coefficients of each column based on knowledge of its support. The key ingredient of our method is a novel spectral procedure that gives us an estimate of the column supports purely from the observed samples. The full algorithm, that we call *Truncated Pairwise Reweighting*, is listed in pseudocode form as Algorithm 1.

Algorithm 1 Truncated Pairwise Reweighting

Initialize $L = \emptyset$
Randomly divide p samples into two disjoint sets $\mathcal{P}_1$ and $\mathcal{P}_2$ of sizes p_1 and p_2 respectively
While $|L| < m$. Pick u and v from $\mathcal{P}_1$ at random
  For every $l = 1, 2, \ldots, n$; compute

$$\widehat{e}_l = \frac{1}{p_2} \sum_{i=1}^{p_2} \langle y^{(i)}, u \rangle \langle y^{(i)}, v \rangle (y_l^{(i)})^2$$

  Sort $(\widehat{e}_1, \widehat{e}_2, \ldots, \widehat{e}_n)$ in descending order
  If $r' \leq r$ s.t $\widehat{e}_{(r')} \geq O(k/mr)$ and $\widehat{e}_{(r'+1)}/\widehat{e}_{(r')} < O^*(r/\log^2 n)$
    Let $\widehat{R}$ be set of the r' largest entries of $\widehat{e}$
    $\widehat{M}_{u,v} = \frac{1}{p_2} \sum_{i=1}^{p_2} \langle y^{(i)}, u \rangle \langle y^{(i)}, v \rangle y^{(i)}_{\widehat{R}} (y^{(i)}_{\widehat{R}})^T$
    $\delta_1, \delta_2 \leftarrow$ top singular values of $\widehat{M}_{u,v}$
    $z_{\widehat{R}} \leftarrow$ top singular vector of $\widehat{M}_{u,v}$
    If $\delta_1 \geq \Omega(k/m)$ and $\delta_2 < O^*(k/m \log n)$
      If $\mathrm{dist}(\pm z, l) > 1/\log n$ for any $l \in L$
        Update $L = L \cup \{z\}$
Return $A^0 = (L_1, \ldots, L_m)$

Let us provide some intuition of our algorithm. Fix a sample $y = A^* x^* + \varepsilon$ from the available training set, and consider samples

$$u = A^* \alpha + \varepsilon_u, v = A^* \alpha' + \varepsilon_v.$$

Now, consider the (very coarse) estimate for the sparse code of u with respect to A^*:

$$\beta = A^{*T} u = A^{*T} A^* \alpha + A^{*T} \varepsilon_u.$$

As long as A^* is incoherent enough and ε_u is small, the estimate β behaves just like α, in the sense that for each sample y:

$$\langle y, u \rangle \approx \langle x^*, \beta \rangle \approx \langle x^*, \alpha \rangle.$$

Moreover, the above inner products are large only if α and x^* share some elements in their supports; else, they are likely to be small. Likewise, the weight $\langle y, u \rangle \langle y, v \rangle$ depends on whether or not x^* shares the support with both α and α'.

Now, suppose that we have a mechanism to isolate pairs u and v who share exactly one atom among their sparse representations. Then by scaling each sample y with an increasing function of $\langle y, u \rangle \langle y, v \rangle$ and linearly adding the samples, we magnify the importance of the samples that are aligned with that atom, and diminish the rest. The final direction can be obtained via the top *principal component* of the reweighted samples and hence can be used as a coarse estimate of the atom. This is exactly the approach adopted in Arora et al. (2015). However, in our double-sparse coding setting, we know that the estimated atom

should be sparse as well. Therefore, we can naturally perform an extra "sparsification" step of the output. An extended algorithm and its correctness are provided in Appendix F. However, as we discussed above, the computational complexity of the re-weighting step still remains.

We overcome this obstacle by first identifying the locations of the nonzero entries in each atom. Specifically, define the matrix:

$$M_{u,v} = \frac{1}{p_2} \sum_{i=1}^{p_2} \langle y^{(i)}, u \rangle \langle y^{(i)}, v \rangle y^{(i)} y^{(i)T}.$$

Then, the diagonal entries of $M_{u,v}$ reveals the support of the atom of A^* shared among u and v: the r-largest entries of $M_{u,v}$ will correspond to the support we seek. Since the desired direction remains unchanged in the r-dimensional subspace of its nonzero elements, we can restrict our attention to this subspace, construct a reduced covariance matrix $\widehat{M}_{u,v}$, and proceed as before. This truncation step alleviates the computational burden by a significant amount; the running time is now $\widetilde{O}(mnp)$, which improves the original by a factor of n.

The success of the above procedure relies upon whether or not we can isolate pairs u and v that share one dictionary atom. Fortunately, this can be done via checking the decay of the singular values of the (reduced) covariance matrix. Here too, we show via our analysis that the truncation step plays an important role. Overall, our proposed algorithm not only accelerates the initialization in terms of running time, but also improves the sample complexity over Arora et al. (2015). The performance of Algorithm 1 is described in the following theorem, whose formal proof is deferred to Appendix B.

Theorem 4 *Suppose that Assumptions* **B1-B4** *hold and Assumptions* **A1-A3** *satify with* $\mu = O^*\big(\frac{\sqrt{n}}{k \log^3 n}\big)$ *and* $r = O^*(\log n)$. *When* $p_1 = \widetilde{\Omega}(m)$ *and* $p_2 = \widetilde{\Omega}(mr)$, *then with high probability Algorithm 1 returns an initial estimate* A^0 *whose columns share the same support as* A^* *and with* $(\delta, 2)$*-nearness to* A^* *with* $\delta = O^*(1/\log n)$.

The limit on r arises from the minimum non-zero coefficient τ of A^*. Since the columns of A^* are standardized, τ should degenerate as r grows. In other words, it is getting harder to distinguish the "signal" coefficients from zero as r grows with n. However, this limitation can be relaxed when a better incoherence available, for example the orthonormal case. We study this in Appendix E.

To provide some intuition about the working of the algorithm (and its proof), let us analyze it in the case where we have access to infinite number of samples. This setting, of course, is unrealistic. However, the analysis is much simpler and more transparent since we can focus on expected values rather than empirical averages. Moreover, the analysis reveals several key lemmas, which we will reuse extensively for proving Theorem 4. First, we give some intuition behind the definition of the "scores", $\widehat{e}_l$.

Lemma 1 *Fix samples* u *and* v *and suppose that* $y = A^* x^* + \varepsilon$ *is a random sample independent of* u, v. *The expected value of the score for the* ℓ^{th} *component of* y *is given by:*

$$e_l \triangleq \mathbb{E}[\langle y, u \rangle \langle y, v \rangle y_l^2] = \sum_{i \in U \cap V} q_i c_i \beta_i \beta_i' A_{li}^{*2} + \text{ perturbation terms}$$

where $q_i = \mathbb{P}[i \in S]$, $q_{ij} = \mathbb{P}[i, j \in S]$ and $c_i = \mathbb{E}[x_i^4 | i \in S]$. Moreover, the perturbation terms have absolute value at most $O^*(k/m \log n)$.

From Assumption **B1**, we know that $q_i = \Theta(k/m)$, $q_{ij} = \Theta(k^2/m^2)$ and $c_i = \Theta(1)$. Besides, we will show later that $|\beta_i| \approx |\alpha_i| = \Omega(1)$ for $i \in U$, and $|\beta_i| = o(1)$ for $i \notin U$. Consider the first term $E_0 = \sum_{i \in U \cap V} q_i c_i \beta_i \beta_i' A_{li}^{*2}$. Clearly, $E_0 = 0$ if $U \cap V = \emptyset$ or that l does not belong to support of any atom in $U \cap V$. On the contrary, as $E_0 \neq 0$ and $U \cap V = \{i\}$, then $E_0 = |q_i c_i \beta_i \beta_i' A_{li}^{*2}| \geq \Omega(\tau^2 k/m) = \Omega(k/mr)$ since $|q_i c_i \beta_i \beta_i'| \geq \Omega(k/m)$ and $|A_{li}^*| \geq \tau$.

Therefore, Lemma 1 suggests that if u and v share a unique atom among their sparse representations, and r is not too large, then we can indeed recover the correct support of the shared atom. When this is the case, the expected scores corresponding to the nonzero elements of the shared atom will dominate the remaining of the scores.

Now, given that we can isolate the support R of the corresponding atom, the remaining questions are how best we can estimate its non-zero coefficients, and when u and v share a unique elements in their supports. These issues are handled in the following lemmas.

Lemma 2 *Suppose that $u = A^*\alpha + \varepsilon_u$ and $v = A^*\alpha' + \varepsilon_v$ are two random samples. Let U and V denote the supports of α and α' respectively. R is the support of some atom of interest. The truncated re-weighting matrix is formulated as*

$$M_{u,v} \triangleq \mathbb{E}[\langle y, u\rangle \langle y, v\rangle y_R y_R^T] = \sum_{i \in U \cap V} q_i c_i \beta_i \beta_i' A_{R,i}^* A_{R,i}^{*T} + \text{ perturbation terms}$$

where the perturbation terms have norms at most $O^*(k/m \log n)$.

Using the same argument for bounding E_0 in Lemma 1, we can see that $M_0 \triangleq q_i c_i \beta_i \beta_i' A_{R,i}^* A_{R,i}^{*T}$ has norm at least $\Omega(k/m)$ when u and v share a unique element i ($\|A_{R,i}^*\| = 1$). According to this lemma, the spectral norm of M_0 dominates those of the other perturbation terms. Thus, given R we can use the first singular vector of $M_{u,v}$ as an estimate of $A_{\bullet i}^*$.

Lemma 3 *Under the setup of Theorem 4, suppose $u = A^*\alpha + \varepsilon_u$ and $v = A^*\alpha' + \varepsilon_v$ are two random samples with supports U and V respectively. $R = \text{supp}(A_i^*)$. If u and v share the unique atom i, the first r largest entries of e_l is at least $\Omega(k/mr)$ and belong to R. Moreover, the top singular vector of $M_{u,v}$ is δ-close to $A_{R,i}^*$ for $O^*(1/\log n)$.*

Proof The recovery of $A_{\bullet i}^*$'s support directly follows Lemma 1. For the latter part, recall from Lemma 2 that

$$M_{u,v} = q_i c_i \beta_i \beta_i' A_{R,i}^* A_{R,i}^{*T} + \text{ perturbation terms}$$

The perturbation terms have norms bounded by $O^*(k/m \log n)$. On the other hand, the first term is has norm at least $\Omega(k/m)$ since $\|A_{R,i}^*\| = 1$ for the correct support R and $|q_i c_i \beta_i \beta_i'| \geq \Omega(k/m)$. Then using Wedin's Theorem to $M_{u,v}$, we can conclude that the top singular vector must be $O^*(k/m \log n)/\Omega(k/m) = O^*(1/\log n)$ -close to $A_{R,i}^*$. ■

Lemma 4 *Under the setup of Theorem 4, suppose $u = A^*\alpha + \varepsilon_u$ and $v = A^*\alpha' + \varepsilon_v$ are two random samples with supports U and V respectively. If the top singular value of $M_{u,v}$ is at least $\Omega(k/m)$ and the second largest one is at most $O^*(k/m \log n)$, then u and v share a unique dictionary element with high probability.*

Proof The proof follows from that of Lemma 37 in Arora et al. (2015). The main idea is to separate the possible cases of how u and v share support and to use Lemma 2 with the bounded perturbation terms to conclude when u and v share exactly one. We note that due to the condition where $\widehat{e}_{(s)} \geq \Omega(k/mr)$ and $\widehat{e}_{(s+1)}/\widehat{e}_{(s)} \leq O^*(r/\log n)$, it must be the case that u and v share only one atom or share more than one atoms with the same support. When their supports overlap more than one, then the first singular value cannot dominate the second one, and hence it must not be the case. ■

Similar to (Arora et al., 2015), our initialization algorithm requires $\widetilde{O}(m)$ iterations in expectation to estimate all the atoms, hence the expected running time is $\widetilde{O}(mnp)$. All the proofs of Lemma 1 and 2 are deferred to Appendix B.

4. Stage 2: Descent

We now adapt the neural sparse coding approach of Arora et al. (2015) to obtain an improved estimate of A^*. As mentioned earlier, at a high level the algorithm is akin to performing approximate gradient descent. The insight is that within a small enough neighborhood (in the sense of δ-closeness) of the true A^*, an estimate of the ground-truth code vectors, X^*, can be constructed using a neurally plausible algorithm.

The innovation, in our case, is the double-sparsity model since we know *a priori* that A^* is itself sparse. Under sufficiently many samples, the support of A^* can be deduced from the initialization stage; therefore we perform an extra *projection* step in each iteration of gradient descent. In this sense, our method is non-trivially different from Arora et al. (2015). The full algorithm is presented as Algorithm 2.

As discussed in Section 2, convergence of noisy approximate gradient descent can be achieved as long as $\widehat{g}^s$ is correlated-w.h.p. with the true solution. However, an analogous convergence result for *projected* gradient descent does not exist in the literature. We fill this gap via a careful analysis. Due to the projection, we only require the correlated-w.h.p. property for *part* of $\widehat{g}^s$ (i.e., when it is restricted to some support set) with A^*. The descent property is still achieved via Theorem 5. Due to various perturbation terms, $\widehat{g}$ is only a biased estimate of $\nabla_A \mathcal{L}(A, X)$; therefore, we can only refine the estimate of A^* until the column-wise error is of order $O(\sqrt{k/n})$. The performance of Algorithm 2 can be characterized via the following theorem.

Theorem 5 *Suppose that the initial estimate A^0 has the correct column supports and is $(\delta, 2)$-near to A^* with $\delta = O^*(1/\log n)$. If Algorithm 2 is provided with $p = \widetilde{\Omega}(mr)$ fresh samples at each step and $\eta = \Theta(m/k)$, then*

$$\mathbb{E}[\|A_{\bullet i}^s - A_{\bullet i}^*\|^2] \leq (1-\rho)^s \|A_{\bullet i}^0 - A_{\bullet i}^*\|^2 + O(\sqrt{k/n})$$

for some $0 < \rho < 1/2$ and for $s = 1, 2, \ldots, T$. Consequently, A^s converges to A^* geometrically until column-wise error $O(\sqrt{k/n})$.

Algorithm 2 Double-Sparse Coding Descent Algorithm

Initialize A^0 is $(\delta, 2)$-near to A^*. $H = (h_{ij})_{n\times m}$ where $h_{ij} = 1$ if $i \in \text{supp}(A^0_{\bullet j})$ and 0 otherwise.

Repeat for $s = 0, 1, \ldots, T$

Decode: $x^{(i)} = \text{threshold}_{C/2}((A^s)^T y^{(i)})$ for $i = 1, 2, \ldots, p$

Update: $A^{s+1} = \mathcal{P}_H(A^s - \eta \widehat{g}^s) = A^s - \eta \mathcal{P}_H(\widehat{g}^s)$

where $\widehat{g}^s = \frac{1}{p}\sum_{i=1}^p (A^s x^{(i)} - y^{(i)})\text{sgn}(x^{(i)})^T$ and $\mathcal{P}_H(G) = H \circ G$

We defer the full proof of Theorem 5 to Section D. In this section, we take a step towards understanding the algorithm by analyzing $\widehat{g}^s$ in the infinite sample case, which is equivalent to its expectation $g^s \triangleq \mathbb{E}[(A^s x - y)\text{sgn}(x)^T]$. We establish the $(\alpha, \beta, \gamma_s)$-correlation of a truncated version of $g^s_{\bullet i}$ with $A^*_{\bullet i}$ to obtain the descent in Theorem 6 for the infinite sample case.

Theorem 6 *Suppose that the initial estimate A^0 has the correct column supports and is $(\delta, 2)$-near to A^*. If Algorithm 2 is provided with infinite number of samples at each step and $\eta = \Theta(m/k)$, then*

$$\|A^{s+1}_{\bullet i} - A^*_{\bullet i}\|^2 \le (1-\rho)\|A^s_{\bullet i} - A^*_{\bullet i}\|^2 + O\big(k^2/n^2\big)$$

for some $0 < \rho < 1/2$ and for $s = 1, 2, \ldots, T$. Consequently, it converges to A^* geometrically until column-wise error is $O(k/n)$.

Note that the better error $O(k^2/n^2)$ is due to the fact that infinitely many samples are given. The term $O(\sqrt{k/n})$ in Theorem 5 is a trade-off between the accuracy and the sample complexity of the algorithm. The proof of this theorem composes of two steps with two main results: 1) an explicit form of g^s (Lemma 5); 2) $(\alpha, \beta, \gamma_s)$-correlation of column-wise g^s with A^* (Lemma 6). The proof of those lemmas are deferred to Appendix C. Since the correlation primarily relies on the $(\delta, 2)$-nearness of A^s to A^* that is provided initially and maintained at each step, then we need to argue that the nearness is preserved after each step.

Lemma 5 *Suppose that the initial estimate A^0 has the correct column supports and is $(\delta, 2)$-near to A^*. The column-wise update has the form $g^s_{R,i} = p_i q_i(\lambda^s_i A^s_{R,i} - A^*_{R,i} + \xi^s_i \pm \zeta)$ where $R = \text{supp}(A^s_{\bullet i})$, $\lambda^s_i = \langle A^s_{\bullet i}, A^*_{\bullet i}\rangle$ and*

$$\xi^s_i = A^s_{R,-i}\text{diag}(q_{ij})(A^s_{\bullet -i})^T A^*_{\bullet i}/q_i.$$

Moreover, ξ_i has norm bounded by $O(k/n)$ for $\delta = O^*(1/\log n)$ and ζ is negligible.

We underline that the correct support of A^s allows us to obtain the closed-form expression of $g^s_{R_i,i}$ in terms of $A^s_{\bullet i}$ and $A^*_{\bullet i}$. Likewise, the gradient form suggests that $g^s_{\bullet i}$ is almost equal to $p_i q_i(A^s_{\bullet i} - A^*_{\bullet i})$ (since $\lambda^s_i \approx 1$), which directs to the desired solution $A^*_{\bullet i}$. With Lemma 5, we will prove the $(\alpha, \beta, \gamma_s)$-correlation of the approximate gradient to each column $A^*_{\bullet i}$ and the nearness of each new update to the true solution A^*.

4.1. $(\alpha, \beta, \gamma_s)$-Correlation

Lemma 6 *Suppose that A^s to be $(\delta, 2)$-near to A^* and $R = \text{supp}(A^*_{\bullet i})$, then $2g^s_{R,i}$ is $(\alpha, 1/2\alpha, \epsilon^2/\alpha)$-correlated with $A^*_{R,i}$; that is*

$$\langle 2g^s_{R,i}, A^s_{R,i} - A^*_{R,i}\rangle \ge \alpha\|A^s_{R,i} - A^*_{R,i}\|^2 + 1/(2\alpha)\|g^s_{R,i}\|^2 - \epsilon^2/\alpha.$$

where $\delta = O^*(1/\log n)$ and $\epsilon = O\big(\frac{k^2}{mn}\big)$. Futhermore, the descent is achieved by

$$\|A^{s+1}_{\bullet i} - A^*_{\bullet i}\|^2 \le (1-2\alpha\eta)^s\|A^0_{\bullet i} - A^*_{\bullet i}\|^2 + \eta\epsilon^2/\alpha.$$

Proof Throughout the proof, we omit the superscript s for simplicity and denote $2\alpha = p_i q_i$. First, we rewrite $g^s_{\bullet i}$ as a combination of the true direction $A^s_{\bullet i} - A^*_{\bullet i}$ and a term with small norm:

$$g_{R,i} = 2\alpha(A_{R,i} - A^*_{R,i}) + v, \tag{2}$$

where $v = 2\alpha[(\lambda_i - 1)A_{\bullet i} + \epsilon_i]$ with norm bounded. In fact, since $A_{\bullet i}$ is δ-close to $A^*_{\bullet i}$, and both have unit norm, then $\|2\alpha(\lambda_i - 1)A_{\bullet i}\| = \alpha\|A_{\bullet i} - A^*_{\bullet i}\|^2 \le \alpha\|A_{\bullet i} - A^*_{\bullet i}\|$ and $\|\xi_i\| \le O(k/n)$ from the inequality (9). Therefore,

$$\|v\| = \|2\alpha(\lambda_i - 1)A_{R,i} + 2\alpha\xi_i\| \le \alpha\|A_{R,i} - A^*_{R,i}\| + \epsilon$$

where $\epsilon = O(k^2/mn)$. Now, we make use of (2) to show the first part of Lemma 6:

$$\langle 2g_{R,i}, A_{R,i} - A^*_{R,i}\rangle = 4\alpha\|A_{R,i} - A^*_{R,i}\|^2 + \langle 2v, A_{R,i} - A^*_{R,i}\rangle. \tag{3}$$

We want to lower bound the inner product term with respect to $\|g_{R_i,i}\|^2$ and $\|A_{R,i} - A^*_{R,i}\|^2$. Effectively, from (2)

$$\begin{aligned} 4\alpha\langle v, A_{\bullet i} - A^*_{\bullet i}\rangle &= \|g_{R,i}\|^2 - 4\alpha^2\|A_{R,i} - A^*_{R,i}\|^2 - \|v\|^2 \\ &\ge \|g_{R,i}\|^2 - 6\alpha^2\|A_{R,i} - A^*_{R,i}\|^2 - 2\epsilon^2, \end{aligned} \tag{4}$$

where the last step is due to Cauchy-Schwarz inequality: $\|v\|^2 \le 2(\alpha^2\|A_{R,i} - A^*_{R,i}\|^2 + \epsilon^2)$.

Substitute $2\langle v, A_{\bullet i} - A^*_{\bullet i}\rangle$ in (3) for the right hand side of (4), we get the first result:

$$\langle 2g_{R,i}, A_{R,i} - A^*_{R,i}\rangle \ge \alpha\|A_{R,i} - A^*_{R,i}\|^2 + \frac{1}{2\alpha}\|g_{R,i}\|^2 - \frac{\epsilon^2}{\alpha}.$$

The second part is directly followed from Theorem 1. Moreover, we have $p_i = \Theta(k/m)$ and $q_i = \Theta(1)$, then $\alpha = \Theta(k/m)$, $\beta = \Theta(m/k)$ and $\gamma_s = O(k^3/mn^2)$. Then $g^s_{R,i}$ is $(\Omega(k/m), \Omega(m/k), O(k^3/mn^2))$-correlated with the true solution $A^*_{R,i}$. □

Proof [Proof of Theorem 6] The descent in Theorem 6 directly follows from the above lemma. Next, we will establish the nearness for the update at step s:

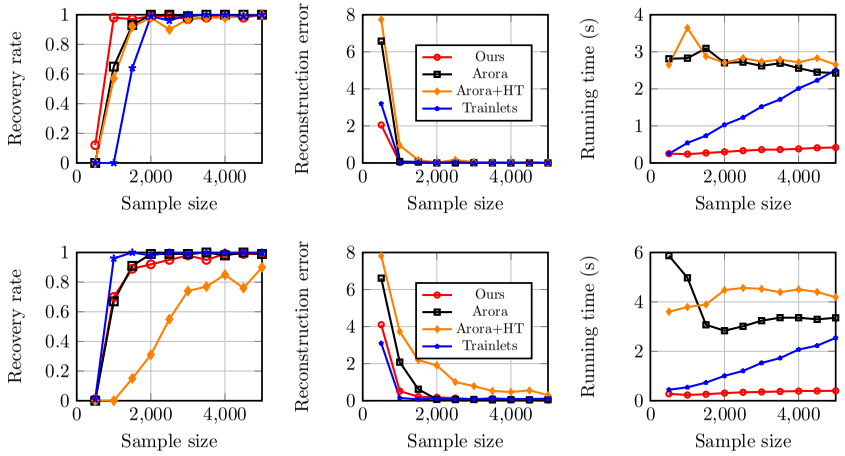


Figure 1: *(top) The performance of four methods on three metrics (recovery rate, reconstruction error and running time (in seconds)) in sample size in the noiseless case. (bottom) The same metrics are measured for the noisy case.*

4.2. Nearness

Lemma 7 *Suppose that A^s is $(\delta, 2)$-near to A^*, then $\|A^{s+1} - A^*\| \leq 2\|A^*\|$*

Proof [Proof] From Lemma 5 we have $g^s_{\bullet i} = p_i q_i(\lambda_i A^s_{\bullet i} - A^*_{\bullet i}) + A_{\bullet -i}\mathrm{diag}(q_{ij})A^T_{\bullet -i}A^*_{\bullet i} \pm \zeta$. Denote $\bar{R} = [n]\backslash R$, then it is obvious that $g^s_{\bar{R},i} = A_{\bar{R},-i}\mathrm{diag}(q_{ij})A^T_{\bullet -i}A^*_{\bullet i} \pm \zeta$ is bounded by $O(k^2/m^2)$. Then we follows the proof of Lemma 24 in (Arora et al., 2015) for the nearness with full $g^s = g^s_{R,i} + g^s_{\bar{R},i}$ to finish the proof for this lemma. ■

In sum, we have shown the descent property of Algorithm 2 in the infinite sample case. The study of the concentration of $\widehat{g}^s$ around its mean to the sample complexity is provided in Section D. In the next section, we corroborate our theory by some numerical results on synthetic data.

5. Empirical Study

We compare our method with three different methods for both standard sparse and double-sparse coding. For a baseline, we implement the algorithm proposed in Arora et al. (2015), which currently is the most theoretically sound approach for provable sparse coding. However, since their approach is not directly designed for the double-sparsity model, we implement a modified version that performs a hard thresholding (HT)-based post-processing step in the initialization and learning procedures (which we dub *Arora + HT*). The final comparison is with *Trainlets*, the proposed approach by Sulam et al. (2016).

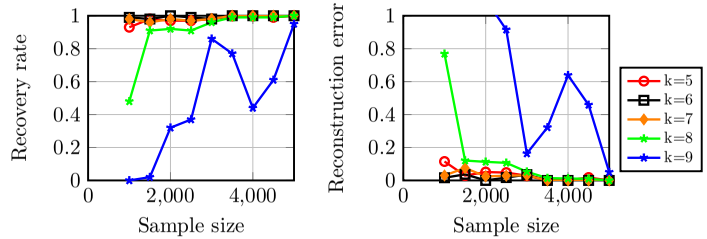


Figure 2: The performance of our method in the noiseless case as the sparsity k varies.

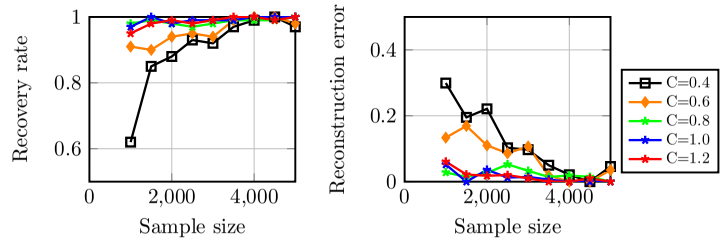


Figure 3: The performance of our method in the noiseless case as the thresholding parameter C varies.

We generate a synthetic training dataset according to the model described in Section 2. The base dictionary Φ is the identity matrix of size $n = 64$, and the square synthesis matrix A^* has a special block structure with 32 blocks. Each block is of size 2×2 and of form $[1\ 1; 1\ \ -1]$ (i.e., the column sparsity of A^* is $r = 2$). The support of x^* is drawn uniformly over all 6-dimensional subsets of $[m]$, and the nonzero coefficients are randomly set to ± 1 with equal probability. In our simulations with noise, we add Gaussian noise ε with entrywise variance $\sigma^2_\varepsilon = 0.01$ to each of those above samples. For all the approaches except Trainlets, we use $T = 2000$ iterations for the initialization procedure, and set the number of steps in the descent stage to 25. Since Trainlets does not have a specified initialization procedure, we initialize it with a random Gaussian matrix upon which column-wise sparse thresholding is then performed. The learning step of Trainlets[2] is executed for 50 iterations, which tolerates its initialization deficiency. For each Monte Carlo trial, we uniformly draw p samples, feed these samples to the four different algorithms, and observe their ability to reconstruct A^*. Matlab implementation of our algorithms is available online[3].

2. We utilize Trainlets's implementation provided at http://jsulam.cswp.cs.technion.ac.il/home/software/.
3. https://github.com/thanh-isu/double-sparse-coding

5.1. Comparison with Other Approaches

We evaluate these approaches on three metrics as a function of the number of available samples: (i) fraction of trials in which each algorithm successfully recovers the ground truth A^*; (ii) reconstruction error; and (iii) running time (in seconds). The synthesis matrix is said to be "successfully recovered" if the Frobenius norm of the difference between the estimate $\widehat{A}$ and the ground truth A^* is smaller than a threshold which is set to 10^{-4} in the noiseless case, and to 0.5 in the other. All three metrics are averaged over 100 Monte Carlo simulations. As discussed above, the Frobenius norm is only meaningful under a suitable permutation and sign flip transformation linking $\widehat{A}$ and A^*. We estimate this transformation using a simple maximum weight matching algorithm. Specifically, we construct a weighted bipartite graph with nodes representing columns of A^* and $\widehat{A}$ and adjacency matrix defined as $G = |A^{*T}\widehat{A}|$, where $|\cdot|$ is taken element-wise. We compute the optimal matching using the Hungarian algorithm, and then estimate the sign flips by looking at the sign of the inner products between the matched columns.

The results of our experiments are shown in Figure 1 with the top and bottom rows respectively for the noiseless and noisy cases. The two leftmost figures suggest that all algorithms exhibit a "phase transition" in sample complexity that occurs in the range of 500-2000 samples. In the noiseless case, our method achieves the phase transition with the fewest number of samples. In the noisy case, our method nearly matches the best sample complexity performance (next to Trainlets, which is a heuristic and computationally expensive). Our method achieves the best performance in terms of (wall-clock) running time in all cases.

5.2. Robustness to Data Assumptions

In this last experiement, we show that our approach is robust to the data assumptions. We numerically study how the initialization and descent algorithms behave when the sparsity k and the thresholding parameter C slightly vary around the groundtruth values. Since our focus is on the recovery property of our approach, we assume that the dictionary size m and sparsity r are known a priori and do not experiement on them.

The results are shown in Figures 2 and 3. When the sparsity and the minimum coefficient are around the true setting, $k_{\text{model}} = 6$ and $C_{\text{model}} = 1.0$, our algorithm is still able recover the dictionary perfectly. When these parameters are set more extreme, the phase transition is not obvious but is gradually achieved with more and more samples.

6. Conclusion

In this paper, we have addressed an open theoretical question on learning sparse dictionaries under a special type of generative model. Our proposed algorithm consists of a novel initialization step followed by a descent-style step, both are able to take advantage of the sparse structure. We rigorously demonstrate its efficacy in both sample- and computation-complexity over existing heuristics as well as provable approaches for double-sparse and regular sparse coding. This results in *the first known provable approach for double-sparse coding problem* with statistical and algorithmic guarantees. Besides, we also show three

benefits of our approach: neural plausibility, robustness to noise and practical usefulness via the numerical experiments.

Nevertheless, several fundamental questions regarding our approach remain. First, our initialization method (in the overcomplete case) achieves its theoretical guarantees under fairly stringent limitations on the sparsity level r. This arises due to our reweighted spectral initialization strategy, and it is an open question whether a better initialization strategy exists (or whether these types of initialization are required at all). Second, our analysis holds for complete (fixed) bases Φ, and it remains open to study the setting where Φ is over-complete. Finally, understanding the reasons behind the very promising practical performance of methods based on heuristics, such as Trainlets, on real-world data remains a very challenging open problem.

Acknowledgments

We would like to thank the anonymous reviewers for the constructive feedback and suggestions. This work is supported in part by the National Science Foundation under the grants CCF-1566281 and DMS-1612985.

Appendix Organization We organize the appendix as follows: we prove the two key lemmas for Theorem 4 of the initialization algorithm 1 in Appendix B. In Appendix C, we prove the result stated in Theorem 5 for the infinite-sample case. The sample complexity results for both stages are proved in Appendix D.

Additionally, we prove some extended results from Arora et al. (2015) and for some special cases in Appendices E and F. The final section details the neural implementation of our approach.

Appendix A. Useful Result

We start our proof with the following claim, which we will use throughout.

Claim 1 (Maximal row ℓ_1-norm) *Given that $\|A^*\|_F^2 = m$ and $\|A^*\| = O(\sqrt{m/n})$, then $\|A^{*T}\|_{1,2} = \Theta(\sqrt{m/n})$.*

Proof Recall the definition of the operator norm:

$$\|A^{*T}\|_{1,2} = \sup_{x\neq 0} \frac{\|A^T x\|}{\|x\|_1} \leq \sup_{x\neq 0} \frac{\|A^T x\|}{\|x\|} = \|A^{*T}\| = O(\sqrt{m/n}).$$

Since $\|A^*\|_F^2 = m$, $\|A^{*T}\|_{1,2} \geq \|A^*\|_F/\sqrt{n} = \sqrt{m/n}$. Combining with the above, we have $\|A^{*T}\|_{1,2} = \Theta(\sqrt{m/n})$. ■

Along with Assumptions **A1** and **A3**, the above claim implies the number of nonzero entries in each row is $O(r)$. This Claim is an important ingredient in our analysis of our initialization algorithm shown in Section 3.

Appendix B. Analysis of Initialization Algorithm

B.1. Proof of Lemma 1

Recall some important notations: $y = A^* x^* + \varepsilon$ and two samples

$$u = A^*\alpha + \varepsilon_u, v = A^*\alpha' + \varepsilon_v.$$

Also, recall the very coarse estimate for the sparse code of u with respect to A^*:

$$\beta = A^{*T} u = A^{*T} A^* \alpha + A^{*T}\varepsilon_u.$$

We split the proof of Lemma 1 into three steps: 1) we first establish useful properties of β with respect to α; 2) we then explicitly derive e_l in terms of the generative model parameters and β; and 3) we finally bound the error terms in E based on the first result and appropriate assumptions.

Claim 2 *In the generative model, $\|x^*\| \leq \widetilde{O}(\sqrt{k})$ and $\|\varepsilon\| \leq \widetilde{O}(\sigma_\varepsilon\sqrt{n})$ with high probability.*

Proof The claim directly follows from the fact that x^* is a k-sparse random vector whose nonzero entries are independent sub-Gaussian with variance 1. Meanwhile, ε has n independent Gaussian entries of variance σ_ε^2. □

Despite its simplicity, this claim will be used in many proofs throughout the paper. Note also that in this section we will calculate the expectation over y and often refer probabilistic bounds (w.h.p.) under the randomness of u and v.

Claim 3 *Suppose that $u = A^*\alpha + \varepsilon_u$ is a random sample and $U = \mathrm{supp}(\alpha)$. Let $\beta = A^{*T}u$, then, w.h.p., we have (a) $|\beta_i - \alpha_i| \leq \frac{\mu k \log n}{\sqrt{n}} + \sigma_\varepsilon \log n$ for each i and (b) $\|\beta\| \leq \widetilde{O}(\sqrt{k} + \sigma_\varepsilon\sqrt{n})$.*

Proof The proof mostly follows from Claim 36 of Arora et al. (2015), with an additional consideration of the error ε_u. Write $W = U\backslash\{i\}$ and observe that

$$|\beta_i - \alpha_i| = |A_{\bullet i}^{*T} A_{\bullet W}^* \alpha_W + A_{\bullet i}^{*T}\varepsilon_u| \leq |\langle A_{\bullet W}^{*T} A_{\bullet i}^*, \alpha_W\rangle| + |\langle A_{\bullet i}^*, \varepsilon_u\rangle|$$

Since A^* is μ-incoherence, then $\|A_{\bullet i}^{*T} A_{\bullet W}^*\| \leq \mu\sqrt{k/n}$. Moreover, α_W has $k-1$ independent sub-Gaussian entries of variance 1, therefore $|\langle A_{\bullet W}^{*T} A_{\bullet i}^*, \alpha_W\rangle| \leq \frac{\mu k \log n}{\sqrt{n}}$ with high probability. Also recall that ε_u has independent Gaussian entries of variance σ_ε^2, then $A_{\bullet i}^{*T}\varepsilon_u$ is Gaussian with the same variance ($\|A_{\bullet i}^*\| = 1$). Hence $|A_{\bullet i}^{*T}\varepsilon| \leq \sigma_\varepsilon \log n$ with high probability. Consequently, $|\beta_i - \alpha_i| \leq \frac{\mu k \log n}{\sqrt{n}} + \sigma_\varepsilon \log n$, which is the first part of the claim.

Next, in order to bound $\|\beta\|$, we express β as

$$\|\beta\| = \|A^{*T} A_{\bullet U}^* \alpha_U + A^{*T}\varepsilon_u\| \leq \|A^*\|\|A_{\bullet U}^*\|\|\alpha_U\| + \|A^*\|\|\varepsilon_u\|$$

Using Claim 2 to get $\|\alpha_U\| \leq \widetilde{O}(\sqrt{k})$ and $\|\varepsilon_u\| \leq \widetilde{O}(\sigma_\varepsilon\sqrt{n})$ w.h.p., and further noticing that $\|A_{\bullet U}^*\| \leq \|A^*\| \leq O(1)$, we complete the proof for the second part. ■

Claim 3 suggests that the difference between β_i and α_i is bounded above by $O^*(1/\log^2 n)$ w.h.p. if $\mu = O^*(\frac{\sqrt{n}}{k\log^3 n})$. Therefore, w.h.p., $C - o(1) \leq |\beta_i| \leq |\alpha_i| + o(1) \leq O(\log m)$ for $i \in U$ and $|\beta_i| \leq O^*(1/\log^2 n)$ otherwise. On the other hand, under Assumption **B4**, $\|\beta\| \leq \widetilde{O}(\sqrt{k})$ w.h.p. We will use these results multiple times in the next few proofs.

Proof [Proof of Lemma 1] We decompose d_l into small parts so that the stochastic model $\mathcal{D}$ is made use.

$$\begin{aligned} e_l &= \mathbb{E}[\langle y, u\rangle\langle y, v\rangle y_l^2] = \mathbb{E}[\langle A^* x^* + \varepsilon, u\rangle\langle A^* x^* + \varepsilon, v\rangle(\langle A_{l\cdot}^*, x^*\rangle + \varepsilon_l)^2] \\ &= \mathbb{E}\big[\big\{\langle x^*, \beta\rangle\langle x^*, \beta'\rangle + x^{*T}(\beta v^T + \beta' u^T)\varepsilon + u^T\varepsilon\varepsilon^T v\big\}\big\{\langle A_{l\bullet}^*, x^*\rangle^2 + 2\langle A_{l\bullet}^*, x^*\rangle\varepsilon_l + \varepsilon_l^2\big\}\big] \\ &= E_1 + E_2 + \cdots + E_9 \end{aligned}$$

where the terms are

$$\begin{aligned} E_1 &= \mathbb{E}[\langle x^*, \beta\rangle\langle x^*, \beta'\rangle\langle A_{l\bullet}^*, x^*\rangle^2] \\ E_2 &= 2\mathbb{E}[\langle x^*, \beta\rangle\langle x^*, \beta'\rangle\langle A_{l\bullet}^*, x^*\rangle\varepsilon_l] \\ E_3 &= \mathbb{E}[\langle x^*, \beta\rangle\langle x^*, \beta'\rangle\varepsilon_l^2] \\ E_4 &= \mathbb{E}\big[\langle A_{l\cdot}^*, x^*\rangle^2 x^{*T}(\beta v^T + \beta' u^T)\varepsilon\big] \\ E_5 &= \mathbb{E}\big[\langle A_{l\cdot}^*, x^*\rangle x^{*T}(\beta v^T + \beta' u^T)\varepsilon\varepsilon_l\big] \\ E_6 &= \mathbb{E}\big[(\beta v^T + \beta' u^T)\varepsilon\varepsilon_l^2\big] \\ E_7 &= \mathbb{E}[u^T\varepsilon\varepsilon^T v\langle A_{l\bullet}^*, x^*\rangle^2] \\ E_8 &= 2\mathbb{E}[u^T\varepsilon\varepsilon^T v\langle A_{l\bullet}^*, x^*\rangle\varepsilon_l] \\ E_9 &= \mathbb{E}[u^T\varepsilon\varepsilon^T v\varepsilon_l^2] \end{aligned} \tag{5}$$

Because x^* and ε are independent and zero-mean, E_2 and E_4 are clearly zero. Moreover,

$$E_6 = (\beta v^T + \beta' u^T)\mathbb{E}[\varepsilon\varepsilon_l^2] = 0$$

due to the fact that $\mathbb{E}[\varepsilon_j\varepsilon_l^2] = 0$, for $j \neq l$, and $\mathbb{E}[\varepsilon_l^3] = 0$. Also,

$$E_8 = A_{l\bullet}^{*T}\mathbb{E}[x^*]\mathbb{E}\big[u^T\varepsilon\varepsilon^T v\varepsilon_l\big] = 0.$$

We bound the remaining terms separately in the following claims.

Claim 4 *In the decomposition* (5)*, E_1 is of the form*

$$E_1 = \sum_{i\in U\cap V} q_i c_i\beta_i\beta_i' A_{li}^{*2} + \sum_{i\notin U\cap V} q_i c_i\beta_i\beta_i' A_{li}^{*2} + \sum_{j\neq i} q_{ij}(\beta_i\beta_i' A_{lj}^{*2} + 2\beta_i\beta_j' A_{li}^* A_{lj}^*)$$

where all those terms except $\sum_{i\in U\cap V} q_i c_i\beta_i\beta_i' A_{li}^{*2}$ have magnitude at most $O^*(k/m\log^2 n)$ w.h.p.

Proof Using the generative model in Assumptions **B1-B4**, we have

$$\begin{aligned}
E_1 &= \mathbb{E}[\langle x^*,\beta\rangle\langle x^*,\beta'\rangle\langle A_{l\bullet}^*, x^*\rangle^2] \\
&= \mathbb{E}_S\big[\mathbb{E}_{x^*|S}[\sum_{i\in S}\beta_i x_i^*\sum_{i\in S}\beta_i' x_i^*\big(\sum_{i\in S} A_{li}^* x_i^*\big)^2]\big] \\
&= \sum_{i\in[m]} q_i c_i\beta_i\beta_i' A_{li}^{*2} + \sum_{i,j\in[m],j\neq i} q_{ij}(\beta_i\beta_i' A_{lj}^{*2} + 2\beta_i\beta_j' A_{li}^* A_{lj}^*) \\
&= \sum_{i\in U\cap V} q_i c_i\beta_i\beta_i' A_{li}^{*2} + \sum_{i\notin U\cap V} q_i c_i\beta_i\beta_i' A_{li}^{*2} + \sum_{j\neq i} q_{ij}(\beta_i\beta_i' A_{lj}^{*2} + 2\beta_i\beta_j' A_{li}^* A_{lj}^*),
\end{aligned}$$

where we have used the $q_i = \mathbb{P}[i\in S]$, $q_{ij} = \mathbb{P}[i,j\in S]$ and $c_i = \mathbb{E}[x_i^4|i\in S]$ and Assumptions **B1-B4**. We now prove that the last three terms are upper bounded by $O^*(k/m\log n)$. The key observation is that all these terms typically involve a quadratic form of the l-th row $A_{l\bullet}^*$ whose norm is bounded by $O(1)$ (by Claim 1 and Assumption **A4**). Moreover, $|\beta_i\beta_i'|$ is relatively small for $i\notin U\cap V$ while $q_{ij} = \Theta(k^2/m^2)$. For the second term, we apply the Claim 3 for $i\in[m]\backslash(U\cap V)$ to bound $|\beta_i\beta_i'|$. Assume $\alpha_i = 0$ and $\alpha_i' \neq 0$, then with high probability

$$|\beta_i\beta_i'| \leq |(\beta_i - \alpha_i)(\beta_i' - \alpha_i')| + |\beta_i\alpha_i'| \leq O^*(1/\log n)$$

Using the bound $q_i c_i = \Theta(k/m)$, we have w.h.p.,

$$\Big|\sum_{i\notin U\cap V} q_i c_i\beta_i\beta_i' A_{li}^{*2}\Big| \leq \max_i |q_i c_i\beta_i\beta_i'|\sum_{i\notin U\cap V} A_{li}^{*2} \leq \max_i |q_i c_i\beta_i\beta_i'|\|A^*\|_{1,2}^2 \leq O^*(k/m\log n).$$

For the third term, we make use of the bounds on $\|\beta\|$ and $\|\beta'\|$ from the previous claim where $\|\beta\|\|\beta'\| \leq \widetilde{O}(k)$ w.h.p., and on $q_{ij} = \Theta(k^2/m^2)$. More precisely, w.h.p.,

$$\begin{aligned}
\Big|\sum_{j\neq i} q_{ij}\beta_i\beta_i' A_{lj}^{*2}\Big| &= \Big|\sum_i \beta_i\beta_i'\sum_{j\neq i} q_{ij} A_{lj}^{*2}\Big| \leq \sum_i |\beta_i\beta_i'|\big(\sum_{j\neq i} q_{ij} A_{lj}^{*2}\big) \\
&\leq (\max_{i\neq j} q_{ij})\sum_i |\beta_i\beta_i'|\Big(\sum_j A_{lj}^{*2}\Big) \leq (\max_{i\neq j} q_{ij})\|\beta\|\|\beta'\|\|A^*\|_{1,2}^2 \leq \widetilde{O}(k^3/m^2),
\end{aligned}$$

where the second last inequality follows from the Cauchy-Schwarz inequality. For the last term, we write it in a matrix form as $\sum_{j\neq i} q_{ij}\beta_i\beta_j' A_{li}^* A_{lj}^* = A_{l\bullet}^{*T} Q_\beta A_{l\bullet}^*$ where $(Q_\beta)_{ij} = q_{ij}\beta_i\beta_j'$ for $i\neq j$ and $(Q_\beta)_{ij} = 0$ for $i = j$. Then

$$|A_{l\bullet}^{*T} Q_\beta A_{l\bullet}^*| \leq \|Q_\beta\|\|A_{l\bullet}^*\|^2 \leq \|Q_\beta\|_F\|A^*\|_{1,2}^2,$$

where $\|Q_\beta\|_F^2 = \sum_{i\neq j} q_{ij}^2\beta_i^2(\beta_j')^2 \leq (\max_{i\neq j} q_{ij}^2)\sum_i \beta_i^2\sum_j(\beta_j')^2 \leq (\max_{i\neq j} q_{ij}^2)\|\beta\|^2\|\beta'\|^2$. Ultimately,

$$\Big|\sum_{j\neq i} q_{ij}\beta_i\beta_j' A_{li}^* A_{lj}^*\Big| \leq (\max_{i\neq j} q_{ij})\|\beta\|\|\beta'\|\|A^*\|_{1,2}^2 \leq \widetilde{O}(k^3/m^2).$$

Under Assumption $k = O^*(\frac{\sqrt{n}}{\log n})$, then $\widetilde{O}(k^3/m^2) \leq O^*(k/m\log^2 n)$. As a result, the two terms above are bounded by the same amount $O^*(k/m\log n)$ w.h.p., so we complete the proof of the claim. ■

Claim 5 *In the decomposition (5), $|E_3|$, $|E_5|$, $|E_7|$ and $|E_9|$ are at most $O^*(k/m\log^2 n)$.*

Proof Recall that $\mathbb{E}[x_i^2|S] = 1$ and $q_i = \mathbb{P}[i\in S] = \Theta(k/m)$ for $S = \mathrm{supp}(x^*)$, then

$$\begin{aligned}
E_3 = \mathbb{E}[\langle x^*,\beta\rangle\langle x^*,\beta'\rangle\varepsilon_l^2] &= \sigma_\varepsilon^2\mathbb{E}_S\big[\mathbb{E}_{x^*|S}[\sum_{i,j\in S}\beta_i\beta_j' x_i^* x_j^*]\big] \\
&= \sigma_\varepsilon^2\mathbb{E}_S[\sum_{i\in S}\beta_i\beta_i'] = \sum_i \sigma_\varepsilon^2 q_i\beta_i\beta_i'
\end{aligned}$$

Denote $Q = \mathrm{diag}(q_1, q_2, \ldots, q_m)$, then $|E_3| = |\sigma_\varepsilon^2\langle Q\beta,\beta'\rangle| \leq \sigma_\varepsilon^2\|Q\|\|\beta\|\|\beta'\| \leq \widetilde{O}(\sigma_\varepsilon^2 k^2/m) = \widetilde{O}(k^3/mn)$ where we have used $\|\beta\| \leq \widetilde{O}(\sqrt{k})$ w.h.p. and $\sigma_\varepsilon \leq O(1/\sqrt{n})$. For convenience, we handle the seventh term before E_5:

$$E_7 = \mathbb{E}[u^T\varepsilon\varepsilon^T v\langle A_{l\bullet}^*, x^*\rangle^2] = \mathbb{E}[\langle A_{l\bullet}^*, x^*\rangle^2]u^T\mathbb{E}[\varepsilon\varepsilon^T]v = \sum_i \sigma_\varepsilon^2\langle u,v\rangle q_i A_{li}^2 = \sigma_\varepsilon^2\langle u,v\rangle A_{l\bullet}^T Q A_{l\bullet}$$

To bound this term, we use Claim 9 in Appendix D to have $\|u\| = \|A^*\alpha + \varepsilon_u\| \leq \widetilde{O}(\sqrt{k})$ w.h.p. and $\langle u,v\rangle \leq \widetilde{O}(\sqrt{k})$ w.h.p. Consequently, $|E_7| \leq \sigma_\varepsilon^2\|Q\|\|A_{l\bullet}\|^2|\langle u,v\rangle| \leq \widetilde{O}(k^2/mn)$ because $\|A_{l\bullet}\|^2 \leq O(m/n)$ and $\sigma_\varepsilon \leq O(1/\sqrt{n})$. Now, the firth term E_5 is expressed as follows

$$\begin{aligned}
E_5 &= \mathbb{E}\big[\langle A_{l\cdot}^*, x^*\rangle x^{*T}(\beta v^T + \beta' u^T)\varepsilon\varepsilon_l\big] \\
&= A_{l\bullet}^{*T}\mathbb{E}\big[x^* x^{*T}\big](\beta v^T + \beta' u^T)\mathbb{E}[\varepsilon\varepsilon_l] \\
&= \sigma_\varepsilon^2 A_{l\bullet}^{*T} Q(v_l\beta + u_l\beta')
\end{aligned}$$

Observe that $|E_5| \leq \sigma_\varepsilon^2\|A_{l\bullet}^{*T}\|\|Q(v_l\beta + u_l\beta')\| \leq \sigma_\varepsilon^2\|A_{l\bullet}^{*T}\|\|Q\|\|v_l\beta + u_l\beta'\|$ and that $\|v_l\beta + u_l\beta'\| \leq 2\|u\|\|\beta\| \leq \widetilde{O}(k)$ w.h.p. using the result $\|u\| \leq \widetilde{O}(k)$ and $\|\beta\| \leq \widetilde{O}(k)$ from Claim 3, then E_5 bounded by $\widetilde{O}(k^2/mn)$.

The last term

$$E_9 = \mathbb{E}[u^T\varepsilon\varepsilon^T v\varepsilon_l^2] = u^T\mathbb{E}\big[\varepsilon\varepsilon^T\varepsilon_l^2\big]v = 9\sigma_\varepsilon^4\langle u,v\rangle$$

because the independent entries of ε and $\mathbb{E}[\varepsilon_l^4] = 9\sigma_\varepsilon^4$. Therefore, $|E_9| \le 9\sigma_\varepsilon^4 \|u\|\|v\| \le \widetilde{O}(k^2/n^2)$. Since $m = O(n)$ and $k \le O^*(\frac{\sqrt{n}}{\log n})$, we obtain the same bound $O^*(k/m \log^2 n)$ for $|E_3|$, $|E_5|$, $|E_7|$ and $|E_9|$, and conclude the proof of the claim. ■

Combining the bounds from Claim 4, 5 for every single term in (5), we finish the proof for Lemma 1. □

B.2. Proof of Lemma 2

We prove this lemma by using the same strategy used to prove Lemma 1.

$$\begin{aligned}
M_{u,v} &\triangleq \mathbb{E}[\langle y, u\rangle\langle y, v\rangle y_R y_R^T] \\
&= \mathbb{E}[\langle A^* x^* + \varepsilon, u\rangle\langle A^* x^* + \varepsilon, v\rangle (A^*_{R\bullet} x^* + \varepsilon_R)(A^*_{R\bullet} x^* + \varepsilon_R)^T] \\
&= \mathbb{E}\big[\big\{\langle x^*, \beta\rangle\langle x^*, \beta'\rangle + x^{*T}(\beta v^T + \beta' u^T)\varepsilon + u^T \varepsilon\varepsilon^T v\big\}\big\{A^*_{R\bullet} x^* x^{*T} A^{*T}_{R\bullet} + A^*_{R\bullet} x^* \varepsilon_R^T + \varepsilon_R x^{*T} A^{*T}_{R\bullet} + \varepsilon_R \varepsilon_R^T\big\}\big] \\
&= M_1 + \cdots + M_8,
\end{aligned}$$

in which only nontrivial terms are kept in place, including

$$\begin{aligned}
M_1 &= \mathbb{E}[\langle x^*, \beta\rangle\langle x^*, \beta'\rangle A^*_{R\bullet} x^* x^{*T} A^{*T}_{R\bullet}] \\
M_2 &= \mathbb{E}[\langle x^*, \beta\rangle\langle x^*, \beta'\rangle \varepsilon_R \varepsilon_R^T] \\
M_3 &= \mathbb{E}[x^{*T}(\beta v^T + \beta' u^T)\varepsilon A^*_{R\bullet} x^* \varepsilon_R^T] \\
M_4 &= \mathbb{E}[x^{*T}(\beta v^T + \beta' u^T)\varepsilon\varepsilon_R x^{*T} A^{*T}_{R\bullet}] \\
M_5 &= \mathbb{E}[u^T \varepsilon\varepsilon^T v A^*_{R\bullet} x^* x^{*T} A^{*T}_{R\bullet}] \\
M_6 &= \mathbb{E}[u^T \varepsilon\varepsilon^T v A^*_{R\bullet} x^* \varepsilon_R^T] \\
M_7 &= \mathbb{E}[u^T \varepsilon\varepsilon^T v \varepsilon_R^T x^{*T} A^{*T}_{R\bullet}] \\
M_8 &= \mathbb{E}[u^T \varepsilon\varepsilon^T v \varepsilon_R \varepsilon_R^T]
\end{aligned} \tag{6}$$

By swapping inner product terms and taking advantage of the independence, we can show that $M_6 = \mathbb{E}[A^*_{R\bullet} x^* u^T \varepsilon\varepsilon^T v \varepsilon_R^T] = 0$ and $M_7 = \mathbb{E}[u^T \varepsilon\varepsilon^T v \varepsilon_R^T x^{*T} A^{*T}_{R\bullet}] = 0$. The remaining are bounded in the next claims.

Claim 6 *In the decomposition* (6),

$$M_1 = \sum_{i \in U\cap V} q_i c_i \beta_i \beta_i' A^*_{R,i} A^{*T}_{R,i} + E_1' + E_2' + E_3'$$

where $E_1' = \sum_{i \notin U\cap V} q_i c_i \beta_i \beta_i' A^*_{R,i} A^{*T}_{R,i}$, $E_2' = \sum_{i \ne j} q_{ij} \beta_i \beta_i' A^*_{R,j} A^{*T}_{R,j}$ *and* $E_3' = \sum_{i \ne j} q_{ij}(\beta_i A^*_{R,i} \beta_j' A^{*T}_{R,j} + \beta_i' A^*_{R,i} \beta_j A^{*T}_{R,j})$ *have norms bounded by* $O^*(k/m \log n)$.

Proof The expression of M_1 is obtained in the same way as E_1 is derived in the proof of Lemma 1. To prove the claim, we bound all the terms with respect to the spectral norm of $A^*_{R\bullet}$ and make use of Assumption **A4** to find the exact upper bound.

For the first term E_1', rewrite $E_1' = A^*_{R,S} D_1 A^{*T}_{R,S}$ where $S = [m]\backslash(U \cap V)$ and D_1 is a diagonal matrix whose entries are $q_i c_i \beta_i \beta_i'$. Clearly, $\|D_1\| \le \max_{i\in S} |q_i c_i \beta_i \beta_i'| \le O^*(k/m \log n)$ as shown in Claim 4, then

$$\|E_1'\| \le \max_{i\in S} |q_i c_i \beta_i \beta_i'| \|A^*_{R,S}\|^2 \le \max_{i\in S} |q_i c_i \beta_i \beta_i'| \|A^*_{R\bullet}\|^2 \le O^*(k/m \log n)$$

where $\|A^*_{R,S}\| \le \|A^*_{R\bullet}\| \le O(1)$. The second term E_2' is a sum of positive semidefinite matrices, and $\|\beta\| \le O(k \log n)$, then

$$E_2' = \sum_{i\ne j} q_{ij} \beta_i \beta_i' A^*_{R,j} A^{*T}_{R,j} \preceq \max_{i\ne j} q_{ij} \Big(\sum_i \beta_i \beta_i'\Big)\Big(\sum_j A^*_{R,j} A^{*T}_{R,j}\Big) \preceq (\max_{i\ne j} q_{ij}) \|\beta\|\|\beta'\| A^*_{R\bullet} A^{*T}_{R\bullet}$$

which implies that $\|E_2'\| \le (\max_{i\ne j} q_{ij})\|\beta\|\|\beta'\|\|A^*_{R\bullet}\|^2 \le \widetilde{O}(k^3/m^2)$. Observe that E_3' has the same form as the last term in Claim 4, which is $E_3' = A^{*T}_{R\bullet} Q_\beta A^*_{R\bullet}$. Then

$$\|E_3'\| \le \|Q_\beta\|\|A^*_{R\bullet}\|^2 \le (\max_{i\ne j} q_{ij})\|\beta\|\|\beta'\|\|A^*_{R\bullet}\|^2 \le \widetilde{O}(k^3/m^2)$$

By Claim 3, we have $\|\beta\|$ and $\|\beta'\|$ are bounded by $O(\sqrt{k}\log n)$, and note that $k \le O^*(\sqrt{n}/\log n)$, then we complete the proof for Lemma 6. ■

Claim 7 *In the decomposition* (6), M_2, M_3, M_4, M_5 *and* M_8 *have norms bounded by* $O^*(k/m \log n)$.

Proof Recall the definition of Q in Claim 5 and use the fact that $\mathbb{E}[x^* x^{*T}] = Q$, we can get $M_2 = \mathbb{E}[\langle x^*, \beta\rangle\langle x^*, \beta'\rangle \varepsilon_R \varepsilon_R^T] = \sum_i \sigma_\varepsilon^2 q_i \beta_i \beta_i' I_r$. Then, $\|M_2\| \le \sigma_\varepsilon^2 \max_i q_i \|\beta\|\|\beta'\| \le O(\sigma_\varepsilon^2 k^2 \log^2 n/m)$.

The next three terms all involve $A^*_{R\bullet}$ whose norm is bounded according to Assumption **A4**. Specifically,

$$\begin{aligned}
M_3 &= \mathbb{E}[x^{*T}(\beta v^T + \beta' u^T)\varepsilon A^*_{R\bullet} x^* \varepsilon_R^T] = \mathbb{E}[A^*_{R\bullet} x^* x^{*T}(\beta v^T + \beta' u^T)\varepsilon \varepsilon_R^T] \\
&= A^*_{R\bullet} \mathbb{E}[x^* x^{*T}](\beta v^T + \beta' u^T)\mathbb{E}[\varepsilon\varepsilon_R^T] \\
&= A^*_{R\bullet} Q(\beta v^T + \beta' u^T)\mathbb{E}[\varepsilon\varepsilon_R^T],
\end{aligned}$$

and

$$\begin{aligned}
M_4 &= \mathbb{E}[x^{*T}(\beta v^T + \beta' u^T)\varepsilon\varepsilon_R x^{*T} A^{*T}_{R\bullet}] = \mathbb{E}[\varepsilon_R \varepsilon^T (v\beta^T + u\beta'^T) x^* x^{*T} A^{*T}_{R\bullet}] \\
&= \mathbb{E}[\varepsilon_R \varepsilon^T](v\beta^T + u\beta'^T)\mathbb{E}[x^* x^{*T}] A^{*T}_{R\bullet} \\
&= \mathbb{E}[\varepsilon_R \varepsilon^T](v\beta^T + u\beta'^T) Q A^{*T}_{R\bullet},
\end{aligned}$$

and the fifth term $M_5 = \mathbb{E}[u^T \varepsilon\varepsilon^T v A^*_{R\bullet} x^* x^{*T} A^{*T}_{R\bullet}] = \sigma_\varepsilon^2 u^T v A^*_{R\bullet} \mathbb{E}[x^* x^{*T}] A^{*T}_{R\bullet} = \sigma_\varepsilon^2 u^T v A^*_{R\bullet} Q A^{*T}_{R\bullet}$. We already have $\|\mathbb{E}[\varepsilon\varepsilon_R^T]\| = \sigma_\varepsilon^2$, $\|Q\| \le O(k/m)$ and $|u^T v| \le \widetilde{O}(k)$ (proof of Claim 9), then the remaining work is to bound $\|\beta v^T + \beta' u^T\|$, then the bound of $v\beta^T + u\beta'^T$ directly follows. We have $\|\beta v^T\| = \|A^* u v^T\| \le \|A^*\|\|u\|\|v\| \le \widetilde{O}(k)$. Therefore, all three terms M_3, M_4 and M_5 are bounded in norm by $\widetilde{O}(\sigma_\varepsilon^2 k^2/m) \le \widetilde{O}(k^3/mn)$.

The remaining term is

$$\begin{aligned}
M_8 &= \mathbb{E}[u^T \varepsilon\varepsilon^T v \varepsilon_R \varepsilon_R^T] = \mathbb{E}[(\sum_{i,j} u_i v_j \varepsilon_i \varepsilon_j)\varepsilon_R \varepsilon_R^T] \\
&= \mathbb{E}[(\sum_{i\in R} u_i v_i \varepsilon_i^2 \varepsilon_R \varepsilon_R^T)] + \mathbb{E}[(\sum_{i\ne j} u_i v_j \varepsilon_i \varepsilon_j)\varepsilon_R \varepsilon_R^T] \\
&= \sigma_\varepsilon^4 u_R v_R^T
\end{aligned}$$

where $u_R = A^*_{R\bullet}\alpha + (\varepsilon_u)_R$ and $v_R = A^*_{R\bullet}\alpha' + (\varepsilon_v)_R$. We can see that $\|u_R\| \le \|A^*_{R\bullet}\|\|\alpha\| + \|(\varepsilon_u)_R\| \le \widetilde{O}(\sqrt{k})$. Therefore, $\|M8\| \le \widetilde{O}(\sigma_\varepsilon^4 k) = \widetilde{O}(k^3/n^2)$. Since $m = O(n)$ and $k \le O^*(\frac{\sqrt{n}}{\log n})$, then we can bound all the above terms by $O^*(k/m \log n)$ and finish the proof of Claim 7. ■

Combine the results of Claim 6 and 7, we complete the proof of Lemma 2.

Appendix C. Analysis of Main Algorithm

C.1. Simple Encoding

We can see that $(A^s x - y)\mathrm{sgn}(x)^T$ is random over y and x that is obtained from the encoding step. We follow (Arora et al., 2015) to derive the closed form of $g^s = \mathbb{E}[(A^s x - y)\mathrm{sgn}(x)^T]$ by proving that the encoding recovers the sign of x^* with high probability as long as A^s is close enough to A^*.

Lemma 8 *Assume that A^s is δ-close to A^* for $\delta = O(r/n\log n)$ and $\mu \le \frac{\sqrt{n}}{2k}$, and $k \ge \Omega(\log m)$ then with high probability over random samples $y = A^* x^* + \varepsilon$*

$$\mathrm{sgn}(\mathrm{threshold}_{C/2}((A^s)^T y) = \mathrm{sgn}(x^*) \tag{7}$$

Proof [Proof of Lemma 8] We follow the same proof strategy from (Arora et al., 2015) (Lemmas 16 and 17) to prove a more general version in which the noise ε is taken into account. Write $S = \mathrm{supp}(x^*)$ and skip the superscript s on A^s for the readability. What we need is to show $S = \{i \in [m] : \langle A_{\bullet i}, y\rangle \ge C/2\}$ and then $\mathrm{sgn}(\langle A^s_{\bullet i}, y\rangle) = \mathrm{sgn}(x^*_i)$ for each $i \in S$ with high probability. Following the same argument of (Arora et al., 2015), we prove in below a stronger statement that, even conditioned on the support S, $S = \{i \in [m] : |\langle A_{\bullet i}, y\rangle| \ge C/2\}$ with high probability.

Rewrite

$$\langle A_{\bullet i}, y\rangle = \langle A_{\bullet i}, A^* x^* + \varepsilon\rangle = \langle A_{\bullet i}, A^*_{\bullet i}\rangle x^*_i + \sum_{j\ne i}\langle A_{\bullet i}, A^*_{\bullet j}\rangle x^*_j + \langle A_{\bullet i}, \varepsilon\rangle,$$

and observe that, due to the closeness of $A_{\bullet i}$ and $A^*_{\bullet i}$, the first term is either close to x^*_i or equal to 0 depending on whether or not $i \in S$. Meanwhile, the rest are small due to the incoherence and the concentration in the weighted average of noise. We will show that both $Z_i = \sum_{S\setminus\{i\}}\langle A_{\bullet i}, A^*_{\bullet j}\rangle x^*_j$ and $\langle A_{\bullet i}, \varepsilon\rangle$ are bounded by $C/8$ with high probability.

The cross-term $Z_i = \sum_{S\setminus\{i\}}\langle A_{\bullet i}, A^*_{\bullet j}\rangle x^*_j$ is a sum of zero-mean independent sub-Gaussian random variables, which is another sub-Gaussian random variable with variance $\sigma^2_{Z_i} = \sum_{S\setminus\{i\}}\langle A_{\bullet i}, A^*_{\bullet j}\rangle^2$. Note that

$$\langle A_{\bullet i}, A^*_{\bullet j}\rangle^2 \le 2\big(\langle A^*_{\bullet i}, A^*_{\bullet j}\rangle^2 + \langle A_{\bullet i} - A^*_{\bullet i}, A^*_{\bullet j}\rangle^2\big) \le 2\mu^2/n + 2\langle A_{\bullet i} - A^*_{\bullet i}, A^*_{\bullet j}\rangle^2,$$

where we use Cauchy-Schwarz inequality and the μ-incoherence of A^*. Therefore,

$$\sigma^2_{Z_i} \le 2\mu^2 k/n + 2\|A^{*T}_{\bullet S}(A_{\bullet i} - A^*_{\bullet i})\|_F^2 \le 2\mu^2 k/n + 2\|A^*_{\bullet S}\|^2\|A_{\bullet i} - A^*_{\bullet i}\|^2 \le O(1/\log n),$$

under $\mu \le \frac{\sqrt{n}}{2k}$, to conclude $2\mu^2 k/n \le O(1/\log n)$ we need $1/k = O(1/\log n)$, i.e. $k = \Omega(\log n)$. Applying Bernstein's inequality, we get $|Z_i| \le C/8$ with high probability. What

remains is to bound the noise term $\langle A_{\bullet i}, \varepsilon\rangle$. In fact, $\langle A_{\bullet i}, \varepsilon\rangle$ is sum of n Gaussian random variables, which is a sub-Gaussian with variance σ^2_ε. It is easy to see that $|\langle A_{\bullet i}, \varepsilon\rangle| \le \sigma_\varepsilon \log n$ with high probability. Notice that $\sigma_\varepsilon = O(1/\sqrt{n})$.

Finally, we combine these bounds to have $|Z_i + \langle A_{\bullet i}, \varepsilon\rangle| \le C/4$. Therefore, for $i \in S$, then $|\langle A_{\bullet i}, y\rangle| \ge C/2$ and negligible otherwise. Using union bound for every $i = 1, 2, \ldots, m$, we finish the proof of the Lemma. □

Lemma 8 enables us to derive the expected update direction $g^s = \mathbb{E}[(A^s x - y)\mathrm{sgn}(x)^T]$ explicitly.

C.2. Approximate Gradient in Expectation

Proof [Proof of Lemma 5] Having the result from Lemma 8, we are now able to study the expected update direction $g^s = \mathbb{E}[(A^s x - y)\mathrm{sgn}(x)^T]$. Recall that A^s is the update at the s-th iteration and $x \triangleq \mathrm{threshold}_{C/2}((A^s)^T y)$. Based on the generative model, denote $p_i = \mathbb{E}[x^*_i \mathrm{sgn}(x^*_i) | i \in S], q_i = \mathbb{P}[i \in S]$ and $q_{ij} = \mathbb{P}[i, j \in S]$. Throughout this section, we will use ζ to denote any vector whose norm is negligible although they can be different across their appearances. A_{-i} denotes the sub-matrix of A whose i-th column is removed. To avoid overwhelming appearance of the superscript s, we skip it from A^s for neatness. Denote $\mathcal{F}_{x^*}$ is the event under which the support of x is the same as that of x^*, and $\bar{\mathcal{F}}_{x^*}$ is its complement. In other words, $\mathbf{1}_{\mathcal{F}_{x^*}} = \mathbf{1}[\mathrm{sgn}(x) = \mathrm{sgn}(x^*)]$ and $\mathbf{1}_{\mathcal{F}_{x^*}} + \mathbf{1}_{\bar{\mathcal{F}}_{x^*}} = 1$.

$$g^s_{\bullet i} = \mathbb{E}[(Ax - y)\mathrm{sgn}(x_i)] = \mathbb{E}[(Ax - y)\mathrm{sgn}(x_i)\mathbf{1}_{\mathcal{F}_{x^*}}] \pm \zeta$$

Using the fact that $y = A^* x^* + \varepsilon$ and that under $\mathcal{F}_{x^*}$ we have $Ax = A_{\bullet S}x_S = A_{\bullet S}A^T_{\bullet S}y = A_{\bullet S}A^T_{\bullet S}A^* x^* + A_{\bullet S}A^T_{\bullet S}\varepsilon$. Using the independence of ε and x^* to get rid of the noise term, we get

$$\begin{aligned}
g^s_{\bullet i} &= \mathbb{E}[(A_{\bullet S}A^T_{\bullet S} - I_n)A^* x^* \mathbf{1}_{\mathcal{F}_{x^*}}] + \mathbb{E}[(A_{\bullet S}A^T_{\bullet S} - I_n)\varepsilon\mathrm{sgn}(x_i)\mathbf{1}_{\mathcal{F}_{x^*}}] \pm \zeta \\
&= \mathbb{E}[(A_{\bullet S}A^T_{\bullet S} - I_n)A^* x^* \mathrm{sgn}(x_i)\mathbf{1}_{\mathcal{F}_{x^*}}] \pm \zeta \quad \text{(Independence of } \varepsilon \text{ and } x\text{'s)} \\
&= \mathbb{E}[(A_{\bullet S}A^T_{\bullet S} - I_n)A^* x^* \mathrm{sgn}(x^*_i)(1 - \mathbf{1}_{\bar{\mathcal{F}}_{x^*}})] \pm \zeta \quad \text{(Under } \mathcal{F}_{x^*} \text{ event)} \\
&= \mathbb{E}[(A_{\bullet S}A^T_{\bullet S} - I_n)A^* x^* \mathrm{sgn}(x^*_i)] \pm \zeta
\end{aligned}$$

Recall from the generative model assumptions that $S = \mathrm{supp}(x^*)$ is random and the entries of x^* are pairwise independent given the support, so

$$\begin{aligned}
g^s_{\bullet i} &= \mathbb{E}_S\mathbb{E}_{x^*|S}[(A_{\bullet S}A^T_{\bullet S} - I_n)A^* x^* \mathrm{sgn}(x^*_i)] \pm \zeta \\
&= p_i\mathbb{E}_{S, i\in S}[(A_{\bullet S}A^T_{\bullet S} - I_n)A^*_{\bullet i}] \pm \zeta \\
&= p_i\mathbb{E}_{S, i\in S}[(A_{\bullet i}A^T_{\bullet i} - I_n)A^*_{\bullet i}] + p_i\mathbb{E}_{S, i\in S}[\sum_{l\in S, l\ne i} A_{\bullet l}A^T_{\bullet l}A^*_{\bullet i}] \pm \zeta \\
&= p_i q_i(A_{\bullet i}A^T_{\bullet i} - I_n)A^*_{\bullet i} + p_i\sum_{l\in[m], l\ne i} q_{il}A_{\bullet l}A^T_{\bullet l}A^*_{\bullet i} \pm \zeta \\
&= p_i q_i(\lambda_i A_{\bullet i} - A^*_{\bullet i}) + p_i A_{\bullet -i}\mathrm{diag}(q_{ij})A^T_{\bullet -i}A^*_{\bullet i} \pm \zeta
\end{aligned}$$

where $\lambda^s_i = \langle A^s_{\bullet i}, A^*_{\bullet i}\rangle$. Let $\xi^s_i = A_{R,-i}\mathrm{diag}(q_{ij})A^T_{\bullet -i}A^*_{\bullet i}/q_i$ for $j = 1, \ldots, m$, we now have the full expression of the expected approximate gradient at iteration s:

$$g^s_{R,i} = p_i q_i(\lambda_i A^s_{R,i} - A^*_{R,i} + \xi^s_i) \pm \zeta_R. \tag{8}$$

What remains is to bound norms of ξ_s and ζ. We have $\|A^s_{R,-i}\| \le \|A^s_{-i}\| \le O(\sqrt{m/n})$ w.h.p. Then, along with the fact that $\|A^*_i\| = 1$, we can bound $\|\xi^s_i\|$

$$\|\xi^s_i\| \le \|A^s_{R_i,-i}\| \max_{j\neq i} \frac{q_{ij}}{q_i} \|A^s_{-i}\| \le O(k/n). \tag{9}$$

Next, we show that norm of ζ is negligible. In fact, $\mathcal{F}_{x^*}$ happens with very high probability, then it suffices to bound norm of $(Ax-y)\mathrm{sgn}(x_i)$ which will be done using Lemma 12 and Lemma 11 in Section D. This concludes the proof for Lemma 5. ■

Appendix D. Sample Complexity

In previous sections, we rigorously analyzed both initialization and learning algorithms as if the expectations g^s, e and $M_{u,v}$ were given. Here we show that corresponding estimates based on empirical means are sufficient for the algorithms to succeed, and identify how may samples are required. Technically, this requires the study of their concentrations around their expectations. Having had these concentrations, we are ready to prove Theorems 4 and 5.

The entire section involves a variety of concentration bounds. Here we make heavy use of Bernstein's inequality for different types of random variables (including scalar, vector and matrix). The Bernstein's inequality is stated as follows.

Lemma 9 (Bernstein's Inequality) *Suppose that $Z^{(1)}, Z^{(2)}, \dots, Z^{(p)}$ are p i.i.d. samples from some distribution $\mathcal{D}$. If $\mathbb{E}[Z] = 0$, $\|Z^{(j)}\| \le \mathcal{R}$ almost surely and $\|\mathbb{E}[Z^{(j)}(Z^{(j)})^T\| \le \sigma^2$ for each j, then*

$$\frac{1}{p}\Big\|\sum_{j=1}^{p} Z^{(j)}\Big\| \le \widetilde{O}\Big(\frac{\mathcal{R}}{p} + \sqrt{\frac{\sigma^2}{p}}\Big) \tag{10}$$

holds with probability $1 - n^{-\omega(1)}$.

Since all random variables (or their norms) are not bounded almost surely in our model setting, we make use of a technical lemma that is used in Arora et al. (2015) to handle the issue.

Lemma 10 (Arora et al. (2015)) *Suppose a random variable Z satisfies $\mathbb{P}[\|Z\| \ge \mathcal{R}(\log(1/\rho))^C] \le \rho$ for some constant $C > 0$, then*

(a) If $p = n^{O(1)}$, it holds that $\|Z^{(j)}\| \le \widetilde{O}(\mathcal{R})$ for each j with probability $1 - n^{-\omega(1)}$.

(b) $\|\mathbb{E}[Z\mathbf{1}_{\|Z\|\ge\widetilde{\Omega}(\mathcal{R})}]\| = n^{-\omega(1)}$.

This lemma suggests that if $\frac{1}{p}\sum_{i=1}^{p} Z^{(j)}(1-\mathbf{1}_{\|Z^{(j)}\|\ge\widetilde{\Omega}(\mathcal{R})})$ concentrates around its mean with high probability, then so does $\frac{1}{p}\sum_{i=1}^{p} Z^{(j)}$ because the part outside the truncation level can be ignored. Since all random variables of our interest are sub-Gaussian or a product of sub-Gaussian that satisfy this lemma, we can apply Lemma 9 to the corresponding truncated random variables with carefully chosen truncation levels. Then the original random variables concentrate likewise.

In the next proofs, we define suitable random variables and identify good bounds of $\mathcal{R}$ and σ^2 for them. Note that in this section, the expectations are taken over y by conditioning

on u and v. This aligns with the construction that the estimators of e and $M_{u,v}$ are empirical averages over i.i.d. samples of y, while u and v are kept fixed. Due to the dependency on u and v, these (conditional) expectations inherit randomness from u and v, and we will formulate probabilistic bounds for them.

The application of Bernstein's inequality requires a bound on $\|\mathbb{E}[ZZ^T(1-\mathbf{1}_{\|Z\|\ge\widetilde{\Omega}(\mathcal{R}))}]\|$. We achieve that by the following technical lemma, where $\tilde{Z}$ is a standardized version of Z.

Lemma 11 *Suppose a random variable $\tilde{Z}\tilde{Z}^T = aT$ where $a \ge 0$ and T is positive semi-definite. They are both random. Suppose $\mathbb{P}[a \ge \mathcal{A}] = n^{-\omega(1)}$ and $\mathcal{B} > 0$ is a constant. Then,*

$$\|\mathbb{E}[\tilde{Z}\tilde{Z}^T(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})]\| \le \mathcal{A}\|\mathbb{E}[T]\| + O(n^{-\omega(1)})$$

Proof To show this, we make use of the decomposition $\tilde{Z}\tilde{Z}^T = aT$ and a truncation for a. Specifically,

$$\begin{aligned}
\|\mathbb{E}[\tilde{Z}\tilde{Z}^T(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})]\| &= \mathbb{E}[aT(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})] \\
&\le \|\mathbb{E}[a(1-\mathbf{1}_{a\ge\mathcal{A}})T(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})]\| + \|\mathbb{E}[a\mathbf{1}_{a\ge\mathcal{A}}T(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})]\| \\
&\le \|\mathbb{E}[a(1-\mathbf{1}_{a\ge\mathcal{A}})T]\| + \mathbb{E}[a\mathbf{1}_{a\ge\mathcal{A}}\|T\|(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})] \\
&\le \mathcal{A}\|\mathbb{E}[T]\| + \big(\mathbb{E}[\|aT\|^2(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})]\mathbb{E}[\mathbf{1}_{a\ge\mathcal{A}}]\big)^{1/2} \\
&\le \mathcal{A}\|\mathbb{E}[T]\| + \big(\mathbb{E}[\|\tilde{Z}\|^4(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})]\mathbb{P}[a\ge\mathcal{A}]\big)^{1/2} \\
&\le \mathcal{A}\|\mathbb{E}[T]\| + \mathcal{B}^2\big(\mathbb{P}[a\ge\mathcal{A}]\big)^{1/2} \\
&\le \mathcal{A}\|\mathbb{E}[T]\| + O(n^{-\omega(1)}),
\end{aligned}$$

where at the third step we used $T(1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}})] \preceq T$ because of the fact that T is the positive semi-definite and $1-\mathbf{1}_{\|\tilde{Z}\|\ge\mathcal{B}} \in \{0,1\}$. Then, we finish the proof of the lemma. ■

D.1. Sample Complexity of Algorithm 1

In Algorithm 1, we empirically compute the "scores" $\widehat{e}$ and the reduced weighted covariance matrix $\widehat{M}_{u,v}$ to produce an estimate for each column of A^*. Since the construction of $\widehat{M}_{u,v}$ depends upon the support estimate $\widehat{R}$ given by ranking $\widehat{e}$, we denote it by $\widehat{M}^{\widehat{R}}_{u,v}$. We will show that we only need $p = \widetilde{O}(m)$ samples to be able to recover the support of one particular atom and up to some specified level of column-wise error with high probability.

Lemma 12 *Consider Algorithm 1 in which p is the given number of samples. For any pair u and v, then with high probability a) $\|\widehat{e} - e\| \le O^*(k/m\log^2 n)$ when $p = \widetilde{\Omega}(m)$ and b) $\|\widehat{M}^{\widehat{R}}_{u,v} - M^R_{u,v}\| \le O^*(k/m\log n)$ when $p = \widetilde{\Omega}(mr)$ where $\widehat{R}$ and R are respectively the estimated and correct support sets of one particular atom.*

D.1.1. Proof of Theorem 4

Using Lemma 12, we are ready to prove the Theorem 4. According to Lemma 1 when $U \cap V = \{i\}$, we can write $\widehat{e}$ as

$$\widehat{e} = q_i c_i \beta_i \beta_i' A^*_{R,i} \circ A^*_{R,i} + \text{ perturbation terms} + (\widehat{e} - e),$$

and consider $\widehat{e} - e$ as an additional perturbation with the same magnitude $O^*(k/m \log^2 n)$ in the sense of $\|\cdot\|_\infty$ w.h.p. The first part of Lemma 3 suggests that when u and v share exactly one atom i, then the set $\widehat{R}$ including r largest elements of $\widehat{e}$ is the same as $\text{supp}(A_i^*)$ with high probability.

Once we have $\widehat{R}$, we again write $\widehat{M}_{u,v}^{\widehat{R}}$ using Lemma 2 as

$$\widehat{M}_{u,v}^{\widehat{R}} = q_i c_i \beta_i \beta_i' A_{R,i}^* A_{R,i}^{*T} + \text{ perturbation terms} + (\widehat{M}_{u,v}^{\widehat{R}} - M_{u,v}^R),$$

and consider $\widehat{M}_{u,v}^{\widehat{R}} - M_{u,v}^R$ as an additional perturbation with the same magnitude $O^*(k/m \log n)$ in the sense of the spectral norm $\|\cdot\|$ w.h.p. Using the second part of Lemma 3, we have the top singular vectors of $\widehat{M}_{u,v}^{\widehat{R}}$ is $O^*(1/\log n)$ -close to $A_{R,i}^*$ with high probability.

Since every vector added to the list L in Algorithm 1 is close to one of the dictionary, then A^0 must be δ-close to A^*. In addition, the nearness ofA^0 to A^* is guaranteed via an appropriate projection onto the convex set $\mathcal{B} = \{A | A \text{ close to } A^0 \text{ and } \|A\| \leq 2\|A^*\|\}$. Finally, we finish the proof of Theorem 4. □

D.1.2. Proof of Lemma 12, Part a

For some fixed $l \in [n]$, consider p i.i.d. realizations $Z^{(1)}, Z^{(2)}, \ldots, Z^{(p)}$ of the random variable $Z \triangleq \langle y, u\rangle \langle y, v\rangle y_l^2$, then $\widehat{e}_l = \frac{1}{p}\sum_{i=1}^p Z^{(i)}$ and $e_l = \mathbb{E}[Z]$. To show that $\|\widehat{e} - e\|_\infty \leq O^*(k/m \log^2 n)$ holds with high probability, we first study the concentration for the l-th entry of $\widehat{e} - e$ and then take the union bound over all $l = 1, 2, \ldots, n$. We derive upper bounds for $|Z|$ and its variance $\mathbb{E}[Z^2]$ in order to apply Bernstein's inequality in (12) to the truncated version of Z.

Claim 8 *$|Z| \leq \widetilde{O}(k)$ and $\mathbb{E}[Z^2] \leq \widetilde{O}(k^2/m)$ with high probability.*

Again, the expectation is taken over y by conditioning on u and v, and therefore is still random due to the randomness of u and v. To show Claim 8, we begin with proving the following auxiliary claim.

Claim 9 *$\|y\| \leq \widetilde{O}(\sqrt{k})$ and $|\langle y, u\rangle| \leq \widetilde{O}(\sqrt{k})$ with high probability.*

Proof From the generative model, we have

$$\|y\| = \|A_{\bullet S}^* x_S^* + \varepsilon\| \leq \|A_{\bullet S}^* x_S^*\| + \|\varepsilon\| \leq \|A_{\bullet S}^*\| \|x_S^*\| + \|\varepsilon\|,$$

where $S = \text{supp}(x^*)$. From Claim 2, $\|x_S^*\| \leq \widetilde{O}(\sqrt{k})$ and $\|\varepsilon\| \leq \widetilde{O}(\sigma_\varepsilon \sqrt{n})$ w.h.p. In addition, A^* is overcomplete and has bounded spectral norm, then $\|A_{\bullet S}^*\| \leq \|A^*\| \leq O(1)$. Therefore, $\|y\| \leq \widetilde{O}(\sqrt{k})$ w.h.p., which is the first part of the proof. To bound the second term, we write it as

$$|\langle y, u\rangle| = |\langle A_{\bullet S}^* x_S^* + \varepsilon, u\rangle| \leq |\langle x_S^*, A_{\bullet S}^{*T} u\rangle| + |\langle \varepsilon, u\rangle|.$$

Similar to y, we have $\|u\| \leq \widetilde{O}(\sqrt{k})$ w.h.p. and hence $\|A_{\bullet S}^{*T} u\| \leq \|A_{\bullet S}^{*T}\| \|u\| \leq O(\sqrt{k})$ with high probability. Since u and x^* are independent sub-Gaussian and $\langle x_S^*, A_{\bullet S}^{*T} u\rangle$ are sub-exponential with variance at most $O(\sqrt{k})$, $|\langle x_S^*, A_{\bullet S}^{*T} u\rangle| \leq \widetilde{O}(k)$ w.h.p. Similarly, $|\langle \varepsilon, u\rangle| \leq \widetilde{O}(\sqrt{k})$ w.h.p. Consequently, $|\langle y, u\rangle| \leq \widetilde{O}(\sqrt{k})$ w.h.p., and we conclude the proof of the claim. ■

Proof [Proof of Claim 8] We have $Z = \langle y, u\rangle \langle y, v\rangle y_l^2 = \langle y, u\rangle \langle y, v\rangle (\langle A_{l\bullet}^*, x^*\rangle + \varepsilon_l)^2$ with $\langle y, u\rangle \langle y, v\rangle \leq \widetilde{O}(k)$ w.h.p. according to Claim 9. What remains is to bound $y_l^2 = (\langle A_{l\bullet}^*, x^*\rangle + \varepsilon_l)^2$. Because $\langle A_{l\bullet}^*, x^*\rangle$ is sub-Gaussian with variance $\mathbb{E}_S(\sum_{i\in S} A_{li}^{*2}) \leq \|A^{*T}\|_{1,2}^2 = O(1)$, then $|\langle A_{l\bullet}^*, x^*\rangle| \leq O(\log n)$ w.h.p. Similarly for ε_l, $|\varepsilon_l| \leq O(\sigma_\varepsilon \log n)$ w.h.p. Ultimately, $|\langle A_{l\bullet}^*, x^*\rangle + \varepsilon_l| \leq O(\log n)$, and hence we obtain with high probability the bound $|Z| \leq \widetilde{O}(k)$.

To bound the variance term, we write $Z^2 = \langle y, v\rangle^2 y_l^2 \langle y, u\rangle^2 y_l^2$. Note that, from the first part, we get $\langle y, v\rangle^2 y_l^2 \leq \widetilde{O}(k)$ and $|Z| \leq \widetilde{O}(k)$ w.h.p.. We apply Lemma 11 with some appropriate scaling to both terms, then

$$\mathbb{E}[Z^2(1 - \mathbf{1}_{|Z| \geq \widetilde{\Omega}(k)})] \leq \widetilde{O}(k)\mathbb{E}[\langle y, u\rangle^2 y_l^2] + O(n^{-\omega(1)}),$$

where $\mathbb{E}[\langle y, u\rangle^2 y_l^2]$ is equal to e_l for pair u, v with $v = u$. From Lemma 1 and its proof in Appendix Section "Analysis of Initialization Algorithm",

$$\mathbb{E}[\langle y, u\rangle^2 y_l^2] = \sum_{i=1}^m q_i c_i \beta_i^2 A_{li}^{*2} + \text{ perturbation terms},$$

in which the perturbation terms are bounded by $O^*(k/m \log^2 n)$ w.h.p. (following Claims 4 and 5). The dominant term $\sum_i q_i c_i \beta_i^2 A_{li}^{*2} \leq (\max q_i c_i \beta_i^2) \|A_{l\bullet}^*\|^2 \leq \widetilde{O}(k/m)$ w.h.p. because $|\beta_i| \leq O(\log m)$ (Claim 3). Then we complete the proof of the second part. ■

Proof [Proof of Lemma 12, Part a] We are now ready to prove Part a of Lemma 12. We apply Bernstein's inequality in Lemma 9 for the truncated random variable $Z^{(i)}(1 - \mathbf{1}_{|Z^{(i)}| \geq \widetilde{\Omega}(\mathcal{R})})$ with $\mathcal{R} = \widetilde{O}(k)$ and variance $\sigma^2 = \widetilde{O}(k^2/m)$ from Claim 8, then

$$\left\| \frac{1}{p} \sum_{i=1}^p Z^{(i)}(1 - \mathbf{1}_{|Z^{(i)}| \geq \widetilde{\Omega}(\mathcal{R})}) - \mathbb{E}[Z(1 - \mathbf{1}_{|Z| \geq \widetilde{\Omega}(\mathcal{R})})] \right\| \leq \frac{\widetilde{O}(k)}{p} + \sqrt{\frac{\widetilde{O}(k^2/m)}{p}} \leq O^*(k/m \log n), \tag{11}$$

w.h.p. for $p = \widetilde{\Omega}(m)$. Then $\widehat{e}_l = \frac{1}{p}\sum_{i=1}^p Z^{(i)}$ also concentrates with high probability. Take the union bound over $l = 1, 2, \ldots, n$, we get $\|\widehat{e} - e\|_\infty \leq O^*(k/m \log n)$ with high probability and complete the proof of 12, Part a. □

D.1.3. Proof of Lemma 12, Part b

Next, we will prove that $\|\widehat{M}_{u,v}^{\widehat{R}} - M_{u,v}^R\| \leq O^*(k/m \log n)$ with high probability. We only need to prove the concentration inequalities for the case when conditioned on the event that $\widehat{R}$ is equivalent to R w.h.p. Again, what we need to derive are an upper norm bound $\mathcal{R}$ of the matrix random variable $Z \triangleq \langle y, u\rangle \langle y, v\rangle y_R y_R^T$ and its variance.

Claim 10 *$\|Z\| \leq \widetilde{O}(kr)$ and $\|\mathbb{E}[ZZ^T]\| \leq \widetilde{O}(k^2 r/m)$ hold with high probability.*

Proof We have $\|Z\| \leq |\langle y, u\rangle \langle y, v\rangle| \|y_R\|^2$ with $|\langle y, u\rangle \langle y, v\rangle| \leq \widetilde{O}(k)$ w.h.p. (according to Claim 9) whereas $\|y_R\|^2 = \sum_{i\in R} y_l^2 \leq O(r \log^2 n)$ w.h.p. because $y_l \leq O(\log n)$ w.h.p. (proof of Claim 8). This implies $\|Z\| \leq \widetilde{O}(kr)$ w.h.p. The second part is handled similarly as in the proof of Claim 8. We take advantage of the bounds of $\widehat{M}_{u,v}$ in Lemma 2. Specifically, using the first part $\|Z\| \leq \widetilde{O}(kr)$ and $\langle y, v\rangle^2 \|y_R\|^2 \leq \widetilde{O}(kr)$, and applying Lemma 11, then

$$\|\mathbb{E}[ZZ^T(1 - \mathbf{1}_{\|Z\| \geq \widetilde{\Omega}(kr)})]\| \leq \widetilde{O}(kr)\|\mathbb{E}[\langle y, u\rangle^2 y_R y_R^T]\| + \widetilde{O}(kr) O(n^{-\omega(1)}) \leq \widetilde{O}(kr)\|M_{u,u}\|,$$

where $M_{u,u}$ arises from the application of Lemma 2. Recall that

$$M_{u,u} = \sum_i q_i c_i \beta_i^2 A^*_{R,i} A^{*T}_{R,i} + \text{ perturbation terms},$$

where the perturbation terms are all bounded by $O^*(k/m \log n)$ w.h.p. by Claims 6 and 7. In addition,

$$\|\sum_i q_i c_i \beta_i^2 A^*_{R,i} A^{*T}_{R,i}\| \leq (\max_i q_i c_i \beta_i^2) \|A^*_{R\bullet}\|^2 \leq \widetilde{O}(k/m)\|A^*\|^2 \leq \widetilde{O}(k/m)$$

w.h.p. Finally, the variance bound is $\widetilde{O}(k^2 r/m)$ w.h.p. ■

Then, applying Bernstein's inequality in Lemma 9 to the truncated version of Z with $\mathcal{R} = \widetilde{O}(kr)$ and variance $\sigma^2 = \widetilde{O}(k^2 r/m)$ and obtain the concentration for the full Z to get

$$\|\widehat{M}^R_{u,v} - M^R_{u,v}\| \leq \frac{\widetilde{O}(kr)}{p} + \sqrt{\frac{\widetilde{O}(k^2 r/m)}{p}} \leq O^*(k/m \log n)$$

w.h.p. when the number of samples is $p = \widetilde{\Omega}(mr)$ under Assumption **A4.1**.

We have proved that $\|\widehat{M}^R_{u,v} - M^R_{u,v}\| \leq O^*(k/m \log n)$ as conditioned on the support consistency event holds w.h.p. $\|\widehat{M}^{\widehat{R}}_{u,v} - M^R_{u,v}\| \leq O^*(k/m \log n)$ is easily followed by the law of total probability through the tail bounds on the conditional and marginal probabilities (i.e. $\mathbb{P}[\|\widehat{M}^R_{u,v} - M^R_{u,v}\| \leq O^*(k/m \log n) | \widehat{R} = R]$) and $\mathbb{P}[\widehat{R} \neq R]$. We finish the proof of Lemma 12, Part b for both cases of the spectral bounds. ■

D.2. Proof of Theorem 5 and Sample Complexity of Algorithm 2

In this section, we prove Theorem 5 and identify sample complexity per iteration of Algorithm 2. We divide the proof into two steps: 1) show that when A^s is $(\delta_s, 2)$-near to A^* for $\delta_s = O^*(1/\log n)$, the approximate gradient estimate $\widehat{g}^s$ is $(\alpha, \beta, \gamma_s)$-correlated-whp with A^* with $\gamma_s \leq O(k^2/mn) + \alpha o(\delta_s^2)$, and 2) show that the nearness is preserved at each iteration. These correspond to showing the following lemmas:

Lemma 13 *At iteration s of Algorithm 2, suppose that A^s has each column correctly supported and is $(\delta_s, 2)$-near to A^* and that $\eta = O(m/k)$. Denote $R = \text{supp}(A^s_{\bullet i})$, then the update $\widehat{g}^s_{R,i}$ is $(\alpha, \beta, \gamma_s)$-correlated-whp with $A^*_{R,i}$ where $\alpha = \Omega(k/m)$, $\beta = \Omega(m/k)$ and $\gamma_s \leq O(k^2/mn) + \alpha o(\delta_s^2)$ for $\delta_s = O^*(1/\log n)$.*

Note that this is a finite-sample version of Lemma 6.

Lemma 14 *If A^s is $(\delta_s, 2)$-near to A^* and number of samples used in step s is $p = \widetilde{\Omega}(m)$, then with high probability $\|A^{s+1} - A^*\| \leq 2\|A^*\|$.*

Proof [Proof of Theorem 5] The correlation of $\widehat{g}_i$ with A^*_i, described in Lemma 13, implies the descent of column-wise error according to Theorem 1. Along with Lemma 14, the theorem follows directly.

D.2.1. Proof of Lemma 13

We prove Lemma 13 by obtaining a tail bound on the difference between $\widehat{g}^s_{R,i}$ and $g^s_{R,i}$ using the Bernstein's inequality in Lemma 9.

Lemma 15 *At iteration s of Algorithm 2, suppose that A^s has each column correctly supported and is $(\delta_s, 2)$-near to A^*. For $R = \text{supp}(A^s_i) = \text{supp}(A^*_i)$, then $\|\widehat{g}^s_{R,i} - g^s_{R,i}\| \leq O(k/m) \cdot (o(\delta_s) + O(\epsilon_s))$ with high probability for $\delta_s = O^*(1/\log n)$ and $\epsilon_s = O(\sqrt{k/n})$ when $p = \widetilde{\Omega}(m + \sigma_\varepsilon^2 \frac{mnr}{k})$.*

To prove this lemma, we study the concentration of $\widehat{g}^s_{R,i}$, which is a sum of random vector of the form $(y - Ax)_R \text{sgn}(x_i)$. We consider random variable $Z \triangleq (y - Ax)_R \text{sgn}(x_i) | i \in S$, with $S = \text{supp}(x^*)$ and $x = \text{threshold}_{C/2}(A^T y)$. Then, using the following technical lemma to bridge the gap in concentration of the two variables. We adopt this strategy from Arora et al. (2015) for our purpose.

Claim 11 *Suppose that $Z^{(1)}, Z^{(2)}, \ldots, Z^{(N)}$ are i.i.d. samples of the random variable $Z = (y - Ax)_R \text{sgn}(x_i) | i \in S$. Then,*

$$\Big\| \frac{1}{N} \sum_{j=1}^N Z^{(j)} - \mathbb{E}[Z] \Big\| \leq o(\delta_s) + O(\epsilon_s) \tag{12}$$

holds with probability when $N = \widetilde{\Omega}(k + \sigma_\varepsilon^2 nr)$, $\delta_s = O^*(1/\log n)$ and $\epsilon_s = O(\sqrt{k/n})$.

Proof [Proof of Lemma 15] Once we have done the proof of Claim 11, we can easily prove Lemma 15. We recycle the proof of Lemma 43 in Arora et al. (2015).

Write $W = \{j : i \in \text{supp}(x^{*(j)})\}$ and $N = |W|$, then express $\widehat{g}_{R,i}$ as

$$\widehat{g}_{R,i} = \frac{N}{p} \frac{1}{N} \sum_j (y^{(j)} - Ax^{(j)})_R \text{sgn}(x_i^{(j)}),$$

where $\frac{1}{|W|} \sum_j (y^{(j)} - Ax^{(j)})_R \text{sgn}(x_i^{(j)})$ is distributed as $\frac{1}{N} \sum_{j=1}^N Z^{(j)}$ with $N = |W|$. Note that $\mathbb{E}[(y - Ax)_R \text{sgn}(x_i)] = \mathbb{E}[(y - Ax)_R \text{sgn}(x_i) \mathbf{1}_{i \in S}] = \mathbb{E}[Z]\mathbb{P}[i \in S] = q_i \mathbb{E}[Z]$ with $q_i = \Theta(k/m)$. Following Claim 11, we have

$$\|\widehat{g}^s_{R,i} - g^s_{R,i}\| \leq O(k/m) \Big\| \frac{1}{N} \sum_{j=1}^N Z^{(j)} - \mathbb{E}[Z] \Big\| \leq O(k/m) \cdot (o(\delta_s) + O(\epsilon_s)),$$

holds with high probability as $p = \Omega(mN/k)$. Substituting N in Claim 11, we obtain the results in Lemma 15. □

Proof [Proof of Claim 11] We are now ready to prove the claim. What we need are good bounds for $\|Z\|$ and its variance, then we can apply Bernstein's inequality in Lemma 9 for the truncated version of Z, then Z is also concentrates likewise.

Claim 12 *$\|Z\| \leq \mathcal{R}$ holds with high probability for $\mathcal{R} = \widetilde{O}(\delta_s \sqrt{k} + \mu k/\sqrt{n} + \sigma_\varepsilon \sqrt{r})$ with $\delta_s = O^*(1/\log n)$.*

Proof From the generative model and the support consistency of the encoding step, we have $y = A^* x^* + \varepsilon = A^*_{\bullet S} x^*_S + \varepsilon$ and $x_S = A^T_{\bullet S} y = A^T_{\bullet S} A^*_{\bullet S} x^*_S + A^T_{\bullet S}\varepsilon$. Then,

$$\begin{aligned}(y - Ax)_R &= (A^*_{R,S} x^*_S + \varepsilon_R) - A_{R,S} A^T_{\bullet S} A^*_{\bullet S} x^*_S - A_{R,S} A^T_{\bullet S}\varepsilon \\ &= (A^*_{R,S} - A_{R,S}) x^*_S + A_{R,S}(I_k - A^T_{\bullet S} A^*_{\bullet S}) x^*_S + (I_n - A_{\bullet S} A^T_{\bullet S})_{R\bullet}\varepsilon.\end{aligned}$$

Using the fact that x^*_S and ε are sub-Gaussian and that $\|Mw\| \le \widetilde{O}(\sigma_w \|M\|_F)$ holds with high probability for a fixed M and a sub-Gaussian w of variance σ_w^2, we have

$$\|(y - Ax)_R \mathrm{sgn}(x_i)\| \le \widetilde{O}(\|A^*_{R,S} - A_{R,S}\|_F + \|A_{R,S}(I_k - A^T_{\bullet S} A^*_{\bullet S})\|_F + \sigma_\varepsilon \|(I_n - A_{\bullet S} A^T_{\bullet S})_{R\bullet}\|_F).$$

Now, we need to bound those Frobenius norms. The first quantity is easily bounded as

$$\|A^*_{R,S} - A_{R,S}\|_F \le \|A^*_{\bullet S} - A_{\bullet S}\|_F \le \delta_s \sqrt{k}, \tag{13}$$

since A is δ_s-close to A^*. To handle the other two, we use the fact that $\|UV\|_F \le \|U\|\|V\|_F$. Using this fact for the second term, we have

$$\|A_{R,S}(I_k - A^T_{\bullet S} A^*_{\bullet S})\|_F \le \|A_{R,S}\| \|(I_k - A^T_{\bullet S} A^*_{\bullet S})\|_F,$$

where $\|A_{R,S}\| \le \|A_{R\bullet}\| \le O(1)$ due to the nearness. The second part is rearranged to take advantage of the closeness and incoherence properties:

$$\begin{aligned}\|I_k - A^T_{\bullet S} A^*_{\bullet S}\|_F &\le \|I_k - A^{*T}_{\bullet S} A^*_{\bullet S} - (A_{\bullet S} - A^*_{\bullet S})^T A^*_{\bullet S}\|_F \\ &\le \|I_k - A^{*T}_{\bullet S} A^*_{\bullet S}\|_F + \|(A_{\bullet S} - A^*_{\bullet S})^T A^*_{\bullet S}\|_F \\ &\le \|I_k - A^{*T}_{\bullet S} A^*_{\bullet S}\|_F + \|A^*_{\bullet S}\| \|A_{\bullet S} - A^*_{\bullet S}\|_F \\ &\le \mu k/\sqrt{n} + O(\delta_s \sqrt{k}),\end{aligned}$$

where we have used $\|I_k - A^{*T}_{\bullet S} A^*_{\bullet S}\|_F \le \mu k/\sqrt{n}$ because of the μ-incoherence of A^*, $\|A_{\bullet S} - A^*_{\bullet S}\|_F \le \delta_s \sqrt{k}$ in (13) and $\|A^*_{\bullet S}\| \le \|A^*\| \le O(1)$. Accordingly, the second Frobenius norm is bounded by

$$\|A_{R,S}(I_k - A^T_{\bullet S} A^*_{\bullet S})\|_F \le O\big(\mu k/\sqrt{n} + \delta_s \sqrt{k}\big). \tag{14}$$

The noise term is handled using the eigen-decomposition $U \Lambda U^T$ of $A_{\bullet S} A^T_{\bullet S}$, then with high probability

$$\|(I_n - A_{\bullet S} A^T_{\bullet S})_{R\bullet}\|_F = \|(UU^T - U\Lambda U^T)_{R\bullet}\|_F = \|U_{R\bullet}(I_n - \Lambda)\|_F \le \|I_n - \Lambda\| \|U_{R\bullet}\|_F \le O(\sqrt{r}), \tag{15}$$

where the last inequality $\|I_n - \Lambda\| \le O(1)$ follows by $\|A_{\bullet S}\| \le \|A\| \le \|A - A^*\| + \|A^*\| \le 3\|A^*\| \le O(1)$ due to the nearness. Putting (13), (14) and (15) together, we obtain the bounds in Claim 12. ■

Next, we determine a bound for the variance of Z.

Claim 13 $\mathbb{E}[\|Z\|^2] = \mathbb{E}[\|(y - Ax)_R \mathrm{sgn}(x_i)\|^2 | i \in S] \le \sigma^2$ *holds with high probability for* $\sigma^2 = O(\delta_s^2 k + k^2/n + \sigma_\varepsilon^2 r)$ *with* $\delta_s = O^*(1/\log n)$.

Proof We explicitly calculate the variance using the fact that x^*_S is conditionally independent given S, and so is ε. x^*_S and ε are also independent and have zero mean. Then we can decompose the norm into three terms in which the dot product is zero in expectation and the others can be shortened using the fact that $E[x^*_S x^{*T}_S] = I_k$, $E[\varepsilon\varepsilon^T] = \sigma_\varepsilon I_n$.

$$\begin{aligned}\mathbb{E}[\|(y - Ax)_R \mathrm{sgn}(x_i)\|^2 | i \in S] &= \mathbb{E}[\|(A^*_{R,S} - A_{R,S} A^T_{\bullet S} A^*_{\bullet S}) x^*_S + (I_n - A_{\bullet S} A^T_{\bullet S})_R . \varepsilon\|^2 | i \in S]] \\ &= \mathbb{E}[\|A^*_{R,S} - A_{R,S} A^T_{\bullet S} A^*_{\bullet S}\|_F^2 | i \in S] + \sigma_\varepsilon^2 \mathbb{E}[\|I_n - A_{\bullet S} A^T_{\bullet S})_{R\bullet}\|_F^2 | i \in S].\end{aligned}$$

Then, by re-writing $A^*_{R,S} - A_{R,S} A^T_{\bullet S} A^*_{\bullet S}$ as before, we get the form $(A^*_{R,S} - A_{R,S}) + A_{R,S}(I_k - A^T_{\bullet S} A^*_{\bullet S})$ in which the first term has norm bounded by $\delta_s \sqrt{k}$. The second is further decomposed as

$$\mathbb{E}[\|A_{R,S}(I_k - A^T_{\bullet S} A^*_{\bullet S})\|_F^2 | i \in S] \le \sup_S \|A_{R,S}\|^2 \mathbb{E}[\|I_k - A^T_{\bullet S} A^*_{\bullet S}\|_F^2 | i \in S], \tag{16}$$

where $\sup_S \|A_{R,S}\| \le \|A_{R\bullet}\| \le O(1)$. We will bound $\mathbb{E}[\|I_k - A^T_{\bullet S} A^*_{\bullet S}\|_F^2 | i \in S] \le O(k\delta_s^2) + O(k^2/n)$ using the proof from Arora et al. (2015):

$$\begin{aligned}\mathbb{E}[\|I_k - A^T_{\bullet S} A^*_{\bullet S}\|_F^2 | i \in S] &= \mathbb{E}[\sum_{j \in S} (1 - A^T_{\bullet j} A^*_{\bullet j})^2 + \sum_{j \in S} \|A^T_{\bullet j} A^*_{\bullet,-j}\|^2 | i \in S] \\ = \mathbb{E}[\sum_{j \in S} \frac{1}{4} \|A_{\bullet j} - A^*_{\bullet j}\|^2] &+ q_{ij} \sum_{j \ne i} \|A^T_{\bullet j} A^*_{\bullet,-j}\|^2 + q_i \|A^T_{\bullet i} A^*_{\bullet,-i}\|^2 + q_i \|A^T_{\bullet,-i} A^*_{\bullet i}\|^2,\end{aligned}$$

where $A_{\bullet,-i}$ is the matrix A with the i-th column removed, $q_{ij} \le O(k^2/m^2)$ and $q_i \le O(k/m)$. For any $j = 1, 2, \ldots, m$,

$$\begin{aligned}\|A^T_{\bullet j} A^*_{\bullet,-j}\|^2 &= \|A^{*T}_{\bullet j} A^*_{\bullet,-j} + (A_{\bullet j} - A^*_{\bullet j})^T A^*_{\bullet,-j}\|^2 \\ &\le \sum_{l \ne j} \langle A^*_{\bullet j}, A^*_{\bullet l}\rangle^2 + \|(A_{\bullet j} - A^*_{\bullet j})^T A^*_{\bullet,-j}\|^2 \\ &\le \sum_{l \ne j} \langle A^*_{\bullet j}, A^*_{\bullet l}\rangle^2 + \|A_{\bullet j} - A^*_{\bullet j}\|^2 \|A^*_{\bullet,-j}\|^2 \le \mu^2 + \delta_s^2.\end{aligned}$$

The last inequality invokes the μ-incoherence, δ-closeness and the spectral norm of A^*. Similarly, we come up with the same bound for $\|A^T_{\bullet i} A^*_{\bullet,-i}\|^2$ and $\|A^T_{\bullet,-i} A^*_{\bullet i}\|^2$. Consequently,

$$\mathbb{E}[\|I_k - A^T_{\bullet S} A^*_{\bullet S}\|_F^2 | i \in S] \le O(k\delta_s^2) + O(k^2/n). \tag{17}$$

For the last term, we invoke the inequality (15) (Claim 12) to get

$$\mathbb{E}[\|(I_n - A_{\bullet S} A^T_{\bullet S})_{R\bullet}\|_F^2 | i \in S] \le r \tag{18}$$

Putting (16), (17) and (18) together and using $\|A_{R\bullet}\| \le 1$, we obtain the variance bound of Z: $\sigma^2 = O(\delta_s^2 k + k^2/n + \sigma_\varepsilon^2 r)$ with $\delta_s = O^*(1/\log n)$. Finally, we complete the proof. □

We now apply truncated Bernstein's inequality to the random variable $Z^{(j)}(1 - 1_{\|Z^{(j)}\| \ge \Omega(\mathcal{R})})$ with $\mathcal{R}$ and σ^2 in Claims 12 and 13, which are $\mathcal{R} = \widetilde{O}(\delta_s \sqrt{k} + \mu k/\sqrt{n} + \sigma_\varepsilon \sqrt{r})$ and $\sigma^2 = O(\delta_s^2 k + k^2/n + \sigma_\varepsilon^2 r)$. Then, $(1/N) \sum_{j=}^N Z^{(j)}$ also concentrates:

$$\Big\| \frac{1}{N} \sum_{i=1}^N Z^{(j)} - E[Z] \Big\| \le \widetilde{O}\Big(\frac{\mathcal{R}}{N}\Big) + \widetilde{O}\Big(\sqrt{\frac{\sigma^2}{N}}\Big) = o(\delta_s) + O(\sqrt{k/n})$$

holds with high probability when $N = \widetilde{\Omega}(k + \sigma_\varepsilon^2 nr)$. Then, we finally finish the proof of Claim 11. ■

Proof [Proof of Lemma 13] With Claim 11, we study the concentration of $\widehat{g}^s_{R,i}$ around its mean $g^s_{R,i}$. Now, we consider this difference as an error term of the expectation $g^s_{R,i}$ and using Lemma 6 to show the correlation of $\widehat{g}^s_{R,i}$. Using the expression in Lemma 5 with high probability, we can write

$$\widehat{g}^s_{R,i} = g^s_{R,i} + (g^s_{R,i} - \widehat{g}^s_{R,i}) = 2\alpha(A_{R,i} - A^*_{R,i}) + v,$$

where $\|v\| \le \alpha\|A_{R,i} - A^*_{R,i}\| + O(k/m) \cdot (o(\delta_s) + O(\epsilon_s))$. By Lemma 6, we have $\widehat{g}^s_{R,i}$ is $(\alpha, \beta, \gamma_s)$-correlated-whp with $A^*_{R,i}$ where $\alpha = \Omega(k/m)$, $\beta = \Omega(m/k)$ and $\gamma_s \le O(k/m) \cdot (o(\delta_s) + O(\sqrt{k/n}))$, then we have done the proof Lemma 13. □

D.2.2. Proof of Lemma 14

We have shown the correlation of $\widehat{g}^s$ with A^* w.h.p. and established the descent property of Algorithm 2. The next step is to show that the nearness is preserved at each iteration. To prove $\|A^{s+1} - A^*\| \le 2\|A^*\|$ holds with high probability, we recall the update rule

$$A^{s+1} = A^s - \eta \mathcal{P}_H(\widehat{g}^s),$$

where $\mathcal{P}_H(\widehat{g}^s) = H \circ \widehat{g}^s$. Here $H = (h_{ij})$ where $h_{ij} = 1$ if $i \in \text{supp}(A_{\bullet j})$ and $h_{ij} = 0$ otherwise. Also, note that A^s is $(\delta_s, 2)$-near to A^* for $\delta_s = O^*(1/\log n)$. We already proved that this holds for the exact expectation g^s in Lemma 7. To prove for $\widehat{g}^s$, we again apply matrix Bernstein's inequality to bound $\|\mathcal{P}_H(g^s) - \mathcal{P}_H(\widehat{g}^s)\|$ by $O(k/m)$ because $\eta = \Theta(m/k)$ and $\|A^*\| = O(1)$.

Consider a matrix random variable $Z \triangleq \mathcal{P}_H((y - Ax)\text{sgn}(x)^T)$. Our goal is to bound the spectral norm $\|Z\|$ and, both $\|\mathbb{E}[ZZ^T]\|$ and $\|\mathbb{E}[Z^TZ]\|$ since Z is asymmetric. To simplify our notations, we denote by x_R the vector x by zeroing out the elements not in R. Also, denote $R_i = \text{supp}(h_i)$ and $S = \text{supp}(x)$. Then Z can be written explicitly as

$$Z = [(y - Ax)_{R_1}\text{sgn}(x_1), \ldots, (y - Ax)_{R_m}\text{sgn}(x_m)],$$

where many columns are zero since x is k-sparse. The following claims follow from the proof of Claim 42 in Arora et al. (2015). Here we state and detail some important steps.

Claim 14 $\|Z\| \le \widetilde{O}(k)$ *holds with high probability.*

Proof With high probability

$$\|Z\| \le \sqrt{\sum_{i \in S} \|(y - Ax)_{R_i}\text{sgn}(x_i)\|^2} \le \sqrt{k}\|(y - Ax)_{R_i}\|$$

where we use Claim 12 with $\|(y - Ax)_R\| \le \widetilde{O}(\delta_s\sqrt{k})$ w.h.p., then $\|Z\| \le \widetilde{O}(k)$ holds w.h.p. ■

Claim 15 $\|\mathbb{E}[ZZ^T]\| \le O(k^2/n)$ *and* $\|\mathbb{E}[Z^TZ]\| \le \widetilde{O}(k^2/n)$ *with high probability.*

Proof The first term is easily handled. Specifically, with high probability

$$\|\mathbb{E}[ZZ^T]\| \le \|\mathbb{E}[\sum_{i \in S}(y - Ax)_{R_i}\text{sgn}(x_i)^2(y - Ax)^T_{R_i}]\| = \|\mathbb{E}[\sum_{i \in S}(y - Ax)_{R_i}(y - Ax)^T_{R_i}]\| \le O(k^2/n),$$

where the last inequality follows from the proof of Claim 42 in Arora et al. (2015), which is tedious to be repeated.

To bound $\|\mathbb{E}[Z^TZ]\|$, we use bound of the full matrix $(y - Ax)\text{sgn}(x)^T$. Note that $\|y - Ax\| \le \widetilde{O}(\sqrt{k})$ w.h.p. is similar to what derived in Claim 12. Then with high probability,

$$\|\mathbb{E}[Z^TZ]\| \le \|\mathbb{E}[\text{sgn}(x)(y - Ax)^T(y - Ax)\text{sgn}(x)^T]\| \le \widetilde{O}(k)\|\mathbb{E}[\text{sgn}(x)\text{sgn}(x)^T]\| \le \widetilde{O}(k^2/m).$$

where $\mathbb{E}[\text{sgn}(x)\text{sgn}(x)^T] = \text{diag}(q_1, q_2, \ldots, q_m)$ has norm bounded by $O(k/m)$. We now can apply Bernstein's inequality for the truncated version of Z with $\mathcal{R} = \widetilde{O}(k)$ and $\sigma^2 = \widetilde{O}(k^2/m)$, then with $p = \widetilde{O}(m)$,

$$\|\mathcal{P}_H(g^s) - \mathcal{P}_H(\widehat{g}^s)\| \le \frac{\widetilde{O}(k)}{p} + \sqrt{\frac{\widetilde{O}(k^2/m)}{p}} \le O^*(k/m)$$

holds with high probability. Finally, we invoke the bound $\eta = O(m/k)$ and complete the proof. ■

Appendix E. A Special Case: Orthonormal A^*

We extend our results for the special case where the dictionary is orthonormal. As such, the dictionary is perfectly incoherent and bounded (i.e., $\mu = 0$ and $\|A^*\| = 1$).

Theorem 7 *Suppose that A^* is orthonormal. When $p_1 = \widetilde{\Omega}(n)$ and $p_2 = \widetilde{\Omega}(nr)$, then with high probability Algorithm 1 returns an initial estimate A^0 whose columns share the same support as A^* and with $(\delta, 2)$-nearness to A^* with $\delta = O^*(1/\log n)$. The sparsity of A^* can be achieved up to $r = O^*\big(\min(\frac{\sqrt{n}}{\log^2 n}, \frac{n}{k^2 \log^2 n})\big)$.*

We use the same initialization procedure for this special case and achieve a better order of r. The proof of Theorem 7 follows the analysis for the general case with following two results:

Claim 16 (Special case of Claim 3) *Suppose that $u = A^*\alpha + \varepsilon_u$ is a random sample and $U = \text{supp}(\alpha)$. Let $\beta = A^{*T}u$, then w.h.p., we have (a) $|\beta_i - \alpha_i| \le \sigma_\varepsilon \log n$ for each i and (b) $\|\beta\| \le O(\sqrt{k}\log n + \sigma_\varepsilon\sqrt{n}\log n)$.*

Proof We have $\beta = A^{*T}u = \alpha + A^{*T}\epsilon_u$, then $\beta_i - \alpha_i = \langle A^*_{\bullet i}, \epsilon_u\rangle$ and $\|\beta - \alpha\| = \|\epsilon_u\|$. Using probability bounds of $\langle A^*_{\bullet i}, \epsilon_u\rangle$, $\|\epsilon_u\|$ and $\|\alpha\|$ in Claim 2, we have the claim proved. □

We draw from the claim that for any $i \notin U \cap V$, $|\beta_i\beta'_i| \le O(\sigma_\varepsilon \log^2 n)$ and have the following result:

Lemma 16 *Fix samples u and v and suppose that $y = A^* x^* + \varepsilon$ is a random sample independent of u, v. The expected value of the score for the lth component of y is given by:*

$$e_l \triangleq \mathbb{E}[\langle y, u\rangle\langle y, v\rangle y_l^2] = \sum_{i \in U \cap V} q_i c_i \beta_i \beta_i' A_{li}^{*2} + \text{ perturbation terms}$$

where $q_i = \mathbb{P}[i \in S]$, $q_{ij} = \mathbb{P}[i, j \in S]$ and $c_i = \mathbb{E}[x_i^4 | i \in S]$. Moreover, the perturbation terms have absolute value at most $O^*\big(k/n \log^2 n \max(1/\sqrt{n}, k^2/n)\big)$.

Proof Lemma follows Lemma 1 via Claim 3 except that the second term of E_1 is bounded by $O(k \log^2 n / n^{3/2})$.

Appendix F. Extensions of Arora et al. (2015)

F.1. Sample complexity in noisy case

In this section, we study the sample complexity of the algorithms in Arora et al. (2015) in the presence of noise. While noise with order $\sigma_\varepsilon = O(1/\sqrt{n})$ does not change the sample complexity of the initialization algorithm, it affects that of the descent stage. The analysis involves producing a sharp bound for $\|\widehat{g}^s_{\bullet,i} - g^s_{\bullet i}\|$.

Lemma 17 *For a regular dictionary A^*, suppose A^s is $(\delta_s, 2)$-near to A^* with $\delta_s = O^*(1/\log n)$, then with high probability $\|\widehat{g}^s_{\bullet,i} - g^s_{\bullet i}\| \le O(k/m) \cdot (o(\delta) + O(\sqrt{k/n}))$ when $p = \widetilde{\Omega}(m + \sigma_\varepsilon^2 \frac{mn^2}{k})$.*

Proof This follows directly from Lemma 15 where $r = n$. □

We tighten the original analysis to obtain the complexity $\widetilde{\Omega}(m)$ instead of $\widetilde{\Omega}(mk)$ for the noiseless case. Putting together with $p = \widetilde{\Omega}(mk)$ required by the initialization, we then have the overall sample complexity $\widetilde{O}(mk + \sigma_\varepsilon^2 \frac{mn^2}{k})$ for the algorithms in Arora et al. (2015) in the noise regime.

F.2. Extension of Arora et al. (2015)'s initialization algorithm for sparse case

We study a simple and straightforward extension of the initialization algorithm of Arora et al. (2015) for the sparse case. This extension is produced by adding an extra projection, and is described in Figure 3. The recovery of the support of A^* is guaranteed by the following Lemma:

Lemma 18 *Suppose that $z^* \in \mathbb{R}^n$ is r-sparse whose nonzero entries are at least τ in magnitude. Provided z is δ-close to z^* and $z^0 = \mathcal{H}_r(z)$ with $\delta = O^*(1/\log n)$ and $r = O^*(\log^2 n)$, then z^0 and z^* has the same support.*

Proof Since z^0 is δ-close to z^*, then $\|z^0 - z^*\| \le \delta$ and $|z_i - z_i^*| \le \delta$ for every i. For $i \in \text{supp}(z^*)$,

$$|z_i| \ge |z_i^*| - |z_i - z_i^*| \ge \tau - \delta$$

and for $i \notin \text{supp}(z^*)$, $|z_i| \le \delta$. Since $\tau > O(1/\sqrt{r}) \gg \delta$, then the r-largest entries of z are in the support z^*, and hence z^0 and z^* has the same support. □

Algorithm 3 Pairwise Reweighting with Hard-Thresholding

Initialize $L = \emptyset$
Randomly divide p samples into two disjoint sets $\mathcal{P}_1$ and $\mathcal{P}_2$ of sizes p_1 and p_2 respectively
While $|L| < m$. Pick u and v from $\mathcal{P}_1$ at random
  Construct the re-weighted covariance matrix $\widehat{M}_{u,v}$:

$$\widehat{M}_{u,v} = \frac{1}{p_2} \sum_{i=1}^{p_2} \langle y^{(i)}, u\rangle \langle y^{(i)}, v\rangle y^{(i)} (y^{(i)})^T$$

  Compute the top singular values δ_1, δ_2 and top singular vector z of $\widehat{M}_{u,v}$
  If $\delta_1 \ge \Omega(k/m)$ and $\delta_2 < O^*(k/m \log n)$
    $z = \mathcal{H}_r(z)$, where $\mathcal{H}_r$ keeps r largest entries of z
    If z is not within distance $1/\log n$ of any vector in L even with sign flip
      $L = L \cup \{z\}$
Return $A^0 = (L_1, \ldots, L_m)$

Theorem 8 *Suppose that Assumptions* **B1-B4** *hold and Assumptions* **A1-A3** *satify with $\mu = O^*\big(\frac{\sqrt{n}}{k \log^3 n}\big)$ and $r = O^*(\log^2 n)$. When $p_1 = \widetilde{\Omega}(m)$ and $p_2 = \widetilde{\Omega}(mk)$, then with high probability Algorithm 3 returns an initial estimate A^0 whose columns share the same support as A^* and with $(\delta, 2)$-nearness to A^* with $\delta = O^*(1/\log n)$.*

This algorithm requires $r = O^*(\log^2 n)$, which is somewhat better than ours. However, the sample complexity and running time is inferior as compared with our novel algorithm.

Appendix G. Neural Implementation of Our Approach

We now briefly describe why our algorithm is "neurally plausible". Basically, similar to the argument in Arora et al. (2015), we describe at a very high level how our algorithm can be implemented via a neural network architecture. One should note that although both our initialization and descent stages are non-trivial modifications of those in Arora et al. (2015), both still inherit the nice neural plausiblity property.

G.1. Neural implementation of Stage 1: Initialization

Recall that the initialization stage includes two main steps: (i) estimate the support of each column of the synthesis matrix, and (ii) compute the top principal component(s) of a certain truncated weighted covariance matrix. Both steps involve simple vector and matrix-vector manipulations that can be implemented plausibly using basic neuronal manipulations.

For the support estimation step, we compute the product $\langle y, u\rangle\langle y, u\rangle y \circ y$, followed by a thresholding. The inner products, $\langle y, u\rangle$ and $\langle y, v\rangle$ can be computed using neurons via an online manner where the samples arrive in sequence; the thresholding can be implemented via a ReLU-type non-linearity.

For the second step, it is well known that the top principal components of a matrix can be computed in a neural (Hebbian) fashion using Oja's Rule Oja (1992).

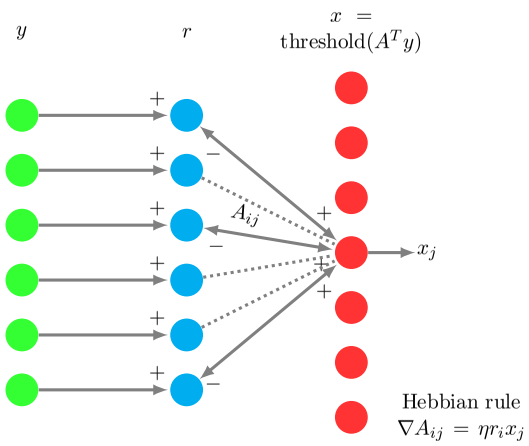


Figure 4: Neural network implementation of Algorithm 2. The network takes the image y as input and produces the sparse representation x as output. The hidden layer represents the residual between the image and its reconstruction Ax. The weights A_{ij}'s are stored on synapses, but most of them are zero and shown by the dotted lines.

G.2. Neural implementation of Stage 2: Descent

Our neural implementation of the descent stage (Algorithm 2), shown in Figure 4, mimics the architecture of Arora et al. (2015), which describes a simple two-layer network architecture for computing a single gradient update of A. The only difference in our case is that most of the value in A are set to zero, or in other words, our network is sparse. The network takes values y from the input layer and produce x as the output; there is an intermediate layer in between connecting the middle layer with the output via synapses. The synaptic weights are stored on A. The weights are updated by Hebbian learning. In our case, since A is sparse (with support given by R, as estimated in the first stage), we enforce the condition the corresponding synapses are inactive. In the output layer, as in the initialization stage, the neurons can use a ReLU-type non-linear activation function to enforce the sparsity of x.

References

Alekh Agarwal, Animashree Anandkumar, Prateek Jain, Praneeth Netrapalli, and Rashish Tandon. Learning sparsely used overcomplete dictionaries. In *Conference on Learning Theory*, pages 123–137, 2014.

Michal Aharon, Michael Elad, and Alfred Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. More algorithms for provable dictionary learning. *arXiv preprint arXiv:1401.0579*, 2014a.

Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *Conference on Learning Theory*, pages 779–806, 2014b.

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on Learning Theory*, pages 113–149, 2015.

Jarosław Błasiok and Jelani Nelson. An improved analysis of the er-spud dictionary learning algorithm. *arXiv preprint arXiv:1602.05719*, 2016.

Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2559–2566. IEEE, 2010.

Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

Niladri Chatterji and Peter Bartlett. Alternating minimization for dictionary learning with random initialization. 2017. arXiv:1711.03634v1.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 2443–2446. IEEE, 1999.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 399–406, 2010.

Rémi Gribonval, Rodolphe Jenatton, and Francis Bach. Sparse and spurious: dictionary learning with noise and outliers. *IEEE Transactions on Information Theory*, 61(11): 6298–6319, 2015a.

Rémi Gribonval, Rodolphe Jenatton, Francis Bach, Martin Kleinsteuber, and Matthias Seibert. Sample complexity of dictionary learning and other matrix factorizations. *IEEE Transactions on Information Theory*, 61(6):3469–3486, 2015b.

Hamid Krim, Dewey Tucker, Stephane Mallat, and David Donoho. On denoising and best signal representation. *IEEE Transactions on Information Theory*, 45(7):2225–2238, 1999.

Rados law Adamczak. A note on the sample complexity of the er-spud algorithm by spielman, wang and wright for exact recovery of sparsely used dictionaries. *Journal of Machine Learning Research*, 17:1–18, 2016.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 689–696, 2009.

Arya Mazumdar and Ankit Singh Rawat. Associative memory using dictionary learning and expander decoding. In *Proc. Conf. American Assoc. Artificial Intelligence (AAAI)*, pages 267–273, 2017.

Erkki Oja. Principal components, minor components, and linear neural networks. *Neural networks*, 5(6):927–935, 1992.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010a.

Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58 (3):1553–1564, 2010b.

Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory*, pages 37–1, 2012.

Jeremias Sulam, Boaz Ophir, Michael Zibulevsky, and Michael Elad. Trainlets: Dictionary learning in high dimensions. *IEEE Transactions on Signal Processing*, 64(12):3180–3193, 2016.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery using nonconvex optimization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2351–2360, 2015.

Lingxiao Wang, Xiao Zhang, and Quanquan Gu. A unified computational and statistical framework for nonconvex low-rank matrix estimation. *arXiv preprint arXiv:1610.05275*, 2016.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(1):3537–3580, 2016.

Journal of Machine Learning Research 20 (2019) 1-39 Submitted 12/17; Revised 4/19; Published 7/19

Model-free Nonconvex Matrix Completion: Local Minima Analysis and Applications in Memory-efficient Kernel PCA

Ji Chen LJICHEN@UCDAVIS.EDU
Department of Mathematics
University of California, Davis
Davis, CA 95616, USA

Xiaodong Li XDGLI@UCDAVIS.EDU
Department of Statistics
University of California, Davis
Davis, CA 95616, USA

Editor: Benjamin Recht

Abstract

This work studies low-rank approximation of a positive semidefinite matrix from partial entries via nonconvex optimization. We characterized how well local-minimum based low-rank factorization approximates a fixed positive semidefinite matrix without any assumptions on the rank-matching, the condition number or eigenspace incoherence parameter. Furthermore, under certain assumptions on rank-matching and well-boundedness of condition numbers and eigenspace incoherence parameters, a corollary of our main theorem improves the state-of-the-art sampling rate results for nonconvex matrix completion with no spurious local minima in Ge et al. (2016, 2017). In addition, we have investigated when the proposed nonconvex optimization results in accurate low-rank approximations even in presence of large condition numbers, large incoherence parameters, or rank mismatching. We also propose to apply the nonconvex optimization to memory-efficient kernel PCA. Compared to the well-known Nyström methods, numerical experiments indicate that the proposed nonconvex optimization approach yields more stable results in both low-rank approximation and clustering.

Keywords: low-rank approximation, matrix completion, nonconvex optimization, model-free analysis, local minimum analysis, kernel PCA

1. Introduction

Let $\boldsymbol{M}$ be an $n \times n$ positive semidefinite matrix and let $r \ll n$ be a fixed integer. It is well known that a rank-r approximation of $\boldsymbol{M}$ can be obtained by truncating the spectral decomposition of $\boldsymbol{M}$. To be specific, let $\boldsymbol{M} = \sum_{i=1}^n \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$ be the spectral decomposition with $\sigma_1 \geqslant \ldots \geqslant \sigma_n \geqslant 0$. Then, the best rank-$r$ approximation of $\boldsymbol{M}$ is $\boldsymbol{M}_r = \sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$. If we denote $\boldsymbol{U}_r = [\sqrt{\sigma_1}\boldsymbol{u}_1 \ \ldots \ \sqrt{\sigma_r}\boldsymbol{u}_r]$, then the best rank-$r$ approximation of $\boldsymbol{M}$ can be written as $\boldsymbol{M} = \boldsymbol{U}_r \boldsymbol{U}_r^\top$. By the well-known Eckart-Young-Mirsky Theorem (Golub and Van Loan, 2012), $\boldsymbol{U}_r$ is actually the global minimum (up to rotation) to the following nonconvex optimization:

$$\min_{\boldsymbol{X} \in \mathbb{R}^{n \times r}} \|\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{M}\|_F^2.$$

©2019 Ji Chen and Xiaodong Li.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-776.html.

This factorization for low-rank approximation has been well-known in the literature (see, e.g., Burer and Monteiro, 2003).

This paper studies how to find a rank-r approximation of $\boldsymbol{M}$ in the case that only partial entries are observed. Let $\Omega \subset [n] \times [n]$ be a symmetric index set, and we assume that $\boldsymbol{M}$ is only observed on the entries in Ω. For convenience of discussion, this subsampling is represented as $\mathcal{P}_\Omega(\boldsymbol{M})$ in that $\mathcal{P}_\Omega(\boldsymbol{M})_{i,j} = M_{i,j}$ if $(i,j) \in \Omega$ and $\mathcal{P}_\Omega(\boldsymbol{M})_{i,j} = 0$ if $(i,j) \notin \Omega$. We are interested in the following question

How to find a rank-r approximation of $\boldsymbol{M}$ in a scalable manner only through $\mathcal{P}_\Omega(\boldsymbol{M})$?

We propose to find such a low-rank approximation through the following nonconvex optimization, which has been exactly proposed in Ge et al. (2016, 2017) for matrix completion. Denote $\boldsymbol{X} = \left[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right]^\top \in \mathbb{R}^{n \times r}$. A rank-$r$ approximation of $\boldsymbol{M}$ can be found through

$$\begin{aligned} \min_{\boldsymbol{X} \in \mathbb{R}^{n \times r}} f(\boldsymbol{X}) &:= \frac{1}{2} \sum_{(i,j) \in \Omega} \left(\boldsymbol{x}_i^\top \boldsymbol{x}_j - M_{ij}\right)^2 + \lambda \sum_{i=1}^n [(\|\boldsymbol{x}_i\|_2 - \alpha)_+]^4 \\ &:= \frac{1}{2} \|\mathcal{P}_\Omega(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{M})\|_F^2 + \lambda G_\alpha(\boldsymbol{X}) \end{aligned} \tag{1}$$

where $G_\alpha(\boldsymbol{X}) := \sum_{i=1}^n [(\|\boldsymbol{x}_i\|_2 - \alpha)_+]^4$. Following the framework of nonconvex optimization *without initialization* in Ge et al. (2016, 2017), our local-minimum based approximation for $\boldsymbol{M}$ is $\boldsymbol{M} \approx \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$ where $\widehat{\boldsymbol{X}}$ is *any* local minimum of (1).

Let's briefly discuss the memory and computational complexity to solve (1) via gradient descent. If Ω is symmetric and does not contain the diagonal entries as later specified in Definition 1, the updating rule of gradient decent

$$\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} - \eta^{(t)} \nabla f(\boldsymbol{X}^{(t)}) \tag{2}$$

is equivalent to

$$\boldsymbol{x}_i^{(t+1)} := \boldsymbol{x}_i^{(t)} - \eta^{(t)} \left[2 \sum_{j:(i,j) \in \Omega} \left(\langle \boldsymbol{x}_i^{(t)}, \boldsymbol{x}_j^{(t)} \rangle - M_{i,j}\right) \boldsymbol{x}_j^{(t)} + \frac{4\lambda}{\|\boldsymbol{x}_i^{(t)}\|_2} \left(\|\boldsymbol{x}_i^{(t)}\|_2 - \alpha\right)^3 1_{\{\|\boldsymbol{x}_i^{(t)}\|_2 \geqslant \alpha\}} \boldsymbol{x}_i^{(t)} \right],$$

where the memory cost is dominated by storing $\boldsymbol{X}^{(t)}$, $\boldsymbol{X}^{(t+1)}$, and $\boldsymbol{M}$ on Ω, which is generally $O(nr + |\Omega|)$. It is also obvious that the computational cost in each iteration is $O(|\Omega| r)$.

1.1. Applications in memory-efficient kernel PCA

Kernel PCA (Schölkopf et al., 1998) is a widely used nonlinear dimension reduction technique in machine learning for the purpose of redundancy removal and preprocessing before prediction, classification or clustering. The method is implemented by finding a low-rank approximation of the kernel-based Gram matrix determined by the data sample. To be concrete, let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ be a data sample of size n and dimension d, and let $\boldsymbol{M}$ be the $n \times n$ positive semidefinite kernel matrix determined by a predetermined kernel function $K(\boldsymbol{x}, \boldsymbol{y})$ in that $M_{i,j} = K(\boldsymbol{z}_i, \boldsymbol{z}_j)$. Non-centered kernel PCA with r principal components amounts to finding the best rank-r approximation of $\boldsymbol{M}$.

However, when the sample size is large, the storage of the kernel matrix itself becomes challenging. Consider the example when the dimension d is in thousands while the sample size n is in millions. The memory cost for the data matrix is $d \times n$ and thus in billions, while the memory cost for the kernel matrix $\boldsymbol{M}$ is in trillions! On the other hand, if not storing $\boldsymbol{M}$, the implementation of standard iterative algorithms of SVD will involve one pass of computing all entries of $\boldsymbol{M}$ in each iteration, usually with formidable computational cost $O(n^2 d)$. A natural question arises: *How to find low-rank approximations of $\boldsymbol{M}$ memory-efficiently?*

The following two are among the most well-known memory-efficient kernel PCA methods in the literature. One is Nyström method (Williams and Seeger, 2001), which amounts to generating *random partial columns* of the kernel matrix, then finding a low-rank approximation based on these columns. In order to generate random partial columns, uniform sampling without replacement is employed in Williams and Seeger (2001), and different sampling strategies are proposed later (e.g., Drineas and Mahoney, 2005). The method is convenient in implementation and efficient in both memory and computation, but relatively unstable in terms of approximation errors as will be shown in Section 3.

Another popular approach is stochastic approximation, e.g., Kernel Hebbian Algorithm (KHA) (Kim et al., 2005), which is memory-efficient and approaches the exact principal component solution as the number of iterations goes to infinity with appropriately chosen learning rate (Kim et al., 2005). However, based on our experience, the method usually requires careful tuning of learning rates even for very slow convergence.

It is also worth mentioning that the randomized one-pass algorithm discussed in, e.g., Halko et al. (2011), where the theoretical properties of a random-projection based low-rank approximation method were fully analyzed. However, although the one-pass algorithm does not require the storage of the whole matrix $\boldsymbol{M}$, in kernel PCA one still needs to compute every entry of $\boldsymbol{M}$, which typically requires $O(n^2 d)$ computational complexity for kernel matrix.

As a result, we aim at finding a memory-efficient method as an alternative to the aforementioned approaches. In particular, we are interested in a method with desirable empirical properties: memory-efficient, no requirement on one or multiple passes to compute the complete kernel matrix, no requirement to tune the parameters carefully, and yielding stable results. To this end, we propose the following method based on entries sampling and nonconvex optimization: In the first step, Ω is generated to follow an Erdős-Rényi random graph with parameter p later specified in Definition 1, and then a partial kernel matrix $\mathcal{P}_\Omega(\boldsymbol{M})$ is generated in that $M_{i,j} = K(\boldsymbol{z}_i, \boldsymbol{z}_j)$ for $(i,j) \in \Omega$. In the second step, the nonconvex optimization (1) is implemented through gradient descent (2). Any local minimum of (1), $\widehat{X}$, is a solution of approximate kernel PCA in that $\boldsymbol{M} \approx \widehat{X}\widehat{X}^\top$.

To store the index set Ω and the sampled entries of $\boldsymbol{M}$ on Ω, the memory cost in the first step is $O(|\Omega|)$, which is comparable to the memory cost $O(nr + |\Omega|)$ in the second step. As to the computational complexity, besides the generation of Ω, the computational cost in the first step is typically $O(|\Omega| d)$, e.g., when the radial kernels or polynomial kernels are employed. This could be dominating the per-iteration computational complexity $O(|\Omega| r)$ in the second step when the target rank r is much smaller than the original dimension d.

Partial entries sampling plus nonconvex optimization has been proposed in the literature for scalable robust PCA and matrix completion (Yi et al., 2016). However, to the best of our knowledge, our work is the first to apply such an idea to memory-efficient kernel PCA. Moreover, the underlying signal matrix is assumed to be exactly low-rank in Yi et al. (2016) while we make no assumptions on the positive semidefinite kernel matrix $\boldsymbol{M}$. Entry-sampling has been proposed in Achlioptas et al. (2002); Achlioptas and McSherry (2007) for scalable low-rank approximation. In particular, it is used to speed up kernel PCA in Achlioptas et al. (2002), but spectral methods are subsequently employed after entries sampling as opposed to nonconvex optimization. Empirical comparisons between spectral methods and nonconvex optimization will be demonstrated in Section 3. It is also noteworthy that matrix completion techniques have been applied to certain kernel matrices when it is costly to generate each single entry (Graepel, 2002; Paisley and Carin, 2010), wherein the proposed methods are not memory-efficient. In contrast, our method is memory-efficient in order to serve a different purpose.

1.2. Related work and our contributions

In recent years, a series of papers have been proposed to study nonconvex matrix completion (see, e.g., Rennie and Srebro, 2005; Keshavan et al., 2010b,a; Jain et al., 2013; Zhao et al., 2015; Sun and Luo, 2016; Chen and Wainwright, 2015; Yi et al., 2016; Zheng and Lafferty, 2016; Ge et al., 2016, 2017). Interested readers are referred to Balcan et al. (2017), where required sampling rates in these papers are summarized in Table 1 therein. Compared to convex approaches for matrix completion (e.g., Candès and Recht, 2009), these nonconvex approaches are not only more computationally efficient, but also more convenient in storing. For the same reason, nonconvex optimization approaches have also been investigated for other low-rank recovery problems including phase retirval (e.g., Candes et al., 2015; Sun et al., 2018; Cai et al., 2016), matrix sensing (e.g., Zheng and Lafferty, 2015; Tu et al., 2015), blind deconvolution (e.g., Li et al., 2018), etc.

Our present work follows the framework of local minimum analysis for nonconvex optimization in the literature. For example, Baldi and Hornik (1989) has described the nonconvex landscape of the quadratic loss for PCA. Loh and Wainwright (2015) studies the local minima of regularized M-estimators. Sun et al. (2018) studies the global geometry of the phase retrieval problem. The conditions for no spurious local minima have been investigated in Bhojanapalli et al. (2016) and Ge et al. (2016) for nonconvex matrix sensing and completion, respectively. The global geometry of nonconvex objective functions with underlying symmetric structures, including low-rank symmetric matrix factorization and sensing, has been studied in Li et al. (2016a). Global geometry of rectangular matrix factorization and sensing is studied Zhu et al. (2017), where the issues of under-parameterization and over-parameterization have been investigated. Similar analysis has been extended to general low-rank optimization problems in Li et al. (2017). Matrix factorization is further studied in Jin et al. (2017) with a novel geometric characterization of saddle points, and this idea is later extended in Ge et al. (2017), where a unified geometric analysis framework is proposed to study the landscapes of nonconvex matrix sensing, matrix completion and robust PCA.

Among these results, Ge et al. (2016) and Ge et al. (2017) are highly relevant to our work in both methodological and technical terms. In fact, exactly the same nonconvex optimization problem (1) has been studied in Ge et al. (2016, 2017) for matrix completion from missing data. To be specific, these papers show that any local minimum $\widehat{\boldsymbol{X}}$ yields $\boldsymbol{M} =$

$\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$, as long as $\boldsymbol{M}$ is exactly rank-r, the condition number $\kappa_r := \sigma_1/\sigma_r$ is well-bounded, the incoherence parameter of the eigenspace of $\boldsymbol{M}$ is well-bounded, and the sampling rate is greater than a function of these quantities. The case with additive stochastic noise has also been discussed in Ge et al. (2016).

In contrast, our paper studies the theoretical properties of $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$ with *no assumptions* on $\boldsymbol{M}$. There are actually two questions of interest: how close $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$ is from $\boldsymbol{M}$, and how close $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$ is from $\boldsymbol{M}_r$ (recall that $\boldsymbol{M}_r$ is the best rank-r approximation of $\boldsymbol{M}$ by spectral truncation). In comparison to Ge et al. (2016, 2017), our main contributions to be introduced in the next section include the following:

- Our main result Theorem 2 that characterizes how well any local-minimum based rank-r factorization $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$ approximates $\boldsymbol{M}$ or $\boldsymbol{M}_r$ requires no assumptions imposed on $\boldsymbol{M}$ regarding its rank, eigenvalues and eigenvectors. The sampling rate is only required to satisfy $p \geqslant C(\log n/n)$ for some absolute constant C. Therefore, for applications such as memory-efficient kernel PCA, our framework provides more suitable guidelines than Ge et al. (2016, 2017). In fact, kernel matrices are in general of full rank and their condition numbers and incoherence parameters may not satisfy the strong assumptions in Ge et al. (2016, 2017).

- When $\boldsymbol{M}$ is assumed to be exactly low-rank as in Ge et al. (2016, 2017), Corollary 3 improves the state-of-the-art no-spurious-local-minima results in Ge et al. (2016, 2017) for exact nonconvex matrix completion in terms of sampling rates. To be specific, assuming both condition numbers and incoherence parameters are on the order of $O(1)$, our result improves the result in Ge et al. (2017) from $\widetilde{O}(r^4/n)$ to $\widetilde{O}(r^2/n)$.

- Theorem 2 also implies the conditions under which the nonconvex optimization (1) yields good low-rank approximation of $\boldsymbol{M}$ in the cases of large condition numbers, high incoherence parameters, or rank-mismatching.

On the other hand, our paper benefits from Ge et al. (2016, 2017) in various aspects. In order to characterize the properties of any local minimum $\widehat{\boldsymbol{X}}$, we follow the idea in Ge et al. (2017) to combine the first and second order conditions of local minima linearly to construct an auxiliary function, denoted as $K(\boldsymbol{X})$ in our paper, and consequently all local minima satisfy the inequality $K(\widehat{\boldsymbol{X}}) \geqslant 0$ as illustrated in Figure 1. If $\boldsymbol{M}$ is exactly rank-r and its eigenvalues and eigenvectors satisfy particular properties, Ge et al. (2017) shows that $K(\boldsymbol{X}) \leqslant 0$ for all $\boldsymbol{X}$ as long as the sampling rate is large enough. This argument can be employed to prove that there is no spurious local minima.

However, $K(\boldsymbol{X}) \leqslant 0$ does not hold for all $\boldsymbol{X}$ if no assumptions are imposed on $\boldsymbol{M}$, so we instead focus on analyzing the inequality $K(\widehat{\boldsymbol{X}}) \geqslant 0$ directly in the model-free manner. Among a few novel technical ideas, the success of such model-free analysis relies crucially on the deterministic inequality (Lemma 8) that controls the difference between the function $K(\boldsymbol{X})$ and its population version $\mathbb{E}[K(\boldsymbol{X})]$.

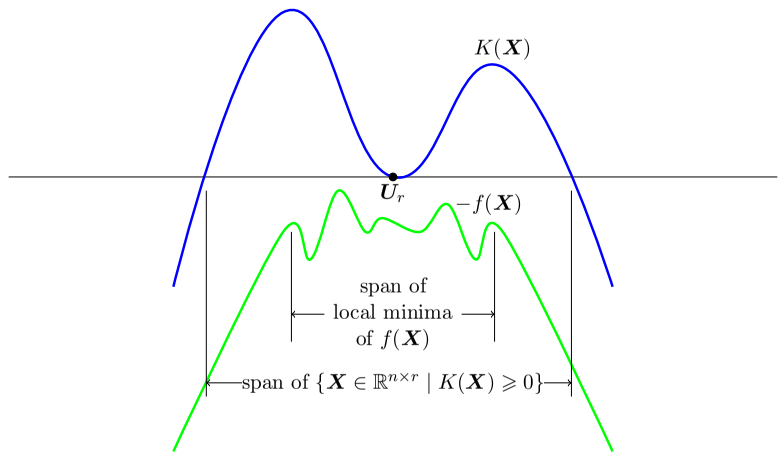


Figure 1: Landscape of $-f(\boldsymbol{X}), K(\boldsymbol{X})$ and $\boldsymbol{U}_r$.

1.3. Organization and notations

The remainder of the paper is organized as follows: Our main theoretical results are stated in Section 2; Numerical simulations and applications in memory-efficient KPCA are given in Section 3. Proofs are deferred to Section 4.

We use bold letters to denote matrices and vectors. For any vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, $\|\boldsymbol{u}\|_2$ denotes its ℓ_2 norm, and $\langle \boldsymbol{u}, \boldsymbol{v}\rangle$ their inner product. For any matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, $M_{i,j}$ denotes its (i,j)-th entry, $\boldsymbol{M}_{i,\cdot} = (M_{i,1}, M_{i,2}, \ldots, M_{i,n})^\top$ its i-th row of $\boldsymbol{M}$, and $\boldsymbol{M}_{\cdot,j} = (M_{1,j}, M_{2,j}, \ldots, M_{n,j})^\top$ its j-th column. Moreover, we use $\|\boldsymbol{M}\|$, $\|\boldsymbol{M}\|_*$, $\|\boldsymbol{M}\|_F$, $\|\boldsymbol{M}\|_{\ell_\infty} := \max_{i,j} |M_{i,j}|$, $\|\boldsymbol{M}\|_{2,\infty} := \max_i \|\boldsymbol{M}_{i,\cdot}\|_2$ to denote its spectral norm, nuclear norm, Frobenius norm, elementwise max norm and $\ell_{2,\infty}$ norm, respectively. The vectorization of $\boldsymbol{M}$ is represented by $\mathrm{vec}(\boldsymbol{M}) = (M_{1,1}, M_{2,1}, \ldots, M_{1,2}, \ldots, M_{n,n})^\top$. For matrices $\boldsymbol{M}, \boldsymbol{N}$ of the same size, denote $\langle \boldsymbol{M}, \boldsymbol{N}\rangle = \sum_{i,j} M_{i,j}N_{i,j} = \mathrm{trace}\left(\boldsymbol{M}^\top \boldsymbol{N}\right)$. Denote by $\nabla f(\boldsymbol{M}) \in \mathbb{R}^{n\times n}$ and $\nabla^2 f(\boldsymbol{M}) \in \mathbb{R}^{n^2\times n^2}$ the gradient and Hessian of $f(\boldsymbol{M})$.

Denote $[x]_+ = \max\{x, 0\}$. We use $\boldsymbol{J}$ to denote a matrix whose all entries equal to one. We use $C, C_1, C_2, \ldots$ to denote absolute constants, whose values may change from line to line.

2. Model-free approximation theory

2.1. Main results

The following sampling scheme is employed throughout the paper:

Definition 1 (Off-diagonal symmetric independent Ber(p) model) *Assume the index set Ω consists only of off-diagonal entries that are sampled symmetrically and independently with probability p, i.e.,*

1. *$(i,i) \notin \Omega$ for all $i = 1, \dots, n$;*
2. *For all $i < j$, sample $(i,j) \in \Omega$ independently with probability p;*
3. *For all $i > j$, $(i,j) \in \Omega$ if and only if $(j,i) \in \Omega$.*

Here we assume all diagonal entries are not in Ω for the generality of the formulation, although they are likely to be obtained in practice. For instance, all diagonal entries of the radial kernel matrix are ones. For any index set $\Omega \subset [n] \times [n]$, define the associated 0-1 matrix $\boldsymbol{\Omega} \in \{0,1\}^{n\times n}$ such that $\Omega_{i,j} = 1$ if and only if $(i,j) \in \Omega$. Then we can write $\mathcal{P}_\Omega(\boldsymbol{X}) = \boldsymbol{X} \circ \boldsymbol{\Omega}$ where $\circ$ is the Hadamard product.

Assume that the positive semidefinite matrix $\boldsymbol{M}$ has the spectral decomposition

$$\boldsymbol{M} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top + \sum_{i=r+1}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top := \boldsymbol{M}_r + \boldsymbol{N}, \tag{3}$$

where $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$ are the spectrum, $\boldsymbol{u}_i \in \mathbb{R}^n$ are unit and mutually perpendicular eigenvectors. The matrix $\boldsymbol{M}_r := \sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$ is the best rank-r approximation of $\boldsymbol{M}$ and $\boldsymbol{N} := \sum_{i=r+1}^n \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$ denotes the residual part. In the case of multiple eigenvalues, the order in the eigenvalue decomposition (3) may not be unique. In this case, we consider the problem for any fixed order in (3) with the fixed $\boldsymbol{M}_r$.

Theorem 2 *Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix with the spectral decomposition (3). Let Ω be sampled according to the off-diagonal symmetric Ber(p) model with $p \geqslant C_S \frac{\log n}{n}$ for some absolute constant C_S. Then in an event E with probability $\mathbb{P}[E] \geqslant 1 - 2n^{-3}$, as long as the tuning parameters α and λ satisfy $100\sqrt{\|\boldsymbol{M}_r\|_{\ell_\infty}} \leqslant \alpha \leqslant 200\sqrt{\|\boldsymbol{M}_r\|_{\ell_\infty}}$ and $100\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \leqslant \lambda \leqslant 200\|\boldsymbol{\Omega} - p\boldsymbol{J}\|$, any local minimum $\widehat{\boldsymbol{X}} \in \mathbb{R}^{n\times r}$ of (1) satisfies*

$$\begin{aligned}\left\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}_r\right\|_F^2 \leqslant & C_1 \sum_{i=1}^{r} \left\{ \left[C_2 \left(\sqrt{\frac{n}{p}} + \frac{\log n}{p} \right) \|\boldsymbol{M}_r\|_{\ell_\infty} + C_2 \sigma_{2r+1-i} - \sigma_i \right]_+ \right\}^2 \\ & + C_1 \frac{[p(1-p)n + \log n] r \|\boldsymbol{N}\|_{\ell_\infty}^2}{p^2}\end{aligned} \tag{4}$$

and

$$\begin{aligned}\left\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\right\|_F^2 \leqslant & C_1 \sum_{i=1}^{r} \left\{ \left[C_2 \left(\sqrt{\frac{n}{p}} + \frac{\log n}{p} \right) \|\boldsymbol{M}_r\|_{\ell_\infty} + C_2 \sigma_{2r+1-i} - \sigma_i \right]_+ \right\}^2 \\ & + C_1 \frac{[p(1-p)n + \log n] r \|\boldsymbol{N}\|_{\ell_\infty}^2}{p^2} + \|\boldsymbol{N}\|_F^2\end{aligned} \tag{5}$$

with C_1, C_2 absolute constants.

Model-free low-rank approximation from partial entries has been studied for for spectral estimators in the literature. For example, under the settings of Theorem 2, the spectral low-rank approximation (denoted as $\boldsymbol{M}_{\text{approx}}$) discussed in Keshavan et al. (2010a, Theorem 1.1) is guaranteed to satisfy

$$\|\boldsymbol{M}_{\text{approx}} - \boldsymbol{M}_r\|_F^2 \leqslant C \left\{ \frac{nr\|\boldsymbol{M}_r\|_{\ell_\infty}^2}{p} + \frac{r\|\mathcal{P}_\Omega(\boldsymbol{N})\|^2}{p^2} \right\},$$

with high probability. However, this cannot imply exact recovery even when $\boldsymbol{M}$ is of low rank and the sampling rate p satisfies the conditions specified in Ge et al. (2017). Similarly, the SVD-based USVT estimator introduced in Chatterjee (2015) does not imply exact recovery. In contrast, as will be discussed in the next subsection, Theorem 2 implies that any local minimum of (1) yields exact recovery of $\boldsymbol{M}$ with high probability under milder conditions than those in Ge et al. (2017).

2.2. Implications in exact matrix completion

Assume in this subsection that the positive semidefinite matrix $\boldsymbol{M}$ is exactly rank-r, i.e.,

$$\boldsymbol{M} = \boldsymbol{M}_r = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top = \boldsymbol{U}_r \boldsymbol{U}_r^\top \tag{6}$$

where $\boldsymbol{U}_r = [\sqrt{\sigma_1}\boldsymbol{u}_1 \ \dots \ \sqrt{\sigma_r}\boldsymbol{u}_r]$. Furthermore, we assume its condition number $\kappa_r = \frac{\sigma_1}{\sigma_r}$ and eigen-space incoherence parameter $\mu_r = \frac{n}{r}\max_i \sum_{j=1}^r u_{i,j}^2$ (Candès and Recht, 2009) are well-bounded. This is a standard setup in the literature of nonconvex matrix completion (e.g., Keshavan et al., 2010b; Sun and Luo, 2016; Chen and Wainwright, 2015; Zheng and Lafferty, 2016; Ge et al., 2016; Yi et al., 2016; Ge et al., 2017).

Notice that Ge et al. (2016) introduces a slightly different version of incoherence

$$\widetilde{\mu}_r := \frac{\sqrt{n}\|\boldsymbol{U}_r\|_{2,\infty}}{\|\boldsymbol{U}_r\|_F} = \sqrt{\frac{n\|\boldsymbol{M}_r\|_{\ell_\infty}}{\operatorname{trace}(\boldsymbol{M}_r)}} \tag{7}$$

as a measure of spikiness. (Note that this is different from the spikiness defined in Negahban and Wainwright (2012).) By $\|\boldsymbol{M}_r\|_{\ell_\infty} = \|\boldsymbol{U}_r\|_{2,\infty}^2 = \max_i \sum_{j=1}^r \sigma_j u_{i,j}^2$, the following relationship between μ and $\widetilde{\mu}$ is straightforward

$$\frac{\widetilde{\mu}_r^2}{\kappa_r} \leqslant \frac{\widetilde{\mu}_r^2 \operatorname{trace}(\boldsymbol{M}_r)}{r\sigma_1} = \frac{n\|\boldsymbol{M}_r\|_{\ell_\infty}}{r\sigma_1} \leqslant \mu_r \leqslant \frac{n\|\boldsymbol{M}_r\|_{\ell_\infty}}{r\sigma_r} = \frac{\widetilde{\mu}_r^2 \operatorname{trace}(\boldsymbol{M}_r)}{r\sigma_r} \leqslant \kappa_r \widetilde{\mu}_r^2. \tag{8}$$

By the fact $\|\boldsymbol{M}\|_{\ell_\infty} \leqslant \frac{r}{n}\sigma_1\mu_r$, Theorem 2 implies the following exact low-rank recovery results:

Corollary 3 *Under the assumptions of Theorem 2, if we further assume $\operatorname{rank}(\boldsymbol{M}) = r$ (i.e., $\boldsymbol{M} = \boldsymbol{M}_r$) and*

$$p \geqslant C \max\left\{ \frac{\mu_r r \kappa_r \log n}{n}, \frac{\mu_r^2 r^2 \kappa_r^2}{n} \right\}$$

or

$$p \geqslant C \max\left\{\frac{\widetilde{\mu}_r^2 r\kappa_r \log n}{n}, \frac{\widetilde{\mu}_r^4 r^2 \kappa_r^2}{n}\right\}$$

for some absolute constant C, then in an event E with probability $\mathbb{P}[E] \geqslant 1-2n^{-3}$, any local minimum $\widehat{\boldsymbol{X}} \in \mathbb{R}^{n\times r}$ of objective function $f(\boldsymbol{X})$ defined in (1) satisfies $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top = \boldsymbol{M}$.

The proof is straightforward and deferred to the appendix. Notice that our results are better than the state-of-the-art results for no spurious local minimum in Ge et al. (2017), where the required sampling rate is $p \geqslant \frac{C}{n}\mu_r^3 r^4 \kappa_r^4 \log n$ (which also implies $p \geqslant \frac{C}{n}\widetilde{\mu}_r^6 r^4 \kappa_r^7 \log n$ by (8)).

2.3. Examples

Besides improving the state-of-the-art no-spurious-local-minima results in nonconvex matrix completion, Theorem 2 is also capable of explaining some nontrivial phenomena in low-rank matrix completion in the presence of large condition numbers, high incoherence parameter, or mismatching between the selected and true ranks.

2.3.1. Nonconvex matrix completion with large condition numbers and high eigen-space incoherence parameters

Assume here $\boldsymbol{M}$ is exactly rank-r and its spectral decomposition is denoted as in (6). However, we assume that μ_r and κ_r can be extremely large, while the condition number and incoherence parameter for $\boldsymbol{M}_{r-1} = \sum_{i=1}^{r-1}\sigma_i \boldsymbol{u}_i\boldsymbol{u}_i^\top$, i.e., $\kappa_{r-1} = \frac{\sigma_1}{\sigma_{r-1}}$ and $\mu_{r-1} = \frac{n}{r-1}\max_i \sum_{j=1}^{r-1} u_{i,j}^2$, are well-bounded. We are interested in figuring out when the local minimum based rank-r factorization $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$ approximates the original $\boldsymbol{M}$ well.

By $\|\boldsymbol{M}_r\|_{\ell_\infty} = \max_i \sum_{j=1}^r \sigma_j u_{i,j}^2$, we have

$$\|\boldsymbol{M}_r\|_{\ell_\infty} \leqslant \frac{r-1}{n}\sigma_1\mu_{r-1} + \sigma_r\|\boldsymbol{u}_r\|_\infty^2.$$

Then by Theorem 2, if

$$p \geqslant C\max\left\{\frac{\left[\mu_{r-1}\kappa_{r-1}(r-1) + n\frac{\sigma_r}{\sigma_{r-1}}\|\boldsymbol{u}_r\|_\infty^2\right]\log n}{n}, \frac{\left[\mu_{r-1}\kappa_{r-1}(r-1) + n\frac{\sigma_r}{\sigma_{r-1}}\|\boldsymbol{u}_r\|_\infty^2\right]^2}{n}\right\}$$

with some absolute constant C, in an event E with probability $\mathbb{P}[E] \geqslant 1-2n^{-3}$, for any local minimum $\widehat{\boldsymbol{X}} \in \mathbb{R}^{n\times r}$ of (1), $\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\|_F^2 \leqslant \frac{1}{100}\sigma_{r-1}^2$ holds. In other words, the relative approximation error satisfies $RE := \frac{\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F} \leqslant \frac{1}{10\sqrt{r-1}}$.

Notice that $\|\boldsymbol{u}_r\|_\infty^2 \leqslant \frac{r}{n}\mu_r$ and $\frac{\sigma_r}{\sigma_{r-1}} = \frac{\kappa_{r-1}}{\kappa_r}$, so the above sampling rate requirement is satisfied as long as $\frac{\mu_r}{\kappa_r} \leqslant C\mu_{r-1}$ and

$$p \geqslant C\max\left\{\frac{\mu_{r-1}\kappa_{r-1} r\log n}{n}, \frac{\mu_{r-1}^2\kappa_{r-1}^2 r^2}{n}\right\}.$$

2.3.2. Rank mismatching

In this subsection, $\boldsymbol{M}$ is assumed to be exactly rank-R, i.e.,

$$\boldsymbol{M} = \boldsymbol{M}_R = \sum_{i=1}^R \sigma_i \boldsymbol{u}_i\boldsymbol{u}_i^\top = \boldsymbol{U}_R\boldsymbol{U}_R^\top$$

where $\boldsymbol{U}_R = [\sqrt{\sigma_1}\boldsymbol{u}_1 \ \ldots \ \sqrt{\sigma_R}\boldsymbol{u}_R]$. However, we consider the case that the selected rank r is not the same as the true rank R, i.e., rank mismatching. As with Section 2.2, we assume the condition number $\kappa_R = \frac{\sigma_1}{\sigma_R}$ and eigen-space incoherence parameter $\mu_R = \frac{n}{R}\max_i \sum_{j=1}^R \sigma_j u_{i,j}^2$ are well-bounded. As with (8), there holds $\|\boldsymbol{M}\|_{\ell_\infty} \leqslant \frac{R}{n}\sigma_1\mu_R$.

Case 1: $R < r$. Theorem 2 implies that if

$$p \geqslant C\max\left\{\frac{\mu_R\kappa_R R\log n}{n}, \frac{\mu_R^2\kappa_R^2 R^2}{n}\right\}$$

for some absolute constant C, then in an event E with probability $\mathbb{P}[E] \geqslant 1-2n^{-3}$, any local minimum $\widehat{\boldsymbol{X}} \in \mathbb{R}^{n\times r}$ of (1) yields $\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\|_F^2 \leqslant \frac{1}{100}(r-R)\sigma_R^2$. This further yields the relative approximation error bound $RE := \frac{\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F} \leqslant \frac{1}{10}\sqrt{\frac{r-R}{R}}$.

Case 2: $R > r$. Recall that $\|\boldsymbol{M}_r\|_{\ell_\infty} \leqslant \frac{r}{n}\sigma_1\mu_r$. Moreover,

$$\|\boldsymbol{N}\|_{\ell_\infty} = \max_i \sum_{j=r+1}^R \sigma_j u_{i,j}^2 \leqslant \sigma_{r+1}\left(\max_i \sum_{j=1}^R u_{i,j}^2\right) = \frac{\mu_R R}{n}\sigma_{r+1}.$$

Theorem 2 implies that if

$$p \geqslant C\max\left\{\frac{\mu_r r\kappa_r \log n}{n}, \frac{\mu_r^2 r^2\kappa_r^2}{n}, \frac{\mu_R^2 R^3}{n}\right\}$$

for some absolute constant C, then with high probability, any local minimum $\widehat{\boldsymbol{X}} \in \mathbb{R}^{n\times r}$ of (1) yields

$$\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}_r\|_F^2 \leqslant C(\sigma_{r+1}^2 + \ldots + \sigma_{2r}^2),$$

which implies that the relative error is well-controlled as long as $\sigma_{r+1}^2 + \ldots + \sigma_R^2$ accounts for a small proportion in $\sigma_1^2 + \ldots + \sigma_R^2$.

If we assume that $2C_2\sigma_{r+1} < \sigma_r$ where C_2 is specified in Theorem 2, under the same sampling rate requirement as above, Theorem 2 implies a much sharper result:

$$\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}_r\|_F^2 \leqslant \frac{1}{100}\sigma_{r+1}^2,$$

which yields the following (perhaps surprising) relative approximation error bound

$$RE := \frac{\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}_r\|_F}{\|\boldsymbol{M}_r\|_F} \leqslant \frac{1}{10}\sqrt{\frac{\sigma_{r+1}^2}{\sigma_1^2 + \ldots + \sigma_r^2}} \leqslant \frac{1}{10\sqrt{r}}.$$

3. Experiments

In the following simulations where the nonconvex optimization (1) is solved, the initialization $\boldsymbol{X}^{(0)}$ is constructed randomly with i.i.d. normal entries with mean 0 and variance 1. The step size $\eta^{(t)}$ for the gradient descent (2) is determined by Armijo's rule (Armijo, 1966). The gradient descent algorithm is implemented with sparse matrix storage in Section 3.2 for the purpose of memory-efficient KPCA, while with full matrix storage in Section 3.1 to test the performance of general low-rank approximations from missing data. In each experiment, the iterations will be terminated when $\|\nabla f(\boldsymbol{X}^{(t)})\|_F \leqslant 10^{-3}$ or $\|\eta^{(t)}\nabla f(\boldsymbol{X}^{(t)})\|_F \leqslant 10^{-10}$ or the number of iterations surpasses 10^3. All methods are implemented in MATLAB. The experiments are running on a virtual computer with Linux KVM, with 12 cores of 2.00GHz Intel Xeon E5 processor and 16 GB memory.

3.1. Numerical simulations

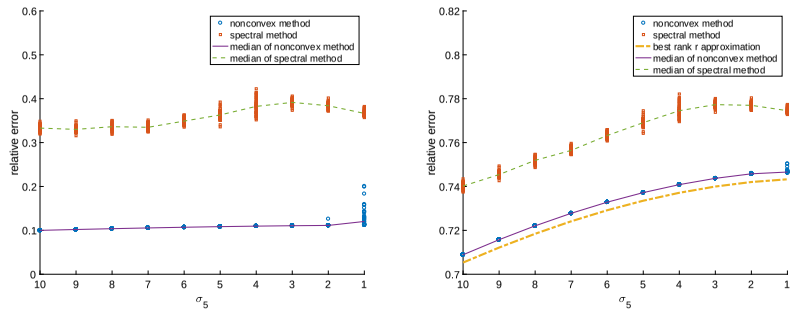
In this section, we conduct numerical tests on the nonconvex optimization (1) under different settings of spectrum for the 500×500 positive semidefinite matrix $\boldsymbol{M}$, whose eigenvectors are the same as the left singular vectors of a random 500×500 matrix with i.i.d. standard normal entries. The generation of eigenvalues for $\boldsymbol{M}$ will be further specified in each test. For each generated $\boldsymbol{M}$, the nonconvex optimization (1) is implemented for 50 times with independent Ω's generated under the off-diagonal symmetric independent $\mathrm{Ber}(p)$ model. To implement the gradient descent algorithm (2), set $\alpha = 100\|\boldsymbol{M}\|_{\ell_\infty}$ and $\lambda = 100\|\boldsymbol{\Omega} - p\boldsymbol{J}\|$ (the performances of our method are empirically not sensitive to the choices of the tuning parameters). In each single numerical experiment, we also conduct spectral method proposed in Achlioptas et al. (2002) to obtain an approximate low-rank approximation of $\boldsymbol{M}$ for the purpose of comparison.

3.1.1. Full rank case

Here $\boldsymbol{M}$ is assumed to have full rank, i.e., $\mathrm{rank}(\boldsymbol{M}) = 500$. To be specific, let $\sigma_1 = \cdots = \sigma_4 = 10$, $\sigma_6 = \cdots = \sigma_{500} = 1$, and $\sigma_5 = 10, 9, 8, \ldots, 2, 1$. The selected rank used in the nonconvex optimization (1) is set as $r = 5$, and the sampling rate is set as $p = 0.2$. With different values of σ_5, the results of our implementations of the gradient descent are plotted in Figure 2. One can observe that the relative errors for our nonconvex method (1) are well-bounded for different σ_5's, and much smaller than those for spectral low-rank approximation. The results indicate that our approach is able to approximate the "true" best rank-r approximation $\boldsymbol{M}_r$ accurately in the presence of heavy spectral tail and possibly large condition number σ_1/σ_5, even with only 20% observed entries.

3.1.2. Low-rank matrix with large condition numbers

Here $\boldsymbol{M}$ is assumed to be of exactly low rank with different condition numbers. Let $\sigma_1 = \cdots = \sigma_4 = 10$, $\sigma_5 = \frac{10}{\kappa}$, and $\sigma_6 = \cdots = \sigma_{500} = 0$. Here the condition number takes on values $\kappa = 10, 20, 30, 40, 50, 100, 200, \infty$, which implies $\mathrm{rank}(\boldsymbol{M}) = 5$ if $\kappa < \infty$ while $\mathrm{rank}(\boldsymbol{M}) = 4$ if $\kappa = \infty$. The selected rank is always assumed to be $r = 5$, while the sampling rate is always $p = 0.2$.



(a) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}_r\|_F}{\|\boldsymbol{M}_r\|_F}$. (b) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}$.

Figure 2: Relative errors for full rank case.

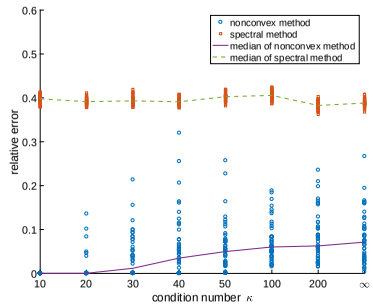


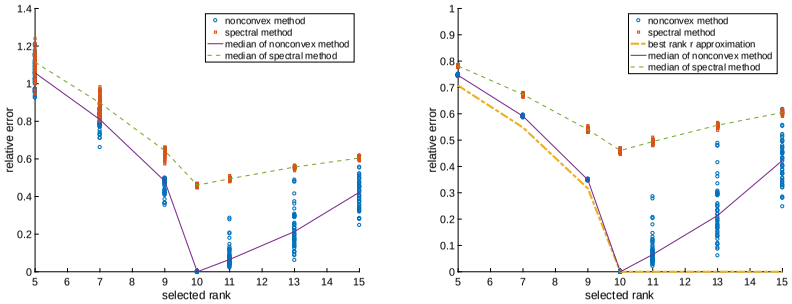
Figure 3: Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}$ for low-rank matrix with extreme condition numbers.

The performance of our nonconvex approach with various choices of κ is demonstrated in Figure 3. One can observe that our nononvex optimization approach yields exact recovery of $\boldsymbol{M}$ when $\kappa = 10$, while yields accurate low-rank approximation for $\boldsymbol{M}$ with relative errors almost always smaller than 0.3 when $\kappa \geqslant 20$. This fact is consistent with the example we discussed in Section 2.3.1, where we have shown that under certain incoherence conditions, the relative approximation error can be well-bounded even when $\kappa_r = \infty$.

3.1.3. Rank mismatching

In this section, we consider rank mismatching, i.e., the rank of $\boldsymbol{M}$ is low but different from the selected rank r. In particular, we consider two settings for simulation: First, we fix $\boldsymbol{M}$ with $\mathrm{rank}(\boldsymbol{M}) = 10$, while the nonconvex optimization is implemented with selected rank $r = 5, 7, 9, 10, 11, 13, 15$; Second, the matrix $\boldsymbol{M}$ is randomly generated with

rank from 1 to 15, while the selected rank is always $r = 5$. The sampling rate is fixed as $p = 0.2$. We perform the simulation on two sets of spectrums: For the first one, all the nonzero eigenvalues are 10; And the second one has decreasing eigenvalues: $\sigma_1 = 20, \sigma_2 = 18, \cdots, \sigma_{10} = 2$ for the case of fixed $\mathrm{rank}(\boldsymbol{M})$, $\sigma_1 = 30, \cdots, \sigma_{\mathrm{rank}(\boldsymbol{M})} = 32 - 2 \times \mathrm{rank}(\boldsymbol{M})$ for the case of fixed selected rank r. Numerical results for the case of fixed $\mathrm{rank}(\boldsymbol{M})$ are demonstrated in Figure 4 (constant nonzero eigenvalues) and Figure 6 (decreasing nonzero eigenvalues), while the case of fixed selected rank in Figure 5 (constant nonzero eigenvalues) and Figure 7 (decreasing nonzero eigenvalues). One can observe from these figures that if the selected rank r is less than the actual rank $\mathrm{rank}(\boldsymbol{M})$, for the approximation of $\boldsymbol{M}$, our nonconvex approach performs almost as well as the complete-data based best low-rank approximation $\boldsymbol{M}_r$. Another interesting phenomenon is that our nonconvex method outperforms simple spectral methods in the approximation of either $\boldsymbol{M}$ or $\boldsymbol{M}_r$ significantly if the selected rank is greater than or equal to the true rank.



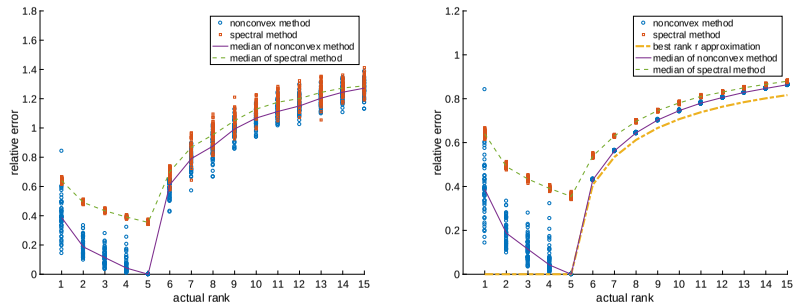
(a) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}_r\|_F}{\|\boldsymbol{M}_r\|_F}$. (b) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}$.

Figure 4: Relative errors for rank mismatching for a fixed $\boldsymbol{M}$ with $\mathrm{rank}(\boldsymbol{M}) = 10$.

3.2. Memory-efficient kernel PCA

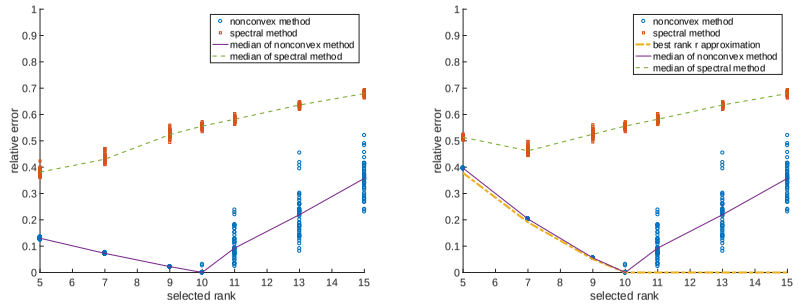
In this section we study the empirical performance of our memory-efficient kernel PCA approach by applying it to the synthetic data set in Wang (2012). The data set is an i.i.d. sample with sample size $n = 10,000$ and dimension $d = 3$, and the data points are partitioned into two classes independently with equal probabilities. Points in the first class are first generated uniformly at random on the three-dimensional sphere $\{\boldsymbol{x} : \|\boldsymbol{x}\|_2 = 0.3\}$, while points in the second class are first generated uniformly at random on the three-dimensional sphere $\{\boldsymbol{x} : \|\boldsymbol{x}\|_2 = 1\}$. Every point is then perturbed independently by $\mathcal{N}(\boldsymbol{0}, \frac{1}{100}\boldsymbol{I}_3)$ noise. We aim to implement memory-efficient uncentered kernel PCA with $r = 2$ on this dataset with the radial kernel $\exp(-\|\boldsymbol{x} - \boldsymbol{y}\|_2^2)$ in order to cluster the data points.

To implement the Nyström method (Williams and Seeger, 2001), 50 columns (and corresponding rows) are selected uniformly at random without replacement, then a rank-2



(a) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}_r\|_F}{\|\boldsymbol{M}_r\|_F}$. (b) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}$.

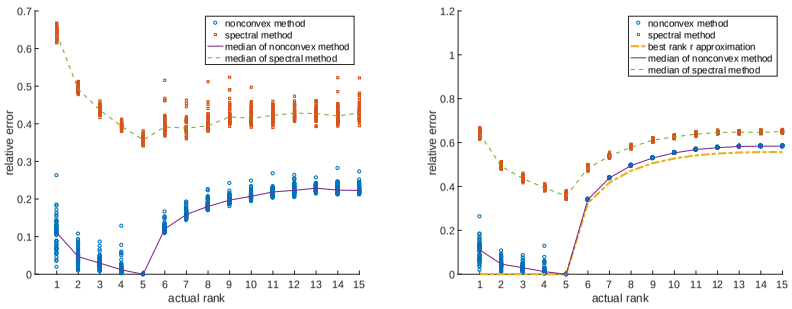
Figure 5: Relative errors for rank mismatching, fixed selected rank.



(a) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}_r\|_F}{\|\boldsymbol{M}_r\|_F}$. (b) Relative error $\frac{\|\boldsymbol{M}_{\mathrm{approx}} - \boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}$.

Figure 6: Relative errors for rank mismatching for a fixed $\boldsymbol{M}$ with $\mathrm{rank}(\boldsymbol{M}) = 10$.

approximation of the kernel matrix $\boldsymbol{M}$ can be efficiently constructed with a smaller scale factorization. The effective sampling rate for Nyström method is $p_{\mathrm{Nys}} = \frac{2 \times 50n - 50^2}{n^2} \approx 0.01$. In contrast, in addition to recording the selected entry values, our nonconvex optimization method also requires to record the row and column indices for each selected entry. By using sparse matrix storage schemes like compressed sparse row (CSR) format (Saad, 2003), it needs $2n^2 p_{NCVX} + n + 1$ entries to store the sparse matrix. Therefore, if $p_{NCVX} \geqslant \frac{3}{n}$, the nonconvex approach requires at most 2.5 times as much memory as Nyström method for the same sampling complexity. Therefore, we choose the sampling rate $p_{\mathrm{NCVX}} = \frac{p_{\mathrm{Nys}}}{2.5}$ in the implementation of the nonconvex optimization (1) such that the memory consumption is less costly than the Nyström method.



(a) Relative error $\frac{\|\boldsymbol{M}_{\text{approx}}-\boldsymbol{M}_r\|_F}{\|\boldsymbol{M}_r\|_F}$. (b) Relative error $\frac{\|\boldsymbol{M}_{\text{approx}}-\boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}$.

Figure 7: Relative errors for rank mismatching, fixed selected rank.

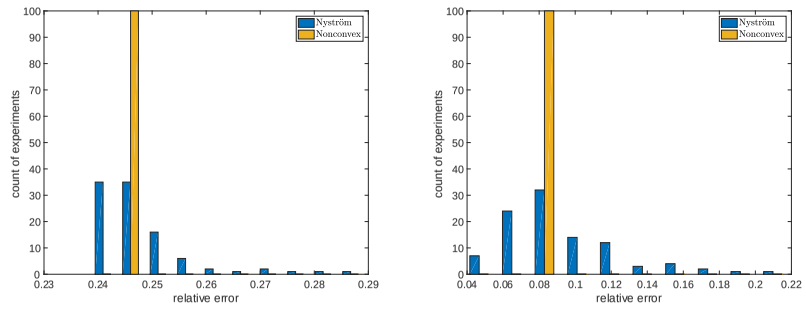
Fixing such a synthetic data set, we apply both the Nyström method and our approach (with $\alpha = 100\|\boldsymbol{M}\|_{\ell_\infty} = 100$ and $\lambda = 500\sqrt{np_{\text{NCVX}}}$) for 100 times. Denote by $\boldsymbol{M}$ the ground truth of the kernel matrix, by $\boldsymbol{M}_2$ the ground truth of the best rank-2 approximation of $\boldsymbol{M}$, and by $\boldsymbol{M}_{\text{approx}}$ the memory efficient rank-2 approximation obtained by Nyström method or our nonconvex optimization. The left and right panels of Figure 8 compare the two methods in approximating $\boldsymbol{M}_2$ and $\boldsymbol{M}$ respectively based on the distributions of relative errors throughout the 100 Monte Carlo simulations. One can see that our approach is comparable with the Nyström method in terms of median performance, but much more stable.

Both Nyström method and our nonconvex optimization (1) give approximation in the form of $\boldsymbol{M} \approx \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top$, so clustering analysis can be directly implemented based on $\widehat{\boldsymbol{X}}$. We implement k-means on the rows of $\widehat{\boldsymbol{X}}$ with 20 repetitions, and Figure 9 compares the two methods in the distribution of clustering accuracies. It clearly shows that our nonconvex optimization (1) yields accurate clustering throughout the 100 tests while the Nyström method results in poor clustering occasionally.

Moreover, during the iterations of the nonconvex method, the regularization term never activate throughout the 100 simulations. Therefore, empirically speaking, the performances of our numerical tests will remain the same if we simply set $\lambda = 0$.

4. Proofs

In this section, we give a proof for main theorem. In Section 4.1, we will present some useful supporting lemmas; in Section 4.2, we present a proof for our main result Theorem 2; finally in Section 4.3 we give proof of lemmas used in former subsections. Our proof ideas benefit from those in Ge et al. (2017) as well as Zhu et al. (2017), Jin et al. (2017).



(a) Relative error $\frac{\|\boldsymbol{M}_{\text{approx}}-\boldsymbol{M}_2\|_F}{\|\boldsymbol{M}_2\|_F}$. (b) Relative error $\frac{\|\boldsymbol{M}_{\text{approx}}-\boldsymbol{M}\|_F}{\|\boldsymbol{M}\|_F}$.

Figure 8: Relative errors for Nyström method with sampling rate $p_{\text{Nys}} \approx 0.01$ and nonconvex method with sampling rate $p_{\text{NCVX}} = \frac{p_{\text{Nys}}}{2.5}$.

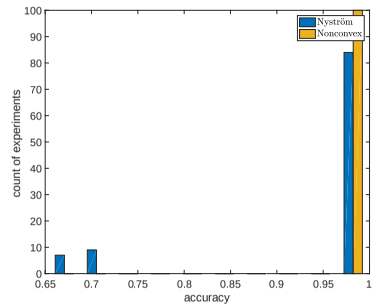


Figure 9: Clustering accuracy for Nyström method with sampling rate $p_{\text{Nys}} \approx 0.01$ and nonconvex method with sampling rate $p_{\text{NCVX}} = \frac{p_{\text{Nys}}}{2.5}$.

4.1. Supporting lemmas

In this section, we give some useful supporting lemmas. The following lemma is well known in the literature, see, e.g., Vu (2018) and Bandeira et al. (2016).

Lemma 4 *There is a constant $C_v > 0$ such that the following holds. If Ω is sampled according to the off-diagonal symmetric Ber(p) model, then*

$$\mathbb{P}\left[\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \geqslant C_v\sqrt{np(1-p)} + C_v\sqrt{\log n}\right] \leqslant n^{-3}.$$

The following eigen-space incoherence parameter has been proposed in Candès and Recht (2009).

Definition 5 (Candès and Recht 2009) *For any subspace $\mathcal{U}$ of $\mathbb{R}^n$ of dimension r, denote $\boldsymbol{P}_{\mathcal{U}} : \mathbb{R}^n \to \mathbb{R}^n$ as the orthogonal projection onto $\mathcal{U}$. Define*

$$\mu(\mathcal{U}) := \frac{n}{r} \max_{1 \leqslant i \leqslant n} \|\boldsymbol{P}_{\mathcal{U}} \boldsymbol{e}_i\|_2^2, \tag{9}$$

where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ represents the standard orthogonal basis of $\mathbb{R}^n$.

As with Theorem 4.1 in Candès and Recht (2009), for the off-diagonal symmetric Ber(p) model, we also have:

Lemma 6 *Let Ω be sampled according to the off-diagonal symmetric Ber(p) model. Define*

$$\mathcal{T} := \{\boldsymbol{M} \in \mathbb{R}^{n \times n} \mid (\boldsymbol{I} - \boldsymbol{P}_{\mathcal{U}})\boldsymbol{M}(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{U}}) = \boldsymbol{0},\ \boldsymbol{M} \text{ symmetric}\},$$

where $\mathcal{U}$ is a fixed subspace of $\mathbb{R}^n$. Let $\mathcal{P}_{\mathcal{T}}$ be the Euclidean projection on to $\mathcal{T}$: For any symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$,

$$\mathcal{P}_{\mathcal{T}}(\boldsymbol{M}) = \boldsymbol{P}_{\mathcal{U}}\boldsymbol{M} + \boldsymbol{M}\boldsymbol{P}_{\mathcal{U}} - \boldsymbol{P}_{\mathcal{U}}\boldsymbol{M}\boldsymbol{P}_{\mathcal{U}}.$$

Then there is an absolute constant C_c, such that for any $\delta \in (0, 1]$, if $p \geqslant C_c \frac{\mu(\mathcal{U}) \dim(\mathcal{U}) \log n}{\delta^2 n}$ with $\mu(\mathcal{U})$ defined in (9), in an event E_c with probability $\mathbb{P}[E_c] \geqslant 1 - n^{-3}$, we have

$$p^{-1}\|\mathcal{P}_{\mathcal{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathcal{T}} - p\mathcal{P}_{\mathcal{T}}\| \leqslant \delta.$$

In Gross (2011) and Gross and Nesme (2010), similar results are given for symmetric uniform sampling with/without replacement. The proof of Lemma 6 is very similar to that in Recht (2011).

The first and second order optimality conditions of $f(\boldsymbol{X})$ satisfy the following properties:

Lemma 7 (Ge et al. 2016, Proposition 4.1) *The first order optimality condition of objective function (1) is*

$$\nabla f(\boldsymbol{X}) = 2\mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{M})\boldsymbol{X} + \lambda \nabla G_\alpha(\boldsymbol{X}) = \boldsymbol{0},$$

and the second order optimality condition requires that for any $\boldsymbol{H} \in \mathbb{R}^{n \times r}$, we have

$$\begin{aligned}&\operatorname{vec}(\boldsymbol{H})^\top \nabla^2 f(\boldsymbol{X}) \operatorname{vec}(\boldsymbol{H})\\ =&\|\mathcal{P}_{\Omega}(\boldsymbol{H}\boldsymbol{X}^\top + \boldsymbol{X}\boldsymbol{H}^\top)\|_F^2 + 2\langle \mathcal{P}_{\Omega}(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{M}), \mathcal{P}_{\Omega}(\boldsymbol{H}\boldsymbol{H}^\top)\rangle + \lambda \operatorname{vec}(\boldsymbol{H})^\top \nabla^2 G_\alpha(\boldsymbol{X}) \operatorname{vec}(\boldsymbol{H})\\ \geqslant& 0.\end{aligned}$$

In the sequel, we are going to present our key lemma which will be used multiple times throughout this section. For any matrix $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathbb{R}^{n_1 \times n_2}$, any set $\Omega_0 \in [n_1] \times [n_2]$ and any real number $t \in \mathbb{R}$, we introduce following notation for simplicity of notations:

$$D_{\Omega_0, t}(\boldsymbol{M}_1, \boldsymbol{M}_2) := \langle \mathcal{P}_{\Omega_0}(\boldsymbol{M}_1), \mathcal{P}_{\Omega_0}(\boldsymbol{M}_2)\rangle - t\langle \boldsymbol{M}_1, \boldsymbol{M}_2\rangle. \tag{10}$$

Our key lemma is given as follows:

Lemma 8 *Let Ω_0 be any index set in $[n_1] \times [n_2]$, and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{n_1 \times n_2}$ be defined correspondingly as in Section 2.1. For any $\boldsymbol{A} \in \mathbb{R}^{n_1 \times r_1}, \boldsymbol{B} \in \mathbb{R}^{n_1 \times r_2}, \boldsymbol{C} \in \mathbb{R}^{n_2 \times r_1}, \boldsymbol{D} \in \mathbb{R}^{n_2 \times r_2}$, and any $t \in \mathbb{R}$, there holds*

$$|D_{\Omega_0, t}(\boldsymbol{A}\boldsymbol{C}^\top, \boldsymbol{B}\boldsymbol{D}^\top)| \leqslant \|\boldsymbol{\Omega}_0 - t\boldsymbol{J}\| \sqrt{\sum_{k=1}^{n_1} \|\boldsymbol{A}_{k,\cdot}\|_2^2 \|\boldsymbol{B}_{k,\cdot}\|_2^2} \sqrt{\sum_{k=1}^{n_2} \|\boldsymbol{C}_{k,\cdot}\|_2^2 \|\boldsymbol{D}_{k,\cdot}\|_2^2}. \tag{11}$$

We will use this result for $\Omega_0 = \Omega, t = p$ for multiple times later. Note that here we do not make any assumptions on Ω_0 and this is a deterministic result. The proof of this lemma is deferred to Section 4.3.1. This result extends the following lemma given in Bhojanapalli and Jain (2014) and Li et al. (2016b):

Lemma 9 (Bhojanapalli and Jain 2014; Li et al. 2016b) *Suppose matrix $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ can be decomposed as $\boldsymbol{M} = \boldsymbol{B}\boldsymbol{D}^\top$, let $\Omega_0 \subset [n_1] \times [n_2]$ be any index set. Then for any $t \in \mathbb{R}$, we have*

$$\|\mathcal{P}_{\Omega_0}(\boldsymbol{M}) - t\boldsymbol{M}\| \leqslant \|\boldsymbol{\Omega}_0 - t\boldsymbol{J}\| \|\boldsymbol{B}\|_{2,\infty} \|\boldsymbol{D}\|_{2,\infty}.$$

Lemma 8 is applied in our proof of Lemma 12 in replace of Theorem D.1 in Ge et al. (2016) to derive tighter control of perturbation terms, i.e., $K_2(\boldsymbol{X}), K3(\boldsymbol{X})$ and $K4(\boldsymbol{X})$ defined in (14). Their result is given here for the purpose of comparison.

Lemma 10 (Ge et al. 2016, Theorem D.1) *With high probability over the choice of Ω, for any two rank-r matrices $\boldsymbol{W}, \boldsymbol{Z} \in \mathbb{R}^{n \times n}$, we have*

$$\begin{aligned}&|\langle \mathcal{P}_{\Omega}(\boldsymbol{W}), \mathcal{P}_{\Omega}(\boldsymbol{Z})\rangle - p\langle \boldsymbol{W}, \boldsymbol{Z}\rangle|\\ \leqslant& O\left(\|\boldsymbol{W}\|_{\ell_\infty}\|\boldsymbol{Z}\|_{\ell_\infty} nr \log n + \sqrt{pnr\|\boldsymbol{W}\|_{\ell_\infty}\|\boldsymbol{Z}\|_{\ell_\infty}\|\boldsymbol{W}\|_F\|\boldsymbol{Z}\|_F \log n}\right).\end{aligned}$$

In Sun and Luo (2016), Chen and Wainwright (2015) and Zheng and Lafferty (2016), upper bounds are given to $\|\mathcal{P}_{\Omega}(\boldsymbol{H}\boldsymbol{H}^\top)\|_F^2$ for any $\boldsymbol{H}$. To be more precise, they assume Ω is sampled according to the i.i.d. Bernoulli model with probability p. If $p \geqslant C\frac{\log n}{n}$ for some sufficient large absolute constant C, there holds

$$\|\mathcal{P}_{\Omega}(\boldsymbol{H}\boldsymbol{H}^\top)\|_F^2 - p\|\boldsymbol{H}\|_F^4 \leqslant C\sqrt{np}\sum_{i=1}^n \|\boldsymbol{H}_{i,\cdot}\|_2^4 \tag{12}$$

with high probability. In contrast, by combining Lemma 4 and Lemma 8, there holds

$$|\|\mathcal{P}_{\Omega}(\boldsymbol{H}\boldsymbol{H}^\top)\|_F^2 - p\|\boldsymbol{H}\boldsymbol{H}^\top\|_F^2| \leqslant C\sqrt{np}\sum_{i=1}^n \|\boldsymbol{H}_{i,\cdot}\|_2^4 \tag{13}$$

with high probability. This is tighter than (12) in that $\|\boldsymbol{H}\boldsymbol{H}^\top\|_F \leqslant \|\boldsymbol{H}\|_F^2$. Moreover, comparing to (12), our result (13) directly measures the difference between $\|\mathcal{P}_{\Omega}(\boldsymbol{H}\boldsymbol{H}^\top)\|_F^2$ and its expectation $p\|\boldsymbol{H}\boldsymbol{H}^\top\|_F^2$, which makes the model-free analysis possible.

4.2. A proof of Theorem 2

This section aims to prove Theorem 2. The proof is basically divided into two parts: In Section 4.2.1, we discuss the landscape of objective function $f(\boldsymbol{X})$ and then define the auxiliary function $K(\boldsymbol{X})$. We show that the span of local minima of $f(\boldsymbol{X})$ can be controlled by the superlevel set of $K(\boldsymbol{X})$: $\{\boldsymbol{X} \in \mathbb{R}^{n\times r} \mid K(\boldsymbol{X}) \geqslant 0\}$. In Section 4.2.2, we give a uniform upper bound of $K(\boldsymbol{X})$ in order to control the above superlevel set.

4.2.1. Landscape of objective function f and auxiliary function K

Denote $\boldsymbol{U}_r \coloneqq [\sqrt{\sigma_1}\boldsymbol{u}_1 \ \dots \ \sqrt{\sigma_r}\boldsymbol{u}_r]$. For a given $\boldsymbol{X} \in \mathbb{R}^{n\times r}$, suppose that $\boldsymbol{X}^\top\boldsymbol{U}_r$ has SVD $\boldsymbol{X}^\top\boldsymbol{U}_r = \boldsymbol{A}\boldsymbol{D}\boldsymbol{B}^\top$, and let $\boldsymbol{R}_{\boldsymbol{X},\boldsymbol{U}_r} \coloneqq \boldsymbol{B}\boldsymbol{A}^\top \in \mathsf{O}(r)$ and $\boldsymbol{U} \coloneqq \boldsymbol{U}_r\boldsymbol{R}_{\boldsymbol{X},\boldsymbol{U}_r}$, where $\mathsf{O}(r)$ denotes the set of $r\times r$ orthogonal matrices $\{\boldsymbol{R} \in \mathbb{R}^{r\times r} \mid \boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}\}$. Then $\boldsymbol{X}^\top\boldsymbol{U} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^\top$ is a positive semidefinite matrix. Then also holds $\boldsymbol{U}_r\boldsymbol{U}_r^\top = \boldsymbol{U}\boldsymbol{U}^\top$.

Denote $\boldsymbol{\Delta} \coloneqq \boldsymbol{X} - \boldsymbol{U}$, and define the following auxiliary function introduced in Jin et al. (2017) and Ge et al. (2017):

$$K(\boldsymbol{X}) \coloneqq \operatorname{vec}(\boldsymbol{\Delta})^\top\nabla^2 f(\boldsymbol{X})\operatorname{vec}(\boldsymbol{\Delta}) - 4\langle\nabla f(\boldsymbol{X}), \boldsymbol{\Delta}\rangle.$$

The first and second order optimality conditions for any local minimum $\widehat{\boldsymbol{X}}$ imply that $K(\widehat{\boldsymbol{X}}) \geqslant 0$. In other words, we have

$$\{\text{All local minima of } f(\boldsymbol{X})\} \subset \{\boldsymbol{X} \in \mathbb{R}^{n\times r} \mid K(\boldsymbol{X}) \geqslant 0\}.$$

To study the properties of the local minima of $f(\boldsymbol{X})$, we can consider the superlevel set of $K(\boldsymbol{X})$: $\{\boldsymbol{X} \in \mathbb{R}^{n\times r} \mid K(\boldsymbol{X}) \geqslant 0\}$ instead. In order to get a clear representation of $K(\boldsymbol{X})$, one can plug in the formulas of gradient and Hessian in Lemma 7. By repacking terms in Ge et al. (2017, Lemma 7), and given $\langle\boldsymbol{U}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle = 0$, due to the definition of $\boldsymbol{U}$ and $\boldsymbol{N}$, $K(\boldsymbol{X})$ can be decomposed as follows:

Lemma 11 (Ge et al. 2017, Lemma 7) *Uniformly for all $\boldsymbol{X} \in \mathbb{R}^{n\times r}$, as well as corresponding $\boldsymbol{U}$ and $\boldsymbol{\Delta}$ defined above, we have*

$$\begin{aligned}
K(\boldsymbol{X}) =& \underbrace{p\left(\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 - 3\|\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2\right)}_{K_1(\boldsymbol{X})} \\
&+ \underbrace{D_{\Omega,p}(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{\Delta}\boldsymbol{\Delta}^\top) - 3D_{\Omega,p}(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top)}_{K_2(\boldsymbol{X})} \\
&+ \underbrace{\lambda\left(\operatorname{vec}(\boldsymbol{\Delta})^\top\nabla^2 G_\alpha(\boldsymbol{X})\operatorname{vec}(\boldsymbol{\Delta}) - 4\langle\nabla G_\alpha(\boldsymbol{X}), \boldsymbol{\Delta}\rangle\right)}_{K_3(\boldsymbol{X})} \\
&+ \underbrace{6D_{\Omega,p}(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}) + 8D_{\Omega,p}(\boldsymbol{U}\boldsymbol{\Delta}^\top, \boldsymbol{N}) + 6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle}_{K_4(\boldsymbol{X})},
\end{aligned} \tag{14}$$

where $D_{\Omega,p}(\cdot,\cdot)$ is defined in (10).

Notice that in Theorem 2, we are only concerned about the difference between $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{M}_r$ (or $\boldsymbol{M}$), which remains the same by replacing $\boldsymbol{X}$ with $\widetilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{R}$, for any $\boldsymbol{R} \in \mathsf{O}(r)$. On the other hand, by the definition of $\boldsymbol{R}_{\boldsymbol{X},\boldsymbol{U}_r}$, we have $\boldsymbol{R}_{\boldsymbol{X}\boldsymbol{R},\boldsymbol{U}_r} = \boldsymbol{R}_{\boldsymbol{X},\boldsymbol{U}_r}\boldsymbol{R}$ for any $\boldsymbol{R} \in \mathsf{O}(r)$, which implies $\widetilde{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{R}$ and $\widetilde{\boldsymbol{\Delta}} = \boldsymbol{\Delta}\boldsymbol{R}$. Now we have

$$\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^\top = \boldsymbol{X}\boldsymbol{X}^\top, \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}^\top = \boldsymbol{U}\boldsymbol{U}^\top, \widetilde{\boldsymbol{\Delta}}\widetilde{\boldsymbol{\Delta}}^\top = \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Delta}}^\top = \boldsymbol{U}\boldsymbol{\Delta}^\top,$$

which means $K_i(\widetilde{\boldsymbol{X}}) = K_i(\boldsymbol{X})$ for $i = 1, 2, 4$. As for K_3, by Ge et al. (2017, Lemma 18), we have

$$\begin{aligned}
&\operatorname{vec}(\boldsymbol{\Delta})^\top\nabla^2 G_\alpha(\boldsymbol{X})\operatorname{vec}(\boldsymbol{\Delta}) - 4\langle\nabla G_\alpha(\boldsymbol{X}), \boldsymbol{\Delta}\rangle \\
=&4\sum_{i=1}^n[(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^3\frac{\|\boldsymbol{X}_{i,\cdot}\|_2^2\|\boldsymbol{\Delta}_{i,\cdot}\|_2^2 - \langle\boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot}\rangle^2}{\|\boldsymbol{X}_{i,\cdot}\|_2^3} + 12\sum_{i=1}^n[(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^2\frac{\langle\boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot}\rangle^2}{\|\boldsymbol{X}_{i,\cdot}\|_2^2} \\
&- 16\sum_{i=1}^n[(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^3\frac{\langle\boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot}\rangle}{\|\boldsymbol{X}_{i,\cdot}\|_2}.
\end{aligned}$$

Since $\boldsymbol{R} \in \mathsf{O}(r)$, we have $\|\widetilde{\boldsymbol{X}}_{i,\cdot}\|_2 = \|\boldsymbol{X}_{i,\cdot}\|_2$, $\|\widetilde{\boldsymbol{\Delta}}_{i,\cdot}\|_2 = \|\boldsymbol{\Delta}_{i,\cdot}\|_2$ and $\langle\widetilde{\boldsymbol{X}}_{i,\cdot}, \widetilde{\boldsymbol{\Delta}}_{i,\cdot}\rangle = \langle\boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot}\rangle$, so we have $K_3(\widetilde{\boldsymbol{X}}) = K_3(\boldsymbol{X})$. Putting things together, we have $K(\widetilde{\boldsymbol{X}}) = K(\boldsymbol{X})$.

Therefore, if we want to show that any $\boldsymbol{X}$ with $K(\boldsymbol{X}) \geqslant 0$ satisfies (4) and (5) with high probability, without loss of generality, we can assume that $\boldsymbol{X}$ satisfies the property that $\boldsymbol{X}^\top\boldsymbol{U}_r$ is a positive semidefinite matrix, i.e., $\boldsymbol{U} = \boldsymbol{U}_r$.

4.2.2. Proof of Theorem 2.

In order to prove our main result, we first give a uniform upper bound of $K(\boldsymbol{X})$. Then for any local minimum $\widehat{\boldsymbol{X}}$, $K(\widehat{\boldsymbol{X}}) \geqslant 0$, the property enables us to solve for the range of possible $\widehat{\boldsymbol{X}}$. For simplicity of notations, denote $\nu_r \coloneqq \|\boldsymbol{M}_r\|_{\ell_\infty}$.

Lemma 12 *Assume that tuning parameters α, λ satisfy $100\sqrt{\nu_r} \leqslant \alpha \leqslant 200\sqrt{\nu_r}$, $100\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \leqslant \lambda \leqslant 200\|\boldsymbol{\Omega} - p\boldsymbol{J}\|$, and $p \geqslant C_S\frac{\log n}{n}$ with some absolute constant C_S. Then, in an event E with probability $\mathbb{P}[E] \geqslant 1 - 2n^{-3}$, uniformly for all $\boldsymbol{X} \in \mathbb{R}^{n\times r}$ and corresponding $\boldsymbol{\Delta}$ defined as before, we have*

$$\begin{aligned}
\sum_{i=2}^4 K_i(\boldsymbol{X}) \leqslant& 10^{-3}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2 + \|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right] \\
&+ C_3 p\sum_{i=1}^r\left\{\left[C_4\left(\sqrt{\frac{n}{p}} + \frac{\log n}{p}\right)\nu_r + C_4\sigma_{2r+1-i} - \sigma_i\right]_+\right\}^2 \\
&+ C_3\frac{[p(1-p)n + \log n]r\|\boldsymbol{N}\|_{\ell_\infty}^2}{p}.
\end{aligned} \tag{15}$$

Note in our proof of Theorem 2, we only use probabilistic tools in the above lemma to control perturbation terms, i.e., $K_2(\boldsymbol{X}), K_3(\boldsymbol{X}), K_4(\boldsymbol{X})$. The rest part of the proof is purely deterministic.

Recall by the way we define $\boldsymbol{\Delta}$,

$$\begin{aligned}\|\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2 =& \|\boldsymbol{U}\boldsymbol{\Delta}^\top + \boldsymbol{\Delta}\boldsymbol{U}^\top + \boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 \\ =& \|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 + 2\|\boldsymbol{\Delta}\boldsymbol{U}^\top\|_F^2 + 2\langle \boldsymbol{\Delta}\boldsymbol{U}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle + 4\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle.\end{aligned} \tag{16}$$

By the definition of matrix inner product, we have

$$\begin{aligned}\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 =& \langle \boldsymbol{U}\boldsymbol{\Delta}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle = \mathrm{trace}(\boldsymbol{\Delta}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{\Delta}^\top) = \mathrm{trace}(\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{\Delta}^\top\boldsymbol{\Delta}) \\ =& \langle \boldsymbol{U}^\top\boldsymbol{U}, \boldsymbol{\Delta}^\top\boldsymbol{\Delta}\rangle,\end{aligned} \tag{17}$$

and

$$\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle = \mathrm{trace}(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\boldsymbol{U}\boldsymbol{\Delta}^\top) = \mathrm{trace}(\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\boldsymbol{U}) = \langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, \boldsymbol{\Delta}^\top\boldsymbol{U}\rangle. \tag{18}$$

Here we use the fact that $\mathrm{trace}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{trace}(\boldsymbol{B}\boldsymbol{A})$ for any matrix $\boldsymbol{A}$ and $\boldsymbol{B}$ with suitable size. Moreover, since we choose $\boldsymbol{U}$ such that $\boldsymbol{U}^\top\boldsymbol{X}$ is positive semidefinite, $\boldsymbol{U}^\top\boldsymbol{\Delta} = \boldsymbol{\Delta}^\top\boldsymbol{U}$ and $\boldsymbol{U}^\top(\boldsymbol{\Delta} + \boldsymbol{U}) \succeq \boldsymbol{0}$. Therefore, we also have

$$\begin{aligned}\langle \boldsymbol{\Delta}\boldsymbol{U}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle =& \mathrm{trace}(\boldsymbol{U}\boldsymbol{\Delta}^\top\boldsymbol{U}\boldsymbol{\Delta}^\top) = \mathrm{trace}(\boldsymbol{\Delta}\boldsymbol{U}^\top\boldsymbol{\Delta}\boldsymbol{U}^\top) = \mathrm{trace}(\boldsymbol{U}^\top\boldsymbol{\Delta}\boldsymbol{U}^\top\boldsymbol{\Delta}) \\ =& \langle \boldsymbol{\Delta}^\top\boldsymbol{U}, \boldsymbol{U}^\top\boldsymbol{\Delta}\rangle = \langle \boldsymbol{\Delta}^\top\boldsymbol{U}, \boldsymbol{\Delta}^\top\boldsymbol{U}\rangle = \|\boldsymbol{\Delta}^\top\boldsymbol{U}\|_F^2\end{aligned} \tag{19}$$

and

$$\langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, \boldsymbol{U}^\top\boldsymbol{U} + \boldsymbol{\Delta}^\top\boldsymbol{U}\rangle = \langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, (\boldsymbol{U} + \boldsymbol{\Delta})^\top\boldsymbol{U}\rangle \geqslant 0. \tag{20}$$

Here (20) also uses the fact that inner product of two positive semidefinite matrices is non-negative.

Now denote $a := \|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F = \|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F, b := \|\boldsymbol{\Delta}^\top\boldsymbol{U}\|_F$ and

$$\psi := C_3\left\{\sum_{i=1}^r\left\{\left[C_4\left(\sqrt{\frac{n}{p}} + \frac{\log n}{p}\right)\nu_r + C_4\sigma_{2r+1-i} - \sigma_i\right]_+\right\}^2 + \frac{[p(1-p)n + \log n]r\|\boldsymbol{N}\|_{\ell_\infty}^2}{p^2}\right\}.$$

Putting Lemma 11 and Lemma 12 together, and using (16), we have

$$\begin{aligned}\frac{K(\boldsymbol{X})}{p} \leqslant& 1.001\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 - 3\|\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2 + 10^{-3}\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + \psi \\ =& 1.001a^2 - 3\left[\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 + 2\|\boldsymbol{\Delta}\boldsymbol{U}^\top\|_F^2 + 2\langle \boldsymbol{\Delta}\boldsymbol{U}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle + 4\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle\right] \\ &+ 10^{-3}\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + \psi,\end{aligned} \tag{21}$$

By putting (17), (18), (19), (21) together,

$$\begin{aligned}\frac{K(\boldsymbol{X})}{p} \leqslant& 1.001a^2 - 3\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 - 6\|\boldsymbol{\Delta}\boldsymbol{U}^\top\|_F^2 - 6\langle \boldsymbol{\Delta}\boldsymbol{U}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle - 12\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top\rangle \\ &+ 10^{-3}\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + \psi \\ =& -1.999a^2 - 6\langle \boldsymbol{U}^\top\boldsymbol{U}, \boldsymbol{\Delta}^\top\boldsymbol{\Delta}\rangle - 6\|\boldsymbol{\Delta}^\top\boldsymbol{U}\|_F^2 - 12\langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, \boldsymbol{\Delta}^\top\boldsymbol{U}\rangle \\ &+ 10^{-3}\langle \boldsymbol{U}^\top\boldsymbol{U}, \boldsymbol{\Delta}^\top\boldsymbol{\Delta}\rangle + \psi \\ =& -1.999a^2 - \langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, 5.999\boldsymbol{U}^\top\boldsymbol{U} + 12\boldsymbol{\Delta}^\top\boldsymbol{U}\rangle - 6b^2 + \psi.\end{aligned} \tag{22}$$

Therefore, combining with (20),

$$\begin{aligned}\frac{K(\boldsymbol{X})}{p} \leqslant& -1.999a^2 - 6.001\langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, \boldsymbol{\Delta}^\top\boldsymbol{U}\rangle - 6b^2 + \psi \\ \leqslant& -1.999a^2 + 6.001ab - 6b^2 + \psi\end{aligned} \tag{23}$$

holds for all $\boldsymbol{X} \in \mathbb{R}^{n\times r}$. For the last line, we apply Cauchy-Schwarz inequality for matrices, i.e.,

$$|\langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, \boldsymbol{\Delta}^\top\boldsymbol{U}\rangle| \leqslant \|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F\|\boldsymbol{\Delta}^\top\boldsymbol{U}\|_F.$$

Note that for any local minimum $\widehat{\boldsymbol{X}}$, we have $K(\widehat{\boldsymbol{X}}) \geqslant 0$. Replacing $\boldsymbol{X}$ with $\widehat{\boldsymbol{X}}$ in (23), there holds

$$-1.999a^2 + 6.001ab - 6b^2 + \psi \geqslant 0,$$

which further implies

$$0 \leqslant a \leqslant C_5\sqrt{\psi}, \qquad 0 \leqslant b \leqslant C_5\sqrt{\psi}. \tag{24}$$

From (22), we have

$$\frac{K(\widehat{\boldsymbol{X}})}{p} \leqslant -1.999a^2 - \langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, 5.999\boldsymbol{U}^\top\boldsymbol{U} + 12\boldsymbol{\Delta}^\top\boldsymbol{U}\rangle - 6b^2 + \psi.$$

Recall from (17), $\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 = \langle \boldsymbol{U}^\top\boldsymbol{U}, \boldsymbol{\Delta}^\top\boldsymbol{\Delta}\rangle$, and $K(\widehat{\boldsymbol{X}}) \geqslant 0$. Therefore, combining with (24),

$$\begin{aligned}5.999\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 \leqslant& -1.999a^2 - \langle \boldsymbol{\Delta}^\top\boldsymbol{\Delta}, 12\boldsymbol{\Delta}^\top\boldsymbol{U}\rangle - 6b^2 + \psi \\ \leqslant& -1.999a^2 + 12\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F\|\boldsymbol{\Delta}^\top\boldsymbol{U}\|_F - 6b^2 + \psi \\ \leqslant& -1.999a^2 + 12ab - 6b^2 + \psi \\ \leqslant& C_6\psi.\end{aligned} \tag{25}$$

From (21),

$$\frac{K(\widehat{\boldsymbol{X}})}{p} \leqslant 1.001\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 - 3\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2 + 10^{-3}\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + \psi.$$

Using the fact that $K(\widehat{\boldsymbol{X}}) \geqslant 0$ again, we have

$$3\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2 \leqslant 1.001\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 + 10^{-3}\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + \psi$$

Combining with (24), (25), we futher have

$$3\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2 \leqslant 1.001a^2 + C_7\psi + \psi \leqslant C_8\psi. \tag{26}$$

Therefore, (4) is directly implied by (26). Notice that

$$\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\|_F^2 = \|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2 - 2\langle \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top, \boldsymbol{N}\rangle + \|\boldsymbol{N}\|_F^2 \leqslant \|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{U}\boldsymbol{U}^\top\|_F^2 + \|\boldsymbol{N}\|_F^2$$

where the inequality holds since $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top \succeq \boldsymbol{0}$ and $\boldsymbol{N} \succeq \boldsymbol{0}$. Therefore, (5) is implied by (4).

4.3. Proofs of supporting lemmas

We present in this section the proofs of lemmas stated in previous sections.

4.3.1. A proof of Lemma 8

Proof First of all, by using the definition of matrix inner product and Hadamard product, we have

$$\begin{aligned}|\langle \mathcal{P}_{\Omega_0}(\boldsymbol{A}\boldsymbol{C}^\top), \mathcal{P}_{\Omega_0}(\boldsymbol{B}\boldsymbol{D}^\top)\rangle - t\langle \boldsymbol{A}\boldsymbol{C}^\top, \boldsymbol{B}\boldsymbol{D}^\top\rangle| =& |\langle \boldsymbol{\Omega}_0 - t\boldsymbol{J}, (\boldsymbol{A}\boldsymbol{C}^\top \circ \boldsymbol{B}\boldsymbol{D}^\top)\rangle| \\ \leqslant& \|\boldsymbol{\Omega}_0 - t\boldsymbol{J}\| \|(\boldsymbol{A}\boldsymbol{C}^\top \circ \boldsymbol{B}\boldsymbol{D}^\top)\|_*,\end{aligned} \tag{27}$$

The inequality holds by matrix Hölder's inequality. So the only thing left over is to give a bound of $\|(\boldsymbol{A}\boldsymbol{C}^\top \circ \boldsymbol{B}\boldsymbol{D}^\top)\|_*$. Notice one can decompose the matrix into sum of rank one matrices as following

$$\boldsymbol{A}\boldsymbol{C}^\top \circ \boldsymbol{B}\boldsymbol{D}^\top = \left(\sum_{k=1}^{r_1} \boldsymbol{A}_{\cdot,k}\boldsymbol{C}_{\cdot,k}^\top\right) \circ \left(\sum_{k=1}^{r_2} \boldsymbol{B}_{\cdot,k}\boldsymbol{D}_{\cdot,k}^\top\right) = \sum_{l=1}^{r_1}\sum_{m=1}^{r_2} (\boldsymbol{A}_{\cdot,l} \circ \boldsymbol{B}_{\cdot,m})(\boldsymbol{C}_{\cdot,l} \circ \boldsymbol{D}_{\cdot,m})^\top.$$

Recall $\boldsymbol{M}_{\cdot,j} = (M_{1,j}, M_{2,j}, \ldots, M_{n,j})^\top$ denotes the j-th column of any matrix $\boldsymbol{M} \in \mathbb{R}^{n\times m}$.

Therefore, one can upper bound the nuclear norm via

$$\begin{aligned}\|(\boldsymbol{A}\boldsymbol{C}^\top \circ \boldsymbol{B}\boldsymbol{D}^\top)\|_* \leqslant& \sum_{l=1}^{r_1}\sum_{m=1}^{r_2} \|(\boldsymbol{A}_{\cdot,l} \circ \boldsymbol{B}_{\cdot,m})(\boldsymbol{C}_{\cdot,l} \circ \boldsymbol{D}_{\cdot,m})^\top\|_* \\ =& \sum_{l=1}^{r_1}\sum_{m=1}^{r_2} \|\boldsymbol{A}_{\cdot,l} \circ \boldsymbol{B}_{\cdot,m}\|_2 \|\boldsymbol{C}_{\cdot,l} \circ \boldsymbol{D}_{\cdot,m}\|_2 \\ =& \sum_{l=1}^{r_1}\sum_{m=1}^{r_2} \sqrt{\sum_{k=1}^{n_1} A_{k,l}^2 B_{k,m}^2}\sqrt{\sum_{k=1}^{n_2} C_{k,l}^2 D_{k,m}^2},\end{aligned}$$

where the first line is by the triangle inequality and we can replace nuclear norm by vector ℓ_2 norms in second line since the summands are all rank one matrices. By applying the Cauchy-Schwarz inequality for twice, we can obtain

$$\begin{aligned}\|(\boldsymbol{A}\boldsymbol{C}^\top \circ \boldsymbol{B}\boldsymbol{D}^\top)\|_* \leqslant& \sqrt{\sum_{l=1}^{r_1}\sum_{m=1}^{r_2}\sum_{k=1}^{n_1} A_{k,l}^2 B_{k,m}^2}\sqrt{\sum_{l=1}^{r_1}\sum_{m=1}^{r_2}\sum_{k=1}^{n_2} C_{k,l}^2 D_{k,m}^2} \\ =& \sqrt{\sum_{k=1}^{n_1} \|\boldsymbol{A}_{k,\cdot}\|_2^2 \|\boldsymbol{B}_{k,\cdot}\|_2^2}\sqrt{\sum_{k=1}^{n_2} \|\boldsymbol{C}_{k,\cdot}\|_2^2 \|\boldsymbol{D}_{k,\cdot}\|_2^2}.\end{aligned} \tag{28}$$

Combining (27) and (28) together, we have

$$\begin{aligned}&|\langle \mathcal{P}_{\Omega_0}(\boldsymbol{A}\boldsymbol{C}^\top), \mathcal{P}_{\Omega_0}(\boldsymbol{B}\boldsymbol{D}^\top)\rangle - t\langle \boldsymbol{A}\boldsymbol{C}^\top, \boldsymbol{B}\boldsymbol{D}^\top\rangle| \\ \leqslant& \|\boldsymbol{\Omega}_0 - t\boldsymbol{J}\| \sqrt{\sum_{k=1}^{n_1} \|\boldsymbol{A}_{k,\cdot}\|_2^2 \|\boldsymbol{B}_{k,\cdot}\|_2^2}\sqrt{\sum_{k=1}^{n_2} \|\boldsymbol{C}_{k,\cdot}\|_2^2 \|\boldsymbol{D}_{k,\cdot}\|_2^2}.\end{aligned}$$

■

4.3.2. A proof of Lemma 12

Proof The proof of Lemma 12 can be divided into the controls of $K_2(\boldsymbol{X})$, $K_3(\boldsymbol{X})$ and $K_4(\boldsymbol{X})$ separately.

For $K_2(\boldsymbol{X})$, we have

Lemma 13 *In an event E_a with probability $\mathbb{P}[E_a] \geqslant 1 - n^{-3}$, uniformly for all $\boldsymbol{X} \in \mathbb{R}^{n\times r}$ and corresponding $\boldsymbol{\Delta}$ defined as before, we have*

$$K_2(\boldsymbol{X}) \leqslant \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[19\sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + 18\nu_r\|\boldsymbol{\Delta}\|_F^2 + 9\nu_r \sum_{i=s+1}^r \sigma_i\right] + 3\times 10^{-4} p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2,$$

where s is defined by

$$s := \max\left\{s \leqslant r,\ \sigma_s \geqslant C_p \frac{\nu_r \log n}{p}\right\} \tag{29}$$

with C_p an absolute constant. Set $s = 0$ if $\sigma_1 < C_p \frac{\nu_r \log n}{p}$.

For $K_3(\boldsymbol{X})$, we use a modified version of Ge et al. (2017, Lemma 11):

Lemma 14 (Ge et al. 2017, Lemma 11) *If $\alpha \geqslant 100\sqrt{\nu_r}$, then uniformly for all $\boldsymbol{X} \in \mathbb{R}^{n\times r}$ and corresponding $\boldsymbol{\Delta}$ defined as before, we have*

$$K_3(\boldsymbol{X}) \leqslant 199.54\lambda\alpha^2\|\boldsymbol{\Delta}\|_F^2 - 0.3\lambda\sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4.$$

The main modification we have made is that we keep the extra negative term. We will give a proof in the appendix for completeness.

For $K_4(\boldsymbol{X})$, we have

Lemma 15 *Uniformly for all $\boldsymbol{X} \in \mathbb{R}^{n\times r}$ and corresponding $\boldsymbol{\Delta}$ defined as before, we have*

$$\begin{aligned}K_4(\boldsymbol{X}) \leqslant& 5\times 10^{-4} p\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 + 2\times 10^{-4} p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + C_{10}\frac{r\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|^2}{p} \\ &+ 6p\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle.\end{aligned}$$

We can apply Lemma 4 together with Lemma 9 to bound $\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|$ and $\|\boldsymbol{\Omega} - p\boldsymbol{J}\|$ (similar result can also be found in Keshavan et al. (2010b)): As long as $p \geqslant C_S \frac{\log n}{n}$ with some absolute constant C_S, there is an absolute constant C_9, such that

$$\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\| \leqslant \left(C_9\sqrt{np(1-p)} + C_9\sqrt{\log n}\right)\|\boldsymbol{N}\|_{\ell_\infty} \tag{30}$$

and

$$\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \leqslant C_9\sqrt{np} \tag{31}$$

hold in an event E_b with probability $\mathbb{P}[E_b] \geqslant 1 - n^{-3}$.

By putting Lemma 13, Lemma 14 and Lemma 15 together, we have

$$\begin{aligned}\sum_{i=2}^{4} K_i(\boldsymbol{X}) \leqslant & \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[19\sum_{i=1}^{n}\|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + 18\nu_r\|\boldsymbol{\Delta}\|_F^2 + 9\nu_r\sum_{i=s+1}^{r}\sigma_i\right] + 3\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 \\ & + 199.54\lambda\alpha^2\|\boldsymbol{\Delta}\|_F^2 - 0.3\lambda\sum_{i=1}^{n}\|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + 5\times 10^{-4}p\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 \\ & + 2\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + C_{10}\frac{r\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|^2}{p} + 6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle.\end{aligned}$$

Replacing α, λ by the assumption $100\sqrt{\nu_r} \leqslant \alpha \leqslant 200\sqrt{\nu_r}, 100\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \leqslant \lambda \leqslant 200\|\boldsymbol{\Omega} - p\boldsymbol{J}\|$, we futher have

$$\begin{aligned}\sum_{i=2}^{4} K_i(\boldsymbol{X}) \leqslant & \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[19\sum_{i=1}^{n}\|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + 18\nu_r\|\boldsymbol{\Delta}\|_F^2 + 9\nu_r\sum_{i=s+1}^{r}\sigma_i\right] + 3\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 \\ & + 7.9816\times 10^8\nu_r\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\|\boldsymbol{\Delta}\|_F^2 - 30\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\sum_{i=1}^{n}\|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + 5\times 10^{-4}p\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 \\ & + 2\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + C_{10}\frac{r\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|^2}{p} + 6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle.\end{aligned}$$

Combining with (30) and (31), and applying union bound,

$$\begin{aligned}\sum_{i=2}^{4} K_i(\boldsymbol{X}) \leqslant & (19-30)\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\sum_{i=1}^{n}\|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + (18 + 7.9816\times 10^8)\nu_r\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\|\boldsymbol{\Delta}\|_F^2 \\ & + 9\nu_r\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\sum_{i=s+1}^{r}\sigma_i + (3+2)\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 \\ & + 5\times 10^{-4}p\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 + C_{10}\frac{r\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|^2}{p} + 6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle \\ \leqslant & 5\times 10^{-4}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2 + \|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right] + C_3\frac{p(1-p)n + \log n}{p}r\|\boldsymbol{N}\|_{\ell_\infty}^2 \\ & + C_{11}\sqrt{np}\nu_r\|\boldsymbol{\Delta}\|_F^2 + C_{12}\sqrt{np}\nu_r\sum_{i=s+1}^{r}\sigma_i + 6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle,\end{aligned} \tag{32}$$

holds in an event E with probability $\mathbb{P}[E] \geqslant 1 - 2n^{-3}$.

For $\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2$, we have

$$\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2 = \langle\boldsymbol{\Delta}^\top\boldsymbol{\Delta}, \boldsymbol{\Delta}^\top\boldsymbol{\Delta}\rangle = \sum_{i=1}^{r}\sigma_i^4(\boldsymbol{\Delta}), \tag{33}$$

where $\sigma_i(\boldsymbol{\Delta})$ denotes i-th largest singular value of $\boldsymbol{\Delta}$.

In order to proceed, we need the following von Neumann's trace inequality:

Lemma 16 (Bhatia 2013, Problem III.6.14) *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ be two symmetric matrices, $\lambda_1(\boldsymbol{A}) \geqslant \lambda_2(\boldsymbol{A}) \geqslant \cdots \geqslant \lambda_n(\boldsymbol{A})$ and $\lambda_1(\boldsymbol{B}) \geqslant \lambda_2(\boldsymbol{B}) \geqslant \cdots \geqslant \lambda_n(\boldsymbol{B})$ are eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$. Then the following holds:*

$$\sum_{i=1}^{n}\lambda_i(\boldsymbol{A})\lambda_{n+1-i}(\boldsymbol{B}) \leqslant \langle\boldsymbol{A}, \boldsymbol{B}\rangle \leqslant \sum_{i=1}^{n}\lambda_i(\boldsymbol{A})\lambda_i(\boldsymbol{B}).$$

This result can also be derived from Schur-Horn theorem (see, e.g., Marshall et al. (2011, Theorem 9.B.1, Theorem 9.B.2)) together with Abel's summation formula.

From Lemma 16, we have

$$\begin{aligned}\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 &= \operatorname{trace}(\boldsymbol{\Delta}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{\Delta}^\top) = \langle\boldsymbol{U}^\top\boldsymbol{U}, \boldsymbol{\Delta}^\top\boldsymbol{\Delta}\rangle \\ &\geqslant \sum_{i=1}^{r}\lambda_{r+1-i}(\boldsymbol{U}^\top\boldsymbol{U})\lambda_i(\boldsymbol{\Delta}^\top\boldsymbol{\Delta}) = \sum_{i=1}^{r}\sigma_i^2(\boldsymbol{\Delta})\sigma_{r+1-i}^2(\boldsymbol{U}),\end{aligned} \tag{34}$$

and

$$\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle \leqslant \sum_{i=1}^{n}\lambda_i(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top)\lambda_i(\boldsymbol{N}) = \sum_{i=1}^{r}\sigma_i^2(\boldsymbol{\Delta})\sigma_i(\boldsymbol{N}). \tag{35}$$

Here we use the fact that $\lambda_i(\boldsymbol{U}^\top\boldsymbol{U}) = \sigma_i^2(\boldsymbol{U})$, $\lambda_i(\boldsymbol{\Delta}^\top\boldsymbol{\Delta}) = \sigma_i^2(\boldsymbol{\Delta})$, $\lambda_i(\boldsymbol{N}) = \sigma_i(\boldsymbol{N})$ and

$$\lambda_i(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top) = \begin{cases}\sigma_i^2(\boldsymbol{\Delta}) & i = 1, \cdots, r \\ 0 & i = r+1, \cdots, n.\end{cases}$$

Putting (33), (34) and (35) together we have

$$\begin{aligned}& -5\times 10^{-4}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2 + \|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right] + C_{11}\sqrt{np}\nu_r\|\boldsymbol{\Delta}\|_F^2 + 6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle \\ \leqslant & -5\times 10^{-4}p\left[\sum_{i=1}^{r}\sigma_i^4(\boldsymbol{\Delta}) + \sum_{i=1}^{r}\sigma_i^2(\boldsymbol{\Delta})\sigma_{r+1-i}^2(\boldsymbol{U})\right] + C_{11}\sqrt{np}\nu_r\sum_{i=1}^{r}\sigma_i^2(\boldsymbol{\Delta}) \\ & + 6p\sum_{i=1}^{r}\sigma_i^2(\boldsymbol{\Delta})\sigma_i(\boldsymbol{N}) \\ \leqslant & 5\times 10^{-4}p\sum_{i=1}^{r}\left\{-\sigma_i^4(\boldsymbol{\Delta}) + \left[C_{13}\sqrt{\frac{n}{p}}\nu_r - \sigma_{r+1-i}^2(\boldsymbol{U}) + C_{13}\sigma_i(\boldsymbol{N})\right]\sigma_i^2(\boldsymbol{\Delta})\right\}.\end{aligned}$$

For the last line, the summands are a series of quadratic functions of $\sigma_i^2(\boldsymbol{\Delta})$. Noticing the fact that for a quadratic function $q(x) = -x^2 + bx$, given the constraint $x \geqslant 0$, the maximum

is taken over $\hat{x} = \frac{1}{2}[b]_+$, and the maximum value is $\frac{1}{4}\{[b]_+\}^2$. Therefore, we have

$$\begin{aligned}
&-5\times 10^{-4}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2+\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right]+C_{11}\sqrt{np}\nu_r\|\boldsymbol{\Delta}\|_F^2+6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top,\boldsymbol{N}\rangle\\
\leqslant& C_{14}p\sum_{i=1}^r\left\{\left[C_{13}\sqrt{\frac{n}{p}}\nu_r-\sigma_{r+1-i}^2(\boldsymbol{U})+C_{13}\sigma_i(\boldsymbol{N})\right]_+\right\}^2\\
=& C_{14}p\sum_{j=1}^r\left\{\left[C_{13}\sqrt{\frac{n}{p}}\nu_r-\sigma_j^2(\boldsymbol{U})+C_{13}\sigma_{r+1-j}(\boldsymbol{N})\right]_+\right\}^2\\
=& C_{14}p\sum_{j=1}^r\left\{\left[C_{13}\sqrt{\frac{n}{p}}\nu_r+C_{13}\sigma_{2r+1-j}-\sigma_j\right]_+\right\}^2.
\end{aligned}\tag{36}$$

In the second last line, we let $j = r+1-i$. In the last line, we use the fact that

$$\sigma_{r+1-j}(\boldsymbol{N}) = \sigma_{r+r+1-j}(\boldsymbol{M}) = \sigma_{2r+1-j}$$

and

$$\sigma_j^2(\boldsymbol{U}) = \sigma_j(\boldsymbol{U}\boldsymbol{U}^\top) = \sigma_j(\boldsymbol{M}_r) = \sigma_j(\boldsymbol{M}) = \sigma_j.$$

Finally putting (32) and (36) together we have

$$\begin{aligned}
\sum_{i=2}^4 K_i(\boldsymbol{X}) \leqslant& 10\times 10^{-4}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2+\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right]+C_3\frac{p(1-p)n+\log n}{p}r\|\boldsymbol{N}\|_{\ell_\infty}^2\\
&-5\times 10^{-4}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2+\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right]+C_{11}\sqrt{np}\nu_r\|\boldsymbol{\Delta}\|_F^2\\
&+C_{12}\sqrt{np}\nu_r\sum_{i=s+1}^r\sigma_i+6p\langle\boldsymbol{\Delta}\boldsymbol{\Delta}^\top,\boldsymbol{N}\rangle\\
\leqslant& 10^{-3}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2+\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right]+C_3\frac{p(1-p)n+\log n}{p}r\|\boldsymbol{N}\|_{\ell_\infty}^2\\
&+C_{14}p\sum_{i=1}^r\left\{\left[C_{13}\sqrt{\frac{n}{p}}\nu_r+C_{13}\sigma_{2r+1-i}-\sigma_i\right]_+\right\}^2+C_{12}\sqrt{np}\nu_r\sum_{i=s+1}^r\sigma_i.
\end{aligned}\tag{37}$$

Recall by the definition of s in (29), for any $i > s$, we have $\sigma_i < C_p\frac{\nu_r\log n}{p}$. By choosing C_{13} sufficient large, i.e., $C_{13} \geqslant 2C_p$, we have

$$\begin{aligned}
C_{13}\left(\sqrt{\frac{n}{p}}+\frac{\log n}{p}\right)\nu_r+C_{13}\sigma_{2r+1-i}-\sigma_i \geqslant& C_{13}\left(\sqrt{\frac{n}{p}}+\frac{\log n}{p}\right)\nu_r-2\sigma_i+\sigma_i\\
\geqslant& C_{13}\left(\sqrt{\frac{n}{p}}+\frac{\log n}{p}\right)\nu_r-2C_p\frac{\nu_r\log n}{p}+\sigma_i\\
=& C_{13}\sqrt{\frac{n}{p}}\nu_r+\sigma_i+(C_{13}-2C_p)\frac{\nu_r\log n}{p}\\
\geqslant& C_{13}\sqrt{\frac{n}{p}}\nu_r+\sigma_i \geqslant 0.
\end{aligned}$$

Therefore, for all $i > s$,

$$\left\{\left[C_{13}\left(\sqrt{\frac{n}{p}}+\frac{\log n}{p}\right)\nu_r+C_{13}\sigma_{2r+1-i}-\sigma_i\right]_+\right\}^2 \geqslant \left[C_{13}\sqrt{\frac{n}{p}}\nu_r+\sigma_i\right]^2 \geqslant \sqrt{\frac{n}{p}}\nu_r\sigma_i.$$

Combining with (37), we have

$$\begin{aligned}
\sum_{i=2}^4 K_i(\boldsymbol{X}) \leqslant& 10^{-3}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2+\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right]\\
&+C_{14}p\sum_{i=1}^r\left\{\left[C_{13}\left(\sqrt{\frac{n}{p}}+\frac{\log n}{p}\right)\nu_r+C_{13}\sigma_{2r+1-i}-\sigma_i\right]_+\right\}^2\\
&+C_{12}p\sum_{i=s+1}^r\left\{\left[C_{13}\left(\sqrt{\frac{n}{p}}+\frac{\log n}{p}\right)\nu_r+C_{13}\sigma_{2r+1-i}-\sigma_i\right]_+\right\}^2\\
&+C_3\frac{p(1-p)n+\log n}{p}r\|\boldsymbol{N}\|_{\ell_\infty}^2\\
\leqslant& 10^{-3}p\left[\|\boldsymbol{\Delta}^\top\boldsymbol{\Delta}\|_F^2+\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2\right]\\
&+C_3p\sum_{i=1}^r\left\{\left[C_4\left(\sqrt{\frac{n}{p}}+\frac{\log n}{p}\right)\nu_r+C_4\sigma_{2r+1-i}-\sigma_i\right]_+\right\}^2\\
&+C_3\frac{p(1-p)n+\log n}{p}r\|\boldsymbol{N}\|_{\ell_\infty}^2
\end{aligned}$$

which finishes the proof. ■

4.3.3. A proof of Lemma 13

Proof Recall that we define $\boldsymbol{\Delta}$ as $\boldsymbol{\Delta} \coloneqq \boldsymbol{X}-\boldsymbol{U}$, $D_{\Omega,p}(\boldsymbol{X}\boldsymbol{X}^\top-\boldsymbol{U}\boldsymbol{U}^\top,\boldsymbol{X}\boldsymbol{X}^\top-\boldsymbol{U}\boldsymbol{U}^\top)$ can be decomposed as following

$$\begin{aligned}
&D_{\Omega,p}(\boldsymbol{X}\boldsymbol{X}^\top-\boldsymbol{U}\boldsymbol{U}^\top,\boldsymbol{X}\boldsymbol{X}^\top-\boldsymbol{U}\boldsymbol{U}^\top)\\
=&D_{\Omega,p}(\boldsymbol{U}\boldsymbol{\Delta}^\top+\boldsymbol{\Delta}\boldsymbol{U}^\top+\boldsymbol{\Delta}\boldsymbol{\Delta}^\top,\boldsymbol{U}\boldsymbol{\Delta}^\top+\boldsymbol{\Delta}\boldsymbol{U}^\top+\boldsymbol{\Delta}\boldsymbol{\Delta}^\top)\\
=&\underbrace{D_{\Omega,p}(\boldsymbol{U}\boldsymbol{\Delta}^\top+\boldsymbol{\Delta}\boldsymbol{U}^\top,\boldsymbol{U}\boldsymbol{\Delta}^\top+\boldsymbol{\Delta}\boldsymbol{U}^\top)}_{①}+\underbrace{D_{\Omega,p}(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top,\boldsymbol{\Delta}\boldsymbol{\Delta}^\top)}_{②}+\underbrace{4D_{\Omega,p}(\boldsymbol{U}\boldsymbol{\Delta}^\top,\boldsymbol{\Delta}\boldsymbol{\Delta}^\top)}_{③}.
\end{aligned}\tag{38}$$

Here we use the fact that Ω is symmetric. Our strategy here is using Lemma 6 to give a tight bound to as many as possible terms, for those terms that Lemma 6 cannot handle, we use Lemma 8 to give a bound. To be more precise, for ② and ③, as Lemma 6 cannot apply here, we use Lemma 8 to give a bound. For ①, we need to split it into two parts, the good part we can use Lemma 6 to control, and the rest part we use Lemma 8 to give a bound.

First for ② and ③, by applying Lemma 8,

$$|②| = |D_{\Omega,p}(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{\Delta}\boldsymbol{\Delta}^\top)| \leqslant \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 \tag{39}$$

and

$$\begin{aligned} |③| = 4|D_{\Omega,p}(\boldsymbol{U}\boldsymbol{\Delta}^\top, \boldsymbol{\Delta}\boldsymbol{\Delta}^\top)| \leqslant & 4\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sqrt{\sum_{i=1}^n \|\boldsymbol{U}_{i,\cdot}\|_2^2 \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2} \sqrt{\sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4} \\ \leqslant & 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \nu_r \|\boldsymbol{\Delta}\|_F^2 + 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4, \end{aligned} \tag{40}$$

where for the second inequality we use the fact that $2xy \leqslant x^2 + y^2$.

Finally for ①, if $\boldsymbol{U}$ is good enough such that the incoherence $\mu(\boldsymbol{U})$ is well-bounded, then we can apply Lemma 6 directly and get a tight bound. If $\mu(\boldsymbol{U})$ is not good enough, we want to split $\boldsymbol{U}$ into two parts and hope first few columns have good incoherence. To be more precise, recall that we assume $\boldsymbol{U} = \boldsymbol{U}_r = [\sqrt{\sigma_1}\boldsymbol{u}_1 \ \dots \ \sqrt{\sigma_r}\boldsymbol{u}_r]$, similar to (8), for the incoherence of the first k columns, we have

$$\begin{aligned} & \mu\left(\text{colspan}([\sqrt{\sigma_1}\boldsymbol{u}_1 \ \dots \ \sqrt{\sigma_k}\boldsymbol{u}_k])\right) \\ = & \frac{n}{k} \max_i \sum_{j=1}^k u_{i,j}^2 \leqslant \frac{n}{k\sigma_k} \max_i \sum_{j=1}^k \sigma_j u_{i,j}^2 \leqslant \frac{n}{k\sigma_k} \max_i \sum_{j=1}^r \sigma_j u_{i,j}^2 \leqslant \frac{n\nu_r}{k\sigma_k}, \end{aligned} \tag{41}$$

where $\mu(\cdot)$ is defined in (9).

For fixed s defined as in (29), denote first s columns of $\boldsymbol{U}$ as $\boldsymbol{U}^1$, and remaining part as $\boldsymbol{U}^2$. Decompose $\boldsymbol{U}$ as $\boldsymbol{U} = [\boldsymbol{U}^1 \ \boldsymbol{U}^2]$, and $\boldsymbol{\Delta}$ can also be decomposed as $\boldsymbol{\Delta} = [\boldsymbol{\Delta}^1 \ \boldsymbol{\Delta}^2]$ correspondingly. Note by our assumption that $\boldsymbol{U} = \boldsymbol{U}_r$, we have $(\boldsymbol{U}^1)^\top \boldsymbol{U}^2 = \boldsymbol{0}$. So we can further decompose the first term of (38) as

$$\begin{aligned} ① = & D_{\Omega,p}(\boldsymbol{U}\boldsymbol{\Delta}^\top + \boldsymbol{\Delta}\boldsymbol{U}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top + \boldsymbol{\Delta}\boldsymbol{U}^\top) \\ = & D_{\Omega,p}\Big([\boldsymbol{U}^1 \ \boldsymbol{U}^2][\boldsymbol{\Delta}^1 \ \boldsymbol{\Delta}^2]^\top + [\boldsymbol{\Delta}^1 \ \boldsymbol{\Delta}^2][\boldsymbol{U}^1 \ \boldsymbol{U}^2]^\top, [\boldsymbol{U}^1 \ \boldsymbol{U}^2][\boldsymbol{\Delta}^1 \ \boldsymbol{\Delta}^2]^\top \\ & \quad + [\boldsymbol{\Delta}^1 \ \boldsymbol{\Delta}^2][\boldsymbol{U}^1 \ \boldsymbol{U}^2]^\top\Big) \\ = & \underbrace{D_{\Omega,p}\left(\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top + \boldsymbol{\Delta}^1(\boldsymbol{U}^1)^\top, \boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top + \boldsymbol{\Delta}^1(\boldsymbol{U}^1)^\top\right)}_{A_1} \\ & + \underbrace{4D_{\Omega,p}\left(\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top, \boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top\right)}_{A_2} + \underbrace{2D_{\Omega,p}\left(\boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top, \boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top\right)}_{A_3} \\ & + \underbrace{2D_{\Omega,p}\left(\boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top, \boldsymbol{\Delta}^2(\boldsymbol{U}^2)^\top\right)}_{A_4} + \underbrace{4D_{\Omega,p}\left(\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top, \boldsymbol{\Delta}^2(\boldsymbol{U}^2)^\top\right)}_{A_5}. \end{aligned} \tag{42}$$

Now we can apply tight approximation Lemma 6 to the first term of (42). By the way we choose s, combining with (41),

$$p \geqslant C_p \frac{\nu_r \log n}{\sigma_s} \geqslant C_p \frac{\nu_r \log n}{\sigma_s} \cdot \frac{\mu\left(\text{colspan}(\boldsymbol{U}^1)\right) s\sigma_s}{n\nu_r} = C_p \frac{\mu\left(\text{colspan}(\boldsymbol{U}^1)\right) s \log n}{n}.$$

By choosing C_p sufficient large, Lemma 6 ensures that

$$\begin{aligned} |A_1| = & \left|D_{\Omega,p}\left(\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top + \boldsymbol{\Delta}^1(\boldsymbol{U}^1)^\top, \boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top + \boldsymbol{\Delta}^1(\boldsymbol{U}^1)^\top\right)\right| \\ \leqslant & 2.5 \times 10^{-5} p \|\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top + \boldsymbol{\Delta}^1(\boldsymbol{U}^1)^\top\|_F^2 \\ \leqslant & 5 \times 10^{-5} p (\|\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top\|_F^2 + \|\boldsymbol{\Delta}^1(\boldsymbol{U}^1)^\top\|_F^2) \\ \leqslant & 10^{-4} p \|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 \end{aligned} \tag{43}$$

hold in an event E_a with probability $\mathbb{P}[E_a] \geqslant 1 - n^{-3}$, where the second inequality uses the fact that $(x+y)^2 \leqslant 2x^2 + 2y^2$, and last inequality uses the fact that $(\boldsymbol{U}^1)^\top \boldsymbol{U}^2 = \boldsymbol{0}$.

For the rest terms in (42), by applying Lemma 8 we have

$$\begin{aligned} |A_2| = & 4|D_{\Omega,p}(\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top, \boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top)| \\ \leqslant & 4\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sqrt{\sum_{i=1}^n \|\boldsymbol{U}_{i,\cdot}^1\|_2^2 \|\boldsymbol{U}_{i,\cdot}^2\|_2^2} \sqrt{\sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}^1\|_2^2 \|\boldsymbol{\Delta}_{i,\cdot}^2\|_2^2} \\ \leqslant & 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[\nu_r \|\boldsymbol{U}^2\|_F^2 + \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4\right] \end{aligned} \tag{44}$$

for the second term in (42), where the second inequality use the fact that $\|\boldsymbol{U}_{i,\cdot}^1\|_2^2 \leqslant \|\boldsymbol{U}_{i,\cdot}\|_2^2 \leqslant \nu_r, \|\boldsymbol{\Delta}_{i,\cdot}^1\|_2^2 \leqslant \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2, \|\boldsymbol{\Delta}_{i,\cdot}^2\|_2^2 \leqslant \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2$ and $2xy \leqslant x^2 + y^2$. For the third term, applying Lemma 8 again we have

$$\begin{aligned} |A_3| = & 2|D_{\Omega,p}(\boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top, \boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top)| \\ \leqslant & 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sqrt{\sum_{i=1}^n \|\boldsymbol{U}_{i,\cdot}^2\|_2^4} \sqrt{\sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}^2\|_2^4} \\ \leqslant & \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[\nu_r \|\boldsymbol{U}^2\|_F^2 + \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4\right], \end{aligned} \tag{45}$$

where for the second inequality we also use the properties used in bounding second term. For the fourth and last term in (42), applying Lemma 8 and properties listed above, we have

$$\begin{aligned} |A_4| = & 2|D_{\Omega,p}(\boldsymbol{U}^2(\boldsymbol{\Delta}^2)^\top, \boldsymbol{\Delta}^2(\boldsymbol{U}^2)^\top)| \\ \leqslant & 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sum_{i=1}^n \|\boldsymbol{U}_{i,\cdot}^2\|_2^2 \|\boldsymbol{\Delta}_{i,\cdot}^2\|_2^2 \leqslant 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \nu_r \|\boldsymbol{\Delta}\|_F^2 \end{aligned} \tag{46}$$

and

$$
\begin{aligned}
|A_5| =& 4|D_{\Omega,p}(\boldsymbol{U}^1(\boldsymbol{\Delta}^1)^\top, \boldsymbol{\Delta}^2(\boldsymbol{U}^2)^\top)| \\
\leqslant& 4\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sqrt{\sum_{i=1}^n \|\boldsymbol{U}^1_{i,\cdot}\|_2^2 \|\boldsymbol{\Delta}^2_{i,\cdot}\|_2^2} \sqrt{\sum_{i=1}^n \|\boldsymbol{U}^2_{i,\cdot}\|_2^2 \|\boldsymbol{\Delta}^1_{i,\cdot}\|_2^2} \\
\leqslant& 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\nu_r \|\boldsymbol{\Delta}^1\|_F^2 + 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\nu_r\|\boldsymbol{\Delta}^2\|_F^2 \\
=& 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\nu_r\|\boldsymbol{\Delta}\|_F^2.
\end{aligned} \tag{47}
$$

Now putting estimations of terms in (42) listed above together, i.e., (43), (44), (45), (46) and (47), we have

$$
\begin{aligned}
|\textcircled{1}| =& |D_{\Omega,p}(\boldsymbol{U}\boldsymbol{\Delta}^\top + \boldsymbol{\Delta}\boldsymbol{U}^\top, \boldsymbol{U}\boldsymbol{\Delta}^\top + \boldsymbol{\Delta}\boldsymbol{U}^\top)| \\
\leqslant& |A_1| + |A_2| + |A_3| + |A_4| + |A_5| \\
\leqslant& 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[\nu_r\|\boldsymbol{U}^2\|_F^2 + \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4\right] \\
&+ \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[\nu_r\|\boldsymbol{U}^2\|_F^2 + \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4\right] + 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\nu_r\|\boldsymbol{\Delta}\|_F^2 \\
&+ 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\nu_r\|\boldsymbol{\Delta}\|_F^2 \\
\leqslant& \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[3\nu_r\|\boldsymbol{U}^2\|_F^2 + 3\sum_{i=1}^n \|\boldsymbol{\Delta}_i\|_2^4 + 4\nu_r\|\boldsymbol{\Delta}\|_F^2\right] + 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2.
\end{aligned} \tag{48}
$$

Plugging estimations (39), (40) and (48) back to (38), we have

$$
\begin{aligned}
&|D_{\Omega,p}(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top)| \\
\leqslant& |\textcircled{1}| + |\textcircled{2}| + |\textcircled{3}| \\
\leqslant& \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[3\nu_r\|\boldsymbol{U}^2\|_F^2 + 3\sum_{i=1}^n \|\boldsymbol{\Delta}_i\|_2^4 + 4\nu_r\|\boldsymbol{\Delta}\|_F^2\right] + 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 \\
&+ \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\|\nu_r\|\boldsymbol{\Delta}\|_F^2 + 2\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 \\
=& \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[3\nu_r\|\boldsymbol{U}^2\|_F^2 + 6\sum_{i=1}^n \|\boldsymbol{\Delta}_i\|_2^4 + 6\nu_r\|\boldsymbol{\Delta}\|_F^2\right] + 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2.
\end{aligned}
$$

Therefore, combining with (39), we have

$$
\begin{aligned}
K_2(\boldsymbol{X}) \leqslant& |D_{\Omega,p}(\boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{\Delta}\boldsymbol{\Delta}^\top)| + 3|D_{\Omega,p}(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top, \boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{U}\boldsymbol{U}^\top)| \\
\leqslant& \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 + 3\|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[3\nu_r\|\boldsymbol{U}^2\|_F^2 + 6\sum_{i=1}^n \|\boldsymbol{\Delta}_i\|_2^4 + 6\nu_r\|\boldsymbol{\Delta}\|_F^2\right] \\
&+ 3\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 \\
\leqslant& \|\boldsymbol{\Omega} - p\boldsymbol{J}\| \left[19\sum_{i=1}^n \|\boldsymbol{\Delta}_i\|_2^4 + 18\nu_r\|\boldsymbol{\Delta}\|_F^2 + 9\nu_r \sum_{i=s+1}^r \sigma_i\right] + 3\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2.
\end{aligned}
$$

The last line uses the fact that $\|\boldsymbol{U}^2\|_F^2 = \sum_{i=s+1}^r \sigma_i$.

■

4.3.4. A proof of Lemma 15

Proof First, by matrix Hölder's inequality,

$$
6|\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\rangle| \leqslant 6\frac{\sqrt{p}\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_*}{\sqrt{r}} \frac{\sqrt{r}\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|}{\sqrt{p}}.
$$

Since $\boldsymbol{\Delta}\boldsymbol{\Delta}^\top$ is at most rank-r, $\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_* \leqslant \sqrt{r}\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F$. Therefore,

$$
\begin{aligned}
6|\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\rangle| \leqslant& 6\sqrt{p}\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F \frac{\sqrt{r}\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|}{\sqrt{p}} \\
\leqslant& 5\times 10^{-4}p\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 + C_{15}\frac{r\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|^2}{p}.
\end{aligned}
$$

For the last inequality, we also use the fact that $2xy \leqslant wx^2 + \frac{y^2}{w}$ for all $w > 0$. Use the same argument we also have

$$
8|\langle \boldsymbol{U}\boldsymbol{\Delta}^\top, \mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\rangle| \leqslant 2\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + C_{15}\frac{r\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|^2}{p},
$$

Therefore, by the way we define $K_4(\boldsymbol{X})$ in (14), we have

$$
\begin{aligned}
K_4(\boldsymbol{X}) \leqslant& |6\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \mathcal{P}_\Omega(\boldsymbol{N})\rangle - 6p\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle| + |8\langle \boldsymbol{U}\boldsymbol{\Delta}^\top, \mathcal{P}_\Omega(\boldsymbol{N})\rangle - 8p\langle \boldsymbol{U}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle| \\
&+ 6p\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle \\
\leqslant& 5\times 10^{-4}p\|\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\|_F^2 + 2\times 10^{-4}p\|\boldsymbol{U}\boldsymbol{\Delta}^\top\|_F^2 + C_{10}\frac{r\|\mathcal{P}_\Omega(\boldsymbol{N}) - p\boldsymbol{N}\|^2}{p} \\
&+ 6p\langle \boldsymbol{\Delta}\boldsymbol{\Delta}^\top, \boldsymbol{N}\rangle.
\end{aligned}
$$

■

5. Discussions

This paper studies low-rank approximation of a positive semidefinite matrix from partial entries via nonconvex optimization. We established a model-free theory for local-minimum based low-rank approximation without any assumptions on its rank, condition number or eigenspace incoherence parameter. We have also improved the state-of-the-art sampling rate results for nonconvex matrix completion with no spurious local minima in Ge et al. (2016, 2017), and have investigated the performance of the proposed nonconvex optimization in presence of large condition numbers, large incoherence parameters, or rank mismatching. The nonconvex optimization is further applied to the problem of memory-efficient kernel PCA. Compared to the well-known Nyström methods, numerical experiments illustrate that the proposed nonconvex optimization approach yields more stable results in both low-rank approximation and clustering.

For future research, we are interested in understanding whether and how fast first-order methods converge to a neighborhood of the set of local minima with theoretical guarantees. In fact, a series of recent works in nonconvex optimization have discussed why and when first-order iterative algorithms can avoid strict saddle points almost surely. For example, in a very recent work by Lee et al. (2017), the authors show that under mild conditions of the nonconvex objective function, a variety of first order algorithms can avoid strict saddle points with almost all initialization, which extends the previous results in Lee et al. (2016) and Panageas and Piliouras (2017). We are particularly interested in the robust version of the strict saddle points condition discussed in Ge et al. (2015) and Jin et al. (2017), referred to as (θ, γ, ζ)-strict saddle, under which noisy stochastic/deterministic gradient descent methods are proven to converge to a neighborhood of the local minima. In fact, Ge et al. (2017, Theorem 12) shows that the nonconvex optimization (1) satisfies certain (θ, γ, ζ)-strict saddle conditions as long as $\boldsymbol{M}$ is exactly of rank r, its condition number and eigenspace incoherence parameter are well-bounded, and the sampling rate is sufficiently large, but their argument cannot be straightforwardly extended to the model-free settings. We plan to explore the (θ, γ, ζ)-strict saddle conditions for (1) under a model-free framework in future.

Acknowledgments

We would like to acknowledge Taisong Jing for pointing us to the reference Bhatia (2013).

References

Dimitris Achlioptas and Frank McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9, 2007.

Dimitris Achlioptas, Frank McSherry, and Bernhard Schölkopf. Sampling techniques for kernel methods. In *Advances in Neural Information Processing Systems*, pages 335–342, 2002.

Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

Maria-Florina Balcan, Yingyu Liang, David P Woodruff, and Hongyang Zhang. Optimal sample complexity for matrix completion and related problems via ℓ_2-regularization. *arXiv preprint arXiv:1704.08683*, 2017.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

Afonso S Bandeira, Ramon Van Handel, et al. Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *The Annals of Probability*, 44(4):2479–2506, 2016.

Rajendra Bhatia. *Matrix Analysis*. Graduate Texts in Mathematics. Springer New York, 2013. ISBN 9781461206538.

Srinadh Bhojanapalli and Prateek Jain. Universal matrix completion. In *International Conference on Machine Learning*, pages 1881–1889, 2014.

Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873 – 3881, 2016.

Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

T Tony Cai, Xiaodong Li, Zongming Ma, et al. Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow. *The Annals of Statistics*, 44(5): 2221–2251, 2016.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Petros Drineas and Michael W Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(Dec):2153–2175, 2005.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points–online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242, 2017.

Gene H Golub and Charles F Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

Thore Graepel. Kernel matrix completion by semidefinite programming. In *International Conference on Artificial Neural Networks*, pages 694–699. Springer, 2002.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

David Gross and Vincent Nesme. Note on sampling without replacing from a finite collection of matrices. *arXiv preprint arXiv:1001.2738*, 2010.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732, 2017.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010a.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010b.

Kwang In Kim, Matthias O Franz, and Bernhard Schölkopf. Iterative kernel principal component analysis for image modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1351–1366, 2005.

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

Jason D. Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. First-order methods almost always avoid saddle points. *arXiv preprint arXiv:1710.07406*, 2017.

Qiuwei Li, Zhihui Zhu, and Gongguo Tang. Geometry of factored nuclear norm regularization. *arXiv preprint arXiv:1704.01265*, 2017.

Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei. Rapid, robust, and reliable blind deconvolution via nonconvex optimization. *Applied and computational harmonic analysis*, 2018.

Xingguo Li, Zhaoran Wang, Junwei Lu, Raman Arora, Jarvis Haupt, Han Liu, and Tuo Zhao. Symmetry, saddle points, and global geometry of nonconvex matrix factorization. *arXiv preprint arXiv:1612.09296*, 2016a.

Yuanzhi Li, Yingyu Liang, and Andrej Risteski. Recovery guarantee of weighted low-rank approximation via alternating minimization. In *International Conference on Machine Learning*, pages 2358–2367, 2016b.

Po-Ling Loh and Martin J. Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16(19):559–616, 2015.

Albert W Marshall, Ingram Olkin, and Barry C Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer, 2011.

Sahand Negahban and Martin J Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13(May):1665–1697, 2012.

John Paisley and Lawrence Carin. A nonparametric bayesian model for kernel matrix completion. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2090–2093. IEEE, 2010.

Ioannis Panageas and Georgios Piliouras. Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions. In *Innovations in Theoretical Computer Science*, 2017.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011.

Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*, volume 82. SIAM, 2003.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank solutions of linear matrix equations via procrustes flow. *arXiv preprint arXiv:1507.03566*, 2015.

Van Vu. A Simple SVD Algorithm for Finding Hidden Partitions. *Combinatorics, Probability and Computing*, 27(1):124–140, 2018.

Quan Wang. Kernel principal component analysis and its applications in face recognition and active shape models. *arXiv preprint arXiv:1207.3538*, 2012.

Christopher KI Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688, 2001.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in Neural Information Processing Systems*, pages 4152–4160, 2016.

Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. In *Advances in Neural Information Processing Systems*, pages 559–567, 2015.

Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems*, pages 109–117, 2015.

Qinqing Zheng and John Lafferty. Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.

Zhihui Zhu, Qiuwei Li, Gongguo Tang, and Michael B Wakin. The global optimization geometry of nonsymmetric matrix factorization and sensing. *arXiv preprint arXiv:1703.01256*, 2017.

Appendix A. Proof of Corollary 3

Proof The inequality (8) gives $\|\boldsymbol{M}\|_{\ell_\infty} \leqslant \frac{\mu_r r \sigma_1}{n}$. Therefore, in the case $\operatorname{rank}(\boldsymbol{M}) = r$, the approximation error bound (5) becomes

$$\left\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\right\|_F^2 \leqslant C_1 \sum_{i=1}^r \left\{ \left[C_2 \left(\sqrt{\frac{n}{p}} + \frac{\log n}{p} \right) \frac{\mu_r r}{n} \sigma_1 - \sigma_i \right]_+ \right\}^2.$$

Therefore, if

$$p \geqslant C \max\left\{ \frac{\mu_r r \kappa_r \log n}{n}, \frac{\mu_r^2 r^2 \kappa_r^2}{n} \right\}$$

with absolute constant C sufficient large, we have

$$C_2 \left(\sqrt{\frac{n}{p}} + \frac{\log n}{p} \right) \frac{\mu_r r}{n} \sigma_1 \leqslant \sigma_i, \quad i = 1, \cdots, r.$$

In other words, $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top = \boldsymbol{M}$.

Similarly, by definition (7), in the case $\operatorname{rank}(\boldsymbol{M}) = r$, we have

$$\|\boldsymbol{M}\|_{\ell_\infty} = \frac{\widetilde{\mu}_r^2 \operatorname{trace}(\boldsymbol{M})}{n} \leqslant \frac{\widetilde{\mu}_r^2 r \sigma_1}{n}.$$

Therefore, the approximation error bound (5) becomes

$$\left\|\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top - \boldsymbol{M}\right\|_F^2 \leqslant C_1 \sum_{i=1}^r \left\{ \left[C_2 \left(\sqrt{\frac{n}{p}} + \frac{\log n}{p} \right) \frac{\widetilde{\mu}_r^2 r}{n} \sigma_1 - \sigma_i \right]_+ \right\}^2.$$

Therefore, if

$$p \geqslant C \max\left\{ \frac{\widetilde{\mu}_r^2 r \kappa_r \log n}{n}, \frac{\widetilde{\mu}_r^4 r^2 \kappa_r^2}{n} \right\}$$

with absolute constant C sufficient large, we have $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^\top = \boldsymbol{M}$. ■

Appendix B. Proof of Lemma 14

Here we present a proof of Lemma 14. This proof is exactly the proof in Ge et al. (2017) except keeping the extra negative term. We include the proof in Ge et al. (2017) here for completeness.

Proof By Ge et al. (2017, Lemma 18), we have

$$\begin{aligned} &\operatorname{vec}(\boldsymbol{\Delta})^\top \nabla^2 G_\alpha(\boldsymbol{X}) \operatorname{vec}(\boldsymbol{\Delta}) - 4\langle \nabla G_\alpha(\boldsymbol{X}), \boldsymbol{\Delta} \rangle \\ =& 4 \sum_{i=1}^n [(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^3 \frac{\|\boldsymbol{X}_{i,\cdot}\|_2^2 \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2 - \langle \boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot} \rangle^2}{\|\boldsymbol{X}_{i,\cdot}\|_2^3} \\ &+ 12 \sum_{i=1}^n [(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^2 \frac{\langle \boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot} \rangle^2}{\|\boldsymbol{X}_{i,\cdot}\|_2^2} - 16 \sum_{i=1}^n [(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^3 \frac{\langle \boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot} \rangle}{\|\boldsymbol{X}_{i,\cdot}\|_2}. \end{aligned} \tag{49}$$

First of all, since we choose $\alpha \geqslant 100\sqrt{\mu_r} = 100\|\boldsymbol{U}\|_{2,\infty}$, then for all $\boldsymbol{X}_{i,\cdot}$ satisfying $\|\boldsymbol{X}_{i,\cdot}\|_2 \geqslant \alpha$, we have

$$\langle \boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot} \rangle = \langle \boldsymbol{X}_{i,\cdot}, \boldsymbol{X}_{i,\cdot} - \boldsymbol{U}_{i,\cdot} \rangle \geqslant \|\boldsymbol{X}_{i,\cdot}\|_2^2 - \|\boldsymbol{X}_{i,\cdot}\|_2 \|\boldsymbol{U}_{i,\cdot}\|_2 \geqslant (1 - 0.01)\|\boldsymbol{X}_{i,\cdot}\|_2^2 \geqslant 0.99\|\boldsymbol{X}_{i,\cdot}\|_2^2, \tag{50}$$

which gives an lower bound of the inner product between $\boldsymbol{X}_{i,\cdot}$ and $\boldsymbol{\Delta}_{i,\cdot}$. At the same time, we can also upper bound $\|\boldsymbol{\Delta}_{i,\cdot}\|_2$ by $\|\boldsymbol{X}_{i,\cdot}\|_2$:

$$\|\boldsymbol{\Delta}_{i,\cdot}\|_2 \leqslant \|\boldsymbol{X}_{i,\cdot}\|_2 + \|\boldsymbol{U}_{i,\cdot}\|_2 \leqslant 1.01\|\boldsymbol{X}_{i,\cdot}\|_2. \tag{51}$$

Plugging the above two estimations (50), (51) together with the fact that $|\langle \boldsymbol{X}_{i,\cdot}, \boldsymbol{\Delta}_{i,\cdot} \rangle|^2 \leqslant \|\boldsymbol{X}_{i,\cdot}\|_2^2 \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2$ into (49), we have

$$\begin{aligned} &\operatorname{vec}(\boldsymbol{\Delta})^\top \nabla^2 G_\alpha(\boldsymbol{X}) \operatorname{vec}(\boldsymbol{\Delta}) - 4\langle \nabla G_\alpha(\boldsymbol{X}), \boldsymbol{\Delta} \rangle \\ \leqslant& -15.68 \sum_{i=1}^n [(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^3 \|\boldsymbol{X}_{i,\cdot}\|_2 + 12 \sum_{i=1}^n [(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^2 \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2. \end{aligned} \tag{52}$$

Moreover, for all $\boldsymbol{X}_{i,\cdot}$ satisfies $\|\boldsymbol{X}_{i,\cdot}\|_2 \geqslant 5\alpha$, we can also upper bound $\|\boldsymbol{\Delta}_{i,\cdot}\|_2$ by $\|\boldsymbol{X}_{i,\cdot}\|_2$:

$$\|\boldsymbol{\Delta}_{i,\cdot}\|_2 \leqslant \|\boldsymbol{X}_{i,\cdot}\|_2 + \|\boldsymbol{U}_{i,\cdot}\|_2 \leqslant 1.002\|\boldsymbol{X}_{i,\cdot}\|_2, \tag{53}$$

and also lower bound $\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha$ by $\|\boldsymbol{\Delta}_{i,\cdot}\|_2$:

$$\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha \geqslant \left(1 - \frac{1}{5}\right)\|\boldsymbol{X}_{i,\cdot}\|_2 \geqslant \frac{400}{501}\|\boldsymbol{\Delta}_{i,\cdot}\|_2. \tag{54}$$

Plugging (53) and (54) back to (52), we have

$$\begin{aligned}
&\operatorname{vec}(\boldsymbol{\Delta})^\top \nabla^2 G_\alpha(\boldsymbol{X}) \operatorname{vec}(\boldsymbol{\Delta}) - 4\langle \nabla G_\alpha(\boldsymbol{X}), \boldsymbol{\Delta}\rangle \\
\leqslant & 12 \sum_{i, \|\boldsymbol{X}_{i,\cdot}\|_2 < 5\alpha} [(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^2 \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2 \\
& + \left[12 - 15.68 \times \frac{400}{501} \times \frac{1}{1.002}\right] \sum_{i, \|\boldsymbol{X}_{i,\cdot}\|_2 \geqslant 5\alpha} [(\|\boldsymbol{X}_{i,\cdot}\|_2 - \alpha)_+]^2 \|\boldsymbol{\Delta}_{i,\cdot}\|_2^2 \\
\leqslant & 192\alpha^2 \|\boldsymbol{\Delta}\|_F^2 - 0.3 \sum_{i, \|\boldsymbol{X}_{i,\cdot}\|_2 \geqslant 5\alpha} \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4 \\
\leqslant & 199.54\alpha^2 \|\boldsymbol{\Delta}\|_F^2 - 0.3 \sum_{i=1}^n \|\boldsymbol{\Delta}_{i,\cdot}\|_2^4,
\end{aligned}$$

where the last inequality uses the fact that $\|\boldsymbol{\Delta}_{i,\cdot}\|_2 \leqslant \|\boldsymbol{X}_{i,\cdot}\|_2 + \|\boldsymbol{U}_{i,\cdot}\|_2$ and $\alpha \geqslant 100\sqrt{\nu_r}$. ■

Journal of Machine Learning Research 20 (2019) 1-57 Submitted 2/18; Revised 8/19; Published 8/19

Minimal Sample Subspace Learning: Theory and Algorithms

Zhenyue Zhang ZYZHANG@ZJU.EDU.CN
School of Mathematics Science
Zhejiang University, Yuquan Campus
Hangzhou 310027, China

Zhejiang Laboratory
1818 Wenyixi Road
Hangzhou 311122, China

Yuqing Xia XIAYQ@ZJU.EDU.CN
School of Mathematics Science
Zhejiang University, Yuquan Campus
Hangzhou 310027, China

Editor: Francis Bach

Abstract

Subspace segmentation, or subspace learning, is a challenging and complicated task in machine learning. This paper builds a primary frame and solid theoretical bases for the minimal subspace segmentation (MSS) of finite samples. The existence and conditional uniqueness of MSS are discussed with conditions generally satisfied in applications. Utilizing weak prior information of MSS, the minimality inspection of segments is further simplified to the prior detection of partitions. The MSS problem is then modeled as a computable optimization problem via the self-expressiveness of samples. A closed form of the representation matrices is first given for the self-expressiveness, and the connection of diagonal blocks is addressed. The MSS model uses a rank restriction on the sum of segment ranks. Theoretically, it can retrieve the minimal sample subspaces that could be heavily intersected. The optimization problem is solved via a basic manifold conjugate gradient algorithm, alternative optimization and hybrid optimization, therein considering solutions to both the primal MSS problem and its pseudo-dual problem. The MSS model is further modified for handling noisy data and solved by an ADMM algorithm. The reported experiments show the strong ability of the MSS method to retrieve minimal sample subspaces that are heavily intersected.

Keywords: Subspace learning, Clustering, Rank restriction, Sparse optimization, Self-expressiveness

1. Introduction

Given a collection of vectors sampled from the union of several unknown low-dimensional subspaces that might intersect with each other, subspace learning, or subspace segmentation, aims to partition the samples into several segments such that each segment contains samples within the same subspace. If the segmentation is correct, the unknown subspaces are estimated well by the segments. The problem of subspace segmentation occurs in several applications. For instance, in single rigid motion, the trajectories of feature points lie in

©2019 Zhenyue Zhang and Yuqing Xia.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-084.html.

an affine subspace with a dimension of at most 3 (Costeira and Kanade, 1998).[1] Moreover, facial images of an individual under various lighting conditions lie in a linear subspace of dimension up to 9 (Basri and Jacobs, 2003). Detecting multiple rigidly moving objects from videos, and recognizing multiple individuals from facial images are potential subspace segmentation tasks.

Algorithms for subspace segmentation can be traced backe to early studies on algorithms such as RANSAC (Fischler and Bolles, 1981), K-subspace (Bradley and Mangasarian, 2000; Tseng, 2000), and generalized principal component analysis (GPCA, Vidal et al., 2005; Ma et al., 2008). In recent years, self-expressiveness methods such as low-rank representation (LRR, Liu et al., 2013) and sparse subspace clustering (SSC, Elhamifar and Vidal, 2013) have attracted a great deal of attention because of their state-of-art empirical performance. Given a set of column vectors X sampled from the union of several subspaces, such algorithms search for a representation matrix C of X, that is, $X = XC$, and try to detect the subspaces based on C. Ideally, the representation matrix C is block-diagonal as that $C = \mathrm{diag}(C_1, \cdots, C_K)$ under permutation. In this case, the samples are also partitioned into K pieces such as $X = [X_1, \cdots, X_K]$, and each X_k contains samples from the same subspace.

To estimate such a representation with the block-diagonal structure in C as much as possible, the SSC minimizes the ℓ_1-norm of C as follows:

$$\text{(SSC)} \quad \min_C \|C\|_1 \quad \text{s.t. } X = XC, \ \mathrm{diag}(C) = 0. \tag{1}$$

The restriction on the diagonals avoids a trivial and meaningless solution. The LRR method searches for a representation matrix with approximate low-rank structure that can take the so-called 'global structure' of the samples into account. Thus, it minimizes the nuclear norm $\|C\|_*$ (sum of the singular values) of C as follows:

$$\text{(LRR)} \quad \min_C \|C\|_* \quad \text{s.t. } X = XC. \tag{2}$$

The main difference is that SSC searches for the sparsest nontrivial representation matrix, while LRR searches for the representation matrix with the lowest rank. SSC and LRR seemingly impede subspace retrieval at two ends: an overly sparse solution may be block-diagonal with a greater-than-expected number of diagonal blocks, and a solution with an overly low-rank may not be block-diagonal or have less blocks. In these two cases, the ground-truth subspaces cannot be detected via classical spectral clustering. To control the number of blocks, Wang et al. (2013) combines the two objective functions. More purposefully, Xia and Zhang (2018) combines the ℓ_1-norm function with a logarithmic-determinant function to balance the sparsity and rank of the solution. Li et al. (2017) modifies the ℓ_1-norm function $\|C\|_1$ to the ℓ_1-norm of the off-diagonal blocks of C with a partition that should also be optimized. This strategy may help to increase the connection of the diagonal blocks in some sense.[2] Since the number of zero eigenvalues of the Laplacian matrix of a block-diagonal matrix is equal to the number of connected diagonal blocks

1. A d-dimensional affine subspace can be embedded into a $(d+1)$-dimensional linear subspace by adding 1 as a new entry to each vector.
2. We say that C_k is connected if the undirected graph constructed from $|C_k| + |C_k|^T$ is connected.

(Von Luxburg, 2007), Lu et al. (2019) minimizes several of the smallest eigenvalues of the Laplacian matrix of C to achieve a block-diagonal solution with connected diagonal blocks.

The effectiveness of these methods has scarcely been exploited in the literature. For instance, Soltanolkotabi and Candès (2012) gave sufficient conditions for SSC to retrieve a representation matrix that can detect subspaces. Liu et al. (2013) proved that LRR can recover mutually independent subspaces.[3] The above modified methods require equivalent or stronger conditions, which are generally very strict in applications.

The latent subspaces we wish to retrieve from samples are not well-defined mathematically, which may explain why theoretical progress in subspace learning has been slow. Subspace segmentation is practically ambiguous and unidentified in the literature. It is also highly possible that the segmented subspaces found by an algorithm may be defined by the algorithm used. For instance, in SSC, segmenting X into $\{X_k\}$ is equivalent to separating a constructed graph A into connected subgraphs $\{A_k\}$ via the following procedure: Let c_i minimize $\|c\|_1$ subjected to $x_i = X_{(i)}c$, where $X_{(i)}$ is the X whose i-th column is reset to zero, $i = 1, \cdots, n$. Graph A takes $\{x_i\}$ as its nodes and has an edge between nodes x_i and x_j if the j-th entry of c_i or the i-th entry of c_j is nonzero. Theoretically, $C = \mathrm{diag}(C_1, \cdots, C_K)$ is block-diagonal of connected blocks under permutation if and only if A has K connected subgraphs $\{A_k\}$. In that case, X_k consists of the samples as nodes involved in A_k. Clearly, the spanning subspaces $\{\mathrm{span}(X_k)\}$ depend on the connection structure of the constructed graph and cannot be predicted. In addition, the number of subspaces cannot be predicted. LRR gives a coarse segmentation corresponding to the independent subspaces, each a sum of several ground-truth subspaces, assuming that the ground-truth subspaces can be separated into several classes such that the subspace sums within classes are independent.[4]

This paper aims to build a theoretical basis for subspace learning from a mathematical viewpoint. The basic, important, and key issues that we keep in mind include the following:

(1) Identifiability of the subspaces that we wish to detect from a finite number of samples. The related basic issues for noiseless samples may include the definition of subspaces that are solely determined by samples, the uniqueness of the corresponding segmentation, the sufficient conditions for uniquely identifying the segmentation, and the consistency of the defined segmentation with the groundtruth segmentation that we expect in applications.

(2) Computability of the defined subspace segmentation. For application purpose, we may be required to formulate the defined segmentation as an optimization problem that should be computable with an acceptable cost. Related issues may include the uniqueness of the solution or conditions of the uniqueness, and the ability of addressing complicated segmentation wherein subspaces intersect with each other heavily, or samples are located near such intersected subspaces.

(3) Efficient algorithms for solving the optimization problem. We may also encounter efficiency issues with the adopted algorithms, such as computational complexity and local optimums.

(4) Stability of solutions and robustness of algorithms. It may be difficult but absolutely worth addressing these issues to further our understanding of subspace learning.

3. LRR cannot recover dependent subspaces; see Theorem 20 in Subsection 5.2.1.
4. This claim can be also concluded from Theorem 20.

(5) Extension to noisy samples which may be more important in applications. Certain necessary modifications are required to this end, together with perturbation theory on subspace segmentation.

In this paper, we partially address the above issues. Below, we briefly describe the main contributions of this paper and our related motivations.

1. The concept of minimal sample space is introduced and used to define a minimal sample segmentation (MSS) of a given set of samples. The existence of the MSS is guaranteed but may not be unique in some special cases; thus, we show that the MSS is conditionally unique. Two kinds of sufficient conditions for this uniqueness are given that focus on data quantity and quality, respectively. These conditions are weak since they are always satisfied in applications with randomly chosen samples from ground-truth subspaces. Hence, the minimal sample subspaces should generally be ground-truth subspaces.

2. It is difficult to check the minimality of a segmentation. We further study how to simplify detection under following the prior information of an MSS: The number of minimal segments, the sum of the segment ranks, and the minimal rank of the segments. We focus on the set of partitions with the same number of pieces and the restrictions of rank sum and minimal rank. Conditions for the singleness of such a set are given based on discreet rank estimations on each segment. These conditions permit subspaces to be heavily intersected within reasonable sense. Singleness means that the MSS can be detected.

3. The sufficient conditions for singleness of the above partition set are tight. We further exploit the properties of the sample segments when the sufficient conditions are incompletely satisfied, leading to two types of partition refinements under weaker conditions: Segment reduction and segment extension.

4. Based on solid theoretical analyses, we formulate the detection of minimal subspace segmentation as a computable optimization problem that adopts the self-expressiveness of samples. The closed-form structure of the representation matrix is given. MSS detection requires a connected and block-diagonal structure of the solution partitioned as the considered MSS. We prove that all the connected diagonal blocks are guaranteed only if the rank sum of the diagonal blocks is equal to that of the minimal sample segments. Under this restriction, the optimization problem gives a minimal subspace detectable representation (MSDR) of the MSS.

5. The objective function of the proposed optimization problem contains discrete variables from index partition J and continuous variables from the representation matrix C over a nonconvex feasible domain. To solve this minimization problem, we alternatively optimize J and C, slightly modifying J to an active index set Ω and adding a penalty on the diagonals of C. A manifold conjugate gradient (MCG) method is used for optimizing C, and an update rule is given for both Ω and J. Combining the two types of update rules yields an alternative optimization for detecting an MSS. An equivalent pseudo-dual problem of the primal problem is further considered and solved via subspace correction. These two kinds of MSS algorithms may drop into local minima, but they seldom have the same local minimizers. Hence, alternatively using these algorithms is an efficient strategy for escaping a local minimum, yielding a hybrid optimization method for the minimal subspace segmentation.

6. We further extend the MSS optimization problem to handle noisy samples. An ADMM method is simply considered for solving this extended optimization problem, and detailed formulas are given for solving the subproblems involved in the ADMM method.

It should be pointed out that we require the sum of subspace dimensions in our sparse model. It is an additional prior as a restriction to the rank of C, compared with algorithms given in the literature for subspace learning. The restriction is not necessary for uniquely determining the MSS (see Theorem 6 given in Section 2.2 for the sufficient conditions). However, it is necessary for guaranteeing the connection of a block-diagonal C (Theorems 22 and 24 in Section 5.2). It is also helpful for simplifying the detection of minimal subspace or MSS as shown in Theorem 11 of Section 3.1.

The remainder of the paper is organized as follows. Sections 2-6 cover the analysis of noiseless samples, while Section 7 discusses the extended model on noisy samples. The definition of minimal sample subspaces and discussions on uniqueness are given in Section 2. In Section 3, we discuss the problem of detecting an MSS, focusing on conditions for the singleness of rank-restricted index partitions. The theoretically supported refinement of conditioned partitions is further discussed in Section 4. Based on these theoretical analyses, we model the MSS problem as a computational optimization problem in Section 5, covering a closed form of representation matrices, the connectivity of diagonal blocks, slight modifications to the model, and a comparison with related work. The MSS algorithms, together with a manifold conjugate gradient method for solving the basic optimization problem, alternative optimization strategies and hybrid optimization, are given in Section 6. In addition, we present an extended model for handling noisy samples and a detailed ADMM algorithm for solving the optimization problem. Finally, we report our numerical results and compare our method with existing algorithms on both noiseless synthetic data and real-world data in Section 8. Comments on further research directions are given in the conclusion section.

2. Minimal Subspace Segmentation

The subspaces that we expect to identify from the set of a finite number of samples may be quite different from those that naturally fit these samples. This occurs when the samples from an expected subspace are exactly located in the expected subspaces' several smaller subspaces. Thus, the basic issue for subspace learning is: what subspaces can we reasonably expect based on the given data points?

In this section, we introduce the concept of a minimal sample subspace and use it to define a segmentation of samples termed minimal subspace segmentation. For the sake of discussion, we refer to X as a data matrix consisting of data points $\{x_i\}$ as its columns, which is also referred to as the set of the data points. In the following discussion, $r(X)$ and $n(X)$ refer to the rank and column number of X, respectively.

2.1. Minimal Sample Subspace

Naturally, the subspace spanned by a set of samples should have a smaller dimension than the sample number. Equivalently, the spanning samples are not linearly independent. We say that a sample-spanned subspace is minimal if it does not have a smaller subspace spanned by a subset of the samples. That is, any linearly dependent partial set of samples

spans the same subspace. Below is an equivalent definition of the minimal sample subspace in linear algebra.

Definition 1 *A sample subspace* $\mathrm{span}(X)$ *is minimal, if*

(1) X is rank deficient, that is, $n(X) > r(X)$, and

(2) X is nondegenerate, that is, any subset with a rank smaller than $r(X)$ is of full rank.

A sample subspace, $\mathrm{span}(X)$*, is pure if X is of full column rank, i.e., $n(X) = r(X)$.*

Nondegeneracy specifies that $r(X') = \min\{n(X'), r(X)\}$ for any subset X' of a nondegenerate X. Hence, any subset of a nondegenerate X must also be nondegenerate. This property implies that, for a minimal subspace $\mathrm{span}(X)$, any rank deficient subset X' of X cannot span a subspace with a smaller dimension. Equivalently, if $\mathrm{span}(X)$ contains a minimal subspace $\mathrm{span}(X')$, the two subspaces must be equal.

With respect to a given data set X if its spanning subspace $\mathrm{span}(X)$ is neither pure nor minimal, that is, if X is rank-deficient but degenerate, then there is a rank-deficient subset X' of smaller rank. Thus, it makes sense to partition the data set X into several nonoverlapping segments $X_0, \cdots, X_K$ such that $\mathrm{span}(X_0)$ is pure and the other $\{\mathrm{span}(X_k)\}$ are minimal.

Definition 2 *A segmentation $\{X_0, X_1, \cdots, X_K\}$ of vector set X is called a minimal subspace segmentation (MSS) of X if*

(1) $\mathrm{span}(X_0)$ *is pure, and each* $\mathrm{span}(X_k)$ *is minimal, $k = 1, \cdots, K$, if it exists;*

(2) $\mathrm{span}(X_k) \neq \mathrm{span}(X_\ell)$ *for $k, \ell = 0, \cdots, K$, $k \neq \ell$;*

(3) If X_0 exists, for any $x_j \in X_0$, $x_j \notin \mathrm{span}(X_k)$, $k = 1, \cdots, K$.

We also call X_0 a pure segment if $\mathrm{span}(X_0)$ is pure. In applications, a pure segment X_0, if it exists, could be a set of outliers. Condition (3) is necessary since some samples may be redundant for spanning a minimal sample space.

Theorem 3 *Any set of nonzero vectors has an MSS.*

Proof The basic idea of the proof was mentioned above. If $\mathrm{span}(X)$ is pure, we set $X = X_0$ and $K = 0$. If $\mathrm{span}(X)$ is minimal, we set $X_1 = X$ and $K = 1$, and X_0 disappears. In the other cases, we have a minimal subspace $\mathrm{span}(X_1)$ that has the smallest dimension, where X_1 is a subset of X. X_1 can be the set containing all the samples belonging to the subspace $\mathrm{span}(X_1)$, since adding these samples does not change the minimality of the subspace because there is no minimal subspace of lower dimension. That is, $\mathrm{span}(X_1)$ remains minimal after adding samples. Repeating the above procedure on the remaining samples, we can complete the proof. ■

However, the MSS of a given set X may be not unique. The following example illustrates an example of nonuniqueness.

Example 1 *Let X be the union of $4T$ five-dimensional vectors in the pieces*

$$X_{1,j} = \begin{bmatrix} a_{j,1} & a_{j,2} \\ b_j & b_j \\ o & o \end{bmatrix}, \quad X_{2,j} = \begin{bmatrix} a_{j,1} & a_{j,2} \\ o & o \\ b_j & b_j \end{bmatrix} \quad \text{with} \quad o = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad j = 1, \cdots, T,$$

where the scales $a_{j,1}$ and $a_{j,2}$ and the vector $b_j \in R^2$ are arbitrarily chosen such that $a_{j,1} \neq a_{j,2}$, each pair (b_i, b_j) pair is linearly independent, and $\begin{bmatrix} a_{i,s_i} & a_{j,s_j} & a_{k,s_k} \\ b_i & b_j & b_k \end{bmatrix}$ is of full rank for different i, j, k and any $s_i, s_j, s_k = 1, 2$. Then X has two types of segmentations without a pure segment,

(1) $X'_1 = [X_{1,1}, \cdots, X_{1,T}]$, $X'_2 = [X_{2,1}, \cdots, X_{2,T}]$, $K = 2$;

(2) $X''_k = [X_{1,k}, X_{2,k}]$, $k = 1, \cdots, K = T$.

Here, each $\text{span}(X'_k)$ or $\text{span}(X''_k)$ is minimal. Hence, both $\{X'_k\}$ and $\{X''_k\}$ are minimal.

This example partially explains why subspace learning is complicated. First, a sample set may have multiple segmentations, and each is an MSS. Second, the segments of an MSS may be very small. A small minimal segment X_k may have the smallest number of samples needed to span a minimal subspace, *i.e.*, $n(X_k) = r(X_k) + 1$. Obviously, if each segment is small in an MSS, this MSS may have a large number of segments. That is, the samples can be clustered into many small classes. Third, two different MSSs may have an equal number of segments with equal ranks. This case occurs when $T = 2$, where each segment has rank 3 with 4 samples.

Fortunately, the MSS is generally unique in applications. In the next subsection, we discuss the conditions of uniqueness.

2.2. Uniqueness of Minimal Subspace Segmentation

The following condition is obviously necessary for a unique MSS of $X = [x_1, \cdots, x_n]$.

$$x_j \notin \text{span}(X_k) \cap \text{span}(X_\ell), \quad \forall j \text{ and } \forall k \neq \ell. \tag{3}$$

Otherwise, a sample belonging to the intersection of two spanning subspaces could be arbitrarily assigned to any one of the two sample sets spanning the subspaces. In this subsection, we describe two types of sufficient conditions that guarantee the uniqueness of an MSS based on either sample quantity or quality.

Theorem 4 *If X has an MSS with K minimal sample subspaces satisfying the condition (3) and $n(X_k) > (K+1)(r(X_k) - 1)$, then a different MSS of X satisfying (3), if it exists, must have more minimal sample subspaces.*

Proof The theorem is obviously true if $\text{span}(X)$ is pure. If $\text{span}(X)$ is not pure, which implies $K \geq 1$, and there is another MSS $\{X'_0, X'_1, \cdots, X'_{K'}\}$ of X satisfying (3) with $K' \leq K$, let

$$n_1 = n(X_1), \quad r_1 = r(X_1), \quad X_{1,k} = X_1 \cap X'_k, \quad n_{1,k} = n(X_{1,k}), \quad k_1 = \arg\max_{k \leq K'} n_{1,k}.$$

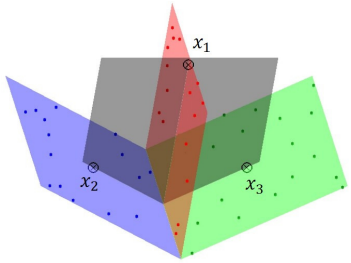


Figure 1: Illustration of nondegenerate intersection. The segmentation of samples according to the blue, red, and green planes is intersected degenerately because of the points x_1, x_2, x_3 that are sampled from the three planes but can span a new minimal subspace marked in gray. After deleting these three points, the segmentation is intersected nondegenerately.

Because $(K+1)(r_1 - 1) < n_1 = \sum_{k=0}^{K'} n_{1,k} \leq (K'+1)n_{1,k_1} \leq (K+1)n_{1,k_1}$, we have $n_{1,k_1} \geq r_1$. Hence, $\text{span}(X_1) = \text{span}(X_{1,k_1}) \subset \text{span}(X'_{k_1})$. By (3), $X'_{k_1} = X_1$. Deleting X_1 from X, the remainder samples have two minimal segmentations $\{X_0, X_2, \cdots, X_K\}$ and $\{X'_0, \cdots, X'_{k_1-1}, X'_{k_1+1}, \cdots, X'_{K'}\}$. Using induction on K, these two minimal segmentations should be equal. Hence, the theorem is proven. ■

Theorem 4 basically says that an MSS is unique if it is 'fat', that is, each segment has enough samples. Multiple minimal subspace segmentations may exist only if some segments have a small number of samples compared with the number of segments. Among those minimal segments with few samples, a union of partial samples from different segments can also form a new minimal segment. This may be the main reason for the multiplicity of minimal subspace segmentations. However, this multiplicity will disappear if the samples are well-distributed. Below, we introduce a new concept to define such a good distribution. For the sake of simplicity, $X \backslash X'$ indicates the remaining samples of X when the subset X' is removed.

Definition 5 *A segmentation $\{X_0, X_1, \cdots, X_K\}$ is intersected nondegenerately, if for any subset X'_k of $X \backslash X_k$, the splitting*

$$X_k = Y_k + Z_k, \quad Y_k \subset \mathcal{S}'_k, \quad Z_k \subset \mathcal{S}''_k$$

according to the direct sum $\text{span}(X_k) = \mathcal{S}'_k \oplus \mathcal{S}''_k$, always gives a zero or nondegenerate Z_k, where $\mathcal{S}'_k = \text{span}(X_k) \cap \text{span}(X'_k)$, and $\mathcal{S}''_k$ is the orthogonal complement of $\mathcal{S}'_k$ restricted in $\text{span}(X_k)$.

Figure 1 illustrates the nondegenerate intersection, in which the minimal segmentation becomes to be intersected degenerately if we add three special points x_1, x_2, and x_3 into

the three planes. In this case, the whole sample set has a new segmentation with 4 minimal segments; However the intersection of two segments may be not empty. In addition, after merging the new points into the original three segments, respectively, the three extended segments also form an MSS. This illustration is limited because of its low dimension, in which the newly added sample x_i belongs to an intersection of two minimal segments. This intersection phenomenon should be removed in uniqueness analysis as we assumed in (3). In a higher dimensional space, some MSSs may be intersected degenerately, although the intersection phenomenon in Figure 1 does not occur. For example, (3) is satisfied for the degenerately intersected segmentation $\{X_k''\}$ given in Example 1.

Theorem 6 *If X has a nondegenerately intersected MSS satisfies (3), then its MSS satisfying (3) is unique.*

Proof We use the same notation as that used in the proof of Theorem 4. Assume that X has a nondegenerately intersected MSS $\{X_0, \cdots, X_K\}$ and another MSS $\{X_0', \cdots, X_{K'}'\}$, and both satisfy (3). We first show that there is a segment X_{k_1}' equal to X_1. Uniqueness is then achieved after applying the method of induction to the number of segments since $\{X_0, X_2, \cdots, X_K\}$ is a minimal segmentation of the remaining samples. To this end, we consider $n_{1,k_1} = \max_{k\neq 0} n_{1,k}$ that is most possible for the equality $X_{k_1}' = X_1$ since X_{k_1}' has the largest intersection with X_1. For the sake of simplicity, we can assume that $k_1 = 1$. The equality $X_1' = X_1$ holds if $\text{span}(X_1') \subseteq \text{span}(X_1)$ or $\text{span}(X_1) \subseteq \text{span}(X_1')$ by (3).

Assume $\text{span}(X_1') \nsubseteq \text{span}(X_1)$ and $\text{span}(X_1) \nsubseteq \text{span}(X_1')$ conversely. That is, both $X_1^c = X_1' \setminus X_{1,1}$ and $X_1 \setminus X_{1,1}$ are not empty. Obviously, $X_1 \nsubseteq \mathcal{S}_0' = \text{span}(X_1^c)$. Hence, splitting X_1 according to the direct sum $\text{span}(X_1) = \mathcal{S}' + \mathcal{S}''$, where $\mathcal{S}' = \mathcal{S}_0' \cap \text{span}(X_1)$ and $\mathcal{S}''$ is the orthogonal complement of $\mathcal{S}'$ restricted in $\text{span}(X_1)$, we can rewrite $X_1 = Y_1 + Z_1$, where $Y_1 \subset \mathcal{S}'$ and nonzero $Z_1 \subset \mathcal{S}''$ that should be nondegenerate since $\{X_0, \cdots, X_K\}$ is intersected nondegenerately. Thus, in the splitting $X_{1,1} = Y_{1,1} + Z_{1,1}$ of the subset $X_{1,1}$ of X_1, where $Z_{1,1} \subset Z_1$, the nondegeneracy of Z_1 gives that $r(Z_{1,1}) = \min\{n_{1,1}, r(Z_1)\}$. However, whether $r(Z_{1,1}) = r(Z_1)$ is true or not, it always leads to a contradiction as shown below. If $r(Z_{1,1}) = r(Z_1)$, then $\text{span}(Z_1) = \text{span}(Z_{1,1}) \subset \text{span}(X_{1,1}) \subset \text{span}(X_1')$, and we get $\text{span}(X_1) \subseteq \text{span}(Y_1) + \text{span}(Z_1) = \mathcal{S}' + \text{span}(X_1') \subseteq \text{span}(X_1')$, a contradiction of the hypothesis $\text{span}(X_1) \nsubseteq \text{span}(X_1')$. If $r(Z_{1,1}) \neq r(Z_1)$, then $r(Z_{1,1}) = n_{1,1}$ and

$$r(Y_{1,1}) + r(Z_{1,1}) = r(X_{1,1}) \leq n_{1,1} = r(Z_{1,1}).$$

Hence, $Y_{1,1} = 0$, *i.e.*, $X_{1,1} = Z_{1,1} \subseteq S'' \perp \big(\mathcal{S}_0' \cap \text{span}(X_{1,1})\big)$. We conclude that $\text{span}(X_{1,1})$ is orthogonal to $\mathcal{S}_0' = \text{span}(X_1^c)$. Thus, $r(X_1^c) < r(X_1')$. By the minimality of X_1', X_1^c is of full column rank, and $n_1 > r(X_1') = r(X_{1,1}) + r(X_1^c) = n_{1,1} + n(X_1^c) = n_1$, which is also a contradiction. ■

The sample quantity condition of Theorem 4 is generally satisfied in many applications since the number of subspaces that we want to be recognized is quite small, compared with the number of samples. In addition, the sample quality condition of Theorem 6 is also satisfied with probability 1 if the samples are randomly chosen from given subspaces.

Theorem 7 *Given different subspaces $\mathcal{S}_1, \cdots, \mathcal{S}_K$, if the columns of X_k are randomly chosen from $\mathcal{S}_k$ with $n_k > \dim(\mathcal{S}_k)$ for $k = 1, \cdots, K$, then $\{X_1, \cdots, X_K\}$ is intersected nondegenerately and (3) is satisfied with probability 1.*

Proof The condition (3) is obviously satisfied with probability 1. Let U_k be an orthogonal basis matrix of $\mathcal{S}_k$, and let $X_k = U_k H_k$. By the assumption, H_k is a random matrix whose entries are i.i.d. To show the nondegenerate intersection of $\{X_0, X_1, \cdots, X_K\}$, we consider an arbitrary subset X_k' of $X_k^c = X \setminus X_k$, and the splitting

$$X_k = Y_k + Z_k, \quad Y_k \subset \mathcal{S}_k' = \mathcal{S}_k \cap \text{span}(X_k'), \quad Z_k \subset \mathcal{S}_k''$$

where $Z_k \neq 0$ and $\mathcal{S}_k''$ is the orthogonal complement of $\mathcal{S}_k'$ restricted to $\mathcal{S}_k$. Let $Y_k = U_k' H_k'$ and $Z_k = U_k'' H_k''$, where U_k' and U_k'' are orthogonal basis matrices of $\mathcal{S}_k'$ and $\mathcal{S}_k''$, respectively. We have $U_k = [U_k', U_k'']$ and $H_k^T = [H_k'^T, H_k''^T]$. Since H_k is a random matrix whose entries are i.i.d., so is H_k''. The entry distribution implies that H_k'' is nondegenerate with probability 1 since a matrix with i.i.d. entries is full rank with probability 1. Therefore, $Z_k = U_k'' H_k''$ is also nondegenerate with probability 1. Hence, the proof is completed. ■

Generally, the pure segment X_0 vanishes in applications. The following corollary further shows that if the samples are randomly chosen from the union of subspaces $\mathcal{S}_1, \cdots, \mathcal{S}_K$, then these subspaces are just the unique minimal sample subspaces of the samples with probability 1.

Corollary 8 *Assume that the columns of X_k are randomly sampled from subspace $\mathcal{S}_k$ and $n_k > \dim(\mathcal{S}_k)$ for $k \leq K$. Then, $\{X_1, \cdots, X_K\}$ is the unique minimal segmentation with probability 1.*

Proof By Theorem 7, $\{X_1, \cdots, X_K\}$ is intersected nondegenerately with probability 1. Hence, with this probability, each X_k is non-degenerate, and hence, $\text{span}(X_k) = \mathcal{S}_k$ is a minimal sample subspace, that is, $\{X_1, \cdots, X_K\}$ is minimal by definition. Since (3) is also satisfied with probability 1, by Theorem 6, this MSS is unique with probability 1. ■

In summary, the MSS of a given a set of finite samples always exists. It is possible to have multiple MSSs, but a fat MSS is unique, as shown in Theorem 4. Furthermore, if the samples are well-distributed, only one MSS exists, as shown in Theorem 6. In applications, samples from ground-truth subspaces are generally well-distributed or the MSS is fat. Therefore, the MSS is unique and generally represents the ground-truth. However, segmentation minimality is extremely difficult to confirm. In the next section, we show how to detect the minimality in a relatively simple way, which provides an insight for MSS detection. It is very helpful for modeling minimality as an optimization problem so that we can practically determine the minimal subspace segmentation via solving the optimization problem.

3. Detection of Minimal Subspace Segmentation

Clearly, it is impractical to inspect the minimality of a given segmentation $\{X_{J_k}\}$ by checking whether each segment is nondegenerate or not, where $X_{J_k} = X(:, J_k)$. Notice that we have

used the notation X_k for the minimal segment for the sake of simplicity, *i.e.*, $X_k = X_{J_k^*}$ with the index set J_k^* of X_k. Fortunately, this complicated task can be relatively simplified if we have a little prior information on the MSS.

The insight for the detection of MSS is that prior information on the MSS may narrow the set of segmentations, and thus enabling relatively easy detection. To this end, and also for the sake of simplicity, we assume that MSS $\{X_k\}$ does not have a pure segment X_0 and is intersected nondegenerately. Thus, MSS $\{X_k\}$ is unique according to Theorem 6. Obviously, there are at least three necessary conditions for segmentation $\{X_{J_k}\}$ to be the MSS:

(a) Its segment number equals the number of the minimal segments;

(b) The rank sum of its segments is not larger than the rank sum $d = \sum_k r(X_k)$;

(c) Each segment size is larger than the smallest rank $d_{\min} = \min_k r(X_k)$.

Here the rank sum d is equal to the dimension sum of the minimal subspaces. We use the prior information to narrow the feasible domain of the MSS to the subset of those satisfying the above three restrictions. Equivalently, we focus the index partitions in the following set:

$$\mathcal{J}(K,d,d_{\min}) = \big\{J = \{J_1,\cdots,J_K\} :\ \min|J_k| > d_{\min},\ r(X_{J_1})+\cdots+r(X_{J_K}) \le d\big\}. \quad (4)$$

Obviously, index partition $J^* = \{J_k^*\}$ of the MSS $\{X_k\}$ belongs to $\mathcal{J}(K,d,d_{\min})$. If $\mathcal{J}(K,d,d_{\min})$ contains only one partition, the detection of the MSS becomes to simply check whether a partition has only K pieces and if the two conditions

$$r(X_{J_1})+\cdots+r(X_{J_K}) \le d, \quad \min|J_k| > d_{\min}$$

are satisfied. Hence, the relevant question is: Could $\mathcal{J}(K,d,d_{\min})$ be a singleton?

We will give a positive answer to this question under weak conditions shown later. To this end, let $\mathcal{S}_k = \mathrm{span}(X_k)$, $d_k = \dim(\mathcal{S}_k) = r(X_k)$, $n_k = n(X_k)$ for $k = 1,\cdots,K$, and

$$d_{\min} = \min_k d_k, \quad d_0 = \max_k \dim\big(\mathcal{S}_k \cap \sum_{j\neq k}\mathcal{S}_j\big), \quad d_{\mathrm{int}} = \max_{k\neq j}\dim(\mathcal{S}_k\cap\mathcal{S}_j). \quad (5)$$

Hence, $d = \sum_k d_k$. We say that $J = \{J_k\}$ is a minimal partition if $\{X_{J_k}\}$ is an MSS of X. Example 1 shows that $\mathcal{J}(K,d,d_{\min})$ may have multiple minimal partitions in special cases. To guarantee a single minimal partition in $\mathcal{J}(K,d,d_{\min})$, certain conditions must be met. In the next subsection, we offer some sufficient conditions that guarantee the singleness of $\mathcal{J}(K,d,d_{\min})$. We may use the assumptions if necessary.

$$(i)\ x_j \notin \mathcal{S}_k\cap\mathcal{S}_\ell, \quad \forall j\le n, k\neq\ell;\ (ii)\ \{X_1,\cdots,X_K\} \text{ is intersected nondegenerately.} \quad (6)$$

These sufficient conditions are tight. We will give some counterexamples in which one of the sufficient conditions is not satisfied and further discuss how to refine $J \in \mathcal{J}(K,d,d_{\min})$ in these cases.

The number K and dimension sum d of minimal subspaces are generally known in applications. The smallest dimension $d_{\min}$ may also be known if the minimal subspaces have equal dimensions. In the computational model given later, we assume that K, d, and $d_{\min}$ are known. However, the minimal dimension restriction is relaxed in our subsequent algorithms.

3.1. Conditions of Singleness

Our analysis on the singleness of $\mathcal{J}(K,d,d_{\min})$ is based on a discreet estimation on the rank of each segment X_{J_k} for a given partition $J = \{J_k\} \in \mathcal{J}(K,d,d_{\min})$. The simple equality for matrix partition $A = [B,C]$

$$r(A) = r(B) + r(C) - \dim(\mathrm{span}(B)\cap\mathrm{span}(C)) \quad (7)$$

will be repeatedly used in the rank estimation. For the sake of simplicity, let $t_+ = \max\{0,t\}$ and

$$J_{ik} = \{j\in J_i : x_j \in X_k\}, \quad |J_{ik}| = n(X_{J_{ik}}),$$

and let $\mathcal{S}\setminus\mathcal{S}'$ be the orthogonal complement of $\mathcal{S}'$ restricted in $\mathcal{S}$ for subspace $\mathcal{S}'$ of $\mathcal{S}$.

Lemma 9 *Let $J = \{J_k\} \in \mathcal{J}(K,d,d_{\min})$. If $x_j \notin \mathcal{S}_k\cap\mathcal{S}_\ell$ for all j and $k\neq\ell$, then for any $s\neq t$ and J_i,*

$$r(X_{J_i}) \ge \min\{|J_{is}|,d_s\} + \big(\min\{|J_{it}|,d_t\} - d_{\mathrm{int}}\big)_+ + \sum_{k\neq s,t}\big(\min\{|J_{ik}|,d_k\} - d_0\big)_+. \quad (8)$$

Furthermore, if $\{X_k\}$ is intersected nondegenerately, then

(a) $r(X_{J_i}) \ge d_{\min}$ for those J_i having a single nonempty piece J_{is}.

(b) $r(X_{J_i}) > d_{\min}$ for those J_i having at least two nonempty pieces J_{is} and J_{it}.

Proof For the sake of simplicity, let $X_k' = X_{J_{ik}}$. We prove (8) with $(s,t) = (1,2)$ only since one can reorder $\{X_k'\}$ to have X_s' and X_t' as the first two segments in the general case. To this end, we merge the first k pieces to $M_k = [X_1',\cdots,X_k']$ and let $\mathcal{S}_k' = \mathrm{span}(M_{k-1})\cap\mathrm{span}(X_k')$. By (7), we have the following recursion:

$$r(M_k) = r(M_{k-1}) + \zeta_k, \quad (9)$$

where $\zeta_k = r(X_k') - \dim(\mathcal{S}_k')$. Since X_k is nondegenerate, $r(X_k') = \min\{|J_{ik}|,d_k\}$. Combining this with $\dim(\mathcal{S}_k') \le d_{\mathrm{int}}$ for $k=2$ or $\dim(\mathcal{S}_k') \le d_0$ for $k>2$, we have the following estimate:

$$\zeta_k \ge \begin{cases} \big(\min\{|J_{ik}|,d_k\} - d_{\mathrm{int}}\big)_+, & k=2;\\ \big(\min\{|J_{ik}|,d_k\} - d_0\big)_+, & k>2.\end{cases} \quad (10)$$

Thus, taking the sum of all the equalities in (9) and using (10), we get (8) with $(s,t) = (1,2)$.

We further show that ζ_k can be represented with $\mathcal{S}_k'' = \mathcal{S}_k\cap\mathrm{span}(M_{k-1})$ as

$$\zeta_k = \min\{|J_{ik}|, d_k - \dim(\mathcal{S}_k'')\}. \quad (11)$$

based on the nondegeneracy of the intersection of $\{X_k\}$. To this end, we split $X_k' = Y_k' + Z_k'$ with $Y_k' \subset \mathcal{S}_k'$ and $Z_k' \subset \mathrm{span}(X_k')\setminus\mathcal{S}_k'$, and rewrite $M_k = [M_{k-1},Y_k'] + [0,Z_k']$. By (7), and $\mathrm{span}([M_{k-1},Y_k'])\cap\mathrm{span}(Z_k') = \{0\}$, we also obtain (9) with $\zeta_k = r(Z_k')$ since

$$r(M_k) = r([M_{k-1},Y_k']) + r(Z_k') = r(M_{k-1}) + r(Z_k').$$

To estimate the rank of Z'_k, we extend the splitting of X'_k to $X_k = Y_k + Z_k$ with $Y_k \subset \mathcal{S}''_k = \mathcal{S}_k \cap \text{span}(M_{k-1})$ and $Z_k \subset \mathcal{S}_k \backslash \mathcal{S}''_k$. Obviously, $r(Y_k) \le \dim(\mathcal{S}''_k)$ and $r(Z_k) \le d_k - \dim(\mathcal{S}''_k)$. These equalities should hold since $r(Y_k) + r(Z_k) = d_k$. Furthermore, Z_k is nondegenerated or $Z_k = 0$ by the nondegenerate intersection of $\{X_1, \cdots, X_K\}$. Thus, as a column submatrix of Z_k, Z'_k should have the rank $\min\{|J_{ik}|, r(Z_k)\} = \min\{|J_{ik}|, d_k - \dim(\mathcal{S}''_k)\}$. This is (11).

We now prove (a) and (b) of this lemma, using (9) and $r(X_{J_i}) = r(M_K) = \sum_{\ell=1}^K \zeta_k$, comparing $|J_{ik}|$ and $\delta_k = d_k - \dim(\mathcal{S}''_k)$ for determining ζ_k by its definition (11).

(1) If $|J_{ik}| \le \delta_k$ for all k, then $\zeta_k = |J_{ik}|$. Hence, $r(X_{J_i}) = \sum_{k=1}^K |J_{ik}| = |J_i| > d_{\min}$ since $J = \{J_k\} \in \mathcal{J}(K, d, d_{\min})$.

(2) If $|J_{i1}| > \delta_1$, and $|J_{ik}| \le \delta_k$ for $k \ge 2$, then $\zeta_1 = \delta_1 = d_1$ and $\zeta_k = |J_{ik}|$ for $k \ge 2$. Hence, $r(X_{J_i}) \ge d_1 + \sum_{k=2}^K |J_{ik}|$.

(3) If $|J_{ik}| > \delta_k$ for a $k \ge 2$, $\zeta_k = \delta_k = d_k - \dim(\mathcal{S}''_k)$. Since $\text{span}(M_{k-1}) \nsubseteq \mathcal{S}_k$, we have $\mathcal{S}''_k \subsetneq \text{span}(M_{k-1})$, *i.e.*, $r(M_{k-1}) > \dim(\mathcal{S}''_k)$. By (9), $r(X_{J_i}) \ge r(M_k) = r(M_{k-1}) + \zeta_k > d_k$.

Hence, in each of the above cases, (a) and (b) are always true. ■

The following lemma further shows that if each minimal segment X_k has a sufficient number of samples, it must be dominated by one piece of any $J \in \mathcal{J}(K, d, d_{\min})$, in the sense that there exists at least one subset J_{ik} whose size is not smaller than d_k. We will use this lemma to prove the singleness of $\mathcal{J}(K, d, d_{\min})$.

Lemma 10 *If $x_j \notin \mathcal{S}_k \cap \mathcal{S}_\ell$ for all j and $k \ne \ell$, and $n_k > d_k + (K-1)d_0$ for all k, then for $J \in \mathcal{J}(K, d, d_{\min})$*

$$\max_i |J_{ik}| \ge d_k, \quad k = 1, \cdots, K.$$

Proof Let $\mathcal{K} = \{k : \max_i |J_{ik}| \ge d_k\}$. This lemma is equivalent to saying that $\mathcal{K} = \{1, \cdots, K\}$. We can prove this by letting $\mathcal{I} = \{i : \max_k |J_{ik}| \ge d_k\}$ and $\mathcal{K}_i = \{k : |J_{ik}| \ge d_k\}$ for each $i \in \mathcal{I}$. Then, $\mathcal{K} = \cup_{i \in \mathcal{I}} \mathcal{K}_i$. If $\mathcal{K}^c$ is not empty, we choose an $s \in \mathcal{K}^c$ and any $t \ne s$ in (8) of Lemma 9 and use $d_{\text{int}} \le d_0$ to obtain the following:

$$r(X_{J_i}) \ge \begin{cases} d_0 + \sum_{k \in \mathcal{K}^c}(|J_{ik}| - d_0) + \sum_{k \in \mathcal{K}_i}(d_k - d_0)_+, & i \in \mathcal{I}; \\ d_0 + \sum_{k \in \mathcal{K}^c}(|J_{ik}| - d_0), & i \in \mathcal{I}^c. \end{cases}$$

Hence, $d = \sum_i r(X_{J_i}) \ge Kd_0 + \sum_{k \in \mathcal{K}^c} \sum_i (|J_{ik}| - d_0) + \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} (d_k - d_0)_+$. In the second term, $\sum_i (|J_{ik}| - d_0) \ge n_k - Kd_0 > d_k - d_0$. Since $\mathcal{I} = \cup_{k \in \mathcal{K}} \mathcal{I}_k$ with $\mathcal{I}_k = \{i : |J_{ik}| \ge d_k\}$, the last term becomes as follows:

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} (d_k - d_0)_+ = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}_k} (d_k - d_0)_+ = \sum_{k \in \mathcal{K}} |\mathcal{I}_k|(d_k - d_0)_+ \ge \sum_{k \in \mathcal{K}} (d_k - d_0). \tag{12}$$

Thus, $d > Kd_0 + \sum_k (d_k - d_0) = d$, which is a contradiction. Therefore, $\mathcal{K}^c$ must be empty. ■

We are now ready to prove the singleness of $\mathcal{J}(K, d, d_{\min})$.

Theorem 11 *Assume that X has an MSS $\{X_1, \cdots, X_K\}$ satisfying the assumption (6). If*

$$d_{\text{int}} < d_{\min}, \quad d_0 \le d_{\min}, \quad n_k > d_k + (K-1)d_0, \quad \forall k, \tag{13}$$

then $\mathcal{J}(K, d, d_{\min})$ is a singleton with the unique J^*.

Proof Let J^*_k be the index set of X_k. By Lemma 10, $\mathcal{I}_k = \{i : |J_{ik}| \ge d_k\}$ is nonempty. We further show that $\mathcal{I}_k$ has only a single index i_k for each k and $|\mathcal{I}| = K$. If it is proven, the mapping from k to i_k is one-to-one; hence for $i = i_k$, $r(X_{J_i}) \ge r(X_{J_{ik}}) = \min\{|J_{ik}|, d_k\} = d_k$. This equality holds since $d \ge \sum_k r(X_{J_{ik}}) \ge \sum_k d_k = d$. Thus, $\text{span}(X_{J_{i_k}}) = \text{span}(X_{J_{i_k,k}}) = \mathcal{S}_k$ and $J_{i_k} = J_{i_k,k} = J^*_k$. That is, $\{X_{J_{i_k}}\}$ is equal to $\{X_k\}$, so $\mathcal{J}(K, d, d_{\min})$ is a singleton with the unique J^*.

We now prove that $|\mathcal{I}_k| = 1$ for each k and $|\mathcal{I}| = K$ by Lemma 9. For $i \in \mathcal{I}^c$, we have $r(X_{J_i}) \ge d_{\min} + 1$ by (b) of Lemma 9, and then $\sum_{i \in \mathcal{I}^c} r(X_{J_i}) \ge |\mathcal{I}^c|(d_{\min} + 1)$. For $i \in \mathcal{I}$, we choose (s, t) in (8) such that $s \ne t \in \mathcal{K}_i$ if $|\mathcal{K}_i| > 1$, or $s \in \mathcal{K}_i$ and $t \in \mathcal{K}_i^c$ if $|\mathcal{K}_i| = 1$. We use the indication function $\delta_{|\mathcal{K}_i|>1} = 1$ if $|\mathcal{K}_i| > 1$ or $\delta_{|\mathcal{K}_i|>1} = 0$, otherwise , and obtain that for $i \in \mathcal{I}$,

$$\begin{aligned} r(X_{J_i}) &\ge d_s + \delta_{|\mathcal{K}_i|>1}(d_t - d_{\text{int}}) + \sum_{k \in \mathcal{K}_i, k \ne s,t} (d_k - d_0)_+ \\ &= d_0 + \delta_{|\mathcal{K}_i|>1}(d_0 - d_{\text{int}}) + \sum_{k \in \mathcal{K}_i} (d_k - d_0), \end{aligned} \tag{14}$$

and $\sum_{i \in \mathcal{I}} r(X_{J_i}) \ge |\mathcal{I}|d_0 + \sum_{i \in \mathcal{I}} \delta_{|\mathcal{K}_i|>1}(d_0 - d_{\text{int}}) + \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} (d_k - d_0)$. Hence,

$$d \ge \sum_i r(X_{J_i}) \ge |\mathcal{I}|d_0 + \sum_{i \in \mathcal{I}} \delta_{|\mathcal{K}_i|>1}(d_0 - d_{\text{int}}) + \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} (d_k - d_0) + |\mathcal{I}^c|(d_{\min} + 1). \tag{15}$$

Since $|\mathcal{I}_k| \ge 1$, $|\mathcal{K}_i| \ge 1 + \delta_{|\mathcal{K}_i|>1}$ for $i \in \mathcal{I}$, and $d_{\min} \ge d_0$, we estimate the third term as

$$\begin{aligned} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} (d_k - d_0) &= \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} (d_k - d_{\min}) + \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} (d_{\min} - d_0) \\ &= \sum_k |\mathcal{I}_k|(d_k - d_{\min}) + \sum_{i \in \mathcal{I}} |\mathcal{K}_i|(d_{\min} - d_0) \\ &\ge d - Kd_{\min} + |\mathcal{I}|(d_{\min} - d_0) + \sum_{i \in \mathcal{I}} \delta_{|\mathcal{K}_i|>1}(d_{\min} - d_0). \end{aligned} \tag{16}$$

Substituting (16) into (15), we obtain that $d \ge d + |\mathcal{I}^c| + \sum_{i \in \mathcal{I}} \delta_{|\mathcal{K}_i|>1}(d_{\min} - d_{\text{int}}) \ge d$. Hence, $|\mathcal{I}^c| = 0$. Furthermore, we have $|\mathcal{K}_i| = 1$ for each $i \in \mathcal{I}$ if $d_{\min} > d_{\text{int}}$. Since $|\mathcal{K}| = |\mathcal{I}| = K$, $|\mathcal{K}_i| = 1$ for each $i \in \mathcal{I}$ is equivalent to $|\mathcal{I}_k| = 1$ for each k. The theorem is then proven. ■

The condition $n_k > d_k + (K-1)d_0$ for all k is generally satisfied in applications. The other conditions on d_{int} and d_0 are also satisfied (in some cases, naturally) basically because $\mathcal{S}_k \nsubseteq \mathcal{S}_j$ for any $k \ne j$. In practice, since $\mathcal{S}_k \cap \mathcal{S}_j \subsetneq \mathcal{S}_k$ for $k \ne j$, $d_{\text{int}} < \max_{k \ne j} \min\{d_k, d_j\} \le \max_k d_k$ and $d_0 \le \max_k d_k$. Thus, if all the d_k's are equal, $d_{\text{int}} < \max_k d_k = d_{\min}$ and $d_0 \le d_{\min}$. The equality restriction on $\{d_k\}$ can be released if $K = 2$ since $d_0 = d_{\text{int}} < \min\{d_1, d_2\} = d_{\min}$. We summarize our conclusions as a corollary.

Corollary 12 *Assume that X has an MSS $\{X_1, \cdots, X_K\}$ satisfying the assumption (6). If $n_k > d_k + (K-1)d_0$ for each k, and $d_1 = \cdots = d_K$ when $K > 2$ or arbitrary d_1 and d_2 when $K = 2$, then J^* is the single partition in $\mathcal{J}(K, d, d_{\min})$.*

3.2. Necessity of the Sufficient Conditions

The conditions of Theorem 11 are tight. In this subsection, we give three counterexamples to show that if one of these conditions, except $n_k > d_k + (K-1)d_0$, is not satisfied, $\mathcal{J}(K, d, d_{\min})$ may not be a singleton. In detail, the MSS $\{X_k\}$ in Example 2 is not interacted nondegenerately, and the other conditions in (13) are satisfied. Example 3 is designed such that $d_{\text{int}} < d_{\min}$ is not obeyed, and in Example 4, there exists a $d_k < d_0$.

Example 2 *Let $X_k = U_k H \in \mathcal{R}^{8\times n_k}$ with $n_k > 30$, $k = 1, \cdots, 6$, where H is nondegenerate and its first three columns are $e, e-e_1, e-e_1-e_2$, e_i is the i-th column of the identity matrix of order 8, and e is a the column vector of all ones. Each U_k consists of 5 columns of the same identity matrix,*

$$U_1 = [e_1, e_2, e_3, e_4, e_5],\ U_2 = [e_1, e_2, e_6, e_7, e_8],\ U_3 = [e_3, e_4, e_1, e_2, e_6],$$
$$U_4 = [e_3, e_4, e_5, e_7, e_8],\ U_5 = [e_5, e_6, e_1, e_2, e_4],\ U_6 = [e_5, e_6, e_4, e_7, e_8].$$

Obviously, $\{X_k\}$ is an MSS of $X = \cup_k X_k$ with $K = 6$ and $d = 30$ since $r(X_k) = 5$ for all $k \le K$. The inequality conditions in (13) are satisfied since $d_0 = 5$, $d_{\text{int}} = 4$, $d_k = 5$, and $n_k > 30 = d_k + (K-1)d_0$ for all k. However, for $\mathcal{S}_1' = \mathcal{S}_1 \cap \mathcal{S}_2$, the splitting $X_1 = Y_1 + Z_1$ with $Y_1 \subset S_1'$ and $Z_1 \subset \mathcal{S}_1 \setminus \mathcal{S}_1'$ results in a degenerate Z_1 whose first three columns are equal to $e_3 + e_4 + e_5$. Hence, $\{X_k\}$ is not interacted nondegenerately. In addition to MSS $\{X_k\}$, we have another segmentation of 6 pieces as

$$\tilde{X}_1 = [X_1(:, 1:3), X_2(:, 1:3)],\ \tilde{X}_2 = [X_3(:, 1:3), X_4(:, 1:3)],$$
$$\tilde{X}_3 = [X_5(:, 1:3), X_6(:, 1:3)],\ \tilde{X}_4 = X_1(:, 4:n_1),\ \tilde{X}_5 = X_2(:, 4:n_2),$$
$$\tilde{X}_6 = [X_3(:, 4:n_3), X_4(:, 4:n_4), X_5(:, 4:n_5), X_6(:, 4:n_6)].$$

Since $r(\tilde{X}_k) = 4$ for $k \le 3$, $r(\tilde{X}_4) = r(\tilde{X}_5) = 5$, $r(\tilde{X}_6) = 8$, we also have $\sum_k r(\tilde{X}_k) = 30$. Hence, the partition $\tilde{J}$ corresponding to $\{\tilde{X}_k'\}$ also belongs to $\mathcal{J}(K, d, d_{\min})$. However, $\{\tilde{X}_k\}$ cannot be minimal since both $\{X_k\}$ and $\{\tilde{X}_k\}$ satisfy $n_k > 7(d_k - 1)$ for each k, and by Theorem 4, the MSS of X with $K = 6$ is unique.

Example 3 *Let $X = [X_1, X_2, X_3]$ with $X_k = U_k H_k$, where $U_1 = [e_1, e_2]$, $U_2 = [e_2, e_3]$, and $U_3 = e_4$ are three orthonormal matrices of four rows, and H_1, H_2, and H_3 are three non-degenerate matrices of 5 columns with 2, 2, and 1 row(s), respectively.*

This segmentation is minimal by definition but does not satisfy $d_{\text{int}} < d_{\min}$. We have a different one $\{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3\}$ where $\tilde{X}_1 = [X_1, X_2]$ and the other two pieces $\tilde{X}_2$ and $\tilde{X}_3$ split from X_3, each having at least two samples $\tilde{X}_2$ and $\tilde{X}_3$ Since $\sum_k r(X_k) = \sum_k r(\tilde{X}_k) = 5$, both partitions belong to $\mathcal{J}(3, 5, 1)$.

Example 4 *Let $X = [X_1, \cdots, X_5]$ of columns in $\mathcal{R}^8$ and $X_k = U_k H_k$ with orthonormal*

$$U_1 = [e_1, e_2, e_3],\ U_2 = [e_3, e_4, e_5],\ U_3 = [e_1, e_4, e_6],\ U_4 = [e_2, e_5, e_6],\ U_5 = [e_7, e_8],$$

and let $\{H_k\}$ be intersected nondegenerately with $n(X_k) = n(H_k) > 15$.

The segmentation $\{X_k\}$ is also minimal with $d = 15$ since each X_k is nondegenerate as H_k. Now, the condition $d_0 \le d_{\min}$ is not satisfied since $d_0 = 3$ and $2 = d_{\min}$. If we merge the first 4 segments to be $\tilde{X}_1$ and split X_5 into 4 pieces as $\tilde{X}_2, \cdots, \tilde{X}_5$ without overlap, and each $\tilde{X}_k$ has at least three samples, then $\sum_k r(\tilde{X}_k) = 15$. Hence, $\mathcal{J}(5, 15, 2)$ has at least two different partitions.

4. Segmentation Refinement

When either of two conditions $d_{\text{int}} < d_{\min}$ or $d_0 \le d_{\min}$ in Theorem 11 are not satisfied, $\mathcal{J}(K, d, d_{\min})$ may have multiple K-partitions. Hence, there may be a partition $J = \{J_k\}$ in $\mathcal{J}(K, d, d_{\min})$ that is not minimal. However, certain segments J_k or X_{J_k} can be further refined to be minimal under some weak conditions. Let us illustrate this scenario on the examples shown in the last subsection.

In Example 3, we take segment of $\{\tilde{X}_k\}$ with the smallest rank, say $\tilde{X}_2$, and extend it to be the largest segment containing all the samples belonging to $\text{span}(\tilde{X}_2)$. This extension merges $\tilde{X}_2$ and $\tilde{X}_3$ as X_3; hence, X_3 is recovered. Then, $\{X_1, X_2\}$ is an MSS of the remaining samples $X' = X \setminus X_3$. One may search for a segmentation from $\mathcal{J}(K', d', d'_{\min})$ on X' with $K' = K - 1 = 2$, $d' = d - r(X_3) = 4$ and $d'_{\min} = 2$. Since the conditions of Theorem 11 are now satisfied , $\mathcal{J}(K', d', d'_{\min})$ has the single segmentation $\{X_1, X_2\}$. Hence, minimal segmentation $\{X_k\}$ is recovered. Similar, we can refine $\{\tilde{X}_k\}$ in Example 4.

In Example 2, each $\tilde{X}_k$, $k \le 3$, has the smallest rank but is nonextendable. However, the extension works on the larger segments $\tilde{X}_4$ or $\tilde{X}_5$. That is, if we extend $\tilde{X}_4$ to the largest one, X_1 can be recovered immediately. Similarly, when $\tilde{X}_5$ is extended, X_2 can also be recovered. Other segments can be determined from $\mathcal{J}(K', d', d'_{\min})$ on the remaining samples with $K' = K - 2$, $d' = d - r(X_1) - r(X_2)$ and $d'_{\min} = 5$.

Motivated by these observations, we offer an approach for refining a segmentation $\{X_{J_k}\}$ for $J \in \mathcal{J}(K, d, d_{\min})$ if it is not minimal. The approach consists of two strategies: reduction and extension. We emphasize that, in this section our analysis is given under the same assumption as that given in the last subsection. Hence, we no longer mention the conditions for simplicity.

4.1. Segment Reduction

We observe that a partition $J = \{J_k\} \in \mathcal{J}(K, d, d_{\min})$ has at least one piece J_k such that X_{J_k} is a minimal segment, even if the whole segmentation $\{X_{J_i}\}$ is not an MSS of X. The following two propositions support this observation.

Proposition 13 *If $r(X_{J_i}) = d_{\min} < |J_i|$, then $J_i \subseteq J_k^*$ with $d_k = d_{\min}$.*

Proof By (b) of Lemma 9, if J_i has two nonempty intersection parts $J_{ik} = J_i \cap J_k^*$ with two different k's, then the condition $|J_i| > d_{\min}$ implies $r(X_{J_i}) > d_{\min}$. Therefore, if we also have $r(X_{J_i}) = d_{\min}$, J_i must have a single nonempty J_{ik}, that is, $J_i \subseteq J_k^*$, and $X_{J_i} \subseteq X_k$. Since X_k is nondegenerate,

$$d_{\min} = r(X_{J_i}) = \min\{|J_i|, d_k\} \ge \min\{d_{\min} + 1, d_k\}.$$

Combining this with $d_k \ge d_{\min}$, we can conclude that $d_k = d_{\min}$. ■

Proposition 14 *If $d_0 \le d_{\min}$ and $|J_k| > d_k + (K-1)d_0$ for all k, then $\min_i r(X_{J_i}) = d_{\min}$.*

Proof If we further have that $d_{\min} > d_{\text{int}}$, this proposition is obviously true since $\{J_k\}$ is unique by Theorem 11. Hence, we can assume $d_{\min} \le d_{\text{int}}$, which implies $d_{\text{int}} = d_0 = d_{\min}$ since $d_{\text{int}} \le d_0 \le d_{\min}$. Thus, in the proof of Theorem 11, we have $|\mathcal{I}^c| = 0$ and each of the inequalities between (14) and (16) holds in equality, where we do not use $d_{\text{int}} < d_{\min}$. The equalities $d_{\text{int}} = d_0 = d_{\min}$ simplify (16) to $\sum_k |\mathcal{I}_k|(d_k - d_{\min}) = d - Kd_{\min}$. Thus, $|\mathcal{I}_k| = 1$ if $d_k > d_{\min}$. Consider the union of those $\mathcal{I}_k$ with $k \in \mathcal{K}' = \{k : d_k > d_{\min}\}$. The size of this union is equal to $|\mathcal{K}'|$ and $|\mathcal{K}'| < K$ since there is a $k \notin \mathcal{K}'$ with $d_k = d_{\min}$. For each $k \in \mathcal{K}_{i_0}$ with $i_0 \notin \cup_{k\in\mathcal{K}'}\mathcal{I}_k$, we also have $i_0 \in \mathcal{I}_k$. Hence, we conclude that for all $k \in \mathcal{K}_{i_0}$, $k \notin \mathcal{K}'$, *i.e.*, $d_k = d_{\min}$. Moreover, (14) becomes $r(X_{J_i}) = d_{\min} + \sum_{k\in\mathcal{K}_i}(d_k - d_{\min})$ for all i. Hence, $r(X_{J_{i_0}}) = d_{\min} \le r(X_{J_i})$ for all i. ■

Therefore, if the conditions of Proposition 14 are satisfied, by Proposition 13, $J_i \subseteq J^*_{k_i}$ for those J_i and $J^*_{k_i}$ with the smallest rank $r(X_{J_i}) = d_{\min} = d_{k_i}$. That is, these minimal segments X_{k_i} have been retrieved. Let K_0 be the number of retrieved segments. The conditions of Proposition 14 remain satisfied for $J' \in \mathcal{J}(K', d', d'_{\min})$, where $K' = K - K_0$, $d' = d - d_{\min}K_0$ and $d'_{\min} \ge d_{\min}$. Thus, repeating this reduction procedure, we can retrieve all the minimal segments. Therefore, we have proven the following theorem.

Theorem 15 *If $d_0 \le d_{\min}$ and $n_k > d_k + (K-1)d_0$ for all k, then the MSS $\{X_k\}$ of X can be recovered via a reduction procedure on any $J \in \mathcal{J}(K, d, d_{\min})$.*

4.2. Segment Extension

We say that J_i or X_{J_i} is extendable if there is at least one $x_j \in \text{span}(X_{J_i})$ with $j \notin J_i$. The extension strategy enlarges an extendable J_i as much as possible by adding all these j's into J_i similar to that given in the proof of Theorem 3 without checking for segment nondegeneracy.

Proposition 16 *If each J_i of $J \in \mathcal{J}(K, d, d_{\min})$ is nonextendable, then $\{J_i\} = \{J^*_k\}$.*

Proof By Lemma 10, for each k, there exists an i_k such that $|J_{i_k,k}| \ge d_k$ and hence, $\text{span}(X_{J_{i_k,k}}) = \mathcal{S}_k$. Since J_{i_k} cannot be extended, we must have $J_{i_k,k} = J^*_k$ and J_{ik} is empty for $i \ne i_k$. Because each J_i is not empty, the mapping from k to i_k is one-to-one and onto. Therefore, $\{J_i\} = \{J^*_k\}$. ■

Proposition 17 *If J_i is extendable, the extended $\tilde{J}_i$ has $\tilde{J}_{ik} = J_{ik}$ or $\tilde{J}_{ik} = J^*_k$ for all k.*

Proof Assume $\tilde{J}_i = J_i \cup J'_i$, where $X_{J'_i} \subset \text{span}(X_{J_i})$. Then, $\tilde{J}_{ik} = J_{ik} \cup J'_{ik}$, where $J'_{ik} = J'_i \cap J^*_k$. If the minimal segment $X_k \subseteq \text{span}(X_{J^c_{ik}}) \subset \text{span}(X_{J_i})$, where $J^c_{ik} = J_i \setminus J_{ik}$, then we must have $\tilde{J}_{ik} = J^*_k$.

We will show that if $X_k \not\subseteq \text{span}(X_{J^c_{ik}})$ and $\tilde{J}_{ik} \ne J_{ik}$, then $r([X_k, X_{J^c_{ik}}]) = r(X_{J_i})$, which implies $X_k \subset \text{span}(X_{J_i})$, and hence, $\tilde{J}_{ik} = J^*_k$. To this end, we split $X_k = Y_k + Z_k$ with

$Y_k \subset \mathcal{S}'_k$ and $Z_k \subset \mathcal{S}''_k$, according to the direct sum $\mathcal{S}_k = \mathcal{S}'_k \oplus \mathcal{S}''_k$ where $\mathcal{S}'_k = \mathcal{S}_k \cap \text{span}(X_{J^c_{ik}})$ and $\mathcal{S}''_k$ is the orthogonal complement of $\mathcal{S}''_k$ restricted in $\mathcal{S}_k$. Obviously, we have $Z_k \ne 0$ when $\tilde{J}_{ik} \ne J_{ik}$, and hence, Z_k is nondegenerate since the MSS $\{X_k\}$ is intersected nondegenerately. Therefore, $r([X_k, X_{J^c_{ik}}]) = r([Y_k, X_{J^c_{ik}}]) + r(Z_k) = r([Z_k, X_{J^c_{ik}}])$. Similarly,

$$r(X_{J_i}) = r([X_{J_{ik}}, X_{J^c_{ik}}]) = r([Z'_k, X_{J^c_{ik}}]), \quad r(X_{\tilde{J}_i}) = r([X_{\tilde{J}_{ik}}, X_{J^c_{ik}}]) = r([\tilde{Z}'_k, X_{J^c_{ik}}]),$$

where Z'_k and $\tilde{Z}'_k$ are two subsets of Z_k corresponding to the subsets $X_{J_{ik}}$ and $X_{\tilde{J}_{ik}}$ of X_k, respectively. We have $r(\tilde{Z}'_k) = r(Z'_k)$ since $r(X_{\tilde{J}_i}) = r(X_{J_i})$. By the nondegeneracy of Z_k, $r(Z'_k) = r(Z_k)$. Therefore, $r([X_k, X_{J^c_{ik}}]) = r(X_{J_i})$, and the proof is completed. ■

Theorem 18 *Assume that $n_k > d_k + (K-1)d_0$ for all k. After segment extension on all the extendable segments of J one-by-one, then each nonempty segment $\tilde{J}_i$ of the resulting segmentation $\tilde{J}$ must be a union of several segments of the MSS $\{X_k\}$. Furthermore, $\tilde{J} = J^*$ if all the $\tilde{J}_i$'s are nonempty.*

Proof Consider the changes of J_{ik} during the extension process. In the step involving an extendable J_i, J_{ik} is unchanged or changed to J^*_k by Proposition 17. After this extension step, for other $j \ne i$, J_{jk} is unchanged or changed to the empty set. That is, after an extension step, J_{ik} is unchanged or becomes to J^*_k or the empty set. By Lemma 10, for each k, there is an i_k such that $|J_{i_k,k}| \ge d_k$ for the original J_{i_k}. Hence, $J_{i_k,k}$ becomes J^*_k in the extension of J_{i_k} if it is unchanged in the earlier extension steps. Otherwise, $J_{i_k,k}$ has already been changed to the empty set. Therefore, after all the extension steps, for each k, the eventually modified $\tilde{J}_{ik}$ must be empty or J^*_k. That is, each $\tilde{J}_i$ must be a union of some J^*_k or the empty set. ■

Note that the extension results may depend on the extending order of each J_i. We suggest extending in the ascending order of $\{r(X_{J_i})\}$ because if X_{J_i} has a smaller rank, most of its pieces $\{J_{ik}\}$ are likely to be empty or small. These small pieces will be removed, leaving the largest pieces after extension. This strategy reduces the risk of merging multiple minimal segments (X_k's) together into a single segment of $\tilde{J}$.

The extension procedure cannot increase the rank sum of the segments, and the rank sum is decreased if a $\tilde{J}_i$ is empty. In this case, $\tilde{J}$ has a smaller number of nonempty segments. Hence, $\tilde{J} \notin \mathcal{J}(K, d, d_{\min})$. Assume that $\tilde{J}$ has ℓ nonempty segments, say $\tilde{J}_1, \cdots, \tilde{J}_\ell$, and let $\mathcal{K}_t$ be the index set of those X_k that are merged to $X_{\tilde{J}_t}$ because of the extension. MSS $\{X_k\}$ is then partitioned into ℓ smaller groups $X^{(t)}$, $t = 1, \cdots, \ell$, each of which is a union of the minimal segments $\{X_k : k \in \mathcal{K}_t\}$. Therefore, one may further determine the MSS of $X^{(t)}$ via determining $\mathcal{J}(K^{(t)}, d^{(t)}, d^{(t)}_{\min})$ on X^t, where $K^{(t)} = |\mathcal{K}_t|$, $d^{(t)} = \sum_{k\in\mathcal{K}_t} r(X_k)$, and $d^{(t)}_{\min} = \min_{k\in\mathcal{K}_t} r(X_k)$. This amounts to a divide-and-conquer approach. We do not touch upon this technique further in this paper.

5. Computable Modeling for Minimal Subspace Segmentation

Our study of MSS detection simplifies its inspection. However, because detecting the MSS works on K-partitions of indices, it is difficult to implement efficiently. Thus, computable modeling is needed. To this end, we adopt the commonly used self-expressiveness approach.

The self-expressiveness method looks for a matrix C with special structures to represent the sample matrix X as $X = XC$, hoping that subspace clustering is well-determined via spectral clustering on the graph matrix $|C| + |C|^T$. The effectiveness of the self-expressiveness method is conditioned by two issues: (1) the correctness of the learned partition $J = \{J_k\}$ under which C has a block-diagonal form, and (2) the connection of each diagonal block $C_k = C(J_k^*, J_k^*)$ of C in the minimal partitions $\{J_k^*\}$. As mentioned before, the connection of matrix C_k refers to the connection of the undirected graph constructed from $|C_k| + |C_k|^T$. Our previous analysis addresses the first issue for theoretically detecting the MSS.

In this section, we address the issue of connection to support a computable optimization problem that we will propose for determining the MSS. Closed-form representation matrices are first given. Based on these closed-form representation matrices, we then exploit the conditions of connected diagonal blocks of a representation matrix in block-diagonal form. In addition, we discuss solutions of SSC and LRR.

5.1. Structures of Representation Matrices

Obviously, the representation matrix C of X is not unique since adding a matrix of null vectors of X to C results in another representation matrix of X. Notice that because C solves the linear system $X = XC$, it should have a closed-form structure. We use the singular value decomposition (SVD) of X in thin form:

$$X = U\Sigma V^T, \tag{17}$$

to represent the closed-form representation matrices, where U and V are the orthonormal matrices of the left and right singular vectors of X corresponding to its nonzero singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$, where $r = r(X)$, which are given in the diagonals of the diagonal matrix Σ. If $r < n$, V has an orthogonal complement $V_\perp$ for forming an orthogonal matrix $[V, V_\perp]$. We use the SVD together with orthogonal complement $V_\perp$ to characterize the representation matrix.

Lemma 19 *C is a representation matrix of X if and only if it has the following form*

$$C = VV^T + V_\perp H, \tag{18}$$

with a matrix $H \in \mathcal{R}^{(n-r)\times n}$. Thus, $r(C) \geq r(X)$ and $\|C\|_* \geq r(X)$. Furthermore,

(a) If C is symmetric, $H = SV_\perp^T$, that is, $C = VV^T + V_\perp SV_\perp^T$ with a symmetric S.

(b) If $\mathrm{tr}(C) = 0$, then $r(C) \geq r(X) + 1$.

Proof Based on the SVD given in (17), C is a representation matrix of X, *i.e.*, $X = XC$, if and only if $V^T = V^TC$. Hence, $C = VV^TC + V_\perp V_\perp^T C = VV^T + V_\perp H$ with an arbitrarily H.

We rewrite $H = TV^T + SV_\perp^T$ with arbitrary T and S. Then $C = [V, V_\perp] \begin{bmatrix} I_r & 0 \\ T & S \end{bmatrix} [V, V_\perp]^T$, where $r = r(X)$. Thus, $r(C) \geq r$ and $\|C\|_* \geq r$.

Furthermore, if C is symmetric, $T = 0$ and S is symmetric obviously. That is (a). Since $\mathrm{tr}(C) = r + \mathrm{tr}(S)$ by the proposition $\mathrm{tr}(AB) = \mathrm{tr}(BA)$, if C is imposed the restriction $\mathrm{tr}(C) = 0$, then $\mathrm{tr}(S) = -r$. Hence, $S \neq 0$ and $r(C) = r + r(S) \geq r + 1$. That is (b). ■

We note that a representation matrix C of X could be of arbitrary rank r' varying from $r(X)$ to $n(X)$. Practically, if we choose $H = \mathrm{diag}(H_{r'}, 0)V_\perp^T$ with any nonsingular matrix $H_{r'}$ of order $r' - r(X)$ in (18), then obviously $r(C) = r'$.

5.2. Minimal Subspace Detectable Representation

The self-expressiveness approach seeks a block-diagonal representation matrix C. That is, there is a permutation matrix Π such that, within a given or existing partition $J = \{J_\ell\}$,

$$C = \Pi \mathrm{diag}(C_{J_1}, \cdots, C_{J_{|J|}})\Pi^T,$$

where $|J|$ defines the number of partition pieces. Simultaneously, X is also partitioned as $X = [X_{J_1}, \cdots, X_{J_{|J|}}]\Pi^T$. A given partition $\{J_\ell\}$ is not naturally assumed to contain all the connected diagonal blocks $\{C_{J_\ell}\}$. In this subsection, we inspect the rank propositions of the state-of-art LRR and SSC, when their solution has a block-diagonal form. The risk of non-connected diagonal blocks is discussed even when partition $\{J_\ell\}$ is ideally chosen as a minimal partition for detecting the MSS. Finally, we prove that the connection is guaranteed under a rank restriction similar to that in the set $\mathcal{J}(K, d, d_{\min})$, and hence, the MSS can be correctly detected.

5.2.1. Propositions of LRR and SSC

LRR is known to give a representation matrix that has the smallest nuclear norm, which implies that $H = 0$ in Lemma 19, and hence, it also has the smallest rank. Meanwhile, an SSC solution has a larger rank or nuclear norm due to a nonzero H. The following lemma further characterizes the solutions of LRR and SSC.[5]

Theorem 20 *LRR provides a block-diagonal solution if and only if $r(X) = \sum_\ell r(X_{J_\ell})$ with a partition $\{J_\ell\}$. If SSC provides a block-diagonal C with a total of T connected blocks, then $r(C) \geq \sum_\ell r(X_{J_\ell}) + T$.*

Proof For the sake of simplicity, let $C'_\ell = C_{J_\ell}$ and $X'_\ell = X_{J_\ell}$. If LRR has a block-diagonal C with diagonal blocks $\{C'_\ell\}$ by $X'_\ell = X'_\ell C'_\ell$, we have $r(X'_\ell) \leq r(C'_\ell)$ and $r(C) = \sum_\ell r(C'_\ell) \geq \sum_\ell r(X'_\ell) \geq r(X)$. On the other hand, since the LRR solution is uniquely given by $C = VV^T$, we have $r(C) = r(X)$. Thus, $r(X) = \sum_\ell r(X'_\ell)$.

Conversely, if $r(X) = \sum_\ell r(X'_\ell)$ for a segmentation $\{X'_\ell\}$ of X, we partition $V^T\Pi = [B_1^T, \cdots, B_{|J|}^T]$ as $\{J_\ell\}$. Based on the thin SVD (17), we get $X'_\ell = U\Sigma B_\ell^T$ and $r(B_\ell) = r(X'_\ell)$. Let $B_\ell = Q_\ell R_\ell$ be the QR decomposition of B_ℓ with an orthonormal Q_ℓ of $r(X'_\ell)$ columns

5. The sufficient condition for LRR was given by Liu et al. (2013).

and a matrix R_ℓ of order $r(X'_\ell) \times r(X)$. Then, $V = \Pi QR$, where $Q = \mathrm{diag}(Q_1, \cdots, Q_{|J|})$ and $R^T = [R_1^T, \cdots, R_{|J|}^T]$. The condition $r(X) = \sum_\ell r(X'_\ell)$ means that R is a square matrix. Since V is orthonormal, R must be orthogonal. Therefore, the LRR solution $C = VV^T$ can be rewritten as follows

$$C = \Pi(\Pi^T VV^T \Pi)\Pi^T = \Pi Q^T Q \Pi^T = \Pi \mathrm{diag}\big(Q_1 Q_1^T, \cdots, Q_{|J|} Q_{|J|}^T\big)\Pi^T.$$

That is, C is block-diagonal.

If SSC provides an MSDR of X with a block-diagonal C of T connected diagonal blocks $\{C'_\ell\}$, then $r(C'_\ell) \geq r(X'_\ell) + 1$ by Lemma 19(b) since $X'_\ell = X'_\ell C'_\ell$ and $\mathrm{tr}(C'_\ell) = 0$. Hence, the lower bound of $r(C)$ follows immediately since $r(C) = \sum_\ell r(C'_\ell)$. ■

Strict sufficient conditions are given by Soltanolkotabi and Candès (2012) for SSC to have a block-diagonal representation matrix according to ideal segmentation $\{X_{J_k^*}\}$. These conditions are very strict and may be difficult to satisfy in applications. We will briefly discuss these sufficient conditions in Section 5.4. In addition, the block-diagonal form does not guarantee the connection of all the diagonal blocks. This phenomenon was reported by Nasihatkon and Hartley (2011). There is a notably large gap between ranks $r(C) = r(X)$ and $r(C) \geq \sum_\ell r(X'_\ell) + T \gg r(X)$ of the possible block-diagonal solutions of LRR and SSC, respectively. In the next subsection, we show how such a block-diagonal representation may be unconnected, even if it is ideally partitioned.

5.2.2. Nonconnectivity

Even if we have a block-diagonal representation matrix C in the ideal partition $J^* = \{J_k^*\}$, it is possible to have unconnected diagonal blocks in C, mainly because the solution does not have a suitable rank. This observation stems from the closed-form structure given in Lemma 19. Practically, because of the block-diagonal form of C, $X_k = X_k C_k$, where $X_k = X(:, J_k^*)$ and $C_k = C(J_k^*, J_k^*)$ as before. Hence, each C_k has the form $C_k = V_k V_k^T + (V_k)_\perp H_k$, where V_k is based on the SVD of segment X_k: $X_k = U_k \Sigma_k V_k^T$. An unsuitable H_k may result in an unconnected C_k. To further verify this observation, we consider the construction of an unconnected representation matrix of a given subset X_k, no matter whether it spans a minimal subspace or not. For the sake of simplicity, let

$$d_k = r(X_k), \quad n_k = n(X_k), \quad d = d_1 + \cdots + d_K.$$

The following lemma shows how to construct such an unconnected C_k with a given rank.

Lemma 21 *Given $r' \in (d_k, n_k]$, there is an unconnected representation C_k of X_k with $r(C_k) = r'$.*

Proof Write the integer $r' \in (d_k, n_k]$ as $r' = pd_k + t$ with $t \in [0, d_k)$. If $t = 0$, we partition $X_k = [X_{k,1}, \cdots, X_{k,p}]$, where each $X_{k,j}$ has at least d_k columns. As metioned below the proof of Lemma 19, we have a representation $C_{k,j}$ of X_{kj} with rank $r(C_{k,j}) = d_k$ since $d_k \leq n(X_{k,j})$. Thus, $C_k = \mathrm{diag}(C_{k,1}, \cdots, C_{k,p})$ is a representation of X_k. If $t \neq 0$, we partition $X_k = [X'_k, X''_k]$, where X'_k has $n_k - t$ columns. Since $pd_k = r' - t \leq n_k - t$, as

mentioned below Lemma 19 again, we have a representation matrix C'_k of X'_k with rank pd_k. Thus, $D_k = \mathrm{diag}(C'_k, I_t)$ is a representation of X_k. Obviously, $r(C_k) = r'$ and C_k is unconnected in both cases. ■

Similarly, we can construct an unconnected block-diagonal representation matrix C of X in a given segmentation $\{X_1, \cdots, X_K\}$ of X.

Theorem 22 *Given a segmentation $\{X_k\}$ of X and an integer $d_+ \in (d, n]$, there is a block-diagonal representation matrix C of X such that $r(C) = d_+$ and some diagonal blocks partitioned as per (19) are not connected.*

Proof Since $d_1 + \cdots + d_K = d < d_+ \leq n = n_1 + \cdots + n_K$, we can write $d_+ = r_1 + \cdots + r_K$ with $\{r_k\}$ satisfying $r_k \in (d_k, n_k]$ for $k \leq \ell$ and $r_k = d_k$ for $k = \ell + 1, \cdots, K$ with a suitable $\ell \geq 1$. Applying Lemma 21 to the first ℓ segments, we obtain the unconnected representation matrices C_k of X_k with rank r_k for $k \leq \ell$. For each $k > \ell$, we also have a C_k with rank d_k via the SVD of X_k as previously mentioned. Thus, $C = \Pi^T \mathrm{diag}(C_1, \cdots, C_K)\Pi$ is obviously a representation matrix of X with rank d_+. ■

5.2.3. MSDR: Minimal Subspace Detectable Representation

We seek a representation matrix of X that can be used to correctly detect the minimal subspace segmentation. This representation matrix should be partitioned block-diagonally as the MSS and all the diagonal blocks are connected. We call such a matrix the minimal subspace detectable representation (MSDR) of X.

Definition 23 *A representation matrix C of X is minimal subspace detectable if there is a permutation matrix Π such that:*

$$X\Pi = [X_1, \cdots, X_K], \quad C = \Pi \mathrm{diag}(C_1, \cdots, C_K)\Pi^T, \tag{19}$$

where $\{X_k\}$ is an MSS of X, and each C_k is connected.

Theorem 22 also shows that, if X has an MSS $\{X_k\}$, its representation matrix C with rank $r(C) > \sum_k r(X_k)$ may be not an MSDR for the MSS, even if C has a block-diagonal form partitioned according to the MSS, since some of the diagonal blocks may be unconnected. In such a case, the unconnected diagonal blocks can be divided into smaller (connected) ones. Thus, C is also a block-diagonal representation matrix with greater than K connected diagonal blocks. In addition, the K-partition learned by spectral clustering may give a nonminimal segmentation. Fortunately, connection issues can be addressed if the representation matrix has a rank equal to $\sum_k r(X_k)$.

Theorem 24 *Under the same assumptions of Theorem 11, if C is a block-diagonal representation matrix of X with $\mathrm{rank}(C) = d$ and K diagonal blocks, each greater than $d_{\min}$ in size, then C is an MSDR. Furthermore, C is unique if it is restricted to be symmetric.*

Proof Let $J = \{J_k\}$ be the partition with K pieces corresponding to the block-diagonal form and let $X_k = X_{J_k}$. Obviously, $X_{J_k} C_k = X_{J_k}$. Since $d = r(C) = \sum_k r(C_k) \geq \sum_k r(X_{J_k})$ and the size $|J_k|$ of C_k is greater than $d_{\min}$, we have $J = \{J_k\} \in \mathcal{J}(K, d, d_{\min})$. By Theorem 11, $\mathcal{J}(K, d, d_{\min})$ is a singleton and thereby $\{X_{J_k}\}$ is an MSS.

Next, we show that all the diagonal blocks of C are connected. Assume, inversely, that there is an unconnected C_i, which can be further partitioned into block-diagonal form with at least two diagonal blocks. That is, there is a permutation Π_i such that:

$$\Pi_i C_i \Pi_i^T = \operatorname{diag}(C_i', C_i''), \quad X_{J_i} \Pi_i = [X_i', X_i''].$$

Let $n_i' = n(X_i')$ and $n_i'' = n(X_i'')$. Since X_{J_i} is minimal, $r(X_i') = \min\{d_i, n_i'\}$ and $r(X_i'') = \min\{d_i, n_i''\}$. Hence, using $n_i' + n_i'' = |J_i| > d_i$,

$$\begin{aligned} r(C_i) = r(C_i') + r(C_i'') &\geq r(X_i') + r(X_i'') = \min\{d_i, n_i'\} + \min\{d_i, n_i''\} \\ &\geq \min\{d_i + n_i', d_i + n_i'', d_i + d_i, n_i' + n_i''\} > d_i. \end{aligned}$$

Combining this inequality with $r(C_k) \geq d_k$ for $k \neq i$, we obtain $r(C) = \sum_k r(C_k) > \sum_k d_k = d$, a contradiction to $\operatorname{rank}(C) = d$. Hence, C is an MSDR of X. If C is symmetric, each C_k is also symmetric, and $C_k = V_k V_k^T$ by $r(C_k) = r(X_{J_k}) = d_k$. Hence, C is unique. ■

5.3. Computable Modeling for MSDR

We are now ready to model the MSDR as an optimization problem, mainly motivated by Theorem 24. As in previous sections, we also assume that the number K and the rank sum d of segments are known for the MSS $\{X_k\}$. By Theorem 24, we restrict the feasible representation C to be of rank d. Since the symmetric MSDR is unique, we further restrict it to be symmetric as $C = C(S)$ with the following:

$$C(S) = VV^T + V_\perp S V_\perp^T, \quad r(S) = d - r, \quad S^T = S,$$

where $r = r(X)$ is known. The mapping from a symmetric S to a symmetric C is one-to-one. That is, given a symmetric C with $\operatorname{rank}(C) = d$ in the imaging domain, there is a unique $S = V_\perp^T C V_\perp$ satisfying $C(S) = C$. To enforce C to be block-diagonal reasonably, we hope that the off-block-diagonal part of C, defined as $C_{\operatorname{off}(J)}(S)$ with entries

$$\left(C_{\operatorname{off}(J)}(S)\right)_{ij} = \begin{cases} 0 & i, j \in J_k, \forall k; \\ c_{ij} & i \in J_s,\ j \in J_t,\ \forall s \neq t, \end{cases}$$

is as small as possible with $J = \{J_k\} \in \mathcal{J}(K, d, d_{\min})$. We adopt the ℓ_1-norm for minimizing this off-block-diagonal part of C. That is, we solve the following optimization model

$$\min_{J \in \mathcal{J}(K,d,d_{\min})} \min_{S \in \mathbb{S}} \|C_{\operatorname{off}(J)}(S)\|_1 \tag{20}$$

for determining an MSDR C of X, where $\mathbb{S}$ is a feasible domain of symmetric matrices of order $n - r$. Theorem 24 supports the model (20) to give an MSDR, because the solution C

of (20) should be a block diagonal matrix of K diagonal blocks, with the size of each block greater than $d_{\min}$.

There are various options for the feasible domain $\mathbb{S}$. By Theorem 24, the symmetric MSDR should be a positive semidefinite matrix and orthogonal projection operator with rank d. A feasible domain should contain such a matrix. For example, choose $\mathbb{S}$ as the set of orthogonal projection matrices as follows:

$$\mathcal{P}_{d-r}^{n-r} = \{S = QQ^T : Q \in \mathcal{O}_{d-r}^{n-r}\},$$

where $\mathcal{O}_{d-r}^{n-r} = \{Q \in \mathcal{R}^{(n-r)\times(d-r)} : Q^T Q = I_{d-r}\}$ is a Stiefel manifold. Obviously, for an $S = QQ^T$ with $Q \in \mathcal{O}_{d-r}^{n-r}$, $C(S) = GG^T \in \mathcal{P}_d^n$ with an orthonormal $G = [V, V_\perp Q] \in \mathcal{O}_d^n$. However, the Stiefel manifold $\mathcal{O}_{d-r}^{n-r}$ is strongly nonconvex; thus, one may encounter a local optimum with a solution far from the MSDR, taking Q as a variable in $\mathcal{O}_{d-r}^{n-r}$. The largely flat domain is the subspace $\mathcal{R}^{(n-r)\times(n-r)}$, but it misses the special structure of the MSDR. In this paper, we choose the feasible domain as the set of symmetric positive semidefinite matrices with rank d for $C = C(S)$,

$$\mathcal{S}_{d-r}^{n-r} = \left\{S = WW^T : W \in \mathcal{R}_+^{(n-r)\times(d-r)}\right\},$$

where $\mathcal{R}_+^{(n-r)\times(d-r)}$ is the set of full rank matrices in $\mathcal{R}^{(n-r)\times(d-r)}$. $\mathcal{S}_{d-r}^{n-r}$ is a slightly larger manifold than $\mathcal{P}_{d-r}^{n-r}$. However, it is much flatter than $\mathcal{P}_{d-r}^{n-r}$, which benefits convergence when we iteratively solve the optimization problem (22) presented later in the paper.

5.4. Comparison with Related Work

The optimization model (22) can handle cases wherein the minimal sample subspaces are intersected with each other and the intersections of pairwise subspaces are potentially significant and variant. To showcase this advantage, we compare the sufficient conditions in Theorem 11 with the conditions for LRR, iPursuit (Rahmani and Atia, 2017), SSC, and LRSSC.

As shown in Theorem 20, the LRR obtains the MSDR if and only if the subspaces are independent, that is, $\dim(\sum_j \mathcal{S}_j) = \sum_j \dim(\mathcal{S}_j)$. This condition implies that each subspace does not intersect with the sum of the other subspaces, or equivalently, $d_{\text{int}} = d_0 = 0$, which is a much stricter condition than that given in Theorem 11. It was proven by Rahmani and Atia (2017) that the iPursuit can separate two subspaces ($K = 2$) with high probability. This amounts to one of the special cases shown in Corollary 12. The condition for LRSSC is similar to that of SSC in the same form, yet it is stricter. We omit a comparison of our method's sufficient condition with that of LRSSC, but a detailed comparison with SSC is given below.

For the SSC, Soltanolkotabi and Candès (2012) showed that if the samples are uniformly distributed in the union of subspaces $\{\mathcal{S}_k\}$, and, for the basis matrices $\{U_k\}$ of $\{\mathcal{S}_k\}$, $\|U_k^T U_\ell\|_F^2 < \min\{\gamma_{k,\ell}, \gamma_{\ell,k}\}$ with the following:

$$\gamma_{k,\ell} = \frac{1}{256} \frac{d_\ell \log((n_k - 1)/d_k)}{(\log(n_k(n_\ell + 1)) + \log(K) + t)^2}, \tag{21}$$

where $t > 0$ is a given parameter, then SSC can give a block-diagonal solution partitioned as the ideal subspace segmentation with a probability approximately equal to one, depending on t, n_k, d_k, and K. We note that this claim does not imply a connected solution as we have explained earlier and mentioned by Nasihatkon and Hartley (2011).

Obviously, a small $\|U_k^T U_\ell\|_F$ implies approximate orthogonality between $\mathcal{S}_k$ and $\mathcal{S}_\ell$. The following lemma further shows that the inequality $\|U_k^T U_\ell\|_F^2 < 1$ implies that the two subspaces are not intersected with each other.

Lemma 25 *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two arbitrary subspaces with basis matrices U_1 and U_2, respectively. Then,*

$$\dim(\mathcal{S}_1 \cap \mathcal{S}_2) \leq \|U_1^T U_2\|_F^2.$$

Proof The orthogonal basis matrices of two intersected subspaces can be extended via the basis of their intersected subspace. That is, using the basis U_0 of $\mathcal{S}_1 \cap \mathcal{S}_2$, the orthogonal basis matrices U_1 and U_2 of $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, can be represented as $U_1 = [U_0, \hat{U}_1]G_1$ and $U_2 = [U_0, \hat{U}_1]G_2$ with orthogonal G_1 and G_2, and orthonormal $\hat{U}_1$ and $\hat{U}_2$ satisfying $U_0^T \hat{U}_1 = 0$, $U_0^T \hat{U}_2 = 0$. Hence,

$$\|U_1^T U_2\|_F^2 = \|[U_0, \hat{U}_1]^T [U_0, \hat{U}_2]\|_F^2 = \|U_0^T U_0\|_F^2 + \|\hat{U}_1^T \hat{U}_2\|_F^2 = \dim(\mathcal{S}_1 \cap \mathcal{S}_2) + \|\hat{U}_1^T \hat{U}_2\|_F^2,$$

which implies $\dim(\mathcal{S}_1 \cap \mathcal{S}_2) \leq \|U_1^T U_2\|_F^2$. ■

Therefore, if $\|U_1^T U_2\|_F < 1$, $\dim(\mathcal{S}_1 \cap \mathcal{S}_2) = 0$, that is, $\mathcal{S}_1$ and $\mathcal{S}_2$ are independent by Lemma 25. Since the upper bound $\gamma_{k,\ell}$ tends to zero quickly as n_k or t increases, even with a small K such as $K = 2$, the sufficient conditions $\|U_k^T U_\ell\|_F^2 < \min\{\gamma_{k,\ell}, \gamma_{\ell,k}\}$ for all k, ℓ with $\gamma_{\ell,k}$ defined in (21) generally imply the existence of pairwise independent subspaces. In practice, if $d_\ell \leq 1937$, then

$$\gamma_{k,\ell} \leq \frac{1}{256} \frac{d_\ell}{\log(n_k(n_\ell+1))} \frac{\log((n_k-1)/d_k)}{\log(n_k(n_\ell+1))} < \frac{1}{256} \frac{d_\ell}{\log(n_k) + \log(n_\ell+1)} < \frac{1}{256} \frac{d_\ell}{\log d_\ell} < 1$$

since $n_\ell > d_\ell$. Furthermore, if $n_k \leq d_\ell^2 n_\ell$, then $\gamma_{k,\ell} < \frac{1}{512} \frac{d_\ell}{\log d_\ell}$, and hence, $\gamma_{k,\ell} < 1$ if $d_\ell \leq 4281$. In real applications, the subspace dimensions are generally much smaller than 1937. Hence, if the conditions for SSC are satisfied, then $d_{\text{int}} = 0$, which differs from the independence condition, *i.e.*, $d_{\text{int}} = d_0 = 0$, for LRR.

In the conditions of Theorem 11, we permit a relatively large intersection that does not depend on the number of samples in each subspace. By Theorems 7 and 11, the minimal sample subspaces can be recovered by the optimal solution of (20) with probability 1.

6. Algorithms

We encounter several computational difficulties when we try to solve the problem (20). First, it is difficult to check whether the restriction $|J_k| > d_{\min}$ is satisfied since $d_{\min}$ is unknown if $\{d_k\}$ are variant.[6] Second, it is inconvenient to check the restriction $\sum_k r(X_{J_k}) \leq d$. Third, the objective function of (22) is neither continuous nor convex, and contains discrete and

6. If all the minimal segments have equal rank, $d_{\min} = d/K$ is known.

continuous variables with respect to the partition J and S or the factor in its symmetric factorization. Fortunately, the strict condition $|J_k| > d_{\min}$ can be implicitly satisfied when the strategy of normalized cutting is adopted for updating J generally. In the case when C is block-diagonal with K diagonal blocks, the inequality $\sum_k r(X_{J_k}) \leq d$ holds automatically since $r(X_{J_k}) \leq r(C_k)$ and $\sum_k r(C_k) = r(C) = d$. Hence, we can remove the restrictions $|J_k| > d_{\min}$ and $\sum_k r(X_{J_k}) \leq d$ in $\mathcal{J}(K, d, d_{\min})$. That is, we relax $\mathcal{J}(K, d, d_{\min})$ to the set $\mathcal{J}(K)$ of all K-partitions, and (20) is slightly modified to

$$\min_{J \in \mathcal{J}(K)} \min_{S \in \mathbb{S}} \|C_{\text{off}(J)}(S)\|_1, \tag{22}$$

where $\mathbb{S} = \mathcal{S}_{d-r}^{n-r} = \left\{S = WW^T : W \in \mathcal{R}_+^{(n-r)\times(d-r)}\right\}$.

The difficulty of mixing discrete and continuous variables can be addressed via alternatively optimizing J and $C(S)$. However, special strategies should be considered to improve the efficiency of this computation. We offer two types of alternative algorithms for this purpose. One algoritm solves (22) directly based on a manifold conjugate gradient (MCG) method for optimizing C. The other algorithm solves an equivalent pseudo-dual problem of (22) based on subspace estimation. Both methods solve the problem using the alternative rule: Optimize C given J, and update J according to the current C.

However, these two methods cannot guarantee a globally optimal solution in any case. Thus, We hybridize them by taking the solution of one method as the initial guess for the other. The motivation for this strategy is the rarity of falling into a common local minimizer of the both problems. Using this hybrid strategy, we can obtain the true minimal segmentation in our experiments if the subspaces are not heavily-intersected with each other.

6.1. Alternative Method for the Primal Problem

In the literature, alternative strategies are commonly used for optimizing multiple variables. For instance, an alternative strategy is adopted by Li et al. (2017) for minimizing the similar objective function $\|C_{\text{off}(J)}\|_1 + \alpha\|C\|_1$. It is potentially easy to optimize C given partition J, and J can be updated via normalized spectral clustering on the symmetric graph $|C| + |C|^T$ given C. However, if the spectral clustering is unstable, it may give an undesired partition when C is far from the ideal solution. Conversely, a poor partition also leads to an unacceptable solution. To decrease instability, a soft version is also considered by Li et al. (2017), in which the function $\|C_{\text{off}(J)}\|_1$ is modified to the weighted ℓ_1-norm function $\sum_{ij} w_{ij}|c_{ij}|$ with weights $w_{ij} = \frac{1}{2}\|u_i - u_j\|^2$, where u_i is a the vector of i-th components of the K eigenvectors corresponding to the K smallest eigenvalues of $|C| + |C|^T$. However, this method blurs block separation, and hence, it may also result in an unacceptable C.

We apply two types of modifications for solving the primal problem (20) using an alternative strategy. The first modification acts on the K-partition J. Different from the commonly used weight strategy, we slightly extend the support domain $\text{off}(J)$ in the objective function $\|C_{\text{off}(J)}(S)\|_1$ to an active index set Ω that covers $\text{off}(J)$. For the sake of simplicity, Ω also refers to an indication matrix whose entries ω_{ij} are 1 for the indices in Ω and zero otherwise. Hence, the function $\|C_{\text{off}(J)}(S)\|_1$ becomes $\|\Omega \odot C(S)\|_1 = \sum_{(i,j)\in\Omega} |c_{ij}(S)|$. This modification can significantly reduce the risk of obtaining an incorrect partition J, especially in the initial case when J is poorly estimated. Initially, we choose Ω to be the coarsest Ω_c with

$\omega_{ij} = 1$ for $i \neq j$ and $\omega_{ii} = 0$, *i.e.*, $\|\Omega_c \odot C(S)\|_1 = \sum_{i\neq j} |c_{ij}|$. In a later subsection, we discuss how to update the active index set Ω so that it can approach the subdomain $\mathrm{off}(J)$ as soon as J is approximately optimal.

The second modification aims to reduce the degree of nonconvexity of the function $\|\Omega \odot C(S)\|_1$ given Ω to render the modified function a bit flatter so that an iteration algorithm is less likely to fall into a local minimizer. To this end, we add the prior term $\frac{\lambda}{2}\|c(S)\|_2^2$ onto the diagonal vector $c(S)$ of $C(S)$ with parameter $\lambda > 0$. This strategy also benefits the search for a block-diagonal solution. Since we relax the strict zero-restriction on the diagonals, the prior term penalizes the diagonals of $C(S)$, and hence, the diagonals are uniformly small in general, which helps to increase the connections within each subspace in the representation $X = XC$.[7]

Combining the two modifications, we modify $\min_{S\in\mathbb{S}} \|C_{\mathrm{off}(J)}(S)\|_1$ to the following:

$$\min_{S\in\mathbb{S}} \Big\{ \|\Omega \odot C(S)\|_1 + \frac{\lambda}{2}\|c(S)\|_2^2 \Big\}. \tag{23}$$

Since S (or $C(S)$) and Ω are updated alternatively, the penalty parameter λ should balance the two terms $\|\Omega \odot C(S)\|_1$ and $\frac{1}{2}\|c(S)\|_2^2$. Thus, it makes sense to set

$$\lambda = \min\big(\lambda_0, 2\|\Omega \odot \hat{C}\|_1 / \|\hat{c}\|_2^2\big) \tag{24}$$

adaptively, using the solution $\hat{C}$ corresponding to the previous $(\hat{\Omega}, \hat{\lambda})$, $\hat{c} = \mathrm{diag}(\hat{C})$, and λ_0 is an initial setting. This strategy is efficient in our experiments.

The basic model (23) works well on some but not all complicated subspaces—it can recover the minimal segmentation of samples from some intersected subspaces if they are not heavily intersected with each other. We show the performance of this basic model compared with other state-of-art methods in the experiment section of this paper.

6.2. MCG: Manifold Conjugate Gradient Method

The problem (23) can be solved using a manifold conjugated gradient (MCG) method, but some computational issues should be addressed before applying MCG on (23). First, the objective function in (23) is not derivable. A subgradient is used as a substitute of the gradient in our analysis. Second, MCG convergence analysis requires the objective function to be smoothed. The gradient of this smooth function is a good approximate of a subgradient of the original function. Third, the gradient vectors should be projected onto the tangent space of the manifold at a point in MCG. However, only a smaller subspace of the tangent space benefits linear searching in MCG. For efficient computation, this subspace must be detected. In this subsection, we give a detailed MCG algorithm for solving (23), taking into account the above concerns and the technique of linear searching, together with convergence analysis. We also discuss some computational details of the MCG.

6.2.1. Subgradients

Writing $S = WW^T \in \mathcal{S}$ with $W \in \mathcal{R}^{(n-r)\times(d-r)}$, the objective function of (23) is as follows:

$$f(W) = \|\Omega \odot C(WW^T)\|_1 + \frac{\lambda}{2}\|c(WW^T)\|_2^2.$$

7. If there is a diagonal $c_{ii} \approx 1$, the connections of sample x_i to the others nearly vanish.

It is known that a subgradient of the function $|x|$ at a real variable x is $\mathrm{sign}(x)$ if $x \neq 0$ or any real $r \in [-1, 1]$ when $x = 0$. Since the function $\|\Omega \odot C\|_1$ is separable on its variables, the set of subgradients of function $f(C)$ at C is

$$\partial_C \|\Omega \odot C\|_1 = \big\{ \Omega \odot (\mathrm{sign}(C) + R) : \ R \in \mathcal{R}_C \big\}$$

where $\mathcal{R}_C = \{R : \ R \odot C = 0, \|R\|_\infty \leq 1\}$.

For $C = C(S) = VV^T + V_\perp S V_\perp^T$ with symmetric S specially, the definition of subgradients gives the inequality $\|\Omega \odot Y\|_1 - \|\Omega \odot C\|_1 \geq \langle B, Y - C\rangle$ for a fixed $B \in \partial_{C(S)}\|\Omega \odot C\|_1$ and all Y. Hence, choosing $Y = C(T)$ with any T, we obtain

$$\|\Omega \odot C(T)\|_1 - \|\Omega \odot C(S)\|_1 \geq \langle B, C(T) - C(S)\rangle = \langle V_\perp^T B V_\perp, T - S\rangle. \tag{25}$$

Obviously, the subgradients of convex function $\|\Omega \odot C(S)\|_1$ at S are $B_V = V_\perp^T B V_\perp$ with symmetric $B \in \partial_{C(S)}\|\Omega \odot C\|_1$, which is concluded by setting symmetric T in (25).

The subgradients of the non-convex function $\|\Omega \odot C(WW^T)\|_1$ at W can be also concluded from (25), based on Definition 8.3 in Rockafellar and Wets (1998) for a non-convex function $g(x)$ at x_0, via the inequality

$$g(x) - g(x_0) \geq \langle s, x - x_0\rangle + o(\|x - x_0\|).$$

Let $S(W) = WW^T$ as a mapping of W, and choose $S = S(W)$ and $T = S(W + \Delta)$ in (25). We see that $\|\Omega \odot C(S(W + \Delta))\|_1 - \|\Omega \odot C(T)\|_1 \geq \langle 2V_\perp^T B V_\perp W, \Delta\rangle + O(\|\Delta\|_F^2)$. Hence,

$$\partial_W \|\Omega \odot C(WW^T)\|_1 = \big\{ 2V_\perp^T B V_\perp W : \ B \in \partial_{C(S)}\|\Omega \odot C\|_1 \big\}.$$

Combining it with the gradient $2\lambda V_\perp^T \mathrm{diag}(c(WW^T))V_\perp W$ of $\frac{\lambda}{2}\|c(WW^T)\|_2^2$, we get

$$\begin{aligned}\partial f(W) &= \big\{ 2\big(\hat{B}_V + \lambda V_\perp^T \mathrm{diag}(c(S))V_\perp\big)W : \ \hat{B}_V \in \partial_S \|\Omega \odot C(S)\|_1, \ S = WW^T \big\} \\ &= \big\{ 2V_\perp^T\big(\Omega \odot (\mathrm{sign}(C) + R) + \lambda \mathrm{diag}(c)\big)V_\perp W : \ R^T = R \in \mathcal{R}_C, \ C = C(WW^T) \big\}.\end{aligned}$$

Convergence analysis requires a differentiable objective function. However, $f(W)$ is continuous but not differentiable on the zero entries of $\Omega \odot C(WW^T)$. To polish $f(W)$, we use the derivable function

$$q_\delta(t) = \begin{cases} |t| & |t| > \delta; \\ \frac{1}{2\delta}(t^2 + \delta^2) & |t| \leq \delta \end{cases} \tag{26}$$

that polishes $|t|$ within a small threshold $\delta > 0$. Its derivative $q_\delta'(t) = \mathrm{sign}(t)\min\{|t|/\delta, 1\}$ is an approximate of subgradient $s_t = \mathrm{sign}(t)$ of $|t|$ with error $s_t - q_\delta'(t) = \mathrm{sign}(t)(1 - |t|/\delta)_+$, where $t_+ = \max\{t, 0\}$. Hence,

$$f_\delta(W) = \|\Omega \odot C(WW^T)\|_{1,\delta} + \frac{\lambda}{2}\|c(WW^T)\|_2^2 \tag{27}$$

is a polishing function of $f(W)$, where $\|C\|_{1,\delta} = \|q_\delta(C)\|_1 = \sum_{ij} q_\delta(c_{ij})$. Since $\|C\|_{1,\delta}$ has the gradient $\mathrm{grad}\|C\|_{1,\delta} = \mathrm{sign}(C) \odot \min\{|C|/\delta, 1\}$, we obtain that, with $C = C(WW^T)$,

$$\begin{aligned}\mathrm{grad} f_\delta(W) &= 2V_\perp^T\big(\nabla\|\Omega \odot C\|_{1,\delta} + \lambda \mathrm{diag}(c)\big)V_\perp W \\ &= 2V_\perp^T\big(\Omega \odot \mathrm{sign}(C) \odot \min\{|\Omega \odot C|/\delta, 1\} + \lambda \mathrm{diag}(c)\big)V_\perp W.\end{aligned} \tag{28}$$

Here, we have used the equality $\Omega \odot \mathrm{sign}(\Omega \odot C) = \Omega \odot \mathrm{sign}(C)$.

The gradient of $f_\delta(W)$ is an approximation of subgradiant $S_W = 2V_\perp^T\big(\Omega \odot \mathrm{sign}(C) + \lambda \mathrm{diag}(c)\big)V_\perp W$ of $f(W)$ corresponding to $R = 0$. The error matrix is as follows:

$$E = S_W - \mathrm{grad} f_\delta(W) = 2V_\perp^T\big(\Omega \odot \mathrm{sign}(C) \odot (1 - |\omega_{ij}c_{ij}|/\delta)_+\big)V_\perp W.$$

If δ is small enough such that $\delta \le \min_{\omega_{ij}c_{ij}\neq 0} |\omega_{ij}c_{ij}|$, then $E = 0$, that is, $\mathrm{grad} f_\delta(W)$ is a subgradient of $f(W)$ at W. Furthermore, $\mathrm{grad} f_\delta(W) = 0$ means $0 \in \partial f(W)$. We use $f_\delta(W)$ with a small δ in our MCG algorithm. The following lemma shows that a local or global optimizer of f_δ is also an approximately local or global optimizer of f with an approximate error $O(\delta)$ in terms of the following:

$$n_\delta(W) = \sum_{|c_{ij}(W)|<\delta} \frac{(\delta - |c_{ij}(W)|)^2}{2\delta} \le \frac{\delta}{2}\sum 1_{|c_{ij}(W)|<\delta}.$$

Lemma 26 *Let W_δ and W_* be the minimizers of $f_\delta(W)$ and $f(W)$, respectively. Then*

$$f(W_\delta) \le f(W_*) + n_\delta(W_*).$$

Proof Since $q_\delta(t) = |t|$ if $|t| \ge \delta$, or $0 \le q_\delta(t) - |t| = \frac{(\delta-|t|)^2}{2\delta}$ otherwise. By definition, we have that $f(W) \le f_\delta(W) \le f(W) + n_\delta(W)$ for any W. Hence, $f(W_\delta) \le f_\delta(W_\delta) \le f_\delta(W_*) \le f(W_*) + n_\delta(W_*)$. ■

6.2.2. Tangent Space

The method of nonlinear conjugate gradient (NCG) updates the current W via a linear searching as $\hat{W} = W + \alpha\Delta$ for minimizing $f_\delta(W)$, where Δ is a conjugate gradient direction involved as a sum of a gradient $\mathrm{grad} f_\delta(W)$ and a conjugate gradient direction at the previous point. However, the NCG does not take into account the manifold $\mathcal{S}_{d-r}^{n-r}$ in our case. To take advantage of the manifold structure, it is required to slightly modify the conjugate gradient formula on the one hand. We will mention it in the next subsection.

On the other hand, the conjugate gradient Δ should be further modified (Journée et al., 2010). Practically, the modified point in the manifold $\mathcal{S}_{d-r}^{n-r}$,

$$S(W + \alpha\Delta) = S(W) + \alpha(W\Delta^T + \Delta W^T) + \alpha^2\Delta\Delta^T$$

contains the tangent component $T_W(\Delta) = W\Delta^T + \Delta W^T$ of the manifold at $S(W) = WW^T$. Obviously, the component of Δ that belongs to the null space $\mathcal{N}_W$ of the linear map $T_W(\Delta)$ does not contribute to the tangent space. If we split $\Delta = \Delta_N + \Delta_H$ with $\Delta_N \in \mathcal{N}_W$ and Δ_H in the orthogonal complement $\mathcal{H}_W$ of $\mathcal{N}_W$, Thus,

$$W\Delta^T + \Delta W^T = W\Delta_H^T + \Delta_H W^T, \quad \|\Delta\|_F^2 = \|\Delta_N\|_F^2 + \|\Delta_H\|_F^2.$$

Clearly, condensing Δ into the horizontal set $\mathcal{H}_W$ does not change the tangent component, but it yields a new point $S(W + \alpha\Delta_H)$. It is closer to the tangent space than $S(W + \alpha\Delta)$. Notice that both $S(W + \alpha\Delta_H)$ and $S(W + \alpha\Delta)$ are retractions of the same modified point

$S(W) + T_W(\Delta_H)$ onto the manifold $\mathcal{S}_{d-r}^{n-r}$ in the technique of manifold conjugate gradient (MCG). Therefore, the updating of W should be modified as

$$W_{\mathrm{new}} = W + \alpha_W \mathcal{P}_W(\Delta) \tag{29}$$

with a suitable step length α_W for linear searching for the (modified) conjugate gradient Δ, where $\mathcal{P}_W(\Delta) = \Delta_H$ is the projection of Δ onto horizontal set $\mathcal{H}_W$.

It is not difficult to determine the projection $\mathcal{P}_W(\Delta)$, via characterizing the subspaces $\mathcal{N}_W$ and $\mathcal{H}_W$. Practically, writing each $\Delta \in \mathcal{N}_W$ as $\Delta = WN + W_\perp F$ with an orthogonal complement $W_\perp$ of W,[8] and using the equality $\Delta W^T + W\Delta^T = 0$, we have that

$$0 = W^T(\Delta W^T + W\Delta^T)W = W^TW(N + N^T)W^TW.$$

Hence, $N + N^T = 0$ since W is of full column rank. Moreover,

$$\begin{aligned} 0 &= -WNW^T - WN^TW^T = (W_\perp F - \Delta)W^T + W(W_\perp F - \Delta)^T \\ &= W_\perp F W^T + WF^TW_\perp^T = [W, W_\perp]\begin{pmatrix} 0 & F^T \\ F & 0 \end{pmatrix}[W, W_\perp]^T, \end{aligned}$$

which implies $F = 0$. Hence, $\mathcal{N}_W = \big\{WN : \ N^T = -N \in \mathcal{R}^{(d-r)\times(d-r)}\big\}$. Furthermore, its orthogonal complement is $\mathcal{H}_W = \big\{H \in \mathcal{R}^{(n-r)\times(d-r)} : \ W^TH = H^TW\big\}$ obviously since $\langle WN, H\rangle = 0$ for all skew-symmetric N of order $d - r$.

To determine a skew-symmetric N and $H \in \mathcal{H}_W$ from the splitting $\Delta = WN + H$, at first, we eliminate the symmetric W^TH in the equality $W^T\Delta = W^TWN + W^TH$, by taking the skew-symmetric part of $W^T\Delta$. It yields the equation $W^TWN + NW^TW = E$, where $E = W^T\Delta - \Delta^TW$ is known. Thus, using the eigen-decomposition $W^TW = Q\Sigma Q^T$ and setting $\tilde{N} = Q^TNQ$ and $\tilde{E} = Q^TEQ$, this equation is simplified to $\Sigma\tilde{N} + \tilde{N}\Sigma = \tilde{E}$, and $\tilde{N}$ and N can be easily obtained as that

$$\tilde{N} = \Big(\frac{\tilde{e}_{ij}}{\sigma_i + \sigma_j}\Big), \quad N = Q\tilde{N}Q^T. \tag{30}$$

Therefore, the linear projection of Δ is $\mathcal{P}_W(\Delta) = H = \Delta - WN$.

6.2.3. Manifold Conjugate Gradients

The conjugate gradient direction Δ in the NCG is recursively defined. In our case, we set $\Delta = G_W$, where the recursive definition of G_W is slightly modified as:

$$G_W = -\mathrm{grad} f_\delta(W) + \beta_W \mathcal{P}_{W_{\mathrm{old}}}(G_{W_{\mathrm{old}}}),$$

and W_{old} is a previous point. Let $P_W = \mathcal{P}_W\big(\mathrm{grad} f_\delta(W)\big)$ be the projection of $\mathrm{grad} f_\delta(W)$ onto $\mathcal{H}_W$. The projection of G_W onto $\mathcal{H}_W$, *i.e.*, the conjugate direction H_W is also recursively defined (Absil et al., 2009, Algorithm 13),

$$H_W = -P_W + \beta_W \mathcal{P}_W\big(H_{W_{\mathrm{old}}}\big). \tag{31}$$

8. We assume that W is of full column rank for simplicity.

Initially, $H_W = -P_W$. Thus, the iteration (29) with $\Delta = G_W$ becomes

$$W_{\text{new}} = W + \alpha_W H_W, \tag{32}$$

which is an iteration of the manifold conjugate gradient method.

We use the following formula for setting the β_W in (31) for updating the conjugate direction in the Riemannian manifold

$$\beta_W = \begin{cases} \frac{\langle P_W, Y_W\rangle}{\langle Y_W, Z_W\rangle} - \frac{2\langle P_W, Z_W\rangle}{\langle Y_W, Z_W\rangle^2}\|Y_W\|_F^2, & \text{if } \langle Y_W, Z_W\rangle \neq 0; \\ 0, & \text{otherwise,} \end{cases} \tag{33}$$

where $Y_W = P_W - \mathcal{P}_W(P_{W_{\text{old}}})$ and $Z_W = \mathcal{P}_W(H_{W_{\text{old}}})$, a slight adaptation of that for the CG method in Euclidean space (Hager and Zhang, 2005). When $\langle Y_W, Z_W\rangle = 0$, the iteration is restarted. Obviously, rescaling $H_{W_{\text{old}}}$ does not change the updating process (31). Hence, one can normalize each H_W in (32) to have a unit Frobenius norm if necessary for numerical stability.

6.2.4. Line Searching

One strategy for linear searching is to choose α_W satisfying the Armijo condition on $f(W)$

$$f(W_{\text{new}}) \leq f(W) + \tau\alpha_W \inf_{B_W \in \partial f(W)} \langle \mathcal{P}_W(B_W), \mathcal{P}_W(\Delta)\rangle. \tag{34}$$

with $\tau \in (0,1)$. Mathematically, $\langle \mathcal{P}_W(Y), \mathcal{P}_W(Z)\rangle = \langle Y, \mathcal{P}_W(Z)\rangle = \langle \mathcal{P}_W(Y), Z\rangle$ for any Y and Z. Hence, only one projection is required in the inner production. In numerical computation, $f(W)$ is replaced by the smooth $f_\delta(W)$, and (34) is changed to that

$$f_\delta(W_{\text{new}}) \leq f_\delta(W) + \tau\alpha_W \langle P_W, H_W\rangle, \tag{35}$$

as suggested in Section 4.2 by Absil et al. (2009). Once the Armijo condition (35) is satisfied and H_W is a descending direction, *i.e.*, $\langle P_W, H_W\rangle < 0$, then $f_\delta(W_{\text{new}}) < f_\delta(W)$ is guaranteed.

We note that the computational cost of checking for the Armijo condition is much lower than that of other strategies for determining an α_W. For example, for the strong Wolfe conditions (Sato and Iwai, 2015)

$$f_\delta(W_{\text{new}}) \leq f_\delta(W) + c_1\alpha_W\langle P_W, H_W\rangle, \quad |\langle P_{W_{\text{new}}}, \mathcal{P}_{W_{\text{new}}}(H_W)\rangle| \leq |c_2\langle P_W, H_W\rangle|, \tag{36}$$

where the constants c_1 and c_2 satisfy $0 < c_1 < c_2 < 1$, an additional condition must be checked. For the convergence $\liminf_{k\to\infty}\|P_{W_k}\|_F = 0$ of MCG under the strong Wolfe conditions, Sato and Iwai (2015) suggests another rule for choosing β_W.

6.2.5. Convergence

The following lemma benefits convergence analysis of the MCG with β_W in (33) and linear searching satisfying the Armijo condition (35).

Lemma 27 *If β_W is chosen as (33), then for arbitrary W,*

$$\langle P_W, H_W\rangle \leq -\frac{7}{8}\|P_W\|_F^2. \tag{37}$$

Proof From updating (31) of H_W, we have $\langle P_W, H_W\rangle = -\|P_W\|_F^2 + \beta_W\langle P_W, Z_W\rangle$. If $\langle Y_W, Z_W\rangle = 0$, then $\beta_W = 0$ by (33), and $\langle P_W, H_W\rangle = -\|P_W\|_F^2$. Otherwise,

$$\begin{aligned}\beta_W\langle P_W, Z_W\rangle =& \frac{1}{8}\Big(8\frac{\langle P_W, Z_W\rangle}{\langle Y_W, Z_W\rangle}\langle P_W, Y_W\rangle - 16\frac{\langle P_W, Z_W\rangle^2}{\langle Y_W, Z_W\rangle^2}\langle Y_W, Y_W\rangle\Big) \\ =& \frac{1}{8}\Big(\|P_W\|_F^2 - \big\|P_W - 4\frac{\langle P_W, Z_W\rangle}{\langle Y_W, Z_W\rangle}Y_W\big\|_F^2\Big) \leq \frac{1}{8}\|P_W\|_F^2.\end{aligned}$$

Hence, $\langle P_W, H_W\rangle \leq -\|P_W\|_F^2 + \frac{1}{8}\|P_W\|_F^2 = -\frac{7}{8}\|P_W\|_F^2$. ■

Thus, if the Armijo condition (35) is satisfied with α_W, we have the decreasing property

$$f_\delta(W_{\text{new}}) \leq f_\delta(W) - \frac{7\tau\alpha_W}{8}\|P_W\|_F^2 \leq f_\delta(W). \tag{38}$$

This equality holds only if $P_W = 0$. Hence, starting with any point, the MCG converges in the sense that $P_{W_k} = 0$ at a W_k or

$$\lim_{k\to\infty} P_{W_k} = 0. \tag{39}$$

That is, the MCG converges globally.

Theoretically, for a sufficiently small δ, the minimizer of f_δ is also a local minimizer of f, as previously mentioned. However, a smaller δ might yield slower convergence of the MCG algorithm, which frequently occurs in numerical experiments. We use the stepped strategy of decreasing δ and use the minimizer W_δ as an initial guess for the MCG with a smaller δ. This strategy can accelerate convergence.

Theorem 28 *Let $\{\delta_\ell\}$ be a decreasing sequence and $W^{(\ell)}$ be a solution of the manifold conjugate gradient method with (33), starting with the previous $W^{(\ell-1)}$ and satisfying the Armijo condition. Then, $\{f_{\delta_\ell}(W^{(\ell)})\}$ is monotonously decreasing.*

Proof Obviously, if $\delta' < \delta''$, then $q_{\delta'}(t) \leq q_{\delta''}(t)$ for $q_\delta(t)$ given in (26) and all t. Hence, $f_{\delta'}(W) \leq f_{\delta''}(W)$ for all W, and $f_{\delta_{\ell+1}}(W^{(\ell)}) \leq f_{\delta_\ell}(W^{(\ell)})$. This equality holds only if $\min_{ij}|c_{ij}^{(\ell)}| \geq \delta_\ell$ since $q_{\delta'}(t) = q_{\delta''}(t)$ only if $|t| \geq \delta''$. By (38), $f_{\delta_{\ell+1}}(W^{(\ell+1)}) \leq f_{\delta_{\ell+1}}(W^{(\ell)})$. This equality holds only if $P_{W^{(\ell)}} = 0$. Therefore, $f_{\delta_{\ell+1}}(W^{(\ell+1)}) \leq f_{\delta_\ell}(W^{(\ell)})$ for all ℓ, and this strict inequality holds if $P_{W^{(\ell)}} \neq 0$ or $\min_{ij}|c_{ij}^{(\ell)}| < \delta_\ell$. ■

6.2.6. Computational Details

Several computational issues may affect the efficiency of the MCG algorithm: the stopping condition of the inner iteration of W given δ, the rule for choosing a suitable α_W satisfying the Armijo condition, the choice of the initial testing value of α_W, and the choice of δ. We offer details on these computational issues below.

Stopping criterion. Given $\delta > 0$, we normalize H_W to have a unit Frobenius norm prior to linear searching. Since $\alpha_W = \alpha_W\|H_W\|_F = \|W_{\text{new}} - W\|_F$, a simple stopping criterion of the iteration of W is that $\alpha_W \leq \epsilon_\alpha$ with a small constant ϵ_α.

Algorithm 1 MSS using manifold conjugated gradients (MSS_MCG)

Input: V, $V_\perp$, Ω, initial guess W, δ_0, and α_{init}, parameters ρ, γ, ϵ_α, ϵ_C, ϵ_δ, $\ell_{\max}$, and $k_{\max}$
Output: W and C.

Compute $C_v = VV^T$, $V_w = V_\perp W$, and $C = C_v + V_w V_w^T$, set $\delta = \delta_0$, and save $W^{(0)} = W$.
For $\ell = 1, 2, \cdots, \ell_{\max}$
  Save $C_{\text{old}} = C$, and compute $f = f_\delta(W)$ as (27) with the current C.
  For $k = 0, 1, 2, \cdots, k_{\max}$
    Compute $\text{grad} f_\delta(W)$ as (28) and $P = \mathcal{P}_W(\text{grad} f_\delta(W))$ as (30).
    Set $H = -P$ if $k = 0$, or compute H as (31) and (33), and $\frac{H}{\|H\|_F} \to H$ if $k > 0$.
    Starting with α_{init}, choose α satisfying the Armijo condition, but α/ρ does not.
    Update $W := W + \alpha H$, $V_w = V_\perp W$, and reset $\alpha_{\text{init}} = \alpha$.
    Update $C = C_v + V_w V_w^T$, $f = f_\delta(W)$.
    If $\alpha < \epsilon_\alpha$, terminate the inner iteration.
  End
  If $\|\Omega \odot (C - C_{\text{old}})\| < \epsilon_C$ and $\delta < \epsilon_\delta$, terminate, otherwise, reduce $\delta := \gamma\delta$.
End

Choosing α_W. To guarantee convergence by Corollary 4.3.2 of Absil et al. (2009), we determine an α_W such that α_W satisfies the Armijo condition but $\alpha'_W = \alpha_W/\rho$ does not. This is accomplished via repeatedly testing α in the rule: $\alpha := \alpha/\rho$ if (34) holds or $\alpha := \alpha * \rho$ otherwise, starting with an initial value α_0. This is basically an estimation of the largest α satisfying the Armijo condition. Taking α^*_W as a good approximation of the minimizer of $\phi(\alpha) = f_\delta(W + \alpha H_W)$, the relative approximation error is bounded,

$$0 \le \frac{\alpha^*_W - \alpha_W}{\alpha_W} \le \frac{\alpha_W/\rho - \alpha_W}{\alpha_W} = \frac{1-\rho}{\rho}.$$

Hence, a ρ closer to 1 yields a better approximate α_W to α^*_W, and hence, a smaller value of f_δ, roughly speaking. We typically choose $\rho \in [0.5, 1)$.

Initial guess of α_W. For simplifying the discussion, we normalize H_W to have a unit Frobenius norm prior to linear searching. Since α_W tends to zero as the iteration of W converges, a good estimate for α_W is $\alpha_{W_{\text{old}}}$ if the previous $\alpha_{W_{\text{old}}}$ is available. This initial guess works well in our experiments—it only takes twice testings for each update of W in general, but may fail when the curvature of $\phi(\alpha) = f_\delta(W + \alpha H_W)$ achieves a local minimum near $\alpha = 0$, which may result in a very small $\alpha_W \approx 0$. This phenomenon happens when W is close to a local minimizer or when the direction H_W is unsuitable, causing very slow descent. Thus, we change H_W back to $-P_W$ if the Armijo condition is unsatisfied under at most $k_{\max}$ testings in case the computational cost becomes prohibitive. In our experiments, we generally set $k_{\max} = 10$.

Setting δ. In practice, a sequence of decreasing $\{\delta_\ell\}$ is used. We simply choose $\delta_\ell = \gamma^{\ell-1}\delta_0$ with $\gamma < 1$. Let $W^{(\ell)}$ be the solution corresponding to δ_ℓ. We terminate the outer iteration if $\|\Omega \odot \left(C_{W^{(\ell)}} - C_{W^{(\ell-1)}}\right)\|_\infty < \epsilon_C$ with a given accuracy or $\delta_\ell \le \epsilon_\delta$, where ϵ_δ is a small constant such as ϵ_α.

The MCG algorithm is summarized in Algorithm 1.

Algorithm 2 Construct active set Ω

Input: a symmetric graph A and the parameter τ.
Output: active set Ω and partition J.

Compute the sum $a = \sum_j a_j$ of all columns of A and set $D = \text{diag}(a)$.
Compute K unit eigenvectors Q of $D^{-1/2}AD^{-1/2}$ with the largest eigenvalues.
Apply k-means on the normalized rows $\{\tilde{y}_j\}$ of Q to get centroids $\{b_k\}$ and partition J.
Compute $q_{i\ell} = \frac{\psi(\tilde{q}_{i\ell})}{\sum_k \psi(\tilde{q}_{i\ell})}$ via (40) and $\psi(t) = \begin{cases} 1, & \text{if } t < \tau; \\ 0, & \text{if } t \ge \tau. \end{cases}$
Set the active set $\Omega = \{(i,j) : q_{i\ell} q_{j\ell} < 1\}$.

6.3. Active Set Updating

Once we obtain a solution $C = C(S)$ of (23) with an active set Ω, as an estimated solution of $\min_{S \in \mathbb{S}} \|C_{\text{off}(J)}(S)\|_1$, we must update the current Ω together with λ as (24). In this subsection, we provide an effective approach for updating the active set Ω, that addresses two issues in the unnormalized spectral clustering for estimating the K-partition J: small segments and instability of classical k-means.

There is an implicit restriction $|J_k| > d_{\min}$ with unknown $d_{\min}$ for partition $J \in \mathcal{J}(K)$ in practice. This restriction implies that J_k should not be small. Hence, we adopt normalized cutting (Shi and Malik, 2000) to avoid small blocks in learning J. For the sake of completeness, we briefly describe the approach taken in this paper, which is similar to that of Von Luxburg (2007).

Normalized cutting modifies $\|C_{\text{off}(J)}\|_1 = \frac{1}{2}\sum_{ij} |c_{ij}| \|u_i - u_j\|_2^2$ to $\frac{1}{2}\sum_{ij} |c_{ij}| \|v_i - v_j\|_2^2$, by just changing the assignment vectors $u_i = e_k$ of J to the rescaled vector $v_i = e_k / \sqrt{\sum_{j \in J_k} \alpha_j}$ for $i \in J_k$, where e_k is the k-th column of I_K, the identity matrix of order K, and $\alpha_j = \sum_i |c_{ji}|$. Hence, $u_i = v_i/\|v_i\|$ can be determined by the solution of the equivalent problem $\min \text{tr}(VLV^T)$ subjected to $V = [v_1, \cdots, v_n]$ with discrete entries and $VDV^T = I_K$, where $L = D - (|C| + |C|^T)/2$ and D is a diagonal matrix of scales $\left\{\sum_{j=1}^n \frac{|c_{ij}| + |c_{ji}|}{2}\right\}$. The discrete restriction is released for computation, and hence, V is estimated by the solution of $\min \text{tr}(YLY^T)$ subjected to $YDY^T = I_K$, which is $Y = Q^T D^{-1/2}$ with Q of K unit eigenvectors of $D^{-1/2}(|C| + |C|^T)D^{-1/2}$ corresponding to the K largest eigenvalues. Therefore, $\{u_i\}$ is estimated by $\{\tilde{y}_i = y_i/\|y_i\|\}$, or equivalently, partition J is estimated by the k-means clustering of $\{\tilde{y}_i\}$. That is, we assign labels for $\{\tilde{y}_i\}$ according to the centroids $\{b_k\}$ given by k-means as follows:

$$\ell(\tilde{y}_j) = \arg\min_k \|\tilde{y}_j - b_k\|_2, \quad j = 1, \cdots, n$$

where $J = \{J_1, \cdots, J_K\}$ with $J_k = \{j : \ \ell(\tilde{y}_j) = k\}$.

However, faulty assignment may occur via k-means clustering, especially when some $\{\tilde{y}_i\}$ are located between two centers. A hard assignment strategy may mislead the partition. To address its effect on the optimization of C, we suggest using the soft strategy of setting the active set Ω based on a probability estimation p_{ij} of points $\tilde{y}_i$ and $\tilde{y}_j$ belonging to different subspaces: $\omega_{ij} = 1$ if $p_{ij} \ge \gamma$ with a constant $\gamma \in (0, 1]$, or $\omega_{ij} = 0$ otherwise. By the law of total probability, we write $p_{ij} = 1 - \sum_\ell q_{i\ell} q_{j\ell}$, where $q_{i\ell}$ is the probability of sample

Algorithm 3 Minimal subspace segmentation via alternative optimization (MSS_AO)

Input: number of subspace K, d, initial active set Ω, λ_0, τ, maximal iteration number $t_{\max}$.
Output: J and C.

1: Initially set $W = [I_d,\ 0]^T$.
2: For $t = 1, 2, \cdots, t_{\max}$
3: If $\ell = 1$, set $\lambda = \lambda_0$. Otherwise, set λ as (24).
4: Save $\Omega_{\text{old}} = \Omega$ and compute C by Algorithm 1 with the current active set Ω.
5: Update the current Ω and J by Algorithm 2 with $A = (|C| + |C|^T)/2$.
6: If $\Omega = \Omega_{\text{old}}$, terminate the iteration.
7: End

x_i belonging to the estimated subspace $\text{span}(X_{J_\ell})$. Hence, the probability of x_i and x_j belonging the same subspace is $\sum_\ell q_{i\ell} q_{j\ell}$. We set $q_{i\ell} = \frac{\psi(\tilde{q}_{i\ell})}{\sum_k \psi(\tilde{q}_{i\ell})}$ with the rescaled distance to a centroid,

$$\tilde{q}_{i\ell} = \frac{\|\tilde{y}_i - b_\ell\|_2 - \min_k \|\tilde{y}_i - b_k\|_2}{\max_k \|\tilde{y}_i - b_k\|_2 - \min_k \|\tilde{y}_i - b_k\|_2}, \tag{40}$$

where ψ is a nonincreasing function. For example, $\psi(t) = 1$ for $t \le \tau$ and $\psi(t) = 0$ otherwise, where $\tau \in (0, 1)$ is a given constant. In our experiments, we simply set $\tau = 1/2$. Algorithm 2 lists the detailed steps of the construction of Ω.

Algorithm 3 summarizes the alternative rule of updating $C(S)$ and Ω for solving (22). Compared with other state-of-art methods, this algorithm provides improved segmentation, especially when the minimal subspaces are significantly intersected with each other. We show the relevant comparisons in the experiment section of this paper. It is possible that the computed solution is locally optimal. In the next subsection, we further consider algorithmic improvements to avoid such localization as much as possible.

6.4. The Pseudo-dual Problem and Solver

The alternative method for solving the primal problem (22) provided in previous subsections may obtain only a locally optimal solution in some cases due to nonconvexity. In this subsection, we consider an algorithm for solving an equivalent pseudo-dual problem of (22) for improved capability to jump out of local minima.

6.4.1. The Pseudo-dual Problem

Changing the objective function $\|C_{\text{off}(J)}\|_1$ of the primal problem (22) as per restriction $C_{\text{off}(J)} = 0$ while changing its restriction $X = XC$ as per function $\|X - XC\|_F^2$ for minimizing and keeping the same restrictions $C^T = C$ and $r(C) = d$, we can easily obtain the following pseudo-dual problem

$$\min_{J \in \mathcal{J}(K), C} \|X - XC\|_F^2 \quad \text{s.t. } C_{\text{off}(J)} = 0,\ C = C^T,\ r(C) = d. \tag{41}$$

The pseudo-dual problem is equivalent to the primal problem under the conditions of Theorem 11, because both problems have the same unique solution by Theorem 24.

The pseudo-dual problem can be further simplified because the off-diagonal blocks of C are zero. Let $C_{J_k} = C(J_k, J_k)$ as before. We see that

$$\|X - XC\|_F^2 = \sum_k \|X_{J_k} - X_{J_k} C_{J_k}\|_F^2, \quad \text{and} \quad r(C) = \sum_k r(C_{J_k}).$$

Hence, (41) becomes

$$\min_{\{C_k\}, \{J_k\} \in \mathcal{J}(K)} \sum_{k=1}^K \|X_{J_k} - X_{J_k} C_{J_k}\|_F^2 \quad \text{s.t. } C = C^T,\ \sum_k r(C_{J_k}) = d. \tag{42}$$

It is convenient to optimize the block-diagonal C in the above problem, since this is equivalent to solving the K independent subproblems

$$\min_{r(C_{J_k}) = d'_k} \|X_{J_k} - X_{J_k} C_{J_k}\|_F^2, \quad k = 1, \cdots, K, \tag{43}$$

on a smaller scale, provided that d can be split as $d = \sum_k d'_k$ with a good estimate d'_k of the true $d_k = r(X_k)$ for each k. We discuss how to split d and how to optimize the partition given C in the next subsection.

6.4.2. Subspace Correction

Since $r(C_{J_k}) = d'_k$, $r\big(X_{J_k} C_{J_k}\big) \le d'_k$ and $\|X_{J_k} - X_{J_k} C_{J_k}\|_F^2 \ge \min_{r(Z) \le d'_k} \|X_{J_k} - Z\|_F^2$. It is known that the minimum is given by the truncated SVD of X_{J_k} with rank d'_k. That is, the minimizer $Z_k = G_k D_k Q_k^T$, where G_k and Q_k consist of the d'_k left and right singular vectors of X_{J_k}, respectively, corresponding to the d'_k largest singular values, and D_k is a diagonal matrix of the d'_k largest singular values. If we choose $C_{J_k} = Q_k Q_k^T$, then $X_{J_k} C_{J_k} = G_k D_k Q_k$. That is, $Q_k Q_k^T$ solves the subproblem $\min_{r(C_{J_k}) = d'_k} \|X_{J_k} - X_{J_k} C_{J_k}\|_F^2$, and

$$\min_{r(C_{J_k}) = d'_k} \|X_{J_k} - X_{J_k} C_{J_k}\|_F^2 = \|X_{J_k} - G_k D_k Q_k^T\|_F^2 = \sum_j \sigma_{k,j}^2 - \sum_{j \le d'_k} \sigma_{k,j}^2,$$

where $\sigma_{k,1} \ge \cdots \ge \sigma_{k,d_k}$ are all the singular values of X_{J_k}.

The splitting $d = \sum_k d'_k$ can be easily determined. Since

$$\sum_k \|X_{J_k} - X_{J_k} C_k\|_F^2 = \sum_k \sum_j \sigma_{k,j}^2 - \sum_k \sum_{j \le d'_k} \sigma_{k,j}^2,$$

minimizing $\sum_k \|X_{J_k} - X_{J_k} C_k\|_F^2$ is equivalent to collecting the d largest values of $\{\sigma_{k,j}\}$. Once the selection is completed, the splitting $d = \sum_k d'_k$ is immediately available by setting d'_k as the number of selected $\{\sigma_{k,j}\}$ in the d largest values.

We now consider how to update partition $J = \{J_k\}$ given $\{C_k\}$. Because we have obtained the spanning subspaces $\{\text{span}(G_k)\}$, partition J can be updated by the new partition $\tilde{J} = \{\tilde{J}_1, \cdots, \tilde{J}_K\}$ according to the rule of the nearest subspace for each sample, that is,

$$\tilde{J}_k = \big\{j :\ k = \arg\min_\ell \|x_j - G_\ell G_\ell^T x_j\|_2^2\big\}. \tag{44}$$

Algorithm 4 Subspace correction for solving the pseudo-dual problem (41)

Input: X, d, initial K-partition J, and max iteration number $s_{\max}$
Output: $\{J_k\}$ and $\{C_k\}$.

1: For $s = 1, 2, \cdots, s_{\max}$
2: Save $J_{\text{old}} = J$ and compute the d largest singular triples $\{g_{ki}, \sigma_{ki}, q_{ki}\}$ of each X_{J_k}.
3: Pick up d largest values from $\{\sigma_{ki}\}$, containing d'_k selected ones for each k.
4: Set $G_k = [g_{k1}, \cdots, g_{kd'_k}]$ and update J_{old} to $J = \{J_k\}$ according to (44).
5: If $J = J_{\text{old}}$, set $C_k = Q_k Q_k^T$ with $Q_k = [q_{k1}, \cdots, q_{kd'_k}]$ for each k, and terminate.
6: End

Our subspace correction method for solving the pseudo-dual problem (41) is summarized in Algorithm 4. We note that the above method is a bit similar to the K-Subspace algorithm proposed by Bradley and Mangasarian (2000) in which the dimension d_k of each true subspace $\mathcal{S}_k$ is known, and each J_k is assumed to match dimension d_k correctly. These two assumptions cannot be satisfied in the complicated case that we consider because the spanning subspaces of the minimal segments are unknown.

6.4.3. Convergence

Algorithm 4 decreases the objective function of (42). On the one hand, given $J = \{J_k\}$, the optimal blocks $\{C_k\}$ are provided by $C_k = Q_k Q_k^T$ as shown above. Hence,

$$\begin{aligned}\sum_k \|X_{J_k} - X_{J_k} C_{J_k}\|_F^2 &= \sum_k \|X_{J_k} - G_k G_k^T X_{J_k}\|_F^2 \\ &= \sum_k \sum_{j \in J_k} \|x_j - G_k G_k^T x_j\|_2^2 \geq \sum_j \min_\ell \|x_j - G_\ell G_\ell^T x_j\|_2^2.\end{aligned}$$

On the other hand, as $X_{J_k} C_{J_k} = G_k D_k Q_k$, we also have $X_{\tilde{J}_k} \tilde{C}_{\tilde{J}_k} = \tilde{G}_k \tilde{D}_k \tilde{Q}_k^T$ for the updated pairs $\{\tilde{C}_{\tilde{J}_k}, \tilde{J}_k\}$ of $\{C_{J_k}, J_k\}$ since $\tilde{G}_k \tilde{D}_k \tilde{Q}_k^T$ is a truncated SVD of $X_{\tilde{J}_k}$. Hence,

$$\begin{aligned}&\sum_j \min_\ell \|x_j - G_\ell G_\ell^T x_j\|_2^2 = \sum_k \sum_{j \in \tilde{J}_k} \|x_j - G_k G_k^T x_j\|_2^2 = \sum_k \|X_{\tilde{J}_k} - G_k G_k^T X_{\tilde{J}_k}\|_F^2 \\ \geq &\sum_k \|X_{\tilde{J}_k} - \tilde{G}_k \tilde{D}_k \tilde{Q}_k^T\|_F^2 = \sum_k \|X_{\tilde{J}_k} - X_{\tilde{J}_k} \tilde{C}_k\|_F^2.\end{aligned}$$

Therefore, $\sum_k \|X_{J_k} - X_{J_k} C_{J_k}\|_F^2 \geq \sum_k \|X_{\tilde{J}_k} - X_{\tilde{J}_k} \tilde{C}_k\|_F^2$. The alternative iteration converges in the sense of decreasing the value of the objective function. Because only a finite number of partitions exist, the alternative iteration can be terminated within a finite number of steps as the function value is unchanged, though J may have a differently modified $\tilde{J}$.

Theorem 29 *The algorithm of subspace correction yields a decreasing sequence of objective values and terminates within a finite number of iterations.*

It should be pointed out that multiple partitions achieving the same objective values may exist if some samples have the same minimal distances to different estimated sample

subspaces. If this happens at a terminated partition J and its modified $\tilde{J}$, the equalities of objective values imply the equalities

$$\sum_k \sum_{j \in J_k} \|x_j - G_k G_k^T x_j\|_2^2 = \sum_j \min_\ell \|x_j - G_\ell G_\ell^T x_j\|_2^2 = \sum_k \sum_{j \in \tilde{J}_k} \|x_j - \tilde{G}_k \tilde{G}_k^T x_j\|_2^2.$$

One may understand the difference between J and $\tilde{J}$ by the arbitrary labeling of such samples because of their equal distances. To clearly show this, let

$$M_j = \left\{k : \|x_j - G_k G_k^T x_j\|_2 = \min_\ell \|x_j - G_\ell G_\ell^T x_j\|_2\right\}$$

for each j. Then, each J_k can be split as $J_k = J_k^0 \cup J'_k$, where J_k^0 consists of the j's with a singleton $M_j = \{k\}$, and J'_k is a set of partial j's whose M_j has at least two indices, one of which is k. Similarly, $\tilde{J}_k = J_k^0 \cup \tilde{J}'_k$. Obviously, $J_k \neq \tilde{J}_k$ is equivalent to $J'_k \neq \tilde{J}'_k$. Randomly labeling these j's according to the multiple k's in M_j results in multiple partitions. That is, there are multiple options for setting $\tilde{J}$ in this case. It is unclear whether there is a $\tilde{J}$ among the multiple choices that achieves a smaller value of the objective function. Choosing such a $\tilde{J}$ may obtain better convergence but requires a complicated labeling rule, rather than the simple one (44). We do not intend to further exploit the multiplicity of partitions because of the nonsingletons $\{M_j\}$.

6.5. Hybrid Optimization

Both Algorithms 3 and 4 may fall into local minimizers, but exhibit their own convergence behaviors. Algorithm 3 is relatively stable on the initial setting of partition J or active set Ω, in the sense that the convergent solution C or J always has good accuracy with respect to the minimal partition, although the solution may not be completely correct. Algorithm 4 heavily depends on the initial guess of J and may give a completely incorrect solution if the initial partition is poor. A good initial J for Algorithm 4 should ensure that each J_k dominates samples from the same minimal segment. In this case, the algorithm 4 converges to the true minimal partition quickly.

In this subsection, we consider a hybrid strategy for minimal subspace learning that combines primal and pseudo-dual optimization, which we term hybrid optimization. Essentially, starting with the coarsest active set Ω_c covering all index pairs (i, j) except the diagonal indices $\{(i, i)\}$, the hybrid strategy first solves the primal problem (22) with an active set Ω and then solves the pseudo-dual problem (41) using the primal solution as its initial guess. This procedure is repeated if necessary.

The key issue for hybrid optimization is constructing an initial guess for the primal (pseudo-dual) algorithm from the solution of the pseudo-dual (primal) algorithm. It is easy to construct an initial partition for the pseudo-dual algorithm (Algorithm 4) based on subspace correction using the solution given by the primal algorithm (Algorithms 3). Here, we focus on constructing a suitable Ω for Algorithms 3, based on partition $J = \{J_\ell\}$ given by Algorithm 4. Here, Ω means the matrix with entries ω_{ij}. We may slightly change the $0-1$ setting of the entries to that with one of the three values $0, 1, \beta$ because of the property of subspace correction.

Algorithm 5 Minimal subspace segmentation via hybrid optimization (MSS_HO)

Input: X, K, parameter β, maximal iteration number $h_{\max}$ of HO
Output: J and C

Initially set $\Omega = \Omega_c$ with $\omega_{ij} = 1$ if $i \neq j$ and $\omega_{ii} = 0$.
For $h = 1, 2, \cdots, h_{\max}$
    Save $\Omega_{\text{old}} = \Omega$, and solve the primal problem (22) to get J_{prim} and C by
        Algorithm 3 with Ω.
    Solve the pseudo-dual problem (41) to get J by Algorithm 4 using J_{prim} initially.
    Update Ω as (46) if there is $J_k = \emptyset$, or as (45) otherwise. If $\Omega = \Omega_{\text{old}}$, terminate.
End

If all the K subsets are not empty, it is highly possible that each J_ℓ is dominated by a single true segment. Thus, we modify Ω as follows:

$$\omega_{ij} = \begin{cases} 1 & \text{if } i \in J_s, j \in J_t,\ s \neq t; \\ 0 & \text{if } i, j \in J_k. \end{cases} \tag{45}$$

However, if there are some empty J_k, without loss of generality, let $J_1, \cdots, J_\ell$ be all the nonempty subsets with $\ell < K$. Since some true minimal segments are approximately merged together into a nonempty J_k because of the subspace correction, the entries in the off-diagonal block $C(J_s, J_t)$ decrease faster than those $C(J^*_{k'}, J^*_{k''})$ with $k' \neq k''$ if $J^*_{k'}$ and $J^*_{k''}$ are merged together. We slightly modify the coarse Ω_c to $\Omega_\beta = (\omega_{ij})$ with the following

$$\omega_{ij} = \begin{cases} \beta & \text{if } i \in J_s, j \in J_t,\ s \neq t; \\ 1 & \text{if } i, j \in J_k,\ i \neq j; \\ 0 & \text{if } i = j. \end{cases} \tag{46}$$

The constant β plays a special role in controlling the convergent behavior of $C = C(S)$ in the iteration of Algorithm 3 using Ω_β. Compared with the iteration of C in Algorithm 3 before shifting to Algorithm 4 for updating the partition, a larger $\beta > 1$ can accelerate the decreasing of the blocks $C(J_s, J_t)$ with $s \neq t$ because of the larger weights in the function

$$\|\Omega_\beta \odot C\|_1 = \sum_k \sum_{i \neq j \in J_k} |c_{ij}| + \beta \sum_{s \neq t} \|C(J_s, J_t)\|_1.$$

Once these $C(J_s, J_t)$ are small, block $C(J^*_{k'}, J^*_{k''})$ begins decreasing, a bit similar to the result of applying Algorithm 3 on the smaller block $C(J_k, J_k)$. Thus, a larger β helps to turn off the decreasing early. Note that a smaller $\beta < 1$ can balance $\{C(J_s, J_t)\}$ and $\{C(J^*_{k'}, J^*_{k''})\}$ since it can delay the decreasing of the $C(J_s, J_t)$, or equivalently, relatively accelerate the decreasing of $C(J^*_{k'}, J^*_{k''})$. When such balance occurs, the merging of multiple minimal segments might also be delayed in Algorithm 4 using such a locally optimal solution of Algorithm 3. However, the value of β must be carefully chosen to balance the decreasing of these blocks. For the sake of simplicity, we just suggest using a $\beta > 1$. In our experiments, we always set $\beta = 1.25$. See Algorithm 5 for the procedure of our hybrid optimization procedure.

Figure 2 plots four indication matrices E_J of four partitions obtained by the hybrid algorithm, where $(E_J)_{ij} = 1$ if $i, j \in J_k$, or $(E_J)_{ij} = 0$ if i, j belong to different J_k's.

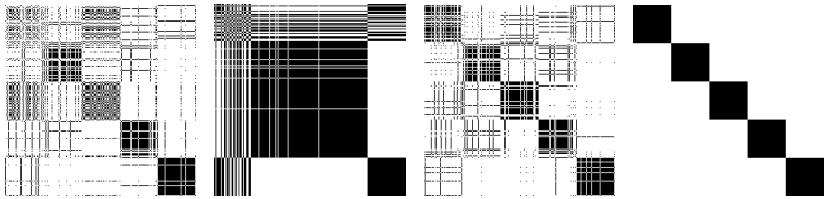


Figure 2: Indication matrix $E_{J^{(t)}}$ of the computed partition of Algorithm 3 using Ω_c (left) or Ω_β (middle right) corresponding to the solution (middle left) of Algorithm 4 starting with the left J. The right J is a solution of Algorithm 4 initially using the middle right J.

The data set has minimal segmentation consisting of five minimal segments of equal size. Starting with the coarsest Ω_c, Algorithm 3 obtains a solution J whose indication matrix is plotted on the left. J contains five nonempty pieces $\{J_k\}$; three of them have relatively dominant indices from a single minimal segment and other two are mixed by multiple minimal segments. Due to this mixture, the pseudo-dual step gives a partition with four nonempty pieces, in which two of them are very small, one almost contains the indices of a minimal segment, and the largest one is dominated by other minimal segments. The initial setting (46) can significantly reduce the mixture at the primal step; see the indication matrix plotted third in Figure 2. Due to the improvement, the second pseudo-dual step correctly recovers all the minimal segments.

7. Generalization for Noisy Samples

Given a noisy sample set $X = X_* + E$, where X_* is a set of unknown clean samples and E is a noise set, we seek an MSDR with respect to the subspaces $\{\mathcal{S}_k\}$ spanned by the minimal segments of X_* theoretically. Our algorithms discussed in previous sections are based on the orthonormal matrix V of the right singular vectors of X_* corresponding to nonzero singular values and its orthogonal complement $V_\perp$. If we can estimate V from noisy samples with high accuracy, these algorithms can also be used for subspace segmentation from noisy samples. Numerically, the required orthonormal matrix V can be estimated by the dominant right singular vectors of X. The classical perturbation theory of singular values shows that the estimation of V, or more preciously, the estimation of the subspace spanned by the dominant singular vectors, is robust if the noise is relatively smaller than the smallest singular value. This means that if the smallest singular value of X_* is large, the data noise could be somewhat relatively large. Further, we could apply the MSS algorithms on the estimated V and its orthogonal complement $V_\perp$ for segmenting the noisy samples. Perturbation theory on solutions is an interesting topic. We leave it as further work.

In this section, we consider a robust approach for the complicated cases of considerable noise or an incorrectly estimated dimension sum d. A generalized sparse model is given to handle noisy samples. We solve this sparse problem by applying an ADMM method.

7.1. The Relaxed Optimization

Let C_* be an MSDR of the latent samples X_*, that is, $X_* = X_* C_*$ and C_* is a block-diagonal matrix with rank d. By Theorem 24 on the one hand, $C_* = G_* G_*^T$ with an orthonormal G_* of d columns. On the other hand, since $X_* = X - E$, the required SDR also satisfies $X = XC_* + E(I - C_*)$. That is, the self-expressive error $X - XC_*$ should be as small as the noise in magnitude. Therefore, it makes sense for approximating the MSDR to minimize the sparsity $\|C_{\text{off}}\|_1$ subjected to $C = GG^T$ with an orthonormal $G = V_* + (V_*)_\perp Q$ of d columns, if the correct sum d of subspace dimensions is known, we should also simultaneously minimize the self-expressive error $X - XC$ in a suitable measurement $\phi(X - XC)$. That is, we may consider the optimization problem

$$\min_{C=GG^T, G^TG=I_d} \Big\{ \|C_{\text{off}}\|_1 + \alpha\phi(X - XC) \Big\}.$$

Some relaxations on the above problem are required for efficient computation. Here, we adopt four kinds of relaxations:

1) relax the first term to $\|\Omega \odot C\|_1$ with an Ω that can be modified iteratively,
2) add a penalty term on the diagonal vector c of C to decrease the nonconvexity,
3) relax the special restriction on $G = V_* + (V_*)_\perp Q$ to be a general G of d columns, and
4) relax the strict expressive matrix $C = GG^T$ to $C \approx GG^T$.

The first two relaxations are similar to those discussed in previous sections without noise, as in the basic model (23). The third one can avoid estimating the dimension r of data space, its orthogonal basis V_*, and orthogonal complement of V_*. The last one considers the case where d may be approximately estimated. The approximation $C \approx GG^T$ can be implicitly obtained by minimizing the error function $\|C - GG^T\|_F^2$. Combining these relaxations together results in the following sparse problem,

$$\min_{C,\Omega,G} \Big\{ \mathcal{F}(C,\Omega,G) = \|\Omega \odot C\|_1 + \frac{\lambda}{2}\|c\|_2^2 + \alpha\phi(X - XC) + \frac{\beta}{2}\|C - GG^T\|_F^2 \Big\}. \tag{47}$$

We may choose ϕ as the norm $\|\cdot\|_1$, $\|\cdot\|_{2,1}$, or $\|\cdot\|_F^2$, depending on the noise distribution. It is reasonable to set the parameters λ, α, and β proportional to the number of samples in each subspace. If the number of samples in each subspace are approximately equal, we can set the parameters proportional to n/K. We will show that this strategy works well numerically.

As in the noiseless case, we also solve (47) via alternatively optimizing (C, G) and Ω: We solve

$$\min_{C,G} \mathcal{F}(C,\Omega,G) \tag{48}$$

with a fixed Ω, and then update Ω when (C, G) is updated. The problem (48) is also addressed via alternatively optimizing C and G because we have the following two sound propositions:

Algorithm 6 Graph construction from C

Input: C, γ, σ, and $s \geq 1$.
Output: graph matrix A.

1: Cut off small entries of C to get a sparse C_γ with column norms $\|C_\gamma e_j\| \approx \gamma\|Ce_j\|$.
2: Compute the left singular vectors $\{u_i\}$ of C_γ corresponding to singular values $\sigma_i \geq \sigma$.
3: Construct $A = \big(\langle g_i, g_j\rangle^s\big)$ with the nomalized rows $\{g_j\}$ of $[\cdots, \sqrt{\sigma_i}u_i, \cdots]$.

(1) The subproblem $\min_G \mathcal{F}(C,\Omega,G)$ is equivalent to $\min_{G\in\mathcal{R}^{n\times d}} \|C - GG^T\|_F^2$ whose solution is as follows:

$$GG^T = P\text{diag}((\lambda_1)_+, \cdots, (\lambda_d)_+)P^T, \tag{49}$$

where $\{\lambda_i\}$ are the d largest eigenvalues of $(C + C^T)/2$ and P consists of the corresponding eigenvectors.

(2) The subproblem $\min_C \mathcal{F}(C,\Omega,G)$ is convex and has a unique solution. We can solve this problem using the ADMM method given in the next subsection.

Therefore, the algorithm for solving (48) consists of an inner-outer iteration scheme. In addition, because the objective function is monotonously decreasing, the algorithm is convergent. Numerically, it is unnecessary to solve the inner problem for updating C with high accuracy. An inaccurate solution by ADMM is sufficient if it can decrease the value of $\mathcal{F}$, which results in an inaccurate inner-outer iteration method for solving (48).

As soon as (C, S) are updated, we modify Ω using the rule given in Subsection 6.3. That is, we construct a new Ω by Algorithm 2 with input graph A. The simple input $A = (|C| + |C|^T)/2$ is no longer suitable for noisy data since it is not positive semidefinite. We give an approach for constructing a symmetric graph A as input to Algorithm 2. The construction combines the advantages of that based on SSC solutions or LRR solutions as shown in Algorithm 6.

7.2. ADMM Approach

We rewrite the subproblem of (47) for optimizing C given (Ω, G) as the ADMM form

$$\min_{C,E,Z} \Big\{ \|\Omega \odot C\|_1 + \frac{\lambda}{2}\|c\|_2^2 + \alpha\phi(E) + \frac{\beta}{2}\|Z - GG^T\|_F^2 \Big\} \quad \text{s.t. } E = X - XZ,\ Z = C.$$

Its augmented Lagrangian function is

$$\begin{aligned} L(C,E,Z) = {} & \|\Omega \odot C\|_1 + \frac{\lambda}{2}\|c\|_2^2 + \alpha\phi(E) + \frac{\beta}{2}\|Z - GG^T\|_F^2 + \frac{\rho'}{2}\|C - Z\|_F^2 \\ & + \frac{\rho''}{2}\|X - XZ - E\|_F^2 + \langle C - Z, Y'\rangle + \langle XZ + E - X, Y''\rangle. \end{aligned}$$

Hence, the ADMM scheme is given by the following alternative rule.

$$(\hat{C}, \hat{E}) = \arg\min_{C,E} L(C, E, Z); \tag{50}$$

$$\hat{Z} = \arg\min_{Z} L(\hat{C}, \hat{E}, Z); \tag{51}$$

$$\hat{Y}' = Y' + \rho'(\hat{C} - \hat{Z}); \tag{52}$$

$$\hat{Y}'' = Y'' + \rho''(X\hat{Z} + \hat{E} - X). \tag{53}$$

Notice that the objective function of (50) is separable as $L(C, E, \cdot) = L_1(C) + L_2(E) +$ constant, where

$$\begin{aligned} L_1(C) &= \|\Omega \odot C\|_1 + \frac{\lambda}{2}\|c\|_2^2 + \frac{\rho'}{2}\|C - Z + \frac{1}{\rho'}Y'\|_F^2, \\ L_2(E) &= \alpha\phi(E) + \frac{\rho''}{2}\|XZ + E - X + \frac{1}{\rho''}Y''\|_F^2, \end{aligned}$$

and the constant means a term not depending on the variables C or E. Hence, $\hat{C} = \arg\min_C L_1(C)$ and $\hat{E} = \arg\min_E L_2(E)$. Below, we provide solutions of the above three subproblems in closed form. The convergence of this ADMM iteration is guaranteed by Boyd et al. (2011).

7.2.1. Updating the Representation Matrix

The step of updating C is separable with respect to its entries. That is, updating each entry of C is an independent procedure. The entry c_{ij} is updated by $\hat{c}_{ij}$, the solution

$$t^* = \arg\min_t \left\{\omega|t| + \frac{a}{2}t^2 + \frac{\rho'}{2}(t-p)^2\right\}$$

with the parameters $\omega = \omega_{ij}$, $a = \lambda$ if $i = j$ or $a = 0$ otherwise, and $p = z_{ij} - y'_{ij}/\rho'$. Since the objective function can be rewritten as $\frac{\rho'+a}{2}\big(t - \frac{\rho' p - \mathrm{sign}(t)\omega}{\rho'+a}\big)^2$, letting $t_1 = \frac{\rho' p - \omega}{\rho'+a}$ and $t_2 = \frac{\rho' p + \omega}{\rho'+a}$, the solution is as follows:

$$t^* = \left\{\begin{array}{ll} t_2, & \text{if } t_2 \le 0 \\ 0, & \text{if } t_1 < 0 < t_2 \\ t_1, & \text{if } t_1 \ge 0 \end{array}\right\} = \frac{\mathrm{shrink}(\rho' p, \omega)}{\rho' + a},$$

where $\mathrm{shrink}(\beta, \alpha) = \mathrm{sign}(\beta)(|\beta| - \alpha)_+$ is a shrinkage operator of β corresponding to α. Hence, the optimal solution $\hat{C}$ is given by the following

$$\hat{C} = R \odot \mathrm{shrink}\left(\rho' Z - Y', \Omega\right), \tag{54}$$

where R has the diagonals $1/(\rho' + \lambda)$ and the off-diagonals $1/\rho'$, and shrink(B, A) is the elementwise operator of $\mathrm{shrink}(\beta, \alpha)$.

Algorithm 7 Minimal subspace segmentation via relaxed optimization (MSS_RO)

Input: X, K, d, accuracy parameters τ and ϵ, and max iteration numbers $t_{\max}, \ell_{\max}, k_{\max}$
Output: J and C.

```
Initially set Ω = Ω_c, Z = C = I, Y' = 0, and Y'' = 0, and save Ω_old = Ω.
Repeat the following produce for at most t_max times.
    Solve (47) with fixed Ω via the following inner iterations:
    For ℓ = 1, ··· , ℓ_max
        Save C_old = C and run the ADMM iteration for updating C:
        For k = 1, 2, ··· , k_max
            Save Z_old = Z and update C, E, and Z as (54), (55), and (56), respectively.
            Modify the multipliers Y' and Y'' as (52) and (53).
            If ‖Z − Z_old‖_F < τ, set C = Z, and terminate the iteration.
        End
        If ‖C − C_old‖_F < ε, terminate. Otherwise, update G as (49).
    End
    Construct the graph A by Algorithm 6 and update Ω and J by Algorithm 2 .
    If Ω = Ω_old, terminate the repeat.
End
```

7.2.2. Updating the Error Matrix

The solution of $\min L_2(E)$ depends on the function ϕ. If $\phi(E)$ is one of the three functions $\|E\|_1$, $\|E\|_{2,1}$, or $\|E\|_F^2$, the solution is closed-form with $\Delta = X - XZ - Y''/\rho''$,

$$\hat{E} = \left\{\begin{array}{ll} \mathrm{shrink}\left(\Delta, \alpha/\rho''\right), & \text{if } \phi(E) = \|E\|_1; \\ \Delta\mathrm{diag}(\beta_1, \cdots, \beta_n), & \text{if } \phi(E) = \|E\|_{2,1}; \\ \Delta/(1 + 2\alpha/\rho''),, & \text{if } \phi(E) = \|E\|_F^2, \end{array}\right. \tag{55}$$

Here, the first form is similar to that given by Beck and Teboulle (2009). In the second form, $\beta_i = (\|\delta_i\|_2 - \alpha/\rho'')_+/\|\delta_i\|_2$ with the columns δ_i of Δ, as shown by Yang et al. (2011).

7.2.3. Updating the Relaxation Variable

Fixing $\hat{C}$ and $\hat{E}$, $L(\hat{C}, \hat{E}, Z)$ is a quadratic function of Z. Hence, its minimizer is unique and is given by the solution of the equation $\frac{\partial}{\partial Z}L(\hat{C}, \hat{E}, Z) = 0$. That is,

$$\beta(Z - GG^T) + \rho'(Z - \hat{C}) + \rho'' X^T(XZ + \hat{E} - X) - Y' + X^T Y'' = 0.$$

Thus, the minimizer of (51) is

$$\hat{Z} = \big((\beta + \rho')I + \rho'' X^T X\big)^{-1}\big(\beta GG^T + \rho'\hat{C} + \rho'' X^T(X - \hat{E}) + Y' - X^T Y''\big). \tag{56}$$

The whole iterative procedure for solving (47) is summarized in Algorithm 7, which combines the closed-forms of solutions of the subproblems (50) and (51), the ADMM iteration, and the updating of Ω if necessary. The number of repeats $t_{\max}$ can be small. In some examples, setting $t_{\max} = 1$ also gives a good solution.

8. Experiments

We evaluate the performance of our algorithms on synthetic data sets without noise and two kinds of real-world data sets for face recognition and motion detection. The synthetic data vectors are sampled from the union of several known subspaces $\{\mathcal{S}_k\}$ such that the segment X_k of samples from $\mathcal{S}_k$ is minimal and $\text{span}(X_k) = \mathcal{S}_k$ for each k. The minimal subspaces are intersected and some samples are close to but do not belong to the intersections of these subspaces. Two key issues known to affect subspace segmentation are subspace intersections and samples located near intersected subspaces. The synthetic data are tested to show how the proposed algorithms perform when the subspaces are heavily intersected and some samples are close to the intersected subspaces. Two real-world data sets are used to evaluate how the proposed algorithms perform on noisy data. Our algorithms are also compared with five algorithms for subspace clustering: LRR (Liu et al., 2013), CLAR (Kang et al., 2015), SSC (Elhamifar and Vidal, 2013), LRSSC (Wang et al., 2013), and SoftS3C (Li et al., 2017).[9] The reported results of these algorithms were obtained using the codes provided by algorithm owners or downloaded from open sources. The parameters are set as suggested by the algorithm owners or carefully chosen by us. The original LRSSC was slightly modified in the experiments on real-world data sets to achieve better results.

8.1. Evaluation Criteria

We use the following four measurements to evaluate the quality of the computed solutions from the partition error, the deviation from block-diagonal form, and the connection of diagonal blocks, according to the ideal minimal segments J^*.

(1) The error of partition J. This error is defined by the percentage of misassigned samples in partition J compared with the true minimal partition J^*,

$$\text{ErrParti}(J) = \min_{\pi} \frac{1}{n}\Big\{n - \sum_{k=1}^{K} |J_{\pi(k)} \cap J_k^*|\Big\}, \tag{57}$$

where $\{\pi(1), \cdots, \pi(K)\}$ is a permutation of $\{1, \cdots, K\}$.

(2) Block-diagonal deviation. Wang et al. (2013) used the metric $\frac{\|C_{\text{off}(J^*)}\|_1}{\|C\|_1}$ to define the relative deviation. However, relatively large diagonals may cause a relatively small value of this function, which may lead to an incorrect gloss for the deviation since such a small value does not imply a small deviation from the ideal block-diagonal form. We modify the metric to the following

$$\text{BdiagDevi}(C) = \frac{\|C_{\text{off}(J^*)}\|_1}{\|C\|_1 - \|c\|_1} \tag{58}$$

by removing the diagonals from the denominator, and use it to measure the deviation of C from the ideal block-diagonal form.

(3) Intra-block connection. The Gini Index was used by Hurley and Rickard (2009) to measure the sparsity of a vector. Wang et al. (2013) adopted it as a sparsity metric

9. We omit a comparison with the hard version proposed in the same paper (Li et al., 2017) since the soft version SoftS3C performed slightly better than the hard version in our experiments.

GiniIndex(C_{J^*}) of all the diagonal blocks $C_{J^*} = \{C_{J_k^*}\}$, that is, the sparsity of the vector of entries in the K diagonal blocks $\{C_{J_k^*}\}$. We slightly modify it to measure the intra-block connection of C by $1 - \text{GiniIndex}(C_{J^*})$. That is, sorting all the $M = \sum_{k=1} |J_k^*|^2$ entries of the diagonal blocks $C_{J^*} = \{C_{J_k^*}\}$ as $\tilde{c}_1 \leq \cdots \leq \tilde{c}_M$, the intra-block connection is defined as follows

$$\text{IntraBConn}(C) = \sum_{\ell=1}^{M} \frac{|\tilde{c}_\ell|}{\|C_{J^*}\|_1}\left(\frac{2(M-\ell)+1}{M}\right). \tag{59}$$

(4) K-block-diagonal structure. It was shown by Von Luxburg (2007) that C is a block-diagonal matrix with K blocks if and only if the (normalized) Laplacian L of $(|C| + |C|^T)/2$ has only K zero eigenvalues. In the approximate case, C is a block-diagonal matrix with K diagonal blocks approximately if L has K small eigenvalues and its other eigenvalues are distinguishable from the K smallest ones. The spectral clustering is just the K partition of the (normalized) rows of the orthonormal matrix with K unit eigenvectors corresponding to the smallest eigenvalues as its columns. By classical subspace perturbation theory (Stewart and Sun, 1990), the stability of the spectral clustering can be characterized by the gap between the K-th and $(K+1)$st smallest eigenvalues $\lambda_K(L)$ and $\lambda_{K+1}(L)$ of L. Hence, we use the relative gap

$$\text{KblockDiag}(L) = \frac{\lambda_{K+1}(L) - \lambda_K(L)}{\lambda_{K+1}(L)} \tag{60}$$

to measure the stability of the spectral clustering, similar to $\frac{\lambda_{K+1}(L)-\lambda_K(L)}{\lambda_K(L)-\lambda_{K-1}(L)}$ used by Lauer and Schnorr (2009). Here, $\text{RelGap}(L) \in [0, 1]$.[10]

Each metric function above has the same range of $[0, 1]$. The first two functions measure the approximation of the K partition to the minimal partition J^* and the approximation of C to have a block-diagonal form as an MSDR. The last two functions measure the. connection of the diagonal blocks from two different viewpoints. Generally, smaller values of ErrParti(J) and BdiagDev(C) and relatively larger values of IntraBCon(C) and KblockDiag(L) mean a better solution (J, C) for learning the minimal sample subspaces.

8.2. Synthetic Data without Noise

The synthetic samples are generated such that the sample spanned subspaces are intersected with each other. We randomly choose an r-dimensional subspace in $\mathcal{R}^m$ given r, or equivalently, choose an orthogonal basis matrix $U \in \mathcal{R}^{m\times r}$. Then, we arbitrarily choose K subspaces $\{\mathcal{S}_k\}$ with bases $U_k = UP_k$, $k = 1, \cdots, K$, where each $P_k \in \mathcal{R}^{r\times d_k}$ is also an orthonormal matrix with a given column number $d_k < r$. The parameters $\{d_k\}$ and r determine the subspace complexity and subspace segmentation difficulty. Generally, a smaller sequence $\{d_k\}$ makes segmentation easier. We choose different kinds of $\{d_k\}$ for generating sample sets with variant complexities of the segmentation. The samples are randomly chosen from each subspace $\mathcal{S}_k$ in the form $x = U_k y$, where the entries of y are independent and identically distributed in the uniform distribution. Thus, the sample set can be written as $\{X_k = U_k Y_k\}$. As shown in Theorem 7, these samples are intersected nondegenerately and

10. The original definition defined by Lauer and Schnorr (2009) could be arbitrarily large.

		$r=10$					$r=14$					$r=20$				
	d_c	4	5	6	7	8	7	8	9	10	11	11	12	13	14	15
Average dimension of $\mathcal{S}_k \cap \mathcal{S}_\ell$	$\tau_\sigma = 0.9$	0.69	1.54	2.68	4.13	6.01	2.10	3.21	4.51	6.12	8.01	4.00	5.24	6.65	8.17	10.03
	0.990	0.03	0.59	2.03	4.00	6.00	0.78	2.11	4.00	6.00	8.00	2.25	4.02	6.00	8.00	10.00
	0.999	0.00	0.20	2.00	4.00	6.00	0.27	2.00	4.00	6.00	8.00	2.01	4.00	6.00	8.00	10.00
Percentage of samples with small dist(x_i)	$[0.00, 0.05)$	0.0	0.0	0.0	0.3	3.9	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0
	$[0.05, 0.10)$	0.0	0.0	0.2	1.9	10.9	0.0	0.0	0.1	0.5	3.5	0.0	0.0	0.0	0.0	0.2
	$[0.10, 0.30)$	1.1	4.8	16.1	40.1	63.1	1.3	4.6	13.8	32.0	59.0	0.7	2.3	6.3	15.1	32.1
	$[0.30, 0.50)$	17.7	37.1	53.9	49.2	21.1	27.6	47.4	62.3	59.1	35.5	31.9	49.3	65.2	71.8	63.6
	$[0.50, 1.00)$	81.2	58.1	29.8	8.6	1.0	71.2	48.0	23.9	8.3	1.5	67.3	48.4	28.5	13.1	4.2

Table 1: Synthetic data: subspace intersection with neighboring samples

hence, the set $\{X_k\}$ is the unique minimal segmentation, and we also have $\mathcal{S}_k = \text{span}(X_k)$ for each k.

There are two kinds of complexities associated with subspace segmentation. One is the degree of subspace intersection. We define this degree by the ratios $\{d_k/r\}$. Generally, a smaller d_k implies better sample construction in this way if r is fixed. The other kind of complexity involves the existence of samples near another subspace and the percentage of samples within a short distance. The minimal distance of a sample $x_i \in \mathcal{S}_k$ to other subspaces is measured by

$$\text{dist}(x_i) = \min_{\ell \neq k} \text{dist}(x, \mathcal{S}_\ell) = \min_{\ell \neq k} \frac{\|x_i - U_\ell(U_\ell^T x_i)\|}{\|x_i\|}.$$

We estimate the dimension of $\mathcal{S}_s \cap \mathcal{S}_t$ as the number of singular values of $U_s^T U_t$ satisfying $\sigma_i(U_s^T U_t) \geq \tau_\sigma$ for a $\tau_\sigma \approx 1$, which takes into account the proximity of two subspaces, excluding the intersection. In the top block of Table 1, we list the average dimensions of the pairwise-intersected subspaces of $K = 5$ minimal sample subspaces with an equal dimension d_c for 100 repeated experiments. The average dimension of the pair-wise subspace intersection is approximately $(2d_c - r)_+$. We test three values of r and, for each r, five values of d_c. In the bottom block, we also list the percentage of samples with the distances dist(x_i) in a given interval. Each time, we randomly choose $n_k = 1000$ samples in each subspace. The quality of the samples improves as r increases if the dimension of the intersected subspaces is approximately fixed.

We run the MSS algorithms with MCG using Ω_c only (MCG(c)), the alternative optimization (AO) or the hybrid optimization (HO) on the synthetic data sets with the same settings of r and d_c as shown in Table 1. To reduce the computation time, we choose $n_k = 50$ samples in each subspace and repeat the experiments 10 times for a total of 150 tests. For each r, the MSS algorithms succeed in recovering the true minimal sample subspaces on all the tests with four smaller values of d_c, but fail when d_c is set to be the largest value. We choose a large d_c for each r to show the case when the MSS algorithms cannot retrieve the minimal subspaces. In approximately 29% of the 120 successful tests of the MSS algorithms, the MCG(c) method successfully gives a true solution where further alternative or hybrid optimization is not required. The MCG(c) solutions for the remaining tests are also suitable as initial guesses for further alternative optimization, and 22% of them can be further improved to retrieve the minimal subspaces using the AO strategy within at most 5

Stage	Convergence behavior in the successful tests		
MCG(c)	35 succeed	85 fail	
AO	at most 5 iterations	19 succeed	66 fail
HO	at most 5 iterations		66 succeed

Table 2: Convergence behavior of the MSS algorithms

AO steps. The other testings require the HO strategy to eventually obtain the true solution within at most 5 HO steps. Table 2 shows the convergence behavior of the MSS algorithms with the three strategies MCG(c), AO, and HO.

Table 3 further characterizes the behaviur of the MSS method with the different strategies MCG(c), AO, and HO, measured by the average values of ErrParti, BdiagDevi, IntraBConn, and KblockDiag on 10 repeated tests. As in previous experiments, the strategies AO and HO improve the connection and estimation accuracy of the K-block structure on both the minimal partition and MSDR. The error functions ErrParti and BdiagDevi decrease quickly, while IntraBConn and KblockDiag increase in most cases when the subspaces are not overly intersected with each other. In the case with the largest d_c for $r = 10$, 14, or 20, the local minimizer of MSS_MCG(c) is far from the ideal solution, due to heavy subspace intersection. The average dimension of the subspace intersections is large compared with the dimension of each subspace. The relative dimensions of the intersected subspaces are approximately 75%, 72%, and 67% of the dimension of the sample space for $r = 10, 14, 20$, respectively. The local optimal solution of MSS_MCG(c) cannot serve as a good initial guess for the MSS_AO or MSS_HO.

In Table 3, we also compare the MSS method with four other algorithms, LRR, SSC, SoftS3C, and LRSSC, on the same data. LRR gives relatively strong intra-block connection but very poor K-block-diagonal structure and large approximation errors to minimal partitions or MSDRs. SoftS3C slightly improves upon SSC based on the four measurements, and LRSSC performs better than LRR, SSC, and SoftS3C on the approximation of K-block-diagonal form, but weakens the connections within diagonal blocks. If the subspace intersection is very weak, the solutions of SSC, SoftS3C, and LRSSC approximate the minimal partitions well, but the block-diagonal structures are unclear since the solutions have large values of BdiagDevi and relatively small values of KblockDiag. The connections within diagonal blocks are also relatively weak. When r is relatively large or the subspace intersection slightly increases, LRR, SSC, SoftS3C and LRSSC still fail to give acceptable solutions. We note that in the noiseless case, the objective function of CLAR is equal to $\log\det(I + C^T C) = \sum_{i=1}^n \log\left(1 + \lambda_i(VV^T + H^T H)\right)$ by Lemma 19. It is obvious that CLAR gives the same solution with $H = 0$ as LRR in the noiseless case. Hence, we omit a comparison with CLAR here.

8.3. Real-world Data

We use the benchmark databases Extended YaleB for face clustering (Georghiades et al., 2001) and Hopkin 155 for motion segmentation (Tron and Vidal, 2007) to evaluate the performance of Algorithm 7 (MSS_RO) on noisy data. Because ignorable noise exists in real-world data, segmentation is obtained by applying spectral clustering on a constructed

	ErrParti					BdiagDevi					IntraBConn					KblockDiag				
$r = 10,\ d_c =$	4	5	6	7	8	4	5	6	7	8	4	5	6	7	8	4	5	6	7	8
MSS_MCG(c)	0.00	0.00	0.02	0.27	0.60	0.02	0.11	0.37	0.66	0.74	0.57	0.58	0.54	0.43	0.37	0.92	0.78	0.39	0.04	0.01
MSS_AO	0.00	0.00	0.01	0.16	0.60	0.00	0.00	0.02	0.30	0.72	0.58	0.58	0.57	0.44	0.24	1.00	0.99	0.97	0.79	0.41
MSS_HO	0.00	0.00	0.00	0.00	0.72	0.00	0.00	0.00	0.01	0.65	0.58	0.58	0.58	0.57	0.27	1.00	1.00	1.00	0.99	0.45
LRR	0.14	0.27	0.50	0.59	0.67	0.65	0.70	0.73	0.76	0.78	0.57	0.57	0.57	0.57	0.57	0.04	0.02	0.01	0.01	0.01
SSC	0.00	0.01	0.09	0.39	0.62	0.08	0.25	0.42	0.55	0.65	0.11	0.11	0.09	0.08	0.07	0.65	0.31	0.10	0.03	0.01
SoftS3C	0.00	0.01	0.08	0.37	0.61	0.11	0.26	0.41	0.53	0.65	0.14	0.12	0.10	0.08	0.07	0.59	0.32	0.12	0.05	0.03
LRSSC	0.00	0.01	0.06	0.30	0.62	0.02	0.12	0.31	0.47	0.61	0.05	0.05	0.05	0.04	0.03	0.85	0.46	0.14	0.03	0.02
$r = 14,\ d_c =$	7	8	9	10	11	7	8	9	10	11	7	8	9	10	11	7	8	9	10	11
MSS_MCG(c)	0.00	0.00	0.09	0.35	0.59	0.22	0.41	0.62	0.70	0.74	0.57	0.54	0.46	0.40	0.37	0.63	0.36	0.09	0.02	0.01
MSS_AO	0.00	0.00	0.01	0.20	0.56	0.00	0.00	0.03	0.40	0.70	0.57	0.57	0.56	0.44	0.31	1.00	1.00	0.95	0.59	0.37
MSS_HO	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00	0.03	0.66	0.57	0.57	0.57	0.57	0.32	1.00	1.00	1.00	0.94	0.37
LRR	0.08	0.29	0.45	0.56	0.64	0.70	0.72	0.74	0.76	0.77	0.57	0.57	0.57	0.57	0.57	0.02	0.01	0.01	0.00	0.00
SSC	0.00	0.02	0.13	0.42	0.59	0.29	0.41	0.51	0.60	0.66	0.12	0.11	0.10	0.09	0.08	0.33	0.16	0.05	0.02	0.01
SoftS3C	0.00	0.03	0.12	0.38	0.61	0.31	0.41	0.50	0.59	0.65	0.14	0.12	0.11	0.09	0.08	0.32	0.18	0.08	0.04	0.03
LRSSC	0.00	0.02	0.14	0.46	0.61	0.15	0.30	0.44	0.55	0.63	0.07	0.07	0.06	0.05	0.05	0.51	0.21	0.06	0.02	0.01
$r = 20,\ d_c =$	11	12	13	14	15	11	12	13	14	15	11	12	13	14	15	11	12	13	14	15
MSS_MCG(c)	0.01	0.04	0.14	0.30	0.51	0.55	0.64	0.68	0.71	0.73	0.49	0.44	0.41	0.38	0.37	0.19	0.08	0.04	0.01	0.01
MSS_AO	0.00	0.00	0.02	0.15	0.43	0.00	0.02	0.05	0.50	0.68	0.56	0.56	0.55	0.47	0.38	0.99	0.97	0.92	0.34	0.24
MSS_HO	0.00	0.00	0.00	0.00	0.49	0.00	0.00	0.00	0.24	0.59	0.56	0.56	0.56	0.56	0.43	1.00	1.00	1.00	0.63	0.24
LRR	0.11	0.18	0.31	0.51	0.58	0.72	0.73	0.75	0.76	0.77	0.56	0.56	0.56	0.56	0.56	0.02	0.01	0.01	0.01	0.00
SSC	0.01	0.04	0.12	0.26	0.50	0.43	0.50	0.56	0.61	0.65	0.13	0.12	0.12	0.11	0.10	0.17	0.11	0.05	0.02	0.01
SoftS3C	0.01	0.04	0.11	0.24	0.47	0.43	0.50	0.55	0.60	0.65	0.15	0.14	0.13	0.12	0.11	0.19	0.13	0.08	0.05	0.03
LRSSC	0.01	0.05	0.13	0.30	0.52	0.33	0.44	0.51	0.57	0.62	0.09	0.08	0.08	0.07	0.07	0.27	0.14	0.05	0.03	0.01

Table 3: Results of the four measurements on the solutions of the compared algorithms

graph A, rather than the solution C itself. Special postprocessing of the computed solution C might be required for constructing the graph A, which aims to strengthen the block-diagonal structure of C or increase class similarities. SSC and LRR/CLAR adopt different postprocessing approaches as shown below, while SoftS3C uses $A = (|C| + |C^T|)/2$.

The postprocessing of a solution C for SSC cuts of the small entries of C in absolute value to obtain a sparser C_γ such that each column has the norm $\|C_\gamma e_j\| \approx \gamma \|Ce_j\|$ with a given positive γ close to 1. Each column of C_γ should be further normalized to have the largest entry equal to one in absolute value. Let $\tilde{C}_\gamma$ be the normalized C_γ. Then, graph A is set as the symmetric part of $\tilde{C}_\gamma$, or equivalently, $A = |\tilde{C}_\gamma| + |\tilde{C}_\gamma^T|$. In the experiments on the two databases, we set $\gamma = 1$ for Extended YaleB and $\gamma = 0.7$ for Hopkins155 as suggested by SSC.

LRR or CLAR truncates the SVD of C to $U_\sigma \Sigma_\sigma V_\sigma^T$ by removing the singular values smaller than threshold σ and the corresponding singular vectors. Then, the rows $\{u_i\}$ of $U_\sigma \Sigma_\sigma^{1/2}$ are used to construct the graph $A = (a_{ij})$ as per $a_{ij} = (\frac{\langle u_i, u_j \rangle}{\|u_i\| \|u_j\|})^s$ with $s \geq 1$. We set $s = 4$, $\sigma = 10^{-4}\|C\|_2$ for LRR and $s = 4$, $\sigma = 10^{-6}$ for CLAR, as suggested by their authors.

We use the strategy of double truncating to construct the graph shown in Algorithm 6 with the same settings $\gamma = 1$, $\sigma = 0$, and $s = 1$ for Extended YaleB, and $\gamma = 0.8$, $\sigma = 0.001$, and $s = 4$ for Hopkins155. Notably, the strategy of double truncating does not work well for SSC or LRR/CLAR. Since no suggestions about graph constructing were given for LRSSC in the literature, We test the three approaches adopted in SSC, LRR, or our method for

Algorithm	$K = 2$				$K = 3$				$K = 5$				$K = 8$				$K = 10$			
	I_0	I_1	I_2	I_3	I_0	I_1	I_2	I_3	I_0	I_1	I_2	I_3	I_0	I_1	I_2	I_3	I_0	I_1	I_2	I_3
LRR	39.3	13.5	41.7	5.5	9.1	13.0	71.2	6.7	1.5	6.3	74.1	18.1	0.0	0.0	55.1	44.9	0.0	0.0	33.3	66.7
CLAR	44.2	24.5	31.3	0.0	16.6	17.1	66.3	0.0	6.8	14.4	78.8	0.0	0.7	6.6	92.6	0.0	0.0	0.0	100.0	0.0
SSC	58.3	21.5	16.6	3.7	27.6	22.8	42.1	7.5	5.5	18.6	63.2	12.7	0.0	4.4	76.5	19.1	0.0	0.0	66.7	33.3
SoftS3C	67.5	20.2	11.7	0.6	39.9	24.0	36.1	0.0	20.9	24.9	50.0	4.2	2.2	10.3	66.9	20.6	0.0	0.0	66.7	33.3
mLRSSC	73.6	15.3	11.0	0.0	46.4	23.8	28.8	1.0	28.3	20.4	45.9	5.3	11.0	8.1	57.4	23.5	0.0	0.0	66.7	33.3
MSS_RO(c)	82.2	12.9	4.9	0.0	63.0	16.6	18.8	1.7	53.3	20.6	22.8	3.3	28.7	28.7	38.2	4.4	0.0	33.3	66.7	0.0
MSS_RO	82.2	12.9	4.9	0.0	63.7	18.5	16.1	1.7	57.1	25.0	15.5	2.3	44.1	19.1	35.3	1.5	33.3	0.0	66.7	0.0

Table 4: Percentage (%) of computed segmentations with errors belonging to each of the intervals $I_0 = [0, 0.005]$, $I_1 = (0.005, 0.01]$, $I_2 = (0.01, 0.1]$, and $I_3 = (0.1, 0.5]$.

LRSSC, including its two modified versions mLRSSC and aLRSSC that will be mentioned latter, and report the best results.

8.3.1. Facial Image Clustering

The Extended YaleB database consists of 2432 facial images (192×168 pixels) from 38 individuals under 64 illumination conditions. Due to various illumination and shadow conditions, these images have a relatively large amount of noise and corruption. It makes sense to use images from the same individual as a groundtruth class, and such images in each class are sampled from the same subspace approximately. It was pointed out by Basri and Jacobs (2003) that a facial image lives in a 9 dimensional subspace. Thus, each of the subspaces has dimension 9, and we estimate $d = 9K$ if we have images from K individuals.

The testing sets are chosen as follows. A total of 38 individuals are divided into 4 groups—each of the first three groups contains 10 individuals and the last group contains the remaining 8 individuals. In each group, we choose K individuals and test the segmentation of all $64K$ images from the chosen individuals. Since there are C_p^K combinations of K individuals among p individuals, we have $3C_{10}^K + C_8^K$ tests for a fixed K. Since the five values of K are set as 2, 3, 5, 8, and 10, we have 163, 416, 812, 136, and 3 tests for the five settings of K, respectively, and we have 1530 tests using different sizes of sample sets for this database. We downsample the large images to 48×42 pixels and vectorize them as 2016-dimensional vectors $\{x_j\}$ to obtain a reasonable computational complexity, as done by Elhamifar and Vidal (2013). The samples are normalized to have the unit norms $\|x_j\|_2 = 1$. The self-expressive error is measured by the ℓ_1-norm $\phi = \|\cdot\|_1$. We set the parameters $\lambda = 5, \alpha = \frac{20}{\|X\|_1^*}$, and $\beta = 5$, where $\|X\|_1^* = \max_j \|x_j\|_1$. The strategy for setting α is the same as that used for SSC and SoftS3C.

Due to the existence of noise, the six algorithms cannot correctly retrieve the true classes completely. To show the efficiency of these algorithms, we separate the computed clustering into four groups according to the partition error ErrParti belonging to one of four intervals $I_0 = [0, 0.005]$, $I_1 = (0.005, 0.01]$, $I_2 = (0.01, 0.1]$, and $I_3 = (0.1, 0.5]$. We point out that LRSSC adopts $\|X - XC\|_F^2$ as a penalty term and imposes the zero restriction on diagonals as in SSC for combining LRR with SSC. Such a combination cannot give better results than LRR or SSC on this data set. We modify LRSSC by changing the Frobenius norm (ℓ_2-norm) penalty to the ℓ_1-norm penalty $\|X - XC\|_1$. The modified LRSSC (marked as

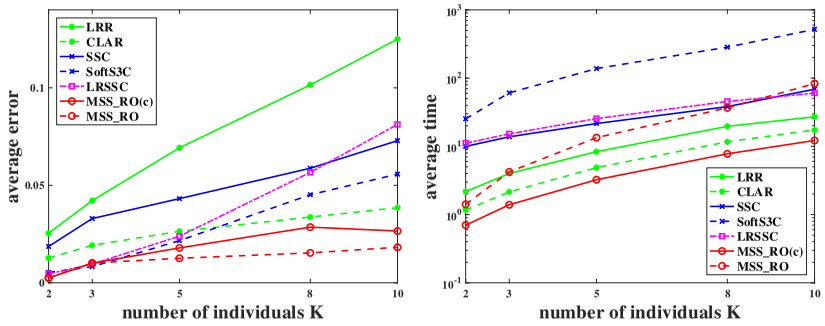


Figure 3: Percentage of average clustering errors (left) and average computation time in seconds (right) of the algorithms for detecting facial images of K individuals with $K = 2, 3, 5, 8, 10$, respectively.

mLRSSC) performs better than LRR, SSC, and CLAR, as shown in Table 4. When $K = 2$, mLRSSC can correctly detect the two individuals in approximately 2/3 tests. When K is slightly increased to 3, its detection percentage decreases to 39.9%, which is also higher than that of the other four methods. When more individuals are detected, the detecting error, that is, the partition error, quickly enlarges. For example, the values of ErrParti are greater than 0.01 for all the solutions of LRR, SSC, mLRSSC, SoftS3C and CLAR when $K = 10$. Our MSS_RO(c) and MSS_RO algorithms perform very well in this experiment. Essentially, the MSS_RO(c) method provides a better solution than those of LRR, SSC, mLRSSC, SoftS3C and CLAR in most cases, especially with large K. The percentage of MSS_RO(c) segmentations with errors in I_0 are 82.2%, 63.0% and 53.3% for $K = 2$, 3, or 5 individuals, respectively, which is much higher than those of LRR, SSC, mLRSSC, SoftS3C and CLAR. Even in the more complicated case for detecting $K = 10$ individuals, 1/3 of the tests achieve a detection error less than 0.01. Within at most two updates of Ω ($\ell_{\max} \leq 3$), MSS_RO presents its final results in this example.

The computational cost of MSS_RO is also competitive. MSS_RO(c) is faster than the other algorithms in the five cases for variant K, without updating the coarsest Ω_c. If Ω is updated within several iterations, the computation time of MSS_RO increases but remains smaller than SoftS3C. On the right side of Figure 3, we plot the computation time in seconds. We also plot the average clustering error with the same number of individuals on the left and do not separate the solutions into several groups as before.[11]

8.3.2. Motion Segmentation

The Hopkins 155 database contains 155 videos of rigidly moving objects in which 120 videos have two objects and 35 videos have three objects. To separate the objects in each video,

11. The results of SoftS3C for $K = 5, 8, 10$ reported in Li et al. (2017) are somehow better than those we obtained using the codes provided by the authors, but still worse than the results of MSS_RO.

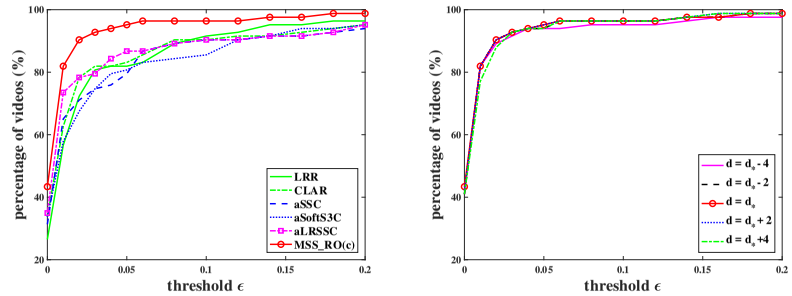


Figure 4: Percentage of videos whose detection error less than given threshold ϵ in the set of 83 'hard' videos. The left one compares different methods, where correct d is set for MSS_RO. The right one shows the sensitivity of MSS_RO to d that varies around the correct sum of subspace dimensions d_*.

the feature points (2D coordinates) of the objects are first extracted from the frames of each video. The 2D coordinates of the same feature point in the sequence of frames form a long sample vector of length $2F$, where F is the number of frames in the video. Because the objects are rigidly moving, the $2F$-dimensional vectors $\{y_i\}$ of the same object belong to an affine subspace with a dimension of at most 3, as pointed out by Costeira and Kanade (1998). The affine data set $\{y_i\}$ corresponding to an object can be modified to be a subspace data set $\{x_i\}$, where each x_i has an additional constant entry 1. The objects in a video are detected by segmenting $\{x_i\}$. Since each of the linear subspaces has a dimension of at most 4, we set $d = 4K$ if the number of objects (subspaces) is K. In this data set, $K = 2$ or $K = 3$. SSC and SoftS3C also have affine versions (termed aSSC or aSoftS3C, respectively) in which the sum of the entries in each column of C is restricted to be one. We report the results of these affine versions on $\{y_i\}$ because of their improved performance over their original versions on $\{x_i\}$. LRR, CLAR, and the noise-handling version MMS_RO of MSS work on $\{x_i\}$. Similar to the original SSC/SoftS3C, LRSSC does not work well on $\{x_i\}$. We modify LRSSC again (termed aLRSSC) by adding a restriction on C as in SSC/SoftS3C so that it can works on the affine data $\{y_i\}$ and gives better results similar to the affine versions of SSC/SoftS3C. In our algorithm MSS_RO, we use $\phi = \|\cdot\|_F^2$ for measuring the self-expressive error, and set the parameters

$$\lambda = 10n/K, \ \alpha = 50n/K, \ \beta = 0.05n/K, \tag{61}$$

where n is the number of sample vectors, which vary from 39 to 556 in this experiment.

In this data set, there are 72 videos whose objects are easily detected—all of the compared algorithms can correctly detect all the objects. The remaining 83 videos are relatively hard to detect—no one can be completely detected by all the compared algorithms. In this experiment, we focus on performances of these algorithms on these 83 challenging videos, by checking the distribution of the detecting errors. The left panel of Figure 4 plots the

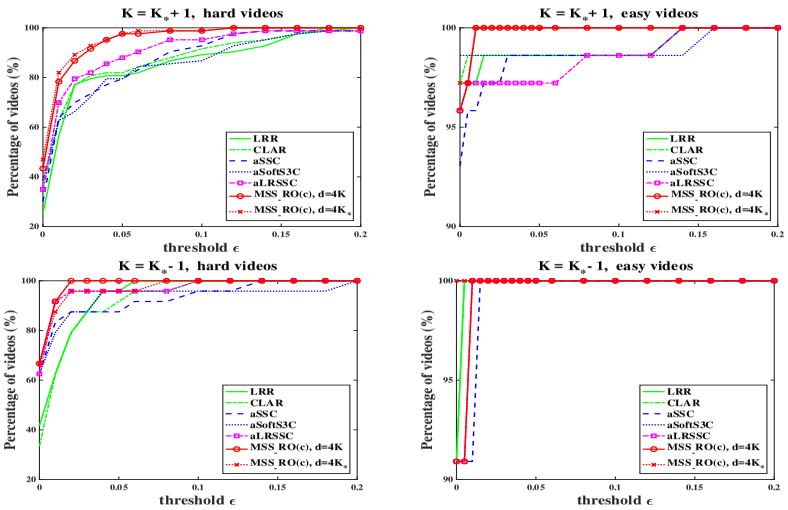


Figure 5: Percentage of videos with dominance rate larger than $1-\epsilon$ in the set of 83 'hard' videos (left) or in the set of 72 'easy' videos (right) when K is chosen as $K = K_*+1$ (top) or $K = K_* - 1$ (bottom, for videos with $K_* = 3$ objects only).

percentage of videos whose detecting errors are not larger than the threshold value ϵ for each of the six algorithms LRR, CLAR, aSSC, aSoftS3C, aLRSSC, and MSS_RO. The LRR and CLAR solutions have detection errors relatively smaller than those of aSSC and aSoftSSC. Our LRSSC modification causes it to perform slightly better than aSSC. MSS_RO(c) still significantly outperforms all of the compared algorithms.

8.4. Sensitivity

Compared with algorithms in the literature for subspace learning, our MSS method needs an estimated value d of the true sum d_* of subspace dimensions as an additional prior. For some real-world data sets like YaleB and Hopkins155, the correct d_* is practically known. In this subsection, we numerically show the sensitivity of MSS_RO when it uses estimated d and K that slightly vary around the correct ones.

At first, we set $d = d_* \pm \delta$, $\delta = 0, 2, 4$, where $d_* = 4K_*$ and $K = K_*$ are the correct ones. The data set is the 83 challenging videos as in the last subsection. As in the left panel of Figure 4, we also plot the percentage curve versus the threshold ϵ for each d in the right panel. It seems that MSS_RO is robust on d in this data set. We also checked the sensitivity of MSS_RO on the set of other 72 videos. Only two videos are wrongly detected by MSS_RO.

Second, we slightly increase the subspace number K_* to $K = K_* + 1$ on all of the 155 videos. Two values are chosen for the dimension sum: $d = d_* = 4K_*$ or $d = 4K = d_* + 4$.[12] Since the number of subspaces is larger than the true one, some true subspaces may not be correctly segmented. In this example, more often than not, only a true subspace is split into two parts, each is learned as a subspace. That is, two pieces J_i and J_j of the learned partition are dominantly contained by a true J_k^*. To show this phenomenon, let $\rho_{k,i}$ be the number of samples belonging to the true set J_k^* that are detected as in J_i. The largest one $\rho_{k_i,i}^+ = \max_k \rho_{k,i}$ means that J_i is dominated by the members from the true subspace $\mathcal{S}_{k_i}$, and we can relabel each J_i to $\tilde{J}_{k_i}$ if necessary. If for each $k \le K_*$, there is a $k_i = k$, then $\{\tilde{J}_k\}$ is a partition of K_* pieces, and the detection accuracy of $\{\tilde{J}_k\}$ is

$$\tau_+(J) = \frac{1}{n}\sum_{i=1}^{K} \rho_{k_i,i}^+.$$

We call $\tau_+(J)$ as the dominance rate of the original partition $J = \{J_i\}$ for $K > K_*$.

We also check the sensitivity when the number of subspace is mistakenly set as $K = K_* - 1$ on those 35 videos, each contains $K_* = 3$ subspaces.[13] Because the number of detected subspaces is less than the number of correct subspaces, it is absolute that some samples from different correct subspaces must be merged together. If we define $\rho_{k,i_k}^- = \max_i \rho_{k,i}$, then there are two equal $i = i_{k'} = i_{k''}$ with different k' and k''. That is, the two subspaces $\mathcal{S}_{k'}$ and $\mathcal{S}_{k''}$ may be merged together. One might split J_i into two parts by picking up the samples belonging to $J_{k''}^*$ to form a new piece $\tilde{J}_{K+1}$, while the remaining part and others have the same labels as $\tilde{J}_j = J_j$. If for each $i \le K$, there is an $i_k = i$, then the detection accuracy of $\{\tilde{J}_i\}$ is

$$\tau_-(J) = \frac{1}{n}\sum_{k=1}^{K_*} \rho_{k,i_k}^-.$$

So, similar with $\tau_+(J)$ for $K > K_*$, we also take $\tau_-(J)$ as the dominance rate of the original partition $J = \{J_i\}$ when $K < K_*$.

Figure 5 plots the percentage curve of videos on which the dominance rate larger than $1-\epsilon$ for each algorithm when K is mistakenly estimated. We also test MMS_RO with two choices for d: $d = 4K_*$ or $d = 4K$. MMS_RO is relatively stable on the estimation of K and d. Its dominance rate is larger than that obtained by other compared algorithms generally, not only on the set of the hard-to-detect videos but also on the set of easy-to-detect videos.

9. Conclusion

Subspace learning is a challenging task not only in theoretical analysis but also in modeling and computing for applications. In applications, ground-truth subspaces may be different from those that can be mathematically defined based on finite samples. Mathematically, these subspaces may be heavily intersected with each other, and some samples may be difficult to separated if they are proximal to the intersected subspaces. The existence of noise further complicates the problem. In this paper, we provided the concept of minimal

12. Same parameters are used in MSS_RO for different $d = d_*$ or $d = d_* + 4$.
13. For those videos with $K_* = 2$, setting $K = K_* - 1$ is meaningless.

sample subspace and considered the segmentation as a union of minimal sample subspaces, together with a so-called pure subspace that is mostly nonexistent in applications. However, even for the minimal subspace segmentation, which is now well-defined, the problem is also difficult and complicated since the MSS may be not unique. We gave sufficient conditions for addressing this uniqueness, and built some solid theoretical bases to support our proposed optimization modeling for conditionally recovering the minimal sample subspaces even if these subspaces themselves are heavily intersected. However, there are still some theoretical problems that need to be addressed, such as the sensitivity of the MSS problem. Perturbation analysis should be given to address the reliability of the retrieved MSS and also help us to understand segmentation on noisy samples. We consider this a difficult but interesting topic for further research.

We proposed several algorithms for solving the MSS problem with or without noise. However, because of the complexity of the optimization problems, a globally optimal solution is not always guaranteed, though we did endeavor to obtain a globally optimal solution of this nonconvex problem as much as possible. In our experiments, finding a local minimum seldom occurs when the subspace intersection is slight, but happens more frequently with increasing intersection. It is unclear how subspace intersection comes into the effect of local minima. The computational complexity of the basic algorithm MSS_MCG is $O(n^3)$. This disadvantage hinders its applications to large numbers of samples. Since the ideal solution is basically low-rank, sufficiently utilizing the low-rank structure may be useful for reducing the computational complexity of MSS_MCG. We will continue our work on this important topic.

Acknowledgments

Acknowledgment The work was supported in part by NSFC projects 11571312 and 91730303, and National Basic Research Program of China (973 Program) 2015CB352503.

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Siam Journal on Imaging Sciences*, 2(1):183–202, 2009.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Paul S. Bradley and Olvi L. Mangasarian. K-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

Joao Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11): 2765–2781, 2013.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

William W. Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005.

Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

Michel Journée, Francis Bach, P.-A. Absil, and Rodolphe Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

Zhao Kang, Chong Peng, and Qiang Cheng. Robust subspace clustering via smoothed rank approximation. *IEEE Signal Processing Letters*, 22(11):2088–2092, 2015.

Fabien Lauer and Christoph Schnorr. Spectral clustering of linear subspaces for motion segmentation. In *IEEE International Conference on Computer Vision*, pages 678–685, 2009.

Chunguang Li, Chong You, and René Vidal. Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework. *IEEE Transactions on Image Processing*, 26(6):2988–3001, 2017.

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, Jan 2013.

Canyi Lu, Jiashi Feng, Zhouchen Lin, Tao Mei, and Shuicheng Yan. Subspace clustering by block diagonal representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):487–501, Feb 2019.

Yi Ma, Allen Y. Yang, Harm Derksen, and Robert Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50 (3):413–458, 2008.

Behrooz Nasihatkon and Richard Hartley. Graph connectivity in sparse subspace clustering. In *Computer Vision and Pattern Recognition*, pages 2137–2144, 2011.

Mostafa Rahmani and George K. Atia. Innovation pursuit: a new approach to subspace clustering. *IEEE Transactions on Signal Processing*, 65(23):6276–6291, 2017.

R. Tyrrell Rockafellar and Roger J. B. Wets. *Variational Analysis*. Springer, 1998.

Hiroyuki Sato and Toshihiro Iwai. A new, globally convergent riemannian conjugate gradient method. *Optimization*, 64(4):1011–1031, 2015.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Mahdi Soltanolkotabi and Emmanuel J. Candès. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

G. W. Stewart and Jiguang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

Paul Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000.

René Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007.

Yuxiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: when LRR meets SSC. In *Advances in Neural Information Processing Systems*, pages 64–72, 2013.

Yuqing Xia and Zhenyue Zhang. Rank–sparsity balanced representation for subspace clustering. *Machine Vision and Applications*, 29(6):979–990, 2018.

Junfeng Yang, Wotao Yin, Yin Zhang, and Yilun Wang. A fast algorithm for edge-preserving variational multichannel image restoration. *Siam Journal on Imaging Sciences*, 2(2):569–592, 2011.

Journal of Machine Learning Research 20 (2019) 1-30 Submitted 3/18; Revised 9/19; Published 10/19

Convergence of Gaussian Belief Propagation Under General Pairwise Factorization: Connecting Gaussian MRF with Pairwise Linear Gaussian Model

Bin Li BINLI@EEE.HKU.HK

Yik-Chung Wu YCWU@EEE.HKU.HK

Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong

Editor: Gal Elidan

Abstract

Gaussian belief propagation (BP) is a low-complexity and distributed method for computing the marginal distributions of a high-dimensional joint Gaussian distribution. However, Gaussian BP is only guaranteed to converge in singly connected graphs and may fail to converge in loopy graphs. Therefore, convergence analysis is a core topic in Gaussian BP. Existing conditions for verifying the convergence of Gaussian BP are all tailored for one particular pairwise factorization of the distribution in Gaussian Markov random field (MRF) and may not be valid for another pairwise factorization. On the other hand, convergence conditions of Gaussian BP in pairwise linear Gaussian model are developed independently from those in Gaussian MRF, making the convergence results highly scattered with diverse settings. In this paper, the convergence condition of Gaussian BP is investigated under a general pairwise factorization, which includes Gaussian MRF and pairwise linear Gaussian model as special cases. Upon this, existing convergence conditions in Gaussian MRF are extended to any pairwise factorization. Moreover, the newly established link between Gaussian MRF and pairwise linear Gaussian model reveals an easily verifiable sufficient convergence condition in pairwise linear Gaussian model, which provides a unified criterion for assessing the convergence of Gaussian BP in multiple applications. Numerical examples are presented to corroborate the theoretical results of this paper.

Keywords: Gaussian belief propagation, convergence analysis, Gaussian Markov random field, pairwise linear Gaussian model, pairwise factorization

1. Introduction

Due to the distributed and parallel computation mechanisms, Gaussian belief propagation (BP) (Weiss and Freeman, 2000) is a low-complexity method for computing the marginal distributions of a high-dimensional joint Gaussian distribution, especially for those with a sparse information matrix. It is known that Gaussian BP is guaranteed to obtain the exact marginal distributions in singly connected or acyclic graphs. For loopy graphs (Kschischang et al., 2001), if Gaussian BP converges, the exact means of the marginal distributions can be obtained (Weiss and Freeman, 2000). These properties make Gaussian BP popular in many applications, ranging from consensus propagation (Moallemi and Roy, 2006), peer-ropeer rating (Bickson et al., 2007), multi-user data detection (Bickson et al., 2008), solving

©2019 Bin Li and Yik-Chung Wu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-133.html.

systems of linear equations (Shental et al., 2008), quadratic minimization (Moallemi and Roy, 2009; Ruozzi and Tatikonda, 2013), to network synchronization (Leng and Wu, 2011; Du and Wu, 2013).

Despite the apparent advantages and popularity, a major hurdle in applying Gaussian BP is that it may not converge in loopy graphs. Consequently, deriving Gaussian BP convergence conditions is one of the major themes in Gaussian BP research. For Gaussian BP running in Gaussian Markov random field (MRF) (Rue and Held, 2005), diagonal dominance (Weiss and Freeman, 2000), walk-summability (Malioutov et al., 2006), pairwise-normalizability (Malioutov et al., 2006) and convex decomposition (Moallemi and Roy, 2009) are well-known sufficient convergence conditions under both synchronous and totally asynchronous schedulings (Bertsekas and Tsitsiklis, 1989). For these sufficient conditions, belief means and variances converge upon the same condition, which has been proved to be unnecessary recently (Su and Wu, 2014, 2015a,b). In particular, the convergence condition of belief variances was proved to be looser than that of belief means (Su and Wu, 2015a). Due to the separate treatment of belief means and variances, the necessary and sufficient convergence condition of Gaussian BP under synchronous scheduling was proposed in Su and Wu (2015a).

On the other hand, under pairwise linear Gaussian model, the convergence of Gaussian BP has been investigated in a case-by-case basis (Moallemi and Roy, 2006; Bickson et al., 2007, 2008; Leng and Wu, 2011; Du and Wu, 2013). In Moallemi and Roy (2006), consensus propagation for distributed averaging was viewed as an asynchronous implementation of Gaussian BP, where the convergence was shown to be guaranteed. In Bickson et al. (2007), Gaussian BP is used to solve the minimization of a quadratic cost function in peer-to-peer rating, where the convergence of Gaussian BP is shown by numerical results. In Bickson et al. (2008), multiuser detection problem is solved by Gaussian BP, where the convergence is checked by the diagonal dominance in the equivalent information matrix. Furthermore, in distributed synchronization applications (Leng and Wu, 2011; Du and Wu, 2013), Gaussian BP is guaranteed to converge under a reference node with perfect prior information.

Obviously, the understanding on the convergence conditions of Gaussian BP is far from complete, as the current results are scattered and derived under different assumptions. For example, existing convergence conditions for Gaussian MRF are derived under different pairwise factorizations of the joint Gaussian distribution. Since different factorizations lead to different Gaussian BP messages, it is not clear if the convergence condition obtained under one pairwise factorization is valid under another pairwise factorization. Another missing link is between the convergence results in Gaussian MRF and that of pairwise linear Gaussian model. On one hand, Bickson et al. (2008) advocate transforming the linear Gaussian model into Gaussian MRF, and then leveraging on the existing convergence condition for determining the convergence of Gaussian BP. On the other hand, in Moallemi and Roy (2006), Leng and Wu (2011), and Du and Wu (2013), Gaussian BP was proved to be convergent without using existing sufficient convergence conditions from Gaussian MRF. One may wonder if there is any special characteristic of pairwise linear Gaussian model that facilitates the Gaussian BP convergence.

To provide a more complete picture, the convergence of Gaussian BP is revisited but starting from a general pairwise factorization of the joint Gaussian distribution. This general pairwise factorization covers all possible pairwise factorizations with the exponents of

factors being quadratic polynomial, and it includes all the existing pairwise factorizations used in Gaussian MRF and pairwise linear Gaussian model as special cases. By establishing the convergence conditions under this general pairwise factorization, existing convergence conditions can be compared and contrasted in a unified way. It is found that existing sufficient convergence conditions derived for Gaussian MRF (including walk-summability, pairwise-normalizability, convex decomposition, and diagonal dominance) are valid under any factorization within the general pairwise factorization model, and the converged beliefs are independent of the choice of pairwise factorization. Moreover, due to the newly established bridge between Gaussian MRF and pairwise linear Gaussian model under the generalized setting, the pairwise-normalizability from Gaussian MRF is further applied to pairwise linear Gaussian model, revealing an easily verifiable sufficient convergence condition in pairwise linear Gaussian model. This sufficient convergence condition is more general than the convergence conditions in Moallemi and Roy (2006), Leng and Wu (2011), and Du and Wu (2013), and further explains the empirical convergence behavior of peer-to-peer rating application (Bickson et al., 2007). Numerical results and applications are presented to corroborate the theoretical convergence results.

Notations: Scalars, vectors, matrices and sets are denoted by lower-case letters, bold lower-case letters, bold upper-case letters and calligraphic upper-case letters, respectively. Notations $\mathbf{a} > \mathbf{b}$, $\mathbf{a} \geq \mathbf{b}$, $\mathbf{a} < \mathbf{b}$, and $\mathbf{a} \leq \mathbf{b}$ indicate $a_i > b_i$, $a_i \geq b_i$, $a_i < b_i$, and $a_i \leq b_i$ for all i, respectively, where a_i and b_i are the i-th element of $\mathbf{a}$ and $\mathbf{b}$. For a matrix, $\mathbf{A} \succ \mathbf{0}$ indicates that $\mathbf{A}$ is positive definite, $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$, and $\mathbf{A}^{-1}$ denotes the inverse of $\mathbf{A}$. The notation $|\mathbf{A}|$ is a matrix with element-wise absolute value of $\mathbf{A}$, and $\rho(\mathbf{A})$ is the spectral radius of $\mathbf{A}$. Moreover, $\mathrm{diag}(\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_n)$ denotes a block diagonal matrix with the diagonal blocks being $\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_n$, starting from the upper left corner. For the notation $\mathcal{B}$, it denotes a set, and $|\mathcal{B}|$ denotes the number of elements in $\mathcal{B}$. The notation $\mathcal{B} \setminus i$ denotes all the elements in set $\mathcal{B}$ except i, and the notation $\mathcal{B} \setminus \mathcal{C}$ denotes all the elements in set $\mathcal{B}$ but not in $\mathcal{C}$. The notation $a \propto b$ denotes that a is proportional to b. For a Gaussian distributed random variable x with mean m and variance v, we denote it as $\mathcal{N}(x; m, v)$. Similarly, for a multivariate Gaussian distributed random vector $\mathbf{x}$ with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{V}$, we denote it as $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{V})$.

2. Gaussian Model and Belief Propagation

In this section, we first discuss the pairwise factorizations of Gaussian model (Section 2.1). Then, BP algorithm under a general pairwise factorization of Gaussian model is presented (Section 2.2).

2.1. Gaussian Model and Its Pairwise Factorizations

Consider the joint Gaussian probability density function (pdf) $p(\mathbf{x})$ of a random vector $\mathbf{x} \triangleq [x_1, x_2, \cdots, x_n]^T$ written in a pairwise factorization form

$$\mathcal{PFF}: \; p(\mathbf{x}) \propto \prod_i f_i(x_i) \prod_{i,j>i} f_{ij}(x_i, x_j), \tag{1}$$

where $f_i(x_i)$ is a local function of x_i while $f_{ij}(x_i, x_j)$ is a local function modeling the interaction between x_i and x_j. In general, given a joint Gaussian distribution[1] $\mathcal{N}(\mathbf{x}; \mathbf{J}^{-1}\mathbf{h}, \mathbf{J}^{-1})$ in Gaussian MRF, the $\mathcal{PFF}$ is not unique. For example, one of the earliest pairwise factorizations in Gaussian MRF is given in Weiss and Freeman (2000) with

$$\mathbb{F}_1: \; f_i(x_i) \propto \exp\left\{-\frac{1}{2}p_i(x_i - \frac{h_i}{p_i})^2\right\}, \tag{2a}$$

$$f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{1}{2}[x_i \; x_j]\begin{bmatrix} a_{ij} & J_{ij} \\ J_{ij} & a_{ji}\end{bmatrix}[x_i \; x_j]^T\right\}, \tag{2b}$$

$$\text{s.t. } \mathbf{J} \succ \mathbf{0}, \; p_i + \sum_{k \in \mathcal{B}_i} a_{ik} = J_{ii}, \; \mathcal{B}_i \triangleq \{k \mid J_{ik} \neq 0, k = 1, 2, \cdots, n\}, \tag{2c}$$

where J_{ik} is the (i,k)-th element of $\mathbf{J}$ and h_i is the i-th element of $\mathbf{h}$. On the other hand, the convex decomposition in Moallemi and Roy (2009) requires

$$\mathbb{F}_2: \; f_i(x_i) \propto \exp\left\{-\frac{1}{2}(J_{ii} - \sum_{k \in \mathcal{B}_i} \xi_{ik} J_{ik}^2)x_i^2 + (h_i - \sum_{k \in \mathcal{B}_i} \varsigma_{ik})x_i\right\}, \tag{3a}$$

$$f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{1}{2}\xi_{ij}J_{ij}^2 x_i^2 - J_{ij}x_i x_j - \frac{1}{2}\xi_{ji}J_{ij}^2 x_j^2 + \varsigma_{ij}x_i + \varsigma_{ji}x_j\right\}, \tag{3b}$$

$$\text{s.t. } \mathbf{J} \succ \mathbf{0}, \; \xi_{ij}, \xi_{ji} > 0, \; J_{ii} - \sum_{k \in \mathcal{B}_i} \xi_{ik} J_{ik}^2 > 0, \; \xi_{ij}\xi_{ji}J_{ij}^2 \geq 1. \tag{3c}$$

However, given an information matrix $\mathbf{J}$, the conditions in $\mathbb{F}_2$ might not be satisfied, thus this decomposition may not exist. Consequently, a more popular factorization is given by

$$\mathbb{F}_3: \; f_i(x_i) \propto \exp\left\{-\frac{1}{2}J_{ii}x_i^2 + h_i x_i\right\}, \tag{4a}$$

$$f_{ij}(x_i, x_j) \propto \exp\left\{-J_{ij}x_i x_j\right\}, \tag{4b}$$

$$\text{s.t. } \mathbf{J} \succ \mathbf{0}. \tag{4c}$$

This factorization directly connects the joint distribution without any constraints and is used in Malioutov et al. (2006), and Su and Wu (2014, 2015a,b).

On the other hand, the $\mathcal{PFF}$ also covers the pairwise linear Gaussian model that frequently appears in distributed inference. In particular, assume there exists a local relationship $z_{ij} = c_{ij}x_i + c_{ji}x_j + n_{ij}$ with known coefficients c_{ij}, c_{ji}. If $n_{ij} \sim \mathcal{N}(n_{ij}; 0, 1/\eta_{ij})$ and the prior distribution of x_i is $p(x_i) = \mathcal{N}(x_i; y_i, 1/\zeta_i)$, then we have the joint posterior distribution given by the $\mathcal{PFF}$ with

$$\mathbb{F}_4: \; f_i(x_i) \propto \exp\left\{-\frac{1}{2}\zeta_i(x_i - y_i)^2\right\}, \tag{5a}$$

$$f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{1}{2}\eta_{ij}[z_{ij} - (c_{ij}x_i + c_{ji}x_j)]^2\right\}, \tag{5b}$$

$$\text{s.t. } \mathrm{diag}(\zeta_1, \zeta_2, \cdots, \zeta_n) + \mathbf{C} \succ \mathbf{0}, \; \zeta_i \geq 0, \; \eta_{ij} > 0, \; \mathbf{C} \in \mathbb{R}^{n \times n}$$
$$\text{with } C_{ii} = \sum\nolimits_{k \in \mathcal{B}_i} \eta_{ik}c_{ik}^2 \text{ and } C_{ij} = \eta_{ij}c_{ij}c_{ji}, \tag{5c}$$

1. The Gaussian distribution is expressed in an information form, where $\mathbf{J}$ is the information matrix and $\mathbf{h}$ is the potential vector.

where $\mathrm{diag}(\zeta_1, \zeta_2, \cdots, \zeta_n) + \mathbf{C} \succ \mathbf{0}$ is a condition guaranteeing the $\mathcal{PFF}$ with factors in $\mathbb{F}_4$ represents a valid Gaussian pdf (in fact, $\mathrm{diag}(\zeta_1, \zeta_2, \cdots, \zeta_n) + \mathbf{C}$ is the information matrix of such Gaussian pdf). This model handles a broad class of distributed estimation applications. In particular, $\mathbb{F}_4$ is exactly the model in distributed clock synchronization (Leng and Wu, 2011) and distributed carrier frequency estimation (Du and Wu, 2013). On the other hand, if $z_{ij} = 0$, $c_{ij} = -c_{ji} = 1$, and y_i is obtained from local observation, $\mathbb{F}_4$ reduces to the model used in consensus propagation (Moallemi and Roy, 2006) and peer-to-peer rating (Bickson et al., 2007).

Currently, different convergence conditions of Gaussian BP are developed under different factorizations. For example, diagonal dominance (Weiss and Freeman, 2000) is developed from $\mathbb{F}_1$, convex decomposition (Moallemi and Roy, 2009) is developed from $\mathbb{F}_2$, walk-summability (Malioutov et al., 2006) and the recent necessary and sufficient condition (Su and Wu, 2014, 2015a) are developed from $\mathbb{F}_3$. One may wonder if the convergence condition of Gaussian BP developed from one factorization is applicable to another factorization. More importantly, for different pairwise factorizations, can we develop a unified treatment in the convergence analysis? In order to answer these questions, we write the general forms of factors $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ in the $\mathcal{PFF}$ as

$$\mathbb{F}_5: \ f_i(x_i) \propto \exp\left\{-\frac{1}{2}\phi_i x_i^2 + \psi_i x_i\right\}, \tag{6a}$$

$$f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{1}{2}\gamma_{ij}x_i^2 - \frac{1}{2}\gamma_{ji}x_j^2 - \tau_{ij}x_i x_j + \kappa_{ij}x_i + \kappa_{ji}x_j\right\}, \tag{6b}$$

$$\text{s.t. } \tau_{ij} = \tau_{ji} \text{ and } \begin{bmatrix} \phi_1 + \sum_{k\in\mathcal{B}_1}\gamma_{1k} & \tau_{12} & \cdots & \tau_{1n} \\ \tau_{21} & \phi_2 + \sum_{k\in\mathcal{B}_2}\gamma_{2k} & \cdots & \tau_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tau_{n1} & \tau_{n2} & \cdots & \phi_n + \sum_{k\in\mathcal{B}_n}\gamma_{nk} \end{bmatrix} \succ \mathbf{0}, \tag{6c}$$

where the constraints in (6c) guarantee that the $\mathcal{PFF}$ with factors in $\mathbb{F}_5$ is a valid Gaussian pdf. Notice that $\mathbb{F}_5$ covers $\mathbb{F}_1$, $\mathbb{F}_2$, $\mathbb{F}_3$ and $\mathbb{F}_4$ as special cases. For example, when $\phi_i = J_{ii}$, $\psi_i = h_i$, $\gamma_{ij} = \gamma_{ji} = \kappa_{ij} = \kappa_{ji} = 0$, $\tau_{ij} = J_{ij}$, $\mathbb{F}_5$ reduces to $\mathbb{F}_3$. Furthermore, when $\phi_i = \zeta_i$, $\psi_i = \zeta_i y_i$, $\gamma_{ij} = \eta_{ij}c_{ij}^2$, $\gamma_{ji} = \eta_{ij}c_{ji}^2$, $\tau_{ij} = \eta_{ij}c_{ij}c_{ji}$, $\kappa_{ij} = \eta_{ij}z_{ij}c_{ij}$ and $\kappa_{ji} = \eta_{ij}z_{ij}c_{ji}$, $\mathbb{F}_5$ reduces to $\mathbb{F}_4$. In fact, $\mathbb{F}_5$ covers pairwise factorizations beyond that in $\mathbb{F}_1$, $\mathbb{F}_2$, $\mathbb{F}_3$ and $\mathbb{F}_4$, as long as the exponents of $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ are quadratic polynomials. For example, when $f_i(x_i) \propto 1$ and $f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{1}{2}x_i^2 - \frac{1}{2}x_i x_j - \frac{1}{2}x_j^2 + x_i + x_j\right\}$ for all $i \neq j \in \{1, 2, \cdots, n\}$ in $\mathbb{F}_5$, it leads to a valid Gaussian pdf, but $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ cannot be represented by $\mathbb{F}_1$—$\mathbb{F}_4$. The relationships among $\mathbb{F}_1$—$\mathbb{F}_5$ are shown in Figure 1 at top of next page (relationships among $\mathbb{F}_1$—$\mathbb{F}_4$ are proved in Appendix A).

2.2. Gaussian Belief Propagation Under $\mathbb{F}_5$

In Gaussian BP, messages are updated and passed among variables. In particular, under synchronous scheduling[2], the message $m_{j\to i}(x_i)$ to be passed from variable x_j to variable

2. Synchronous scheduling requires that all messages at each iteration are updated before starting a new one.

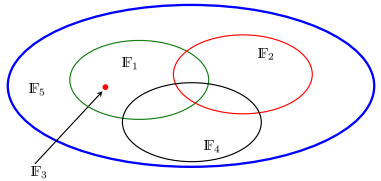


Figure 1: Relationships among different factorization sets.

x_i at the l-th iteration is computed as (Malioutov et al., 2006)

$$m_{j\to i}^{(l)}(x_i) \propto \int_{-\infty}^{\infty} f_{ij}(x_i, x_j) f_j(x_j) \prod_{k\in\mathcal{B}_j\setminus i} m_{k\to j}^{(l-1)}(x_j) dx_j, \tag{7}$$

where $m_{k\to j}^{(l-1)}(x_j)$ denotes the message $m_{k\to j}(x_i)$ passed from variable x_k to variable x_j at the $(l-1)$-th iteration.

Based on the expressions of factors in $\mathbb{F}_5$ and message update rule in (7), we can derive a general formula of messages $m_{j\to i}^{(l)}(x_i)$ for all $(i,j) \in \mathcal{E}$, where $\mathcal{E} \triangleq \{(i,j) \mid i = 1, 2, \cdots, n, j \in \mathcal{B}_i\}$ denotes the set of index pair (i,j) with $\tau_{ij} \neq 0$. Without loss of generality, it is assumed that the message $m_{j\to i}^{(l-1)}(x_i)$ with $(i,j) \in \mathcal{E}$ at the $(l-1)$-th iteration takes the expression

$$m_{j\to i}^{(l-1)}(x_i) \propto \exp\left\{-\frac{1}{2}\alpha_{j\to i}^{(l-1)} x_i^2 + \beta_{j\to i}^{(l-1)} x_i\right\}, \tag{8}$$

where the parameters $\alpha_{j\to i}^{(l-1)}$ and $\beta_{j\to i}^{(l-1)}$ are the precision and linear coefficient, respectively. Substituting (8) into (7) and if $\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)} > 0$, the message $m_{j\to i}(x_i)$ at the l-th iteration is well-defined and is shown in Appendix B to be in the form $m_{j\to i}^{(l)}(x_i) \propto \exp\{-\frac{1}{2}\alpha_{j\to i}^{(l)}x_i^2 + \beta_{j\to i}^{(l)}x_i\}$, with

$$\alpha_{j\to i}^{(l)} = \gamma_{ij} - \frac{\tau_{ij}^2}{\phi_j + \gamma_{ji} + \sum\limits_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)}}, \tag{9}$$

$$\beta_{j\to i}^{(l)} = \kappa_{ij} - \frac{\tau_{ij}\psi_j + \tau_{ij}\kappa_{ji} + \sum\limits_{k\in\mathcal{B}_j\setminus i}\tau_{ij}\beta_{k\to j}^{(l-1)}}{\phi_j + \gamma_{ji} + \sum\limits_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)}}. \tag{10}$$

After updating the messages $m_{j\to i}^{(l)}(x_i)$ by that of the parameters $\alpha_{j\to i}^{(l)}$ and $\beta_{j\to i}^{(l)}$ for all $(i,j) \in \mathcal{E}$, we can compute the belief of x_i at the l-th iteration as

$$b^{(l)}(x_i) \propto f_i(x_i) \prod_{j\in\mathcal{B}_i} m_{j\to i}^{(l)}(x_i). \tag{11}$$

By substituting the expression of $f_i(x_i)$ in $\mathbb{F}_5$ and $m^{(l)}_{j\to i}(x_i) \propto \exp\{-\frac{1}{2}\alpha^{(l)}_{j\to i}x_i^2 + \beta^{(l)}_{j\to i}x_i\}$ into (11), it becomes

$$b^{(l)}(x_i) \propto \exp\left\{-\frac{1}{2}\left(\phi_i + \sum_{j\in\mathcal{B}_i}\alpha^{(l)}_{j\to i}\right)x_i^2 + \left(\psi_i + \sum_{j\in\mathcal{B}_i}\beta^{(l)}_{j\to i}\right)x_i\right\}, \tag{12}$$

where the precision $\nu^{(l)}_{x_i}$ and linear coefficient $\varphi^{(l)}_{x_i}$ of belief $b^{(l)}(x_i)$ can be denoted as

$$\nu^{(l)}_{x_i} = \phi_i + \sum_{j\in\mathcal{B}_i}\alpha^{(l)}_{j\to i}, \tag{13}$$

$$\varphi^{(l)}_{x_i} = \psi_i + \sum_{j\in\mathcal{B}_i}\beta^{(l)}_{j\to i}. \tag{14}$$

For the belief $b^{(l)}(x_i)$, it represents a valid Gaussian pdf if and only if $\nu^{(l)}_{x_i} > 0$. Under $\nu^{(l)}_{x_i} > 0$, we can explicitly write $b^{(l)}(x_i) = \mathcal{N}(x_i; \varphi^{(l)}_{x_i}/\nu^{(l)}_{x_i}, 1/\nu^{(l)}_{x_i})$, where $\varphi^{(l)}_{x_i}/\nu^{(l)}_{x_i}$ and $1/\nu^{(l)}_{x_i}$ are the belief mean and belief variance of x_i, respectively. It has been proved (Pearl, 1988) that beliefs $b^{(l)}(x_i)$ are guaranteed to converge to the exact marginal distributions in singly connected graphs, such as tree-structured and chain-structured graphs. But for loopy graphs, it is known that beliefs may not converge. Fortunately, if beliefs do converge in Gaussian BP, belief means converge to the exact means of the marginal distributions and belief variances are approximate variances of the marginal distributions (Weiss and Freeman, 2000). This makes convergence analysis in loopy graphs a core research topic of Gaussian BP.

3. Convergence Analysis Under General Pairwise Factorization $\mathbb{F}_5$

In this section, we analyze the convergence of Gaussian BP under the $\mathcal{PFF}$ with general factors in $\mathbb{F}_5$. For notational convenience, we stack $\alpha^{(l)}_{j\to i}$, $\beta^{(l)}_{j\to i}$, γ_{ij} and κ_{ij} for all $(i,j)\in\mathcal{E}$ into vectors $\boldsymbol{\alpha}^{(l)}$, $\boldsymbol{\beta}^{(l)}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\kappa}$, respectively, where the order of $(i,j)\in\mathcal{E}$ is ascending first on i and then on j. For the order of $(i,j)\in\mathcal{E}$ ascending first on i and then on j, there exists a one-to-one mapping function $order(\cdot,\cdot)$ from all $(i,j)\in\mathcal{E}$ to $1,2,\cdots,|\mathcal{E}|$. That is, for any $k\in\{1,2,\cdots,|\mathcal{E}|\}$, there exists only one $(i,j)\in\mathcal{E}$ satisfying $k = order(i,j)$. Moreover, we define a set

$$\begin{aligned}\mathcal{Q} \triangleq \Big\{\mathbf{q}\in\mathbb{R}^{|\mathcal{E}|} \mid \phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i} > 0, \frac{-\tau_{ij}^2}{\phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i}} - q_{i\to j} \geq 0,\\ \forall (i,j)\in\mathcal{E}, \text{ and } \phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i} q_{k\to i} > 0 \text{ for } i=1\Big\},\end{aligned} \tag{15}$$

where $\mathbf{q}\in\mathbb{R}^{|\mathcal{E}|}$ is a vector obtained by stacking $q_{j\to i}$ for all $(i,j)\in\mathcal{E}$ ordered in the same way as that in $\boldsymbol{\alpha}^{(l)}$. For the set $\mathcal{Q}$, it is non-empty if and only if there exists at least one vector $\mathbf{q}\in\mathbb{R}^{|\mathcal{E}|}$ satisfying the inequalities in (15). As shown later, the set $\mathcal{Q}$ would be used to verify the convergence of $\boldsymbol{\alpha}^{(l)}$.

First, we consider the convergence of BP beliefs under synchronous scheduling. For synchronous scheduling, if $\boldsymbol{\alpha}^{(l)}$ converge to $\boldsymbol{\alpha}^*$ with elements $\alpha^*_{j\to i}$ for all $(i,j)\in\mathcal{E}$ ordered

in the same way as that in $\boldsymbol{\alpha}^{(l)}$, the update of $\beta^{(l)}_{j\to i}$ in (10) for all $(i,j)\in\mathcal{E}$ can be written in a vector form as

$$\boldsymbol{\beta}^{(l)} = \mathbf{A}\boldsymbol{\beta}^{(l-1)} + \mathbf{d}, \tag{16}$$

where $\mathbf{A}$ is a $|\mathcal{E}|\times|\mathcal{E}|$ matrix with the element

$$A_{i'j'} = \begin{cases} -\frac{\tau_{i_1 i_2}}{\phi_{i_1}+\gamma_{i_1 i_2}+\sum_{k\in\mathcal{B}_{i_1}\setminus i_2}\alpha^*_{k\to i_1}} & \text{if } j_1\in\mathcal{B}_{i_1}\setminus i_2 \text{ and } j_2 = i_1,\\ 0 & \text{otherwise,}\end{cases} \tag{17}$$

for $i' = order(i_1,i_2)$ and $j' = order(j_1,j_2)$, and $\mathbf{d}$ is a column vector containing elements $\kappa_{ij} - \frac{\tau_{ij}\psi_j+\tau_{ij}\kappa_{ji}}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i}\alpha^*_{k\to j}}$ for all $(i,j)\in\mathcal{E}$ ordered in the same way as that in $\boldsymbol{\alpha}^{(l)}$. Conditioned on the convergence of $\boldsymbol{\alpha}^{(l)}$, the convergence of $\boldsymbol{\beta}^{(l)}$ can be easily analyzed with the linear update equation in (16). In the following, we present the convergence conditions of Gaussian BP under general pairwise factorization and synchornous scheduling.

Theorem 1 ***(Convergence conditions under synchronous scheduling)*** *Under synchronous scheduling and any initialization $\boldsymbol{\alpha}^{(0)}\geq\boldsymbol{\gamma}$, belief variances converge to the same positive values if and only if the set $\mathcal{Q}$ is non-empty. Furthermore, if $\mathcal{Q}$ is non-empty, belief means converge to unique values if and only if $\rho(\mathbf{A})<1$ under synchronous scheduling.*

Proof See Appendix C. ■

Theorem 1 states that after we fix the factorization parameters in $\mathbb{F}_5$, Gaussian BP converges if and only if the set $\mathcal{Q}$ is non-empty and $\rho(\mathbf{A})<1$. Furthermore, the converged values of BP beliefs are independent of the choice of initialization as long as $\boldsymbol{\alpha}^{(0)}\geq\boldsymbol{\gamma}$. Notice that the spectral radius $\rho(\mathbf{A})$ depends on the converged value of $\boldsymbol{\alpha}^{(l)}$, which can be obtained by running the update of $\boldsymbol{\alpha}^{(l)}$ until convergence. On the other hand, the condition $\rho(\mathbf{A})<1$ will be useful in analyzing the invariant convergence properties of Gaussian BP in Section 4 without the need to know the converged value $\boldsymbol{\alpha}^*$.

Synchronous scheduling requires each variable receiving updated messages from all neighbors before computing the next round messages. In contrast, asynchronous scheduling allows distributed processors to operate more efficiently and flexibly without waiting for all its neighbors' messages. For totally asynchronous scheduling (Bertsekas and Tsitsiklis, 1989), the convergence condition is given by the following Theorem.

Theorem 2 ***(Convergence conditions under totally asynchronous scheduling)*** *Under totally asynchronous scheduling and any initialization $\boldsymbol{\alpha}^{(0)}\geq\boldsymbol{\gamma}$, belief variances converge to the same positive values if and only if the set $\mathcal{Q}$ is non-empty. Moreover, if $\mathcal{Q}$ is non-empty and $\rho(|\mathbf{A}|)<1$, belief means converge to unique values under totally asynchronous scheduling.*

Proof See Appendix D. ■

From Theorems 1 and 2, it is noticed that the convergence condition of belief variances under synchronous and totally asynchronous schedulings stays the same, i.e., $\mathcal{Q} \neq \emptyset$. Furthermore, since totally asynchronous scheduling includes synchronous scheduling as a special case, the converged value of BP message precision $\boldsymbol{\alpha}^*$ stays the same for both synchronous and totally asynchronous schedulings. Consequently, the matrix $\mathbf{A}$ in both Theorems 1 and 2 are the same. On the other hand, for the convergence condition of belief means, the condition in Theorem 2 is stricter than that in Theorem 1 since $\rho(\mathbf{A}) \leq \rho(|\mathbf{A}|)$. This implies that if the convergence condition in Theorem 2 for totally asynchronous scheduling is satisfied, the convergence condition in Theorem 1 for synchronous scheduling would automatically be satisfied. But Theorem 1 has its own value since it is the necessary and sufficient condition, while Theorem 2 is only a sufficient condition. Notice that the initialization $\boldsymbol{\alpha}^{(0)}$ also plays a role in the convergence of BP beliefs, but $\boldsymbol{\alpha}^{(0)}$ is a parameter controllable by the user, therefore the requirement $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$ can easily be satisfied.

For the verification of the non-emptiness of the set $\mathcal{Q}$, one may directly solve the following semidefinite programming (SDP) (Vandenberghe and Boyd, 1996) problem:

$$\min_{\mathbf{q}\in\mathbb{R}^{|\mathcal{E}|},a} \quad a, \tag{18a}$$

$$\text{s.t.} \quad \begin{bmatrix} \phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i} & \tau_{ij} \\ \tau_{ij} & -q_{i\to j} \end{bmatrix} \succeq \mathbf{0}, \forall (i,j)\in\mathcal{E}, \tag{18b}$$

$$a + \phi_1 + \sum_{k\in\mathcal{B}_1}\gamma_{1k} + \sum_{k\in\mathcal{B}_1} q_{k\to 1} \geq 0, \tag{18c}$$

where the constraints in (18b) are equivalent to the first two constraints in $\mathcal{Q}$, and the constraint in (18c) corresponds to the third constraint in $\mathcal{Q}$ after introducing a slack variable a. If the SDP problem in (18a)—(18c) is feasible with the solution $(\mathbf{q}^\dagger, a^\dagger)$ and $a^\dagger < 0$, then $\mathbf{q}^\dagger$ satisfies the constraints in (18b) and (18c) with $\phi_1 + \sum_{k\in\mathcal{B}_1}\gamma_{1k} + \sum_{k\in\mathcal{B}_1} q^\dagger_{k\to 1} \geq -a^\dagger > 0$, which implies $\mathcal{Q}$ is non-empty. Otherwise, $\mathcal{Q}$ would be empty. For solving the SDP problem in (18a)—(18c), interior point methods, with available solvers such as SeDuMi, SDPT3, and MOSEK, can be used.

On the other hand, the optimization problem in (18a)—(18c) can be reformulated in the form:

$$\min_{\mathbf{S}\in\mathcal{S}^{2|\mathcal{E}|+1},\mathbf{q}\in\mathbb{R}^{|\mathcal{E}|},a} \quad a, \tag{19a}$$

$$\text{s.t.} \quad \mathbf{S} \succeq \mathbf{0}, \tag{19b}$$

$$a\mathbf{M}_a + \sum_{(i,j)\in\mathcal{E}} q_{j\to i}\mathbf{M}_{j\to i} + \mathbf{S} = \mathbf{B}, \tag{19c}$$

where $\mathcal{S}^m$ denotes the set of $m\times m$ symmetric matrix, $\mathbf{M}_a \triangleq \text{diag}(-1,\mathbf{0}) \in \mathcal{S}^{2|\mathcal{E}|+1}$, $\mathbf{B} \triangleq \text{diag}(\phi_1 + \sum_{k\in\mathcal{B}_1}\gamma_{1k}, \mathbf{D}^{(B)}_{1,k_1}, \cdots, \mathbf{D}^{(B)}_{i,k_i}, \cdots, \mathbf{D}^{(B)}_{n,k_n}) \in \mathcal{S}^{2|\mathcal{E}|+1}$ with

$$\mathbf{D}^{(B)}_{i,k_i} = \begin{bmatrix} \phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} & \tau_{ik_i} \\ \tau_{ik_i} & 0 \end{bmatrix} \text{ for all } i = 1,2,\cdots,n \text{ and } k_i \in \mathcal{B}_i,$$

and $\mathbf{M}_{j\to i} \triangleq \text{diag}(d^{(M)}, \mathbf{D}^{(M)}_{1,k_1}, \cdots, \mathbf{D}^{(M)}_{i,k_i}, \cdots, \mathbf{D}^{(M)}_{n,k_n}) \in \mathcal{S}^{2|\mathcal{E}|+1}$ with

$$d^{(M)} = \begin{cases} -1 & \text{if } i = 1 \text{ and } j \in \mathcal{B}_1, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathbf{D}^{(M)}_{i_1,j_1} = \begin{cases} \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} & \text{if } i_1 = i \text{ and } j_1 \in \mathcal{B}_i \setminus j, \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & \text{else if } i_1 = j \text{ and } j_1 = i, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

The form of the SDP problem in (19a)—(19c) is exactly the same as that of the SDP problem in equation (3) of Wen et al. (2010). Therefore, the alternating direction augmented Lagrangian methods in Wen et al. (2010) can also be used to solve the SDP problem in (19a)—(19c).

Furthermore, we can interpret the physical meaning of the convergence condition in Theorems 1 and 2 as follows. Under initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, the first two constraints in the set $\mathcal{Q}$ correspond to the condition for guaranteeing the validity of integration in (32). Together with the third constraint in the set $\mathcal{Q}$, they correspond to the condition for guaranteeing the BP beliefs in (12) remain as valid Gaussian pdfs at each iteration. This is summarized and formally proved in the following Corollary.

Corollary 3 ***(Physical meaning of non-emptiness of $\mathcal{Q}$)*** *Under any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, BP beliefs at each iteration are valid Gaussian pdfs if and only if the set $\mathcal{Q}$ is non-empty.*

Proof See Appendix E. ■

Note that being a valid Gaussian pdf in $\mathbb{F}_5$ does not necessarily imply that the BP beliefs are valid Gaussian pdfs at all iterations. Rather, being a valid Gaussian pdf in $\mathbb{F}_5$ is a prerequisite for Theorems 1, 2, and Corollary 3.

4. Invariant Properties of Convergence Conditions Under $\mathbb{F}_5$

For the $\mathcal{PFF}$ with factors in $\mathbb{F}_5$, it can be equivalently written as a Gaussian pdf $\mathcal{N}(\mathbf{x}; \mathbf{J}^{-1}\mathbf{h}, \mathbf{J}^{-1})$ with the information matrix

$$\mathbf{J} \triangleq \begin{bmatrix} \phi_1 + \sum_{k\in\mathcal{B}_1}\gamma_{1k} & \tau_{12} & \cdots & \tau_{1n} \\ \tau_{21} & \phi_2 + \sum_{k\in\mathcal{B}_2}\gamma_{2k} & \cdots & \tau_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tau_{n1} & \tau_{n2} & \cdots & \phi_n + \sum_{k\in\mathcal{B}_n}\gamma_{nk} \end{bmatrix}, \tag{20}$$

and the potential vector

$$\mathbf{h} \triangleq \left[\psi_1 + \sum_{k\in\mathcal{B}_1}\kappa_{1k}, \psi_2 + \sum_{k\in\mathcal{B}_2}\kappa_{2k}, \cdots, \psi_n + \sum_{k\in\mathcal{B}_n}\kappa_{nk}\right]. \tag{21}$$

Conversely, for a given Gaussian pdf $\mathcal{N}(\mathbf{x}; \mathbf{J}^{-1}\mathbf{h}, \mathbf{J}^{-1})$, there are many possible pairwise factorizations with each obtained by choosing different ϕ_i, ψ_i, γ_{ij}, τ_{ij} and κ_{ij} while satisfying the constraints in $\mathbb{F}_5$. Since the convergence conditions in Theorems 1 and 2 seem to depend on the factorization parameters. One may wonder would different factorizations demonstrate different convergence behaviors? We will give the answer by the following discussions.

From (20) and (21), it is obvious that for a Gaussian pdf with information matrix $\mathbf{J}$ and potential vector $\mathbf{h}$, the factorization parameters of any pairwise factorization with factors in the form of $\mathbb{F}_5$ should satisfy $\phi_i + \sum_{k\in\mathcal{B}_i} \gamma_{ik} = J_{ii}$, $\tau_{ij} = J_{ij}$ and $\psi_i + \sum_{k\in\mathcal{B}_i} \kappa_{ik} = h_i$. Using these relationships, the definition of the set $\mathcal{Q}$ can be rewritten as

$$\mathcal{Q} \triangleq \Big\{\mathbf{q} \in \mathbb{R}^{|\mathcal{E}|} \mid J_{ii} + \sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i} > 0, \frac{-J_{ij}^2}{J_{ii} + \sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i}} - q_{i\to j} \geq 0, \forall (i,j) \in \mathcal{E}, \\ \text{and } J_{ii} + \sum_{k\in\mathcal{B}_i} q_{k\to i} > 0 \text{ for } i = 1\Big\}. \tag{22}$$

From (22), one can obtain that the emptiness or non-emptiness of the set $\mathcal{Q}$ only depends on the information matrix $\mathbf{J}$ and is independent of ϕ_i, ψ_i, γ_{ij}, τ_{ij} and κ_{ij} as long as they obey $\mathbb{F}_5$. The above discussion is summarized in the following Lemma.

Lemma 4 *For a Gaussian pdf with information matrix $\mathbf{J}$, the emptiness or non-emptiness of the set $\mathcal{Q}$ is independent of factorization parameters in $\mathbb{F}_5$.*

Furthermore, when belief variances converge, it can be proved that the matrix $\mathbf{A}$ is independent of the choice of factorization parameters ϕ_i, ψ_i, γ_{ij}, τ_{ij} and κ_{ij}. This is formalized in the following Lemma.

Lemma 5 *For a Gaussian pdf with information matrix $\mathbf{J}$, if the set $\mathcal{Q}$ is non-empty, matrix $\mathbf{A}$ is independent of factorization parameters in $\mathbb{F}_5$ for any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$.*

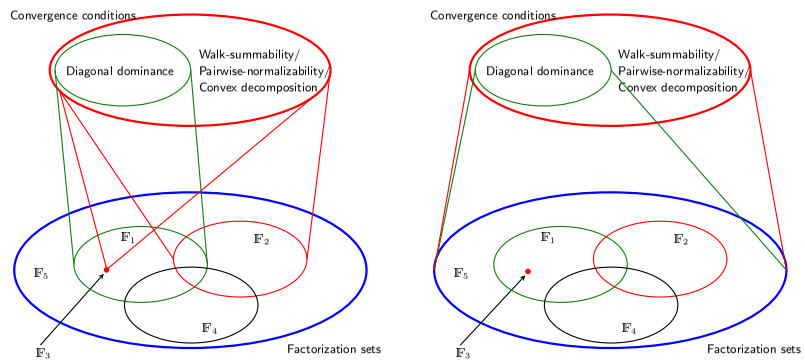
Proof See Appendix F. ■

Since $\mathcal{Q}$ and $\mathbf{A}$ are independent of the specific pairwise factorization, and the convergence conditions in Theorems 1 and 2 only depend on $\mathcal{Q}$ and $\mathbf{A}$, we obtain the invariant properties of Gaussian BP convergence in the following two Corollaries.

Corollary 6 ***(Convergence or divergence being independent of factorization)*** *For a Gaussian pdf with information matrix $\mathbf{J}$, under synchronous scheduling and any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, the convergence or divergence of Gaussian BP is independent of factorization parameters in $\mathbb{F}_5$.*

Proof See Appendix G. ■

Corollary 6 is a very strong result as convergence (or divergence) in one factorization automatically implies convergence (or divergence) in all other factorizations. On the other



(a) Previous results. (b) New relationships revealed by Corollary 7.

Figure 2: Relationships of classical convergence conditions and factorization sets.

hand, for totally asynchronous scheduling, due to the fact that Theorem 2 is only a sufficient condition under $\mathbb{F}_5$, we cannot have a similar conclusion to that of Corollary 6.

Previously, walk-summability was proved to be equivalent to pairwise-normalizability (Malioutov et al., 2006) and convex decomposition (Moallemi and Roy, 2009), thus they are all valid within factorizations $\mathbb{F}_2$ and $\mathbb{F}_3$. On the other hand, diagonal dominance is valid within $\mathbb{F}_1$ (Weiss and Freeman, 2000), and was proved to be a special case of walk-summability (Malioutov et al., 2006). These relationships are illustrated in Figure 2(a). But whether these classical conditions are valid for other pairwise factorizations is still unknown. To this end, we obtain a result that is practically very useful in the following Corollary.

Corollary 7 ***(Validity of existing sufficient conditions under any pairwise factorization)*** *For a Gaussian pdf with information matrix $\mathbf{J}$, if $\mathbf{J}$ is diagonally dominant, walk-summable, pairwise-normalizable or convex decomposable, Gaussian BP converges under any pairwise factorization within $\mathbb{F}_5$ in both synchronous and totally asynchronous schedulings as long as initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$.*

Proof See Appendix H. ■

With Corollary 7, it gives a more complete picture: diagonal dominance, walk-summability, pairwise-normalizability, and convex decompositions are all valid sufficient conditions under $\mathbb{F}_5$, which includes $\mathbb{F}_1$, $\mathbb{F}_2$ and $\mathbb{F}_3$ as special cases. This extension is illustrated in Figure 2(b). Moreover, the initialization set of $\boldsymbol{\alpha}^{(0)}$ in Corollary 7 is larger than that in previous conditions (Weiss and Freeman, 2000; Malioutov et al., 2006; Moallemi and Roy, 2009). Notice that from Corollary 7, the initialization set is related to the parameter $\boldsymbol{\gamma}$, which means that typical initialization $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ will not always work. This point will be further illustrated in Section 6.1. For comparison with the results in Su and Wu (2015a), it is no-

ticed that Theorems 1 and 2 can be considered as extensions of the convergence conditions in Su and Wu (2015a) from a single factorization $\mathbb{F}_3$ to a much larger set $\mathbb{F}_5$.

Corollaries 6 and 7 only give us the information about the convergence or divergence of Gaussian BP under different pairwise factorizations. Another important question is if BP beliefs converge, would the converged values of beliefs be the same under different pairwise factorizations? The answer is given in the following Theorem.

Theorem 8 ***(Invariant converged beliefs)*** *For a Gaussian pdf with information matrix* $\mathbf{J}$ *and potential vector* $\mathbf{h}$*, under any initialization* $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$*, if Gaussian BP converges in synchronous or totally asynchronous scheduling, the converged beliefs are independent of factorization parameters in* $\mathbb{F}_5$*.*

Proof See Appendix I. ■

5. A Simple Convergence Condition in Pairwise Linear Gaussian Model

In the pairwise linear Gaussian model $\mathbb{F}_4$, we have the parameters of $\mathbb{F}_5$ as $\phi_i = \zeta_i$, $\psi_i = \zeta_i y_i$, $\gamma_{ij} = \eta_{ij} c_{ij}^2$, $\gamma_{ji} = \eta_{ij} c_{ji}^2$, $\tau_{ij} = \eta_{ij} c_{ij} c_{ji}$, $\kappa_{ij} = \eta_{ij} z_{ij} c_{ij}$ and $\kappa_{ji} = \eta_{ij} z_{ij} c_{ji}$. Further setting different values of ζ_i, y_i, η_{ij}, z_{ij}, c_{ij} and c_{ji}, the model covers various applications (Leng and Wu, 2011; Du and Wu, 2013; Moallemi and Roy, 2006; Bickson et al., 2007), where the convergence of Gaussian BP is studied separately. While we can use the general results in Theorems 1 and 2 to verify whether Gaussian BP converges in a particular pairwise linear Gaussian model, this approach requires us to verify the non-emptiness of the set $\mathcal{Q}$ and find the spectral radius of matrix $\mathbf{A}$. In the following, by further making use of the special structure of $\mathbb{F}_4$, we will provide an easily verifiable sufficient convergence condition for the pairwise linear Gaussian model. Before we state the result, we need the following Lemma.

Lemma 9 *A Gaussian model obeying* $\mathbb{F}_5$ *is pairwise-normalizable if there exist* ω_{ij} *for all* $(i,j) \in \mathcal{E}$ *such that*

$$\omega_{ij} \geq 0 \text{ for all } (i,j) \in \mathcal{E}, \tag{23a}$$

$$\omega_{ij}\omega_{ji} - \tau_{ij}^2 \geq 0 \text{ for all } (i,j > i) \in \mathcal{E}, \tag{23b}$$

$$\phi_i + \sum_{j \in \mathcal{B}_i} \gamma_{ij} - \sum_{j \in \mathcal{B}_i} \omega_{ij} > 0 \text{ for all } i = 1, 2, \cdots, n. \tag{23c}$$

Proof See Appendix J. ■

Making use of Lemma 9, we can prove that the pairwise linear Gaussian model $\mathbb{F}_4$ with at least one $\zeta_i > 0$ is pairwise-normalizable, therefore by Theorem 8, Gaussian BP is convergent under synchronous and totally asynchronous schedulings. The detail is given in the following Theorem.

Theorem 10 ***(At least one*** $\zeta_i > 0$ ***guarantees convergence)*** *For pairwise linear Gaussian model* $\mathbb{F}_4$*, under any initialization* $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$*, if* $\zeta_i > 0$ *for at least one* i*, Gaussian BP converges in both synchronous and totally asynchronous schedulings.*

Proof See Appendix K. ■

From Theorem 10, the convergence of Gaussian BP in pairwise linear Gaussian model can be guaranteed if there exists one $\zeta_i > 0$, which is much easier to check than directly verifying the conditions in Theorems 1 and 2. For example, in clock synchronization, ζ_i has the physical interpretation of precision of prior distribution of unknown clock offset. Theorem 10 implies that as long as there is at least one node having an informative prior on the clock parameter, the distributed clock synchronization algorithm (Leng and Wu, 2011) would converge. This is a much stronger result than that in Leng and Wu (2011) since Leng and Wu (2011) proves the convergence under one reference node with perfect prior information (i.e., $p(x_i) = \delta(x_i)$). In contrast, Theorem 10 allows the reference node to have imperfect prior, and more importantly, Theorem 10 allows more than one node having prior information, possibly with different precisions.

On the other hand, for peer-to-peer rating applications, ζ_i represents the confidence of node i's initial rating y_i. Theorem 10 shows that Gaussian BP converges with a minimum of one initial rating with corresponding confidence $\zeta_i > 0$. This is obviously true for a practical peer-to-peer rating system. Notice that while Bickson et al. (2007) requires all ζ_i to be the same for different nodes, our model $\mathbb{F}_4$ can accommodate different confidences at different nodes and Gaussian BP still converges due to Theorem 10. This is the first time that the convergence of Gaussian BP is proved in such an application. Furthermore, if all the observations $\{y_i\}$ exist with the same confidence $\zeta_i = 1$, the model $\mathbb{F}_4$ becomes the consensus propagation model in Moallemi and Roy (2006). Due to $\zeta_i > 0$ for all i, Theorem 10 guarantees the convergence of Gaussian BP in consensus propagation.

6. Numerical Results and Applications

In this section, numerical results are presented to corroborate the newly established theoretical results. In the following simulations, we consider two message passing schedules: synchronous scheduling, and an asynchronous schedule with each variable updating its message with probability 0.7 at each iteration.

6.1. Impact of Different Pairwise Factorizations

Consider a joint Gaussian distribution of 6 random variables, where the information matrix $\mathbf{J}$ and potential vector $\mathbf{h}$ are

$$\mathbf{J} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 6 & -1 & 2 & 0 & 0 \\ 0 & -1 & 5 & -6 & 0 & 0 \\ 0 & 2 & -6 & 36 & 3 & -10 \\ 0 & 0 & 0 & 3 & 9 & 0 \\ 0 & 0 & 0 & -10 & 0 & 4 \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

We consider two pairwise factorizations with different parameters in $\mathbb{F}_5$. In pairwise factorization I, we set $\phi_1 = 1$, $\phi_2 = 6$, $\phi_3 = 5$, $\phi_4 = 36$, $\phi_5 = 9$, $\phi_6 = 4$, $\psi_i = 1$ for all $i = 1, 2, \cdots, 6$, $\tau_{12} = 1$, $\tau_{23} = -1$, $\tau_{24} = 2$, $\tau_{34} = -6$, $\tau_{45} = 3$, $\tau_{46} = -10$, and $\gamma_{ij} = \kappa_{ij} = 0$

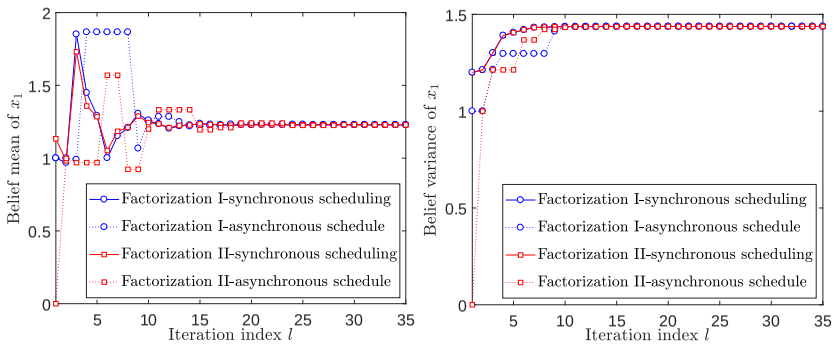


Figure 3: Convergence of beliefs under different pairwise factorizations and message schedulings.

for all $(i,j) \in \mathcal{E}$. The factors in this factorization are used in Malioutov et al. (2006), and Su and Wu (2014, 2015a). In pairwise factorization II, we set $\phi_i = 0$ and $\psi_i = 0$ for all $i = 1, 2, \cdots, 6$, $\gamma_{12} = \gamma_{21} = \gamma_{23} = \gamma_{32} = \gamma_{42} = \gamma_{45} = 1$, $\gamma_{24} = \gamma_{34} = \gamma_{64} = 4$, $\gamma_{43} = \gamma_{54} = 9$, $\gamma_{46} = 25$, $\tau_{12} = 1$, $\tau_{23} = -1$, $\tau_{24} = 2$, $\tau_{34} = -6$, $\tau_{45} = 3$, $\tau_{46} = -10$, $\kappa_{12} = \kappa_{54} = \kappa_{64} = 1$, $\kappa_{21} = \kappa_{23} = \kappa_{24} = 1/3$, $\kappa_{32} = \kappa_{34} = 1/2$, and $\kappa_{42} = \kappa_{43} = \kappa_{45} = \kappa_{46} = 1/4$.

It can be verified that the set $\mathcal{Q}$ is non-empty for the considered Gaussian distribution by solving the SDP problem in (18a)—(18c) (as discussed in Lemma 4, the set $\mathcal{Q}$ is independent of the chosen factorization). Furthermore, under initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, numerical computation shows that $\rho(\mathbf{A}) = 0.6934 < 1$ and $\rho(|\mathbf{A}|) = 0.6934 < 1$ for both pairwise factorizations. This corroborates Lemma 5 that $\mathbf{A}$ is independent of factorizations. By Theorems 1 and 2, under initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, Gaussian BP beliefs converge in both synchronous and totally asynchronous schedulings. Figure 3 shows belief means and variances of variable x_1 during the iterations, where $\boldsymbol{\alpha}^{(0)}$ is set to $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}^{(0)}$ is set to $\mathbf{0}$ in both pairwise factorizations (i.e., $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ for factorization I and $\boldsymbol{\alpha}^{(0)} = [1\ 1\ 1\ 4\ 1\ 4\ 1\ 9\ 1\ 25\ 9\ 4]^T$ for factorization II). It can be seen that belief means (variances) converge to the same values for both factorizations under synchronous scheduling and asynchronous scheduling, which corroborates Theorem 8.

Nevertheless, if we set the initialization $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ in factorization II, by the update equation in (9), $\boldsymbol{\alpha}^{(l)}$ can be easily shown to maintain at $\mathbf{0}$ for all $l \geq 0$. Then belief variance $1/(\phi_i + \sum_{j \in \mathcal{B}_i} \alpha_{j \to i}^{(l)}) = 1/0$ is not defined. But for factorization I, BP beliefs do converge when $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$. Therefore, the initialization $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ will not always work for different pairwise factorizations. To guarantee the convergence of Gaussian BP, we need to choose the initialization $\boldsymbol{\alpha}^{(0)}$ according to the factorization parameter $\boldsymbol{\gamma}$.

6.2. Distributed Clock Synchronization

In clock synchronization of wireless sensor networks (Leng and Wu, 2011), x_i represents the unknown clock offset at node i, and the likelihood function can be regarded as $f_{ij}(x_i, x_j)$

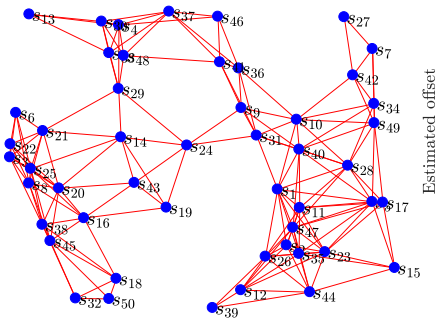


Figure 4: Network topology in clock synchronization.

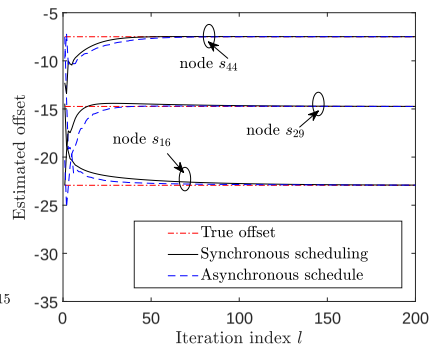


Figure 5: Estimated offsets.

with the expression

$$f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{N}{2\sigma^2}\left[2\left(x_i - x_j\right) + \frac{\mathbf{1}^T \mathbf{t}_{ij}}{N}\right]^2\right\}, \tag{24}$$

where $\mathbf{1}$ is the all-ones vector of length N, $\mathbf{t}_{ij}$ is a column vector collecting N observations, and the observation errors are Gaussian distributed with zero mean and variance σ^2. Moreover, the prior of x_i is Gaussian distributed, and it can be regarded as $f_i(x_i)$ with the expression

$$f_i(x_i) \propto \exp\left\{-\frac{1}{2\bar{\sigma}_i^2}\left(x_i - \bar{\mu}_i\right)^2\right\}, \tag{25}$$

where $\bar{\mu}_i$ and $\bar{\sigma}_i^2$ are the mean and variance, respectively. In case there is no prior information for x_i, we can set $\bar{\sigma}_i^2 = \infty$ and $f_i(x_i) \propto 1$. On the other hand, if the prior of x_i is perfect, then $\bar{\sigma}_i^2 = 0$ and $f_i(x_i) \propto \delta(x_i - \bar{\mu}_i)$.

We consider a network with topology shown in Figure 4, where there are 50 nodes. Among these nodes, the clock offsets of nodes with no prior information are drawn uniformly from $[-30, 30]$. Moreover, we set $N = 4$, $\sigma^2 = 1$ and two reference nodes s_1 and s_4 with their offsets drawn from $\mathcal{N}(x_1; 0, 10^{-1})$ and $\mathcal{N}(x_4; 0, 10^{-2})$, respectively. By Theorem 10, if there exists at least one node with prior information in the network, BP beliefs always converge under both synchronous and totally asynchronous schedulings. Figure 5 shows the estimated offsets of nodes s_{16}, s_{29}, s_{44} under synchronous scheduling and asynchronous scheduling. It can be seen that the estimated offsets of each node converge to the same values under both synchronous and asynchronous schedulings, corroborating the results in Theorem 8.

6.3. Peer-to-Peer Rating

The ratings of items, such as movies, doctors and vendors, play an important role in the social networks and can affect the decisions of people in some degree. Define x_i as the rating

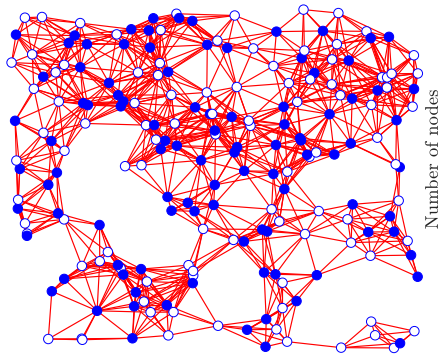


Figure 6: Network topology of peer-to-peer rating.

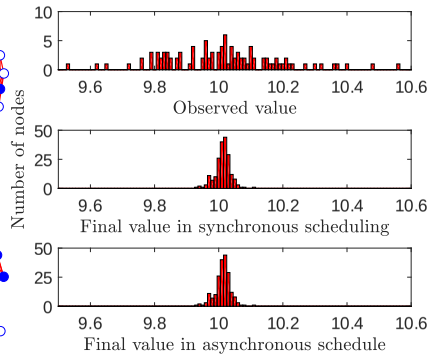


Figure 7: Histograms of consensus values.

at node i, and the peer-to-peer rating problem can be formulated as

$$\min_{x_1,x_2,\cdots,x_n} \sum_i \frac{1}{2}\varphi_{ii}(x_i - z_i)^2 + \sum_{i\neq j} \frac{1}{2}\varphi_{ij}(x_i - x_j)^2, \tag{26}$$

where z_i is the initial rating of node i, φ_{ii} and φ_{ij} denote the self confidence of node i and mutual trust between node i and node j, respectively. The term in the first summation of (26) denotes the rating difference between the final result and its initialization, and the term in the second summation of (26) denotes the rating difference between a pair of neighboring nodes. In particular, if we set

$$f_i(x_i) \propto \exp\left\{-\frac{1}{2}\varphi_{ii}(x_i - z_i)^2\right\}, \tag{27}$$

$$f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{1}{2}\varphi_{ij}(x_i - x_j)^2\right\}, \tag{28}$$

then Gaussian BP can be performed to estimate $\{x_i\}$.

We consider a network with topology in Figure 6, where there are 200 nodes. Out of the 200 nodes, 100 randomly selected nodes (denoted as solid nodes in Figure 6) have observations z_i on a common object or variable with true value 10. The observations are disturbed by zero-mean Gaussian noise with variance φ_{ii} uniformly sampled from $[10, 10^2]$, and we set φ_{ij} to $10^{1.5}$. Comparing $f_i(x_i)$ in (27) with that in $\mathbb{F}_4$, it is noticed that $\xi_i = \varphi_{ii} > 0$ for the nodes with observations. By Theorem 10, Gaussian BP beliefs converge under both synchronous and totally asynchronous schedulings. This can be seen from Figure 7 that the final values at all nodes concentrate much more closely to the true common value compared to the initial observed values. Furthermore, the final consensus values are consistent under synchronous scheduling and asynchronous scheduling.

Figure 8: (a) Original image, (b) Corrupted image, (c) Recovered image by Gaussian BP, (d) Recovered image by median filter (5×5) for all pixels, (e) Recovered image by median filter (5×5) only for pixels with values 0 or 255.

6.4. Image Denoising

Consider a $M \times N$ gray level image (with level $[0, 255]$) corrupted by salt-and-pepper noise. With the true gray level of pixel at location (i, j) in the original image denoted by $x_{i,j}$, we can formulate the image restoration problem as

$$\begin{aligned}\min_{\{x_{i,j}\}} \quad & \sum_{1\le i\le M, 1\le j\le N} \frac{1}{2} w_{i,j}(x_{i,j} - z_{i,j})^2 + \\ & \sum_{(i,j)\in\mathcal{D}} \frac{1}{2}(x_{i,j} - x_{i,j-1})^2 + \frac{1}{2}(x_{i,j} - x_{i,j+1})^2 + \frac{1}{2}(x_{i,j} - x_{i-1,j})^2 + \frac{1}{2}(x_{i,j} - x_{i+1,j})^2,\end{aligned} \tag{29}$$

where $w_{i,j}$ is the confidence of the observed pixel value $z_{i,j}$ at the location (i, j), and $\mathcal{D}$ denotes the set of the locations of pixels with values 0 or 255. From (29), if we set the factors

$$f(x_{i,j}) \propto \exp\left\{-\frac{1}{2} w_{i,j}(x_{i,j} - z_{i,j})^2\right\}, \tag{30}$$

$$f(x_{i,j}, \widetilde{x}) \propto \exp\left\{-\frac{1}{2}(x_{i,j} - \widetilde{x})^2\right\}, \tag{31}$$

where $\widetilde{x}$ denotes the neighboring pixel of $x_{i,j}$ (i.e., $x_{i,j-1}$ or $x_{i,j+1}$ or $x_{i-1,j}$ or $x_{i+1,j}$), by treating all $x_{i,j}$ as continuous variables, we can perform Gaussian BP to recover the image.

As a demonstration, we choose a 256×256 gray-scale Lena image. The original image (shown in Figure 8(a)) is corrupted by 50% salt-and-pepper noise on the face and 5% salt-and-pepper noise on other parts, as shown in Figure 8(b). Figure 8(c) shows the recovered image by Gaussian BP under synchronous scheduling, where we set $w_{i,j} = 10^{-6}$ for $(i, j) \in \mathcal{D}$ and $w_{i,j} = 1$ for $(i, j) \notin \mathcal{D}$. Since $w_{i,j} > 0$, by Theorem 10, we know that Gaussian BP converges in solving the problem in (29) under both synchronous and totally asynchronous schedulings. Moreover, for comparison, Figure 8(d) and (e) show the images recovered by the median filter (5×5) for all pixels and only for pixels with values 0 or 255, respectively. It can be seen that the image recovered by Gaussian BP has a much better visual quality than those by the median filters.

7. Conclusions

In this paper, a unified convergence analysis of Gaussian BP in Gaussian MRF and pairwise linear Gaussian model was presented. By using a general pairwise factorization of the joint Gaussian distribution, general convergence conditions of the Gaussian BP beliefs were derived for both synchronous and totally asynchronous schedulings. With the general convergence conditions, existing convergence conditions such as walk-summability, pairwise-normalizability, convex decomposition, and diagonal dominance, were extended from their original considered pairwise factorizations to the proposed general pairwise factorization. Moreover, by further linking the pairwise-normalizability in Gaussian MRF to pairwise linear Gaussian model, an easily verifiable sufficient convergence condition was proposed. Numerical examples and applications were presented to corroborate the newly established convergence results.

Appendix A. Relationships Among Factorizations $\mathbb{F}_1$—$\mathbb{F}_4$

In $\mathbb{F}_1$, if we set $a_{ij} = a_{ji} = 0$, $p_i = J_{ii}$, then $\mathbb{F}_1$ is reduced to $\mathbb{F}_3$, which implies that $\mathbb{F}_3$ is a specific setting of $\mathbb{F}_1$. On the other hand, from the constraint of $\mathbb{F}_2$, ξ_{ij}, $\xi_{ji} > 0$, so the coefficients of the terms x_i^2 and x_j^2 in $f_{ij}(x_i, x_j)$ cannot be zero. Therefore, $\mathbb{F}_2$ cannot reduce to $\mathbb{F}_3$, and vice versa. Furthermore, in the constraint of $\mathbb{F}_4$, due to $\eta_{ij} > 0$, the coefficients of the terms x_i^2 and x_j^2 in $f_{ij}(x_i, x_j)$ cannot be zero. Therefore, $\mathbb{F}_4$ cannot reduce to $\mathbb{F}_3$, and vice versa.

In $\mathbb{F}_2$, if $\varsigma_{ij} = \varsigma_{ji} = 0$, by comparing $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ in $\mathbb{F}_2$ with those in $\mathbb{F}_1$, we have $\xi_{ij}J_{ij}^2 = a_{ij}$, $\xi_{ji}J_{ij}^2 = a_{ji}$, $J_{ii} - \sum_{k\in\mathcal{B}_i}\xi_{ik}J_{ik}^2 = p_i$, therefore $\mathbb{F}_2$ becomes the form as $\mathbb{F}_1$. However, if we set $\varsigma_{ij} \neq 0$ or $\varsigma_{ji} \neq 0$, $f_{ij}(x_i, x_j)$ in $\mathbb{F}_2$ will never become the form as that in $\mathbb{F}_1$ since $f_{ij}(x_i, x_j)$ in $\mathbb{F}_1$ does not contain the first-order terms of variable x_i and x_j. On the other hand, in $\mathbb{F}_1$, if $a_{ij}a_{ji} - J_{ij}^2 < 0$, $\mathbb{F}_1$ will not satisfy the convex decomposition in $\mathbb{F}_2$, which implies $\mathbb{F}_1$ cannot be included in $\mathbb{F}_2$ for these cases. Therefore, $\mathbb{F}_1$ and $\mathbb{F}_2$ are overlapping, but one cannot include the other.

In $\mathbb{F}_1$, if $p_i, a_{ij}, a_{ji} > 0$ and $a_{ij}a_{ji} = J_{ij}^2$, $\mathbb{F}_1$ can be converted into $\mathbb{F}_4$ with $\zeta_i = p_i$, $y_i = h_i/p_i$, $\eta_{ij} = 1$, $z_{ij} = 0$, $c_{ij} = \sqrt{a_{ij}}$ and $c_{ji} = \sqrt{a_{ji}}$. But if $p_i < 0$ in $\mathbb{F}_1$, due to $\zeta_i \geq 0$ in $\mathbb{F}_4$, $f_i(x_i)$ in $\mathbb{F}_1$ cannot be converted into that in $\mathbb{F}_4$, which implies $\mathbb{F}_1$ is not included in $\mathbb{F}_4$. On the other hand, if $z_{ij} \neq 0$ in $\mathbb{F}_4$, $f_{ij}(x_i, x_j)$ in $\mathbb{F}_4$ contains the first-order terms of variable x_i and x_j while $f_{ij}(x_i, x_j)$ in $\mathbb{F}_1$ does not contain the first-order terms of x_i and x_j, which implies $\mathbb{F}_4$ cannot be included in $\mathbb{F}_1$. Therefore, $\mathbb{F}_1$ and $\mathbb{F}_4$ are overlapping, but one cannot include the other.

In $\mathbb{F}_4$, if $\zeta_i = 0$, $f_i(x_i)$ in $\mathbb{F}_4$ does not satisfy the condition $J_{ii} - \sum_{k\in\mathcal{B}_i}\xi_{ik}J_{ik}^2 > 0$ of $f_i(x_i)$ in $\mathbb{F}_2$, which implies $\mathbb{F}_4$ is not included in $\mathbb{F}_2$ for this case. But if $\zeta_i > 0$, then $\mathbb{F}_4$ can be converted into $\mathbb{F}_2$ with $J_{ii} - \sum_{k\in\mathcal{B}_i}\xi_{ik} = \zeta_i > 0$, $h_i - \sum_{k\in\mathcal{B}_i}\varsigma_{ik} = \zeta_i y_i$, $\xi_{ij}J_{ij}^2 = \eta_{ij}c_{ij}^2 > 0$, $\xi_{ji}\eta_{ij}^2 = \eta_{ij}c_{ji}^2 > 0$, $J_{ij} = \eta_{ij}c_{ij}c_{ji}$, $\varsigma_{ij} = \eta_{ij}z_{ij}c_{ij}$ and $\varsigma_{ji} = \eta_{ij}z_{ij}c_{ji}$. On the other hand, comparing $\mathbb{F}_2$ with $\mathbb{F}_4$, if $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ in $\mathbb{F}_2$ can be expressed by those in $\mathbb{F}_4$, we must have $\xi_{ij}J_{ij}^2 = \eta_{ij}c_{ij}^2$, $\xi_{ji}J_{ij}^2 = \eta_{ij}c_{ji}^2$, and $J_{ij} = \eta_{ij}c_{ij}c_{ji}$. From these conditions, we obtain that $\xi_{ij}\xi_{ji}J_{ij}^4 = \eta_{ij}^2c_{ij}^2c_{ji}^2 = J_{ij}^2$. Due to $\eta_{ij}, c_{ij}, c_{ji} \neq 0$, $J_{ij} \neq 0$ and the above result reduces to $\xi_{ij}\xi_{ji}J_{ij}^2 = 1$ if $\mathbb{F}_2$ can be expressed by $\mathbb{F}_4$. Hence, when $\xi_{ij}\xi_{ji}J_{ij}^2 > 1$ in $\mathbb{F}_2$, it implies that $\mathbb{F}_2$ cannot be expressed by $\mathbb{F}_4$. Therefore, $\mathbb{F}_2$ and $\mathbb{F}_4$ are overlapping, but one cannot include the other.

Finally, in $\mathbb{F}_4$, if $\zeta_i > 0$ for all $i = 1, 2, \cdots, n$ and $z_{ij} = 0$ for all $(i, j) \in \mathcal{E}$, then $f_{ij}(x_i, x_j)$ reduces to $f_{ij}(x_i, x_j) \propto \exp\left\{-\frac{1}{2}\eta_{ij}\left(c_{ij}x_i + c_{ji}x_j\right)^2\right\}$. Then, comparing $\mathbb{F}_4$ with $\mathbb{F}_1$, if we put $\zeta_i = p_i$, $y_i = h_i/p_i$, $\eta_{ij}c_{ij}^2 = a_{ij}$, $\eta_{ij}c_{ji}^2 = a_{ji}$ and $\eta_{ij}c_{ij}c_{ji} = J_{ij}$, then $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ in $\mathbb{F}_4$ become those in $\mathbb{F}_1$. On the other hand, comparing $\mathbb{F}_4$ with $\mathbb{F}_2$, if we put $\zeta_i = J_{ii} - \sum_{k\in\mathcal{B}_i}\xi_{ik}J_{ik}^2$, $\zeta_i y_i = h_i - \sum_{k\in\mathcal{B}_i}\varsigma_{ik}$, $\eta_{ij}c_{ij}^2 = \xi_{ij}J_{ij}^2$, $\eta_{ij}c_{ji}^2 = \xi_{ji}J_{ij}^2$, $\eta_{ij}c_{ij}c_{ji} = J_{ij}$ and $\varsigma_{ij} = \varsigma_{ji} = 0$, then $f_i(x_i)$ and $f_{ij}(x_i, x_j)$ in $\mathbb{F}_4$ reduce to those in $\mathbb{F}_2$ and satisfy the convex decomposition. Hence, we found a model that is a special case of $\mathbb{F}_4$, but at the same time can be expressed in the form of $\mathbb{F}_1$ and $\mathbb{F}_2$. Therefore, there exists an intersection of $\mathbb{F}_1$, $\mathbb{F}_2$ and $\mathbb{F}_4$.

Appendix B. Derivation of Gaussian BP Messages

By substituting the factors of $\mathbb{F}_5$ and (8) into (7), we obtain the expression

$$m_{j\to i}^{(l)}(x_i) \propto \exp\left\{-\frac{1}{2}\gamma_{ij}x_i^2 + \kappa_{ij}x_i\right\} \times \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\left(\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)}\right)x_j^2 + \left(\psi_j + \kappa_{ji} - \tau_{ij}x_i + \sum_{k\in\mathcal{B}_j\setminus i}\beta_{k\to j}^{(l-1)}\right)x_j\right\}dx_j. \tag{32}$$

For the integration in (32), it remains finite if and only if $\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)} > 0$. If the condition $\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)} > 0$ is satisfied, by performing the integration in (32), we have

$$m_{j\to i}^{(l)}(x_i) \propto \exp\left\{-\frac{1}{2}\underbrace{\left(\gamma_{ij} - \frac{\tau_{ij}^2}{\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)}}\right)}_{\alpha_{j\to i}^{(l)}}x_i^2 + \underbrace{\left(\kappa_{ij} - \frac{\tau_{ij}\psi_j + \tau_{ij}\kappa_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\tau_{ij}\beta_{k\to j}^{(l-1)}}{\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)}}\right)}_{\beta_{j\to i}^{(l)}}x_i\right\}, \tag{33}$$

where the message $m_{j\to i}^{(l)}(x_i)$ is maintained the same form as in (8) and parameterized by a pair of parameters $\alpha_{j\to i}^{(l)}$ and $\beta_{j\to i}^{(l)}$. If $\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(l-1)} \leq 0$, the message $m_{j\to i}^{(l)}(x_i)$ is not defined.

Appendix C. Proof of Theorem 1

From (9) and (10), the message parameters update can be rewritten as

$$\alpha_{j\to i}^{(l)} - \gamma_{ij} = -\frac{\tau_{ij}^2}{\phi_j + \sum\limits_{k\in\mathcal{B}_j}\gamma_{jk} + \sum\limits_{k\in\mathcal{B}_j\setminus i}(\alpha_{k\to j}^{(l-1)} - \gamma_{jk})}, \tag{34}$$

$$\beta_{j\to i}^{(l)} - \kappa_{ij} = -\frac{\tau_{ij}\psi_j + \tau_{ij}\sum\limits_{k\in\mathcal{B}_j}\kappa_{jk} + \sum\limits_{k\in\mathcal{B}_j\setminus i}\tau_{ij}(\beta_{k\to j}^{(l-1)} - \kappa_{jk})}{\phi_j + \sum\limits_{k\in\mathcal{B}_j}\gamma_{jk} + \sum\limits_{k\in\mathcal{B}_j\setminus i}(\alpha_{k\to j}^{(l-1)} - \gamma_{jk})}. \tag{35}$$

Furthermore, from (13) and (14), the belief parameters update can be rewritten as $\nu_{x_i}^{(l)} = \phi_i + \sum_{j\in\mathcal{B}_i}\gamma_{ij} + \sum_{j\in\mathcal{B}_i}(\alpha_{j\to i}^{(l)} - \gamma_{ij})$ and $\varphi_{x_i}^{(l)} = \psi_i + \sum_{j\in\mathcal{B}_i}\kappa_{ij} + \sum_{j\in\mathcal{B}_i}(\beta_{j\to i}^{(l)} - \kappa_{ij})$. By taking $\alpha_{j\to i}^{(l)} - \gamma_{ij}$ and $\beta_{j\to i}^{(l)} - \kappa_{ij}$ as two new sequences in the updates of messages and beliefs, it is observed that the updates of messages and beliefs are in the same form as those in Su and Wu (2014) if we set $\phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} = J_{ii}$, $\tau_{ij} = J_{ij}$ for $i \neq j$ and $\psi_i + \sum_{k\in\mathcal{B}_i}\kappa_{ik} = h_i$. By using Su and Wu (2014, Theorem 4), under any initialization satisfying $\boldsymbol{\alpha}^{(0)} - \boldsymbol{\gamma} \geq \mathbf{0}$, we obtain that belief variances $\{1/\nu_{x_i}^{(l)}\}$ converge to the same positive values if and only if the set $\widetilde{\mathcal{Q}} = \{\mathbf{q} \in \mathbb{R}^{|\mathcal{E}|} \mid J_{ii} + \sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i} > 0, -\frac{J_{ij}^2}{J_{ii}+\sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i}} \geq q_{i\to j}, \forall (i,j) \in \mathcal{E}$, and $J_{ii} + \sum_{k\in\mathcal{B}_i} q_{k\to i} > 0$ for $i = 1, 2, \cdots, n\}$ is non-empty. Furthermore, by Su and Wu (2014, Lemma 3), the constraint $J_{ii} + \sum_{k\in\mathcal{B}_i} q_{k\to i} > 0$ will be either satisfied for all $i = 1, 2, \cdots, n$ or dissatisfied for all $i = 1, 2, \cdots, n$. Without loss of generality, checking $i = 1$ is sufficient. By putting $\phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} = J_{ii}$, $\tau_{ij} = J_{ij}$ for $i \neq j$ into $\widetilde{\mathcal{Q}}$, it leads to the set $\mathcal{Q} \triangleq \{\mathbf{q} \in \mathbb{R}^{|\mathcal{E}|} \mid \phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i} > 0, -\frac{\tau_{ij}^2}{\phi_i+\sum_{k\in\mathcal{B}_i}\gamma_{ik}+\sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i}} \geq q_{i\to j}, \forall (i,j) \in \mathcal{E}$, and $\phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i} q_{k\to i} > 0$ for $i = 1\}$. Furthermore, if the set $\mathcal{Q}$ is non-empty, $\boldsymbol{\alpha}^{(l)}$ converges, and the update of $\boldsymbol{\beta}^{(l)}$ becomes the linear update equation in (16). With Bertsekas and Tsitsiklis (1989, Proposition 2.6.1), $\boldsymbol{\beta}^{(l)}$ converges to a unique value if and only if $\rho(\mathbf{A}) < 1$ under synchronous scheduling. With a similar proof to Su and Wu (2015a, Theorem 3), belief means converge to unique values if and only if $\boldsymbol{\beta}^{(l)}$ converges to a unique value. Therefore, belief means converge to unique values if and only if $\rho(\mathbf{A}) < 1$ under synchronous scheduling.

Appendix D. Proof of Theorem 2

By the proof in Theorem 1, it is shown that the update of beliefs is in the same form as that in Su and Wu (2014) if we set $f_i(x_i) \propto \exp\{-\frac{1}{2}(\phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik})x_i^2 + (\psi_i + \sum_{k\in\mathcal{B}_i}\kappa_{ik})x_i\}$ and $f_{ij}(x_i, x_j) \propto \exp\{-\tau_{ij}x_ix_j\}$. By Su and Wu (2014, Theorem 4), under any initialization $\boldsymbol{\alpha}^{(0)} - \boldsymbol{\gamma} \geq \mathbf{0}$, belief variances converge to the same positive values under totally asynchronous scheduling if and only if $\mathcal{Q}$ is non-empty. Moreover, if $\mathcal{Q}$ is non-empty, $\boldsymbol{\alpha}^{(l)}$ converges and thus the update of $\boldsymbol{\beta}^{(l)}$ becomes a linear equation as shown in (16). According to the asynchronous convergence theorem in Bertsekas and Tsitsiklis (1989, Proposition 6.2.1), $\boldsymbol{\beta}^{(l)}$ converge to a unique value if $\rho(|\mathbf{A}|) < 1$ under totally asynchronous scheduling. Furthermore, with the converged value of x_i's belief variance being $1/\nu_{x_i}^*$, the x_i's belief

mean at the l-th iteration is computed as $\varphi_{x_i}^{(l)}/\nu_{x_i}^* = (\psi_i + \sum_{j\in\mathcal{B}_i}\beta_{j\to i}^{(l)})/\nu_{x_i}^*$. If $\boldsymbol{\beta}^{(l)}$ converges, belief means would also converge. Therefore, belief means converge to unique values if $\rho(|\mathbf{A}|) < 1$ under totally asynchronous scheduling.

Appendix E. Proof of Corollary 3

Since synchronous scheduling is a special case of totally asynchronous scheduling, we only need to prove the result under totally asynchronous scheduling. We define a positive integer set $\mathcal{T}_{j\to i}$ denoting the iteration indices when the message $m_{j\to i}^{(l)}(x_i)$ is updated under totally asynchronous scheduling. If $l \in \mathcal{T}_{j\to i}$, we have the update of $\alpha_{j\to i}^{(l)}$ as

$$\alpha_{j\to i}^{(l)} = \gamma_{ij} - \frac{\tau_{ij}^2}{\phi_j + \gamma_{ji} + \sum\limits_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l-1))}}, \tag{36}$$

where $0 \leq T_{k\to j}^{j\to i}(l-1) \leq l-1$ and $\alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l-1))}$ denotes the most recently received value of $\alpha_{k\to j}$ for the update of $\alpha_{j\to i}$ at the l-th iteration. Otherwise, $\alpha_{j\to i}^{(l)}$ will be maintained as the previous value at the $(l-1)$-th iteration.

<u>Necessary condition:</u>

We need to prove that if the set $\mathcal{Q}$ is non-empty, $\nu_{x_i}^{(l)} > 0$ for all $l \geq 0$. Firstly, we can rewrite the update of $\nu_{x_i}^{(l)}$ in (25) as $\nu_{x_i}^{(l)} = \phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i}(\alpha_{k\to i}^{(l)} - \gamma_{ik})$. If we can prove that $\alpha_{k\to i}^{(l)} - \gamma_{ik} > q_{k\to i}$, where $q_{k\to i}$ is the component of any $\mathbf{q} \in \mathcal{Q}$, then $\nu_{x_i}^{(l)} > \phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} + \sum_{k\in\mathcal{B}_i} q_{k\to i} > 0$, where the last inequality is due to the third constraint of $\mathcal{Q}$. Hence, our task is to prove $\alpha_{k\to i}^{(l)} - \gamma_{ik} > q_{k\to i}$ for all $l \geq 0$. This is done by induction. First, consider $l = 0$. By putting the first constraint of $\mathcal{Q}$ into its second constraint, we must have $q_{i\to j} \leq -\frac{\tau_{ij}^2}{\phi_i+\sum_{k\in\mathcal{B}_i}\gamma_{ik}+\sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i}} < 0$. On the other hand, under any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, we have $\alpha_{i\to j}^{(0)} - \gamma_{ji} \geq 0$. Combining with $q_{i\to j} < 0$, we obtain $\alpha_{i\to j}^{(0)} - \gamma_{ji} > q_{i\to j}$. Therefore, the statement is true for $l = 0$. Now, suppose that $\alpha_{i\to j}^{(l-1)} - \gamma_{ji} > q_{i\to j}$ is true. On one hand, if $l \notin \mathcal{T}_{i\to j}$, then $\alpha_{i\to j}^{(l)}$ is not updated at the l-th iteration and $\alpha_{i\to j}^{(l)} = \alpha_{i\to j}^{(l-1)}$. Hence, we can directly obtain $\alpha_{i\to j}^{(l)} - \gamma_{ji} > q_{i\to j}$. On the other hand, if $l \in \mathcal{T}_{i\to j}$, then $\alpha_{i\to j}^{(l)}$ is updated at the l-th iteration. Due to the assumption $\alpha_{i\to j}^{(l-1)} - \gamma_{ji} > q_{i\to j}$ and $T_{k\to i}^{i\to j}(l-1) \leq l-1$, we have $\alpha_{k\to i}^{(T_{k\to i}^{i\to j}(l-1))} - \gamma_{ik} > q_{k\to i}$ and then $\phi_i + \gamma_{ij} + \sum_{k\in\mathcal{B}_i\setminus j}\alpha_{k\to i}^{(T_{k\to i}^{i\to j}(l-1))} > \phi_i + \gamma_{ij} + \sum_{k\in\mathcal{B}_i\setminus j}(q_{k\to i} + \gamma_{ik})$. Furthermore, due to the third constraint of $\mathcal{Q}$, we can obtain $\phi_i + \gamma_{ij} + \sum_{k\in\mathcal{B}_i\setminus j}\alpha_{k\to i}^{(T_{k\to i}^{i\to j}(l-1))} > \phi_i + \gamma_{ij} + \sum_{k\in\mathcal{B}_i\setminus j}(q_{k\to i} + \gamma_{ik}) > 0$, which implies $-\frac{\tau_{ij}^2}{\phi_i+\gamma_{ij}+\sum_{k\in\mathcal{B}_i\setminus j}\alpha_{k\to i}^{(T_{k\to i}^{i\to j}(l-1))}} > -\frac{\tau_{ij}^2}{\phi_i+\sum_{k\in\mathcal{B}_i}\gamma_{ik}+\sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i}}$. Since we have $-\frac{\tau_{ij}^2}{\phi_i+\sum_{k\in\mathcal{B}_i}\gamma_{ik}+\sum_{k\in\mathcal{B}_i\setminus j} q_{k\to i}} \geq q_{i\to j}$ due to the second constraint of $\mathcal{Q}$, we can obtain $-\frac{\tau_{ij}^2}{\phi_i+\gamma_{ij}+\sum_{k\in\mathcal{B}_i\setminus j}\alpha_{k\to i}^{(T_{k\to i}^{i\to j}(l-1))}} > q_{i\to j}$. Noticing that the left hand side of this equation is

simply $\alpha_{i\to j}^{(l)} - \gamma_{ji}$ (see (36)) under totally asynchronous scheduling, we obtain $\alpha_{i\to j}^{(l)} - \gamma_{ji} > q_{i\to j}$. Therefore, we have proved $\alpha_{k\to i}^{(l)} - \gamma_{ik} > q_{k\to i}$ for all $l \geq 0$.

Sufficient condition:

We need to prove that if BP beliefs stay as valid Gaussian pdfs at all iterations, the set $\mathcal{Q}$ will be non-empty. When BP beliefs stay as valid Gaussian pdfs at all iterations under totally asynchronous scheduling, we can obtain $\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l))} > 0$ and $\phi_i + \sum_{k\in\mathcal{B}_i} \alpha_{k\to i}^{(l)} > 0$ for all $l \geq 0$. When $l = 1$, if $1 \in \mathcal{T}_{j\to i}$, then $\alpha_{j\to i}^{(l)}$ is updated. Due to $\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(0))} > 0$, further by the update equation (36), we can obtain $\alpha_{j\to i}^{(1)} = \gamma_{ij} - \frac{\tau_{ij}^2}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(0))}} < \gamma_{ij}$. Due to $\alpha_{j\to i}^{(0)} \geq \gamma_{ij}$, we have $\alpha_{j\to i}^{(1)} < \alpha_{j\to i}^{(0)}$. If $1 \notin \mathcal{T}_{j\to i}$, we have $\alpha_{j\to i}^{(1)} = \alpha_{j\to i}^{(0)}$. Therefore, $\alpha_{j\to i}^{(l)} \leq \alpha_{j\to i}^{(l-1)}$ is true for $l = 1$. Suppose that $\alpha_{j\to i}^{(l)} \leq \alpha_{j\to i}^{(l-1)}$ is true. If $l+1 \notin \mathcal{T}_{j\to i}$, $\alpha_{j\to i}^{(l+1)}$ is not updated and $\alpha_{j\to i}^{(l+1)} = \alpha_{j\to i}^{(l)}$. If $l+1 \in \mathcal{T}_{j\to i}$, then $\alpha_{j\to i}^{(l+1)}$ is updated. We divide the discussion into two cases:

1) If $\alpha_{j\to i}^{(l)}$ is updated for the first time at the $(l+1)$-th iteration, we have $\alpha_{j\to i}^{(l)} = \alpha_{j\to i}^{(0)}$. From (36), we can obtain $\alpha_{j\to i}^{(l+1)} = \gamma_{ij} - \frac{\tau_{ij}^2}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{i\to j}^{j\to i}(l))}} < \gamma_{ij}$. Due to $\alpha_{j\to i}^{(0)} > \gamma_{ij}$, we can obtain $\alpha_{j\to i}^{(l+1)} < \alpha_{j\to i}^{(l)}$.

2) If $\alpha_{j\to i}^{(l)}$ has been updated before the $(l+1)$-th iteration, we suppose that l_1 is the most recent iteration for the update of $\alpha_{j\to i}^{(l)}$, where $l_1 \leq l$. Since $T_{k\to j}^{j\to i}(l) \geq T_{k\to j}^{j\to i}(l_1 - 1)$ and due to the assumption $\alpha_{k\to j}^{(l)} \leq \alpha_{k\to j}^{(l-1)}$, we can obtain $\alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l))} \leq \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l_1-1))}$. Further due to $\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l))} > 0$ and $\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l_1-1))} > 0$, we can obtain $\gamma_{ij} - \frac{\tau_{ij}^2}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l))}} \leq \gamma_{ij} - \frac{\tau_{ij}^2}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i} \alpha_{k\to j}^{(T_{k\to j}^{j\to i}(l_1-1))}}$, i.e., $\alpha_{j\to i}^{(l+1)} \leq \alpha_{j\to i}^{(l_1)}$. Moreover, since $\alpha_{j\to i}^{(l)}$ is not updated after the l_1-th iteration and before the $(l+1)$-th iteration, we have $\alpha_{j\to i}^{(l)} = \alpha_{j\to i}^{(l_1)}$. Hence, we can obtain $\alpha_{j\to i}^{(l+1)} \leq \alpha_{j\to i}^{(l)}$.

Therefore, $\alpha_{j\to i}^{(l)}$ is a monotonically decreasing sequence under totally asynchronous scheduling.

Furthermore, if $\alpha_{j\to i}^{(l)}$ is not bounded below, then it goes to $-\infty$ when the iterations goes to infinity, which leads to $\phi_i + \sum_{k\in\mathcal{B}_i} \alpha_{k\to i}^{(l)} < 0$ when l goes to infinity. This leads to a contradiction with the condition $\phi_i + \sum_{k\in\mathcal{B}_i} \alpha_{k\to i}^{(l)} > 0$ for all $l \geq 0$. Therefore, $\alpha_{j\to i}^{(l)}$ must be bounded below. Together with the monotonically decreasing property of $\alpha_{j\to i}^{(l)}$, $\alpha_{j\to i}^{(l)}$ converges. Here, we suppose that $\boldsymbol{\alpha}^*$ is the converged value of $\boldsymbol{\alpha}^{(l)}$. Defining $\alpha_{j\to i}^* = q_{j\to i} + \gamma_{ij}$, and putting it into the condition guaranteeing the validity of integration in (32) and $\nu_{x_i}^{(l)} > 0$, we can obtain $\phi_i+\gamma_{ij}+\sum_{k\in\mathcal{B}_i\setminus j}(q_{k\to i}+\gamma_{ik}) > 0$ and $\phi_i+\sum_{k\in\mathcal{B}_i}(q_{k\to i}+\gamma_{ik}) > 0$ for $l \geq 0$. Moreover, putting $\alpha_{j\to i}^* = q_{j\to i} + \gamma_{ij}$ into the update equation (36), we have

$q_{j\to i} = -\frac{\tau_{ij}^2}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i}(q_{k\to j}+\gamma_{jk})}$. But these are in fact the three constraints of the set $\mathcal{Q}$. Therefore, we have found a $\mathbf{q} \in \mathcal{Q}$ with the element $q_{j\to i} = \alpha_{j\to i}^* - \gamma_{ij}$, and the set $\mathcal{Q}$ is non-empty.

Appendix F. Proof of Lemma 5

For synchronous scheduling, by Corollary 3, under any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, if the set $\mathcal{Q}$ is non-empty, BP beliefs stay being valid Gaussian pdfs. Furthermore, from the proof of sufficient condition in Corollary 3, it is proved that BP beliefs stay in valid Gaussian form leads to $\boldsymbol{\alpha}^{(l)}$ converges. Then, the matrix $\mathbf{A}$ consists of elements $\frac{\tau_{ij}}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^*}$ for all $(i,j) \in \mathcal{E}$, where $\boldsymbol{\alpha}^*$ is the converged value of $\boldsymbol{\alpha}^{(l)}$. Since $\tau_{ij} = J_{ij}$ and $\phi_j + \gamma_{ji} + \sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^* = \phi_j + \sum_{k\in\mathcal{B}_j}\gamma_{jk} + \sum_{k\in\mathcal{B}_j\setminus i}(\alpha_{k\to j}^* - \gamma_{jk}) = J_{jj} + \sum_{k\in\mathcal{B}_j\setminus i}(\alpha_{k\to j}^* - \gamma_{jk})$ for a Gaussian pdf with information matrix $\mathbf{J}$, the elements of $\mathbf{A}$ become $\frac{J_{ij}}{J_{jj}+\sum_{k\in\mathcal{B}_j\setminus i}(\alpha_{k\to j}^*-\gamma_{jk})}$ for all $(i,j) \in \mathcal{E}$. On the other hand, from (34), if we treat $\alpha_{j\to i}^{(l)} - \gamma_{ij} \triangleq \widetilde{\alpha}_{j\to i}^{(l)}$ as a whole and putting $\phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} = J_{ii}$ and $\tau_{ij} = J_{ij}$, the resultant equation becomes (12) in Su and Wu (2014). Furthermore, putting $\phi_i + \sum_{k\in\mathcal{B}_i}\gamma_{ik} = J_{ii}$ and $\tau_{ij} = J_{ij}$, the set $\mathcal{Q}$ reduces to $\widetilde{\mathcal{Q}}$ defined in the proof of Theorem 1. Therefore, non-empty $\mathcal{Q}$ implies that $\widetilde{\mathcal{Q}}$ is non-empty. By Su and Wu (2014, Theorem 2), under any initialization $\widetilde{\boldsymbol{\alpha}}^{(0)} \geq \mathbf{0}$ and non-empty $\widetilde{\mathcal{Q}}$, $\widetilde{\boldsymbol{\alpha}}^{(l)}$ converges to the same value $\widetilde{\boldsymbol{\alpha}}^*$. Recognizing that $\widetilde{\boldsymbol{\alpha}}^* = \boldsymbol{\alpha}^* - \boldsymbol{\gamma}$ is the same for any $\boldsymbol{\gamma}$, we can conclude that under synchronous scheduling, the elements of matrix $\mathbf{A}$ are independent of factorization parameters for any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$. Since the matrix $\mathbf{A}$ in totally asynchronous scheduling is the same as that in synchronous scheduling, Lemma 5 also holds for totally asynchronous scheduling.

Appendix G. Proof of Corollary 6

For a Gaussian pdf with information matrix $\mathbf{J}$, under any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, by Lemmas 4 and 5, $\mathcal{Q}$ and $\mathbf{A}$ are independent of factorization parameters in $\mathbb{F}_5$. Furthermore, by Theorem 1, belief variances converge if and only if the set $\mathcal{Q}$ is non-empty and belief means converge if and only if $\rho(\mathbf{A}) < 1$. Therefore, the convergence or divergence of Gaussian BP under synchronous scheduling is independent of factorization parameters in $\mathbb{F}_5$.

Appendix H. Proof of Corollary 7

Since synchronous scheduling is a special case of totally asynchronous scheduling, proving the result under totally asynchronous scheduling is sufficient.

Under totally asynchronous scheduling, walk-summability guarantees that the set $\mathcal{Q}$ is non-empty and $\rho(|\mathbf{A}|) < 1$ under pairwise factorization $\mathbb{F}_3$ and initialization $\boldsymbol{\alpha}^{(0)} \geq \mathbf{0}$ (Su and Wu, 2015a, Theorem 6). Using Lemma 4, $\mathcal{Q}$ is non-empty under any pairwise factorization within $\mathbb{F}_5$. Further by Lemma 5, we have $\rho(|\mathbf{A}|) < 1$ under any pairwise factorization within $\mathbb{F}_5$ and any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$. Then applying Theorem 2, we obtain walk-summability leads to Gaussian BP convergence under any pairwise factorization within $\mathbb{F}_5$ and any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$. Since walk-summability is equivalent to con-

vex decomposition (Moallemi and Roy, 2009) or pairwise-normalizability (Malioutov et al., 2006), and includes diagonal dominance as a special case (Malioutov et al., 2006), Gaussian BP converges under any one of these conditions, and the convergence result is valid under any pairwise factorization within $\mathbb{F}_5$ and any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$.

Appendix I. Proof of Theorem 8

As given in the proof of Lemma 5, for a Gaussian pdf with information matrix $\mathbf{J}$, under any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, if Gaussian beliefs converge, then $\widetilde{\boldsymbol{\alpha}}^* = \boldsymbol{\alpha}^* - \boldsymbol{\gamma}$ is the same for any $\boldsymbol{\gamma}$. On the other hand, when belief variances converge under synchronous scheduling, we have $\boldsymbol{\beta}^{(l)} = \mathbf{A}\boldsymbol{\beta}^{(l-1)} + \mathbf{d}$. By taking $\beta_{j\to i}^{(l)} - \kappa_{ij} \triangleq \widetilde{\beta}_{j\to i}^{(l)}$, we have the update equation $\widetilde{\boldsymbol{\beta}}^{(l)} = \mathbf{A}\widetilde{\boldsymbol{\beta}}^{(l-1)} + \widetilde{\mathbf{d}}$, where $\widetilde{\mathbf{d}}$ contains the elements $-\frac{\tau_{ij}(\psi_j+\sum_{k\in\mathcal{B}_j}\kappa_{jk})}{\phi_j+\gamma_{ji}+\sum_{k\in\mathcal{B}_j\setminus i}\alpha_{k\to j}^*}$ for all $(i,j)\in\mathcal{E}$ ordered in the same way as that in $\boldsymbol{\alpha}^{(l)}$. Since $J_{jj} = \phi_j + \sum_{k\in\mathcal{B}_j}\gamma_{jk}$, $J_{ij} = \tau_{ij}$ and $\psi_j + \sum_{k\in\mathcal{B}_j}\kappa_{jk} = h_j$ for a Gaussian pdf with information matrix $\mathbf{J}$ and potential vector $\mathbf{h}$ (see (20) and (21)), the elements of $\widetilde{\mathbf{d}}$ become $-\frac{J_{ij}h_j}{J_{jj}+\sum_{k\in\mathcal{B}_j\setminus i}(\alpha_{k\to j}^*-\gamma_{jk})}$ for all $(i,j)\in\mathcal{E}$. Due to J_{jj}, J_{ij}, h_j are fixed and $\boldsymbol{\alpha}^* - \boldsymbol{\gamma}$ being the same under initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, the elements of $\widetilde{\mathbf{d}}$ are independent of factorization parameters in $\mathbb{F}_5$. Furthermore, from Theorem 1, when Gaussian BP beliefs converge under synchronous scheduling, we have $\rho(\mathbf{A}) < 1$. This leads to $\widetilde{\boldsymbol{\beta}}^{(l)}$ converge to $\widetilde{\boldsymbol{\beta}}^* = (\mathbf{I}-\mathbf{A})^{-1}\widetilde{\mathbf{d}}$ (Bertsekas and Tsitsiklis, 1989, Proposition 6.1). Since $\mathbf{A}$ and $\widetilde{\mathbf{d}}$ are independent of factorization parameters, we can obtain $\widetilde{\boldsymbol{\beta}}^* = \boldsymbol{\beta}^* - \boldsymbol{\kappa}$ is unique for any $\boldsymbol{\kappa}$. Moreover, Su and Wu (2015a, Theorem 5) has proved that the converged $\widetilde{\boldsymbol{\alpha}}^* = \boldsymbol{\alpha}^* - \boldsymbol{\gamma}$ and $\widetilde{\boldsymbol{\beta}}^* = \boldsymbol{\beta}^* - \boldsymbol{\kappa}$ under totally asynchronous scheduling and any initialization $\boldsymbol{\alpha}^{(0)} - \boldsymbol{\gamma} \geq \mathbf{0}$ are the same values as those under synchronous scheduling. Therefore, under any initialization $\boldsymbol{\alpha}^{(0)} \geq \boldsymbol{\gamma}$, the converged $\widetilde{\boldsymbol{\alpha}}^* = \boldsymbol{\alpha}^* - \boldsymbol{\gamma}$ and $\widetilde{\boldsymbol{\beta}}^* = \boldsymbol{\beta}^* - \boldsymbol{\kappa}$ are the same for different pairwise factorizations under both synchronous and totally asynchronous schedulings. Since $\phi_i + \sum_{j\in\mathcal{B}_i}\gamma_{ij} = J_{ii}$ and $\psi_i + \sum_{j\in\mathcal{B}_i}\kappa_{ij} = h_i$ are fixed for a Gaussian pdf with information matrix $\mathbf{J}$ and potential vector $\mathbf{h}$, and $\widetilde{\boldsymbol{\alpha}}^* = \boldsymbol{\alpha}^* - \boldsymbol{\gamma}$, $\widetilde{\boldsymbol{\beta}}^* = \boldsymbol{\beta}^* - \boldsymbol{\kappa}$ are unique when Gaussian BP converges, x_i's converged belief mean $\frac{\psi_i+\sum_{j\in\mathcal{B}_i}\beta_{j\to i}^*}{\phi_i+\sum_{j\in\mathcal{B}_i}\alpha_{j\to i}^*} = \frac{\psi_i+\sum_{j\in\mathcal{B}_i}\kappa_{ij}+\sum_{j\in\mathcal{B}_i}(\beta_{j\to i}^*-\kappa_{ij})}{\phi_i+\sum_{j\in\mathcal{B}_i}\gamma_{ij}+\sum_{j\in\mathcal{B}_i}(\alpha_{j\to i}^*-\gamma_{ij})} = \frac{h_i+\sum_{j\in\mathcal{B}_i}\widetilde{\beta}_{j\to i}^*}{J_{ii}+\sum_{j\in\mathcal{B}_i}\widetilde{\alpha}_{j\to i}^*}$ and belief variance $\frac{1}{\phi_i+\sum_{j\in\mathcal{B}_i}\alpha_{j\to i}^*} = \frac{1}{\phi_i+\sum_{j\in\mathcal{B}_i}\gamma_{ij}+\sum_{j\in\mathcal{B}_i}(\alpha_{j\to i}^*-\gamma_{ij})} = \frac{1}{J_{ii}+\sum_{j\in\mathcal{B}_i}\widetilde{\alpha}_{j\to i}^*}$ are unique and independent of factorization parameters under both synchronous and totally asynchronous schedulings.

Appendix J. Proof of Lemma 9

For any valid Gaussian pdf, $\mathbf{x}^T\mathbf{J}\mathbf{x}$ with $\mathbf{J}$ defined in (20) can be written as

$$\mathbf{x}^T\mathbf{J}\mathbf{x} = \sum_{i=1}^{n}(J_{ii} - \sum_{j\in\mathcal{B}_i}\omega_{ij})x_i^2 + \sum_{(i,j>i)\in\mathcal{E}}[x_i\ x_j]\begin{bmatrix}\omega_{ij} & J_{ij}\\ J_{ij} & \omega_{ji}\end{bmatrix}\begin{bmatrix}x_i\\ x_j\end{bmatrix} \text{ for all } \omega_{ij}. \tag{37}$$

While there are many possible ways to write (37) for an information matrix $\mathbf{J}$, the Gaussian model is pairwise-normalizable if there exists at least one decomposition (37) with $\omega_{ij} \geq 0$ for all $(i,j)\in\mathcal{E}$, $\omega_{ij}\omega_{ji} - J_{ij}^2 \geq 0$ for all $(i,j>i)\in\mathcal{E}$ and $J_{ii} - \sum_{j\in\mathcal{B}_i}\omega_{ij} > 0$ for all

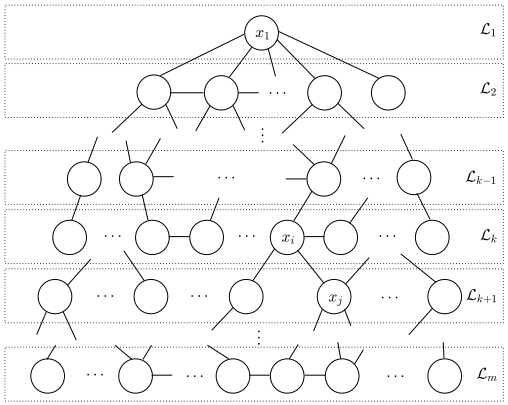


Figure 9: Clustering variables into multiple layers.

$i = 1, 2, \cdots, n$ (Koller and Friedman, 2009). Putting $J_{ii} = \phi_i + \sum_{j\in\mathcal{B}_i}\gamma_{ij}$ and $J_{ij} = \tau_{ij}$ into the above conditions, we have the constraints as in (23a)—(23c) if a Gaussian model is pairwise-normalizable.

Appendix K. Proof of Theorem 10

We put $\phi_i = \zeta_i$, $\gamma_{ij} = \eta_{ij}c_{ij}^2$, $\tau_{ij} = \eta_{ij}c_{ij}c_{ji}$ into (23a)—(23c) to reduce the conditions for the special case of pairwise linear Gaussian model $\mathbb{F}_4$:

$$\omega_{ij} \geq 0 \text{ for all } (i,j)\in\mathcal{E}, \tag{38a}$$

$$\omega_{ij}\omega_{ji} - \eta_{ij}^2c_{ij}^2c_{ji}^2 \geq 0 \text{ for all } (i,j>i)\in\mathcal{E}, \tag{38b}$$

$$\zeta_i + \sum_{j\in\mathcal{B}_i}\eta_{ij}c_{ij}^2 - \sum_{j\in\mathcal{B}_i}\omega_{ij} > 0, \text{ for all } i = 1, 2, \cdots, n. \tag{38c}$$

Below we will find ω_{ij} such that (38a)—(38c) are satisfied.

To find the ω_{ij}, we cluster all the variables into multiple layers as shown in Figure 9. Without loss of generality, we assume $\zeta_1 > 0$ and put x_1 into the first layer. Then we define a variable index set $\mathcal{L}_1 = \{1\}$ for variable x_1 in the first layer and the collection of variables indices in the second layer as a set $\mathcal{L}_2 = \{i \mid i\in\mathcal{B}_1\}$. In general, the set of variables' indices in the k-th layer is defined as $\mathcal{L}_k \triangleq \{i | i\in\mathcal{B}_t\setminus\{\mathcal{L}_{k-1}\cup\mathcal{L}_{k-2}\}, \text{ for all } t\in\mathcal{L}_{k-1}\}$ with $2\leq k\leq m$, where m is the number of total layers and $\mathcal{L}_1\cup\mathcal{L}_2\cup\cdots\cup\mathcal{L}_m = \{1, 2, \cdots, n\}$. Furthermore, based on the network in Figure 9, we can decompose the set $\mathcal{E} = \bigcup_{k=2}^{m}(\mathcal{E}_{k,k}\cup\mathcal{E}_{k-1,k}^{+}\cup\mathcal{E}_{k-1,k}^{-})$, where $\mathcal{E}_{k,k} \triangleq \{(i,j)\in\mathcal{E} \mid i,j\in\mathcal{L}_k\}$, $\mathcal{E}_{k-1,k}^{+} = \{(i,j)\in\mathcal{E} \mid i\in\mathcal{L}_{k-1}, j\in\mathcal{L}_k\}$, and $\mathcal{E}_{k-1,k}^{-} = \{(i,j)\in\mathcal{E} \mid i\in\mathcal{L}_k, j\in\mathcal{L}_{k-1}\}$. For the preparation of finding ω_{ij}, we first define ϵ_{ij} for all $(i,j)\in\mathcal{E}$ as follows.

For $(i,j) \in \mathcal{E}^-_{k-1,k}$ with $k \geq 2$, we define

$$\epsilon_{ij} = \begin{cases} -\frac{\eta_{i1}c_{i1}^2\zeta_1/(|\mathcal{B}_1|+1)}{\eta_{i1}c_{1i}^2+\zeta_1/(|\mathcal{B}_1|+1)} & \text{if } (i,j) \in \mathcal{E}^-_{1,2}, \\ \frac{\eta_{ij}c_{ij}^2\sum_{(j,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{jt}/(|\mathcal{B}_j\setminus\mathcal{L}_{k-2}|+1)}{\eta_{ij}c_{ji}^2-\sum_{(j,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{jt}/(|\mathcal{B}_j\setminus\mathcal{L}_{k-2}|+1)} & \text{otherwise.} \end{cases} \tag{39}$$

We can prove that $\epsilon_{ij} < 0$ for all $(i,j) \in \mathcal{E}^-_{k-1,k}$ with $k \geq 2$ by induction. When $k = 2$, we have $j = 1$. Due to $\eta_{i1} > 0$ and $\zeta_1 > 0$, we can obtain $\epsilon_{i1} = -\frac{\eta_{i1}c_{i1}^2\zeta_1/(|\mathcal{B}_1|+1)}{\eta_{i1}c_{1i}^2+\zeta_1/(|\mathcal{B}_1|+1)} < 0$. Therefore, $\epsilon_{i1} < 0$ is true for $k = 2$. Suppose $\epsilon_{ij} < 0$ is true for all $(i,j) \in \mathcal{E}^-_{k-1,k}$ for some $k \geq 2$. Then, for $(i,j) \in \mathcal{E}^-_{k,k+1}$, we have $\epsilon_{ij} = \frac{\eta_{ij}c_{ij}^2\sum_{(j,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{jt}/(|\mathcal{B}_j\setminus\mathcal{L}_{k-1}|+1)}{\eta_{ij}c_{ji}^2-\sum_{(j,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{jt}/(|\mathcal{B}_j\setminus\mathcal{L}_{k-1}|+1)}$. Since on the right hand side of this equation, $(j,t) \in \mathcal{E}^-_{k-1,k}$, by assumption, $\epsilon_{jt} < 0$. Furthermore, since $\eta_{ij} > 0$, we can obtain $\epsilon_{ij} < 0$. Therefore, we have proved $\epsilon_{ij} < 0$ for all $(i,j) \in \mathcal{E}^-_{k-1,k}$ with $k \geq 2$.

Based on ϵ_{ij} with $(i,j) \in \mathcal{E}^-_{k-1,k}$, we can define ϵ_{ij} for $(i,j) \in \mathcal{E}^+_{k-1,k}$ with $k \geq 2$ as

$$\epsilon_{ij} = \begin{cases} 0 & \text{if } (i,j) \in \mathcal{E}^+_{1,2}, \\ -\frac{\sum_{(i,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-2}|+1} & \text{otherwise.} \end{cases} \tag{40}$$

Furthermore, we define $\epsilon_{ij} = -\frac{\sum_{(i,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-1}|+1}$ for $(i,j) \in \mathcal{E}_{k,k}$ with $k \geq 2$.

Based on ϵ_{ij} for all $(i,j) \in \mathcal{E}$, we define $\omega_{ij} = \eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} + \epsilon_{ij}$ for all $(i,j) \in \mathcal{E}$. Then we will prove such defined ω_{ij} satisfies the conditions in (38a)—(38c).

Condition in (38a):

For $(i,j) \in \mathcal{E}^+_{k-1,k}$ with $k = 2$, we have $i = 1$. Due $\eta_{1j}, \zeta_1 > 0$ and $\epsilon_{1j} = 0$, we can obtain $\omega_{1j} = \eta_{1j}c_{1j}^2 + \frac{\zeta_1}{|\mathcal{B}_1|+1} + \epsilon_{1j} = \eta_{1j}c_{1j}^2 + \frac{\zeta_1}{|\mathcal{B}_1|+1} > 0$. For $(i,j) \in \mathcal{E}^+_{k-1,k}$ with $k > 2$, we have $\omega_{ij} = \eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} + \epsilon_{ij} = \eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-2}|+1}$. Due to $\epsilon_{it} < 0$ for $(i,t) \in \mathcal{E}^-_{k-2,k-1}$, and $\eta_{ij} > 0$, $\zeta_i \geq 0$, we have $\omega_{ij} > 0$.

For $(i,j) \in \mathcal{E}^-_{k-1,k}$ with $k = 2$, we have $j = 1$. We can obtain $\omega_{i1} = \eta_{i1}c_{i1}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} + \epsilon_{i1} = \eta_{i1}c_{i1}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\eta_{i1}c_{i1}^2\zeta_1/(|\mathcal{B}_1|+1)}{\eta_{i1}c_{1i}^2+\zeta_1/(|\mathcal{B}_1|+1)} = \frac{\zeta_i}{|\mathcal{B}_i|+1} + \frac{\eta_{i1}^2c_{i1}^2c_{1i}^2}{\eta_{i1}c_{1i}^2+\zeta_1/(|\mathcal{B}_1|+1)}$. Due to $\zeta_i \geq 0$ and $\eta_{i1} > 0$, we have $\omega_{i1} > 0$. For $(i,j) \in \mathcal{E}^-_{k-1,k}$ with $k > 2$, we have

$$\begin{aligned} \omega_{ij} &= \eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} + \epsilon_{ij} \\ &= \eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} + \frac{\eta_{ij}c_{ij}^2\sum_{(j,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{jt}/(|\mathcal{B}_j\setminus\mathcal{L}_{k-2}|+1)}{\eta_{ij}c_{ji}^2-\sum_{(j,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{jt}/(|\mathcal{B}_j\setminus\mathcal{L}_{k-2}|+1)} \\ &= \frac{\zeta_i}{|\mathcal{B}_i|+1} + \frac{\eta_{ij}^2c_{ij}^2c_{ji}^2}{\eta_{ij}c_{ji}^2-\sum_{(j,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{jt}/(|\mathcal{B}_j\setminus\mathcal{L}_{k-2}|+1)} > 0, \end{aligned} \tag{41}$$

due to $\zeta_i \geq 0$, $\eta_{ij} > 0$, and $\epsilon_{jt} < 0$ with $(j,t) \in \mathcal{E}^-_{k-2,k-1}$.

Furthermore, for $(i,j) \in \mathcal{E}_{k,k}$ with $k \geq 2$, we have $\omega_{ij} = \eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} + \epsilon_{ij} = \eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-1}|+1}$. Due to $\eta_{ij} > 0$, $\zeta_i \geq 0$ and $\epsilon_{it} < 0$ with $(i,t) \in \mathcal{E}^-_{k-1,k}$, we can obtain $\omega_{ij} > 0$.

Condition in (38b):

First notice that $(i, j > i) \in \mathcal{E}$ is equivalent to $(i,j) \in \bigcup_{k=2}^{m} \mathcal{E}^+_{k-1,k}$ or $(i, j > i) \in \bigcup_{k=2}^{m} \mathcal{E}_{k,k}$. Therefore, we only need to examine (38b) for the sets $(i,j) \in \mathcal{E}^+_{k-1,k}$ and $(i, j > i) \in \mathcal{E}_{k,k}$ for $k \geq 2$. For $(i,j) \in \mathcal{E}^+_{k-1,k}$ with $k = 2$, since $\eta_{1j}, \zeta_1 > 0$ and $\zeta_j \geq 0$, we can obtain $\omega_{1j}\omega_{j1} - \eta_{1j}^2c_{1j}^2c_{j1}^2 = (\eta_{1j}c_{1j}^2 + \frac{\zeta_1}{|\mathcal{B}_1|+1})(\frac{\zeta_j}{|\mathcal{B}_j|+1} + \frac{\eta_{j1}^2c_{j1}^2c_{1j}^2}{\eta_{j1}c_{1j}^2+\zeta_1/(|\mathcal{B}_1|+1)}) - \eta_{1j}^2c_{1j}^2c_{j1}^2 = (\eta_{1j}c_{1j}^2 + \frac{\zeta_1}{|\mathcal{B}_1|+1})\frac{\zeta_j}{|\mathcal{B}_j|+1} \geq 0$. For $(i,j) \in \mathcal{E}^+_{k-1,k}$ with $k > 2$, we have

$$\begin{aligned} \omega_{ij}\omega_{ji} - \eta_{ij}^2c_{ij}^2c_{ji}^2 &= \left(\eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-2}|+1}\right) \\ &\quad\times\left(\frac{\zeta_j}{|\mathcal{B}_j|+1} + \frac{\eta_{ij}^2c_{ij}^2c_{ji}^2}{\eta_{ij}c_{ij}^2-\sum_{(i,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{it}/(|\mathcal{B}_i\setminus\mathcal{L}_{k-2}|+1)}\right) - \eta_{ij}^2c_{ij}^2c_{ji}^2 \\ &= \left(\eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-2}|+1}\right)\frac{\zeta_j}{|\mathcal{B}_j|+1} \\ &\quad+\frac{\zeta_i}{|\mathcal{B}_i|+1}\frac{\eta_{ij}^2c_{ij}^2c_{ji}^2}{\eta_{ij}c_{ij}^2-\sum_{(i,t)\in\mathcal{E}^-_{k-2,k-1}}\epsilon_{it}/(|\mathcal{B}_i\setminus\mathcal{L}_{k-2}|+1)} \geq 0, \end{aligned} \tag{42}$$

due to $\eta_{ij} > 0$, $\zeta_i, \zeta_j \geq 0$, and $\epsilon_{it} < 0$ with $(i,t) \in \mathcal{E}^-_{k-2,k-1}$.

Furthermore, for $(i, j > i) \in \mathcal{E}_{k,k}$, we have

$$\begin{aligned} \omega_{ij}\omega_{ji} - \eta_{ij}^2c_{ij}^2c_{ji}^2 &= \left(\eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-1}|+1}\right)\left(\eta_{ij}c_{ji}^2 + \frac{\zeta_j}{|\mathcal{B}_j|+1} - \frac{\sum_{(j,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{jt}}{|\mathcal{B}_j\setminus\mathcal{L}_{k-1}|+1}\right) \\ &\quad-\eta_{ij}^2c_{ij}^2c_{ji}^2 \\ &= \left(\frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-1}|+1}\right)\eta_{ij}c_{ji}^2 \\ &\quad+\left(\eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{it}}{|\mathcal{B}_i\setminus\mathcal{L}_{k-1}|+1}\right)\left(\frac{\zeta_j}{|\mathcal{B}_j|+1} - \frac{\sum_{(j,t)\in\mathcal{E}^-_{k-1,k}}\epsilon_{jt}}{|\mathcal{B}_j\setminus\mathcal{L}_{k-1}|+1}\right). \end{aligned} \tag{43}$$

Due to $\eta_{ij} > 0$, $\zeta_i, \zeta_j \geq 0$ and $\epsilon_{it}, \epsilon_{jt} < 0$ with $(i,t), (j,t) \in \mathcal{E}^-_{k-1,k}$, we can obtain $\omega_{ij}\omega_{ji} - \eta_{ij}^2c_{ij}^2c_{ji}^2 > 0$.

Condition in (38c):

For $i = 1$, we have $\zeta_1 + \sum_{j\in\mathcal{B}_1} \eta_{1j}c_{1j}^2 - \sum_{j\in\mathcal{B}_1} \omega_{1j} = \zeta_1 + \sum_{j\in\mathcal{B}_1} \eta_{1j}c_{1j}^2 - \sum_{j\in\mathcal{B}_1}(\eta_{1j}c_{1j}^2 + \frac{\zeta_1}{|\mathcal{B}_1|+1}) = \frac{\zeta_1}{|\mathcal{B}_1|+1} > 0$. For $i \in \mathcal{L}_k$, we have

$$\begin{aligned}
\zeta_i + \sum_{j\in\mathcal{B}_i} \eta_{ij}c_{ij}^2 - \sum_{j\in\mathcal{B}_i} \omega_{ij} =& \zeta_i + \sum_{j\in\mathcal{B}_i} \eta_{ij}c_{ij}^2 - \sum_{j\in\mathcal{B}_i\cap\mathcal{L}_{k-1}} \omega_{ij} - \sum_{j\in\mathcal{B}_i\setminus\mathcal{L}_{k-1}} \omega_{ij} \\
=& \zeta_i + \sum_{j\in\mathcal{B}_i} \eta_{ij}c_{ij}^2 - \sum_{j\in\mathcal{B}_i\cap\mathcal{L}_{k-1}} (\eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} + \epsilon_{ij}) \\
& - \sum_{j\in\mathcal{B}_i\setminus\mathcal{L}_{k-1}} (\eta_{ij}c_{ij}^2 + \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}_{k-1,k}^-} \epsilon_{it}}{|\mathcal{B}_i \setminus \mathcal{L}_{k-1}|+1}) \\
=& \frac{\zeta_i}{|\mathcal{B}_i|+1} - \frac{\sum_{(i,t)\in\mathcal{E}_{k-1,k}^-} \epsilon_{it}}{|\mathcal{B}_i \setminus \mathcal{L}_{k-1}|+1}. \qquad (44)
\end{aligned}$$

Due to $\zeta_i \geq 0$, $\epsilon_{it} < 0$ with $(i,t) \in \mathcal{E}_{k-1,k}^-$, and $\sum_{j\in\mathcal{B}_i\cap\mathcal{L}_{k-1}} \epsilon_{ij} = \sum_{(i,t)\in\mathcal{E}_{k-1,k}^-} \epsilon_{it}$, we obtain $\zeta_i + \sum_{j\in\mathcal{B}_i} \eta_{ij}c_{ij}^2 - \sum_{j\in\mathcal{B}_i} \omega_{ij} > 0$.

Hence, we have found ω_{ij} satisfying (38a)—(38c). Therefore, pairwise factorization $\mathbb{F}_4$ with at least one $\zeta_i > 0$ is pairwise-normalizable. By Corollary 7, Gaussian BP converges in both synchronous and totally asynchronous schedulings.

References

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice Hall, 1989.

D. Bickson, D. Malkhi, and L. Zhou. Peer-to-peer rating. In *IEEE International Conference on Peer-to-Peer Computing*, pages 211–218. IEEE, 2007.

D. Bickson, D. Dolev, O. Shental, P. H. Siegel, and J. K. Wolf. Gaussian belief propagation based multiuser detection. In *IEEE International Symposium on Information Theory*, pages 1878–1882. IEEE, 2008.

J. Du and Y. C. Wu. Network-wide distributed carrier frequency offsets estimation and compensation via belief propagation. *IEEE Transactions on Signal Processing*, 61(23): 5868–5877, 2013.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

M. Leng and Y. C. Wu. Distributed clock synchronization for wireless sensor networks using belief propagation. *IEEE Transactions on Signal Processing*, 59(11):5404–5414, 2011.

D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 7(1):2031–2064, 2006.

C. C. Moallemi and B. V. Roy. Consensus propagation. *IEEE Transactions on Information Theory*, 52(11):4753–4766, 2006.

C. C. Moallemi and B. V. Roy. Convergence of min-sum message passing for quadratic optimization. *IEEE Transactions on Information Theory*, 55(5):2413–2423, 2009.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

N. Ruozzi and S. Tatikonda. Message-passing algorithms for quadratic minimization. *Journal of Machine Learning Research*, 14(1):2287–2314, 2013.

O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson, and D. Dolev. Gaussian belief propagation solver for systems of linear equations. In *IEEE International Symposium on Information Theory*, pages 1863–1867. IEEE, 2008.

Q. Su and Y. C. Wu. Convergence analysis of the variance in Gaussian belief propagation. *IEEE Transactions on Signal Processing*, 62(19):5119–5131, 2014.

Q. Su and Y. C. Wu. On convergence conditions of Gaussian belief propagation. *IEEE Transactions Signal Processing*, 63(5):1144–1155, 2015a.

Q. Su and Y. C. Wu. Distributed estimation of variance in Gaussian graphical model via belief propagation: Accuracy analysis and improvement. *IEEE Transactions on Signal Processing*, 63(23):6258–6271, 2015b.

L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In *Advances in Neural Information Processing Systems*, pages 673–679, 2000.

Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3-4):203–230, 2010.

Journal of Machine Learning Research 20 (2019) 1-30 Submitted 4/18; Revised 3/19; Published 9/19

Bayesian Optimization for Policy Search via Online-Offline Experimentation

Benjamin Letham BLETHAM@FB.COM

Eytan Bakshy EYTAN@FB.COM

Facebook
Menlo Park, California, USA

Editor: Bayesian Optimization Special Issue

Abstract

Online field experiments are the gold-standard way of evaluating changes to real-world interactive machine learning systems. Yet our ability to explore complex, multi-dimensional policy spaces—such as those found in recommendation and ranking problems—is often constrained by the limited number of experiments that can be run simultaneously. To alleviate these constraints, we augment online experiments with an offline simulator and apply multi-task Bayesian optimization to tune live machine learning systems. We describe practical issues that arise in these types of applications, including biases that arise from using a simulator and assumptions for the multi-task kernel. We measure empirical learning curves which show substantial gains from including data from biased offline experiments, and show how these learning curves are consistent with theoretical results for multi-task Gaussian process generalization. We find that improved kernel inference is a significant driver of multi-task generalization. Finally, we show several examples of Bayesian optimization efficiently tuning a live machine learning system by combining offline and online experiments.

Keywords: Bayesian optimization, multi-task Gaussian process, policy search, A/B testing, multi-fidelity optimization

1. Introduction

Online field experiments, commonly referred to as "A/B tests," are the gold-standard method for evaluating changes to Internet services and are an important tool for driving improvements (Kohavi et al., 2007; Manzi, 2012; Bakshy et al., 2014). Such experiments typically test only a few possible discrete variants, such as the presence or absence of a new feature, or a test of a new ranking model. Bayesian optimization (Jones et al., 1998) is an efficient approach to exploring and optimizing large, continuous parameter spaces in noisy environments, including field experiments (Letham et al., 2019). These methods open the possibility of solving reinforcement learning-type problems with Bayesian optimization (Lizotte et al., 2007; Wilson et al., 2014; Calandra et al., 2016; Metzen, 2016; Marco et al., 2017) in the context of real-world machine learning systems.

Bayesian optimization works by fitting a response surface model to a set of evaluated design points (e.g., a parameter configuration that determines the behavior of software, such as a policy) and iteratively deploying new points based on an explore/exploit algorithm. A

©2019 Benjamin Letham and Eytan Bakshy.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-225.html.

practical challenge in tuning online machine learning systems with Bayesian optimization is that each online field experiment can require millions of samples due to low signal-to-noise ratios in target outcomes (Kohavi et al., 2014). Given a limited available population, it can then be challenging to run enough experiments and evaluate enough design points to yield an accurate response surface model and to effectively optimize many parameters simultaneously.

Offline simulators can provide a higher-throughput mechanism for evaluating the effects of changes to production systems. For Internet services, an offline simulator can provide estimates of humans' interactions with a machine learning system by replaying sessions to the system under an alternative configuration policy. Behavioral models are then used to simulate events and compute aggregate outcomes (e.g., average cumulative rewards) to provide a counterfactual estimate of the alternative policy (Bottou et al., 2013; Dudík et al., 2014).

There are many sources of bias in simulators that can limit their utility. Even in cases where accurate predictive models are available, as the policy under evaluation pushes system behavior further from the status quo, it can be increasingly difficult to model individuals' responses to off-policy states that may be very far from previously observed states. Additionally, unobserved confounding can make it difficult to obtain high-quality counterfactual estimates. For example, Bottou et al. (2013) described how estimation of many interventions of interest in a production advertising system at Microsoft required careful consideration of the causal structure of the underlying system. Such analyses are often difficult to perform adequately in arbitrary machine learning systems that may not have randomized assignment policies.

It generally will not be possible to accurately simulate the interactions between humans and a machine learning system, especially for outcomes with more complex dynamics than click-through-rates. Take for example the case of ranking stories shown to people in the Facebook News Feed. The number of posts an individual is exposed to, and which posts an individual interacts with within the span of a single session, will be influenced by the composition (e.g., videos, images, text) and ordering of posts. Any interactions with specific content may in turn affect patterns of activity in present and future sessions (Eckles et al., 2016). Techniques from reinforcement learning and causal inference can be used to improve simulator accuracy in specific applications, including some multi-armed bandit settings (Tang et al., 2013; Hofmann et al., 2013; Li et al., 2015a,b), but generally these dynamics will not be captured. As a result, offline experiments are often considered as a precursor to evaluating a machine learning system with online experiments (Shani and Gunawardana, 2011). Here we show how they can be used in tandem with online experiments to increase the accuracy and throughput of online Bayesian optimization.

Bayesian optimization with the aid of a simulator is a special case of multi-fidelity optimization, in which there is a target function to be optimized along with a collection of cheaper approximations of varying fidelity. By modeling the relationship between the approximations and the target function, the approximations can be used to more efficiently explore the design space and to accelerate the optimization. Swersky et al. (2013) developed multi-task Bayesian optimization (MTBO), in which a multi-task Gaussian process (MTGP) is used to model a response surface jointly across tasks, which could be a discrete set

of fidelity levels. They showed that MTBO is effective for multi-fidelity hyperparameter optimization with downsampled data sets providing low-fidelity estimates.

In this paper, we provide a rigorous, empirical evaluation of how MTBO can be used effectively for real-world policy search and configuration problems. We first describe the empirical context of this work (Section 2), which is an online ranking system in a non-stationary, noisy environment (i.e., live Facebook traffic). We describe a simple simulator of that environment which uses the existing event prediction models to provide counterfactual estimates. In Section 3 we describe the MTGP model that enables borrowing of strength from offline measurements to online measurements, and review the assumptions behind the ICM kernel that we use. Section 4 provides the first set of empirical results on the suitability of the MTGP for response surface modeling in an online ranking system. Importantly, we show that with the MTGP we can directly use a simple, biased simulator together with a small number of online experiments to accurately predict online outcomes. In Section 5 we evaluate MTGPs and MTBO on a corpus of real-world experiments conducted using our proposed methodology. These results show that the addition of the simulator leads to better user experiences during the optimization and an improved final state. Finally, in Section 6 we review theoretical results on MTGP generalization, and then construct empirical learning curves for generalization in this setting. We provide an empirical analysis of the factors behind MTGP generalization and show how theory provides practical guidance on best practices for policy search with combined online and offline experimentation.

2. Empirical Context and the Simulator

In this paper we describe the tuning of a particular recommendation system, the Facebook News Feed. Generally, recommendation systems consist of a number of machine learning models that predict different events, like commenting or clicking. When an item is sent to the system for scoring, the score is based on these event predictions as well as features of the item, such as the strength of connection between viewer and content producer. The actual score is computed by applying a collection of rules and weights, which encode business logic and provide a way of combining all of the event predictions into an overall score. This collection of rules is called the *value model*, and can be seen as parameterizing the ranking policy used in a recommender system. Individual terms in the value model may target statistical surrogates (Prentice, 1989; Athey et al., 2016) that are expected to improve long-term outcomes of interest that are not directly modeled by event-based models that measure more immediate rewards. The value model forms a convenient way of balancing multi-objective decision problems faced in industrial settings, and similar approaches can be found at Yahoo!, LinkedIn, and other organizations (Agarwal et al., 2011, 2015).

A typical value model contains many parameters that affect its performance, such as weights for combining event prediction probabilities and rules that apply additional weights to the scores under certain conditions. These rules and weights constitute the policy for the system. The policy directly impacts long-term individual-level outcomes and must be tuned to achieve the desired system performance. The outcomes of interest can only be measured with online field experiments, and so the policy search must be done online. For example, in the case of News Feed ranking, a team may wish to tune value model parameters to maximize a measure of meaningful interactions (Mosseri, 2018), subject to constraints

on other engagement and ecosystem-related outcomes. This is the type of policy search problem that we wish to solve using Bayesian optimization.

We use the term *experiment* to refer to the process of tuning a particular set of parameters. An experiment represents a policy search problem and comprises potentially many A/B tests and simulator runs. The set of parameters being tuned and their ranges form the design space for the experiment, and we refer to the parameter values of any particular configuration as a *policy*. When policies are evaluated, either online or offline, we evaluate several in parallel and refer to the policies being simultaneously evaluated as a *batch*. For each evaluation, we record measurements of several *outcomes*, which are summary statistics of the various measurements of system performance (including individuals' behaviors). These measured outcomes of either an online or offline test are referred to as *observations* for those policies.

Because of the limitations in online experimentation throughput, policy search would be greatly accelerated if the outcomes of interest could be measured offline, via simulation. In fact, we already have predictive models for the event probabilities of many outcomes of interest, which are incorporated in the value model. A naïve approach to offline counterfactual estimation is to replay user sessions to the recommender system under a modified value model, apply the event models to the items selected by the system (e.g., probability of comment), and aggregate the model predictions across many replayed sessions to estimate top-level outcomes (e.g., total comments).

This simulation approach is attractive because it requires little additional infrastructure by making use of existing event prediction models, and it can evaluate arbitrary changes to the policy. However, as described in the Introduction, it suffers from significant limitations that prevent it from producing accurate estimates. Examples of the bias between offline estimates with this technique and online results are shown in Fig. 1. This figure shows three outcomes from experiments that varied value model parameters. In each experiment, a batch of 20 policies were evaluated both in the simulator and with online tests. Fig. 1 compares the observations of each policy, online and with the simulator.

Observations in Fig. 1 are shown relative to the status quo, meaning 0 on each axis corresponds to the status quo value. Generally the simulator observations were correlated with the observations of online experiments, but showed increasingly large discrepancies as the outputs deviated further from the status quo. Some of this is off-policy bias to be expected with this simulation approach since it uses only on-policy trained predictive models. There are also relevant dynamics of the system that are not being simulated that provide another source of bias. Since some of these outcomes will be used as constraints in the policy search problem, the monotonic relationship between simulator and online results is not sufficient to do the optimization with the simulator alone. Moreover, the bias between simulator and online results depends on the experiment (which parameters are being tuned), so any adjustment must be learned separately for each experiment. In general, we observe that simulators tend to systematically overestimate the size of effects, likely because of the difficulties with modeling elasticities in time and attention (Simon, 1971, 1996). We expect to see similar behaviors in other settings, such as markets, that may be subject to difficult-to-model elasticities.

The methodology that we develop here allows for these offline estimates to be directly combined with online observations to improve the throughput of online policy search, despite

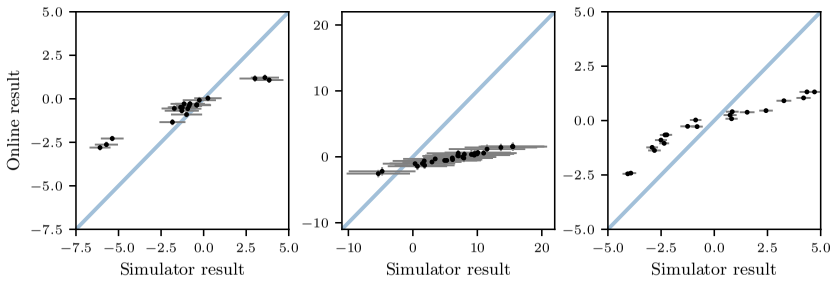


Figure 1: Simulator observations compared to results of online tests for three outcomes (panels). Each marker shows a policy that was evaluated in both the simulator and with an online test. The simulator measurements were correlated with the actual online results, but were too biased to be used in the place of online experiments.

the bias. We now describe the MTGP model for online-offline optimization. While the empirical setting of this work is value model tuning, many other parameters in information ranking and retrieval systems induce changes in system behavior that can be estimated offline in a similar way. The methodology can be applied to many such other policy search problems.

3. Response Surface Modeling with the Gaussian Process

Bayesian optimization relies on modeling the outcome of an online test as a function of policy parameters. A typical approach is to use a Gaussian process (GP) prior as the response surface model. The GP prior, $f \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$, is defined by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a covariance function $k(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[f(\mathbf{x}), f(\mathbf{x}')]$. The covariance function specifies the covariance between any two points in the design space and is typically chosen so that the covariance decays with distance, such as in the ARD RBF kernel:

$$k(\mathbf{x}, \mathbf{x}') = \tau^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{m}\left(\frac{x_j - x_j'}{\ell_j}\right)^2\right).$$

The kernel induces smoothness in $f(\mathbf{x})$ since points that are nearby in space are given high covariance. The degree of smoothness depends on the kernel variance and lengthscales, τ^2 and ℓ_j, which are inferred from data, typically by maximizing the marginal likelihood or with slice sampling (Murray and Adams, 2010).

Under a GP, any finite collection of function evaluations $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ are jointly normally distributed. For notational convenience, let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ denote the evaluated

points, $m(X) = [m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)]^\top$ denote the vector of means, and

$$K(X, X) = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

denote the covariance matrix across X. Then, the GP prior has

$$f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n) \sim \mathcal{N}(m(X), K(X, X)).$$

We typically do not directly observe the "true" outcome $f(\mathbf{x}_i)$, rather we observe a noisy measurement $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \eta_i^2)$. For online field experiments, we measure an aggregate outcome over a large test population and so are able to directly measure both the mean outcome y_i and its sample variance η_i^2. Let $\mathbf{y} = [y_1, \ldots, y_n]$ and $\boldsymbol{\eta}^2 = [\eta_1^2, \ldots, \eta_n^2]$. It follows that

$$\mathbf{y} \sim \mathcal{N}(m(X), K(X, X) + \mathrm{diag}(\boldsymbol{\eta}^2)).$$

Given observations $\{X, \mathbf{y}\}$, Gaussian closure under conditioning leads to an analytic posterior for the function value at any $\mathbf{x}_*$

$$f(\mathbf{x}_*)|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(\mu(\mathbf{x}_*, X, \mathbf{y}), \sigma^2(\mathbf{x}_*, X)), \tag{1}$$

where

$$\mu(\mathbf{x}_*, X, \mathbf{y}) = m(\mathbf{x}_*) + K(\mathbf{x}_*, X)[K(X, X) + \mathrm{diag}(\boldsymbol{\eta}^2)]^{-1}(\mathbf{y} - m(X)), \tag{2a}$$

$$\sigma^2(\mathbf{x}_*, X) = k(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, X)[K(X, X) + \mathrm{diag}(\boldsymbol{\eta}^2)]^{-1}K(X, \mathbf{x}_*), \tag{2b}$$

with $K(\mathbf{x}_*, X)$ the $1 \times n$ vector of covariances between $\mathbf{x}_*$ and each point in X.

An interesting property of GPs that can be seen from (2b) is that the posterior variance does not depend on $\mathbf{y}$, a fact that will be useful for characterizing generalization. GP regression was largely developed in the geostatistics community, where (2a) and (2b) are referred to as the kriging equations (Cressie, 1990). Rasmussen and Williams (2006) provide a thorough machine learning perspective on GP regression.

The GP has several properties that make it especially useful as a response surface model for Bayesian optimization. First, it provides uncertainty estimates for the function value at unobserved points. This is critical for applying an explore/exploit algorithm since exploration is driven by the uncertainty estimates. Second, both mean and variance predictions are available in closed form, in (2a) and (2b). This allows for fast gradient optimization of functions of the posterior when trying to identify the optimal point to test next. Finally, the smoothness assumption implied by the kernel tends in practice to be a good fit for the response surfaces of systems parameter tuning experiments. We have used the GP for parameter tuning via A/B tests in a variety of settings and consistently found good predictive performance (Letham et al., 2019). Fig. 2 shows examples of using a GP to predict online outcomes for two online policy search problems: a value model tuning experiment, and optimizing parameters of a video playback controller policy. For each of these, a batch of policies were tested online, and a GP was fit to the observations. For these experiments, leave-one-out cross-validation shows that the GP was able to make accurate out-of-sample predictions of the results of online tests. This model performance is typical of many successful Bayesian optimization experiments that we have run.

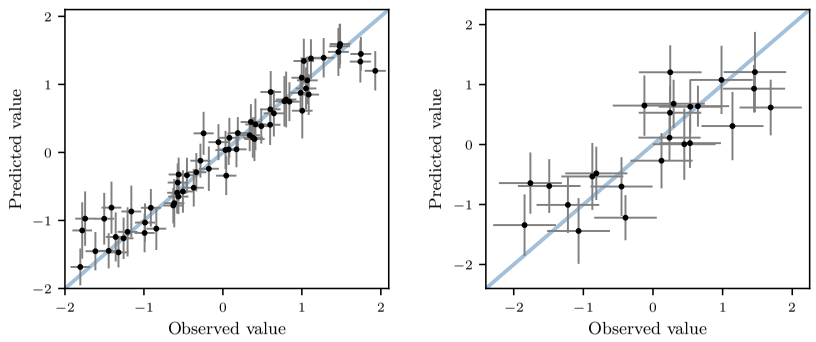


Figure 2: Leave-one-out cross validation predicting the outcome of online tests with a GP. (Left) A GP fit to 64 observations in a 9-dimensional value model tuning experiment. (Right) A GP fit to 24 observations in a 3-dimensional video playback controller policy tuning experiment. GPs provide good predictions for online field experiment outcomes, making them a suitable model for Bayesian optimization.

3.1. The Multi-Task Gaussian Process

Bayesian optimization with a GP response surface model is effective for policy search in online systems, however the number of observations (online tests) required for good GP predictions with 10–24 parameters can be prohibitive for systems with limited capacity for online experiments. The MTGP provides a natural way to include simulator observations into the model and reduce the number of online observations needed.

The MTGP extends the GP from a single function f to a collection of functions $f_1, \ldots, f_D$, each of these the response surface for a task. For our offline-online policy search, we initially have two tasks: an online task consisting of the online observations, and an offline task with the simulator observations. In Section 5 we will also model different simulator batches with separate tasks as well, and so will describe the model for an arbitrary number of tasks D.

The MTGP covariance function models the covariance across tasks in addition to the covariance across the design space: $k((d, \mathbf{x}), (d', \mathbf{x}')) = \mathrm{Cov}[f_d(\mathbf{x}), f_{d'}(\mathbf{x}')]$. We will make two assumptions to construct a multi-task covariance function. The first assumption is that all tasks share the same spatial kernel, which we will denote $\kappa(\mathbf{x}, \mathbf{x}')$. This assumption is especially useful because it allows us to rely on the simulator to estimate the spatial kernel hyperparameters, which can have many more observations than the online task. A similar approach is taken in reinforcement learning in robotics, where information sharing across tasks is facilitated by sharing a kernel (Chai et al., 2008). The second assumption is separability between the task covariance and the spatial covariance, which assumption yields

$$\mathrm{Cov}[f_d(\mathbf{x}), f_{d'}(\mathbf{x}')] = B_{d,d'}\kappa(\mathbf{x}, \mathbf{x}'). \tag{3}$$

Here B is a positive semidefinite task covariance matrix with element $B_{d,d'}$ the covariance between tasks d and d', and $\kappa(\mathbf{x}, \mathbf{x}')$ is the parameter covariance function. This is called the intrinsic coregionalization model (ICM) (Álvarez and Lawrence, 2011). The covariance function (3) can be substituted into (2a) and (2b) to immediately extend GP regression to the multi-task setting. The MTGP maintains all of the desirable properties of the GP for Bayesian optimization, while also modeling the relationship between the online and offline tasks.

The ICM kernel implicitly models each task function, f_d, as being a linear combination of independent latent functions. In the case of two tasks, it models each task as a sum of two latent functions

$$f_1(\mathbf{x}) = a_{11}u_1(\mathbf{x}) + a_{12}u_2(\mathbf{x}), \tag{4a}$$
$$f_2(\mathbf{x}) = a_{21}u_1(\mathbf{x}) + a_{22}u_2(\mathbf{x}), \tag{4b}$$

where $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$ have GP priors with the same covariance function $\kappa(\mathbf{x}, \mathbf{x}')$ (Álvarez et al., 2012). The weights are related to the cross-task covariance of (3) as $B_{12} = a_{11}a_{21} + a_{12}a_{22}$. High cross-task covariance thus corresponds to the two task functions concentrating their weight on the same latent function. In the extreme case of $a_{12} = a_{21} = 0$, the ICM kernel corresponds exactly to modeling the two tasks with independent GPs that have the same covariance function.

This formulation gives insight into what type of simulator biases the MTGP with ICM kernel will be capable of capturing. Rearranging (4a) and (4b), we have that $f_2(\mathbf{x}) = f_1(\mathbf{x}) + u'(\mathbf{x})$ where $u'(\mathbf{x}) = (a_{21} - a_{11})u_1(\mathbf{x}) + (a_{22} - a_{12})u_2(\mathbf{x})$. Here $u'(\mathbf{x})$ is the bias from one task to the other, and we see that $u'(\mathbf{x})$ is a GP with covariance function $\kappa(\mathbf{x}, \mathbf{x}')$. Thus, the ICM kernel implicitly models the simulator bias with a GP, and can capture any simulator bias that has the same degree of smoothness as the response surface itself. The biases shown in Fig. 1 are smooth over the output and suggest that an ICM kernel should be successful at modeling and thus adjusting for the bias seen with this simulator.

MTGP inference requires fitting the hyperparameters of the spatial covariance function, as in regular GP regression, along with the cross-task covariance matrix B. This is done by reparameterizing the cross-task covariance matrix with its Cholesky decomposition $B = LL^\top$, to ensure positive semidefiniteness. The matrix L is then chosen to maximize marginal likelihood. We inferred all kernel hyperparameters (B and the spatial kernel hyperparameters τ^2 and ℓ_j) from the data by maximizing marginal likelihood, using the Scipy interface to L-BFGS (Byrd et al., 1995; Zhu et al., 1997).

Estimating B requires fitting $D(D+1)/2$ parameters. In a Bayesian optimization setting with relatively few function evaluations, for even moderate D there can be more parameters than can reliably be fit. Bonilla et al. (2007) proposed using an incomplete Cholesky decomposition

$$B = \tilde{L}\tilde{L}^\top,$$

where $\tilde{L}$ is a $D \times P$ matrix, $P < D$. This reduces the number of parameters from $\mathcal{O}(D^2)$ to $\mathcal{O}(DP)$ and induces the constraint $\mathrm{rank}(B) \leq P$. They show that in practice lower rank parameterizations can have lower generalization error than higher rank parameterizations. For two tasks (online and offline) this is not necessary, but later in this paper we will model different batches with a task and will make use of a low-rank kernel. The rank

of the task covariance matrix corresponds to the number of latent functions in (4a) and (4b). Those equations correspond to a rank-2 task covariance matrix while a rank-1 task covariance matrix would imply each task function is a scaled version of the same, single latent function.

There are many alternatives to the ICM kernel with more complex structures—Álvarez et al. (2012) provide a review of multi-task kernels. In our application, the ICM assumption of shared spatial covariance is important for enabling inference of the kernel lengthscales. There are not enough online observations to infer a separate ARD kernel for the online task, and so we rely on the shared kernel and the leverage we can gain from simulator observations for hyperparameter inference. In Section 4 we show empirically that the transfer of information via the shared spatial kernel is appropriate in this setting and performs better than restricting the fitting to the online observations only. In Section 6 we use empirical learning curves to show that the shared spatial kernel significantly improves generalization.

4. Multi-Task Response Surface Models with the Simulator

We now study empirically how the MTGP performs for predicting the results of online value model tuning tests, with the benefit of the naïve simulator described in Section 2. Our study is based on 11 experiments that were run to improve the performance of the particular recommendation system, according to various online metrics. Each experiment optimized different sets of policy parameters, with the dimensionality ranging from 10 to 20. Each optimization began with a batch of 20 online observations and 100 simulator observations. The policies in these batches were generated from separate scrambled Sobol sequences (Owen, 1998) using a different seed. For each experiment, nine outcomes were measured which were treated independently for a total of 99 outcomes to be modeled. The data for each outcome were standardized by subtracting the mean of the observed values and dividing by their standard deviation. A 2-task MTGP was fit separately to each of the 99 outcomes, as described in Section 3.1.

Across this large body of experiments and models, we have found that including simulator experiments with the MTGP provides significant gains in model performance, for nearly every outcome and experiment. These gains are often substantial enough that the optimization would not be possible without the use of the simulator. Fig. 3 shows the benefit of the MTGP for one of the outcomes by comparing a single-task GP fit only to the 20 online observations with the MTGP fit to the same 20 online observations plus 100 simulator observations. We evaluated prediction performance using leave-one-out cross-validation, in which each online observation was removed from the data and the model fit to the remaining 19 observations (plus 100 simulator observations for the MTGP) was used to predict the online outcome of the held-out policy. Because the policies deployed online were generated with a space-filling design, this procedure provides a useful evaluation of out-of-sample prediction ability. For these experiments, none of the online policies were included in the simulator batch, so the results for the MTGP are also representative of out-of-sample prediction performance.

The particular experiment shown in Fig. 3 was over a 10-dimensional parameter space. The cross-validation results for the single-task model show that the 20 online observations alone were not sufficient to be able to make useful predictions in this space. This particular

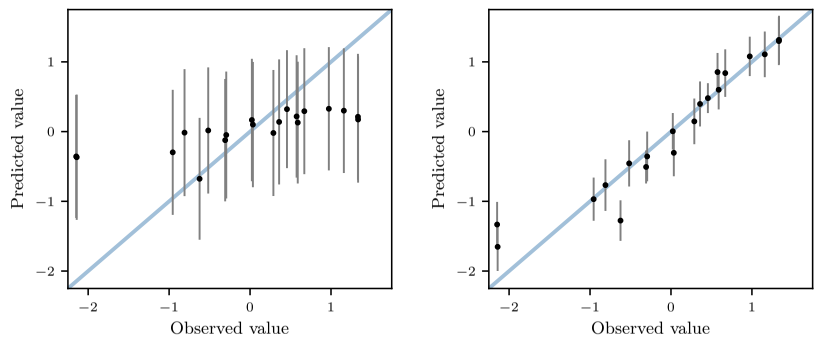


Figure 3: Leave-one-out cross validation results predicting the outcome of online tests of 20 policies in a 10-dimensional parameter space. (Left) Results using a GP fit only to the online observations. (Right) Results using an MTGP fit to both the online observations and 100 simulator observations. Due to the relatively high dimensionality, the online observations alone cannot be modeled well enough to make useful predictions, but incorporating the simulator observations allows for accurate out-of-sample predictions.

outcome was used as a constraint in the optimization, and the predictive uncertainty is too high to determine if a policy is likely to be feasible or not. Our practical experience with model fits like that of the single-task model is that the policy search would not be successful, and we typically would not continue the experiment. After incorporating simulator experiments with the MTGP, the model performs remarkably well and is able to make accurate predictions.

The improvement in model predictions from incorporating simulator observations seen in Fig. 3 have been seen across many outcomes and experiments. Fig. 4 shows the mean squared error (MSE) of the 20 online leave-one-out predictions for each of the 99 outcomes described above. Because the data for each outcome were transformed to be mean 0 and standard deviation 1, the expected squared error of a constant prediction at the mean is 1. Fig. 4 shows that for most of the outcomes, the single-task model with online experiments only did not do much better than predicting the mean. The poor predictions of the single-task GP are not surprising considering there are only 20 observations in 10–20 dimensions. The MTGP, however, produced useful predictions for most outcomes, and importantly, when the prediction errors were high they were still at least as good as the online-only model on all outcomes.

In situations where the simulator performs poorly, as discussed in Section 3.1 the MTGP contains the single-task GP as a special case and so should in general do no worse than not using the simulator at all, so long as overfitting is avoided. This was the case for all of the outcomes in Fig. 4. For those in which the MTGP had high MSE, the lack of substantial

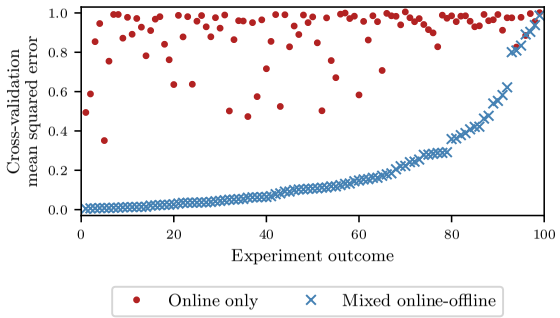


Figure 4: For each of 99 experiment outcomes, 20 policies were tested online and 100 policies were tested on the simulator, and two models were separately fit to the observations: a single-task GP to online observations only, and an MTGP to both online and offline observations. Markers show the cross-validation MSE for each of these 99 experiment outcomes, ordered by MTGP MSE. Across this large suite of experiments and outcomes, incorporating simulator observations significantly improved model predictions.

improvement over the single-task GP was driven by low correlation between the online and offline tasks. In some instances the low correlation came from a high level of observation noise, while for others it was poor fidelity of the simulator—these factors are analyzed in Online Appendix 1.

5. Bayesian Optimization by Combining Online and Offline Experiments

The ultimate purpose of fitting the MTGP is to produce an accurate model with which to do Bayesian optimization and find an optimal policy. Bayesian optimization with a mix of online and simulator observations proceeds in a similar way as regular single-task Bayesian optimization. We begin with a small number of randomly selected policies to test online, and a much larger set to test offline. The MTGP is fit to the outcomes of these first two batches, as in Section 4, and is then used to maximize an acquisition function that identifies valuable policies to test in the next batch. We now describe the acquisition function used and how a mix of online and offline batches are used throughout the optimization, and then provide examples of actual value model tuning optimizations done using the method.

5.1. The Acquisition Function

To select policies to evaluate with future tests, we use expected improvement integrated over observation noise (Letham et al., 2019). The optimization problem that we consider has objective $\max_{\mathbf{x}} f(\mathbf{x})$ and constraints $c_j(\mathbf{x}) \geq 0$, $j = 1, \ldots, J$. Here f and c_j represent online outcomes that are modeled with separate MTGPs. Consider first a single-task problem

with noisy observations of objective and constraint outcomes at a set of policy parameters X.

We let $\mathcal{D}$ denote the set of all observed data, and use the shorthand $f(X)$ to denote the vector of (unobserved, true) objective function evaluations at each policy in X. The unobserved, true constraint evaluations are likewise denoted $\mathbf{c}(X) = [c_1(X), \ldots, c_J(X)]$. Without observation noise, the improvement at $\mathbf{x}$ over the current best feasible policy can be expressed as[1]

$$I(\mathbf{x}, X) = \max \left(0, f(\mathbf{x}) - \max_{\mathbf{x}^* \in X : \mathbf{c}(\mathbf{x}^*) \geq \mathbf{0}} f(\mathbf{x}^*)\right) \mathbb{1}_{\mathbf{c}(\mathbf{x}) \geq \mathbf{0}}.$$

EI is the expectation over the posterior of $f(\mathbf{x})$ of this quantity, and the noisy EI acquisition function is defined by further integrating over the observation noise,

$$\alpha_{\text{NEI}}(\mathbf{x}|X, \mathcal{D}) = \mathbb{E}_{f(X), \mathbf{c}(X)} \left[\mathbb{E}_{f(\mathbf{x})} \left[I(\mathbf{x}, X) | f(X), \mathbf{c}(X) \right] | \mathcal{D} \right].$$

The inner expectation has an analytic solution. The outer expectation is over the multivariate normal GP posterior and can be effectively handled with numerical (quasi-Monte Carlo) integration.

This acquisition function has several properties that make it especially useful for optimizing with online experiments—it robustly handles noise and constraints, is easily extended to batch optimization, and is easy to implement and maintain in production. There are other acquisition functions that have been used for multi-fidelity problems, which we describe in Section 7.

We adapt noisy EI to the multi-task setting by restricting the definition of the best-feasible policy to include only observations from the primary task (online observations). Given online observations at X_T and simulator observations at X_S, we now let $\mathcal{D}_T$ represent online observations and $\mathcal{D}_S$ simulator observations. We then use as the acquisition function

$$\alpha_{\text{NEI}}(\mathbf{x}|X_T, \mathcal{D}_T \cup \mathcal{D}_S). \tag{5}$$

Simulator observations are thus used to improve the model posteriors $f(X_T)$ and $\mathbf{c}(X_T)$, but are not considered to have been an observation for the purposes of evaluating improvement.

5.2. The Bayesian Optimization Loop

Optimizing the acquisition function produces a batch of points that maximizes the expected improvement over the policies that have been observed online, in expectation over the model predictions for their online outcomes. We then have the choice to launch the batch either online or to the simulator. Here we used a fixed strategy where batches of online experiments and batches of simulator experiments are interleaved. The full Bayesian optimization loop used here is given in Algorithm 1.

In our experiments here, the initial quasi-random batches consisted of $n_T = 20$ online tests and $n_S = 100$ simulator tests. The initial batches do not necessarily have to cover the same space—the simulator can be used to explore policies on a wider design space than the

1. For ease of exposition, the definition given here assumes the existence of a current feasible point. The case where all observations are infeasible is handled in Letham et al. (2019).

Algorithm 1: Online-Offline Bayesian Optimization

Data: Online batch size n_T, simulator batch size n_S, optimization batch size $n_o \geq n_T$, number of optimization iterations K.

1 Construct a quasi-random batch of n_T policies to be tested online, X_T, and n_S policies to be tested on the simulator, X_S.
2 Run the online and simulator tests. Let $\mathcal{D}$ represent the complete collection of observations.
3 **for** $k = 1, \ldots, K$ **do**
4 Fit an MTGP to the observations $\mathcal{D}$ for each outcome of interest.
5 Optimize the acquisition function (5) to obtain a batch of n_o optimized policies to be tested.
6 Test the n_o optimized policies on the simulator. Let $\mathcal{D}_{S,k}$ represent the observations and update $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_{S,k}$.
7 Re-fit the MTGPs for each outcome to incorporate the new simulator observations.
8 Select the best n_T of the optimized policies to be tested online. Let $\mathcal{D}_{T,k}$ represent the observations, and update $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_{T,k}$.
9 **return** *The best policy observed online.*

online tests, allowing for more exploration while not veering too far off-policy in the online tests. In Line 4, for the first iteration ($k = 1$), we fit a 2-task MTGP as done in Section 4. For later iterations where there were multiple batches of simulator observations, we introduced additional tasks to handle non-stationarity across simulator batches, which are handled with a low-rank ICM kernel. This aspect of the modeling is described in Appendix A. In Line 5, we used the acquisition function to generate a batch of $n_o = 30$ optimized policies. In Line 8, we used Thompson sampling (Thompson, 1933) to select 8–10 policies for online tests, depending on available capacity.

Interleaving online and simulator batches as is done in Algorithm 1 improves the quality of the online tests, and thus user experience in the test groups, by reducing the predictive uncertainty of candidates that have high EI. EI measures a balance between exploration and exploitation, so a policy selected in the first acquisition function optimization (Line 5) may have high EI because it has high predictive uncertainty. Running that policy on the simulator (Line 6) will likely reduce that uncertainty and can filter out from the online batch policies that are no longer likely to be good after their simulator observation. The online batch can thus be more exploitative (lower regret), while still maintaining thorough exploration through the simulator. As an alternative to interleaving the batches, one could use the acquisition function to weigh the relative efficiency and resource requirements of online and simulator experiments—we discuss this further in Section 7.

In Online Appendix 2, we apply Algorithm 1 to a synthetic online-offline problem that is modeled after our application here. There we compare to regular, single-task BO, and also evaluate how much improvement comes from the interleaved offline batches (Lines 6 and 7). Our focus here, however, is on real online optimizations, which we turn to now.

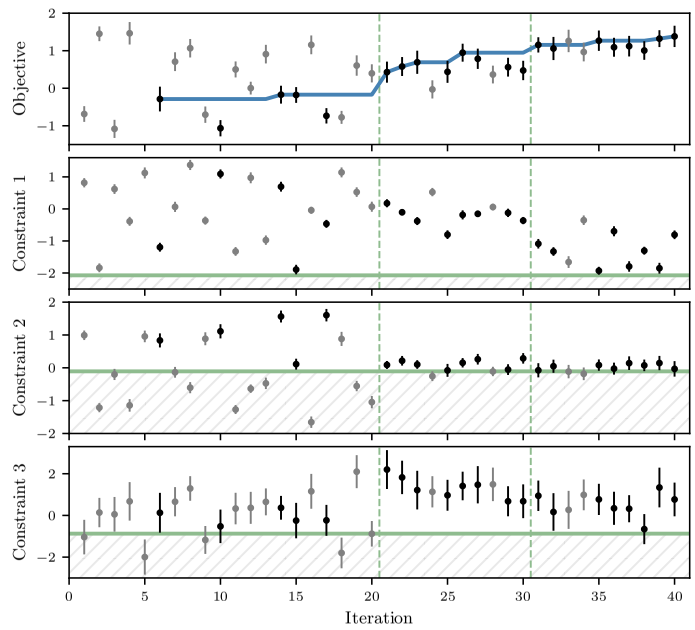


Figure 5: Results of a multi-task optimization to maximize the objective subject to three constraints. Each point shows an online observation (mean and two standard errors). Gray points were infeasible in expectation, and the infeasible region for each constraint is shaded. The solid line for the objective tracks the expected best-feasible objective. Dashed vertical lines delineate batches of policies evaluated in parallel. A simulator batch was run prior to each online batch. Bayesian optimization with the MTGP was able to find significantly better policies with a limited number of online tests.

5.3. Examples of Value Model Tuning with Online-Offline Experiments

We now provide five examples of Bayesian optimization using both online and simulator observations according to the loop in Algorithm 1. Each of these experiments sought to maximize a particular objective outcome of the recommendation system's online performance, subject to 3–5 constraints on other outcomes. The constraint outcomes here are other metrics that have trade-offs with the objective, so improving one can often be to the detriment of the other. The experiments targeted different sets of value model parameters, with the design space dimensionality ranging from 10 to 20.

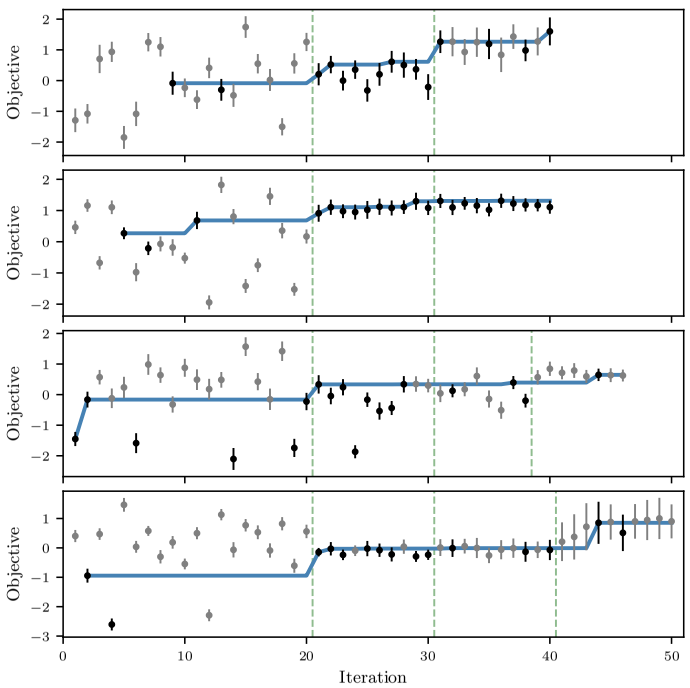


Figure 6: Results of four separate policy searches over different sets of value model parameters. Each maximized the objective shown, subject to 3–5 constraints (not shown). As in Fig. 5, grayed-out points were infeasible in expectation, the solid line tracks the expected best-feasible policy, and vertical lines delineate batches of policies evaluated online in parallel. These examples show the success of multi-task Bayesian optimization in 10–20 dimensions.

Fig. 5 shows the results of a round of Bayesian optimization in a 13-dimensional space with three constraints. The quasi-random initialization of 20 online observations and 100 simulator observations was followed by two rounds of interleaved simulator and online batches, as described in Section 5.2, for a total of 160 policies evaluated in the simulator and 40 policies evaluated online. Outcomes and constraints were each modeled with an MTGP. The figure shows the results of each online observation. Most of the policies in the initialization were infeasible in expectation, and none of the feasible policies improved over the status quo value of 0. The status quo policy had previously been manually tuned and so quasi-random search was not sufficient to improve the objective. Optimizing with

the MTGP, on the other hand, led to significantly better feasible policies in the optimized batches. Not only did the optimization improve the best-feasible objective, it also increased the proportion of online observations that satisfied the constraints, which corresponds to improved user experience in the test groups. The second optimized batch continued to improve over the first as it was able to push closer to the constraint bounds. The fact that we are able to jointly optimize 13 parameters with only 40 total online tests is remarkable, and is entirely due to the MTGP and the simulator.

The optimization gains in Fig. 5 are typical of a collection of successful optimizations we have done using the MTGP with the simulator. Fig. 6 shows the results of four other optimizations, with 3–5 constraints and 10–20 parameters. As in Fig. 5, for these problems the constraints were restrictive enough that most of the initial policies were infeasible. Subsequent optimized batches found feasible policies that substantially improved the objective value. For all of these optimizations the total number of online observations required was greatly reduced compared to single-task Bayesian optimization: 40–50 total online observations for 10–20 parameters.

6. MTGP Generalization and the Value of a Simulator Observation

We have seen that the MTGP is able to make accurate predictions of online outcomes using a small number of online observations and a large number of biased offline observations. The model is essentially able to use offline observations as a substitute for online observations. We now study the extent to which simulator observations can replace online tests, and in particular the number of simulator observations required and the conditions in which the model can effectively borrow strength from the simulator observations. Both of these are ultimately a question of MTGP generalization behavior, which we now describe first theoretically and then empirically.

6.1. Learning Curves for the MTGP

We now define the quantities used to evaluate MTGP generalization and discuss their theoretical properties. We begin with the single-task setting of (1) where we have observed data $\{X, y\}$ and make a prediction at point $\mathbf{x}_*$. We suppose that there is some probability density from which training points X and test points $\mathbf{x}_*$ are drawn, i.i.d. Rasmussen and Williams (2006) show that if the true f is a draw from a GP, then squared prediction error equals the predictive variance:

$$\mathbb{E}_f\left[(\mu(\mathbf{x}_*, X, \mathbf{y}) - f(\mathbf{x}_*))^2\right] = \sigma^2(\mathbf{x}_*, X).$$

The single-task *learning curve* for n training points is defined as the expected squared prediction error, which is

$$\epsilon(n) = \mathbb{E}_{X,\mathbf{x}_*}\left[\sigma^2(\mathbf{x}_*, X)\right], \tag{6}$$

for X of size n.

These definitions extend naturally to the multi-task case, where each point in X has an associated task and there is a target task associated with $\mathbf{x}_*$. For the purposes of the theoretical results, for the remainder of this section we will focus on the two-task case described by Chai (2009). There is a "primary" task (online observations) for which we

make predictions and a "secondary" task (simulator observations) which can be used to improve the predictions on the primary task. For the theoretical results here, we assume that the two tasks have the same output variance, which is then pushed into the parameter kernel $\kappa(\mathbf{x}, \mathbf{x}')$ so that

$$B = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

We do not make this assumption when actually using the model in the other sections of this paper, in which the task output variances are inferred from the data—it is only for the purposes of deriving the bound in Proposition 1.

We suppose that we have training observations at X_T on the primary task and X_S on the secondary task, with n_T and n_S the sizes of X_T and X_S respectively. The posterior predictive variance $\sigma_T^2(\mathbf{x}_*, \rho, X_T, X_S)$ can then be computed in closed form by substituting the multi-task kernel into (2b).

The multi-task learning curve for n_T primary task observations and n_S secondary task observations is defined in the same way as the single-task learning curve:

$$\epsilon_T(\rho, n_T, n_S) = \mathbb{E}_{X_T, X_S, \mathbf{x}_*}\left[\sigma_T^2(\mathbf{x}_*, \rho, X_T, X_S)\right]. \tag{7}$$

Chai (2009) proved an elegant bound on the multi-task predictive variance that provides insight into the relative value of online and offline observations for predictive accuracy.

Proposition 1 *(Chai, 2009, Proposition 5(a))*

$$\sigma_T^2(\mathbf{x}_*, \rho, X_T, X_S) \geq \rho^2 \sigma_T^2(\mathbf{x}_*, \rho = 1, X_T, X_S) + (1 - \rho^2)\sigma_T^2(\mathbf{x}_*, \rho = 0, X_T, X_S).$$

When $\rho = 1$, observations from each task are equivalent, and the model reduces to a single-task GP with the combined training set $X = [X_T, X_S]$. Similarly, with $\rho = 0$ the secondary task observations are ignored, and the model reduces to a single-task GP but with $X = X_T$. Taking the expectation over i.i.d. X_T, X_S, and $\mathbf{x}_*$ yields the following result.

Corollary 2 *Let $\epsilon_\kappa(n)$ be the learning curve for the single-task GP with covariance function $\kappa(\mathbf{x}, \mathbf{x}')$. Then,*

$$\epsilon_T(\rho, n_T, n_S) \geq \rho^2 \epsilon_\kappa(n_T + n_S) + (1 - \rho^2)\epsilon_\kappa(n_T).$$

The result in Corollary 2 is a bound, however Chai (2009) showed empirically that it is relatively tight and closely approximates the multi-task learning curve. This result shows that the ability of the multi-task model to generalize depends on the ability to generalize on each task separately, $\epsilon_\kappa(n)$, and the inter-task correlation ρ. When tasks are highly correlated, the $\rho^2\epsilon_\kappa(n_T + n_S)$ term dominates and secondary task observations propel the learning forward on the single-task learning curve. Alternatively, when task correlation is low the $(1 - \rho^2)\epsilon_\kappa(n_T)$ term dominates and secondary task observations have little effect.

Besides providing insight into the factors behind generalization, a useful property of the bound is that the quantities on the right-hand-side can be empirically estimated without primary task observations. The single-task learning curve $\epsilon_\kappa(n)$ can be estimated from secondary task observations only. Given an estimate of ρ^2 (perhaps from past usage of the simulator), we can then use the bound to understand the expected generalization to the primary task before even making primary task observations.

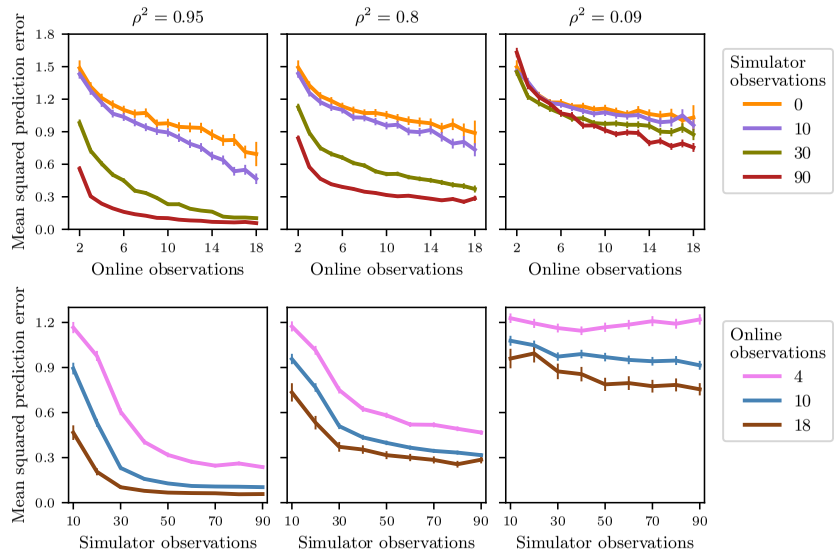


Figure 7: Empirical learning curves from value model tuning experiments for the MTGP. Each column shows an outcome with different levels of squared inter-task correlation ρ^2 as indicated. Curves show the average over 500 training set draws, with two standard errors as error bars. Learning curves are shown as a function of the number of online observations n_T with number of offline observations n_S fixed (top row), and as a function of n_S with n_T fixed (bottom row). Simulator observations greatly accelerate learning when the inter-task correlation ρ^2 is high.

6.2. Empirical Learning Curves

We use cross-validation to estimate empirical learning curves $\epsilon_T(\rho, n_T, n_S)$ for the value model policy models in Section 4. To estimate the MTGP learning curve at n_T online observations and n_S simulator observations, we randomly sampled (without replacement) n_T of the 20 online observations and n_S of the 100 simulator observations and fit a model to only those data. We then made predictions at the held-out $20 - n_T$ online points and evaluated mean squared prediction error and mean predictive variance. This was repeated with 500 samples of the data to approximate the expectation over X_T, X_S, and $\mathbf{x}_*$ in (7). Empirical learning curves are shown in Fig. 7 for three outcomes from an experiment over a 10-d parameter space. These outcomes were chosen to illustrate learning curves with weak, medium, and strong inter-task correlations; the square of the inter-task correlation, ρ^2, is indicated for each outcome.

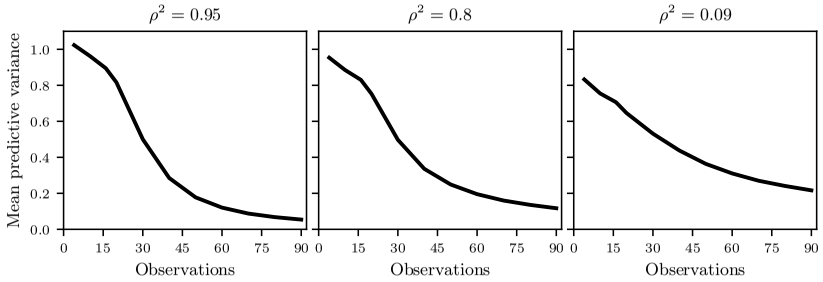


Figure 8: Single-task empirical learning curves estimated from simulator observations only. Panels correspond to the same three outcomes as Fig. 7, in a 10-d parameter space. There is an initial slow phase of learning in which there are not enough observations to infer the kernel, after which learning accelerates.

The empirical learning curves show how the simulator observations significantly improve generalization throughout the learning curve. As expected from theory, the extent to which simulator observations improve learning depends on the squared inter-task correlation ρ^2. For ρ^2 high, 2 online observations combined with 90 simulator observations produce a better model than 18 online observations. For ρ^2 small, at least 5 online observations are needed before the simulator even begins to provide benefit. The rate at which prediction error decreases with simulator observations is much lower than with high correlation. Comparing the $\rho^2 = 0.8$ and $\rho^2 = 0.09$ outcomes in Fig. 7 with 0 simulator observations shows that the online response surfaces are of a similar difficulty to predict with the GP; the dramatic difference in the response to simulator observations comes from the difference in inter-task correlation.

6.3. Comparing Theory to Empirical Learning Curves

The theoretical result in Corollary 2 provides an elegant relationship between the single-task learning curve, $\epsilon_\kappa(n)$, and the multi-task learning curve. Under the ICM assumption of a shared covariance function, the single-task learning curve can be estimated using only simulator observations. To estimate the learning curve, we use the same cross-validation procedure used for the MTGP learning curves in Fig. 7: n simulator observations were chosen at random, a single-task GP was fit to those data, and predictive variance was measured on the $100 - n$ held-out points. Fig. 8 shows the single-task learning curves for the three outcomes from Fig. 7.

The single-task learning curves are characterized by an initial slow period of learning, followed by a rapid and saturating reduction in predictive variance. The initial slow period can be attributed to the difficulty in inferring the kernel in 10 dimensions until a minimum number of observations have been obtained. Together with the estimated ρ^2, these learning curves are used to evaluate the bound $\rho^2\epsilon_\kappa(n_T + n_S) + (1 - \rho^2)\epsilon_\kappa(n_T)$. Importantly, this is

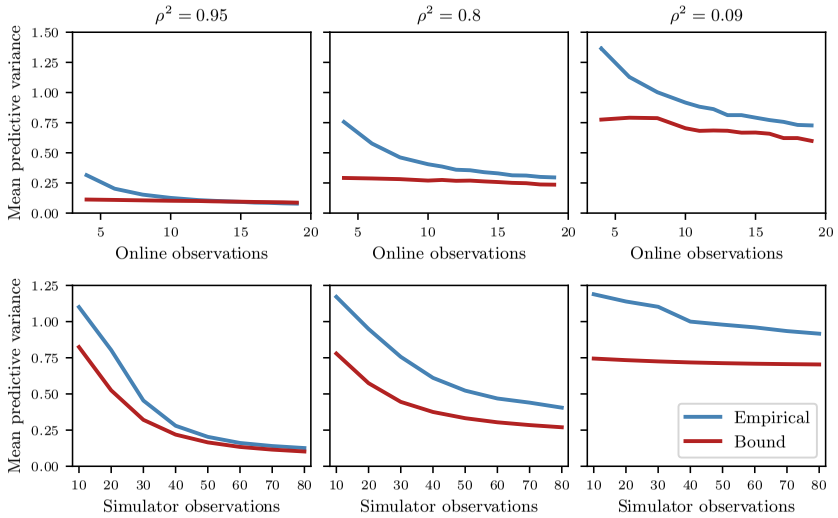


Figure 9: Empirical multi-task learning curves compared to the theoretical bound applied to the empirical single-task learning curves. Top row shows learning curves as a function of the number of online observations n_T with the number of simulator observations n_S fixed at 80. Bottom row shows learning curves as a function of the number of simulator observations n_S, with $n_T = 10$ online observations. Theoretical results provide a useful indication of real-world performance.

computed using only simulator observations and ρ^2. This bound is compared to the actual empirical multi-task learning curve $\epsilon_T(\rho, n_T, n_S)$ in Fig. 9.

For these real experiments, although the MTGP fits the data well it is not the actual data generating process as assumed by the bound, and so we do not expect the bound to necessarily be satisfied by empirical learning curves. For instance, the real data do not have the same output variance for each task as assumed by Proposition 1. Although the output variances will be similar because we standardize the observations from each task prior to fitting, we do infer a separate output variance for each task. Despite these differences between the theoretical framing and practical setting, we do find here that the bound provides useful guidance for understanding the role of online and simulator observations in learning.

The empirical learning curves are generally consistent with theory, and in particular the role of ρ^2 on learning is well predicted by theory. With fewer than 10 online observations, the empirical learning curve shows a steeper decline than the bound. The source of this discrepancy can be seen by comparing the single-task learning curve of Fig. 8 to the multi-task learning curves of Fig. 7. In the single-task learning curve, the variance reduces

slowly with n up until around 20 observations. With simulator observations, the multi-task model avoids this slow initial phase of learning and exhibits a steep learning curve from the start. This suggests that improved kernel inference plays an important role in the MTGP performance, a factor that we analyze in the next section. As n_T grows, the value and slope of the empirical learning curves are consistent with the theoretical bound.

These results show that theory can provide practical guidance on the number of online and simulator observations required to reach a desired generalization error, in addition to quantifying the value of incremental observations. In the single-task learning curves the $\rho^2 = 0.8$ outcome had higher mean predictive variance at $n = 19$ than the $\rho^2 = 0.09$ outcome, however the bound correctly predicts that with simulator observations the $\rho^2 = 0.09$ outcome has higher mean predictive variance at $n_T = 19$. Computing the bound for higher values of n_T could guide a decision when trading off predictive variance with the resource requirements of online observations.

6.4. Kernel Inference and MTGP Generalization

The theoretical results and empirical learning curves show that simulator observations increase the effective amount of data, with the relative value of simulator and online observations depending on ρ^2. More data improves the model predictions in two distinct ways. First, the GP is a linear smoother, and having more data nearby the prediction point reduces predictive uncertainty. Second, the additional data improves inference for the kernel hyperparameters: the kernel output variance, lengthscales, and the task covariance matrix. All of the learning curves computed so far included kernel inference as part of the learning. That is, the kernel hyperparameters were inferred from the subset of data, and so the learning curves incorporate both of these sources of learning.

The single-task learning curves in Fig. 8 showed an initial slow phase of learning due to kernel inference. Because of the ICM assumption of a shared spatial kernel, the MTGP is able to use offline observations to infer the kernel and thus avoid this initial slow phase.

How much of the MTGP performance is due solely to improved kernel inference and the ICM assumption? To answer this question, we computed empirical learning curves for three models, on the outcomes used in the previous section. The first model was a single-task GP fit only to the online data, with the kernel inferred as part of model fitting as usual. The third was an MTGP with the same online data plus 100 offline observations, also with the kernel inferred as part of model fitting. The third model was a single-task GP that was given only online data for computing the posterior, however the kernel was fixed at the best offline kernel (that is, the kernel inferred when fitting a single-task GP to the 100 offline observations). These empirical learning curves are shown in Fig. 10.

The two single-task GPs use the same data to compute the posterior, but we see that substituting the offline kernel for the kernel inferred from online data significantly accelerates learning. In fact, much of the gap in performance between the single-task GP and the MTGP can be attributed solely to improved ability to infer the kernel—not because of conditioning on more data in the GP posterior. For the outcomes in Fig. 10 with medium and high values of ρ^2 there was still substantial value to the MTGP beyond just the offline kernel, but that diminished as the number of online observations grew. For the outcome with low ρ^2, practically all of the MTGP's performance improvement came from the kernel inference.

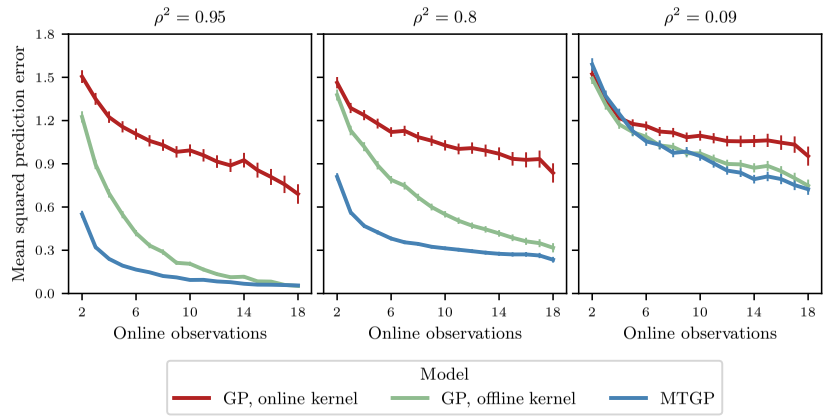


Figure 10: Empirical learning curves for three models: A single-task GP fit to online observations, including kernel inference; a single-task GP that uses only online observations to compute the posterior but has the kernel fixed as that inferred from the simulator observations; and an MTGP with 100 offline observations. For higher numbers of online observations, much of the performance improvement with the MTGP comes from the shared kernel with the offline task.

7. Discussion

We have provided empirical results and described practical techniques for using Bayesian optimization to tune an online machine learning system with the aid of a naïve simulator. Across a large set of experiment results, the MTGP with simulator observations provided remarkable gains in predictive performance over a single-task GP fit to online observations only. Section 4 showed that the single-task model failed badly to predict for most of the experiment outcomes we studied here—the optimizations in Section 5 would not have been possible without the simulator. These results showed that with the modeling power of the MTGP, even a simple, biased simulator provides a powerful way of enabling policy search via Bayesian optimization in settings where the limited pool for online experiments would otherwise be prohibitive.

Multi-fidelity Bayesian optimization has been studied with both continuous (Kandasamy et al., 2017) and discrete (Kandasamy et al., 2016) fidelities—here we had two fidelities, the online outputs and the simulator outputs. MTGPs have been used to model discrete fidelities in many physics and engineering applications (Wackernagel, 1994; Forrester et al., 2007; Fernández-Godino et al., 2016). Huang et al. (2006) first proposed Bayesian optimization with multiple fidelities using a heuristic extension to EI, albeit with a different kernel structure than the ICM kernel used here. In machine learning, the most common application of multi-fidelity Bayesian optimization is hyperparameter optimization with downsampled

data sets (Swersky et al., 2013), an approach that has been developed with several kernels and acquisition functions (Klein et al., 2015, 2017; Poloczek et al., 2017). MTGPs have also been used for transfer learning in hyperparameter optimization, alongside a number of other models for learning relationships across tasks (Yogatama and Mann, 2014; Poloczek et al., 2016; Shilton et al., 2017). Other substantial machine learning applications of MTGPs and Bayesian optimization have been contextual bandit optimization (Krause and Ong, 2011) and reinforcement learning (Chai et al., 2008; Metzen, 2016; Marco et al., 2017).

The empirical learning curves characterized the generalization of the multi-task model in practice, and showed the relative values of online and simulator observations. The theory behind MTGP identifies ρ^2 as the key factor determining the relative value of simulator observations. We found that our empirical learning curves were consistent with the theoretical bound; that ρ^2 was an important factor for multi-task learning, and that the theory could be used to guide decisions in sizing experiments. Previous empirical validation of the MTGP theory used simulations from the generating process. To our knowledge, these are the first results to compare empirics to theory on real-world multi-task modeling problems.

The ICM kernel was shown to be particularly effective at modeling the relationship between the online and offline observations, and enabling the simulator to improve predictions for online outcomes. The assumption of a shared spatial kernel allows kernel inference to rely primarily on the numerous simulator observations, and the implicit assumption of smooth bias was empirically appropriate for these experiments. Our results showed that a substantial part of the gain from using the MTGP can be attributed directly to the ICM shared spatial kernel. These results are consistent with work on Bayesian optimization for learning robot locomotion policies, where simulators were used for learning a custom kernel that was then transferred to real-world contexts for safe and sample-efficient learning (Antonova et al., 2017; Rai et al., 2019).

Finally, we found that the MTGP allowed for effective optimization of a live recommendation system. We used noisy EI as the acquisition function because it naturally handles the properties of online experiments—noisy observations, noisy constraints, and large batches. EI does not, however, directly provide a way of selecting the modality of experimentation (online or simulator), hence we used a fixed strategy of interleaving simulator and online batches. Alternative acquisition functions could potentially identify whether an online batch or simulator batch should be done. Several heuristic extensions to EI allow for jointly selecting the point to be tested and the fidelity (Huang et al., 2006; Lam et al., 2015). The knowledge gradient provides a principled way to measure the expected improvement in the online task that a simulator batch would provide (Poloczek et al., 2017). This could be compared to the improvement provided by an online batch, and the relative gains and efficiencies could be used to evaluate if it is most beneficial to run the simulator batch or the online batch. However, knowledge gradient has not been developed to handle constraints, a necessary requirement for our experiments. Predictive entropy search (Hernández-Lobato et al., 2014, 2015; McLeod et al., 2018) allows for measuring the effect a simulator batch has in reducing the entropy of the location of the online optimizer. However, it is not immediately clear how to construct a decision rule for evaluating the trade-off between information gain and experiment time. Building such a decision rule into the acquisition function remains an important area of future work.

Acknowledgments

We thank Si Chen, Yang Liu, Ashwin Murthy, and Pengfei Wang who worked on the simulator and systems integration. We also thank Brian Karrer for providing valuable modeling insights and David Arbour for work on model validation. Finally, we thank Roberto Calandra, Mohammad Ghavamzadeh, Ron Kohavi, Alex Deng, and our anonymous reviewers for helpful feedback on this work.

Appendix A. Handling Simulator Non-stationarity during Optimization

As the optimization progresses, new batches of policies are constructed and launched either online or offline as we search for the optimal policy. In later iterations when we have accumulated several batches of simulator observations, optimization with the aid of the simulator has an additional challenge that must be handled in the modeling: non-stationarity across simulator batches.

For practical reasons, the sessions replayed on the simulator often cannot be a fully representative sample of sessions, rather they are sampled from live sessions during the simulator run. This is similar to the "live benchmarking" paradigm in Bakshy and Frachtenberg (2015), where requests are streamed to multiple servers in parallel. Such live setups are often more efficient from a resource utilization perspective, since relevant state information can remain "hot" in distributed caches, and need not be saved to disk. This leads to a biased sample that is more likely to include sessions that occur at the same time as the simulator experiment. If the effects of the experiment are heterogeneous across people logged in at different times of the day or on different days, then we can expect the simulator outputs to have a bias that depends on when the simulation was run. This bias could be reduced through the use of propensity score weighting or regression adjustment (Sekhon, 2011; Bottou et al., 2013; Hartman et al., 2015), but such approaches involve substantially more infrastructure. Moreover, the user population is not the only source of non-stationarity: there is also non-stationarity in the characteristics of items being scored by the machine learning system.

In online field experiments, these sources of non-stationarity are avoided by running the experiment for several days, as a matter of routine practice (Kohavi et al., 2007). Offline experiments on the simulator are run for much less time to improve throughput, and so the bias across simulator batches must be corrected in the modeling. Fig. 11 shows an example of the effect of non-stationarity on simulator observations. This figure compares simulator observations from two one-hour periods of replayed sessions, separated by several days. Each experiment tested a batch of the same 100 points, and there are clear discrepancies in the simulator outcomes in observations at different times.

Bias across simulator batches can be handled naturally in the MTGP the same way the online-offline bias is handled, by treating each offline batch as a separate task. The need to treat each simulator batch as a separate task could make it challenging to estimate the $\mathcal{O}(D^2)$ cross-task covariances, however we can use a low-rank task covariance matrix as described in Section 3.1. We use a low-rank ICM kernel to learn adjustments for simulator batches, and then apply a full rank (rank-2) kernel to the online-offline tasks as done in Section 4.

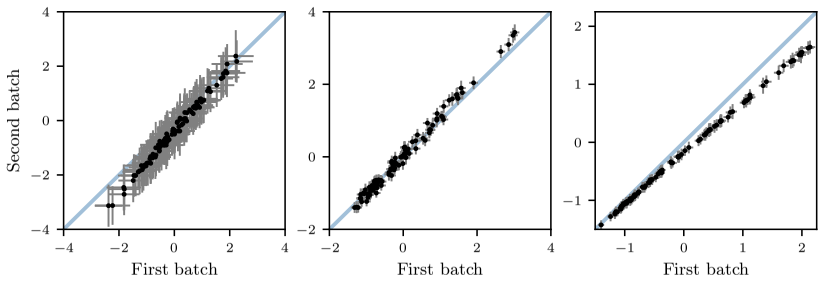


Figure 11: Examples of the non-stationarity bias across batches run on the simulator, for three experiments. Each point was run on the simulator twice, in batches separated by several days. Sessions non-stationarity leads to differences in simulator outcomes when points are tested at different times.

Empirically, we find that the bias across simulator batches due to non-stationarity nearly always admits a low-rank representation. For example, the biases shown in Fig. 11 are all linear shifts of the response surface. Linear corrections are often used as multi-fidelity models in engineering (Fernández-Godino et al., 2016), and correspond to a rank-1 ICM kernel, as discussed in 3.1. Bonilla et al. (2007) show that low-rank approximations such as that can be beneficial for generalization, and we use cross-validation to ensure that the model is performing well. In the Bayesian optimization loop in Algorithm 1, a subset of the initial points were repeated in every simulator batch so that there were repeated policies across batches to aid in learning this shift.

References

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. Click shaping to optimize multiple objectives. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 132–140, 2011.

Deepak Agarwal, Shaunak Chatterjee, Yang Yang, and Liang Zhang. Constrained optimization for homepage relevance. In *Proceedings of the 24th International Conference on World Wide Web*, WWW, pages 375–384, 2015.

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

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

Rika Antonova, Akshara Rai, and Christopher G. Atkeson. Deep kernels for optimizing locomotion controllers. In *Proceedings of the 1st Conference on Robot Learning*, CoRL, 2017.

Susan Athey, Raj Chetty, Guido Imbens, and Hyunseung Kang. Estimating treatment effects using multiple surrogates: the role of the surrogate score and the surrogate index. *arXiv preprint arXiv:1603.09326*, 2016.

Eytan Bakshy and Eitan Frachtenberg. Design and analysis of benchmarking experiments for distributed internet services. In *Proceedings of the 24th International Conference on World Wide Web*, WWW, pages 108–118, 2015.

Eytan Bakshy, Dean Eckles, and Michael S. Bernstein. Designing and deploying online field experiments. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW, pages 283–292, 2014.

Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, NIPS, pages 153–160, 2007.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: the example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1–2):5–23, 2016.

Kian Ming A. Chai. Generalization errors and learning curves for regression with multi-task Gaussian processes. In *Advances in Neural Information Processing Systems 22*, NIPS, pages 279–287, 2009.

Kian Ming A. Chai, Christopher K. I. Williams, Stefan Klanke, and Sethu Vijayakumar. Multi-task Gaussian process learning of robot inverse dynamics. In *Advances in Neural Information Processing Systems 21*, NIPS, pages 265–272, 2008.

Noel Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252, 1990.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Dean Eckles, René F Kizilcec, and Eytan Bakshy. Estimating peer effects in networks with peer encouragement designs. *Proceedings of the National Academy of Sciences*, 113(27): 7316–7322, 2016.

M Giselle Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T Haftka. Review of multi-fidelity models. *arXiv preprint arXiv:1609.07196*, 2016.

Alexander I.J. Forrester, András Sóbester, and Andy J. Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269, 2007.

Erin Hartman, Richard Grieve, Roland Ramsahai, and Jasjeet S Sekhon. From sample average treatment effect to population average treatment effect on the treated: Combining experimental with observational studies to estimate population treatment effects. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178(3):757–778, 2015.

José Miguel Hernández-Lobato, Matthew W. Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems 27*, NIPS, pages 918–926, 2014.

José Miguel Hernández-Lobato, Michael A. Gelbart, Matthew W. Hoffman, Ryan P. Adams, and Zoubin Ghahramani. Predictive entropy search for Bayesian optimization with unknown constraints. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML, pages 1699–1707, 2015.

Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. Reusing historical interaction data for faster online learning to rank for IR. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, WSDM, pages 183–192, 2013.

D. Huang, T. T. Allen, W. I. Notz, and R. A. Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5): 369–382, 2006.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

Kirthevasan Kandasamy, Gautam Dasarathy, Junier Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimization with multi-fidelity evaluations. In *Advances in Neural Information Processing Systems 29*, NIPS, pages 992–1000, 2016.

Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Bayesian optimisation with continuous approximations. In *Proceedings of the 34th International Conference on Machine Learning*, ICML, pages 1799–1808, 2017.

Aaron Klein, Simon Bartels, Stefan Falkner, Philipp Hennig, and Frank Hutter. Towards efficient Bayesian optimization for big data. In *Proceedings of the 2015 NIPS Workshop on Bayesian Optimization*, BayesOpt, 2015.

Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2017.

Ron Kohavi, Randal M Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web: Listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 959–967, 2007.

Ron Kohavi, Alex Deng, Roger Longbotham, and Ya Xu. Seven rules of thumb for web site experimenters. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 1857–1866, 2014.

Andreas Krause and Cheng S. Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems 24*, NIPS, pages 2447–2455, 2011.

Rémi Lam, Douglas L. Allaire, and Karen E. Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *Proceedings of the 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA SciTech Forum, 2015.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. Counterfactual estimation and optimization of click metrics in search engines: a case study. In *Proceedings of the 24th International Conference on World Wide Web*, WWW, pages 929–934, 2015a.

Lihong Li, Jin Young Kim, and Imed Zitouni. Toward predicting the outcome of an A/B experiment for search relevance. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, WSDM, pages 37–46, 2015b.

Daniel Lizotte, Tao Wang, Michael Bowling, and Dale Schuurmans. Automatic gait optimization with Gaussian process regression. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI, pages 944–949, 2007.

Jim Manzi. *Uncontrolled: The Surprising Payoff of Trial-and-Error for Business, Politics, and Society*. Basic Books, New York, New York, 2012.

Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P. Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, ICRA, pages 1557–1563, 2017.

Mark McLeod, Michael A. Osborne, and Stephen J. Roberts. Practical Bayesian optimization for variable cost objectives. *arXiv preprint arXiv:1703.04335*, 2018.

Jan Hendrik Metzen. Minimum regret search for single- and multi-task optimization. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML, pages 192–200, 2016.

Adam Mosseri. Bringing people closer together, 2018. URL `https://newsroom.fb.com/news/2018/01/news-feed-fyi-bringing-people-closer-together/`.

Iain Murray and Ryan P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems 24*, NIPS, pages 1732–1740, 2010.

Art B Owen. Scrambling Sobol' and Niederreiter–Xing points. *Journal of Complexity*, 14 (4):466–489, 1998.

Matthias Poloczek, Jialei Wang, and Peter I. Frazier. Warm starting Bayesian optimization. In *Winter Simulation Conference*, WSC, 2016. arXiv preprint arXiv:1608.03585.

Matthias Poloczek, Jialei Wang, and Peter I. Frazier. Multi-information source optimization. In *Advances in Neural Information Processing Systems 30*, NIPS, pages 4288–4298, 2017.

Ross L Prentice. Surrogate endpoints in clinical trials: definition and operational criteria. *Statistics in Medicine*, 8(4):431–440, 1989.

Akshara Rai, Rika Antonova, Franziska Meier, and Christopher G. Atkeson. Using simulation to improve sample-efficiency of Bayesian optimization for bipedal robots. *Journal of Machine Learning Research*, 20(49):1–24, 2019.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, Massachusetts, 2006.

Jasjeet Sekhon. Multivariate and propensity score matching software with automated balance optimization: the matching package for R. *Journal of Statistical Software*, 42(7): 1–52, 2011.

Guy Shani and Asela Gunawardana. Evaluating recommender systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 257–297. Springer, Boston, MA, 2011.

Alistair Shilton, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Regret bounds for transfer learning in Bayesian optimisation. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2017.

Herbert A. Simon. Designing organizations for an information-rich world. In M Greenberger, editor, *Computers, Communications, and the Public Interest*, chapter 2, pages 28–72. Johns Hopkins Press, 1971.

Herbert A. Simon. *The Sciences of the Artificial*. The MIT Press, Cambridge, Massachusetts, 1996.

Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems 26*, NIPS, pages 2004–2012, 2013.

Liang Tang, Rómer Rosales, Ajit P. Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM, pages 1587–1594, 2013.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294, 1933.

Hans Wackernagel. Cokriging versus kriging in regionalized multivariate data anslysis. *Geoderma*, 62:83–92, 1994.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. Using trajectory data to improve Bayesian optimization for reinforcement learning. *Journal of Machine Learning Research*, 15(1): 253–282, 2014.

Dani Yogatama and Gideon Mann. Efficient transfer learning method for automatic hyperparameter tuning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2014.

Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.

Journal of Machine Learning Research 20 (2019) 1-33 Submitted 5/18; Published 9/19

Characterizing the Sample Complexity of Pure Private Learners*

Amos Beimel BEIMEL@CS.BGU.AC.IL
Ben-Gurion University

Kobbi Nissim KOBBI.NISSIM@GEORGETOWN.EDU
Georgetown University

Uri Stemmer U@URI.CO.IL
Ben-Gurion University

Editor: Moritz Hardt

Abstract

Kasiviswanathan et al. (FOCS 2008) defined *private learning* as a combination of PAC learning and differential privacy. Informally, a private learner is applied to a collection of labeled individual information and outputs a hypothesis while preserving the privacy of each individual. Kasiviswanathan et al. left open the question of characterizing the sample complexity of private learners.

We give a combinatorial characterization of the sample size sufficient and necessary to learn a class of concepts under pure differential privacy. This characterization is analogous to the well known characterization of the sample complexity of non-private learning in terms of the VC dimension of the concept class. We introduce the notion of *probabilistic representation* of a concept class, and our new complexity measure RepDim corresponds to the size of the smallest probabilistic representation of the concept class.

We show that any private learning algorithm for a concept class $\mathcal{C}$ with sample complexity m implies $\mathrm{RepDim}(\mathcal{C}) = O(m)$, and that there exists a private learning algorithm with sample complexity $m = O(\mathrm{RepDim}(\mathcal{C}))$. We further demonstrate that a similar characterization holds for the database size needed for computing a large class of optimization problems under pure differential privacy, and also for the well studied problem of private data release.

Keywords: Differential privacy, PAC learning, Sample complexity

1. Introduction

Motivated by the observation that learning generalizes many of the analyses applied to large collections of data, Kasiviswanathan et al. (2011) defined *private learning* as a combination of probably approximately correct (PAC) learning (Valiant, 1984) and differential privacy (Dwork et al., 2006). A PAC learner is given a collection of labeled examples (sampled according to an unknown probability distribution and labeled according to an unknown concept) and generalizes the labeled examples into a hypothesis h that should predict with high accuracy the labeling of fresh examples.

The privacy requirement is that the choice of h preserves differential privacy of sample points. Intuitively this means that this choice should not be significantly affected by any particular sample. Differential privacy is increasingly accepted as a standard for rigorous privacy, and recent research has shown that differentially private variants exists to many analyses. We refer the reader to the excellent surveys by Dwork and Roth (2014) and Vadhan (2016).

A natural problem is to characterize the *sample complexity* – the minimum number of examples necessary in order to identify a good hypothesis – as a function of the target class $\mathcal{C}$. This important measure determines the amount of data that must be collected before starting the analysis. Without privacy, it is well-known that the sample complexity of PAC learning is proportional to the Vapnik–Chervonenkis (VC) dimension of the class $\mathcal{C}$ (Vapnik and Chervonenkis, 1971; Blumer et al., 1989; Ehrenfeucht et al., 1989).

In analogy to this characterization of the sample complexity of non-private PAC learners via the VC-dimension, we give a combinatorial characterization of the sample size sufficient and necessary for PAC learners satisfying pure differential privacy. Towards obtaining this characterization, we introduce the notion of *probabilistic representation* of a concept class. We note that our characterization, as the VC-dimension characterization, ignores the computation required by the learner. Some of our algorithms are, however, computationally efficient.

1.1. Related Work

In the initial work on private learning, Kasiviswanathan et al. (2011) proved that a private learner exists for every *finite* concept class. Their construction of is based on the exponential mechanism of McSherry and Talwar (2007), and exhibits a sample complexity logarithmic in $|\mathcal{C}|$. The VC dimension of a concept class $\mathcal{C}$ is always at most $\log|\mathcal{C}|$, but is significantly lower for many interesting classes. Hence, the results of Kasiviswanathan et al. (2011) left open the possibility that the sample complexity of private learning may be significantly higher than that of non-private learning.

Consider the task of *properly* learning a concept class $\mathcal{C}$ where, after consulting its sample, the learner outputs a hypothesis that is by itself in $\mathcal{C}$. While non-privately this restriction has no effect on the sample complexity, Beimel et al. (2014) showed that it can have a big impact for pure differentially private learners. Specifically, Beimel et al. proved lower bounds on the sample complexity of *properly* learning the class of point functions under pure differential privacy, implying that the VC dimension of a class does not characterize the sample complexity of pure private proper learning. On the other hand, they observed that the sample complexity can be improved for *improper* private learners whenever there exists a smaller hypothesis class $\mathcal{H}$ that represents $\mathcal{C}$ in the sense that for every concept $c \in \mathcal{C}$ and for every distribution on the examples, there is a hypothesis $h \in \mathcal{H}$ that is close to c. Using the exponential mechanism to choose among the hypotheses in $\mathcal{H}$ instead of $\mathcal{C}$, the sample complexity is reduced to $\ln|\mathcal{H}|$ (this is why the *size* of the representation $\mathcal{H}$ is defined to be $\ln|\mathcal{H}|$). For some classes this can dramatically improve the sample complexity, e.g., for the class of point functions, the sample complexity is improved from $O(\ln|\mathcal{C}|)$ to

*. A preliminary version of this paper appeared in ITCS'13 under the title "Characterizing the Sample Complexity of Private Learners". Subsequent works showed that the sample complexity of private learning can be quite different under approximate differential privacy, and we chose to change the title in order to reflect the fact that our characterization only applies to learners satisfying pure differential privacy. See discussion in Section 1.4

©2019 Amos Beimel and Kobbi Nissim and Uri Stemmer.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-269.html.

$O(\ln \ln |\mathcal{C}|)$. Using other techniques, Beimel et al. showed that the sample complexity of learning point functions can be reduced even further to $O(1)$, hence showing the largest possible gap between proper and non proper private learning. Such a gap does not exists for non-private learning.

1.2. Our Results

Beimel et al. (2014) showed how to use a representation of a class to privately learn it. We make an additional step in improving the sample complexity by considering a *probabilistic* representation of a concept class $\mathcal{C}$. Instead of one collection $\mathcal{H}$ representing $\mathcal{C}$, we consider a list of collections $\mathcal{H}_1, \ldots, \mathcal{H}_r$ such that for every $c \in \mathcal{C}$ and every distribution on the examples, if we sample a collection $\mathcal{H}_i$ from the list, then with high probability there is a hypothesis $h \in \mathcal{H}_i$ that is close to c. To privately learn $\mathcal{C}$, the learning algorithm first samples $i \in \{1, \ldots, r\}$ and then uses the exponential mechanism to select a hypothesis from $\mathcal{H}_i$. This reduces the sample complexity to $O(\max_i \ln |\mathcal{H}_i|)$; the *size* of the probabilistic representation is hence defined to be $\max_i \ln |\mathcal{H}_i|$. We define the representation dimension (RepDim) of a class $\mathcal{C}$ as the size of its smallest such probabilistic representation.

We show that for point functions there exists a probabilistic representation of size $O(1)$. This results in a private learning algorithm with sample complexity $O(1)$, matching a different algorithm of Beimel et al. (2014). Our new algorithm offers some improvement in the sample complexity compared to the algorithm of Beimel et al. (2014) when considering the learning and privacy parameters. Furthermore, our algorithm can be made computationally efficient without making any computational hardness assumptions, while the efficient version of Beimel et al. (2014) assumes the existence of one-way functions. Finally, it is conceptually simpler.

One can ask if there are private learning algorithms with smaller sample complexity than the size of the smallest probabilistic representation. We show that under pure differential privacy the answer is no — the size of the smallest probabilistic representation is a lower bound on the sample complexity. Thus, the size of the smallest probabilistic representation of a class $\mathcal{C}$, which we call the *representation dimension* and denote by $\mathrm{RepDim}(\mathcal{C})$, characterizes (up to constants) the sample size necessary and sufficient for learning the class $\mathcal{C}$ under pure differential privacy.

The notion of probabilistic representation applies not only to private learning, but also to optimization problems. We consider a scenario where there is a domain X, a database S of m records, each taken from the domain X, a set of solutions $\mathcal{F}$, and a quality function $q : X^* \times \mathcal{F} \to [0, 1]$ that we wish to maximize. If the exponential mechanism is used for (approximately) solving the problem, then the size of the database should be $\Omega(\ln |\mathcal{F}|)$ in order to achieve a reasonable approximation. Using our notions of a representation of $\mathcal{F}$ and of a probabilistic representation of $\mathcal{F}$, one can reduce the size of the minimal database without paying too much in the quality of the solution. Interestingly, a similar notion to representation, called "solution list algorithms", was considered by Beimel et al. (2008) for constructing secure protocols for search problems while leaking only a few bits on the input. Curiously, their notion of leakage is very different from that of differential privacy.

We give two examples of such optimization problems. First, an example inspired by Beimel et al. (2008): each record in the database is a clause with exactly 3 literals and we

want to find an assignment satisfying at least 7/8 fraction of the clauses while protecting the privacy of the clauses. A construction of Beimel et al. (2008) yields a deterministic representation for this problem where the size of the database can be much smaller. Using a probabilistic representation, we can give a good assignment even for databases of constant size. This example is a simple instance of a scenario, where each individual has a preference on the solution and we want to choose a solution maximizing the number of individuals whose preferences are met, while protecting the privacy of the preference. Another example of optimization is sanitization, where given a database we want to publish a synthetic database, which gives a similar utility as the original database while protecting the privacy of the individual records of the database. Using our techniques, we study the minimal database size for which sanitization gives reasonable performance with respect to a given family of queries.

1.3. Subsequent Work: Private Learning and Communication Complexity

Following our work, Feldman and Xiao (2015) showed an equivalence between $\mathrm{RepDim}(\mathcal{C})$ and the randomized one-way communication complexity of the evaluation problem for concepts from $\mathcal{C}$. Using this equivalence, they separated the sample complexity of pure private learners from that of non-private ones, and showed that the sample complexity of pure private learners (proper or improper) can generally be much higher than what is required for non-private learning.

We next present a high level overview of the ideas of Feldman and Xiao (2015). Let $\mathcal{C}$ be a concept class over a domain X, and consider the following communication problem: Alice holds a function $c \in \mathcal{C}$ and Bob holds an input $x \in X$. Together they want to compute $c(x)$, where we can assume that the function c and the input x are sampled from some distribution μ (known to both Alice and Bob). The randomized one-way communication complexity of this problem is the length of the shortest message that Alice needs to send Bob to allow him to compute $c(x)$ correctly (w.h.p. over the choice of c and x and over the shared randomness in the protocol).

To see how $\mathrm{RepDim}(\mathcal{C})$ upper bounds the communication complexity, consider a list of collections $\mathcal{H}_1, \ldots, \mathcal{H}_r$ that probabilistically represents $\mathcal{C}$ where $\mathrm{RepDim}(\mathcal{C}) = \max_i \ln |\mathcal{H}_i|$. We can now use $\mathcal{H}_1, \ldots, \mathcal{H}_r$ to design a protocol as follows. Alice and Bob begin by using their shared randomness in order to sample a collection $\mathcal{H}_i$ from the list. By the definition of probabilistic representation, with high probability there is a hypothesis $h \in \mathcal{H}_i$ that is close to the function c that Alice holds, in the sense that $h(x') = c(x')$ w.h.p. over $x' \sim \mu|c$. Alice now identifies such a hypothesis h, and sends it to Bob, who computes $h(x)$. Observe that specifying h requires sending $\log |\mathcal{H}_i| \leq \mathrm{RepDim}(\mathcal{C})$ bits.

In the case of threshold functions over a domain X, computing $c(x)$ is exactly the well-known "greater than" communication problem (where Alice holds $a \in X$ and Bob holds $b \in X$, and together they want to learn if $a > b$), for which a lower bound of $\Omega(\log |X|)$ is known (Miltersen et al., 1998). Hence, as RepDim characterizes the sample complexity of pure private learners, the sample complexity of every pure private (proper or improper) learner for the class of threshold functions over a domain X is at least $\Omega(\log |X|)$. This is a strong separation from the non-private sample complexity, which is $O(1)$ as the VC dimension of this class is constant.

1.4. Subsequent Work: Sample Complexity of Approximate Private Learners

Beimel et al. (2013) showed that relaxing the privacy requirement from pure to approximate differential privacy can drastically reduce the sample complexity of private learners. In particular, they showed that threshold functions can be learned with approximate privacy using $2^{O(\log^* |X|)}$ examples, a dramatic improvement over the $\Omega(\log |X|)$ lower bound on the sample complexity of every pure private learner for this class (Feldman and Xiao, 2015). For point functions, they constructed an approximate private *proper* learner with constant sample complexity, again circumventing the lower bound of $\Omega(\log |X|)$ for pure private *proper* learners.

In light of these positive results on the sample complexity of approximate private learners, one might hope that the sample complexity of such learners is actually characterized by the VC dimension and is of the same order as that of non-private learning. However, this is not the case. First, Bun et al. (2015) showed that any approximate private *proper* learner for threshold functions over X must have sample complexity $\Omega(\log^* |X|)$. This result separated the sample complexity of approximate private proper learners from that of non-private ones, but left open the question of understanding the sample complexity of approximate private *improper* learners.

Recently, Alon et al. (2018) showed that a similar lower bound holds also for *improper* learners. This means that learning threshold functions over an infinite domain is impossible with approximate privacy. In fact, the negative result of Alon et al. (2018) holds for any concept class $\mathcal{C}$ with infinite *Littlestone Dimension* (Littlestone, 1987). Informally, the *Littlestone Dimension* of a concept class $\mathcal{C}$ over a domain X, denoted $\mathrm{Ldim}(\mathcal{C})$, is the maximal depth of a complete binary tree such that each root-to-leaf path in the tree can be "explained" by some concept $c \in \mathcal{C}$. In more details, consider a complete binary tree T in which each node is labeled by a domain element $x \in X$. Every concept $c \in \mathcal{C}$ *realizes* a root-to-leaf path in the tree T, where from a node that is labeled by an element x we proceed to the left child if $c(x) = 0$ and proceed to the right child if $c(x) = 1$. The Littlestone Dimension of $\mathcal{C}$ is the depth of the largest complete tree T such that every root-to-leaf path in T is realized by some concept $c \in \mathcal{C}$. Alon et al. (2018) showed that any approximate private learner (proper or improper) for a class $\mathcal{C}$ must have sample complexity $\Omega(\log^*(\mathrm{Ldim}(\mathcal{C})))$. In particular, the class of thresholds over a domain X has Littlestone Dimension $\log |X|$, and hence, every approximate private learner for it requires $\Omega(\log^* |X|)$ examples.

We remark that the connection between the Littlestone Dimension and the sample complexity of private learners was first identified by Feldman and Xiao (2015) in the context of *pure* private learners. They showed that every *pure* private learner (proper or improper) for a class C must have sample complexity $\Omega(\mathrm{Ldim}(\mathcal{C}))$. Tables 1 summarizes the currently known bounds on the sample complexity of private learners.

1.5. Other Related Work

Chaudhuri and Hsu (2011) studied learning algorithms that are only required to protect the privacy of the labels (and do not necessarily protect the privacy of the examples themselves). They proved upper and lower bounds on the sample complexity of such algorithms. In particular, they proved a lower bound on the sample complexity using the doubling dimension of the disagreement metric of the hypothesis class with respect to the unlabeled data distribution. Following their results, Beimel et al. (2013) showed that the sample complexity of such learners is actually fully characterized by the VC dimension, and is of the same order as that of non-private learners.

	Pure privacy	Approximate privacy
Generic bounds for properly learning a class $\mathcal{C}$	$O(\log \lvert\mathcal{C}\rvert)$ $\Omega(\mathrm{Ldim}(\mathcal{C}))$	$O(\log \lvert\mathcal{C}\rvert)$ $\Omega(\log^*(\mathrm{Ldim}(\mathcal{C})))$
Generic bounds for improperly learning a class $\mathcal{C}$	$\Theta(\mathrm{RepDim}(\mathcal{C}))$ $\Omega(\mathrm{Ldim}(\mathcal{C}))$	$O(\mathrm{RepDim}(\mathcal{C}))$ $\Omega(\log^*(\mathrm{Ldim}(\mathcal{C})))$
Properly Learning Points over domain X	$\Theta(\log \lvert X\rvert)$	$\Theta(1)$
Improperly Learning Points over domain X	$\Theta(1)$	$\Theta(1)$
Learning Thresholds over domain X (properly or improperly)	$\Theta(\log \lvert X\rvert)$	$2^{O(\log^* \lvert X\rvert)}$ $\Omega(\log^* \lvert X\rvert)$

Table 1: Bounds on the sample complexity of private learning.

Beimel et al. (2015) studied the labeled sample complexity of *semi-supervised* private learners, and showed that the sample complexity of such learners is also characterized by the VC dimension.

A line of research (started by Schapire (1990)) that is very relevant to our paper is boosting learning algorithms, that is, taking learning algorithms that have a big classification error and producing a learning algorithm with small error. Dwork et al. (2010) show how to privately boost accuracy, that is, given a *private* learning algorithms that have a big classification error, they produce a *private* learning algorithm with small error. In Lemma 18, we show how to boost the accuracy α for probabilistic representations. This gives an alternative private boosting, whose proof is simpler. However, as it uses the exponential mechanism, it is (generally) not computationally efficient.

1.6. Open Questions

Our understanding of the sample complexity of learning under approximate (ϵ, δ)-differential privacy is still very limited. On the one hand, there are currently no *generic* upper bounds on the sample complexity of approximate private learners that achieve a better sample complexity than the generic constructions for pure private learners. On the other hand, the only currently known lower bounds are the above mentioned bounds that scale with $\Omega(\log^* |X|)$. While $\log^* |X|$ is super constant, we would like to know whether there are cases in which the gap in the sample complexity (compared to the non-private sample complexity) is larger, say $\Omega(\log |X|)$. In addition, we consider the task of *characterizing* the sample complexity of approximate private learners to be an important question for future work.

2. Preliminaries

Notation. We use $O_\gamma(g(n))$ as a shorthand for $O(h(\gamma)\cdot g(n))$ for some non-negative function h. Given a set $\mathcal{B}$ of cardinality r, and a distribution $\mathcal{P}$ on $\{1,2,\ldots,r\}$, we use the notation $b \in_{\mathcal{P}} \mathcal{B}$ to denote a random element of $\mathcal{B}$ chosen according to $\mathcal{P}$.

2.1. Preliminaries from Privacy

A database is a vector $S = (z_1,\ldots,z_m)$ over a domain X, where each entry $z_i \in S$ represents information contributed by one individual. Databases S_1 and S_2 are called *neighboring* if they differ in exactly one entry. An algorithm preserves differential privacy if neighboring databases induce nearby outcome distributions. Formally,

Definition 1 (Dwork et al. (2006)) *A randomized algorithm A is (ϵ,δ)-differentially private if for all neighboring databases S_1, S_2, and for all sets $\mathcal{F}$ of outputs,*

$$\Pr[A(S_1) \in \mathcal{F}] \le \exp(\epsilon)\cdot \Pr[A(S_2) \in \mathcal{F}] + \delta. \tag{1}$$

The probability is taken over the random coins of A. When $\delta = 0$ we omit it and say that A preserves ϵ-differential privacy.

We use the term *pure* differential privacy when $\delta = 0$ and the term *approximate* differential privacy when $\delta > 0$, in which case δ is typically a negligible function of the database size. The focus of this work is on pure differential privacy.

An immediate consequence of the definition of pure differential privacy is that for *any* two databases $S_1, S_2 \in X^m$, and for all sets $\mathcal{F}$ of outputs,

$$\Pr[A(S_1) \in \mathcal{F}] \ge \exp(-\epsilon m)\cdot \Pr[A(S_2) \in \mathcal{F}].$$

2.2. Preliminaries from Learning Theory

Let $X_d = \{0,1\}^d$. A concept $c : X_d \to \{0,1\}$ is a function that labels *examples* taken from the domain X_d by either 0 or 1. A *concept class* $\mathcal{C}$ over X_d is a class of concepts mapping X_d to $\{0,1\}$.

PAC learning algorithms are given examples sampled according to an unknown probability distribution $\mathcal{D}$ over X_d, and labeled according to an unknown *target* concept $c \in \mathcal{C}$. The *generalization error* of a hypothesis $h : X_d \to \{0,1\}$ is defined as

$$\text{error}_{\mathcal{D}}(c,h) = \Pr_{x \in_{\mathcal{D}} X_d}[h(x) \ne c(x)].$$

For a labeled sample $S = (x_i, y_i)_{i=1}^m$, the *empirical error* of h is

$$\text{error}_S(h) = \frac{1}{m}|\{i : h(x_i) \ne y_i\}|.$$

Definition 2 *An* α-good *hypothesis for c and $\mathcal{D}$ is a hypothesis h such that* $\text{error}_{\mathcal{D}}(c,h) \le \alpha$.

Definition 3 (Valiant (1984)) *Algorithm A is an* (α,β)-PAC learner *for a concept class $\mathcal{C}$ over X_d using hypothesis class $\mathcal{H}$ and sample size m if for all concepts $c \in \mathcal{C}$, all distributions $\mathcal{D}$ on X_d, given an input of m samples $S = (z_1,\ldots,z_m)$, where $z_i = (x_i, c(x_i))$ and x_i are drawn i.i.d. from $\mathcal{D}$, algorithm A outputs a hypothesis $h \in \mathcal{H}$ satisfying*

$$\Pr[\text{error}_{\mathcal{D}}(c,h) \le \alpha] \ge 1-\beta.$$

The probability is taken over the random choice of the examples in S according to $\mathcal{D}$ and the coin tosses of the learner A.

Definition 4 *An algorithm satisfying Definition 3 with $\mathcal{H} \subseteq \mathcal{C}$ is called a* proper *PAC learner; otherwise it is called an* improper *PAC learner.*

2.3. Private Learning

As a private learner is a PAC learner, its outcome hypothesis should also be a good predictor of labels. Hence, the privacy requirement from a private learner is not that an application of the hypothesis h on a new sample (pertaining to an individual) should leak no information about the sample.

Definition 5 (Kasiviswanathan et al. (2011)) *Let A be an algorithm that gets an input $S = (z_1,\ldots,z_m)$. Algorithm A is an* (α,β,ϵ)-PPAC learner *for a concept class $\mathcal{C}$ over X_d using hypothesis class $\mathcal{H}$ and sample size m if*

Privacy. *Algorithm A is ϵ-differentially private (as formulated in Definition 1);*

Utility. *Algorithm A is an* (α,β)-PAC learner *for $\mathcal{C}$ using $\mathcal{H}$ and sample size m (as formulated in Definition 3).*

2.4. The Exponential Mechanism

We next describe the exponential mechanism of McSherry and Talwar (2007). We present its private learning variant; however, it can be used in more general scenarios. The goal here is to choose a hypothesis $h \in \mathcal{H}$ approximately minimizing the empirical error. The choice is probabilistic, where the probability mass that is assigned to each hypothesis decreases exponentially with its empirical error.

Inputs: a privacy parameter ϵ, a hypothesis class $\mathcal{H}$, and m labeled samples $S = (x_i, y_i)_{i=1}^m$.

1. $\forall h \in \mathcal{H}$ define $q(S,h) = |\{i : h(x_i) = y_i\}|$.
2. Randomly choose $h \in \mathcal{H}$ with probability

$$\frac{\exp\left(\epsilon \cdot q(S,h)/2\right)}{\sum_{f\in\mathcal{H}} \exp\left(\epsilon \cdot q(S,f)/2\right)}.$$

Proposition 6 *Denote $\hat{e} \triangleq \min_{f\in\mathcal{H}}\{\text{error}_S(f)\}$. The probability that the exponential mechanism outputs a hypothesis h such that $\text{error}_S(h) > \hat{e} + \Delta$ is at most $|\mathcal{H}|\cdot\exp(-\epsilon\Delta m/2)$. Moreover, the exponential mechanism is ϵ differentially private.*

2.5. Concentration Bounds

Let $X_1, \ldots, X_n$ be independent random variables where $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$ for some $0 < p < 1$. Clearly, $\mathbb{E}[\sum_i X_i] = pn$. Chernoff and Hoeffding bounds show that the sum is concentrated around this expected value:

$$\begin{aligned}
\Pr\left[\sum\nolimits_i X_i > (1+\delta)pn\right] &\le \exp\left(-pn\delta^2/3\right) \quad \text{for } \delta > 0,\\
\Pr\left[\sum\nolimits_i X_i < (1-\delta)pn\right] &\le \exp\left(-pn\delta^2/2\right) \quad \text{for } 0 < \delta < 1,\\
\Pr\left[\left|\sum\nolimits_i X_i - pn\right| > \delta\right] &\le 2\exp\left(-2\delta^2/n\right) \quad \text{for } \delta \ge 0.
\end{aligned}$$

The first two inequalities are known as the multiplicative Chernoff bounds (Chernoff, 1952), and the last inequality is known as the Hoeffding bound (Hoeffding, 1963).

3. The Representation Dimension

In this section we present a combinatorial measure of a concept class $\mathcal{C}$ that characterizes the sample complexity necessary and sufficient for learning $\mathcal{C}$ under pure differential privacy. The measure is a *probabilistic representation* of the class $\mathcal{C}$. We start with the notation of deterministic representation of Beimel et al. (2014).

Definition 7 (Beimel et al. (2014)) *A hypothesis class $\mathcal{H}$ is an α-representation for a class $\mathcal{C}$ if for every $c \in \mathcal{C}$ and every distribution $\mathcal{D}$ on X_d there exists a hypothesis $h \in \mathcal{H}$ such that* $\mathrm{error}_{\mathcal{D}}(c, h) \le \alpha$.

Example 1 ($\mathtt{POINT}_d$) *For $j \in X_d$, define $c_j : X_d \to \{0,1\}$ as $c_j(x) = 1$ if $x = j$, and $c_j(x) = 0$ otherwise. Define $\mathtt{POINT}_d = \{c_j\}_{j \in X_d}$. Beimel et al. (2014) showed that for $\alpha < 1/2$, every α-representation for $\mathtt{POINT}_d$ must be of cardinality at least d, and that an α-representation $\mathcal{H}_d$ for $\mathtt{POINT}_d$ exists where $|\mathcal{H}_d| = O(d/\alpha^2)$.*

The above representation can be used for non-private learning, by taking a big enough sample and finding a hypothesis $h \in \mathcal{H}_d$ minimizing the empirical error. For *private* learning, Beimel et al. (2014) showed that a sample of size $O_{\alpha,\beta,\epsilon}(\log|\mathcal{H}_d|)$ suffices, with a learner that employs the exponential mechanism to choose a hypothesis from $\mathcal{H}_d$.

Definition 8 *For a hypothesis class $\mathcal{H}$ we denote* $\mathrm{size}(\mathcal{H}) = \ln|\mathcal{H}|$. *We define the Deterministic Representation Dimension of a concept class $\mathcal{C}$ as*

$$\mathrm{DRepDim}(\mathcal{C}) = \min\left\{\mathrm{size}(\mathcal{H}) : \mathcal{H} \text{ is a } \frac{1}{4}\text{-representation for } \mathcal{C}\right\}.$$

Remark 9 *Choosing $\frac{1}{4}$ is arbitrary; we could have chosen any (smaller than $\frac{1}{2}$) constant.*

Example 2 *By the results of Beimel et al. (2014), stated in the previous example,* $\mathrm{DRepDim}(\mathtt{POINT}_d) = \theta(\ln(d))$.

We are now ready to present the notion of a probabilistic representation. The idea behind this notion is that we have a list of hypothesis classes, such that for every concept c and distribution $\mathcal{D}$, if we sample a hypothesis class from the list, then with high probability it contains a hypothesis that is close to c.

Definition 10 *Let $\mathcal{P}$ be a distribution over $\{1, 2, \ldots, r\}$, and let $\mathscr{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_r\}$ be a family of hypothesis classes (every $\mathcal{H}_i \in \mathscr{H}$ is a set of boolean functions). We say that $(\mathscr{H}, \mathcal{P})$ is an (α, β)-probabilistic representation for a class $\mathcal{C}$ if for every $c \in \mathcal{C}$ and every distribution $\mathcal{D}$ on X_d:*

$$\Pr_{\mathcal{P}}\left[\exists h \in \mathcal{H}_i \ \text{ s.t. } \ \mathrm{error}_{\mathcal{D}}(c, h) \le \alpha\right] \ge 1 - \beta.$$

The probability is over randomly choosing a set $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$.

Remark 11 *As we will see in Section 4, the existence of such a probabilistic representation $(\mathscr{H}, \mathcal{P})$ for a concept class $\mathcal{C}$ implies the existence of a private learning algorithm for $\mathcal{C}$ with sample complexity that depends on the cardinality of the hypothesis classes $\mathcal{H}_i \in \mathscr{H}$. The sample complexity will not depend on $r = |\mathscr{H}|$. Nevertheless, there always exists a probabilistic representation in which r is bounded (see Appendix 7 for more details).*

Example 3 ($\mathtt{POINT}_d$) *In Section 7 we construct for every α and every β a pair $(\mathscr{H}, \mathcal{P})$ that (α, β)-probabilistically represents the class $\mathtt{POINT}_d$, where $\mathscr{H}$ contains all the sets of at most $\frac{4}{\alpha}\ln(1/\beta)$ boolean functions.*

Definition 12 *Let $\mathscr{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_r\}$ be a family of hypothesis classes. We denote $|\mathscr{H}| = r$, and* $\mathrm{size}(\mathscr{H}) = \max\{\ \ln|\mathcal{H}_i| : \mathcal{H}_i \in \mathscr{H}\ \}$. *We define the Representation Dimension of a concept class $\mathcal{C}$ as*

$$\mathrm{RepDim}(\mathcal{C}) = \min\left\{\ \mathrm{size}(\mathscr{H}) \ : \ \begin{array}{l}\exists \mathcal{P} \text{ s.t. } (\mathscr{H}, \mathcal{P}) \text{ is a}\\ (\frac{1}{4}, \frac{1}{4})\text{-probabilistic}\\ \text{representation for } \mathcal{C}\end{array}\ \right\}.$$

Remark 13 *Choosing $\alpha = \beta = \frac{1}{4}$ is arbitrary; we could have chosen any constants $0 < \alpha < \frac{1}{2}$ and $0 < \beta < 1$.*

Example 4 ($\mathtt{POINT}_d$) *The* size *of the probabilistic representation mentioned in Example 3 is $\ln(\frac{4}{\alpha}\ln(1/\beta))$. Placing $\alpha = \beta = \frac{1}{4}$, we see that the Representation Dimension of $\mathtt{POINT}_d$ is constant.*

4. Equivalence of Probabilistic Representation and Private Learning

We now show that $\mathrm{RepDim}(\mathcal{C})$ characterizes the sample complexity of private learners. We start by showing in Lemma 14 that an (α, β)-probabilistic representation of $\mathcal{C}$ implies a private learning algorithm whose sample complexity is the size of the representation. We then show in Lemma 16 that if there is a private learning algorithm with sample complexity m, then there is probabilistic representation of $\mathcal{C}$ of size $O(m)$; this lemma implies that $\mathrm{RepDim}(\mathcal{C})$ is a lower bound on the sample complexity. Recall that $\mathrm{RepDim}(\mathcal{C})$ is the size

of the smallest probabilistic representation for $\alpha = \beta = 1/4$. Thus, to complete the proof we show in Lemma 18 that a probabilistic representation with $\alpha = \beta = 1/4$ implies a probabilistic representation for arbitrary α and β.

Lemma 14 *If there a exists pair $(\mathscr{H}, \mathcal{P})$ that (α, β)-probabilistically represents a class $\mathcal{C}$, then for every ϵ there exists an algorithm A that $(6\alpha, 4\beta, \epsilon)$-PPAC learns $\mathcal{C}$ with a sample size $m = O\left(\frac{1}{\alpha\epsilon}(\text{size}(\mathscr{H}) + \ln(\frac{1}{\beta}))\right)$.*

Proof Let $(\mathscr{H}, \mathcal{P})$ be an (α, β)-probabilistic representation for the class $\mathcal{C}$, and consider the following algorithm A:

> Inputs: $S = (x_i, y_i)_{i=1}^m$, and a privacy parameter ϵ.
> 1. Randomly choose $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$.
> 2. Choose $h \in \mathcal{H}_i$ using the exp. mechanism with ϵ.

By the properties of the exponential mechanism, A is ϵ-differentially private. We will show that with sample size $m = O\left(\frac{1}{\alpha\epsilon}(\text{size}(\mathscr{H}) + \ln(\frac{1}{\beta}))\right)$, algorithm A is a $(6\alpha, 4\beta)$-PAC learner for $\mathcal{C}$. Fix some $c \in \mathcal{C}$ and $\mathcal{D}$, and define the following 3 good events:

E_1 $\mathcal{H}_i$ chosen in step 1 contains at least one hypothesis h s.t. $\text{error}_S(h) \leq 2\alpha$.

E_2 For every $h \in \mathcal{H}_i$ s.t. $\text{error}_S(h) \leq 3\alpha$, it holds that $\text{error}_{\mathcal{D}}(c, h) \leq 6\alpha$

E_3 The exponential mechanism chooses an h such that $\text{error}_S(h) \leq \alpha + \min_{f \in \mathcal{H}_i}\{\text{error}_S(f)\}$.

We first show that if those 3 good events happen, algorithm A returns a 6α-good hypothesis. Event E_1 ensures the existence of a hypothesis $f \in \mathcal{H}_i$ s.t. $\text{error}_S(f) \leq 2\alpha$. Thus, event $E_1 \cap E_3$ ensures algorithm A chooses (using the exponential mechanism) a hypothesis $h \in \mathcal{H}_i$ s.t. $\text{error}_S(h) \leq 3\alpha$. Event E_2 ensures therefore that this h obeys $\text{error}_{\mathcal{D}}(c, h) \leq 6\alpha$.

We will now show that those 3 events happen with high probability. As $(\mathscr{H}, \mathcal{P})$ is an (α, β)-probabilistic representation for the class $\mathcal{C}$, the chosen $\mathcal{H}_i$ contains a hypothesis h s.t. $\text{error}_{\mathcal{D}}(c, h) \leq \alpha$ with probability at least $1 - \beta$; by the Chernoff bound with probability at least $1 - \exp(-m\alpha/3)$ this hypothesis has empirical error at most 2α. Event E_1 happens with probability at least $(1 - \beta)(1 - \exp(-m\alpha/3)) > 1 - (\beta + \exp(-m\alpha/3))$, which is at least $(1 - 2\beta)$ for $m \geq \frac{3}{\alpha}\ln(1/\beta)$.

Using the Chernoff bound, the probability that a hypothesis h s.t. $\text{error}_{\mathcal{D}}(c, h) > 6\alpha$ has empirical error $\leq 3\alpha$ is less than $\exp(-m\alpha 3/4)$. Using the union bound, the probability that there is such a hypothesis in $\mathcal{H}_i$ is at most $|\mathcal{H}_i| \cdot \exp(-m\alpha 3/4)$. Therefore, $\Pr[E_2] \geq 1 - |\mathcal{H}_i| \cdot \exp(-m\alpha 3/4)$. For $m \geq \frac{4}{3\alpha}(\ln(\frac{|\mathcal{H}_i|}{\beta}))$, this probability is at least $(1 - \beta)$.

The exponential mechanism ensures that the probability of event E_3 is at least $1 - |\mathcal{H}_i| \cdot \exp(-\epsilon\alpha m/2)$ (see Section 2.4), which is at least $(1 - \beta)$ for $m \geq \frac{2}{\alpha\epsilon}\ln(\frac{|\mathcal{H}_i|}{\beta})$.

All in all, by setting $m = \frac{3}{\alpha\epsilon}(\text{size}(\mathscr{H}) + \ln(\frac{1}{\beta}))$ we ensure that the probability of A failing to output a 6α-good hypothesis is at most 4β. ■

We will demonstrate the above lemma with two examples:

Example 5 (Efficient learner for $\texttt{POINT}_d$) *As described in Example 3, there exists an $(\mathscr{H}, \mathcal{P})$ that $(\alpha/6, \beta/4)$-probabilistically represents the class $\texttt{POINT}_d$, where $\text{size}(\mathscr{H}) = O_{\alpha,\beta,\epsilon}(1)$. By Lemma 14, there exists an algorithm that $(\alpha, \beta, \epsilon)$-PPAC learns $\mathcal{C}$ with sample size $m = O_{\alpha,\beta,\epsilon}(1)$.*

The existence of an algorithm with sample complexity $O(1)$ was already proven by Beimel et al. (2014). Moreover, assuming the existence of oneway functions, their learner is efficient. Our constructions yields an efficient learner, without assumptions. To see this, consider again algorithm A presented in the above proof, and note that as $\text{size}(\mathscr{H})$ is constant, step 2 could be done in constant time. Step 1 can be done efficiently as we can efficiently sample a set $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$. In Claim 35 we initially construct a probabilistic representation in which the description of every hypothesis is exponential in d. The representation is then revised using pairwise independence to yield a representation in which every hypothesis h has a short description, and given x the value $h(x)$ can be computed efficiently.

Example 6 ($\texttt{POINT}_{\mathbb{N}}$) *Consider the class $\texttt{POINT}_{\mathbb{N}}$, which is exactly like $\texttt{POINT}_d$, only over the natural numbers. By results of Chaudhuri and Hsu (2011) and Beimel et al. (2014), it is impossible to properly PPAC learn the class $\texttt{POINT}_{\mathbb{N}}$. Our construction can yield an (inefficient) improper private learner for $\texttt{POINT}_{\mathbb{N}}$ with $O_{\alpha,\beta,\epsilon}(1)$ samples. The details are deferred to Section 7.*

The next lemma shows that a private learning algorithm implies a probabilistic representation. This lemma can be used to lower bound the sample complexity of private learners.

Lemma 15 *If there exists an algorithm A that $(\alpha, \frac{1}{2}, \epsilon)$-PPAC learns a concept class $\mathcal{C}$ with a sample size m, then there exists a pair $(\mathscr{H}, \mathcal{P})$ that $(\alpha, 1/4)$-probabilistically represents the class $\mathcal{C}$ such that $\text{size}(\mathscr{H}) = O(m\epsilon)$.*

Proof Let A be an $(\alpha, \frac{1}{2}, \epsilon)$-PPAC learner for a class $\mathcal{C}$ using hypothesis class $\mathcal{F}$ whose sample size is m. For a target concept $c \in \mathcal{C}$ and a distribution $\mathcal{D}$ on X_d, we define G as the set of all hypotheses $h \in \mathcal{F}$ such that $\text{error}_{\mathcal{D}}(c, h) \leq \alpha$. Fix some $c \in \mathcal{C}$ and a distribution $\mathcal{D}$ on X_d. As A is an $(\alpha, \frac{1}{2})$-PAC learner, $\Pr_{\mathcal{D},A}[A(S) \in G] \geq \frac{1}{2}$, where the probability is over A's randomness and over sampling the examples in S (according to $\mathcal{D}$). Therefore, there exists a database S of m samples such that $\Pr_A[A(S) \in G] \geq \frac{1}{2}$, where the probability is only over the randomness of A. As A is ϵ-differentially private, $\Pr_A\left[A(\vec{0}) \in G\right] \geq e^{-m\epsilon} \cdot \Pr_A[A(S) \in G] \geq \frac{1}{2}e^{-m\epsilon}$, where $\vec{0}$ is a database with m zeros.[1] That is, $\Pr_A\left[A(\vec{0}) \notin G\right] \leq 1 - \frac{1}{2}e^{-m\epsilon}$. Now, consider a set $\mathcal{H}$ containing the outcomes of $2\ln(4)e^{m\epsilon}$ executions of $A(\vec{0})$. The probability that $\mathcal{H}$ does not contain an α-good hypothesis is at most $(1 - \frac{1}{2}e^{-m\epsilon})^{2\ln(4)e^{m\epsilon}} \leq \frac{1}{4}$. Thus, $\mathscr{H} = \{\mathcal{H} \subseteq \mathcal{F} : |\mathcal{H}| \leq 2\ln(4)e^{m\epsilon}\}$, and $\mathcal{P}$, the distribution induced by $A(\vec{0})$, are an $(\alpha, 1/4)$-probabilistic representation for class $\mathcal{C}$. It follows that $\text{size}(\mathscr{H}) = \max\{\ \ln|\mathcal{H}| : \mathcal{H} \in \mathscr{H}\ \} = \ln(2\ln(4)) + m\epsilon$. ■

1. Choosing $\vec{0}$ is arbitrary; we could have chosen any database.

The above lemma yields a lower bound of $\Omega\left(\frac{1}{\epsilon}\operatorname{RepDim}(\mathcal{C})\right)$ on the sample complexity of private learners for a concept class $\mathcal{C}$. To see this, fix $\alpha \le \frac{1}{4}$ and let A be an $(\alpha, \frac{1}{2}, \epsilon)$-PPAC learner for $\mathcal{C}$ with sample size m. By the above lemma, there exists a pair $(\mathscr{H}, \mathcal{P})$ that $(\alpha, 1/4)$-probabilistically represents $\mathcal{C}$ s.t. $\operatorname{size}(\mathscr{H}) = \ln(2\ln(4)) + m\epsilon$. Therefore, by definition, $\operatorname{RepDim}(\mathcal{C}) \le \ln(2\ln(4)) + m\epsilon$. Thus, $m \ge \frac{1}{\epsilon}(\operatorname{RepDim}(\mathcal{C}) - \ln(2\ln(4))) = \Omega\left(\frac{1}{\epsilon}\operatorname{RepDim}(\mathcal{C})\right)$.

In order to refine this lower bound (and incorporate α in it), we will need a somewhat stronger version of this lemma:

Lemma 16 *Let $\alpha \le 1/4$. If there exists an algorithm A that $(\alpha, \frac{1}{2}, \epsilon)$-PPAC learns a concept class $\mathcal{C}$ with a sample size m, then there exists a pair $(\mathscr{H}, \mathcal{P})$ that $(1/4, 1/4)$-probabilistically represents the class $\mathcal{C}$ such that* $\operatorname{size}(\mathscr{H}) = O(m\epsilon\alpha)$.

Proof Let A be an $(\alpha, \frac{1}{2}, \epsilon)$-PPAC learner for the class $\mathcal{C}$ using hypothesis class $\mathcal{F}$ whose sample size is m. Without loss of generality, we can assume that $m \ge \frac{3\ln(4)}{4\alpha}$ (since A can ignore part of the sample). For a target concept $c \in \mathcal{C}$ and a distribution $\mathcal{D}$ on X_d, we define

$$G^{\alpha}_{\mathcal{D}} = \{h \in \mathcal{F} : \operatorname{error}_{\mathcal{D}}(c, h) \le \alpha\}.$$

Fix some $c \in \mathcal{C}$ and a distribution $\mathcal{D}$ on X_d, and define the following distribution $\widetilde{\mathcal{D}}$ on X_d:

$$\Pr_{\widetilde{\mathcal{D}}}[x] = \begin{cases} 1 - 4\alpha + 4\alpha \cdot \Pr_{\mathcal{D}}[x], & x = 0^d. \\ 4\alpha \cdot \Pr_{\mathcal{D}}[x], & x \ne 0^d. \end{cases}$$

Note that for every $x \in X_d$,

$$\Pr_{\widetilde{\mathcal{D}}}[x] \ge 4\alpha \cdot \Pr_{\mathcal{D}}[x]. \tag{2}$$

As A is an $(\alpha, \frac{1}{2})$-PAC learner, it holds that

$$\Pr_{\widetilde{\mathcal{D}}, A}\left[A(S) \in G^{\alpha}_{\mathcal{D}}\right] \ge \frac{1}{2},$$

where the probability is over A's randomness and over sampling the examples in S (according to $\widetilde{\mathcal{D}}$). In addition, by inequality (2), every hypothesis h with $\operatorname{error}_{\mathcal{D}}(c, h) > 1/4$ has error strictly greater than α under $\widetilde{\mathcal{D}}$:

$$\operatorname{error}_{\widetilde{\mathcal{D}}}(c, h) \ge 4\alpha \cdot \operatorname{error}_{\mathcal{D}}(c, h) > \alpha.$$

So, every α-good hypothesis for c and $\widetilde{\mathcal{D}}$ is a $\frac{1}{4}$-good hypothesis for c and $\mathcal{D}$. That is, $G^{\alpha}_{\widetilde{\mathcal{D}}} \subseteq G^{1/4}_{\mathcal{D}}$. Therefore, $\Pr_{\widetilde{\mathcal{D}}, A}\left[A(S) \in G^{1/4}_{\mathcal{D}}\right] \ge \frac{1}{2}$.

We say that a database S of m labeled examples is *good* if the unlabeled example 0^d appears in S at least $(1 - 8\alpha)m$ times. Let S be a database constructed by taking m i.i.d. samples from $\widetilde{\mathcal{D}}$, labeled by c. By the Chernoff bound, S is good with probability at least $1 - \exp(-4\alpha m/3)$. Hence,

$$\Pr_{\widetilde{\mathcal{D}}, A}\left[(A(S) \in G^{1/4}_{\mathcal{D}}) \wedge (S \text{ is good})\right] \ge \frac{1}{2} - \exp(-4\alpha m/3) \ge \frac{1}{4}.$$

Therefore, there exists a database S_{good} of m samples that contains the unlabeled sample 0^d at least $(1 - 8\alpha)m$ times, and $\Pr_A\left[A(S_{\text{good}}) \in G^{1/4}_{\mathcal{D}}\right] \ge \frac{1}{4}$, where the probability is only over the randomness of A. All of the examples in S_{good} (including the example 0^d) are labeled by c.

For $\sigma \in \{0, 1\}$, denote by $\vec{0}_\sigma$ a database containing m copies of the example 0^d labeled as σ. As A is ϵ-differentially private, and as the target concept c labels the example 0^d by either 0 or 1, for at least one $\sigma \in \{0, 1\}$ it holds that

$$\begin{aligned} \Pr_A[A(\vec{0}_\sigma) \in G^{1/4}_{\mathcal{D}}] &\ge \exp(-8\alpha\epsilon m) \cdot \Pr_A\left[A(S_{\text{good}}) \in G^{1/4}_{\mathcal{D}}\right] \\ &\ge \exp(-8\alpha\epsilon m) \cdot 1/4. \end{aligned} \tag{3}$$

That is, $\Pr_A[A(\vec{0}_\sigma) \notin G^{1/4}_{\mathcal{D}}] \le 1 - \frac{1}{4}e^{-8\alpha\epsilon m}$. Now, consider a set $\mathcal{H}$ containing the outcomes of $4\ln(4)e^{8\alpha\epsilon m}$ executions of $A(\vec{0}_0)$, and the outcomes of $4\ln(4)e^{8\alpha\epsilon m}$ executions of $A(\vec{0}_1)$. The probability that $\mathcal{H}$ does not contain a $\frac{1}{4}$-good hypothesis for c and $\mathcal{D}$ is at most $(1 - \frac{1}{4}e^{-8\alpha\epsilon m})^{4\ln(4)e^{8\alpha\epsilon m}} \le \frac{1}{4}$. Thus, $\mathscr{H} = \left\{\mathcal{H} \subseteq \mathcal{F} \,:\, |\mathcal{H}| \le 2 \cdot 4\ln(4)e^{8\alpha\epsilon m}\right\}$, and $\mathcal{P}$, the distribution induced by $A(\vec{0}_0)$ and $A(\vec{0}_1)$, are a $(1/4, 1/4)$-probabilistic representation for the class $\mathcal{C}$. Note that the value $c(0^d)$ is unknown, and can be either 0 or 1. Therefore the construction uses the two possible values (one of them correct).

It holds that $\operatorname{size}(\mathscr{H}) = \max\{\ln|\mathcal{H}| : \mathcal{H} \in \mathscr{H}\} = \ln(8\ln(4)) + 8\alpha\epsilon m = O(m\epsilon\alpha)$. ■

Lemma 18 shows how to construct a probabilistic representation for an arbitrary α and β from a probabilistic representation with $\alpha = \beta = 1/4$; in other words we boost α and β. The proof of this lemma is combinatorial. It allows us to start with a private learning algorithm with constant α and β, move to a representation, use the combinatorial boosting, and move back to a private algorithm with small α and β. This should be contrasted with the private boosting of Dwork et al. (2010) which is algorithmic and more complicated (however, the algorithm of Dwork et al. (2010) is computationally efficient).

We first show how to construct a probabilistic representation for arbitrary β from a probabilistic representation with $\beta = 1/4$.

Claim 17 *For every concept class $\mathcal{C}$ and for every β, there exists a pair $(\mathscr{H}, \mathcal{P})$ that $(1/4, \beta)$-probabilistically represents $\mathcal{C}$ where* $\operatorname{size}(\mathscr{H}) \le \operatorname{RepDim}(\mathcal{C}) + \ln\ln(1/\beta)$.

Proof Let $\beta < 1/4$, and let $(\mathscr{H}^0, \mathcal{P}^0)$ be a $(\frac{1}{4}, \frac{1}{4})$- probabilistic representation for $\mathcal{C}$ with $\operatorname{size}(\mathscr{H}^0) = \operatorname{RepDim}(\mathcal{C}) \triangleq k_0$ (that is, for every $\mathcal{H}^0_i \in \mathscr{H}^0$ it holds that $|\mathcal{H}^0_i| \le e^{k_0}$). Denote $\mathscr{H}^0 = \{\mathcal{H}^0_1, \mathcal{H}^0_2, \ldots, \mathcal{H}^0_r\}$, and consider the following family of hypothesis classes:

$$\mathscr{H}^1 = \left\{\mathcal{H}^0_{i_1} \cup \cdots \cup \mathcal{H}^0_{i_{\ln(1/\beta)}} \,:\, 1 \le i_1 \le \cdots \le i_{\ln(1/\beta)} \le r\right\}.$$

Note that for every $\mathcal{H}^1_i \in \mathscr{H}^1$ it holds that $|\mathcal{H}^1_i| \le \ln(1/\beta)e^{k_0}$ and so $\operatorname{size}(\mathscr{H}^1) \triangleq k_1 \le k_0 + \ln\ln(1/\beta)$. We will now show an appropriate distribution $\mathcal{P}^1$ on $\mathscr{H}^1$ s.t. $(\mathscr{H}^1, \mathcal{P}^1)$ is

a $(\frac{1}{4}, \beta)$-probabilistic representation for $\mathcal{C}$. To this end, consider the following process for randomly choosing an $\mathcal{H}^1 \in \mathscr{H}^1$:

1. Denote $M = \ln(1/\beta)$
2. For $i = 1, \ldots, M$:
   Randomly choose $\mathcal{H}_i^0 \in_{\mathcal{P}_0} \mathscr{H}^0$.
3. Return $\mathcal{H}^1 = \bigcup_{i=1}^{M} \mathcal{H}_i^0$.

The above process induces a distribution on $\mathscr{H}^1$, denoted as $\mathcal{P}^1$. As $\mathscr{H}^0$ is a $(\frac{1}{4}, \frac{1}{4})$-probabilistic representation for $\mathcal{C}$, we have that

$$\Pr_{\mathcal{P}^1}\left[\nexists h \in \mathcal{H}^1 \; s.t. \; \mathrm{error}_{\mathcal{D}}(c,h) \leq 1/4\right] =$$
$$= \prod_{i=1}^{M} \Pr_{\mathcal{P}^0}\left[\nexists h \in \mathcal{H}_i^0 \; s.t. \; \mathrm{error}_{\mathcal{D}}(c,h) \leq 1/4\right] \leq$$
$$\leq \left(\frac{1}{4}\right)^M \leq \beta.$$

■

Lemma 18 *For every concept class $\mathcal{C}$, every α, and every β, there exists $(\mathscr{H}, \mathcal{P})$ that (α, β)-probabilistically represents $\mathcal{C}$ where*

$$\mathrm{size}(\mathscr{H}) = O\Big(\ln(\frac{1}{\alpha}) \cdot \big(\mathrm{RepDim}(\mathcal{C}) + \ln\ln\ln(\frac{1}{\alpha}) + \ln\ln(\frac{1}{\beta})\big)\Big).$$

Lemma 18 corresponds to standard accuracy amplification arguments. We defer the proof to Appendix 7. The next theorem states the main result of this section – RepDim characterizes the sample complexity of private learning.

Theorem 19 *Let $\mathcal{C}$ be a concept class. $\widetilde{\Theta}_\beta\left(\frac{\mathrm{RepDim}(\mathcal{C})}{\alpha\epsilon}\right)$ samples are necessary and sufficient for the private learning of the class $\mathcal{C}$.*

Proof Fix some $\alpha \leq 1/4, \beta \leq 1/2$, and ϵ. By Lemma 18, there exists a pair $(\mathscr{H}, \mathcal{P})$ that $(\frac{\alpha}{6}, \frac{\beta}{4})$-represent class $\mathcal{C}$, where $\mathrm{size}(\mathscr{H}) = O\Big(\ln(1/\alpha) \cdot \big(\mathrm{RepDim}(\mathcal{C}) + \ln\ln\ln(1/\alpha) + \ln\ln(1/\beta)\big)\Big)$. Therefore, by Lemma 14, there exists an algorithm A that $(\alpha, \beta, \epsilon)$-PPAC learns the class $\mathcal{C}$ with a sample size

$$m = O_\beta\left(\frac{1}{\alpha\epsilon}\ln(\frac{1}{\alpha}) \cdot \left(\mathrm{RepDim}(\mathcal{C}) + \ln\ln\ln(\frac{1}{\alpha})\right)\right).$$

For the lower bound, let A be an $(\alpha, \beta, \epsilon)$-PPAC learner for the class $\mathcal{C}$ with a sample size m, where $\alpha \leq 1/4$ and $\beta \leq 1/2$. By Lemma 16, there exists an $(\mathscr{H}, \mathcal{P})$ that $(\frac{1}{4}, \frac{1}{4})$-

probabilistically represents the class $\mathcal{C}$ and $\mathrm{size}(\mathscr{H}) = \ln(8) + \ln\ln(4) + 8\alpha\epsilon m$. Therefore, by definition, $\mathrm{RepDim}(\mathcal{C}) \leq \ln(8\ln(4)) + 8\alpha\epsilon m$. Thus,

$$m \geq \frac{1}{8\alpha\epsilon} \cdot \big(\mathrm{RepDim}(\mathcal{C}) - \ln(8\ln(4))\big) = \Omega\left(\frac{\mathrm{RepDim}(\mathcal{C})}{\alpha\epsilon}\right).$$

■

5. Probabilistic Representation for Privately Solving Optimization Problems

The notion of probabilistic representation applies not only to private learning, but also to a broader task of optimization problems. We consider the following scenario:

Definition 20 *An* optimization problem OPT *over a universe X and a set of solutions $\mathcal{F}$ is defined by a quality function $q : X^* \times \mathcal{F} \to [0,1]$. Given a database S, the task is to choose a solution $f \in \mathcal{F}$ such that $q(S,f)$ is maximized.*

Notation. We will refer to the optimization problem defined by a quality function q as OPT_q.

Definition 21 *An α-*good *solution for a database S is a solution s such that $q(S,s) \geq \max_{f\in\mathcal{F}}\{q(S,f)\} - \alpha$.*

Given an optimization problem OPT_q, one can use the exponential mechanism to choose a solution $s \in \mathcal{F}$. In general, this method achieves a reasonable solution only for databases of size $\Omega(\log|\mathcal{F}|/\epsilon)$. To see this, consider a case where there exists a database S of m records such that exactly one solution $t \in \mathcal{F}$ has a quality of $q(S,t) = 1$, and every other $f \in \mathcal{F}$ has a quality of $q(S,f) = 1/2$. The probability of the exponential mechanism choosing t is:

$$\Pr[t \text{ is chosen}] = \frac{\exp(\epsilon m/2)}{(|\mathcal{F}|-1)\cdot\exp(\epsilon m/4) + \exp(\epsilon m/2)}.$$

Unless

$$m \geq \tfrac{4}{\epsilon}\ln(|\mathcal{F}|-1) = \Omega(\tfrac{1}{\epsilon}\ln|\mathcal{F}|), \tag{4}$$

the above probability is strictly less than $1/2$. Using our notations of probabilistic representation, it might be possible to reduce the necessary database size.

Consider using the exponential mechanism for choosing a solution s, not out of $\mathcal{F}$, but rather from a smaller set of solutions $\mathcal{B}$. Roughly speaking, the factor of $\ln|\mathcal{F}|$ in requirement (4) will now be replaced with $\ln|\mathcal{B}|$, which corresponds to size of the representation. Therefore, the database size m should be at least $\ln|\mathcal{B}|/\epsilon$. So m needs to be bigger than the size of the representation by at least a factor of $1/\epsilon$.

In the following analysis we will denote this required gap, i.e., $m/\ln|\mathcal{B}|$, as Δ. We will see that the existence of a private approximation algorithm implies a probabilistic representation with $1 < \Delta \approx \frac{1}{\epsilon}$, and that a probabilistic representation with $\Delta > 1$ implies a private approximation algorithm. Bigger Δ corresponds to better privacy; however, it might be harder to achieve.

Definition 22 *Let* OPT_q *be an optimization problem over a universe* X *and a set of solutions* $\mathcal{F}$*. Let* $\mathcal{B}$ *be a set of solutions, and denote* $\mathrm{size}(\mathcal{B}) = \ln|\mathcal{B}|$*. We say that* $\mathcal{B}$ *is an* α*-deterministic representation of* OPT_q *for databases of* m *elements if for every* $S \in X^m$ *there exists a solution* $s \in \mathcal{B}$ *such that* $q(S,s) \geq \max_{f\in\mathcal{F}}\{q(S,f)\} - \alpha$*.*

Definition 23 *Let* $\mathcal{B}$ *be an* α*-deterministic representation of* OPT_q *for databases of* m *elements. Denote* $\Delta \triangleq \frac{m}{\mathrm{size}(\mathcal{B})}$*. If* $\Delta > 1$*, then we say that the* ratio *of* $\mathcal{B}$ *is* Δ*.*

An α-deterministic representation $\mathcal{B}$ with ratio Δ is required to support all the databases of $m = \Delta \cdot \mathrm{size}(\mathcal{B})$ elements. That is, for every $S \in X^m$, the set $\mathcal{B}$ is required to contain at least one α-good solution.

Fix $S \in X^m$. Intuitively, Δ controls the ratio between m and number of bits needed to represent an α-good solution for S. As $\mathcal{B}$ contains an α-good solution for S, and assuming $\mathcal{B}$ is publicly known, this solution could be represented with $\ln|\mathcal{B}| = \mathrm{size}(\mathcal{B}) = m/\Delta$ bits.

Definition 24 *Let* OPT_q *be an optimization problem over a universe* X *and a set of solutions* $\mathcal{F}$*. Let* $\mathcal{P}$ *be a distribution over* $\{1,2,\ldots,r\}$*, and let* $\mathscr{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_r\}$ *be a family of solution sets for* OPT_q*. We denote* $\mathrm{size}(\mathscr{B}) = \max\{\ \ln|\mathcal{B}_i| : \mathcal{B}_i \in \mathscr{B}\ \}$*. We say that* $(\mathscr{B},\mathcal{P})$ *is an* (α,β)*-probabilistic representation of* OPT_q *for databases of* m *elements if for every* $S \in X^m$*:*

$$\Pr_{\mathcal{P}}\left[\exists s \in \mathcal{B}_i \ \ s.t.\ \ q(S,s) \geq \max_{f\in\mathcal{F}}\{q(S,f)\} - \alpha\right] \geq 1-\beta.$$

Definition 25 *Let* $(\mathscr{B},\mathcal{P})$ *be an* (α,β)*-probabilistic representation of* OPT_q *for databases of* m *elements. Denote* $\Delta \triangleq \frac{m}{\mathrm{size}(\mathscr{B})}$*. If* $\Delta > 1$*, then we say that the* ratio *of the representation is* Δ*.*

Definition 26 *An optimization problem* OPT_q *is* bounded *if* $\Big||S_1|\cdot q(S_1,f) - |S_2|\cdot q(S_2,f)\Big| \leq 1$ *for every solution* f *and every two neighboring databases* S_1, S_2*.*

We are interested in approximating bounded optimization problems, while guaranteeing differential privacy:

Definition 27 *Let* OPT_q *be a bounded optimization problem over a universe* X *and a set of solutions* $\mathcal{F}$*. An algorithm* A *is an* (α,β,ϵ)-private approximation algorithm *for* OPT_q *with a database of* m *records if:*

1. *Algorithm* A *is* ϵ*-differentially private (as formulated in Definition 1);*
2. *For every* $S \in X^m$*, algorithm* A *outputs with probability at least* $(1-\beta)$ *a solution* s *such that* $q(S,s) \geq \max_{f\in\mathcal{F}}\{q(S,f)\} - \alpha$*.*

Example 7 (Sanitization) *Consider a class of predicates* $\mathcal{C}$ *over* X*. A database* S *contains points taken from* X*. A predicate query* Q_c *for* $c \in \mathcal{C}$ *is defined as* $Q_c(S) = \frac{1}{|S|}\cdot|\{x_i \in S : c(x_i) = 1\}|$*. Blum et al. (2008) defined a sanitizer (or data release mechanism) as a differentially private algorithm that, on input a database* S*, outputs another database* $\hat{S}$ *with entries taken from* X*. A sanitizer* A *is* (α,β)*-useful for predicates in the class* $\mathcal{C}$ *if for every database* S *it holds that*

$$\Pr_{A}\left[\forall c \in C\ \ \left|Q_c(S) - Q_c(\hat{S})\right| \leq \alpha\right] \geq 1-\beta.$$

This scenario can be viewed as a bounded optimization problem: The solutions are sanitized databases. For an input database S *and and a sanitized database* $\hat{S}$*, the quality function is*

$$q(S,\hat{S}) = 1 - \max_{c\in C}\Big\{|Q_c(S) - Q_c(\hat{S})|\Big\}.$$

To see that this optimization problem is bounded, note that for every two neighboring databases S_1, S_2 *of* m *elements, and every* $c \in C$ *it holds that* $|Q_c(S_1) - Q_c(S_2)| \leq \frac{1}{m}$*. Therefore, for every sanitized database* f*,*

$$m\cdot|q(S_1,f) - q(S_2,f)| = m\cdot\left|\max_{c\in C}\{|Q_c(S_1) - Q_c(f)|\} - \max_{c\in C}\{|Q_c(S_2) - Q_c(f)|\}\right| \leq 1$$

Example 8 (Center points) *Let* $X \subseteq \mathbb{R}^d$ *be a finite domain. Given a database* $S \in X^m$*, consider the task of privately identifying a point* $x \in \mathbb{R}$ *that is "deep inside" the convex-hull of* S*. This problem was shown to be closely related to (privately) learning halfspaces (Beimel et al., 2019). In this context, "deep inside" is quantified using the notion of* Tukey depth *(Tukey, 1975). Specifically, the* Tukey depth *of a point* $x \in \mathbb{R}^d$ *w.r.t. the database* S*, denoted* $\mathrm{td}(S,x)$*, is the minimum number of points that need to be removed from* S *such that* x *is not in the convex-hull of the remaining points.*

The task of identifying a point with high Tukey depth can also be viewed as a bounded optimization problem, where the solutions are points in some subset $Y \subseteq \mathbb{R}^d$*. For an input database* $S \in X^m$ *and a solution* $y \in Y$*, the quality function is* $q(S,y) = \mathrm{td}(S,y)/|S|$*.*

The next two lemmas establish an equivalence between a private approximation algorithm and a probabilistic representation for a bounded optimization problem.

Lemma 28 *Let* OPT_q *be a bounded optimization problem over a universe* X*. If there exists a pair* $(\mathscr{B},\mathcal{P})$ *that* (α,β)*-probabilistically represents* OPT_q *for databases of* m *elements, s.t. the ratio of* $(\mathscr{B},\mathcal{P})$ *is* $\Delta > 1$*, then for every* $\hat{\alpha},\hat{\beta},\epsilon$ *satisfying*

$$\Delta \geq \frac{2}{\epsilon\hat{\alpha}}\left(1 + \frac{\ln(1/\hat{\beta})}{\mathrm{size}(\mathscr{B})}\right),$$

there exists an $\big((\alpha+\hat{\alpha}),(\beta+\hat{\beta}),\epsilon\big)$*-approximation algorithm for* OPT_q *with a database of size* m*.*

Proof Consider the following algorithm A:

Inputs: a database $S \in X^m$, and a privacy parameter ϵ.

1. Randomly choose $\mathcal{B}_i \in_{\mathcal{P}} \mathscr{B}$.
2. Choose $s \in \mathcal{B}_i$ using the exponential mechanism, that is, with probability

$$\frac{\exp(\epsilon \cdot m \cdot q(S,s)/2)}{\sum_{f \in \mathcal{B}_i} \exp(\epsilon \cdot m \cdot q(S,f)/2)}.$$

By the properties of the exponential mechanism, A is ϵ-differentially private. Fix a database $S \in X^m$, and define the following 2 bad events:

E_1 The set $\mathcal{B}_i$ chosen in step 1 does not contain a solution s s.t. $q(S,s) \geq \max_{f \in \mathcal{F}}\{q(S,f)\} - \alpha$.

E_2 The solution s chosen in step 2 is such that $q(S,s) < \max_{t \in \mathcal{B}_i} q(S,t) - \hat{\alpha}$.

Note that if those two bad events do not occur, algorithm A outputs a solution s such that $q(S,s) \geq \max_{f \in \mathcal{F}}\{q(S,f)\} - \alpha - \hat{\alpha}$. As $(\mathscr{B}, \mathcal{P})$ is an (α, β)-probabilistic representation of OPT_q for databases of size m, event E_1 happens with probability at most β. By the properties of the exponential mechanism, the probability of event E_2 is bounded by $|\mathcal{B}_i| \cdot \exp(-\epsilon m \hat{\alpha}/2)$. As $m = \Delta \operatorname{size}(\mathscr{B})$, this probability is at most

$$\begin{aligned}
\Pr[E_2] &\leq \operatorname{size}(\mathscr{B}) \cdot \exp(-\epsilon m \hat{\alpha}/2) \\
&= \operatorname{size}(\mathscr{B}) \cdot \exp(-\epsilon \Delta \operatorname{size}(\mathscr{B}) \hat{\alpha}/2) \\
&\leq \operatorname{size}(\mathscr{B}) \cdot \exp\left(-\left(1 + \frac{\ln(1/\hat{\beta})}{\operatorname{size}(\mathscr{B})}\right) \operatorname{size}(\mathscr{B})\right) \\
&= \operatorname{size}(\mathscr{B}) \cdot \exp(-\operatorname{size}(\mathscr{B}) - \ln(1/\hat{\beta})) = \hat{\beta}.
\end{aligned}$$

Therefore, algorithm A outputs an $(\alpha + \hat{\alpha})$-good solution with probability at least $(1 - \beta - \hat{\beta})$. ■

Lemma 29 *Let OPT_q be an optimization problem. If there exists an $(\alpha, \beta, \epsilon)$-private approximation algorithm for OPT_q with a database of m records, then for every $\hat{\beta}$ satisfying*

$$\Delta \triangleq \frac{m}{\ln(\frac{1}{1-\beta}) + \ln\ln(\frac{1}{\hat{\beta}}) + m \cdot \epsilon} > 1,$$

there exists a pair $(\mathscr{B}, \mathcal{P})$ that $(\alpha, \hat{\beta})$-probabilistically represents OPT_q for databases of m elements, where the ratio of the representation is Δ.

Proof Let A be an $(\alpha, \beta, \epsilon)$-private approximation algorithm for OPT_q, with a sample size m. Fix an arbitrary input database $S \in X^m$. Define G as the set of all solutions s, possibly outputted by A, such that $q(S,s) \geq \max_{f \in \mathcal{F}}\{q(S,f)\} - \alpha$. As A is an $(\alpha, \beta, \epsilon)$-approximation algorithm, $\Pr_A[A(S) \in G] \geq 1 - \beta$. As A is ϵ-differentially private, $\Pr_A\left[A(\vec{0}) \in G\right] \geq$

$(1-\beta)e^{-m\epsilon}$, where $\vec{0}$ is a database with m zeros. That is, $\Pr_A\left[A(\vec{0}) \notin G\right] \leq 1-(1-\beta)e^{-m\epsilon}$. Now, consider a set $\mathcal{B}$ containing the outcomes of $\Gamma \triangleq \frac{1}{1-\beta}\ln(\frac{1}{\hat{\beta}})e^{m\epsilon}$ executions of $A(\vec{0})$. The probability that $\mathcal{B}$ does not contain a solutions $s \in G$ is at most $(1-(1-\beta)e^{-m\epsilon})^{\Gamma} \leq \hat{\beta}$. Thus, $\mathscr{B} = \{\mathcal{B} \subseteq \mathit{support}(A) : |\mathcal{B}| \leq \Gamma\}$, and $\mathcal{P}$, the distribution induced by $A(\vec{0})$, are an $(\alpha, \hat{\beta})$-probabilistic representation of OPT_q for databases with m elements. Moreover, the ratio of the representation is

$$\begin{aligned}
\frac{m}{\operatorname{size}(\mathscr{B})} &= \frac{m}{\max\{\ln|\mathcal{B}| : \mathcal{B} \in \mathscr{B}\}} \\
&= \frac{m}{\ln(\frac{1}{1-\beta}) + \ln\ln(\frac{1}{\hat{\beta}}) + m\epsilon} = \Delta.
\end{aligned}$$

■

5.1. Exact 3SAT

Consider the following bounded optimization problem, denoted as $\mathrm{OPT}_{\mathrm{E3SAT}}$: The universe X is the set of all possible clauses with exactly 3 different literals over n variables, and the set of solutions $\mathcal{F}$ is the set of all possible 2^n assignments. Given a database $S = (\sigma_1, \sigma_2, \ldots, \sigma_m)$ containing m E3CNF clauses, the quality of an assignment $a \in \mathcal{F}$ is

$$q(S,a) = \frac{|\{i : a(\sigma_i) = 1\}|}{m}.$$

Aiming at the (very different) objective of secure protocols for search problems, Beimel et al. (2008) defined the notation of solution-list algorithms, which corresponds to our notation of deterministic representation. We next rephrase their results using our notations.

$R1$ For every $\alpha > 0$ and every $\Delta > 1$, there exists a set $\mathcal{B}$ that $(\alpha + 1/8)$-deterministically represents $\mathrm{OPT}_{\mathrm{E3SAT}}$ for databases of size $m = O\big(\Delta(\ln\ln(n) + \ln(1/\alpha))\big)$, and a ratio of Δ.

$R2$ Let $\alpha < 1/2$ and $\Delta > 1$. For every set $\mathcal{B}$ that α- deterministically represents $\mathrm{OPT}_{\mathrm{E3SAT}}$ for databases of size m with a ratio of Δ, it holds that $m = \Omega\big(\ln\ln(n)\big)$.

Using $(R1)$ and a deterministic version of Lemma 28, for every $\alpha, \beta, \epsilon > 0$, there exists an $\big((1/8 + \alpha), \beta, \epsilon\big)$- approximation algorithm for $\mathrm{OPT}_{\mathrm{E3SAT}}$ with a database of $m = O_{\alpha,\beta,\epsilon}(\ln\ln(n))$ clauses. By $(R2)$, this is the best possible using a deterministic representation.

We can reduce the necessary database size, using a probabilistic representation. Fix a clause with three different literals. If we pick an assignment at random, then with probability at least $7/8$ it satisfies the clause. Now, fix any exact 3CNF formula. If we pick an assignment at random, then the expected fraction of satisfied clauses is at least $7/8$. Moreover, for every $0 < \alpha < 7/8$, the fraction of satisfied clauses is at least $(7/8 - \alpha)$ with

probability at least $\frac{\alpha}{\alpha+1/8}$. So, if we pick $t = \frac{\ln(1/\beta)}{\ln(\alpha+1/8)+\ln(1/\alpha)}$ random assignments, the probability that none of them will satisfy at least $(7/8-\alpha)m$ clauses is at most $\left(\frac{\alpha}{\alpha+1/8}\right)^t = \beta$. So, for every $\Delta > 1$,

$$\mathscr{B} = \{\mathcal{B} : \mathcal{B} \text{ is a set of at most } t \text{ assignments}\},$$

and $\mathcal{P}$, the distribution induced on $\mathscr{B}$ by randomly picking t assignments, are an $\big((1/8+\alpha),\beta\big)$-probabilistic representation of $\mathrm{OPT}_{\mathrm{E3SAT}}$ for databases of size $\Delta \cdot \ln(t)$ and a ratio of Δ. By Lemma 28, for every ϵ there exists an $\big((1/8+\alpha),\beta,\epsilon\big)$-approximation algorithm for $\mathrm{OPT}_{\mathrm{E3SAT}}$ with a database of $m = O_{\alpha,\beta,\epsilon}(1)$ clauses.

6. Extensions

6.1. (ϵ,δ)-Differential Privacy

Recall the definition of (ϵ,δ)-differential privacy:

$$\Pr[A(S_1) \in \mathcal{F}] \le \exp(\epsilon) \cdot \Pr[A(S_2) \in \mathcal{F}] + \delta.$$

The proof of Lemma 16 remains valid even if algorithm A is only (ϵ,δ)-differential private for

$$\delta \le \tfrac{1}{8} e^{-8\alpha\epsilon m}(1-e^{-\epsilon}). \tag{5}$$

To see this, note that inequality (3) changes to

$$\begin{aligned}
&\Pr_A\left[A(\vec{0}) \in G\right] \ge \\
&\ge \left(\left(\left(\Pr_A[A(S) \in G] \cdot e^{-\epsilon} - \delta\right) e^{-\epsilon} - \delta\right) \cdots \right) e^{-\epsilon} - \delta \\
&\ge \frac{1}{4} e^{-8\alpha\epsilon m} - \delta \left(\sum_{i=0}^{8\alpha m - 1} e^{-i\epsilon}\right) \\
&\ge \frac{1}{4} e^{-8\alpha\epsilon m} - \delta \left(\frac{1}{1-e^{-\epsilon}}\right) \ge \frac{1}{8} e^{-8\alpha\epsilon m}.
\end{aligned}$$

The rest of the proof remains almost intact (only minor changes in the constants). With that in mind, we see that the lower bound showed in Theorem 19 for ϵ-differentially private (that is, with $\delta = 0$) learners also applies for (ϵ,δ)-differentially private learners satisfying inequality (5). That is, every such learner for a class $\mathcal{C}$ must use $\Omega\left(\frac{\mathrm{RepDim}(\mathcal{C})}{\alpha\epsilon}\right)$ samples.

When using (ϵ,δ)-differential privacy, it is desirable for δ to be negligible in the security parameter, e.g., in d – the representation length of elements in X_d. In such a case, using (ϵ,δ)-differential privacy instead of ϵ-differential privacy cannot reduce the sample complexity for PPAC learning a concept class $\mathcal{C}$ whenever $\mathrm{RepDim}(\mathcal{C}) = O(\log(d))$.

6.2. Probabilistic Representation Using a Hypothesis Class

We will now consider a generalization of our representation notations that can be useful when considering PPAC learners that use a specific hypothesis class. In particular, those notation can be useful when considering proper PPAC learners, that is, a learner that learns a class $\mathcal{C}$ using a hypothesis class $\mathcal{B} \subseteq \mathcal{C}$.

Definition 30 *We define the α-Deterministic Representation Dimension of a concept class $\mathcal{C}$ using a hypothesis class $\mathcal{B}$ as*

$$\mathrm{DRepDim}_\alpha(\mathcal{C},\mathcal{B}) = \min\left\{\mathrm{size}(\mathcal{H}) : \begin{array}{c}\mathcal{H} \subseteq \mathcal{B} \text{ is an} \\ \alpha\text{-representation} \\ \text{for class } \mathcal{C}\end{array}\right\}.$$

Note that $\mathrm{DRepDim}_{\frac{1}{4}}(\mathcal{C}, 2^{X_d}) = \mathrm{DRepDim}(\mathcal{C})$. The dependency on α in the above definition is necessary: if $\mathcal{C}$ is not contained in $\mathcal{B}$ then for every small enough α, the hypothesis class $\mathcal{B}$ itself does not α-represents $\mathcal{C}$ (and therefore no subset $\mathcal{H} \subseteq \mathcal{B}$ can α-represent $\mathcal{C}$). Moreover, when considering the notations of representation using a hypothesis class, our boosting technique for α does not work (as the boosting uses more complex hypotheses).

Example 9 *Beimel et al. (2014) showed that for every $\alpha < 1$, every subset $\mathcal{H} \subsetneq \mathtt{POINT}_d$ does not α-represent the class $\mathtt{POINT}_d$. Therefore, $\mathrm{DRepDim}_\alpha(\mathtt{POINT}_d, \mathtt{POINT}_d) = \theta(d)$ for every $\alpha < 1$.*

Definition 31 *A pair $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for a concept class C using a hypothesis class $\mathcal{B}$ if:*

1. *$(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for the class C, as formulated in Definition 10.*
2. *Every $\mathcal{H}_i \in \mathscr{H}$ is a subset of $\mathcal{B}$.*

Note that whenever $\mathcal{B} = 2^{X_d}$, this definition is identical to Definition 10. Using this general notation, we can restate Lemma 14 and Lemma 16 as follows:

Lemma 32 *If there exists a pair $(\mathscr{H},\mathcal{P})$ that (α,β)- probabilistically represents a class $\mathcal{C}$ using a hypothesis class $\mathcal{B}$, then for every ϵ and every γ there exists an algorithm A that $(\alpha+\gamma, 3\beta, \epsilon)$-PPAC learns $\mathcal{C}$ using $\mathcal{B}$ and a sample size $m = O((\mathrm{size}(\mathscr{H}) + \ln(\frac{1}{\beta})) \max\{\frac{1}{\gamma\epsilon}, \frac{1}{\gamma^2}\})$.*

Note that in the above lemma the resulting algorithm A has accuracy $(\alpha+\gamma)$ as opposed to 6α in lemma 14, where γ is arbitrary. While in section 3 we did not mind the multiplicative factor of 6 in the accuracy parameter (as we could boost it back), replacing it with an additive factor of γ might be of value in this section as our boosting technique for the accuracy parameter does not work here. As an example, consider a representation with $\alpha = \frac{1}{10}$. Without boosting capabilities, this change makes the difference between the ability to generate an algorithm with $\alpha = \frac{6}{10}$, or an algorithm with $\alpha = \frac{1}{10} + \frac{1}{1000}$.

Proof Let $(\mathscr{H}, \mathcal{P})$ be an (α, β)-probabilistic representation for class $\mathcal{C}$ using a hypothesis class $\mathcal{B}$, and consider the following algorithm A:

> Inputs: $S = (x_i, y_i)_{i=1}^m$, and a privacy parameter ϵ.
> 1. Randomly choose $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$.
> 2. Choose $h \in \mathcal{H}_i$ using the exp. mechanism with ϵ.

First note that the support of A is indeed (a subset of) $\mathcal{B}$. By the properties of the exponential mechanism, A is ϵ-differentially private. Fix some $c \in \mathcal{C}$ and $\mathcal{D}$, and define the following 3 good events:

E_1 $\mathcal{H}_i$ chosen in step 1 contains at least one hypothesis h s.t. $\text{error}_{\mathcal{D}}(h) \leq \alpha$.

E_2 For every $h \in \mathcal{H}_i$ it holds that $|\text{error}_S(h) - \text{error}_{\mathcal{D}}(c, h)| \leq \frac{\gamma}{3}$.

E_3 The exponential mechanism chooses an h such that $\text{error}_S(h) \leq \frac{\gamma}{3} + \min_{f \in \mathcal{H}_i} \{\text{error}_S(f)\}$.

Note that if those 3 good events happen, algorithm A returns an $(\alpha + \gamma)$-good hypothesis. We will now show that those 3 events happen with high probability.

As $(\mathscr{H}, \mathcal{P})$ is an (α, β)-probabilistic representation for the class $\mathcal{C}$, event E_1 happens with probability at least $1 - \beta$.

Using the Hoeffding bound, event E_2 happens with probability at leat $1 - 2|\mathcal{H}_i| \exp(-\frac{2}{9}\gamma^2 m)$. For $m \geq \frac{9}{2\gamma^2} \ln(\frac{2|\mathcal{H}_i|}{\beta})$, this probability is at leat $1 - \beta$.

The exponential mechanism ensures that the probability of event E_3 is at least $1 - |\mathcal{H}_i| \cdot \exp(-\epsilon\gamma m/6)$ (see Section 2.4), which is at least $(1 - \beta)$ for $m \geq \frac{6}{\gamma\epsilon} \ln(\frac{|\mathcal{H}_i|}{\beta})$.

All in all, by setting $m = 6(\text{size}(\mathscr{H}) + \ln(\frac{2}{\beta})) \max\{\frac{1}{\gamma^2}, \frac{1}{\gamma\epsilon}\}$ we ensure that the probability of A failing to output an $(\alpha + \gamma)$-good hypothesis is at most 3β. ■

Lemma 33 *If there exists an algorithm A that $(\alpha, \frac{1}{2}, \epsilon)$-PPAC learns a concept class $\mathcal{C}$ using a hypothesis class $\mathcal{B}$ and a sample size m, then there exists a pair $(\mathscr{H}, \mathcal{P})$ that $(\alpha, 1/4)$-probabilistically represents the class $\mathcal{C}$ using the hypothesis class $\mathcal{B}$ where $\text{size}(\mathscr{H}) = O(m\epsilon)$.*

The proof of Lemma 33 is identical to the proof of Lemma 15.

Definition 34 *We define the α-Probabilistic Representation Dimension of a concept class $\mathcal{C}$ using a hypothesis class $\mathcal{B}$ as*

$$\text{RepDim}_\alpha(\mathcal{C}, \mathcal{B}) = \min \left\{ \text{size}(\mathscr{H}) : \begin{array}{l} \exists \mathcal{P} \text{ s.t. } (\mathscr{H}, \mathcal{P}) \\ \text{is an } (\alpha, \frac{1}{4})\text{-prob.} \\ \text{representation} \\ \text{for } \mathcal{C} \text{ using } \mathcal{B} \end{array} \right\}.$$

Example 10 *Beimel et al. (2014) showed that for every $\alpha < 1$, every proper PPAC learner for $\texttt{POINT}_d$ requires $\Omega((d + \log(1/\beta))/(\epsilon\alpha))$ labled examples. Using Lemma 32, we get that $\text{RepDim}_\alpha(\texttt{POINT}_d, \texttt{POINT}_d) = \Omega(d)$.*

The above example shows a strong separation between the VC dimension of the class $\texttt{POINT}_d$ and $\text{RepDim}_\alpha(\texttt{POINT}_d, \texttt{POINT}_d)$.

7. A Probabilistic Representation for Points

Example 3 states the existence of a constant size probabilistic representation for the class $\texttt{POINT}_d$. We now give the construction. Similar ideas were used by Feldman (2009) who studied the smilingly unrelated *model of evolvability* of Valiant (2009) (a theoretical model for quantifying how complex mechanisms, such as those found in living cells, can evolve as result of a random search guided by selection).

Claim 35 *There exists an (α, β)-probabilistic representation for $\texttt{POINT}_d$ of size $\ln(4/\alpha) + \ln\ln(1/\beta)$. Furthermore, each hypothesis h in each $\mathcal{H}_i$ has a short description and given x, the value $h(x)$ can be computed efficiently.*

Proof Consider the following set of hypothesis classes

$$\mathscr{H} = \left\{ \mathcal{H} \subseteq 2^{X_d} \ : \ |\mathcal{H}| \leq \frac{4}{\alpha} \ln(\frac{1}{\beta}) \right\}.$$

That is, $\mathcal{H} \in \mathscr{H}$ if $\mathcal{H}$ contains at most $\frac{4}{\alpha} \ln(\frac{1}{\beta})$ boolean functions. We will show an appropriate distribution $\mathcal{P}$ s.t. $(\mathscr{H}, \mathcal{P})$ is an (α, β)-probabilistic representation of the class $\texttt{POINT}_d$. To this end, fix a target concept $c_j \in \texttt{POINT}_d$ and a distribution $\mathcal{D}$ on X_d (remember that j is the unique point on which $c_j(j) = 1$). We need to show how to randomly choose an $\mathcal{H} \in_R \mathscr{H}$ such that with probability at least $(1 - \beta)$ over the choice of $\mathcal{H}$, there will be at least one $h \in \mathcal{H}$ such that $\text{error}_{\mathcal{D}}(c_j, h) \leq \alpha$. Consider the following process for randomly choosing an $\mathcal{H} \in \mathscr{H}$:

> 1. Denote $M = \frac{4}{\alpha} \ln(\frac{1}{\beta})$
> 2. For $i = 1, \ldots, M$ construct hypothesis h_i as follows:
> For each $x \in X_d$ (independently):
> Let $h_i(x) = 1$ with probability $\alpha/2$,
> and $h_i(x) = 0$ otherwise.
> 3. Return $\mathcal{H} = \{h_1, h_2, \ldots, h_M\}$.

The above process induces a distribution on $\mathscr{H}$, denoted as $\mathcal{P}$. We will next analyze the probability that the returned $\mathcal{H}$ does not contain an α-good hypothesis. We start by fixing some i and analyzing the expected error of h_i, conditioned on the event that $h_i(j) = 1$. The probability is taken over the random coins used to construct h_i.

$$\begin{aligned} &\mathop{\mathbb{E}}_{h_i}\left[\text{error}_{\mathcal{D}}(c_j, h_i) \,\middle|\, h_i(j) = 1\right] = \\ &= \mathop{\mathbb{E}}_{h_i}\left[\mathop{\mathbb{E}}_{x \in \mathcal{D}}\left[\,|c_j(x) - h_i(x)|\,\right] \,\middle|\, h_i(j) = 1\right] \\ &= \mathop{\mathbb{E}}_{x \in \mathcal{D}}\left[\mathop{\mathbb{E}}_{h_i}\left[\,|c_j(x) - h_i(x)| \,\middle|\, h_i(j) = 1\right]\right] \leq \frac{\alpha}{2}. \end{aligned}$$

Using Markov's Inequality,

$$\Pr_{h_i}\left[\text{error}_{\mathcal{D}}(c_j, h_i) \geq \alpha \,\middle|\, h_i(j) = 1\right] \leq \frac{1}{2}.$$

So, the probability that h_i is α-good for c_j and $\mathcal{D}$ is:

$$\begin{aligned}
&\Pr_{h_i}\left[\mathrm{error}_{\mathcal{D}}(c_j,h_i)\le\alpha\right]\ge\\
&\ge\Pr_{h_i}\left[h_i(j)=1\right]\cdot\Pr_{h_i}\left[\mathrm{error}_{\mathcal{D}}(c_j,h_i)\le\alpha\;\middle|\;h_i(j)=1\right]\\
&\ge\frac{\alpha}{2}\cdot\frac{1}{2}=\frac{\alpha}{4}.
\end{aligned}$$

Thus, the probability that $\mathcal{H}$ fails to contain an α-good hypothesis is at most $\left(1-\frac{\alpha}{4}\right)^M$, which is less than β for our choice of M. This concludes the proof that $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for $\texttt{POINT}_d$.

When a hypothesis $h_i()$ was constructed in the above random process, the value of $h_i(x)$ was independently drawn for every $x\in X_d$. This results in a hypothesis whose description size is $O(2^d)$, which in turn, will result in a non efficient learning algorithm. We next construct hypotheses whose description is short. To achieve this goal, we note that in the above analysis we only care about the probability that $h_i(x)=0$ given that $h_i(j)=1$. Thus, we can choose the values of h_i in a pairwise independent way, e.g., using a random polynomial of degree 2. The size of the description in this case is $O(d)$. ■

Consider the class $\texttt{POINT}_{\mathbb{N}}$, defined in Example 6. The above construction can be adjusted to yield an (inefficient) improper private learner for $\texttt{POINT}_{\mathbb{N}}$ with $O_{\alpha,\beta,\epsilon}(1)$ samples. The only adjustments necessary are in the construction of the (α,β)-probabilistic representation. Specifically, we need to specify how to randomly draw a boolean function h over the natural numbers, such that for every $x\in\mathbb{N}$ the probability of $h(x)=1$ is $\alpha/2$, and the values of h on every two distinct points in $\mathbb{N}$ are independent. This can be done assuming that the learner is allowed to output a real number, as a random real number could be interpreted as a random function over $\mathbb{N}$. Note however, that this means that the learner outputs a hypothesis with infinite description. As was shown by Bun et al. (2015), this barrier is unavoidable, and every pure private (proper or improper) learner for $\texttt{POINT}_{\mathbb{N}}$ must output a hypothesis with infinite description.

Acknowledgments

Amos Beimel and Kobbi Nissim were supported by NSF grant no. 1565387. TWC: Large: Collaborative: Computing Over Distributed Sensitive Data. Amos Beimel was supported by ISF grant no. 152/17 and a grant from the cyber security center at Ben Gurion university. Uri Stemmer was supported by a gift from Google Ltd.

Appendix A.

In this section we prove Lemma 18.

Proof Let $\mathcal{C}$ be a concept class, and let $(\mathscr{H}^1,\mathcal{P}^1)$ be a $(\frac{1}{4},\beta/T)$-probabilistic representation for $\mathcal{C}$ (where T will be set later). By Claim 17, such a representation exists with $\mathrm{size}(\mathscr{H}^1)\triangleq k_1\le\mathrm{RepDim}(\mathcal{C})+\ln\ln(T/\beta)$. We use $\mathscr{H}^1$ and $\mathcal{P}^1$ to create an (α,β)- probabilistic representation for $\mathcal{C}$. We begin with two notations:

1. For T hypotheses $h_1,\ldots,h_T$ we denote by $\mathrm{maj}_{h_1,\ldots,h_T}$ the majority hypothesis. That is, $\mathrm{maj}_{h_1,\ldots,h_T}(x)=1$ if and only if $|\{h_i\,:\,h_i(x)=1\}|\ge T/2$.
2. For T hypothesis classes $\mathcal{H}_1,\ldots,\mathcal{H}_T$ we denote $\mathrm{MAJ}(\mathcal{H}_1,\ldots,\mathcal{H}_T)=\left\{\mathrm{maj}_{h_1,\ldots,h_T}\,:\,\forall_{1\le i\le T}\,h_i\in\mathcal{H}_i\right\}$.

Consider the following family of hypothesis classes:

$$\mathscr{H}=\left\{\mathrm{MAJ}(\mathcal{H}_{i_1},\ldots,\mathcal{H}_{i_T})\,:\,\mathcal{H}_{i_1},\ldots,\mathcal{H}_{i_T}\in\mathscr{H}^1\right\}.$$

Moreover, denote the distribution on $\mathscr{H}$ induced by the following random process as $\mathcal{P}$:

For $j=1,\ldots,T$:
Randomly choose $\mathcal{H}_{i_j}\in_{\mathcal{P}^1}\mathscr{H}^1$
Return $\mathrm{MAJ}(\mathcal{H}_{i_1},\ldots,\mathcal{H}_{i_T})$.

Next we show that $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for $\mathcal{C}$: For a fixed pair of a target concept c and a distribution $\mathcal{D}$, randomly choose $\mathcal{H}_{i_1},\ldots,\mathcal{H}_{i_T}\in_{\mathcal{P}^1}\mathscr{H}^1$. We now show that with probability at least $(1-\beta)$ the set $\mathrm{MAJ}(\mathcal{H}_{i_1},\ldots,\mathcal{H}_{i_T})$ contains at least one α-good hypothesis for $c,\mathcal{D}$.

To this end, denote $\mathcal{D}_1=\mathcal{D}$ and consider the following thought experiment, inspired by the Adaboost Algorithm of Freund and Schapire (1997):

For $t=1.\ldots,T$:
1. Fail if $\mathcal{H}_{i_t}$ does not contain a $\frac{1}{4}$-good hypothesis for $c,\mathcal{D}_t$.
2. Denote by $h_t\in\mathcal{H}_{i_t}$ a $\frac{1}{4}$-good hypothesis for $c,\mathcal{D}_t$.
3. $\mathcal{D}_{t+1}(x)=\begin{cases}2\mathcal{D}_t(x), & \text{if } h_t(x)\ne c(x).\\ \left(1-\frac{\mathrm{error}_{\mathcal{D}_t}(c,h_t)}{1-\mathrm{error}_{\mathcal{D}_t}(c,h_t)}\right)\mathcal{D}_t(x), & \text{otherwise.}\end{cases}$

Note that as $\mathcal{D}_1$ is a probability distribution on X_d; the same is true for $\mathcal{D}_2,\mathcal{D}_3,\ldots,\mathcal{D}_T$. As $(\mathscr{H}^1,\mathcal{P}^1)$ is a $(\frac{1}{4},\beta/T)$-probabilistic representation for $\mathcal{C}$, the failure probability of every iteration is at most β/T. Thus (using the union bound), with probability at least $(1-\beta)$ the whole thought experiment will succeed, and in this case we show that the error of $h_{\mathrm{fin}}=\mathrm{maj}_{h_1,\ldots,h_T}$ is at most α.

Consider the set $R=\{x\,:\,h_{\mathrm{fin}}(x)\ne c(x)\}\subseteq X_d$. This is the set of points on which at least $T/2$ of $h_1,\ldots,h_T$ err. Next consider the partition of R to the following sets:

$$R_t=\left\{x\in R\,:\,\left(h_t(x)\ne c(x)\right)\wedge\left(\forall_{i>t}\,h_i(x)=c(x)\right)\right\}.$$

That is, R_t contains the points $x \in R$ on which h_t is last to err. Clearly $\mathcal{D}_t(R_t) \leq 1/4$, as R_t is a subset of the set of points on which h_t errs. Moreover,

$$\begin{aligned}
\mathcal{D}_t(R_t) &\geq \mathcal{D}_1(R_t) \cdot 2^{T/2} \cdot \left(1 - \frac{\mathrm{error}_{\mathcal{D}_t}(c,h_t)}{1-\mathrm{error}_{\mathcal{D}_t}(c,h_t)}\right)^{t-T/2} \\
&\geq \mathcal{D}_1(R_t) \cdot 2^{T/2} \cdot \left(1 - \frac{1/4}{1-1/4}\right)^{t-T/2} \\
&\geq \mathcal{D}_1(R_t) \cdot 2^{T/2} \cdot \left(1 - \frac{1/4}{1-1/4}\right)^{T/2} \\
&= \mathcal{D}(R_t) \cdot \left(\frac{4}{3}\right)^{T/2},
\end{aligned}$$

so,

$$\mathcal{D}(R_t) \leq \mathcal{D}_t(R_t) \cdot \left(\frac{4}{3}\right)^{-T/2} \leq \frac{1}{4} \cdot \left(\frac{4}{3}\right)^{-T/2}.$$

Finally,

$$\begin{aligned}
\mathrm{error}_{\mathcal{D}}(c,h_{\mathrm{fin}}) = \mathcal{D}(R) = \sum_{t=T/2}^{T} \mathcal{D}(R_t) \leq \\
\leq \frac{T}{2} \cdot \frac{1}{4} \cdot \left(\frac{4}{3}\right)^{-T/2} = \frac{T}{8} \cdot \left(\frac{4}{3}\right)^{-T/2}.
\end{aligned}$$

Choosing $T = 14\ln(\frac{2}{\alpha})$, we get that $\mathrm{error}_{\mathcal{D}}(c,h_{\mathrm{fin}}) \leq \alpha$. Hence, $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for $\mathcal{C}$. Moreover, for every $\mathcal{H}_i \in \mathscr{H}$ we have that $|\mathcal{H}_i| \leq \left(e^{k_1}\right)^T$, and so

$$\begin{aligned}
&\mathrm{size}(\mathscr{H}) \leq k_1 \cdot T \leq \big(\mathrm{RepDim}(\mathcal{C}) + \ln\ln(T/\beta)\big)T \\
&= O\Big(\ln(\frac{1}{\alpha}) \cdot \big(\mathrm{RepDim}(\mathcal{C}) + \ln\ln\ln(\frac{1}{\alpha}) + \ln\ln(\frac{1}{\beta})\big)\Big).
\end{aligned}$$

■

Appendix B.

As we mentioned in the introduction (Section 1.3), Feldman and Xiao (2015) showed an equivalence between $\mathrm{RepDim}(\mathcal{C})$ and the randomized one-way communication complexity of the evaluation problem for concepts from $\mathcal{C}$ (in the public coin model). A similar argument shows that the private-coin model is equivalent to $\mathrm{DRepDim}(\mathcal{C})$. Thus, a relationship of $\mathrm{DRepDim}(\mathcal{C}) = O(\mathrm{RepDim}(\mathcal{C}) + \ln(d))$ follows from the classical result of Newman (1991) on the difference in communication complexity between the public and private coin models. In this section we present a direct proof for this relationship, essentially following the same strategy as the proof of Newman.

Observation 36 *Let $(\mathscr{H},\mathcal{P})$ be an (α,β)-probabilistic representation for a concept class $\mathcal{C}$. Then, $\mathcal{B} = \bigcup_{\mathcal{H}_i \in \mathscr{H}} \mathcal{H}_i$ is an α-representation of $\mathcal{C}$.*

Proof As $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for $\mathcal{C}$, for every c and every $\mathcal{D}$

$$\Pr_{\mathcal{P}}[\exists h \in \mathcal{H}_i \ \ s.t \ \ \mathrm{error}_{\mathcal{D}}(c,h) \leq \alpha] \geq 1-\beta > 0.$$

The probability is over choosing a set $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$. In particular, for every c and every $\mathcal{D}$ there exists an $\mathcal{H}_i \in \mathscr{H}$ that contains an α-good hypothesis. ■

The simple construction in Observation 36 may result in a very large deterministic representation. For example, in Claim 35 we show an $(\mathscr{H},\mathcal{P})$ that (α,β)- probabilistically represents the class $\mathtt{POINT}_d$, where $\mathscr{H}$ contains all the sets of at most $\frac{4}{\alpha}\ln(\frac{1}{\beta})$ boolean functions. While $\bigcup_{\mathcal{H}_i \in \mathscr{H}} \mathcal{H}_i = 2^{X_d}$ is indeed an α-representation for $\mathtt{POINT}_d$, it is extremely over-sized.

We will show that it is not necessary to take the union of all the $\mathcal{H}_i$'s in $\mathscr{H}$ in order to get an α-representation for $\mathcal{C}$. As $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation, for every c and every $\mathcal{D}$, with probability at least $1-\beta$ a randomly chosen $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$ contains an α-good hypothesis. The straight forward strategy here is to first boost β as in Claim 17, and then use the union bound over all possible $c \in \mathcal{C}$ and over all possible distributions $\mathcal{D}$ on X_d. Unfortunately, there are infinitely many such distributions, and the proof will be somewhat more complicated.

Definition 37 *Let $\mathscr{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_r\}$ be a family of hypothesis classes, and $\mathcal{P}$ be a distribution over $\{1,\ldots,r\}$. We will denote the following non private algorithm as Learner$(\mathscr{H},\mathcal{P},m,\gamma)$:*

> *Input: a sample $S = (x_i,y_i)_{i=1}^m$.*
> 1. *Randomly choose $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$.*
> 2. *If for every $h \in \mathcal{H}_i$ $\mathrm{error}_S(h) > \gamma$, then fail.*
> 3. *Return $h \in \mathcal{H}_i$ minimizing $\mathrm{error}_S(h)$.*

*We will say that Learner$(\mathscr{H},\mathcal{P},m,\gamma)$ is β-*successful *for a class $\mathcal{C}$ over X_d, if for every $c \in \mathcal{C}$ and every distribution $\mathcal{D}$ on X_d, given an input sample drawn i.i.d. according to $\mathcal{D}$ and labeled by c, algorithm Learner fails with probability at most β.*

Claim 38 *If $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for a class $\mathcal{C}$, then, for $m \geq \frac{3}{\alpha}\ln(1/\beta)$, algorithm Learner$(\mathscr{H},\mathcal{P},m,2\alpha)$ is 2β-successful for $\mathcal{C}$.*

Proof We will show that with probability at least $1-2\beta$, the set $\mathcal{H}_i$ (chosen in Step 1) contains at least one hypothesis h s.t. $\mathrm{error}_S(h) \leq 2\alpha$. As $(\mathscr{H},\mathcal{P})$ is an (α,β)-probabilistic representation for class $\mathcal{C}$, the chosen $\mathcal{H}_i$ will contain a hypothesis h s.t. $\mathrm{error}_{\mathcal{D}}(c,h) \leq \alpha$ with probability at least $1-\beta$; by the Chernoff bound with probability at least $1-\exp(-m\alpha/3)$ this hypothesis has empirical error at most 2α. The set $\mathcal{H}_i$ contains a hypothesis h s.t. $\mathrm{error}_S(h) \leq 2\alpha$ with probability at least $(1-\beta)(1-\exp(-m\alpha/3)) > 1-(\beta+\exp(-m\alpha/3))$, which is at least $(1-2\beta)$ for $m \geq \frac{3}{\alpha}\ln(1/\beta)$. ■

Claim 39 *Let $\mathscr{H}$ be a family of hypothesis classes, and $\mathcal{P}$ a distribution on it. Let γ, β and m be such that $m \geq \frac{4}{\gamma}(\text{size}(\mathscr{H}) + \ln(\frac{1}{\beta}))$. If $Learner(\mathscr{H}, \mathcal{P}, m, \gamma)$ is β-successful for a class $\mathcal{C}$ over X_d, then there exists $\widehat{\mathscr{H}} \subseteq \mathscr{H}$ and a distribution $\widehat{\mathcal{P}}$ on it, s.t. $Learner(\widehat{\mathscr{H}}, \widehat{\mathcal{P}}, m, \gamma)$ is a $(2\gamma, 3\beta)$-PAC learner for $\mathcal{C}$ and $\left|\widehat{\mathscr{H}}\right| = \frac{d \cdot m}{\beta^2}$.*

Proof For every input $S = (x_i, y_i)_{i=1}^m$, denote by p_S the probability of $Learner(\mathscr{H}, \mathcal{P}, m, \gamma)$ failing on step 2 (the probability is only over the choice of $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$ in the first step). As $Learner(\mathscr{H}, \mathcal{P}, m, \gamma)$ is β-successful,

$$\Pr_{\mathcal{P},\mathcal{D}}\left[Learner(\mathscr{H}, \mathcal{P}, m, \gamma) \text{ fails}\right] = \sum_S \Pr_{\mathcal{D}}[S] \cdot p_S \leq \beta.$$

Consider the following process, denoted by Proc, for randomly choosing a multiset $\widetilde{\mathscr{H}}$ of size t (t will be set later):

For $i = 1, \ldots, t$:
Randomly choose $\mathcal{H}_i \in_{\mathcal{P}} \mathscr{H}$
Return $\widetilde{\mathscr{H}} = (\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_t)$.

Denote by $\mathcal{U}_t$ the uniform distribution on $\{1, 2, \ldots, t\}$. As before, for every input $S = (x_i, y_i)_{i=1}^m$, denote by $\widetilde{p_S}$ the probability of $Learner(\widetilde{\mathscr{H}}, \mathcal{U}_t, m, \gamma)$ failing on its second step (again, the probability is only over the choice of $\mathcal{H}_i \in_{\mathcal{U}_t} \widetilde{\mathscr{H}}$ in the first step). Using those notations:

$$\Pr_{\mathcal{U}_t,\mathcal{D}}\left[Learner(\widetilde{\mathscr{H}}, \mathcal{U}_t, m, \gamma) \text{ fails}\right] = \sum_S \Pr_{\mathcal{D}}[S] \cdot \widetilde{p_S}.$$

Fix a sample S. As the choice of $\mathcal{H}_i \in_{\mathcal{U}_t} \widetilde{\mathscr{H}}$ is uniform,

$$\widetilde{p_S} = \frac{\left|\left\{\mathcal{H}_i \in \widetilde{\mathscr{H}} \ : \ \forall h \in \mathcal{H}_i \ \text{error}_S(h) > \gamma\right\}\right|}{\left|\widetilde{\mathscr{H}}\right|}.$$

Using the Hoeffding bound,

$$\Pr_{Proc}\left[|\widetilde{p_S} - p_S| \geq \beta\right] \leq 2e^{-2t\beta^2}.$$

The probability is over choosing the multiset $\widetilde{\mathscr{H}}$. There are at most $2^{m(d+1)}$ samples of size m (as every entry in the sample is an element of X_d, concatenated with a label bit). Using the union bound over all possible samples S,

$$\Pr_{Proc}\left[\exists S \ s.t. \ |\widetilde{p_S} - p_S| \geq \beta\right] \leq 2^{m(d+1)} \cdot 2 \cdot e^{-2t\beta^2}.$$

For $t \geq \frac{m \cdot d}{\beta^2}$ the above probability is strictly less than 1. This means that for $t = \frac{m \cdot d}{\beta^2}$ there exists a multiset $\widehat{\mathscr{H}}$ such that $|\widehat{p_S} - p_S| \leq \beta$ for every sample S. We will show that for this $\widehat{\mathscr{H}}$, $Learner(\widehat{\mathscr{H}}, \mathcal{U}_t, m, \gamma)$ is a $(2\gamma, 3\beta)$-PAC learner. Fix a target concept $c \in \mathcal{C}$ and a distribution $\mathcal{D}$ on X_d. Define the following two good events:

E_1 $Learner(\widehat{\mathscr{H}}, \mathcal{U}_t, m, \gamma)$ outputs a hypothesis h such that $\text{error}_S(h) \leq \gamma$.

E_2 For every $h \in \mathcal{H}_i$ s.t. $\text{error}_S(h) \leq \gamma$, it holds that $\text{error}_{\mathcal{D}}(c, h) \leq 2\gamma$.

Note that if those two events happen, $Learner(\widehat{\mathscr{H}}, \mathcal{U}_t, m, \gamma)$ returns a 2γ-good hypothesis for c and $\mathcal{D}$. We will show that those two events happen with high probability. We start by bounding the failure probability of $Learner(\widehat{\mathscr{H}}, \mathcal{U}_t, m, \gamma)$.

$$\begin{aligned}
&\Pr_{\mathcal{U}_t,\mathcal{D}}\left[Learner(\widehat{\mathscr{H}}, \mathcal{U}_t, m, \gamma) \text{ fails}\right] \\
&= \sum_S \Pr_{\mathcal{D}}[S] \cdot \widehat{p_S} \\
&\leq \sum_S \Pr_{\mathcal{D}}[S] \cdot (p_S + \beta) \\
&= \Pr_{\mathcal{P},\mathcal{D}}\left[Learner(\mathscr{H}, \mathcal{P}, m, \gamma) \text{ fails}\right] + \beta \leq 2\beta.
\end{aligned}$$

When $Learner(\widehat{\mathscr{H}}, \mathcal{U}_t, m, \gamma)$ does not fail, it returns a hypothesis h with empirical error at most γ. Thus, $\Pr[E_1] \geq 1 - 2\beta$.

Using the Chernoff bound, the probability that a hypothesis h with $\text{error}_{\mathcal{D}}(c, h) > 2\gamma$ has empirical error $\leq \gamma$ is less than $\exp(-m\gamma/4)$. Using the union bound, the probability that there is such a hypothesis in $\mathcal{H}_i$ is at most $|\mathcal{H}_i| \cdot \exp(-m\gamma/4)$. Therefore, $\Pr[E_2] \geq 1 - |\mathcal{H}_i| \cdot \exp(-m\gamma/4)$. For $m \geq \frac{4}{\gamma}\ln(\frac{|\mathcal{H}_i|}{\beta})$, this probability is at least $(1 - \beta)$.

All in all, the probability of $Learner(\mathscr{H}, \mathcal{P}, m, \gamma)$ failing to output a 2γ-good hypothesis is at most 3β. ■

Theorem 40 *If there exists a pair $(\mathscr{H}, \mathcal{P})$ that (α, β)-probabilistically represents a class $\mathcal{C}$ over X_d (where $|\mathscr{H}|$ might be very big), then there exists a pair $(\widehat{\mathscr{H}}, \widehat{\mathcal{P}})$ that $(4\alpha, 6\beta)$-probabilistically represents $\mathcal{C}$, where $\widehat{\mathscr{H}} \subseteq \mathscr{H}$, and*

$$\left|\widehat{\mathscr{H}}\right| = \frac{3d}{4\alpha\beta^2}\left(\text{size}(\mathscr{H}) + \ln(\frac{1}{\beta})\right).$$

Proof Let $(\mathscr{H}, \mathcal{P})$ be an (α, β)-probabilistic representation for a class $\mathcal{C}$. Set $m = \frac{3}{\alpha}(\text{size}(\mathscr{H}) + \ln(\frac{1}{\beta}))$. By Claim 38, $Learner(\mathscr{H}, \mathcal{P}, m, 2\alpha)$ is 2β-successful for class $\mathcal{C}$. By Claim 39, there exists an $\widehat{\mathscr{H}} \subseteq \mathscr{H}$ and a distribution $\widehat{\mathcal{P}}$ on it, such that $Learner(\widehat{\mathscr{H}}, \widehat{\mathcal{P}}, m, 2\alpha)$ is a $(4\alpha, 6\beta)$-PAC learner for $\mathcal{C}$ and $\left|\widehat{\mathscr{H}}\right| = \frac{d \cdot m}{4\beta^2} = \frac{3d}{4\alpha\beta^2}(\text{size}(\mathscr{H}) + \ln(\frac{1}{\beta}))$.

Assume towards contradiction that $(\widehat{\mathscr{H}}, \widehat{\mathcal{P}})$ does not $(4\alpha, 6\beta)$-represent $\mathcal{C}$. So, there exist a concept $c \in \mathcal{C}$ and a distribution $\mathcal{D}$ s.t., with probability strictly greater than 6β, a randomly chosen $\mathcal{H}_i \in_{\widehat{\mathcal{P}}} \widehat{\mathscr{H}}$ does not contain a 4α-good hypothesis for $c, \mathcal{D}$. Therefore, for those c and $\mathcal{D}$, $Learner(\widehat{\mathscr{H}}, \widehat{\mathcal{P}}, m, 2\alpha)$ will fail to return a 4α-good hypothesis with probability strictly greater than 6β. ■

Theorem 41 *For every class $\mathcal{C}$ over X_d there exists a $\frac{1}{4}$-representation $\mathcal{B}$ such that* $\mathrm{size}(\mathcal{B}) = O(\ln(d) + \mathrm{RepDim}(\mathcal{C}))$.

Proof By Lemma 18, there exists a pair $(\mathscr{H}, \mathcal{P})$ that $(\frac{1}{16}, \frac{1}{12})$-probabilistically represents $\mathcal{C}$ such that $\mathrm{size}(\mathscr{H}) = O(\mathrm{RepDim}(\mathcal{C}))$. Using Theorem 40, there exists a pair $(\widehat{\mathscr{H}}, \widehat{\mathcal{P}})$ that $(\frac{1}{4}, \frac{1}{2})$-probabilistically represents $\mathcal{C}$, such that $\mathrm{size}(\widehat{\mathscr{H}}) = \mathrm{size}(\mathscr{H})$ and

$$\left|\widehat{\mathscr{H}}\right| = O\left(d \cdot \mathrm{size}(\mathscr{H})\right).$$

We can now use Observation 36 and construct the set $\mathcal{B} = \bigcup_{\mathcal{H}_i \in \widehat{\mathscr{H}}} \mathcal{H}_i$ which is a $\frac{1}{4}$-representation for the class $\mathcal{C}$. In addition,

$$|\mathcal{B}| = O\left(\left|\widehat{\mathscr{H}}\right| \cdot e^{\mathrm{size}(\mathscr{H})}\right) = O\left(d \cdot \mathrm{size}(\mathscr{H}) \cdot e^{\mathrm{size}(\mathscr{H})}\right).$$

Thus, $\mathrm{size}(\mathcal{B}) = \ln|\mathcal{B}| = O\left(\ln(d) + \mathrm{RepDim}(\mathcal{C})\right)$. ■

Corollary 42 *For every concept class $\mathcal{C}$ over X_d,* $\mathrm{DRepDim}(\mathcal{C}) = O(\ln(d) + \mathrm{RepDim}(\mathcal{C}))$.

Corollary 43 *There exists a constant N s.t. for every concept class C over X_d where* $\mathrm{DRepDim}(\mathcal{C}) \geq N \log(d)$*, the sample complexity that is necessary and sufficient for privately learning $\mathcal{C}$ is* $\Theta_{\alpha,\beta}(\mathrm{DRepDim}(\mathcal{C}))$.

References

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. *CoRR*, abs/1806.00949, 2018. URL `http://arxiv.org/abs/1806.00949`.

Amos Beimel, Paz Carmi, Kobbi Nissim, and Enav Weinreb. Private approximation of search problems. *SIAM J. Comput.*, 38(5):1728–1760, 2008.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, volume 8096 of *Lecture Notes in Computer Science*, pages 363–378. Springer, 2013.

Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. In *SODA*, pages 461–477. SIAM, 2015.

Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. *CoRR*, abs/1902.10731, 2019. URL `http://arxiv.org/abs/1902.10731`.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618. ACM, 2008.

Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, October 1989. ISSN 0004-5411. doi: 10.1145/76359.76371. URL `http://doi.acm.org/10.1145/76359.76371`.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *FOCS*, pages 634–649, 2015.

Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In Sham M. Kakade and Ulrike von Luxburg, editors, *COLT*, volume 19 of *JMLR Proceedings*, pages 155–186. JMLR.org, 2011.

Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Statist.*, 23:493–507, 1952.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014. doi: 10.1561/0400000042. URL `http://dx.doi.org/10.1561/0400000042`.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60. IEEE Computer Society, 2010.

Andrzej Ehrenfeucht, David Haussler, Michael J. Kearns, and Leslie G. Valiant. A general lower bound on the number of examples needed for learning. *Inf. Comput.*, 82(3):247–261, 1989.

Vitaly Feldman. Robustness of evolvability. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009. URL `http://www.cs.mcgill.ca/%7Ecolt2009/papers/028.pdf#page=1`.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. *SIAM J. Comput.*, 44(6):1740–1764, 2015. doi: 10.1137/140991844. URL `http://dx.doi.org/10.1137/140991844`.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987. doi: 10.1007/BF00116827. URL https://doi.org/10.1007/BF00116827.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE Computer Society, 2007.

Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998. doi: 10.1006/jcss.1998.1577. URL https://doi.org/10.1006/jcss.1998.1577.

Ilan Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67 – 71, 1991. ISSN 0020-0190. doi: https://doi.org/10.1016/0020-0190(91)90157-D. URL http://www.sciencedirect.com/science/article/pii/002001909190157D.

Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5(2):197–227, 1990.

John W. Tukey. Mathematics and the picturing of data. In *Proc. Int. Congress of Mathematicians*, volume 2, pages 523–532, 1975.

Salil Vadhan. *The Complexity of Differential Privacy*. 2016.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL http://doi.acm.org/10.1145/1968.1972.

Leslie G. Valiant. Evolvability. *J. ACM*, 56(1):3:1–3:21, 2009. doi: 10.1145/1462153.1462156. URL https://doi.org/10.1145/1462153.1462156.

Vladimir N. Vapnik and Alexey Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16 (2):264–280, 1971.

Journal of Machine Learning Research 20 (2019) 1–50 Submitted 6/18; Revised 4/19; Published 10/19

Robustifying Independent Component Analysis by Adjusting for Group-Wise Stationary Noise

Niklas Pfister* PFISTER@STAT.MATH.ETHZ.CH
Seminar for Statistics, ETH Zürich
Rämisstrasse 101, 8092 Zürich, Switzerland

Sebastian Weichwald* SWEICHWALD@MATH.KU.DK
Department of Mathematical Sciences, University of Copenhagen
Universitetsparken 5, 2100 Copenhagen, Denmark

Peter Bühlmann BUHLMANN@STAT.MATH.ETHZ.CH
Seminar for Statistics, ETH Zürich
Rämisstrasse 101, 8092 Zürich, Switzerland

Bernhard Schölkopf BS@TUE.MPG.DE
Empirical Inference Department, Max Planck Institute for Intelligent Systems
Max-Planck-Ring 4, 72076 Tübingen, Germany

Editor: Kenji Fukumizu

Abstract

We introduce coroICA, confounding-robust independent component analysis, a novel ICA algorithm which decomposes linearly mixed multivariate observations into independent components that are corrupted (and rendered dependent) by hidden group-wise stationary confounding. It extends the ordinary ICA model in a theoretically sound and explicit way to incorporate group-wise (or environment-wise) confounding. We show that our proposed general noise model allows to perform ICA in settings where other noisy ICA procedures fail. Additionally, it can be used for applications with grouped data by adjusting for different stationary noise within each group. Our proposed noise model has a natural relation to causality and we explain how it can be applied in the context of causal inference. In addition to our theoretical framework, we provide an efficient estimation procedure and prove identifiability of the unmixing matrix under mild assumptions. Finally, we illustrate the performance and robustness of our method on simulated data, provide audible and visual examples, and demonstrate the applicability to real-world scenarios by experiments on publicly available Antarctic ice core data as well as two EEG data sets. We provide a scikit-learn compatible pip-installable Python package coroICA as well as R and Matlab implementations accompanied by a documentation at https://sweichwald.de/coroICA/.

Keywords: blind source separation, causal inference, confounding noise, group analysis, heterogeneous data, independent component analysis, non-stationary signal, robustness

1. Introduction

The analysis of multivariate data is often complicated by high dimensionality and complex inter-dependences between the observed variables. In order to identify patterns in such data it is therefore desirable and often necessary to separate different aspects of the data. In multivariate statistics, for example, principal component analysis (PCA) is a common preprocessing step that decomposes the data into orthogonal principle components which are sorted according to how much variance of the original data each component explains. There are two important applications of this. Firstly, one can reduce the dimensionality of the data by projecting it onto the lower dimensional space spanned by the leading principal components which maximize the explained variance. Secondly, since the principle components are orthogonal, they separate in some sense different (uncorrelated) aspects of the data. In many situations this enables a better interpretation and representation.

Often, however, PCA may not be sufficient to separate the data in a desirable way due to more complex inter-dependences in the multivariate data (see e.g., Section 1.3.3 in Hyvärinen et al. (2002) for an instructive example). This observation motivates the development of independent component analysis (ICA), formally introduced in its current form by Cardoso (1989b) and Comon (1994). ICA is a widely used unsupervised blind source separation technique that aims at decomposing an observed mixture of independent source signals. More precisely, assuming that the observed data is a linear mixture of underlying independent variables, one seeks the unmixing matrix that maximizes the independence between the signals it extracts. There has been a large amount of research on different types of ICA procedures and their interpretations, e.g., Bell and Sejnowski (1995, Infomax) who maximize the entropy, Hyvärinen (1999, fastICA) maximizing the kurtosis or Belouchrani et al. (1997, SOBI) who propose to minimize time-lagged dependences, to name only some of the widespread examples.

ICA has applications in many fields, for example in finance (e.g., Back and Weigend, 1997), the study of functional magnetic resonance imaging (fMRI) data (e.g., McKeown et al., 1998a,b; Calhoun et al., 2003), and notably in the analysis of electroencephalography (EEG) data (e.g., Makeig et al., 1995, 1997; Delorme and Makeig, 2004). The latter is motivated by the common assumption that the signals recorded at EEG electrodes are a (linear) superposition of cortical dipole signals (Nunez and Srinivasan, 2006). Indeed, ICA-based preprocessing has become the de facto standard for the analysis of EEG data. The extracted components are interpreted as corresponding to cortical sources (e.g., Ghahremani et al., 1996; Zhukov et al., 2000; Makeig et al., 2002) or used for artifact removal by dropping components that are dominated by ocular or muscular activity (e.g., Jung et al., 2000; Delorme et al., 2007).

In many applications, the data at hand is heterogeneous and parts of the samples can be grouped by the different settings (or environments) under which the observations were taken. For example, we can group those samples of a multi-subject EEG recording that belong to the same subject. For the analysis and interpretation of such data across different groups, it is desirable to extract one set of common features or signals instead of obtaining individual ICA decompositions for each group of samples separately. Here, we present a novel, methodologically sound framework that extends the ordinary ICA model, respects the group structure and is robust by explicitly accounting for group-wise stationary confounding. More precisely, we consider a model of the form

$$X_i = A \cdot S_i + H_i, \tag{1}$$

*Authors contributed equally. Most of this work was done while SW was at the Max Planck Institute for Intelligent Systems, Tübingen, Germany.

©2019 Niklas Pfister*, Sebastian Weichwald*, Peter Bühlmann, and Bernhard Schölkopf.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-399.html.

where i denotes the sample index, A remains fixed across different groups, S_i is a vector of independent source signals and H_i is a vector of stationary confounding noise variables with fixed covariance within each group (an intuitive example where such a scenario may be encountered in practice is illustrated in Figure 7). Based on this extension to ordinary ICA, we construct a method and an easy to implement algorithm to extract one common set of sources that are robust against confounding within each group and can be used for across-group analyses. The unmixing also generalizes to previously unseen groups.

1.1. Relation to Existing Work

ICA is well-studied with a tremendous amount of research related to various types of extensions and relaxations of the ordinary ICA model. In light of this, it is important to understand where our proposed procedure is positioned and why it is an interesting and useful extension. Here, we look at ICA research from three perspectives and illustrate how our proposed coroICA methodology relates to existing work. First off, in Section 1.1.1 we compare our proposed methodology with other noisy ICA models. In Section 1.1.2, we review ICA procedures based on approximate joint matrix diagonalization. Finally, in Section 1.1.3 we summarize the existing literature on ICA procedures for grouped data and highlight the differences to coroICA.

1.1.1. Noisy ICA Models

The ordinary ICA model assumes that the observed process X is a linear mixture of independent source signals S *without* a confounding term H. Identifiability of the source signals S is guaranteed by assumptions on S such as non-Gaussianity or specific time structures. For coroICA we require—similar to other second-order based methods (cf. Section 1.1.2)—that the source process S is non-stationary. More precisely, we require that either the variance or the auto-covariance of S changes across time. An important extension of the ordinary ICA model is known as noisy ICA (e.g., Moulines et al., 1997) in which the data generating process is assumed to be an ordinary ICA model with additional additive noise. In general, this leads to further identifiability issues. These can be resolved by assuming that the additive noise is Gaussian and the signal sources non-Gaussian (e.g., Hyvärinen, 1999), which enables correct identification of the mixing matrix. Another possibility is to assume that the noise is independent over time, while the source signals are time-dependent[1] (e.g., Choi and Cichocki, 2000b). In contrast, our assumption on the noise term H is much weaker, since we only require it to be stationary and hence in particular allow for time-dependent noise in coroICA. As we show in our simulations in Section 4.2.3 this renders our method robust with respect to confounding noise: coroICA is more robust against time-dependent noise while remaining competitive in the setting of time-independent noise. We refer to the book by Hyvärinen et al. (2002) for a review of most of the existing ICA models and the assumptions required for identifiability.

1. Autocorrelated signals are time-dependent, while the absence of autocorrelation does not necessarily imply time-independence of the signal. We thus use the terms time-dependence and time-independence throughout this article.

1.1.2. ICA based on Approximate Joint Diagonalization

As an extension of PCA, the concept of ICA is naturally connected to the notion of joint diagonalization of covariance-type matrices. One of the first procedures for ICA was FOBI introduced by Cardoso (1989a), which aims to jointly diagonalize the covariance matrix and a fourth order cumulant matrix. Extending on this idea Cardoso and Souloumiac (1993) introduced the method JADE which improves on FOBI by diagonalizing several different fourth order cumulant matrices. Unlike FOBI, JADE uses a general joint matrix diagonalization algorithm which is the de facto standard for all modern approaches. In fact, there is a still-active field that focuses on approximate joint matrix diagonalization, commonly restricted to positive semi-definite matrices, and often with the purpose of improving ICA procedures (e.g., Cardoso and Souloumiac, 1996; Ziehe et al., 2004; Tichavsky and Yeredor, 2009; Ablin et al., 2018).

Both JADE and FOBI are based on the assumption that the signals are non-Gaussian. This ensures that the sources are identifiable given independent and identically distributed observations. A different stream of ICA research departs from this assumption and instead assumes that the data are a linear mixture of independent weakly stationary time-series. This model is often referred to as a second-order source-separation model (SOS). The time structure in these models allows to identify the sources by jointly diagonalizing the covariance and auto-covariance. The first method developed for this setting is AMUSE by Tong et al. (1990) who diagonalize the covariance matrix and the auto-covariance matrix for one fixed lag. The performance of AMUSE is, however, fragile with respect to the exact choice of the lag, which complicates practical application (Miettinen et al., 2012). Instead of only using a single lag, Belouchrani et al. (1997) proposed the method SOBI which uses all lags up to a certain order and jointly diagonalizes all the resulting auto-covariance matrices. SOBI is to date still one of the most commonly employed ICA methods, in particular in EEG analysis.

The SOS model is based on the assumption of weak stationarity of the sources which implies that the signals have fixed variance and auto-covariance structure across time. This assumption can be dropped and the resulting models are often termed non-stationary source separation models (NSS). The non-stationarity can be leveraged to boost the performance of ICA methods in various ways (see Matsuoka et al., 1995; Hyvärinen, 2001; Choi and Cichocki, 2000a,b; Choi et al., 2001; Choi and Cichocki, 2001; Pham and Cardoso, 2001). All aforementioned methods make use of the non-stationarity by jointly diagonalizing different sets of covariance or auto-covariance matrices and mainly differ by how they perform the approximate joint matrix diagonalization. For example, the methods introduced by Choi and Cichocki (2000a,b); Choi et al. (2001) make use of non-stationarity across sources by separating the data into blocks and jointly diagonalizing either the covariance matrices, the auto-covariances or both across all blocks. For our experimental comparisons, we implemented all three of these methods with the slight modification that we use the recent uwedge approximate joint matrix diagonalization procedure due to Tichavsky and Yeredor (2009). We denote the resulting three ICA variants as

- choiICA (var): jointly diagonalize blocks of covariances,
- choiICA (TD): jointly diagonalize blocks of auto-covariances,
- choiICA (var & TD): jointly diagonalize blocks of covariances and auto-covariances.

method	signal type	allowed noise
choiICA (TD)	varying time-dependence	time-independent
choiICA (var)	varying variance	none
choiICA (var & TD)	varying variance and time-dependence	none
SOBI	fixed time-dependence	time-independent
fastICA[2]	non-Gaussian	none
coroICA	varying time-dependence *and/or* variance	group-wise stationary

Table 1. Important ICA procedures and the signal types they require as well as the noise they can deal with. coroICA is a confounding-robust ICA variant and is the only method for which an identifiability result under time-dependent noise is available.

Depending on the type of matrix which is diagonalized, each procedure detects different types of signals and behaves differently with respect to noise. Choi and Cichocki (2001) suggest a modification of choiICA (TD) in which instead of auto-covariance matrices, differences of auto-correlation matrices are diagonalized. The advantage being that it captures the non-stationarity of a signal more explicitly. Our proposed method similarly aims to use this type of signal but instead of considering the noise-free case, we explicitly formalize a model class that generalizes to noisy settings. Furthermore, we provide an identifiability theorem allowing for group-wise stationary confounding. Such a result has not been proven for the aforementioned method in the noise-free case. For a detailed description of both SOS- and NSS-based methods we refer the reader to the review by Nordhausen (2014) and for recent developments on leveraging non-stationarity for identifiability in non-linear ICA see Hyvärinen and Morioka (2016).

An exhaustive comparison of all methods is infeasible on the one hand due to the sheer amount of different models and methods and on the other hand due to the fact that appropriately maintained and easy adaptable code—for most methods—simply does not exist. Therefore, we focus our comparison on the following representative, modern methods that are most closely related to coroICA: fastICA, SOBI, choiICA (TD), choiICA (var), choiICA (TD & var). The methods and their respective assumptions on the source and noise characteristics are summarized in Table 1.

1.1.3. ICA Procedures for Grouped Data

Applications in EEG and fMRI have motivated the development of a wide variety of blind source separation techniques which are capable of dealing with grouped data, e.g., where groups correspond to different subjects or recording sessions. A short review is given in Hyvärinen (2013) and a detailed exposition in the context of fMRI data is due to Calhoun et al. (2003).

Consider we are given m groups $\{g_1, \dots, g_m\}$ and observe a corresponding data matrix $\mathbf{X}_{g_i} \in \mathbb{R}^{d \times n_i}$ for each group, where d is the number of observed signals and n_i the number of observations. Using this notation, all existing ICA procedures for grouped data can be related to one of three underlying models extending the classical mixing model $\mathbf{X} = A \cdot \mathbf{S}$. The first, often also referred to as "temporal concatenation", assumes that the mixing

2. The fastICA method can be extended to include Gaussian noise (see Hyvärinen, 1999).

remains equal while the sources are allowed to change across groups leading to data of the form

$$(\mathbf{X}_{g_1}, \dots, \mathbf{X}_{g_m}) = A \cdot (\mathbf{S}_{g_1}, \dots, \mathbf{S}_{g_m}) \,. \tag{2}$$

The second model, often also referred to as "spatial concatenation", assumes the sources remain fixed ($n_1 = \cdots = n_m$) while the mixing matrices are allowed to change, i.e.,

$$\begin{pmatrix} \mathbf{X}_{g_1} \\ \vdots \\ \mathbf{X}_{g_m} \end{pmatrix} = \begin{pmatrix} A_{g_1} \\ \vdots \\ A_{g_m} \end{pmatrix} \cdot \mathbf{S}. \tag{3}$$

Finally, the third model assumes that both the sources and the mixing remains fixed across groups which implies that for all $k \in \{1, \dots, m\}$ it holds that

$$\mathbf{X}_{g_k} = A \cdot \mathbf{S}. \tag{4}$$

In all three settings the baseline approach to ICA is to simply apply a classical ICA to the corresponding concatenated or averaged data, i.e., to apply the algorithm to the temporally/spatially concatenated data matrices on the left-hand side of above equations or the average over groups. These ad-hoc approaches are appealing, since they postulate straightforward procedures to solving the problem on grouped data and facilitate interpretability of the resulting estimates. It is these ad-hoc approaches that are implemented as the default behavior in toolboxes like the widely used eeglab for EEG analyses (Delorme and Makeig, 2004).

Several procedures have been proposed tailored to specific applications that extend on these baselines by employing additional assumptions. The most prominent such extensions are tensorial methods that have found popularity in fMRI analysis. They express the group index as an additional dimension (the data is thus viewed as a $\mathbb{R}^{d \times n \times m}$ tensor) and construct an estimate factorization of the tensor representation. Many of these procedures build on the so called PARAFAC (parallel factor analysis) model (Harshman, 1970). Recasting the tensor notation, this model is of the form (3) with $A_{g_k} = A \cdot D_{g_k}$ for all groups and for diagonal matrices $D_{g_1}, \dots, D_{g_m}$. As can be seen from this representation, the PARAFAC model allows the mixing matrices to change across groups while they are constrained to be the same up to different scaling of the mixing matrix columns (intuitively, across groups the source dimensions are allowed to project with different strengths onto the observed signal dimensions). Given that the matrices $D_{g_1}, \dots, D_{g_m}$ are sufficiently different it is possible to estimate this model uniquely without further assumptions. However, in the case that some of these diagonal matrices are equal identifiability is lost. In such cases Beckmann and Smith (2005) suggest to additionally require that the individual components of the sources be independent. This is comparable to the case where uncorrelatedness may not be sufficient for the separation of sources while independence is.

The coroICA procedure also allows for grouped-data but aims at inferring a fixed mixing matrix A, i.e., a model as given in (2) is considered. In contrast to vanilla concatenation procedures, our methodology naturally incorporates changes across groups by allowing and adjusting for different stationary confounding noise in each group. We argue why this leads to a more robust procedure and also illustrate this in our simulations and real data

experiments. More generally, our goal is to learn an unmixing which allows to generalize to new and previously unseen groups; think for example about learning an unmixing based on several different training subjects and extending it to new so far unseen subjects. Such tasks can appear in brain-computer interfacing applications and can also be of relevance more broadly in feature learning for classification tasks where classification models are to be transferred from one group/domain to another. Since our aim is to learn a fixed mixing matrix A that is confounding-robust and readily applicable to new groups, coroICA cannot naturally be compared to models that are based on spatial concatenation (3) or fixed sources *and* mixings (4); these methods employ fundamentally different assumptions on the model underlying the data generating process, the crucial difference being that we allow the sources and their time courses to change between groups.

1.2. Our Contribution

One strength of our methodology is that it explicates a statistical model that is sensible for data with group structure and can be estimated efficiently, while being supported by provable identification results. Furthermore, providing an explicit model with all required assumptions enables a constructive discussion about the appropriateness of such modeling decisions in specific application scenarios. The model itself is based on a notion of invariance against confounding structures from groups, an idea that is also related to invariance principles in causality (Haavelmo, 1944; Peters et al., 2016); see also Section 3 for a discussion on the relation to causality.

We believe that coroICA is a valuable contribution to the ICA literature on the following grounds:

- We introduce a methodologically sound framework which extends ordinary ICA to settings with grouped data and confounding noise.
- We prove identifiability of the unmixing matrix under mild assumptions, importantly, we explicitly allow for time-dependent noise thereby lessening the assumptions required by existing noisy ICA methods.
- We provide an easy to implement estimation procedure.
- We illustrate the usefulness, robustness, applicability, and limitations of our newly introduced coroICA algorithm as well as characterize the advantage of coroICA over existing ICAs: The source separation by coroICA is more stable across groups since it explicitly accounts for group-wise stationary confounding.
- We provide an open-source scikit-learn compatible ready-to-use Python implementation available as coroICA from the Python Package Index repository as well as R and Matlab implementations and an intuitive audible example which is available at `https://sweichwald.de/coroICA/`.

2. Methodology

We consider a general noisy ICA model inspired by ideas employed in causality research (see Section 3). We argue below that it allows to incorporate group structure and enables joint inference on multi-group data in a natural way. For the model description, let $S_i = (S_i^1, \dots, S_i^d)^\top \in \mathbb{R}^{d\times 1}$ and $H_i = (H_i^1, \dots, H_i^d)^\top \in \mathbb{R}^{d\times 1}$ be two independent vector-valued sequences of random variables where $i \in \{1, \dots, n\}$. The components $S_i^1, \dots, S_i^d$ are

assumed to be mutually independent for each i while, importantly, we allow for any weakly stationary noise H. Let $A \in \mathbb{R}^{d\times d}$ be an invertible matrix. The d-dimensional data process $(X_i)_{i\in\{1,\dots,n\}}$ is generated by the following noisy linear mixing model

$$X_i = A \cdot S_i + H_i, \quad \text{for all } i \in \{1, \dots, n\}. \tag{5}$$

X is a linear combination of source signals S and confounding variables H. In this model, both S and H are unobserved. One aims at recovering the mixing matrix A as well as true source signals S from observations of X. Without additional assumptions, the confounding H makes it impossible to identify the mixing matrix A. Even with additional assumptions it remains a difficult task (see Section 1.1.1 for an overview of related ICA models). Given the mixing matrix A it is straightforward to recover the confounded source signals $\widetilde{S}_i = S_i + A^{-1} \cdot H_i$.

Throughout this paper, we denote by $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{R}^{d\times n}$ the observed data matrix and similarly by $\mathbf{S}$ and $\mathbf{H}$ the corresponding (unobserved) source and confounding data matrices. For a finite data sample generated by this model we hence have

$$\mathbf{X} = A \cdot \mathbf{S} + \mathbf{H}.$$

In order to distinguish between the confounding H and the source signals S we assume that the two processes are sufficiently different. This can be achieved by assuming the existence of a group structure such that the covariance of the confounding H remains stationary within a group and only changes across groups.

Assumption 1 (group-wise stationary confounding) *There exists a collection of m disjoint groups $\mathcal{G} = \{g_1, \dots, g_m\}$ with $g_k \subseteq \{1, \dots, n\}$ and $\cup_{k=1}^m g_k = \{1, \dots, n\}$ such that for all $g \in \mathcal{G}$ the process $(H_i)_{i\in g}$ is weakly stationary.*

Under this assumption and given that the source signals change enough within groups, the mixing matrix A is identifiable (see Section 2.2). Similar to existing ICA methods discussed in Section 1.1.2, we propose to estimate the mixing matrix A by jointly diagonalizing empirical estimates of dependence matrices. In contrast to existing methods, we explicitly allow and adjust for the confounding H. The process of finding a matrix V that simultaneously diagonalizes a set of matrices is known as joint matrix diagonalization and has been studied extensively (e.g., Ziehe et al., 2004; Tichavsky and Yeredor, 2009). In Section 2.3, we show how to construct an estimator for V based on approximate joint matrix diagonalization.

The key step in adjusting for the confounding is to make use of the assumption that in contrast to the signals S the confounding H remains stationary within groups. Depending on the type of signal in the sources one can consider different sets of matrices. Here, we distinguish between two types of signals.

Variance signal In case of a variance signal, the variance process of each signal source $\mathrm{Var}(S_i^j)$ changes over time. These changes can be detected by examining the covariance matrix $\mathrm{Cov}(X_i)$ over time. For $V = A^{-1}$ and using (5) it holds for all $i \in \{1, \dots, n\}$ that

$$V\,\mathrm{Cov}(X_i)V^\top = \mathrm{Cov}(S_i) + V\,\mathrm{Cov}(H_i)V^\top.$$

Since the source signal components S_i^j are mutually independent, the covariance matrix $\mathrm{Cov}(S_i)$ is diagonal. Moreover, due to Assumption 1 the covariance matrix of the confounding H is constant, though not necessarily diagonal, within each group. This implies for all groups $g \in \mathcal{G}$ and for all $k, l \in g$ that

$$V\left(\mathrm{Cov}(X_k) - \mathrm{Cov}(X_l)\right)V^\top = \mathrm{Cov}(S_k) - \mathrm{Cov}(S_l) \tag{6}$$

is a diagonal matrix.

Time-dependence signal In case of a time-dependence signal, the time-dependence of each signal source S_i^j changes over time, i.e., for fixed τ, $\mathrm{Cov}(S_i^j, S_{i-\tau}^j)$ changes over time. These changes lead to changes in the auto-covariance matrices $\mathrm{Cov}(X_i, X_{i-\tau})$. Analogous to the variance signal it holds for all $i \in \{\tau+1, \ldots, n\}$ that

$$V\,\mathrm{Cov}(X_i, X_{i-\tau})V^\top = \mathrm{Cov}(S_i, S_{i-\tau}) + V\,\mathrm{Cov}(H_i, H_{i-\tau})V^\top.$$

Since the source signal components S_i^j are mutually independent, the auto-covariance matrix $\mathrm{Cov}(S_i, S_{i-\tau})$ is diagonal and due the stationarity of H (see Assumption 1) the auto-covariance $\mathrm{Cov}(H_i, H_{i-\tau})$ is constant within each group. This implies for all groups $g \in \mathcal{G}$, for all $k, l \in g$ and for all τ that

$$V\left(\mathrm{Cov}(X_k, X_{k-\tau}) - \mathrm{Cov}(X_l, X_{l-\tau})\right)V^\top = \mathrm{Cov}(S_k, S_{k-\tau}) - \mathrm{Cov}(S_l, S_{l-\tau}) \tag{7}$$

is a diagonal matrix.

For both signal types, we can identify V by simultaneously diagonalizing differences of (auto-)covariance matrices. Details and identifiability results are given in Section 2.3. The two signal types considered differ from both, the more classical settings of non-Gaussian time-independent signals as considered for example by fastICA, and the stationary signals with fixed time-dependence assumed for SOBI (cf. Table 1). Owing to the non-stationarity of the signal we can allow for more general forms of noise.

2.1. Motivating Examples

To get a better understanding of our proposed ICA model in (5), we illustrate two different aspects: the group structure and the noise model.

Noise model coroICA can be viewed as a noisy ICA, where the noise is allowed to be group-wise non-stationary. This generalizes existing noisy ICA methods, which, to the best of our knowledge, all assume that the noise is independent over time with various additional assumptions. The following example illustrates the intuition behind our model via a toy-application to natural images.

Example 1 (unmixing noisy images) *We provide an illustration of how our proposed method compares to other ICA approaches under the presence of noise. Four images, each* 450×300 *pixels and with three RGB color channels, are used to construct four sources* S^1, S^2, S^3, S^4 *as follows.*[3] *Every color channel is converted to a one dimensional vector by cutting each image into* 15×10 *equally sized patches (i.e., each patch consists of* 30×30 *pixels) and concatenating the row-wise vectorized patches. This procedure preserves the local structure of the image. We concatenate the three color channels and consider them as separate groups for our model. Thus, each of the four sources* $S^1, \ldots, S^4$ *consists of* $n = 3 \cdot 450 \cdot 300 = 405.000$ *observations, that is, three groups of* 135.000 *observations corresponding to the RGB color channels. Next, we construct locally dependent noise that differs across color channels. Here, locally dependent means that the added noise is similar (and dependent) for pixels which are close to each other. This results in four noise processes* $H^1, \ldots, H^4$. *We combine the sources with the noise and apply a random mixing matrix* A *to obtain the following observed data*

$$X = A \cdot S + H.$$

The recast noisy images $\widetilde{S} = S + A^{-1}H$ *are illustrated in the first row and the recast observed mixtures* X *in the second row of Figure 1. The last three rows are the resulting reconstructions of three different ICA procedures, coroICA (TD), fastICA and choiICA (TD). As expected, fastICA as a noise-free ICA method, appears frail to the noise in the images. While choiICA (TD) is able to adjust for independent noise, it is unable to properly adjust for the spatial dependence of the noise process and thus leads to undesired reconstruction results. In contrast, coroICA (TD) is able to recover the noisy images. It is the noise and its characteristics that break the two competing ICA methods, since all three methods are able to unmix the images in the noise-free case (not shown here).*

3. The images are freely available from Pexels GmbH (2018).

The noise model we employ is motivated by recent advances in causality research where the group-wise stationary noise can be interpreted as unobserved confounding factors in linear causal feedback models. We describe this in more detail with an explicit example application to Antarctic ice core data in Section 3.

Group structure A key aspect of our model is that it aims to leverage group-structure to improve the stability of the umixing under the presence of group-wise confounding. Here we refer to the following notion of stability: A stable unmixing matrix extracts the same set of independent sources when applied to the different groups; it is robust against the confounding that varies across groups and introduces dependences. A standard ICA method is not able to estimate the correct unmixing $V = A^{-1}$, if the data generating process follows our confounded ICA model in (5). These methods extract signals that are not only corrupted by the group-wise confounding but also are mixtures of the independent sources and are thus not stable in the aforementioned sense. This is illustrated by the "America's Got Talent Duet Problem" (cf. Example 2), an extension and alteration of the classical "cocktail party problem".

Example 2 (America's Got Talent Duet Problem) *Consider the problem of evaluating two singers at a duet audition individually. This requires to listen to the two voices separately, while the singers perform simultaneously. There are two sound sources in the audition room (the two singers) and additionally several noise sources which corrupt the recordings at the two microphones (or the jury member's two ears). A schematic of such a setting is illustrated in Supplement B, Figure 12. The additional noise comes from an audience and two open windows. One can assume that this noise satisfies our Assumption 1*

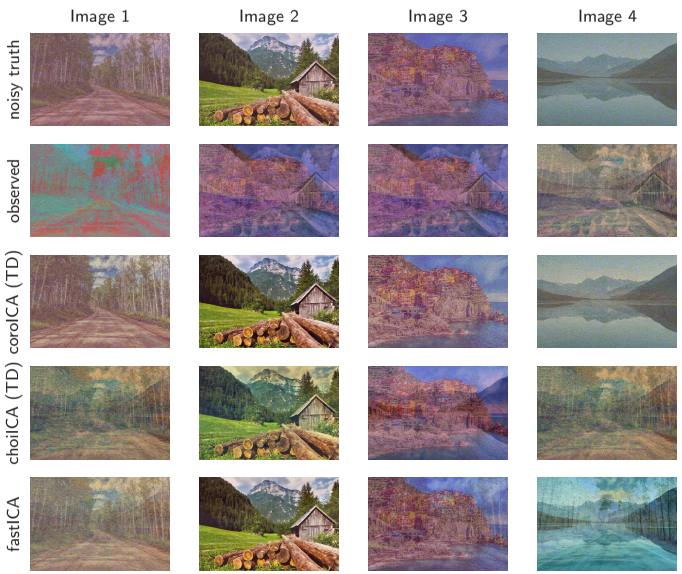


Figure 1. Images accompanying example 1. The top row shows noisy unmixed images, the second row shows mixed images, and the last three rows show unmixed and rescaled images resulting from an application of coroICA (TD), choiICA (TD) and fastICA (cf. Table 1). Here, only coroICA (TD) is able to correctly unmix the images and recover the original (noise-corrupted) images.

on a single group. The sound stemming from the audience can be seen as an average of many sounds, hence remaining approximately stationary over time. Typical sounds from an open window also satisfy this assumption, for example sound from a river or a busy road. Our methodology, however, also allows for more complicated settings in which the noise shifts at known points in times, for example if someone opens or closes a window or starts mowing the lawn outside. In such cases we use the known time blocks of stationary noise as groups and apply coroICA (var) on this grouped data. An example with artificial sound data related to this setting is available at `https://sweichwald.de/coroICA/`. *We show that coroICA (var) is able to recover useful sound signals with the two voices being separated into different dimensions and thus allows to listen to them individually. In contrast, existing ICAs applied to the time concatenated data fail to unmix the two singers.*

2.2. Identifiability

Identifiability requires that the source signals S change sufficiently strong within groups. The precise notion of a strong signal depends on the type of signal. As discussed previously,

we consider two types of non-stationary signals (i) variance signals and (ii) time-dependence signals. Depending on the signal type we formalize two slightly different assumptions that characterize source signals that ensure identifiability. Firstly, in the case of a variance signal, we have the following assumption.

Assumption 2 (signals with independently changing variance) *For each pair of components $p, q \in \{1, \dots, d\}$ we require the existence of three (not necessarily unique) groups $g_1, g_2, g_3 \in \mathcal{G}$ and three corresponding pairs $l_1, k_1 \in g_1$, $l_2, k_2 \in g_2$ and $l_3, k_3 \in g_3$ such that the two vectors*

$$\begin{pmatrix} \mathrm{Var}\left(S^p_{l_1}\right) - \mathrm{Var}\left(S^p_{k_1}\right) \\ \mathrm{Var}\left(S^p_{l_2}\right) - \mathrm{Var}\left(S^p_{k_2}\right) \\ \mathrm{Var}\left(S^p_{l_3}\right) - \mathrm{Var}\left(S^p_{k_3}\right) \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} \mathrm{Var}\left(S^q_{l_1}\right) - \mathrm{Var}\left(S^q_{k_1}\right) \\ \mathrm{Var}\left(S^q_{l_2}\right) - \mathrm{Var}\left(S^q_{k_2}\right) \\ \mathrm{Var}\left(S^q_{l_3}\right) - \mathrm{Var}\left(S^q_{k_3}\right) \end{pmatrix}$$

are neither collinear nor equal to zero.

In case of time-dependence signals we have the analogous assumption.

Assumption 3 (signals with independently changing time-dependence) *For each pair of components $p, q \in \{1, \dots, d\}$ we require the existence of three (not necessarily unique) groups $g_1, g_2, g_3 \in \mathcal{G}$ and three corresponding pairs $l_1, k_1 \in g_1$, $l_2, k_2 \in g_2$ and $l_3, k_3 \in g_3$ for which there exists $\tau \in \{1, \dots, n\}$ such that the two vectors*

$$\begin{pmatrix} \mathrm{Cov}\left(S^p_{l_1}, S^p_{l_1-\tau}\right) - \mathrm{Cov}\left(S^p_{k_1}, S^p_{k_1-\tau}\right) \\ \mathrm{Cov}\left(S^p_{l_2}, S^p_{l_2-\tau}\right) - \mathrm{Cov}\left(S^p_{k_2}, S^p_{k_2-\tau}\right) \\ \mathrm{Cov}\left(S^p_{l_3}, S^p_{l_3-\tau}\right) - \mathrm{Cov}\left(S^p_{k_3}, S^p_{k_3-\tau}\right) \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} \mathrm{Cov}\left(S^q_{l_1}, S^q_{l_1-\tau}\right) - \mathrm{Cov}\left(S^q_{k_1}, S^q_{k_1-\tau}\right) \\ \mathrm{Cov}\left(S^q_{l_2}, S^q_{l_2-\tau}\right) - \mathrm{Cov}\left(S^q_{k_2}, S^q_{k_2-\tau}\right) \\ \mathrm{Cov}\left(S^q_{l_3}, S^q_{l_3-\tau}\right) - \mathrm{Cov}\left(S^q_{k_3}, S^q_{k_3-\tau}\right) \end{pmatrix}$$

are neither collinear nor equal to zero.

Intuitively, these assumptions ensure that the signals are not changing in exact synchrony across components, which removes degenerate types of signals. In particular, they are satisfied in the case that the variance or auto-covariance processes change pair-wise independently over time. Whenever one of these assumptions is satisfied, the mixing matrix A is uniquely identifiable.

Theorem 1 (identifiability of the mixing matrix)
Assume the data process $(X_i)_{i \in \{1,\dots,n\}}$ satisfies the model in (5) and Assumption 1 holds. If additionally either Assumption 2 or Assumption 3 is satisfied, then A is unique up to permutation and rescaling of its columns.

Proof A proof is given in Supplement A. ■

2.3. Estimation

In order to estimate V from a finite observed sample $\mathbf{X} \in \mathbb{R}^{d \times n}$, we first partition each group into subgroups. We then compute the empirical (auto-)covariance matrices on each

coroICA

subgroup. Finally, we estimate a matrix that simultaneously diagonalizes the differences of these empirical (auto-)covariance matrices using an approximate joint matrix diagonalization technique. This procedure results in three methods depending on which type of matrices we diagonalize. Similar to our notation for the different versions of choiICAs we denote these methods by coroICA (var) if we diagonalize differences of covariances, coroICA (TD) if we diagonalize differences of auto-covariances, and coroICA (var & TD) if we diagonalize both differences of covariance and auto-covariances.

More precisely, for each group $g \in \mathcal{G}$, we first construct a partition $\mathcal{P}_g$ consisting of subsets of g such that each $e \in \mathcal{P}_g$ satisfies that $e \subseteq g$ and $\cup_{e \in \mathcal{P}_g} e = g$. This partition $\mathcal{P}_g$ should be granular enough to capture the changes in the signals described in Assumption 2 or 3. We propose partitioning each group based on a grid such that the separation between grid points is large enough for a reasonable estimation of the covariance matrix and at the same time small enough to capture variations in the signals. In our experiments, we observed robustness with respect to the exact choice; only too small partitions should be avoided since otherwise the procedure is fragile due to poorly estimated covariance matrices. More details on the choice of the partition size are given in Remark 2. Depending on whether a variance or time-dependence signal or a hybrid thereof is considered, we fix time lags $\mathrm{T} \subset \mathbb{N}_0$.

Next, for each group $g \in \mathcal{G}$, each distinct pair $e, f \in \mathcal{P}_g$, and each $\tau \in \mathrm{T}$ we define the matrix

$$M_{e,f}^{g,\tau} := \widehat{\mathrm{Cov}}_\tau(\mathbf{X}_e) - \widehat{\mathrm{Cov}}_\tau(\mathbf{X}_f),$$

where $\widehat{\mathrm{Cov}}_\tau(\cdot)$ denotes the empirical (auto-)covariance matrix for lag τ and $\mathbf{X}_e$ is the data matrix restricted to the columns corresponding to the subgroup e. Assumption 1 ensures that $V M_{e,f}^{g,\tau} V^\top$ is approximately diagonal. We are therefore interested in finding an invertible matrix V which approximately jointly diagonalizes the matrices in the set

$$\mathcal{M}^{\mathrm{all}} := \left\{ M_{e,f}^{g,\tau} \,\middle|\, g \in \mathcal{G} \text{ and } e, f \in \mathcal{P}_g \text{ and } \tau \in \mathrm{T} \right\}. \tag{8}$$

The number of matrices in this set grows quadratically in the number of partitions. This can lead to large numbers of matrices to be diagonalized. Another option that reduces the computational load is to compare each partition to its complement, which leads to the following set of matrices

$$\mathcal{M}^{\mathrm{comp}} := \left\{ M_{e,\bar{e}}^{g,\tau} \,\middle|\, g \in \mathcal{G} \text{ and } e \in \mathcal{P}_g \text{ (with } \bar{e} := g \setminus e) \text{ and } \tau \in \mathrm{T} \right\} \tag{9}$$

or to compare only neighboring partitions as in

$$\mathcal{M}^{\mathrm{neighbor}} := \left\{ M_{e,\mathrm{neighbor}(e)}^{g,\tau} \,\middle|\, g \in \mathcal{G} \text{ and } e \in \mathcal{P}_g \text{ and } \tau \in \mathrm{T} \right\}, \tag{10}$$

where $\mathrm{neighbor}(e)$ is the partition to the right of e.

The task of jointly diagonalizing a set of matrices is a well-studied topic in the literature and is referred to as approximate joint matrix diagonalization. Many solutions have been proposed for different assumptions made on the matrices to be diagonalized. In this paper, we use the uwedge algorithm[4] introduced by Tichavsky and Yeredor (2009). The basic idea

4. As a byproduct of our work, we are able to provide a new stable open-source Python/R/Matlab implementation of the uwedge algorithm which is also included in our respective coroICA packages.

13 JMLR 20(147):1-50, 2019

Pfister*, Weichwald*, Bühlmann, and Schölkopf

behind uwedge is to find a minimizer of a proxy for the loss function

$$\ell(V) = \sum_{M \in \mathcal{M}^*} \left(\sum_{k \neq l} \left[V M V^\top \right]_{k,l}^2 \right),$$

over the set of invertible matrices, where in our case $\mathcal{M}^* \in \{\mathcal{M}^{\mathrm{all}}, \mathcal{M}^{\mathrm{comp}}, \mathcal{M}^{\mathrm{neighboring}}\}$.

The full estimation procedure based on the set $\mathcal{M}^{\mathrm{neighbouring}}$ defined in (9) is made explicit in the pseudo code in Algorithm 1 (where ApproximateJointDiagonalizer stands for a general approximate joint diagonalizer; here we use uwedge).

Remark 2 (choosing the partition and the lags) *Whenever there is no obvious partition of the data, we propose to partition the data into equally sized blocks with a fixed partition size. The decision on how to choose a partition size should be driven by type of non-stationary signal one expects and the dimensionality of the data. For example, in the case of a variance signal the partition should be fine enough to capture areas of high and low variance, while at the same time being coarse enough to allow for sufficiently good estimates of the covariance matrices. That said, for applications to real data sets the signals are often of various length implying that there is a whole range of partition sizes which all work well. In cases with few data points, it can then be useful to consider several grids with different partition sizes and diagonalize across all resulting differences simultaneously. This somewhat removes the dependence of the results on the exact choice of a partition size and increases the power of the procedure. We employ this approach in Section 3.1. In general, the lags* T *should be chosen as* $\mathrm{T} = \{0\}$, $\mathrm{T} \subset \mathbb{N}$, *or* $\mathrm{T} \subset \mathbb{N}_0$, *depending on whether a variance signal, time-dependence signal, or a hybrid thereof is considered. For time-dependence signal, we recommend to determine up to which time-lag the autocorrelation of the observed signals has sufficiently decayed, and use all lags up to that point.*

2.4. Assessing the Quality of Recovered Sources

Assessing the quality of the recovered sources in an ICA setting is an inherently difficult task, as is typical for unsupervised learning procedures. The unidentifiable scale and ordering of the sources as well as the unclear choice of a performance measure render this task difficult. Provided that ground truth is known, several scores have been proposed, most notably the Amari measure introduced by Amari et al. (1995) and the minimum distance (MD) index due to Ilmonen et al. (2010). Here, we use the MD index, which is defined as

$$\mathrm{MD}(\hat{V}, A) = \frac{1}{\sqrt{p-1}} \inf_{C \in \mathcal{C}} \| C \hat{V} A - \mathrm{Id} \|,$$

where the set $\mathcal{C}$ consists of matrices for which each row and column has exactly one nonzero element. Intuitively, this score measures how close $\hat{V} A$ is to a rescaled and permuted version of the identity matrix. One appealing property of this score is that it can be computed efficiently by solving a linear sum assignment problem. In contrast to the Amari measure, the MD index is affine invariant and has desirable theoretical properties (see Ilmonen et al., 2010).

14 JMLR 20(147):1-50, 2019

Algorithm 1: coroICA

input : data matrix $\mathbf{X}$
group index $\mathcal{G}$ (user selected)
group-wise partition $(\mathcal{P}_g)_{g\in\mathcal{G}}$ (user selected)
lags $\mathrm{T} \subset \mathbb{N}_0$ (user selected)
initialize empty list $\mathcal{M}$
for $g \in \mathcal{G}$ **do**
  for $e \in \mathcal{P}_g$ **do**
    for $\tau \in \mathrm{T}$ **do**
      append $\widehat{\mathrm{Cov}}_\tau(\mathbf{X}_e) - \widehat{\mathrm{Cov}}_\tau(\mathbf{X}_{\mathrm{neighbour}(e)})$ to list $\mathcal{M}$
    end
  end
end
$\widehat{V} \leftarrow$ ApproximateJointDiagonalizer($\mathcal{M}$)
$\widehat{\mathbf{S}} \leftarrow \widehat{V}\mathbf{X}$
output: unmixing matrix $\widehat{V}$
sources $\widehat{\mathbf{S}}$

We require a different performance measure for our real data experiments where the true unmixing matrix is unknown. Here, we check whether the desired independence (after adjustment for the constant confounding) is achieved by computing the following covariance instability score (CIS) matrix. It measures the instability of the covariance structure of the unmixed sources $\widehat{\mathbf{S}}$ and is defined for a each groups $g \in \mathcal{G}$ and a corresponding partition $\mathcal{P}_g$ (see Section 2.3) by

$$\mathrm{CIS}(\widehat{\mathbf{S}}, \mathcal{P}_g) := \frac{1}{|\mathcal{P}_g|} \sum_{e\in\mathcal{P}_g} \left(\frac{\widehat{\mathrm{Cov}}(\widehat{\mathbf{S}}_e) - \widehat{\mathrm{Cov}}(\widehat{\mathbf{S}}_{\mathrm{neighbour}(e)})}{\widehat{\sigma}_{\widehat{\mathbf{S}}_g} \cdot \widehat{\sigma}_{\widehat{\mathbf{S}}_g}^\top} \right)^2,$$

where $\widehat{\sigma}_{\widehat{\mathbf{S}}} \in \mathbb{R}^{d\times 1}$ is the empirical standard deviation of $\widehat{\mathbf{S}}$ and the fraction is taken element-wise. The CIS matrix is approximately diagonal whenever $\widehat{\mathbf{S}}$ can be written as the sum of independent source signals $\mathbf{S}$ and confounding $\mathbf{H}$ with fixed covariance. This is condensed into one scalar that reflects how stable the sources' covariance structure is by averaging the off-diagonals of the CIS matrix

$$\mathrm{MCIS}(\widehat{\mathbf{S}}, \mathcal{P}_g)^2 := \frac{1}{d(d-1)} \sum_{\substack{i,j=1\\ i\neq j}}^{d} \left[\mathrm{CIS}(\widehat{\mathbf{S}}, \mathcal{P}_g)\right]_{i,j}.$$

The differences taken in the CIS score extract the variance signals such that the mean covariance instability score (MCIS) can be understood as a measure of independence between the recovered variance signal processes. High values of MCIS imply strong dependences beyond stationary confounding between the signals. Low values imply weak dependences. MCIS is a reasonable score whenever there is a *variance signal* (as described in Section 2)

in sources and is a sensible evaluation metric of ICA procedures in such cases. In case of time-dependence signal (as described in Section 2), one can define an analogous score based on the auto-covariances. Here, we restrict ourselves to the variance signal case as for all our applications this appeared to constitute the dominant part of the signal.

In case of *variance signals* the MCIS appears natural and appropriate as independence measure: It measures how well the individual variance signals (and hence the relevant information) are separated. To get a better intuition, let $A = (a_1, \ldots, a_d) \in \mathbb{R}^{d\times d}$ denote the mixing and $V = (v_1, \ldots, v_d)^\top \in \mathbb{R}^{d\times d}$ the corresponding unmixing matrix (i.e., $V = A^{-1}$, a_i are columns of A and v_i are rows of V). Then it holds that,

$$\begin{aligned}\mathrm{Cov}(X_i)v_j^\top &= A\,\mathrm{Cov}(S_i)A^\top v_j^\top + \mathrm{Cov}(H_i)v_j^\top \\ &= A\,\mathrm{Cov}(S_i)e_j^\top + A\,\mathrm{Cov}(H_i)e_j^\top \\ &= a_j\,\mathrm{Var}(S_i^j) + A\,\mathrm{Cov}(H_i)e_j^\top \end{aligned} \tag{11}$$

Under our group-wise stationary confounding assumption (Assumption 1) this implies that within all groups $g \in \mathcal{G}$, it holds for all $l, k \in g$ that

$$\left(\mathrm{Cov}(X_l) - \mathrm{Cov}(X_k)\right) v_j^\top = a_j \left(\mathrm{Var}(S_l^j) - \mathrm{Var}(S_k^j)\right). \tag{12}$$

This equation holds also in the confounding-free case and it reflects the contribution of the signal (in terms of variance signal) of the j-th recovered source S^j to the the variance signal in all components of the observed multivariate data X.

While in the population case the equality in (12) is satisfied exactly, this is no longer the case when the (un-)mixing matrix is estimated on finite data. Consider two subsets $e, f \in g$ for some group $g \in \mathcal{G}$, then using the notation from Section 2.3 and denoting by $\widehat{v}_j$ and $\widehat{a}_j$ the estimates of v_j and a_j, respectively, it holds that

$$\begin{aligned} M_{e,f}^g \widehat{v}_j^\top &= \left[\widehat{\mathrm{Cov}}(\mathbf{X}_e) - \widehat{\mathrm{Cov}}(\mathbf{X}_f)\right]\widehat{v}_j^\top \\ &= \widehat{A}\left[\widehat{\mathrm{Cov}}(\widehat{S}_e) - \widehat{\mathrm{Cov}}(\widehat{S}_f)\right]\widehat{A}^\top \widehat{v}_j^\top \\ &= \widehat{A}\left[\widehat{\mathrm{Cov}}(\widehat{S}_e) - \widehat{\mathrm{Cov}}(\widehat{S}_f)\right]e_j^\top \\ &\approx \widehat{a}_j(\mathrm{Var}(S_e^j) - \mathrm{Var}(S_f^j)). \end{aligned} \tag{13}$$

The approximation is close only if the empirical estimate $\widehat{V}$ correctly unmixes the j-th source. Essentially, MCIS measures the extent to which this approximation holds true for all components simultaneously across the subsets specified by the partition $\mathcal{P}_g$. It is also possible to consider individual components by assessing how closely the following proportionality is satisfied

$$\sum_{M\in\mathcal{M}^*} \mathrm{sign}(\widehat{v}_j M \widehat{v}_j^\top) M \widehat{v}_j^\top \propto \widehat{a}_j. \tag{14}$$

In EEG experiments, this can also be assessed visually by comparing the topographic maps corresponding to columns of A with so-called activation maps corresponding to the left-hand side in (14). More details on this are provided in Section 4.3.3.

3. Causal Perspective

Our underlying noisy ICA model (5) and the assumption on the noise (Assumption 1) are motivated by causal structure learning scenarios. ICA is closely linked to the problem of identifying structural causal models (SCMs) (see Pearl, 2009; Imbens and Rubin, 2015; Peters et al., 2017). Shimizu et al. (2006) were the first to make this connection explicit and used ICA to infer causal structures. To make this more precise consider the following linear SCM

$$X_i = B \cdot X_i + \widetilde{S}_i, \tag{15}$$

where X_i are observed covariates and $\widetilde{S}_i$ are noise terms. An SCM induces a corresponding causal graph over the involved variables by drawing an edge from variables on the right-hand side to the one on the left-hand side of (15). Moreover, we can define noise interventions (Pearl, 2009) by allowing the distributions of the noise terms $\widetilde{S}_i$ to change for different i. In the language of ICA, this means that the signals $\widetilde{S}_i$ encode the different interventions (over time) on the noise variables. Assuming that the matrix $\mathrm{Id} - B$ is invertible, we can rewrite (15) as

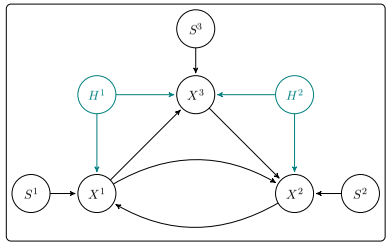
$$X_i = (\mathrm{Id} - B)^{-1} \widetilde{S}_i,$$

which can be viewed as an ICA model with mixing matrix $A = (\mathrm{Id} - B)^{-1}$. Instead of taking the noise term $\widetilde{S}_i$ as independent noise sources one can also consider $\widetilde{S}_i = S_i + H_i$. In that case the linear SCM in (15) describes a causal model between the observed variables X_i in which hidden confounding is allowed. This is illustrated in Figure 2, which depicts a 3 variable SCM with feedback loops and confounding. Learning a causal model as in (15) with ICA is generally done by performing the following two steps.

(i) **(ICA)** The matrix $(\mathrm{Id} - B)$ is inferred by ICA up to an undefined scale and permutation of its rows by using an appropriate ICA procedure. This step is often infeasible in the presence of confounding H since existing ICA methods only allow noise under restrictive assumptions (cf. Table 1).

(ii) **(identify B)** There are essentially two assumptions that one can make in order for this to work. The first is to assume the underlying causal model has an acyclic structure as in Shimizu et al. (2006). In such cases the matrix B needs to be permuted to an upper triangular matrix. The second option is to allow for feedback loops in the causal model but restrict the types of feedback to exclude infinite loops as in Hoyer et al. (2008) and Rothenhäusler et al. (2015).

When performing step (i) there are two important modeling assumptions that are made when selecting the ICA procedure: (a) the type of allowed signals (types of interventions) and (b) the type of allowed confounding. For the classic ICA setting with non-Gaussian source signals and no noise this translates to the class of linear non-Gaussian models, such as Linear Non-Gaussian Acyclic Models (LiNGAMs) introduced by Shimizu et al. (2006). While such models are a sensible choice in a purely observational setting (i.e., no samples from interventional settings) they are somewhat misspecified in terms of (a) when data from different interventional settings or time-continuous intervention shifts are observed (see Remark 3). In those settings, it is more natural to use ICA methods that are tailored to sequential shifts as for example choiICA or coroICA. Moreover, most common ICA methods



$$\begin{aligned} X^1 &\leftarrow b_{1,2} X^2 && + S^1 + H^1 \\ X^2 &\leftarrow b_{2,1} X^1 + b_{2,3} X^3 && + S^2 + H^2 \\ X^3 &\leftarrow b_{3,1} X^1 && + S^3 + H^1 + H^2 \end{aligned}$$

Figure 2. Illustration of an SCM with (including colored nodes H^1, H^2) and without (excluding colored nodes) confounding.

consider noise-free mixing, which from a causal perspective implies that no hidden confounding is allowed. While noisy ICA weakens this assumption, existing methods only allow for time-independent or even iid noise, which again greatly restricts the type of confounding. In contrast, our proposed coroICA allows for any type of block-wise stationary confounding, hence greatly increasing the class of causal models which can be inferred. This is attractive for causal modeling as it is a priori unknown whether hidden confounding exists. Therefore, our proposed procedure allows for robust causal inference under general confounding settings. In Section 3.1, we illustrate a potential application to climate science and how the choice of ICA can have a strong impact on the estimates of the causal parameters.

Remark 3 (relation between interventions and non-stationarity) *A causal model does not only describe the observational distribution but also the behavior of the data generating model under all of the allowed interventions. Here, we restrict the allowed interventions to distribution shifts in the source signals, that either change the distribution block-wise (e.g., abruptly changing environmental conditions) or continuously (e.g., continuous shifts in the environmental conditions). Any such shifts are by definition synonymous with the process S_i being non-stationary. In our proposed causal model (15) the non-stationarity of the signal therefore corresponds to shifts in the environmental conditions which can be utilized, using coroICA, to infer the underlying causal structure. From this perspective, the causal inference procedure we propose here is a method based on interventional data rather than plainly observational data, while the interventions are not exactly known.*

3.1. Application to Climate Science

To motivate the foregoing causal model we consider a prominent example from climate science: the causal relationship between carbon dioxide concentration (CO_2) and temperature (T). More precisely, we consider Antarctic ice core data that consists of temperature and carbon dioxide measurements of the past 800'000 years due to Bereiter et al. (2015, carbon dioxide) and Jouzel et al. (2007, temperature). We combined both temperature and carbon dioxide data and recorded measurements every 500 years by a cubic interpolation of the raw data. The data is shown in Figure 3 (right). Oversimplifying, one can model this data

as an SCM with time-lags as follows

$$\begin{pmatrix} \log(\mathrm{CO_2})_t \\ \mathrm{T}_t \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \beta \\ \alpha & 0 \end{pmatrix}}_{=B_0} \begin{pmatrix} \log(\mathrm{CO_2})_t \\ \mathrm{T}_t \end{pmatrix} + \sum_{k=1}^{p} B_k \begin{pmatrix} \log(\mathrm{CO_2})_{t-k} \\ \mathrm{T}_{t-k} \end{pmatrix} + \widetilde{S}_t, \tag{16}$$

where $\widetilde{S}_t = S_t + H_t$ with S_t component-wise independent non-stationary source signals and H_t a stationary confounding process. Vector-valued linear time-series models of this type are referred to as structural auto regressive models (SVARs) (see e.g., Lütkepohl, 2005). They have been previously analyzed in the confounding free-case by Hyvärinen et al. (2010), using an ICA based causal inference approach. A graphical representation of such a model is shown in Supplement B.2, Figure 13. In this example, we can think of the source signals S_t as being two independent summaries of important factors that affect both temperature and carbon dioxide and vary over time, e.g., environmental catastrophes like volcano eruptions and large wildfires, sunspot activity or ice-coverage. These variations can be considered as changing environmental conditions or interventions (see Remark 3). On the other hand the stationary confounding process H_t can be thought of as factors which affect both temperature and carbon dioxide in a constant fashion over time, for example this could be effects due the shifts in the earth's rotation axis.

Assuming that this was the true underlying causal model, we could use it to predict what happens under interventions. From a climate science perspective an interesting intervention is given by doubling the concentration of CO_2 and determining the resulting instantaneous (faster than 1000 years) effect on the temperature. This effect is commonly referred to as equilibrium climate sensitivity (ECS) due to CO_2 which is loosely defined as the change in degrees temperature associated with a doubling of the concentration of carbon dioxide in the earth's atmosphere. In the fifth assessment report of the United Nations Intergovernmental Panel on Climate Change it has been stated that "there is high confidence that ECS is extremely unlikely less than 1 °C and medium confidence that the ECS is likely between 1.5 °C and 4.5 °C and very unlikely greater than 6 °C" (Intergovernmental Panel on Climate Change, 2014, Chapter 10). Since the measurement frequency in our model is quite low (500 years) and we model the logarithm of carbon dioxide the ECS corresponds to

$$\mathrm{ECS} = \log(2)\alpha.$$

Estimating the model in (16) can be done by first fitting a vector auto-regressive model of the time lags using OLS resulting in a vector of residuals

$$R_t = \begin{pmatrix} \log(\mathrm{CO_2})_t \\ \mathrm{T}_t \end{pmatrix} - \begin{pmatrix} \widehat{\log(\mathrm{CO_2})}_t \\ \widehat{\mathrm{T}}_t. \end{pmatrix}$$

Then, one can apply the two-step causal inference procedure described in Section 3 to

$$R_t = B_0 R_t + \widetilde{S}_t.$$

Since we are in a two-dimensional setting, step (ii) (i.e., identifying the causal parameters α and β from the estimated mixing matrix) only requires to assume that feedback loops

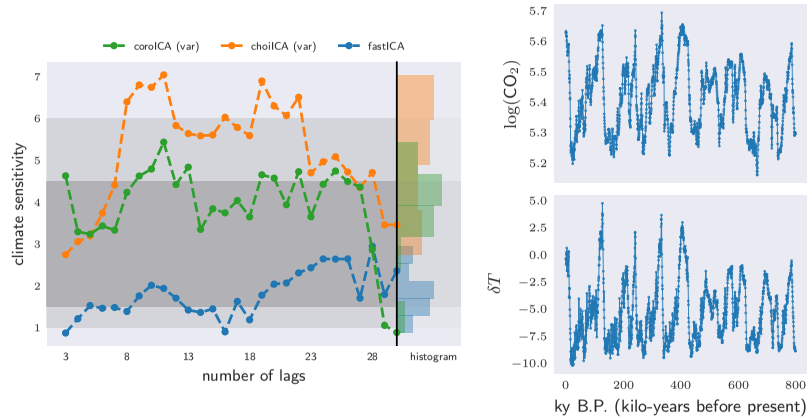


Figure 3. (left) Estimated equilibrium climate sensitivity (ECS) for different ICAs depending on the number of lags included into the SVAR model. The light gray and dark gray overlay indicate likely and very likely value ranges, respectively, for the true value of climate sensitivity as per the fifth assessment report of the United Nations Intergovernmental Panel on Climate Change (cf. Section 3.1). The differences across procedures illustrate that the choice of ICA has a large effect on the estimation. (right) Interpolated time-series data, which we model with an SVAR model.

do not blow-up, which translates into B_0 having spectral norm less than one. Given that the signal is sufficiently strong (i.e., there are sufficient interventions on both CO_2 and T), it is possible to recover the causal parameters by trying both potential permutations of the sources with subsequent scaling and assessing whether the aforementioned condition is satisfied.

We applied this procedure based on coroICA (var) to the data in order to estimate climate sensitivity and compared it with results obtained when using fastICA or choiICA (var). The results are given in Figure 3. We believe the results illustrate two important aspects. Firstly, the choice of the lags has a strong effect on the estimation of the causal effect parameters, particularly for boundary cases. If it is chosen too small the remaining time-dependence in the data can obscure the signal. If it is chosen too big part of the signal starts being removed. Choosing an appropriate number of lags is therefore crucial. One option would be to apply an information criterion (AIC or BIC) for this. Secondly, the results illustrate that the choice of ICA has a large impact on the estimated causal effect parameters. More specifically, both the assumed signal as well as the assumed confounding have an impact on the estimation. Compare the results between fastICA (non-Gaussian signal) and choiICA/coroICA (variance signal) for the former and observe the differences between fastICA/choiICA (no confounding) and coroICA (adjusted for stationary confounding) for the latter. The choice of the ICA algorithm should therefore be driven by the assumptions

(both on signal type and confounding) one is willing to employ on the underlying model. Considering a variance signal and adjusting for confounding, coroICA appears to lead to estimates of equilibrium climate sensitivity that are more closely in line with the highly likely bands previously identified by the United Nations Intergovernmental Panel on Climate Change. This observation is only indicative as all three methods yield highly variable results and also the panel's highly likely band rests on certain assumptions that may become refuted at some later point. coroICA can be considered a conservative choice if no assumptions on confounding can be made, while noise-free methods may outperform if indeed there were no confounding factors.

4. Experiments

In this section, we analyze empirical properties of coroICA. To this end, we first illustrate the performance of coroICA as compared to time-concatenated versions of (noisy) ICA variants on simulated data with and without confounding. We also compare on real data and outline potential benefits of using our method when analyzing multi-subject EEG data.

4.1. Competing Methods

In all of our numerical experiments, we apply coroICA as outlined in Algorithm 1, where we partition each group based on equally spaced grids and run a fixed number of $10 \cdot 10^3$ iterations of the uwedge approximate joint diagonalizer. Unless specified otherwise, coroICA refers to coroICA(var) (i.e., the variance signal based version) and we explicitly write coroICA(var), coroICA(TD) and coroICA(var & TD) whenever appropriate to avoid confusion. We compare with all of the methods in Table 1. Since no Python implementation was publicly available, we implemented the choiICAs and SOBI methods ourselves also based on a fixed number of $10 \cdot 10^3$ iterations of the uwedge approximate joint diagonalizer. For fastICA we use the implementation from the scikit-learn Python library due to Pedregosa et al. (2011) and use the default parameters.

For the simulation experiments in Section 4.2, we also compare to random projections of the sources, where the unmixing matrix is simply sampled with iid standard normal entries. The idea of this comparison is to give a baseline of the unmixing problem and enhance intuition about the scores' behavior. In order to illustrate the variance in this method, we generally sample 100 random projections and show the results for each of them. A random mixing does not lead to interpretable sources, thus we do not compare with random projections in the EEG experiments in Section 4.3.

4.2. Simulations

In this section, we investigate empirical properties of coroICA in well-controlled simulated scenarios. First off, we show that we can recover the correct mixing matrix given that the data is generated according to our model (5) and Assumptions 1 and 2 hold, while the other ICAs necessarily fall short in this setting (cf. Section 4.2.1). Moreover, in Section 4.2.2 we show that even in the absence of any confounding (i.e., when the data follows the ordinary ICA model and $H \equiv 0$ in our model) we remain competitive with all competing ICAs. Finally, in Section 4.2.3 we analyze the performance of coroICA for various types of signals

and noise settings. Our first two simulation experiments are based on block-wise shifting variance signals, which we describe in Data Set 1 and our third simulation experiment is based on GARCH type models described in Data Set 2.

4.2.1. Dependence on Confounding Strength

For this simulation experiment, we sample data according to Data Set 1 and choose to simulate $n = 10^5$ (dimension $d = 22$) samples from $m = 10$ groups where each group contains $n/m = 10^4$ observations. Within each group, we select a random partition consisting of $|\mathcal{P}_g| = 10$ subsets while ensuring that these have the same size on average. We fix the signal strength to $c_1 = 1$ and consider the behavior of coroICA (trained on half of the groups with an equally spaced grid of 10 partitions per group) for different confounding strengths $c_1 = \{0.125, 0.25, 0.5, 1, 1.5, 2, 2.5, 3\}$. The results for 1000 repetitions are shown in Figure 4. To allow for a fair comparison we take the same partition size for choiICA (var).

Data Set 1: Block-wise shifting variance signals

For our simulations we select m equally sized groups $\mathcal{G} := \{g_1, \dots, g_m\}$ of the data points $\{1, \dots, n\}$ and for each group $g \in \mathcal{G}$ construct a partition $\mathcal{P}_g$. Then, we sample a model of the form

$$X_i = A \cdot (S_i + C \cdot H_i)\,,$$

where the values on the right-hand side are sampled as follows:

- $A, C \in \mathbb{R}^{d \times d}$ are sampled with iid entries from $\mathcal{N}(0,1)$ and $\mathcal{N}(0, \frac{1}{d})$, respectively.
- For each $g \in \mathcal{G}$ the variables $H_i \in \mathbb{R}^d$ are sampled from $\mathcal{N}(0, \sigma_g^2 \mathrm{Id}_d)$, where the σ_g^2 are sampled iid from $\mathrm{Unif}(0.1, b_1)$.
- For each $g \in \mathcal{G}$ and $e \in \mathcal{P}_g$ the variables $S_i \in \mathbb{R}^d$ are sampled from $\mathcal{N}(0, \eta_e^2 \mathrm{Id}_d)$, where the η_e^2 are sampled iid from $\mathrm{Unif}(0.1, b_2)$.

The parameters b_1 and b_2 are selected in such a way that the expected confounding strength $c_1 = \mathbb{E}(\sigma_g^2)$ and variance signal strength $c_2 := \mathbb{E}(|\eta_e^2 - \eta_f^2|)$ are as dictated by the respective experiment. Due to the uniform distribution this reduces to

$$b_1 = 2c_1 - 0.1 \quad \text{and} \quad b_2 = 3c_2 + 0.1.$$

The results indicate that in terms of the MD index the competitors all become worse as the confounding strength increases. All competing ICAs systematically estimate an incorrect unmixing matrix. coroICA on the other hand only shows a very small loss in precision as confounding increases; the small loss is expected due to the decreasing signal to noise ratio. In terms of MCIS, the behavior is analogous but slightly less well resolved; with increasing confounding strength the unmixing estimation of all competing ICAs is systematically biased resulting in bad separation of sources and high MCIS scores both out-of-sample and in-sample.

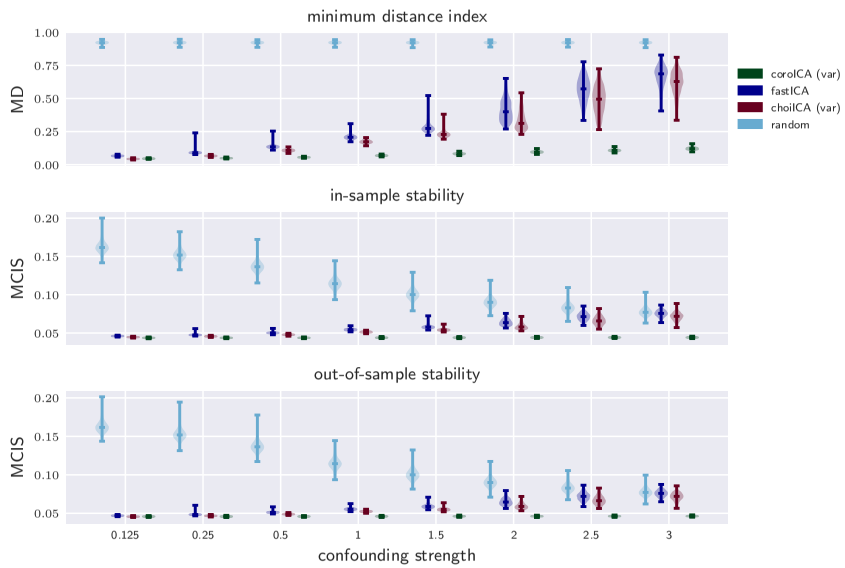


Figure 4. Results of the simulation experiment described in Section 4.2.1. Plot shows performance measures (MD: small implies close to truth; MCIS: small implies stable) for fixed signal strength and various confounding strengths. The difference between the competing ICAs and coroICA is more prominent for higher confounding strengths where the estimates of the competing ICAs are increasingly different from the true unmixing matrix and the sources become increasingly unstable.

4.2.2. Efficiency in Absence of Group Confounding

For this simulation experiment, we sample data according to Data Set 1 and choose to simulate $n = 2 \cdot 10^4$ (dimension $d = 22$) samples from $m = 10$ groups where each group contains $n/m = 2 \cdot 10^3$ observations. Within each group, we then select a random partition consisting of $|\mathcal{P}_g| = 10$ subsets while ensuring that these have the same size on average. This time, to illustrate performance in the absence of confounding, we fix the confounding strengths $c_1 = 0$ and consider the behavior of coroICA (applied to half of the groups with an equally spaced grid of 10 partitions per group) for different signal strengths $c_2 = \{0.025, 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4\}$. The results for 1000 repetitions are shown in Figure 5. Again, choiICA (var) is applied with the same partition size.

The results indicate that overall coroICA performs competitive in the confounding-free case. In particular, there is no drastic negative hit on the performance of coroICA as compared to choiICA (var) in settings where the data follows the ordinary ICA model. The

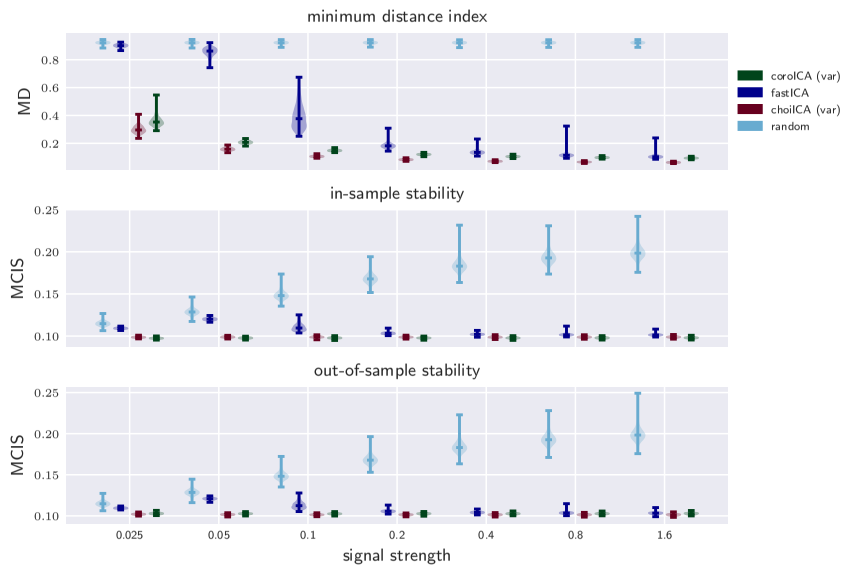


Figure 5. Results of the simulation experiment described in Section 4.2.2. Plot shows performance measures (MD: small implies close to truth; MCIS: small implies stability) for data generated without confounding and for various signal strengths. These results are reassuring, as they indicate that when applied to data that follows the ordinary ICA model, coroICA still performs competitive to competing ICAs even though it allows for a richer model class.

slight advantage compared to fastICA in this setting is due to the signal type which favors ICA methods that focus on variance signals.

4.2.3. Comparison with Other Noisy ICA Procedures

To get a better understanding of how our proposed ICA performs for different signal and noise types, we compare it on simulated data as described in Data Set 2. We illustrate the different behavior with respect to the different types of signal by applying all three of our proposed coroICA procedures (coroICA (var), coroICA (TD) and coroICA (var & TD)) and compare them to the corresponding choiICA variants which do not adjust for confounding (choiICA (var), choiICA (TD) and choiICA (var & TD)). While all coroICA procedures can deal with any type of stationary noise, choiICA (TD) only works for time-independent noise and choiICA (var) and choiICA (var & TD) cannot handle any type of noise at all (see Table 1). Additionally, we also compare with fastICA to assess its performance in the various noise settings. The results are depicted in Figure 6.

Data Set 2: GARCH simulation

For this simulation we consider different settings of the confounded mixing model

$$X_t = AS_t + H_t.$$

More precisely, we consider the following three different GARCH type signals: (i) changing variance, (ii) changing time-dependence, and (iii) both changing variance and changing time-dependence. For each of these signal types we consider two types of confounding (noise) terms: (a) time-independent and (b) time-dependent auto-regressive noise. For both we construct d independent processes $\tilde{H}^1, \ldots, \tilde{H}^d$ and then combine them with a random mixing matrix C as follows

$$H_t = C \cdot \tilde{H}_t.$$

Full details are given in Supplement B.3.

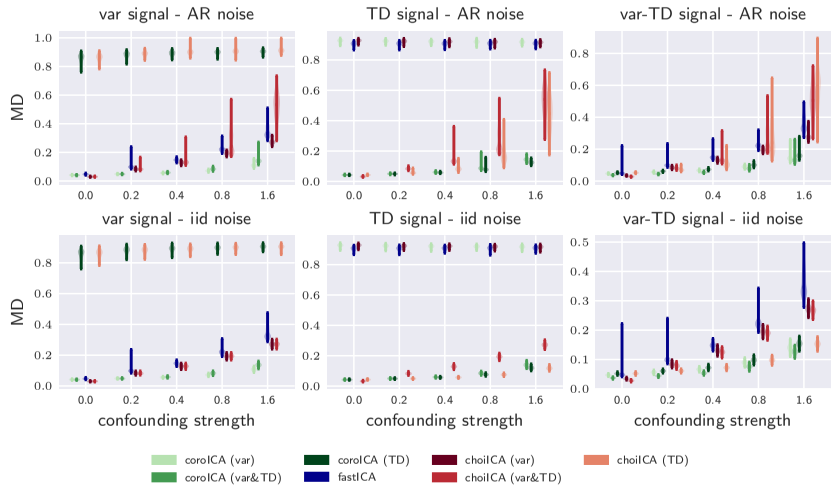


Figure 6. Results of the simulation experiment described in Section 4.2.3 and Data Set 2. Plots show performance (MD: small implies close to truth) for data generated with auto-regressive (AR) or iid noise and for var, TD, and var & TD signal as described in Data Set 2. coroICA (var & TD) is able to estimate the correct mixing in all of the considered settings, while others break whenever the more restrictive signal/noise assumptions are not met.

In all settings the most general method coroICA (var & TD) is able to estimate the correct mixing. The two signal specific methods coroICA (TD) and coroICA (var) are also able to accurately estimate the mixing in settings where a corresponding signal exists. It is also

worth noting that they slightly outperform coroICA (var & TD) in these settings. In contrast, when comparing with the choiICA variants, coroICA is in general able to outperform the corresponding method. Only in the setting of a changing time-dependence with time-independent noise, choiICA (TD) is able to slightly outperform coroICA (TD).

4.2.4. Summary of the Performance of coroICA

In summary, coroICA performs well on a larger model class consisting of both the group-wise confounded as well as the confounding-free case. An advantage over all competing ICAs is gained in confounded settings (as shown in Section 4.2.1) while there is at most a small disadvantage in the unconfounded case (cf. Section 4.2.2). This suggests that whenever the data is expected to contain at least small amounts of stationary noise or confounding, one may be better off using coroICA as the richer model class will guard against wrong results. The results in Section 4.2.3 further underline the robustness of our proposed method to various types of noise (and signals) for which other methods break. Again, even in settings that satisfy the assumptions of the more tailored methods coroICA remains competitive.

4.3. EEG Experiments

ICA is often applied in the analysis of EEG data. Here, we illustrate the potential benefit and use of coroICA for this. Specifically, we consider a multi-subject EEG experiment as depicted in Figure 7. The goal is to find a single mixing matrix that separates the sources simultaneously on all subjects. Our proposed model allows that the EEG recordings for each subject have a different but stationary noise term H. We illustrate the applicability

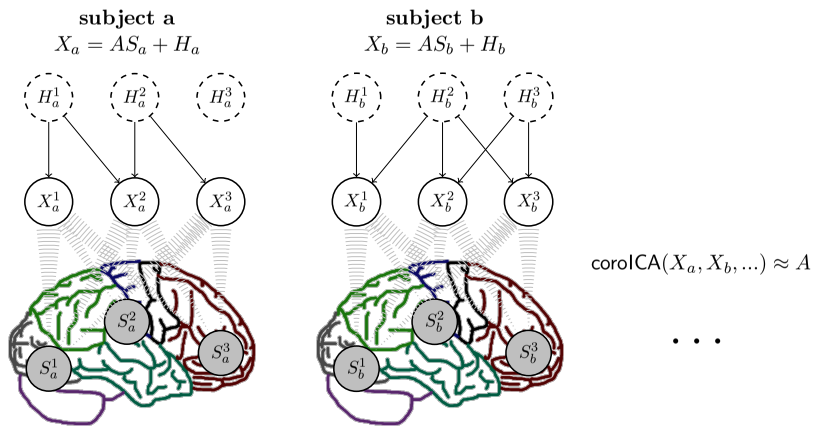


Figure 7. Illustration of a multi-subject EEG recording. For each subject, EEG signals X are recorded which are assumed to be corrupted by subject-specific (but stationary) noise terms H. The goal is to recover a single mixing matrix A that separates signals well across all subjects.

of our method to this setting based on two publicly available EEG data sets.

Data Set 3: CovertAttention data

This data set is due to Treder et al. (2011) and consists of EEG recordings of 8 subjects performing multiple trials of covertly shifting visual attention to one out of 6 cued directions. The data set contains recordings of

- 8 subjects,
- for each subject there exist 6 runs with 100 trials,
- each recording consists of 60 EEG channels recorded at 1000 Hz sampling frequency, while we work with the publicly available data that is downsampled to 200 Hz.

Since visual inspection of the data revealed data segments with huge artifacts and details about how the publicly available data was preprocessed was unavailable to us, we removed outliers and high-pass filtered the data at 0.5 Hz. In particular, along each dimension we set those values to the median along its dimension that deviate more than 10 times the median absolute distance from this median. We further preprocess the data by re-referencing to common average reference (car) and projecting onto the orthogonal complement of the null component. For our unmixing estimations, we use the entire data, i.e., including intertrial breaks.

For classification experiments (cf. Section 4.3.2) we use, in line with Treder et al. (2011), the 8–12 Hz bandpass-filtered data during the 500–2000 ms window of each trial, and use the log-variance as bandpower feature (Lotte et al., 2018). The classification analysis is restricted to valid trials (approximately 311 per subject) with the desired target latency as described in Treder et al. (2011).

Results on the CovertAttention Data Set 3 are presented here, while the results of the analogous experiments on the BCICompIV2a Data Set 4 are deferred to Supplement C. For both data sets, we compare the recovered sources of coroICA with those recovered by competing ICA methods. Since ground truth is unknown we report comparisons based on the following three criteria:

stability and independence
We use MCIS (cf. Section 2.4) to assess the stability and independence of the recovered sources both in- and out-of-sample.

classification accuracy
For both data sets there is label information available that associates certain time windows of the EEG recordings with the task the subjects were performing at that time. Based on the recovered sources, we build a classification pipeline relying on feature extraction and classification techniques that are common in the field (Lotte et al., 2018). The achieved classification accuracy serves as a proxy of how informative and suitable the extracted signals are.

topographies
For a qualitative assessment, we inspect the topographic maps of the extracted sources, as well as the corresponding power spectra and a raw time-series chunk. This is used to illustrate that the sources recovered by coroICA do not appear random or implausible for EEG recordings and are qualitatively similar to what is expected from other ICAs. Furthermore, we provide an overview over all components achieved on Data Set 3 by SOBI, fastICA, and coroICA in the Supplementary Section D, where components are well resolved when the corresponding topographic map and activation map are close to each other (cf. Section 2.4).

4.3.1. Stability and Independence

We aim to probe stability not only in-sample but also verify the expected increase in stability when applying the unmixing matrix to data of new unseen subjects, i.e., to new groups of samples with different confounding specific to that subject. In order to assess stability and independence of the recovered sources in terms of the MCIS both in- and out-of-sample and for different amounts of training samples, we proceed by repeatedly splitting the data into a training and a test data set. More precisely, we construct all possible splits into training and test subjects for any given number of training subjects. For each pair of training and test set, we fit an unmixing matrix using coroICA and all competing methods described in Section 4.1. We then compute the MCIS on the training and test data for each method separately and collect the results of each training-test split for each number of training subjects.

Results obtained on the CovertAttention data set (with equally spaced partitions of ≈15 seconds length) are given in Figure 8 and the results for the BCICompIV2a data set (with equally spaced partitions of ≈15 seconds length) are shown in Supplement C.1, Figure 14. For both data sets the results are qualitatively similar and support the claim that the unmixing obtained by coroICA is more stable when transferred to new unseen subjects. While for the competing ICAs the instability on held-out subjects does not follow a clear decreasing trend with increasing number of training subjects, coroICA can successfully make use of additional training subjects to learn a more stable unmixing matrix.

Due to the characteristics and low signal-to-noise ratio in EEG recordings, the evaluation based on the absolute MCIS score is less well resolved than what we have seen in the simulations before. For this reason we additionally provide a more focused evaluation by considering the MCIS fraction: the fraction of the MCIS achieved on a subject by the respective competitor method divided by the MCIS achieved on that subject by coroICA when trained on the same subjects. Thus, this score compares MCIS on a per subject basis, where values greater than 1 indicate that the respective competing ICA method performed worse than coroICA. Figure 9 shows the results on the CovertAttention Data Set 3 confirming that coroICA can successfully incorporate more training subjects to derive a better unmixing of signals.

4.3.2. Classification based on Recovered Sources

While the results in the previous section indicate that coroICA can lead to more stable separations of sources in EEG than the competing methods, in scenarios with an unknown

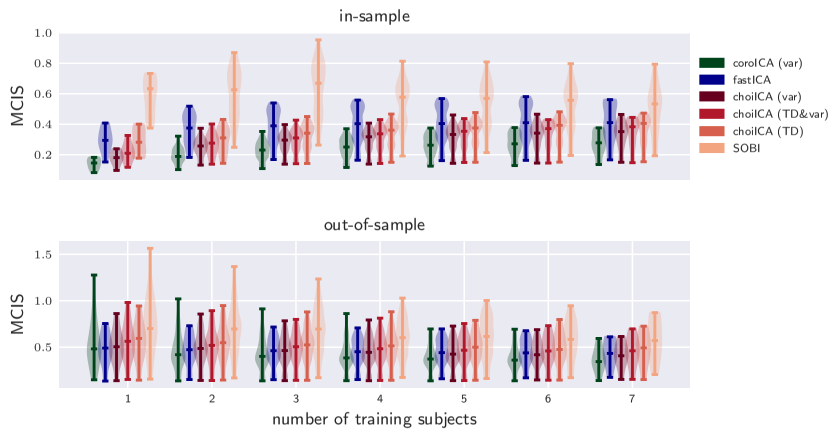


Figure 8. Experimental results for comparing the stability of sources (MCIS: small implies stable) trained on different numbers of training subjects (cf. Section 4.3.1), here on the CovertAttention Data Set 3, demonstrating that coroICA, in contrast to the competing ICA methods, can successfully incorporate more training subjects to learn more stable unmixing matrices when applied to new unseen subjects.

ground truth the stability of the recovered sources cannot serve as the sole determining criterion for assessing the quality of recovered sources. In addition to asking whether the recovered sources are stable and independent variance signals, we hence also need to investigate whether the sources extracted by coroICA are in fact reasonable or meaningful. In the "America's Got Talent Duet Problem" (cf. Example 2) this means that each of the recovered sources should only contain the voice of one (independent) singer (plus some confounding noise that is not the other singer). For EEG data, this assessment is not as easy. Here, we approach this problem from two angles: (a) in this section we show that the recovered sources are informative and suitable for common EEG classification pipelines, (b) in Section 4.3.3 we qualitatively assess the extracted sources based on their power spectra and topographic maps.

In both data sets there are labeled trials, i.e., segments of data during which the subject covertly shifts attention to one of six cues (cf. Data Set 3) or performs one of four motor imagery tasks (cf. Data Set 4). Based on these, one can try to predict the trial label given the trial EEG data. To mimic a situation where the sources are transferred from other subjects, we assess the informativeness of the extracted sources in a leave-k-subjects-out fashion as follows. We estimate an unmixing matrix on data from all but k subjects, compute bandpower features for each extracted signal and for each trial (as described in Data Set 3 and Data Set 4), and on top of those we train an ensemble of 200 bootstrapped shrinkage linear discriminant analysis classifiers where each is boosted by a random forest classifier on the wrongly classified trials. This pipeline (signal unmixing, bandpower-feature

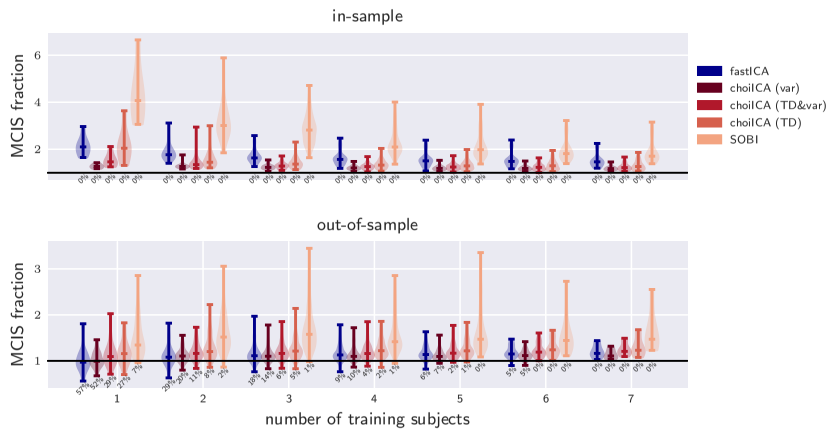


Figure 9. Experimental results for comparing the stability of sources of the competing methods relative to the stability obtained by coroICA (MCIS fraction: above 1 implies less stable than coroICA) trained on different numbers of training subjects (cf. Section 4.3.1), here on the CovertAttention Data Set 3, demonstrating that coroICA can successfully incorporate more training subjects to learn more stable unmixing matrices when applied to new unseen subjects.

computation, trained ensemble classifier), is then used to predict the trials on the k held-out subjects.

The results are reported in Figure 10 and Supplement C.2, Figure 16 which show for each number of training subjects, the accuracies achieved on the respective held-out subjects when using the unmixing obtained on the remaining subjects by either coroICA or one of the competitor methods. The results on both data sets support the claim that the sources recovered by coroICA are not only stable but in addition also capture meaningful aspects of the data that enable competitive classification accuracies in fully-out-of-sample classification. The mean improvement in classification accuracy of coroICA over the other methods increases with increasing number of training subjects. This behavior is expected since it is difficult to disambiguate signal from subject-specific confounding for few training subjects, while coroICA is expected to learn an unmixing which better adjusts for the confounding with more training subjects.

It is worth noting that these classification results depend heavily on the employed classification pipeline subsequent to the source separation. Here, our goal is only to show that coroICA does indeed separate the data into informative sources. In practice, and when only classification accuracy matters, one might also consider using a label-informed source separation (Dähne et al., 2014), employ common spatial patterns (Koles et al., 1990) or use decoding techniques based on Riemannian geometry (Barachant et al., 2012).

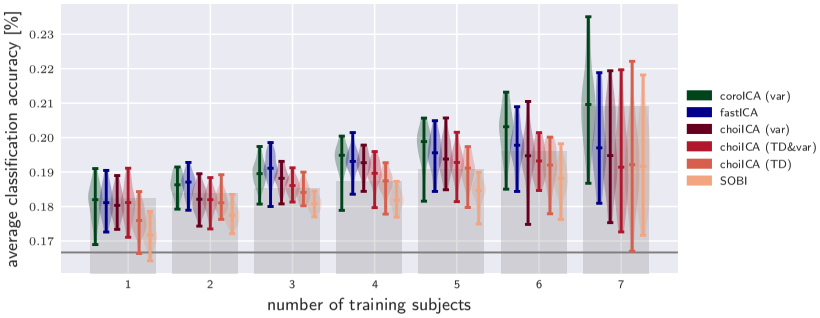


Figure 10. Classification accuracies on held-out subjects (cf. Section 4.3.2), here on the CovertAttention Data Set 3. Gray regions indicate a 95% confidence interval of random guessing accuracies.

4.3.3. Topographic Maps

The components that coroICA extracts from EEG signals are stable (cf. Section 4.3.1) and meaningful in the sense that they contain information that enables classification of trial labels, which is a common task in EEG studies (cf. Section 4.3.2). In this section, we complement the assessment of the recovered sources by demonstrating that the results obtained by coroICA lead to topographies, activation maps, power spectra and raw time-series that are similar to what is commonly obtained during routine ICA analyses of EEG data when the plausibility and nature of ICA components is to be judged.

Topographies are common in the EEG literature to depict the relative projection strength of extracted sources to the scalp sensors. More precisely, the column-vector a_j of $A = V^{-1}$ that specifies the mixing of the j-th source component is visualized as follows. A sketched top view of the head is overlayed with a heatmap where the value at each electrodes' position is given by the corresponding entry in a_j. These topographies are indicative of the nature of the extracted sources, for example the dipolarity of source topographies is a criterion invoked to identify cortical sources (Delorme et al., 2012) or the topographies reveal that the source mainly picks up changes in the electromagnetic field induced by eye movements. Another way to visualize an extracted source is an activation map, which is commonly obtained by depicting the vector $\widehat{\mathrm{Cov}}(X)v_j^\top$ (where v_j is j-th row of unmixing matrix V) and shows for each electrode how the signal observed at that electrode covaries with the signal extracted by v_j (Haufe et al., 2014). Besides inspecting the raw time-series data, another criterion invoked to separate cortical from muscular components is the log power spectrum. For example, a monotonic increase in spectral power starting at around 20 Hz is understood to indicate muscular activity (Goncharova et al., 2003) and peaks in typical EEG frequency ranges are used to identify brain-related components.[5].

5. These are commonly employed criteria which are also advised in the eeglab tutorial (Delorme and Makeig, 2004, https://sccn.ucsd.edu/wiki/Chapter_09:_Decomposing_Data_Using_ICA) and the neurophys-

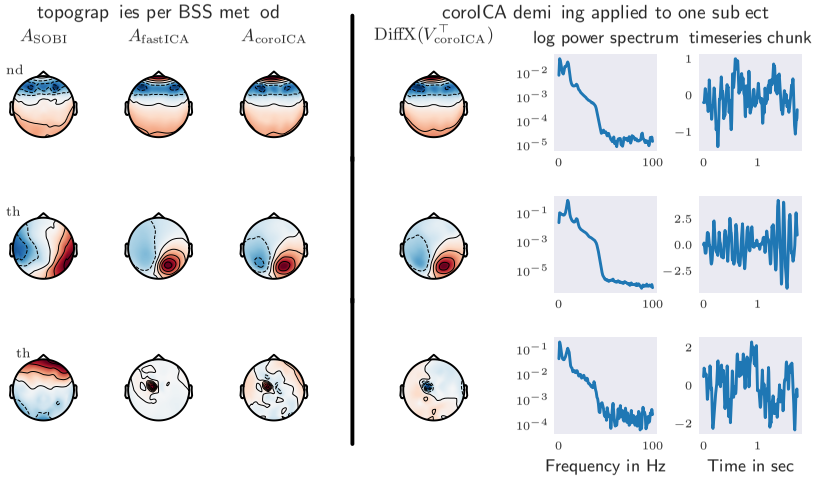


Figure 11. Visualization of exemplary EEG components recovered on the CovertAttention Data Set 3. On the left the topographies of three components are shown where the mixing matrix is the inverse of the unmixing matrix obtained by SOBI (A_{SOBI}), the unmixing matrix obtained by fastICA (A_{fastICA}) and that of coroICA (var) (A_{coroICA}). On the right we depict, for a randomly chosen subject, the activation maps (cf. Section 4.3.3 and 2.4), the log power spectra, and randomly chosen chunks of the raw time-series data corresponding to the respective coroICA (var) components. Components extracted by coroICA (var) are qualitatively similar to those of the commonly employed ICA procedures; see Section 4.3.3 for details.

In Figure 11, we depict the aforementioned criteria for three exemplary components extracted by coroICA on the CovertAttention Data Set 3. Following the discussion in Section 2.4 we show the activation maps as

$$\mathrm{DiffX}(v_j^\top) = \sum_{\mathcal{M}\in\mathcal{M}^*} \mathrm{sign}(v_j M v_j^\top) M v_j^\top,$$

which captures variance changing signal and allows to asses the quality of a recovered source by comparison to the topographic map a_j (cf. Equation 2.4). Here, the idea is to demonstrate that coroICA components are qualitatively similar to components extracted by commonly employed SOBI-ICA or fastICA. Therefore, we choose to display one example of an ocular component (2nd where the topography is indicative of eye movement), a cortical component (7th where the dipolar topography, the typical frequency peak at around 8–12 Hz, and the amplitude modulation visible in the raw time-series are indicative of the

iological biomarker toolbox wiki (Hardstone et al., 2012, https://www.nbtwiki.net/doku.php?id=tutorial:how_to_use_ica_to_remove_artifacts).

cortical nature), and an artifactual component (51st where the irregular topography and the high frequency components indicate an artifact). For comparison, we additionally show for each component the topographies of the components extracted by SOBI-ICA or fastICA by matching the recovered source which most strongly correlates with the one extracted by coroICA. The components extracted by coroICA closely resemble the results one would obtain from a commonly employed ICA analysis on EEG data.

For completeness, we provide an overview over all components extracted on Data Set 3 by SOBI, fastICA, and coroICA (var) in the Supplementary Section D. Components are well resolved when the corresponding topographic map and activation map are close to each other (cf. Section 2.4), which, by visual inspection, appears to be more often the case for coroICA than for the competing methods.

5. Conclusion

In this paper, we propose a method for recovering independent sources corrupted by group-wise stationary confounding. It extends ordinary ICA to an easily interpretable model, which we believe is relevant for many practical problems as is demonstrated in Section 3.1 for climate data and Section 4.3 for EEG data. We give explicit assumptions under which the sources are identifiable in the population case (cf. Section 2.2). Moreover, we introduce a straightforward algorithm for estimating the sources based on the well-understood concept of approximate joint matrix diagonalization. As illustrated in the simulations in Section 4.2, this estimation procedure performs competitive even for data from an ordinary ICA model, while additionally being robust and able to adjust for group-wise stationary confounding. For real data, we show that the coroICA model indeed performs reasonably on EEG data and leads to improvements in comparison to commonly employed approaches, while at the same time preserving an enhanced interpretation of the recovered sources.

Acknowledgments

The authors thank Vinay Jayaram, Nicolai Meinshausen, Jonas Peters, Gian Thanei, the action editor Kenji Fukumizu, and anonymous reviewers for helpful discussions and constructive comments. NP and PB were partially supported by the European Research Commission grant 786461 CausalStats - ERC-2017-ADG.

Supplementary material

The supplementary material consists of the following appendices.

A Identifiability Proof

B Complementary Material

C EEG Experiments on Data Set 4

D All Topographies on Data Set 3

Appendix A. Identifiability Proof

The proof is based on Theorem 1 from Kleinsteuber and Shen (2013). For completeness, we introduce some of the notation therein and state their result with adapted notation to ease following our proof of Theorem 1. We begin by defining the empirical correlation between two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$ as

$$\widehat{\mathrm{Corr}}(\mathbf{v}, \mathbf{w}) := \begin{cases} \frac{\mathbf{v}^\top \mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|}, & \text{if } \mathbf{v} \neq 0 \text{ and } \mathbf{w} \neq 0, \\ 1, & \text{otherwise.} \end{cases}$$

Moreover, for a collection of $(d \times d)$-real diagonal matrices $\{Z_1, \ldots, Z_m\}$, we define the following collinearity measure

$$\rho(Z_1, \ldots, Z_m) := \max_{1 \leq k < l \leq d} |\widehat{\mathrm{Corr}}(\mathbf{z}_k, \mathbf{z}_l)|, \tag{17}$$

where $\mathbf{z}_j := (z_1(j) \ldots, z_m(j))$ and $z_i(j)$ is the j-th diagonal element of the matrix Z_i. Using this notation we can state the uniqueness result due to Kleinsteuber and Shen (2013, Theorem 1) as follows.

Theorem 4 (Kleinsteuber and Shen (2013, Theorem 1)) *Let $D_i \in \mathbb{R}^{d \times d}$, for $i \in \{1, \ldots, m\}$ be diagonal, and let $M \in \mathbb{R}^{d \times d}$ be an invertible matrix so that $M^\top D_i M$ is diagonal as well. Then M is essentially, up to scaling and permutation of its columns, unique if and only if $\rho(D_1, \ldots, D_m) < 1$.*

Using this result we prove Theorem 1.

Proof The theorem is proven by the correct invocation of Theorem 4. We first define the unmixing matrix $V := A^{-1}$ and introduce the sets of matrices

$$\mathcal{D}_{\mathrm{var}} := \{V(\mathrm{Cov}(X_k) - \mathrm{Cov}(X_l))V^\top \,|\, g \in \mathcal{G} \text{ and } k, l \in g\}.$$

and

$$\mathcal{D}_{\mathrm{TD}} := \{V(\mathrm{Cov}(X_k, X_{k-\tau}) - \mathrm{Cov}(X_l, X_{l-\tau}))V^\top \,|\, g \in \mathcal{G} \text{ and } k, l \in g\}.$$

Due to the assumed ICA model and Assumption 1, all matrices in the sets $\mathcal{D}_{\text{var}}$ and $\mathcal{D}_{\text{TD}}$ are diagonal (cf. (6) and (7)). Moreover, for $g \in \mathcal{G}$ and $k, l \in g$ it holds that

$$\begin{aligned} &V(\text{Cov}(X_k) - \text{Cov}(X_l))V^\top \\ &\quad = \text{Cov}(S_k) - \text{Cov}(S_l) \\ &\quad = \text{diag}(\text{Var}(S_k^1) - \text{Var}(S_l^1), \ldots, \text{Var}(S_k^d) - \text{Var}(S_l^d)) \end{aligned}$$

and

$$\begin{aligned} &V(\text{Cov}(X_k, X_{k-\tau}) - \text{Cov}(X_l, X_{l-\tau}))V^\top \\ &\quad = \text{Cov}(S_k, S_{k-\tau}) - \text{Cov}(S_l, S_{l-\tau}) \\ &\quad = \text{diag}(\text{Cov}(S_k^1, S_{k-\tau}^1) - \text{Cov}(S_l^1, S_{l-\tau}^1), \ldots, \text{Cov}(S_k^d, S_{k-\tau}^d) - \text{Cov}(S_l^d, S_{l-\tau}^d)). \end{aligned}$$

Next, we define for all $j \in \{1, \ldots, d\}$ the vectors

$$\begin{aligned} \mathbf{z}_j &= \left(\left(\text{Var}(S_k^j) - \text{Var}(S_l^j) \right)_{k,l \in g} \right)_{g \in \mathcal{G}} \\ \text{or} \quad \mathbf{z}_j &= \left(\left(\text{Cov}(S_k^j, S_{k-\tau}^j) - \text{Cov}(S_l^j, S_{l-\tau}^j) \right)_{k,l \in g} \right)_{g \in \mathcal{G}}, \end{aligned}$$

depending on whether a variance signal or time-dependence signal is being considered, respectively. Then, Assumption 2 or Assumption 3 implies for all distinct pairs $p, q \in \{1, \ldots, d\}$ that

$$|\widehat{\text{Corr}}(\mathbf{z}_p, \mathbf{z}_q)| = \frac{|\mathbf{z}_p \cdot \mathbf{z}_q|}{\|\mathbf{z}_p\| \|\mathbf{z}_q\|} < 1.$$

Hence, for either $\mathcal{D} = \mathcal{D}_{\text{var}}$ or $\mathcal{D} = \mathcal{D}_{\text{TD}}$ it holds that $\rho(\mathcal{D}) < 1$, where ρ is defined in (17). Since the identity matrix satisfies that $\text{Id}\, D\, \text{Id}^\top$ is diagonal for all $D \in \mathcal{D}$, we can invoke Theorem 4 to conclude that any matrix $M \in \mathbb{R}^{d \times d}$ for which $MDM^\top$ is diagonal for all $D \in \mathcal{D}$, is equal to the identity matrix up to scaling and permutation of its columns. Next, we consider the two signal types separately.

- **variance signal:** If there is a variance signal that satisfies Assumption 2, assume there exists an invertible matrix $\widetilde{A}$ such that for all $g \in \mathcal{G}$ and all $k, l \in g$ it holds that

  $$\widetilde{A}^{-1}(\text{Cov}(X_k) - \text{Cov}(X_l))(\widetilde{A}^{-1})^\top = \text{Cov}(S_k) - \text{Cov}(S_l).$$

  Then, it also holds that

  $$(V\widetilde{A}) \underbrace{(\text{Cov}(S_k) - \text{Cov}(S_l))}_{\in \mathcal{D}_{\text{var}}} (V\widetilde{A})^\top = V(\text{Cov}(X_k) - \text{Cov}(X_l))V^\top,$$

  which is diagonal.

- **time-dependence signal:** If there is a time-dependence signal that satisfies Assumption 3, assume there exists an invertible matrix $\widetilde{A}$ such that for all $g \in \mathcal{G}$ and all $k, l \in g$ it holds that

  $$\widetilde{A}^{-1}(\text{Cov}(X_k, X_{k-\tau})) - \text{Cov}(X_l, X_{l-\tau}))(\widetilde{A}^{-1})^\top = \text{Cov}(S_k, S_{k-\tau}) - \text{Cov}(S_l, S_{l-\tau}).$$

  Then, it also holds that

  $$(V\widetilde{A}) \underbrace{(\text{Cov}(S_k, S_{k-\tau}) - \text{Cov}(S_l, S_{l-\tau}))}_{\mathcal{D}_{\text{TD}}} (V\widetilde{A})^\top = V(\text{Cov}(X_k, X_{k-\tau})) - \text{Cov}(X_l, X_{l-\tau}))V^\top,$$

  which is diagonal.

Using the above reasoning, either of the two cases—depending on whether Assumption 2 or 3 holds—shows that $V\widetilde{A}$ is equal to the identity matrix up to permutation and rescaling of its columns. Moreover, this implies that $\widetilde{A}$ is equal to A up to scaling and permutation of its columns. This completes the proof of Theorem 1. ■

Appendix B. Complementary Material

B.1. America's Got Talent

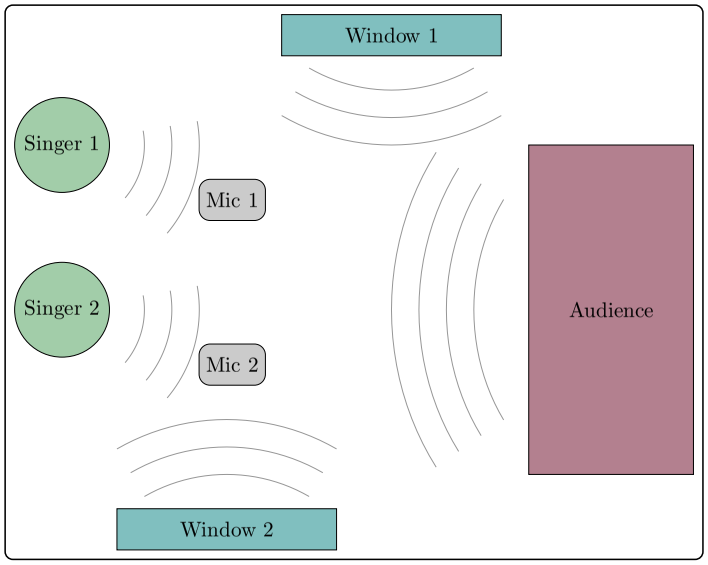


Figure 12. Schematic of the "America's Got Talent Duet Problem" described in Example 2. The sound from the windows and audience is taken to be confounding noise which has fixed covariance structure over given time blocks. The challenge is to recover the sound signals from the individual singers given the recordings of the two microphones.

B.2. Causality Example

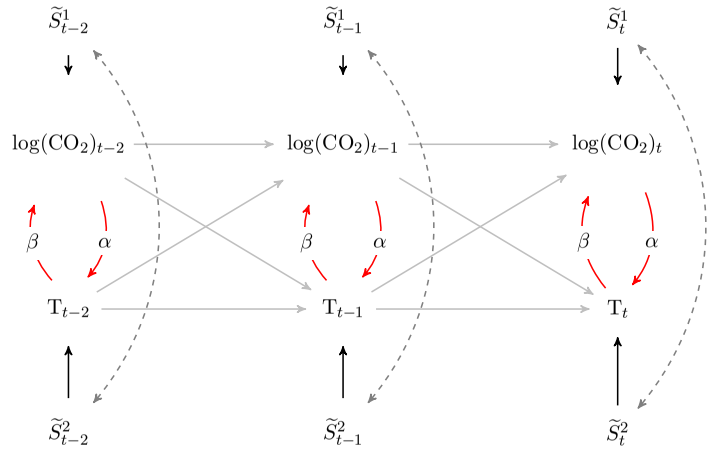


Figure 13. Graphical representation of the causal feedback model between carbon dioxide (CO_2) and temperature (T). The dashed line corresponds to stationary confounding.

B.3. Simulations

The GARCH model that we simulate from in Section 4.2.3 is specified as follows. We simulate sources $S^1, \dots, S^d$ from the following GARCH-type model

$$\sigma_i^2 = a_1 + a_2 \cdot (S_{i-1}^j)^2 + a_3 \cdot \sigma_{i-1}^2$$
$$S_i^j = b_1 S_{i-1}^j + \cdots + b_p S_{i-p}^j + \sigma_i \varepsilon_i,$$

where the ε_i are independent and standard normal. Moreover, the noise terms $H^1, \dots, H^d$ are assumed to be either given by the following AR-process

$$H_i^j = c_1 H_{i-1}^j + \cdots + c_q H_{i-q}^j + \nu_i,$$

where ν_i are independent standard normal, q is uniformly distributed on $\{1, \dots, 10\}$ and c_i independent $\mathcal{N}\left(0, 1/(i+1)^2\right)$ or simply as iid $\mathcal{N}(0,1)$ random variables. The final data is then constructed according to the following equation

$$X_i = A \cdot S_i + \widetilde{H}_i,$$

where $\widetilde{H}_i = ACH_i$ and $A, C \in \mathbb{R}^{d \times d}$ are sampled with iid entries from $\mathcal{N}(0,1)$ and $\mathcal{N}(0, \frac{1}{d})$, respectively. To illustrate, the effect of the signal type we consider the following three settings.

- **Setting 1 (time-independent with changing variance)**
  Set $a = (0.005, 0.026, 0.97)$ such that the variance changes over time and $p = 0$ to ensure time-independent signals. Based on these settings we sample $n = 200000$ observations.

- **Setting 2 (varying time-dependence structure with constant variance)**
  Set $a = (1, 0, 0)$ such that the variance is fixed to 1. Then, sample p 100 times uniformly from $\{1, \ldots, 10\}$ and b_i independent from $\mathcal{N}\left(0, 1/(i+1)^2\right)$ and simulate 2000 observations for each of the 100 parameter settings, leading to a total of $n = 200000$ observations.

- **Setting 3 (varying time-dependence structure with changing variance)**
  Set $a = (0.005, 0.026, 0.97)$ such that the variance changes over time. Then, we sample p 100 times uniformly from $\{1, \ldots, 10\}$ and b_i independent from $\mathcal{N}\left(0, 1/(i+1)^2\right)$ and simulate 2000 observations for each of the 100 parameter settings, leading to a total of $n = 200000$ observations.

Appendix C. EEG Experiments on Data Set 4

Analogous to Sections 4.3.1 and 4.3.2 we conducted experiments on the BCICompIV2a Data Set 4, the results of which are presented in the subsequent sections.

Data Set 4: BCICompIV2a data

This data set is due to Tangermann et al. (2012, Section 5) and consists of EEG recordings of 9 subjects performing multiple trials of 4 different motor imagery tasks. The data set contains recordings of

- 9 subjects, each recorded on 2 different days,
- for each subject and day there exist 6 runs with 48 trials,
- each recording consists of 22 EEG channels recorded at 250 Hz sampling frequency,
- and is bandpass filtered between 0.5 and 100 Hz and is 50 Hz notch filtered.

For our analysis we only use the trial-data, i.e., the concatenated segments of seconds 3–6 of each trial (corresponding to the motor imagery part of the trials (Tangermann et al., 2012)). We further preprocess the data by re-referencing to common average reference (car) and projecting onto the orthogonal complement of the null component.

As features for classification experiments (cf. Section 4.3.2) on this data set we use bandpower in the 8–30 Hz band as measured by the log-variance of the 8–30 Hz bandpass-filtered trial data (Lotte et al., 2018).

C.1. Stability and Independence

Figure 14. Experimental results for comparing the stability of sources (MCIS: small implies stable) trained on different numbers of training subjects (cf. Section 4.3.1), here on the BCICompIV2a Data Set 4, demonstrating that coroICA, in contrast to the competing ICA methods, can successfully incorporate more training subjects to learn more stable unmixing matrices when applied to new unseen subjects.

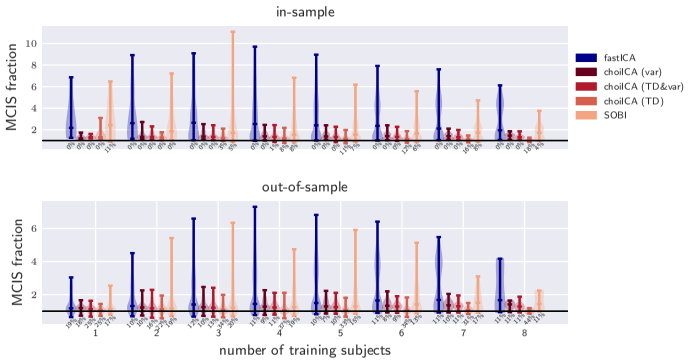


Figure 15. Experimental results for comparing the stability of sources of competitors relative to the stability obtained by coroICA (MCIS fraction: > 1 implies less stable than coroICA) trained on different numbers of training subjects (cf. Section 4.3.1), here on the BCICompIV2a Data Set 4, demonstrating that coroICA can successfully incorporate more training subjects to learn more stable unmixing matrices when applied to new unseen subjects.

C.2. Classification based on Recovered Sources

Figure 16. Classification accuracies on held-out subjects (cf. Section 4.3.2), here on the BCICompIV2a Data Set 4. Gray regions indicate a 95% confidence interval of random guessing accuracies.

Appendix D. All Topographies and Activation Maps on Data Set 3

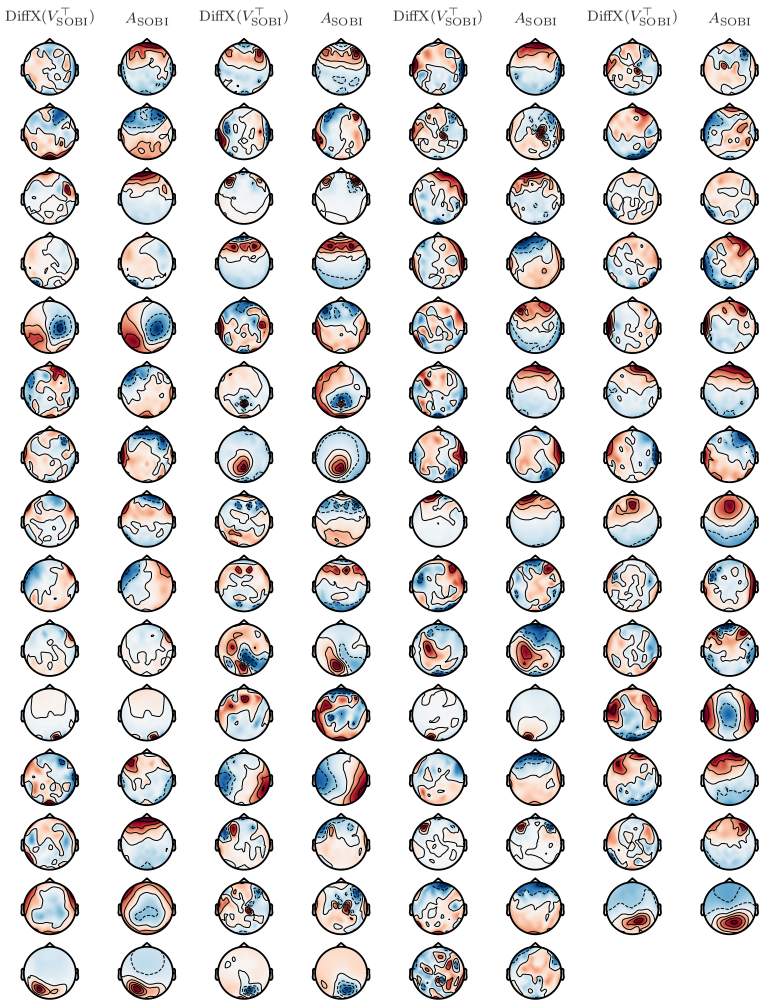


Figure 17. Activation maps (left of each pair of columns) and topographies (right of each pair of columns) of 59 components extracted by SOBI on the CovertAttention Data Set 3. For components that are well resolved, both should look similar (cf. Section 2.4 and 4.3.3).

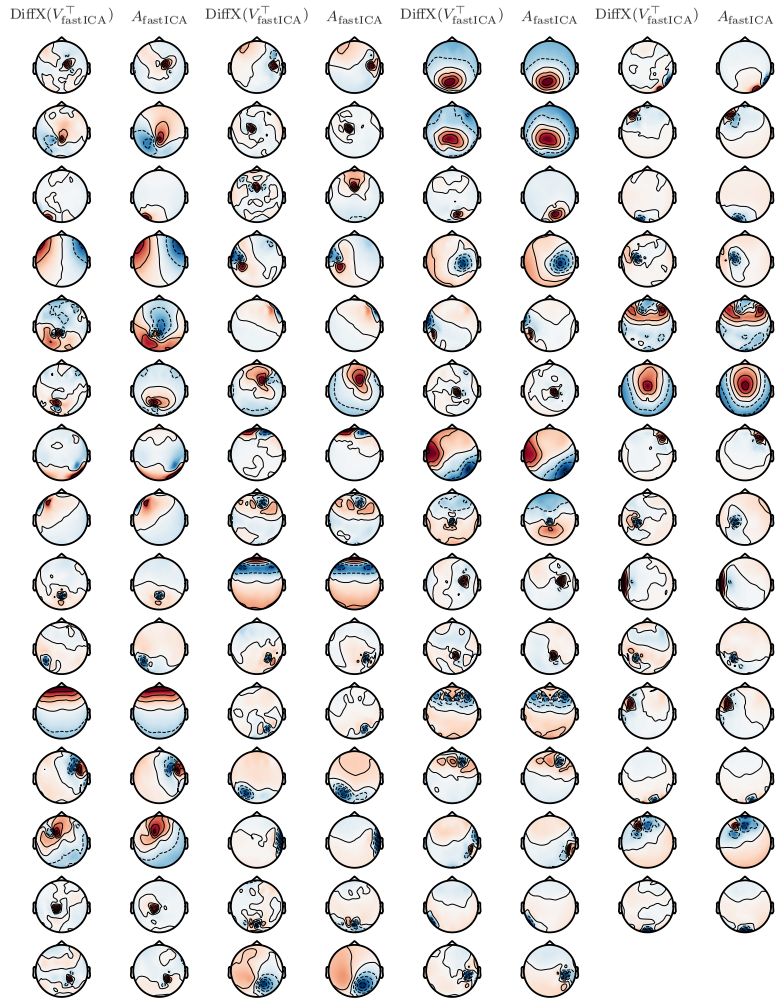


Figure 18. Activation maps (left of each pair of columns) and topographies (right of each pair of columns) of 59 components extracted by fastICA on the CovertAttention Data Set 3. For components that are well resolved, both should look similar (cf. Section 2.4 and 4.3.3).

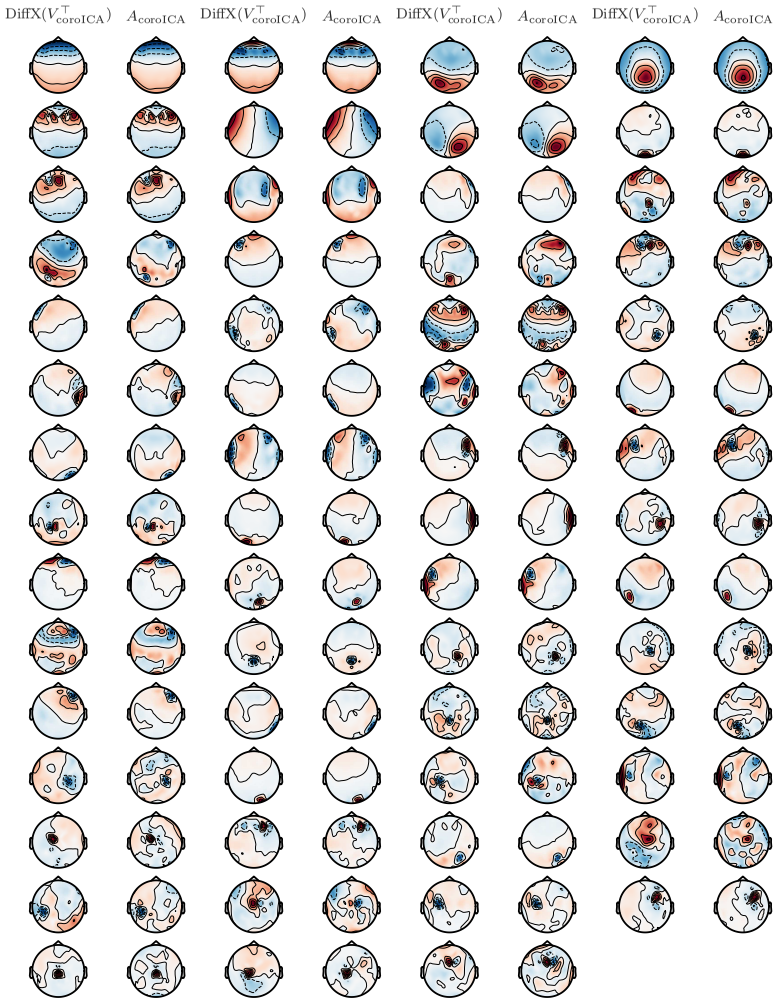


Figure 19. Activation maps (left of each pair of columns) and topographies (right of each pair of columns) of 59 components extracted by coroICA (var) on the CovertAttention Data Set 3. For components that are well resolved, both should look similar (cf. Section 2.4 and 4.3.3).

References

P. Ablin, J.-F. Cardoso, and A. Gramfort. Beyond Pham's algorithm for joint diagonalization. *arXiv preprint arXiv:1811.11433v1*, 2018.

S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems 8 (NIPS 1995)*, pages 757–763. NIPS Foundation, 1995.

A. Back and A. Weigend. A first application of independent component analysis to extracting structure from stock returns. *International Journal of Neural Systems*, 8(4):473–484, 1997.

A. Barachant, S. Bonnet, M. Congedo, and C. Jutten. Multiclass brain-computer interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2012.

C. Beckmann and S. M. Smith. Tensorial extensions of independent component analysis for multisubject fMRI analysis. *NeuroImage*, 25(1):294–311, 2005.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2): 434–444, 1997.

B. Bereiter, S. Eggleston, J. Schmitt, C. Nehrbass-Ahles, T. F. Stocker, H. Fischer, S. Kipfstuhl, and J. Chappellaz. Revision of the EPICA Dome C CO_2 record from 800 to 600 kyr before present. *Geophysical Research Letters*, 42(2):542–549, 2015.

V. D. Calhoun, T. Adalı, L. K. Hansen, J. Larsen, and J. J. Pekar. ICA of functional MRI data: An overview. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 281–288, 2003.

J.-F. Cardoso. Source separation using higher order moments. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 2109–2112 vol.4, May 1989a.

J.-F. Cardoso. Blind identification of independent components with higher-order statistics. In *Workshop on Higher-Order Spectral Analysis*, pages 157–162, 06 1989b.

J.-F. Cardoso and A. Souloumiac. Blind beamforming for non-Gaussian signals. *IEE Proceedings F - Radar and Signal Processing*, 140(6):362–370, 1993.

J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.

S. Choi and A. Cichocki. Blind separation of nonstationary and temporally correlated sources from noisy mixtures. In *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, pages 405–414. IEEE, 2000a.

S. Choi and A. Cichocki. Blind separation of nonstationary sources in noisy mixtures. *Electronics Letters*, 36(9):848–849, 2000b.

S. Choi and A. Cichocki. Algebraic differential decorrelation for nonstationary source separation. *Electronics Letters*, 37(23):1414–1415, 2001.

S. Choi, A. Cichocki, and A. Belouchrani. Blind separation of second-order nonstationary and temporally colored sources. In *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pages 444–447. IEEE, 2001.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3): 287–314, 1994.

S. Dähne, F. Meinecke, S. Haufe, J. Höhne, M. Tangermann, K.-R. Müller, and V. Nikulin. SPoC: a novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters. *NeuroImage*, 86:111–122, 2014.

A. Delorme and S. Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9–21, 2004.

A. Delorme, T. Sejnowski, and S. Makeig. Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. *NeuroImage*, 34(4): 1443–1449, 2007.

A. Delorme, J. Palmer, J. Onton, R. Oostenveld, and S. Makeig. Independent EEG Sources Are Dipolar. *PLOS One*, 7:1–14, 02 2012.

D. Ghahremani, S. Makeig, T. Jung, A. Bell, and T. Sejnowski. Independent component analysis of simulated EEG using a three-shell spherical head model. Technical report, Naval Health Research Center San Diego CA, 1996.

I. Goncharova, D. McFarland, T. Vaughan, and J. Wolpaw. EMG contamination of EEG: spectral and topographical characteristics. *Clinical Neurophysiology*, 114(9):1580–1593, 2003.

T. Haavelmo. The Probability Approach in Econometrics. *Econometrica*, 12:iii–115, 1944.

R. Hardstone, S.-S. Poil, G. Schiavone, R. Jansen, V. Nikulin, H. Mansvelder, and K. Linkenkaer-Hansen. Detrended fluctuation analysis: a scale-free view on neuronal oscillations. *Frontiers in Physiology*, 3:450, 2012.

R. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis, 1970.

S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, 87:96–110, 2014.

P. Hoyer, S. Shimizu, A. Kerminen, and M. Palviainen. Estimation of causal effects using linear non-Gaussian causal models with hidden variables. *International Journal of Approximate Reasoning*, 49(2):362–378, 2008.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen. Fast ICA for noisy data using Gaussian moments. In *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI*, pages 57–61. IEEE, 1999.

A. Hyvärinen. Blind source separation by nonstationarity of variance: a cumulant-based approach. *IEEE Transactions on Neural Networks*, 12(6):1471–1474, 2001.

A. Hyvärinen. Independent component analysis: recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371 (1984):20110534, 2013.

A. Hyvärinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pages 3765–3773. NIPS Foundation, 2016.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2002.

A. Hyvärinen, K. Zhang, S. Shimizu, and P. Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11 (5):1709–1731, 2010.

P. Ilmonen, K. Nordhausen, H. Oja, and E. Ollila. A new performance index for ICA: Properties, computation and asymptotic analysis. In *Latent Variable Analysis and Signal Separation*, pages 229–236. Springer, 2010.

G. W. Imbens and D. B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences*. Cambridge University Press, 2015.

Intergovernmental Panel on Climate Change. *Climate Change 2013 – The Physical Science Basis*. Cambridge University Press, 2014.

J. Jouzel, V. Masson-Delmotte, O. Cattani, G. Dreyfus, S. Falourd, G. Hoffmann, B. Minster, J. Nouet, J.-M. Barnola, J. Chappellaz, H. Fischer, J. C. Gallet, S. Johnsen, M. Leuenberger, L. Loulergue, D. Lüthi, H. Oerter, F. Parrenin, Raisbeck G., D. Raynaud, J. Schilt, A. Schwander, E. Selmo, R. Souchez, R. Spahni, B. Stauffer, J. P. Steffensen, B. Stenni, T. Stocker, J. L. Tison, M. Werner, and E. W. Wolff. Orbital and millennial antarctic climate variability over the past 800,000 years. *Science*, 317(5839): 793–796, 2007.

T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. Mckeown, V. Iragui, and T. Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(2):163–178, 2000.

M. Kleinsteuber and H. Shen. Uniqueness analysis of non-unitary matrix joint diagonalization. *IEEE Transactions on Signal Processing*, 61(7):1786–1796, 2013.

Z. J. Koles, M. S. Lazar, and S. Z. Zhou. Spatial patterns underlying population differences in the background EEG. *Brain Topography*, 2(4):275–284, 1990.

F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger. A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3):031005, 2018.

H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media, 2005.

S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski. Independent component analysis of electroencephalographic data. In *Advances in Neural Information Processing Systems 8 (NIPS 1995)*, pages 145–151. NIPS Foundation, 1995.

S. Makeig, T.-P. Jung, A. Bell, D. Ghahremani, and T. Sejnowski. Blind separation of auditory event-related brain responses into independent components. *Proceedings of the National Academy of Sciences*, 94(20):10979–10984, 1997.

S. Makeig, M. Westerfield, T.-P. Jung, S. Enghoff, J. Townsend, E. Courchesne, and T. Sejnowski. Dynamic brain sources of visual evoked responses. *Science*, 295(5555):690–694, 2002.

K. Matsuoka, M. Ohoya, and M. Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.

M. McKeown, T.-P. Jung, S. Makeig, G. Brown, S. Kindermann, T.-W. Lee, and T. Sejnowski. Spatially independent activity patterns in functional MRI data during the Stroop color-naming task. *Proceedings of the National Academy of Sciences*, 95(3):803–810, 1998a.

M. McKeown, S. Makeig, G. Brown, T.-P. Jung, S. Kindermann, A. Bell, and T.-J. Sejnowski. Analysis of fMRI data by blind separation into independent spatial components. *Human Brain Mapping*, 6(3):160–188, 1998b.

J. Miettinen, K. Nordhausen, H. Oja, and S. Taskinen. Statistical properties of a blind source separation estimator for stationary time series. *Statistics & Probability Letters*, 82 (11):1865–1873, 2012.

E. Moulines, J.-F. Cardoso, and E. Gassiat. Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3617–3620. IEEE, 1997.

K. Nordhausen. On robustifying some second order blind source separation methods for nonstationary time series. *Statistical Papers*, 55(1):141–156, Feb 2014.

P.L. Nunez and R. Srinivasan. *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, 2006.

Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(10):2825–2830, 2011.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference*. MIT Press, 2017.

Pexels GmbH. Pexels. https://www.pexels.com/, 2018. Accessed: 2018-10-30.

D.-T. Pham and J.-F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Transactions on Signal Processing*, 49(9):1837–1848, 2001.

D. Rothenhäusler, C. Heinze, J. Peters, and N. Meinshausen. BACKSHIFT: Learning causal cyclic graphs from unknown shift interventions. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 1513–1521. NIPS Foundation, 2015.

S. Shimizu, P. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10):2003–2030, 2006.

M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. Miller, G. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz. Review of the BCI Competition IV. *Frontiers in Neuroscience*, 6:55, 2012.

P. Tichavsky and A. Yeredor. Fast approximate joint diagonalization incorporating weight matrices. *IEEE Transactions on Signal Processing*, 57(3):878–891, 2009.

L. Tong, V. C. Soon, Y. F. Huang, and R. Liu. AMUSE: A new blind identification algorithm. In *IEEE International Symposium on Circuits and Systems*, pages 1784–1787. IEEE, 1990.

M. Treder, A. Bahramisharif, N. Schmidt, M. Van Gerven, and B. Blankertz. Brain-computer interfacing using modulations of alpha activity induced by covert shifts of attention. *Journal of NeuroEngineering and Rehabilitation*, 8(1):24, 2011.

L. Zhukov, D. Weinstein, and C. Johnson. Independent component analysis for EEG source localization. *IEEE Engineering in Medicine and Biology Magazine*, 19(3):87–96, 2000.

A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5(7):777–800, 2004.

Journal of Machine Learning Research 20 (2019) 1-43 Submitted 7/18; Revised 10/19; Published 10/19

Collective Matrix Completion

Mokhtar Z. Alaya MOKHTARZAHDI.ALAYA@GMAIL.COM
Modal'X, UPL, Univ Paris Nanterre,
F92000 Nanterre France

Olga Klopp KLOPPOLGA@MATH.CNRS.FR
ESSEC Business School and CREST,
F95021 Cergy France

Editor: Xiaotong Shen

Abstract

Matrix completion aims to reconstruct a data matrix based on observations of a small number of its entries. Usually in matrix completion a single matrix is considered, which can be, for example, a rating matrix in recommendation system. However, in practical situations, data is often obtained from multiple sources which results in a collection of matrices rather than a single one. In this work, we consider the problem of collective matrix completion with multiple and heterogeneous matrices, which can be count, binary, continuous, etc. We first investigate the setting where, for each source, the matrix entries are sampled from an exponential family distribution. Then, we relax the assumption of exponential family distribution for the noise. In this setting, we do not assume any specific model for the observations. The estimation procedures are based on minimizing the sum of a goodness-of-fit term and the nuclear norm penalization of the whole collective matrix. We prove that the proposed estimators achieve fast rates of convergence under the two considered settings and we corroborate our results with numerical experiments.

Keywords: High-dimensional prediction; Exponential families; Low-rank matrix estimation; Nuclear norm minimization; Low-rank optimization;

1. Introduction

Completing large-scale matrices has recently attracted great interest in machine learning and data mining since it appears in a wide spectrum of applications such as recommender systems (Koren et al., 2009; Bobadilla et al., 2013), collaborative filtering (Netflix challenge) (Goldberg et al., 1992; Rennie and Srebro, 2005), sensor network localization (So and Ye, 2005; Drineas et al., 2006; Oh et al., 2010), system identification (Liu and Vandenberghe, 2009), image processing (Hu et al., 2013), among many others. The basic principle of matrix completion consists in recovering all the entries of an unknown data matrix from incomplete and noisy observations of its entries.

To address the high-dimensionality in matrix completion problem, statistical inference based on low-rank constraint is now an ubiquitous technique for recovering the underlying data matrix. Thus, matrix completion can be formulated as minimizing the rank of the matrix given a random sample of its entries. However, this rank minimization problem is in general NP-hard due to the combinatorial nature of the rank function (Fazel et al., 2001; Fazel, 2002). To alleviate this problem and make it tractable, convex relaxation strategies

©2019 Mokhtar Z. Alaya and Olga Klopp.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-483.html.

were proposed, e.g., the nuclear norm relaxation (Srebro et al., 2005; Candes and Tao, 2010; Recht et al., 2010; Negahban and Wainwright, 2011; Klopp, 2014) or the max-norm relaxation (Cai and Zhou, 2016). Among those surrogate approximations, nuclear norm, which is defined as the sum of the singular values of the matrix or the ℓ_1-norm of its spectrum, is probably the most widely used penalty for low-rank matrix estimation, since it is the tightest convex lower bound of the rank (Fazel et al., 2001).

Motivations. Classical matrix completion focus on a single matrix, whereas in practical situations data is often obtained from a collection of matrices that may cover multiple and heterogeneous sources. For example, in e-commerce users express their feedback for different items such as books, movies, music, etc. In social networks like Facebook and Twitter users often share their opinions and interests on a variety of topics (politics, social events, health). In this examples, informations from multiple sources can be viewed as a collection of matrices coupled through a common set of users.

Rather than exploiting user preference data from each source independently, it may be beneficial to leverage all the available user data provided by various sources in order to generate more encompassing user models (Cantador et al., 2015). For instance, some recommender system runs into the so-called *cold-start problem* (Lam et al., 2008). A user is new or "cold" in a source when he has few to none rated items. Such user may have a rating history in auxiliary sources and we can use his profile in the auxiliary sources to recommend relevant items in the target source. For example, a user's favorite movie genres may be derived from his favorite book genres. Therefore, this shared structure among the sources can be useful to get better predictions (Singh and Gordon, 2008; Bouchard et al., 2013; Gunasekar et al., 2016).

More generally speaking, collective matrix completion finds a natural application in the problem of recommender system with side information. In this problem, in addition to the conventional user-item matrix, it is assumed that we have side information about each user (Chiang et al., 2015; Jain and Dhillon, 2013; Fithian and Mazumder, 2018; Agarwal et al., 2011). For example, in blog recommendation task, we may have access to user generated content (images, tags and text) or user activity (e.g., likes and reblogs). Such side information may be used to improve the quality of recommendation of blogs of interest (Shin and Lee, 2015).

Based on the type of available side information, various methods for recommender systems with side information have been proposed. It can be user generated content (Armentano et al., 2013; Hannon et al., 2010), user/item profile or attribute (Agarwal et al., 2011), social network (Jamali and Ester, 2010; Ma et al., 2011) and context information (Natarajan et al., 2013). A very interesting surveys of the state-of-the-art methods can be found in (Fithian and Mazumder, 2018; Natarajan et al., 2013).

On the other hand, our framework includes the model of Mixed Data Frames with missing observations (Pagès, 2014; Udell et al., 2016). Here matrices collect categorical, numerical and count observations. They appear in numerous applications including in ecology, patient records in health care (Gunasekar et al., 2016), quantitative gene expression values (Natarajan and Dhillon, 2014; Zitnik and Zupan, 2014, 2015), and also in recommender systems and survey data.

Main contributions and related literature. In this paper, we extend the theory of low-rank matrix completion to a collection of multiple and heterogeneous matrices. We first consider general matrix completion setting where we assume that for each matrix its entries are sampled from natural exponential distributions (Lehmann and Casella, 1998). In this setting, we may have Gaussian distribution for continuous data; Bernoulli for binary data; Poisson for count-data, etc. In a second part, we relax the assumption of exponential family distribution for the noise and we do not assume any specific model for the observations. This approach is more popular and widely used in machine learning. The proposed estimation procedure is based on minimizing the sum of a goodness-of-fit term and the nuclear norm penalization of the whole collective matrix. The key challenge in our analysis is to use joint low-rank structure and our algorithm is far from the trivial one which consists in estimating each source matrix separately. We provide theoretical guarantees on our estimation method and show that the collective approach provides faster rate of convergences. We further corroborate our theoretical findings through simulated experiments.

Previous works on collective matrix completion are mainly based on matrix factorization (Srebro et al., 2005). In a nutshell, this approach fits the target matrix as the product of two low-rank matrices. Matrix factorization gives rise to non-convex optimization problems and its theoretical understanding is quite limited. For example, Singh and Gordon (2008) proposed the collective matrix factorization that jointly factorizes multiple matrices sharing latent factors. As in our setting, each matrix can have a different value type and error distribution. In Singh and Gordon (2008), the authors use Bregman divergences to measure the error and extend standard alternating projection algorithms to this setting. They consider a quite general setting which includes as a particular case the nuclear norm penalization approach that we study in the present paper. They do not provide any theoretical guarantee. A Bayesian model for collective matrix factorization was proposed in Singh and Gordon (2010). Horii et al. (2014) and Xu et al. (2016) also consider collective matrix factorization and investigate the strength of the relation among the source matrices. Their estimation procedure is based on penalization by the sum of the nuclear norms of the sources. The convex formulation for collective matrix factorization was proposed in Bouchard et al. (2013) where the authors consider a general situation when the set of matrices do not necessarily have a common set of rows/columns. When this is the case, the estimator proposed in Bouchard et al. (2013) is quite similar to ours. Their algorithm is based on the iterative Singular Value Thresholding and the authors conduct empirical evaluations of this approach on two real data sets.

Most of the previous papers focus on the algorithmic side without providing theoretical guarantees for the collective approach. One exception is the paper by Gunasekar et al. (2015) where the authors prove consistency of the estimate under two observation models: noise-free and additive noise models. Their estimation procedure is based on minimizing the least squares loss penalized by the nuclear norm. To prove the consistency of their estimator, Gunasekar et al. (2015) assume that all the source matrices share the same low-rank factor. They consider the uniform sampling scheme for the observations (see Assumptions 1 and 4 in Gunasekar et al. (2015)). Uniform sampling is an usual assumption in matrix completion literature (see, e.g., (Candes and Tao, 2010; Candès and Recht, 2009; Davenport et al., 2014)). This assumption is restrictive in many applications such as recommendations

systems. The theoretical analysis in the present paper is carried out for general sampling distributions.

Similar to our setting, matrix completion with side information explores the available user data provided by various sources. For instance Jain and Dhillon (2013) and Xu et al. (2013) introduce the so-called Inductive Matrix Completion (IMC). It models side information as knowledge of feature spaces. They show that if the features are perfect (e.,g., see Definition 1 in Chiang et al. (2018) for perfect side information), the sample complexity can be reduced. More precisely, in works on matrix completion with side information, it is usually assumed that one has partially observed low-rank matrix of interest $\boldsymbol{M} \in \mathbb{R}^{d_1 \times d_2}$ and, additionally, one has access to two matrices of features $\boldsymbol{A} \in \mathbb{R}^{d_1 \times r_1}$ and $\boldsymbol{B} \in \mathbb{R}^{d_2 \times r_2}$ where each row of $\boldsymbol{A}$ (or $\boldsymbol{B}$) denotes the feature of the i-th row (or column) entity of $\boldsymbol{M}$, $r_i < d_i$ for $i = 1, 2$ and $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^T$. The main difference with our setting is that, here, $\boldsymbol{A}$ and $\boldsymbol{B}$ are assumed to be fully observed while our model allows also missing observations for the set of features. The perfect side information assumption is strong and hard to meet in practice. Chiang et al. (2015) relaxed it by assuming that the side information may be noisy (not perfect). In this approach, referred as DirtyIMC, they assume that the unknown matrix is modeled as $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^T + \boldsymbol{N}$ where the residual matrix $\boldsymbol{N}$ models imperfections and noise in the features.

Several works consider matrix completion with side information. For example, Chiang et al. (2015) proposes a method based on penalization by the sum of the nuclear norms of $\boldsymbol{M}$ and of each feature. Our method is based on the penalization by the nuclear norm of the whole matrix built of the matrix $\boldsymbol{M}$ and the features $\boldsymbol{A}$ and $\boldsymbol{B}$. In Jain and Dhillon (2013), the authors study the problem of low-rank matrix estimation using rank one measurements. In the noise-free setting, they assume that all the features are known and that the matrices of features are incoherent. The method proposed in Jain and Dhillon (2013) is based on non-convex matrix factorization. In Fithian and Mazumder (2018), the authors consider a general framework for reduced-rank modeling of matrix-valued data. They use a generalized weighted nuclear norm penalty where the matrix is multiplied by positive semidefinite matrices P and Q which depend on the matrix of features. In Agarwal et al. (2011), the authors introduce a per-item user covariate logistic regression model augmenting with user-specific random effects. Their approach is based on a multilevel hierarchical model.

In the case of the heterogeneous data coming from different sources, these approaches can be applied for recovering each source separately. In contrast, our approach aims at collecting all the available information in a single matrix which results in faster rates of convergence. On the other hand, popular algorithms for matrix completion with side information, such as Maxide in Xu et al. (2013) and AltMin in Jain and Dhillon (2013), are based on the least square loss which could be not suitable for data coming from non-Gaussian distributions.

If we consider a single matrix, our model includes as particular case 1-bit matrix completion and, more generally, matrix completion with exponential family noise. 1-bit matrix completion was first studied in Davenport et al. (2014), where the observed entries are assumed to be sampled uniformly at random. This problem was also studied among others by (Cai and Zhou, 2013; Klopp et al., 2015; Alquier et al., 2017). Matrix completion with exponential family noise (for a single matrix) was previously considered in Lafond (2015) and Gunasekar et al. (2014). In these papers authors assume sampling with replacement

where there can be multiple observations for the same entry. In the present paper, we consider more natural setting for matrix completion where each entry may be observed at most once. Our result improves the known results on 1-bit matrix completion and on matrix completion with exponential family noise. In particular, we obtain exact minimax optimal rate of convergence for 1-bit matrix completion and matrix completion with exponential noise which was known up to a logarithmic factor (for more details see Remark 9 in Section 3).

Organization of the paper. The remainder of the paper is organized as follow. In Section 1.1, we introduce basic notation and definitions. Section 2 sets up the formalism for the collective matrix completion. In Section 3, we investigate the exponential family noise model. In Section 4, we study distribution-free setup and we provide the upper bound on the excess risk. To verify the theoretical findings, we corroborate our results with numerical experiments in Section 5, where we present an efficient iterative algorithm that solves the maximum likelihood approximately. The proofs of the main results and key technical lemmas are postponed to the appendices.

1.1. Preliminaries

For the reader's convenience, we provide a brief summary of the standard notation and the definitions that will be frequently used throughout the paper.

Notation. For any positive integer m, we use $[m]$ to denote $\{1, \dots, m\}$. We use capital bold symbols such as $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{A}$, to denote matrices. For a matrix $\boldsymbol{A}$, we denote its (i,j)-th entry by A_{ij}. As usual, let $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$ be the Frobenius norm and let $\|\boldsymbol{A}\|_\infty = \max_{i,j} |A_{ij}|$ denote the elementwise ℓ_∞-norm. Additionally, $\|\boldsymbol{A}\|_*$ stands for the nuclear norm (trace norm), that is $\|\boldsymbol{A}\|_* = \sum_i \sigma_i(\boldsymbol{A})$ where $\sigma_1(\boldsymbol{A}) \geq \sigma_2(\boldsymbol{A}) \geq \cdots$ are singular values of $\boldsymbol{A}$, and $\|\boldsymbol{A}\| = \sigma_1(\boldsymbol{A})$ to denote the operator norm. The inner product between two matrices is denoted by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathrm{tr}(\boldsymbol{A}^\top \boldsymbol{B}) = \sum_{ij} A_{ij} B_{ij}$, where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. We write $\partial\Psi$ the subdifferential mapping of a convex functional Ψ. Given two real numbers a and b, we write $a \vee b = \max(a,b)$ and $a \wedge b = \min(a,b)$. The symbols $\mathbb{P}$ and $\mathbb{E}$ denote generic probability and expectation operators whose distribution is determined from the context. The notation $\boldsymbol{c}$ will be used to denote positive constant, that might change from one instance to the other.

Definition 1 *A distribution of a random variable X is said to belong to the natural exponential family, if its probability density function characterized by the parameter η is given by:*

$$X|\eta \sim f_{h,G}(x|\eta) = h(x) \exp\big(\eta x - G(\eta)\big),$$

where h is a nonnegative function, called the base measure function, which is independent of the parameter η. The function $G(\eta)$ is strictly convex, and is called the log-partition *function, or the cumulant function. This function uniquely defines a particular member distribution of the exponential family, and can be computed as: $G(\eta) = \log\big(\int h(x)\exp(\eta x) dx\big)$.*

If G is smooth enough, we have that $\mathbb{E}[X] = G'(\eta)$ and $\mathbb{V}ar[X] = G''(\eta)$, where G' stands for the derivative of G. The exponential family encompasses a wide large of standard distributions such as:

- Normal, $\mathcal{N}(\mu, \sigma^2)$ (known σ), is typically used to model continuous data, with natural parameter $\eta = \frac{\mu}{\sigma^2}$ and $G(\eta) = \frac{\sigma^2}{2}\eta^2$.
- Gamma, $\Gamma(\lambda, \alpha)$ (known α), is often used to model positive valued continuous data, with natural parameter $\eta = -\lambda$ and $G(\eta) = -\alpha \log(-\eta)$.
- Negative binomial, $\mathcal{NB}(p, r)$ (known r), is a popular distribution to model overdispersed count data, whose variance is larger than their mean, with natural parameter $\eta = \log(1-p)$ and $G(\eta) = -r\log(1-\exp(\eta))$.
- Binomial, $\mathcal{B}(p, N)$ (known N), is used to model number of successes in N trials, with natural parameter $\eta = \log(\frac{p}{1-p})$ (logit function) and $G(\eta) = N\log(1+\exp(\eta))$.
- Poisson, $\mathcal{P}(\lambda)$, is used to model count data, with natural parameter $\eta = \log(\lambda)$ and $G(\eta) = \exp(\eta)$.

Exponential, chi-squared, Rayleigh, Bernoulli and geometric distributions are special cases of the above five distributions.

Definition 2 *Let S be a closed convex subset of $\mathbb{R}^m$ and $\Phi : S \subset \boldsymbol{dom}(\Phi) \to \mathbb{R}$ a continuously-differentiable and strictly convex function. The Bregman divergence associated with Φ (Bregman, 1967; Censor and Zenios, 1997) $d_\Phi : S \times S \to [0, \infty)$ is defined as*

$$d_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle x - y, \nabla\Phi(y) \rangle,$$

where $\nabla\Phi(y)$ represents the gradient vector of Φ evaluated at y.

The value of the Bregman divergence $d_\Phi(x, y)$ can be viewed as the difference between the value of Φ at x and the first Taylor expansion of Φ around y evaluated at point x. For exponential family distributions, the Bregman divergence corresponds to the Kullback-Leibler divergence (Banerjee et al., 2005) with $\Phi = G$.

2. Collective matrix completion

Assume that we observe a collection of matrices $\boldsymbol{\mathcal{X}} = (\boldsymbol{X}^1, \dots, \boldsymbol{X}^V)$. In this collection components $\boldsymbol{X}^v \in \mathbb{R}^{d_u \times d_v}$ have a common set of rows. This common set of rows corresponds, for example, to a common set of users in a recommendation system. The set of columns of each matrix $\boldsymbol{X}^v$ corresponds to a different type of entity. In the case of recommender system it can be books, films, video game, etc. Then, the entries of each matrix $\boldsymbol{X}^v$ corresponds to the user's rankings for this particular type of products.

We assume that the distribution of each matrix $\boldsymbol{X}^v$ depends on the matrix of parameters $\boldsymbol{M}^v$. This distribution can be different for different v. For instance, we can have binary observations for one matrix $\boldsymbol{X}^{v_1}$ with entries which correspond, for example, to like/dislike labels for a certain type of products, multinomial for another matrix $\boldsymbol{X}^{v_2}$ with ranking going from 1 to 5 and Gaussian for a third matrix $\boldsymbol{X}^{v_3}$.

As it happens in many applications, we assume that for each matrix $\boldsymbol{X}^v$ we observe only a small subset of its entries. We consider the following model: for $v \in [V]$ and $(i,j) \in [d_u] \times [d_v]$, let B_{ij}^v be independent Bernoulli random variables with parameter π_{ij}^v.

We suppose that B^v_{ij} are independent from X^v_{ij}. Then, we observe $Y^v_{ij} = B^v_{ij} X^v_{ij}$. We can think of the B^v_{ij} as masked variables. If $B^v_{ij} = 1$, we observe the corresponding entry of $\boldsymbol{X}^v$, and when $B^v_{ij} = 0$, we have a missing observation.

In the simplest situation each coefficient is observed with the same probability, i.e. for every $v \in [V]$ and $(i,j) \in [d_u] \times [d_v], \pi^v_{ij} = \pi$. In many practical applications, this assumption is not realistic. For example, for a recommendation system, some users are more active than others and some items are more popular than others and thus rated more frequently. Hence, the sampling distribution is in fact non-uniform. In the present paper, we consider general sampling model where we only assume that each entry is observed with a positive probability:

Assumption 1 *Assume that there exists a positive constant $0 < p < 1$ such that*

$$\min_{v\in[V]} \min_{(i,j)\in[d_u]\times[d_v]} \pi^v_{ij} \geq p.$$

Let Π denotes the joint distribution of the Bernoulli variables $\{B^v_{ij} : (i,j) \in [d_u] \times [d_v], v \in [V]\}$. For any matrix $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_u \times D}$ where $D = \sum_{v\in[V]} d_v$, we define the weighted Frobenius norm

$$\|\boldsymbol{\mathcal{A}}\|^2_{\Pi,F} = \sum_{v\in[V]} \sum_{(i,j)\in[d_u]\times[d_v]} \pi^v_{ij} (A^v_{ij})^2.$$

Assumption 1 implies $\|\boldsymbol{\mathcal{A}}\|^2_{\Pi,F} \geq p\|\boldsymbol{\mathcal{A}}\|^2_F$. For each $v \in [V]$ let us denote $\pi^v_{i\bullet} = \sum_{j=1}^{d_v} \pi^v_{ij}$ and $\pi^v_{\bullet j} = \sum_{i=1}^{d_u} \pi^v_{ij}$. Note we can easily get an estimations of $\pi^v_{i\bullet}$ and $\pi^v_{\bullet j}$ using the empirical frequencies:

$$\widehat{\pi^v_{i\bullet}} = \sum_{j\in[d_v]} B^v_{ij} \quad \text{and} \quad \widehat{\pi^v_{\bullet j}} = \sum_{i\in[d_u]} B^v_{ij}.$$

Let $\pi_{i\bullet} = \sum_{v\in[V]} \pi^v_{i\bullet}$, $\pi_{\bullet j} = \max_{v\in[V]} \pi^v_{\bullet j}$, and μ be an upper bound of its maximum, that is

$$\max_{(i,j)\in[d_u]\times[d_v]} (\pi_{i\bullet}, \pi_{\bullet j}) \leq \mu. \tag{1}$$

3. Exponential family noise

In this section we assume that for each v distribution of $\boldsymbol{X}^v$ belongs to the exponential family, that is

$$X^v_{ij} | M^v_{ij} \sim f_{h^v,G^v}(X^v_{ij}|M^v_{ij}) = h^v(X^v_{ij}) \exp\left(X^v_{ij} M^v_{ij} - G^v(M^v_{ij})\right).$$

We denote $\boldsymbol{\mathcal{M}} = (\boldsymbol{M}^1, \ldots, \boldsymbol{M}^V)$ and let γ be an upper bound on the sup-norm of $\boldsymbol{\mathcal{M}}$, that is $\gamma = |\gamma_1| \vee |\gamma_2|$, where $\gamma_1 \leq M^v_{ij} \leq \gamma_2$ for every $v \in [V]$ and $(i,j) \in [d_u] \times [d_v]$. Hereafter, we denote by $\mathscr{C}_\infty(\gamma) = \{\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d_u \times D} : \|\boldsymbol{\mathcal{W}}\|_\infty \leq \gamma\}$, the ℓ_∞-norm ball with radius γ in the space $\mathbb{R}^{d_u \times D}$. We need the following assumptions on densities f_{h^v,G^v}:

Assumption 2 *For each $v \in [V]$, we assume that the function $G^v(\cdot)$ is twice differentiable and there exits two constants L^2_γ, U^2_γ satisfying:*

$$\sup_{\eta\in[-\gamma-\frac{1}{K},\gamma+\frac{1}{K}]} (G^v)''(\eta) \leq U^2_\gamma, \tag{2}$$

and

$$\inf_{\eta\in[-\gamma-\frac{1}{K},\gamma+\frac{1}{K}]} (G^v)''(\eta) \geq L^2_\gamma, \tag{3}$$

for some $K > 0$.

The first statement, (2), in Assumption 2 ensures that the distributions of X^v_{ij} have uniformly bounded variances and sub-exponential tails (see Lemma 28 in Appendix C). The second one, (3), is the strong convexity condition satisfied by the log-partition function G^v. This assumption is satisfied for most standard distributions presented in the previous section. In Table 1, we list the corresponding constants in Assumption 2.

Model	$(G^v)'(\eta)$	$(G^v)''(\eta)$	L^2_γ	U^2_γ
Normal	$\sigma^2\eta$	σ^2	σ^2	σ^2
Binomial	$\frac{Ne^\eta}{1+e^\eta}$	$\frac{Ne^\eta}{(1+e^\eta)^2}$	$\frac{Ne^{-(\gamma+\frac{1}{K})}}{(1+e^{\gamma+\frac{1}{K}})^2}$	$\frac{N}{4}$
Gamma (if $\gamma_1\gamma_2 > 0$)	$-\frac{\alpha}{\eta}$	$\frac{\alpha}{\eta^2}$	$\frac{\alpha}{(\gamma+\frac{1}{K})^2}$	$\frac{\alpha}{(\lvert\gamma_1\rvert\wedge\lvert\gamma_2\rvert)^2}$
Negative binomial	$\frac{re^\eta}{1-e^\eta}$	$\frac{re^\eta}{(1-e^\eta)^2}$	$\frac{re^{-(\gamma+\frac{1}{K})}}{(1-e^{-(\gamma+\frac{1}{K})})^2}$	$\frac{re^{(\gamma+\frac{1}{K})}}{(1-e^{\gamma+\frac{1}{K}})^2}$
Poisson	e^η	e^η	$e^{-(\gamma+\frac{1}{K})}$	$e^{(\gamma+\frac{1}{K})}$

Table 1: Examples of the corresponding constants L^2_γ and U^2_γ from Assumption 2.

3.1. Estimation procedure

To estimate the collection of matrices of parameters $\boldsymbol{\mathcal{M}} = (\boldsymbol{M}^1, \ldots, \boldsymbol{M}^V)$, we use penalized negative log-likelihood. Let $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d_u \times D}$, we divide it in V blocks $\boldsymbol{W}^v \in \mathbb{R}^{d_u \times d_v}$: $\boldsymbol{\mathcal{W}} = (\boldsymbol{W}^1, \ldots, \boldsymbol{W}^V)$. Given observations $\boldsymbol{\mathcal{Y}} = (\boldsymbol{Y}^1, \ldots, \boldsymbol{Y}^V)$, we write the negative log-likelihood as

$$\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{W}}) = -\frac{1}{d_u D} \sum_{v\in[V]} \sum_{(i,j)\in[d_u]\times[d_v]} B^v_{ij}\big(Y^v_{ij} W^v_{ij} - G^v(W^v_{ij})\big).$$

The nuclear norm penalized estimator $\widehat{\boldsymbol{\mathcal{M}}}$ of $\boldsymbol{\mathcal{M}}$ is defined as follows:

$$\widehat{\boldsymbol{\mathcal{M}}} = (\widehat{\boldsymbol{M}}^1, \ldots, \widehat{\boldsymbol{M}}^V) = \underset{\boldsymbol{\mathcal{W}}\in\mathscr{C}_\infty(\gamma)}{\operatorname{argmin}} \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{W}}) + \lambda\|\boldsymbol{\mathcal{W}}\|_*, \tag{4}$$

where λ is a positive regularization parameter that balances the trade-off between model fit and privileging a low-rank solution. Namely, for large value of λ the rank of the estimator $\widehat{\boldsymbol{\mathcal{M}}}$ is expected to be small.

Let the collection of matrices $(E^v_{11}, \ldots, E^v_{d_u d_v})$ form the canonical basis in the space of matrices of size $d_u \times d_v$. The entry of (E^v_{ij}) is 0 everywhere except for the (i,j)-th entry

where it equals to 1. For (ε_{ij}^v), an *i.i.d* Rademacher sequence, we define $\mathbf{\Sigma}_R = (\mathbf{\Sigma}_R^1, \dots, \mathbf{\Sigma}_R^V)$ where for all $v \in [V]$

$$\mathbf{\Sigma}_R^v = \frac{1}{d_u D} \sum_{(i,j)\in[d_u]\times[d_v]} \varepsilon_{ij}^v B_{ij}^v E_{ij}^v.$$

We now state the main result concerning the recovery of $\boldsymbol{\mathcal{M}}$. Theorem 3 gives a general upper bound on the estimation error of $\widehat{\boldsymbol{\mathcal{M}}}$ defined by (4). Its proof is postponed in Appendix A.1.

Theorem 3 *Assume that Assumptions 1 and 2 hold, and $\lambda \geq 2\|\nabla \mathscr{L}_{\mathbf{y}}(\boldsymbol{\mathcal{M}})\|$. Then, with probability exceeding $1 - 4/(d_u + D)$ we have*

$$\frac{1}{d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\mathrm{II},F}^2 \leq \frac{\mathcal{C}}{p} \max\Big\{ d_u D \operatorname{rank}(\boldsymbol{\mathcal{M}}) \Big(\frac{\lambda^2}{L_\gamma^4} + \gamma^2 (\mathbb{E}[\|\mathbf{\Sigma}_R\|])^2 \Big), \frac{\gamma^2 \log(d_u + D)}{d_u D} \Big\},$$

where $\mathcal{C}$ is a numerical constant.

Using Assumption 1, Theorem 3 implies the following bound on the estimation error measured in normalized Frobenius norm.

Corollary 4 *Under assumptions of Theorem 3 and with probability exceeding $1-4/(d_u+D)$, we have*

$$\frac{1}{d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F^2 \leq \frac{\mathcal{C}}{p^2} \max\Big\{ d_u D \operatorname{rank}(\boldsymbol{\mathcal{M}}) \Big(\frac{\lambda^2}{L_\gamma^4} + \gamma^2 (\mathbb{E}[\|\mathbf{\Sigma}_R\|])^2 \Big), \frac{\gamma^2 \log(d_u + D)}{d_u D} \Big\}.$$

In order to get a bound in a closed form we need to obtain a suitable upper bounds on $\mathbb{E}[\|\mathbf{\Sigma}_R\|]$ and on $\|\nabla \mathscr{L}_{\mathbf{y}}(\boldsymbol{\mathcal{M}})\|$ with high probability. Therefore we use the following two lemmas.

Lemma 5 *There exists an absolute constant $\mathcal{C}$ such that*

$$\mathbb{E}[\|\mathbf{\Sigma}_R\|] \leq \mathcal{C}\Big(\frac{\sqrt{\mu} + \sqrt{\log(d_u \wedge D)}}{d_u D} \Big).$$

Lemma 6 *Let Assumption 2 holds. Then, there exists an absolute constant $\mathcal{C}$ such that, with probability at least $1 - 4/(d_u + D)$, we have*

$$\|\nabla \mathscr{L}_{\mathbf{y}}(\boldsymbol{\mathcal{M}})\| \leq \mathcal{C}\Big(\frac{(U_\gamma \vee K)\big(\sqrt{\mu} + (\log(d_u \vee D))^{3/2}\big)}{d_u D} \Big).$$

The proofs of Lemmas 5 and 6 are postponed to Appendices A.2 and A.3. Recall that the condition on λ in Theorem 3 is that $\lambda \geq 2\|\nabla \mathscr{L}_{\mathbf{y}}(\boldsymbol{\mathcal{M}})\|$. Using Lemma 6, we can choose

$$\lambda = 2\mathcal{C} \frac{(U_\gamma \vee K)\big(\sqrt{\mu} + (\log(d_u \vee D))^{3/2}\big)}{d_u D}.$$

With this choice of λ, we obtain the following theorem:

Theorem 7 *Let Assumptions 1 and 2 be satisfied. Then, with probability exceeding $1 - 4/(d_u + D)$ we have*

$$\frac{1}{d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\mathrm{II},F}^2 \leq \frac{\mathcal{C} \operatorname{rank}(\boldsymbol{\mathcal{M}})}{p d_u D} \Big(\gamma^2 + \frac{(U_\gamma \vee K)^2}{L_\gamma^4} \Big) \big(\mu + \log^3(d_u \vee D) \big),$$

and

$$\frac{1}{d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F^2 \leq \frac{\mathcal{C} \operatorname{rank}(\boldsymbol{\mathcal{M}})}{p^2 d_u D} \Big(\gamma^2 + \frac{(U_\gamma \vee K)^2}{L_\gamma^4} \Big) \big(\mu + \log^3(d_u \vee D) \big),$$

where $\mathcal{C}$ is an absolute constant.

Remark 8 *Note that the rate of convergence in Theorem 7 has the following dominant term:*

$$\frac{1}{d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F^2 \lesssim \frac{\operatorname{rank}(\boldsymbol{\mathcal{M}})\mu}{p^2 d_u D},$$

where the symbol $\lesssim$ means that the inequality holds up to a multiplicative constant. If we assume that the sampling distribution is close to the uniform one, that is that there exists positive constants c_1 and c_2 such that for every $v \in [V]$ and $(i,j) \in [d_u] \times [d_v]$ we have $c_1 p \leq \pi_{ij}^v \leq c_2 p$, then Theorem 7 yields

$$\frac{1}{d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F^2 \lesssim \frac{\operatorname{rank}(\boldsymbol{\mathcal{M}})}{p(d_u \wedge D)}.$$

If we complete each matrix separately, the error will be of the order $\sum_{v=1}^V \operatorname{rank}(\boldsymbol{M}^v)/p(d_u \wedge D)$. As $\operatorname{rank}(\boldsymbol{\mathcal{M}}) \leq \sum_{v=1}^V \operatorname{rank}(\boldsymbol{M}^v)$, the rate of convergence achieved by our estimator is faster compared to the penalization by the sum-nuclear-norm.

In order to get a small estimation error, p should be larger than $\operatorname{rank}(\boldsymbol{\mathcal{M}})/(d_u \wedge D)$. We denote $n = \sum_{v\in[V]} \sum_{(i,j)\in[d_u]\times[d_v]} \pi_{ij}^v$, the expected number of observations. Then, we get the following condition on n:

$$n \geq \mathcal{C} \operatorname{rank}(\boldsymbol{\mathcal{M}})(d_u \vee D).$$

Remark 9 *In 1-bit matrix completion (Davenport et al., 2014; Klopp et al., 2015; Alquier et al., 2017), instead of observing the actual entries of the unknown matrix $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{d\times D}$, for a random subset of its entries Ω we observe $\{Y_{ij} \in \{+1, -1\} : (i,j) \in \Omega\}$, where $Y_{ij} = 1$ with probability $f(M_{ij})$ for some link-function f. In Davenport et al. (2014) the parameter $\boldsymbol{\mathcal{M}}$ is estimated by minimizing the negative log-likelihood under the constraints $\|\boldsymbol{\mathcal{M}}\|_\infty \leq \gamma$ and $\|\boldsymbol{\mathcal{M}}\|_* \leq \gamma\sqrt{rdD}$ for some $r > 0$. Under the assumption that $\operatorname{rank}(\boldsymbol{\mathcal{M}}) \leq r$, the authors prove that*

$$\frac{1}{dD}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F^2 \leq \mathcal{C}_\gamma \sqrt{\frac{r(d \vee D)}{n}}, \tag{5}$$

where $\mathcal{C}_\gamma$ is a constant depending on γ (see Theorem 1 in Davenport et al. (2014)). A similar result using max-norm minimization was obtained in Cai and Zhou (2013). In (Klopp et al.,

2015) the authors prove a faster rate. Their upper bound (see Corollary 2 in Klopp et al. (2015)) is given by

$$\frac{1}{dD}\|\widehat{\boldsymbol{\mathcal{M}}}-\boldsymbol{\mathcal{M}}\|_F^2 \le c_\gamma \frac{\mathrm{rank}(\boldsymbol{\mathcal{M}})(d\vee D)\log(d\vee D)}{n}. \tag{6}$$

In the particular case of 1*-bit matrix completion for a single matrix under uniform sampling scheme, Theorem 7 implies the following bound:*

$$\frac{1}{dD}\|\widehat{\boldsymbol{\mathcal{M}}}-\boldsymbol{\mathcal{M}}\|_F^2 \le c_\gamma \frac{\mathrm{rank}(\boldsymbol{\mathcal{M}})(d\vee D)}{n},$$

which improves (6) *by a logarithmic factor. Furthermore, Klopp et al. (2015) provide* $\mathrm{rank}(M)(d\vee D)/n$ *as the lower bound for* 1*-bit matrix completion (see Theorem 3 in Klopp et al. (2015)). So our result answers the important theoretical question what is the exact minimax rate of convergence for* 1*-bit matrix completion which was previously known up to a logarithmic factor.*

In a more general setting of matrix completion with exponential family noise, the minimax optimal rate of convergence was also known only up to logarithmic factor (see Lafond (2015)). Our result provides the exact minimax optimal rate in this more general setting too. It is easy to see, by inspection of the proof of the lower bound in Lafond (2015), that the upper bound provided by Theorem 7 is optimal for the collective matrix completion.

Remark 10 *Note that our estimation method is based on the minimization of the nuclear-norm of the whole collective matrix* $\boldsymbol{\mathcal{M}}$*. Another possibility is to penalize by the sum of the nuclear norms* $\sum_{v\in[V]}\|\boldsymbol{M}^v\|_*$ *(see, e.g., Klopp et al. (2015)). This approach consists in estimating each component matrix independently.*

4. General losses

In the previous section we assume that the link functions G^v are known. This assumption is not realistic in many applications. In this section we relax this assumption in the sense that we do not assume any specific model for the observations. Recall that our observations are a collection of partially observed matrices $\boldsymbol{Y}^v = (B^v_{ij}X^v_{i,j}) \in \mathbb{R}^{d_u\times d_v}$ for $v = 1,\dots,V$ and $\boldsymbol{X}^v = (X^v_{ij}) \in \mathbb{R}^{d_u\times d_v}$. We are interested in the problem of prediction of the entries of the collective matrix $\boldsymbol{\mathcal{X}} = (\boldsymbol{X}^1,\dots,\boldsymbol{X}^V)$. We consider the risk of estimating $\boldsymbol{X}^v$ with a loss function ℓ^v, which measures the discrepancy between the predicted and actual value with respect to the given observations. We focus on non-negative convex loss functions that are Lipschitz:

Assumption 3 *(Lipschitz loss function) For every* $v\in[V]$*, we assume that the loss function* $\ell^v(y,\cdot)$ *is* ρ_v*-Lipschitz in its second argument:* $|\ell^v(y,x)-\ell^v(y,x')|\le\rho_v|x-x'|$.

Some examples of the loss functions that are 1-Lipschitz are: hinge loss $\ell(y,y') = \max(0,1-yy')$, logistic loss $\ell(y,y') = \log(1+\exp(-yy'))$, and quantile regression loss $\ell(y,y') = \ell_\tau(y'-y)$ where $\tau\in(0,1)$ and $\ell_\tau(z) = z(\tau-\mathbb{1}(z\le 0))$.

For a matrix $\boldsymbol{\mathcal{M}} = (\boldsymbol{M}^1,\dots,\boldsymbol{M}^V)\in\mathbb{R}^{d_u\times D}$, we define the empirical risk as

$$R_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}}) = \frac{1}{d_uD}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B^v_{ij}\ell^v(Y^v_{ij},M^v_{ij}).$$

We define the oracle as:

$$\overset{\star}{\boldsymbol{\mathcal{M}}} = (\overset{\star}{\boldsymbol{M}}{}^1,\dots,\overset{\star}{\boldsymbol{M}}{}^V) = \underset{\boldsymbol{\mathcal{Q}}\in\mathscr{C}_\infty(\gamma)}{\mathrm{argmin}}\ R(\boldsymbol{\mathcal{Q}}) \tag{7}$$

where $R(\boldsymbol{\mathcal{Q}}) = \mathbb{E}[R_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Q}})]$. Here the expectation is taken over the joint distribution of $\{(Y^v_{ij},B^v_{ij}) : (i,j)\in[d_u]\times[d_v] \text{ and } v\in[V]\}$. We use machine learning approach and will provide an estimator $\widehat{\boldsymbol{\mathcal{M}}}$ that predicts almost as well as $\overset{\star}{\boldsymbol{\mathcal{M}}}$. Thus we will consider excess risk $R(\widehat{\boldsymbol{\mathcal{M}}})-R(\overset{\star}{\boldsymbol{\mathcal{M}}})$. By construction, the excess risk is always positive.

For a tuning parameter $\Lambda>0$, the nuclear norm penalized estimator $\widehat{\boldsymbol{\mathcal{M}}}$ is defined as

$$\widehat{\boldsymbol{\mathcal{M}}} \in \underset{\boldsymbol{\mathcal{Q}}\in\mathscr{C}_\infty(\gamma)}{\mathrm{argmin}}\ \big\{R_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Q}})+\Lambda\|\boldsymbol{\mathcal{Q}}\|_*\big\}. \tag{8}$$

We next turn to the assumption needed to establish an upper bound on the performance of the estimator $\widehat{\boldsymbol{\mathcal{M}}}$ defined in (8).

Assumption 4 *Assume that there exists a constant* $\varsigma>0$ *such that for every* $\boldsymbol{\mathcal{Q}}\in\mathscr{C}_\infty(\gamma)$*, we have*

$$R(\boldsymbol{\mathcal{Q}})-R(\overset{\star}{\boldsymbol{\mathcal{M}}}) \ge \frac{\varsigma}{d_uD}\|\boldsymbol{\mathcal{Q}}-\overset{\star}{\boldsymbol{\mathcal{M}}}\|^2_{\Pi,F}.$$

This assumption has been extensively studied in the learning theory literature (Mendelson, 2008; Zhang, 2004; Bartlett et al., 2004; Alquier et al., 2017; Elsener and van de Geer, 2018), and it is called "Bernstein" condition. It is satisfied in various cases of loss function (Alquier et al., 2017) and it ensures a sufficient convexity of the risk around the oracle defined in (7). Note that when the loss function ℓ^v is strongly convex, the risk function inherits this property and automatically satisfies the margin condition. In other cases, this condition requires strong assumptions on the distribution of the observations, for instance for hinge loss or quantile loss (see Section 6 in Alquier et al. (2017)). The following result gives an upper bound on the excess risk of the estimator $\widehat{\boldsymbol{\mathcal{M}}}$.

Theorem 11 *Let Assumptions 1, 3 and 4 hold and set* $\rho = \max_{v\in[V]}\rho_v$*. Suppose that* $\Lambda\ge 2\sup\{\|\boldsymbol{\mathcal{G}}\| : \boldsymbol{\mathcal{G}}\in\partial R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\}$*. Then, with probability at least* $1-4/(d_u+D)$*, we have*

$$R(\widehat{\boldsymbol{\mathcal{M}}})-R(\overset{\star}{\boldsymbol{\mathcal{M}}}) \le \frac{c}{p}\max\Big\{\mathrm{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})d_uD\Big(\rho^{3/2}\sqrt{\gamma/\varsigma}(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2+\frac{\Lambda^2}{\varsigma}\Big), \frac{\big(\rho\gamma+\rho^{3/2}\sqrt{\gamma/\varsigma}\big)\log(d_u+D)}{d_uD}\Big\}.$$

Theorem 11 gives a general upper bound on the prediction error of the estimator $\widehat{\boldsymbol{\mathcal{M}}}$. Its proof is presented in Appendix A.4. In order to get a bound in a closed form we need to obtain a suitable upper bounds on $\sup\{\|\boldsymbol{\mathcal{G}}\| : \boldsymbol{\mathcal{G}}\in\partial(R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}}))\}$ with high probability.

Lemma 12 *Let Assumption 3 holds. Then, there exists an absolute constant c such that, with probability at least $1-4/(d_u+D)$, we have*

$$\|\mathcal{G}\| \leq c\frac{\rho\big(\sqrt{\mu}+\sqrt{\log(d_u \vee D)}\big)}{d_u D},$$

for all $\mathcal{G} \in \partial R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})$.

The proof of Lemma 12 is given in Appendix A.5. Using Lemma 12 , we can choose

$$\Lambda = 2c\frac{\rho\big(\sqrt{\mu}+\sqrt{\log(d_u \vee D)}\big)}{d_u D}$$

and with this choice of Λ and Lemma 5, we obtain the following theorem:

Theorem 13 *Let Assumptions 1, 3 and 4 hold. Then, we have*

$$R(\widehat{\mathcal{M}}) - R(\overset{\star}{\mathcal{M}}) \leq \frac{c}{p}\operatorname{rank}(\overset{\star}{\mathcal{M}})\frac{(\rho^2+\rho^{3/2}\sqrt{\gamma/\varsigma})(\mu+\log(d_u \vee D))}{d_u D},$$

with probability at least $1-4/(d_u+D)$.

Using Assumption 4, we get the following corollary:

Corollary 14 *With probability at least $1-4/(d_u+D)$, we have*

$$\frac{1}{d_u D}\|\widehat{\mathcal{M}} - \overset{\star}{\mathcal{M}}\|_F^2 \leq \frac{c}{p^2\varsigma}\operatorname{rank}(\overset{\star}{\mathcal{M}})\frac{(\rho^2+\rho^{3/2}\sqrt{\gamma/\varsigma})(\mu+\log(d_u \vee D))}{d_u D}.$$

1-bit matrix completion. In 1-bit matrix completion with logistic (resp. hinge) loss, the Bernstein assumption is satisfied with $\varsigma = 1/(4e^{2\gamma})$ (resp. $\varsigma = 2\tau$, for some τ that verifies $|\overset{\star}{M}{}^v_{ij} - 1/2| \geq \tau, \forall v \in [V], (i,j) \in [d_u]\times[d_v]$). More details for these constants can be found in Propositions 6.1 and 6.3 in Alquier et al. (2017). Then, the excess risk with respect to these two losses under the uniform sampling is given by:

Corollary 15 *With probability at least $1-4/(d_u+D)$, we have*

$$R(\widehat{\mathcal{M}}) - R(\overset{\star}{\mathcal{M}}) \leq c\frac{\operatorname{rank}(\overset{\star}{\mathcal{M}})}{p(d_u \wedge D)}.$$

These results are obtained without a logarithmic factor, and it improves the ones given in Theorems 4.2 and 4.4 in Alquier et al. (2017). The natural loss in this context is the 0/1 loss which is often replaced by the hinge or the logistic loss. We assume without loss of generality that $\gamma = 1$, since the Bayes classifier has its entries in $[-1,1]$, and we define the classification excess risk by:

$$R_{0/1}(\mathcal{M}) = \frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} \pi^v_{ij}\mathbb{P}[X^v_{ij} \neq \operatorname{sign}(M^v_{ij})],$$

for all $\mathcal{M} \in \mathbb{R}^{d_u\times D}$. Using Theorem 2.1 in Zhang (2004), we have

$$R_{0/1}(\widehat{\mathcal{M}}) - R_{0/1}(\overset{\star}{\mathcal{M}}) \leq c\sqrt{\frac{\operatorname{rank}(\overset{\star}{\mathcal{M}})}{p(d_u \wedge D)}}.$$

5. Numerical experiments

In this section, we first provide algorithmic details of the numerical procedure for solving the problem (4), then we conduct experiments on synthetic data to further illustrate the theoretical results of the collective matrix completion.

5.1. Algorithm

The collective matrix completion problem (4) is a semidefinite program (SDP), since it is a nuclear norm minimization problem with a convex feasible domain (Fazel et al., 2001; Srebro et al., 2005). We may solve it, for example, via the interior-point method (Liu and Vandenberghe, 2010). However, SDP solvers can handle a moderate dimensions, thus such formulation is not scalable due to the storage and computation complexity in low-rank matrix completion tasks. In the following, we present an algorithm that solves the problem (4) approximately and in a more efficient way than solving it as SDP.

Proximal Gradient. Problem (4) can be solved by first-order optimization methods such as proximal gradient (PG) which has been popularly used for optimizations problems of the form of (4) (Beck and Teboulle, 2009; Nesterov, 2013; Parikh and Boyd, 2014; Ji and Ye, 2009b; Mazumder et al., 2010; Yao and Kwok, 2015). When $\mathscr{L}_{\mathcal{Y}}$ has L-Lipschitz continuous gradient, that is $\|\nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{W}) - \nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{Q})\|_F \leq L\|\mathcal{W}-\mathcal{Q}\|_F$, the PG generates a sequence of estimates $\{\mathcal{W}_t\}$ as

$$\begin{aligned}\mathcal{W}_{t+1} &= \operatorname*{argmin}_{\mathcal{W}} \mathscr{L}_{\mathcal{Y}}(\mathcal{W}) + (\mathcal{W}-\mathcal{W}_t)^\top\nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{W}_t) + \frac{L}{2}\|\mathcal{W}-\mathcal{W}_t\|_F^2 + \lambda\|\mathcal{W}\|_* \\ &= \operatorname{prox}_{\frac{\lambda}{L}\|\cdot\|_*}(\mathcal{Z}_t), \text{ where } \mathcal{Z}_t = \mathcal{W}_t - \frac{1}{L}\nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{W}_t) \end{aligned} \tag{9}$$

and for any convex function $\Psi : \mathbb{R}^{d_u\times D} \mapsto \mathbb{R}$, the associated proximal operator at $\mathcal{W} \in \mathbb{R}^{d_u\times D}$ is defined as

$$\operatorname{prox}_\Psi(\mathcal{W}) = \operatorname{argmin}\big\{\frac{1}{2}\|\mathcal{W}-\mathcal{Q}\|_F^2 + \Psi(\mathcal{Q}) : \mathcal{Q} \in \mathbb{R}^{d_u\times D}\big\}.$$

The proximal operator of the nuclear norm at $\mathcal{W} \in \mathbb{R}^{d_u\times D}$ corresponds to the singular value thresholding (SVT) operator of $\mathcal{W}$ (Cai. et al., 2010). That is, assuming a singular value decomposition $\mathcal{W} = \mathcal{U}\mathbf{\Sigma}\mathcal{V}^\top$, where $\mathcal{U} \in \mathbb{R}^{d_u\times r}$, $\mathcal{V} \in \mathbb{R}^{D\times r}$ have orthonormal columns, $\mathbf{\Sigma} = (\sigma_1,\ldots,\sigma_r)$, with $\sigma_1 \geq \cdots \geq \sigma_r > 0$ and $r = \operatorname{rank}(\mathcal{W})$, we have

$$\operatorname{SVT}_{\lambda/L}(\mathcal{W}) = \mathcal{U}\operatorname{diag}((\sigma_1-\lambda/L)_+,\ldots,(\sigma_r-\lambda/L)_+)\mathcal{V}^\top, \tag{10}$$

where $(a)_+ = \max(a,0)$.

Although PG can be implemented easily, it converges slowly when the Lipschitz constant L is large. In such scenarios, the rate is $\mathcal{O}(1/T)$, where T is the number of iterations (Parikh and Boyd, 2014). Nevertheless, it can be accelerated by replacing $\mathcal{Z}_t$ in (9) with

$$\mathcal{Q}_t = (1+\theta_t)\mathcal{W}_t - \theta_t\mathcal{W}_{t-1}, \quad \mathcal{Z}_t = \mathcal{Q}_t - \eta\nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{Q}_t). \tag{11}$$

Several choices for θ_t can be used. The resultant accelerated proximal gradient (APG) (see Algorithm 1) converges with the optimal $\mathcal{O}(1/T^2)$ rate (Nesterov, 2013; Ji and Ye, 2009a).

Algorithm 1: APG for Collective Matrix Completion

1. **initialize:** $\mathcal{W}_0 = \mathcal{W}_1 = \mathcal{Y}$, and $\alpha_0 = \alpha_1 = 1$.
2. **for** $t = 1, \ldots, T$ **do**
3. $\quad \mathcal{Q}_t = \mathcal{W}_t + \frac{\alpha_{t-1}-1}{\alpha_t}(\mathcal{W}_t - \mathcal{W}_{t-1})$;
4. $\quad \mathcal{W}_{t+1} = \mathrm{SVT}_{\frac{\lambda}{L}}(\mathcal{Q}_t - \frac{1}{L}\nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{Q}_t))$;
5. $\quad \alpha_{t+1} = \frac{1}{2}(\sqrt{4\alpha_t^2 + 1} + 1)$;
6. **return** $\mathcal{W}_{T+1}$.

Approximate SVT (Yao and Kwok, 2015). To compute $\mathcal{W}_{t+1}$ in the proximal step (SVT) in Algorithm 1, we need first perform SVD of $\mathcal{Z}_t$ given in (11). In general, obtaining the SVD of $d_u \times D$ matrix $\mathcal{Z}_t$ requires $\mathcal{O}((d_u \wedge D)d_u D)$ operations, because its most expensive steps are computing matrix-vector multiplications. Since the computation of the proximal operator of the nuclear norm given in (10) does not require to do the full SVD, only a few singular values of $\mathcal{Z}_t$ which are larger than λ/L are needed. Assume that there are $\hat{k}$ such singular values. As $\mathcal{W}_t$ converges to a low-rank solution $\mathcal{W}_*$, $\hat{k}$ will be small during iterating. The power method (Halko et al., 2011) at Algorithm 2 is a simple and efficient to capture subspace spanned by top-k singular vectors for $\hat{k} \geq k$. Additionally, the power method also allows warm-start, which is particularly useful because the iterative nature of APG algorithm. Once an approximation $\mathcal{Q}$ is found, we have $\mathrm{SVT}_{\lambda/L}(\mathcal{Z}_t) = \mathcal{Q}\mathrm{SVT}_{\lambda/L}(\mathcal{Q}^\top \mathcal{Z}_t)$ (see Proposition 3.1 in Yao and Kwok (2015)). We therefore reduce the time complexity on SVT from $\mathcal{O}((d_u \wedge D)d_u D)$ to $\mathcal{O}(\hat{k}d_u D)$ which is much cheaper.

Algorithm 2: Power Method: `PowerMethod`$(\mathcal{Z}, \mathcal{R}, \epsilon)$

1. **input:** $\mathcal{Z} \in \mathbb{R}^{d_u \times D}$, initial $\mathcal{R} \in \mathbb{R}^{D \times k}$ for warm-start, tolerance δ;
2. **initialize** $\mathcal{W}_1 = \mathcal{Z}\mathcal{R}$;
3. **for** $t = 1, 2, \ldots,$ **do**
4. $\quad \mathcal{Q}_{t+1} = \mathrm{QR}(\mathcal{W}_t)$;// QR denotes the QR factorization
5. $\quad \mathcal{W}_{t+1} = \mathcal{Z}(\mathcal{Z}^\top \mathcal{Q}_{t+1})$;
6. $\quad$ **if** $\|\mathcal{Q}_{t+1}\mathcal{Q}_{t+1}^\top - \mathcal{Q}_t\mathcal{Q}_t^\top\|_F \leq \delta$ **then**
$\quad\quad$ break;
7. **return** $\mathcal{Q}_{t+1}$.

Algorithm 3 shows how to approximate $\mathrm{SVT}_{\lambda/L}(\mathcal{Z}_t)$. Let the target (exact) rank-k SVD of $\mathcal{Z}_t$ be $\mathcal{U}_k\Sigma_k\mathcal{V}_k^\top$. Step 1 first approximates $\mathcal{U}_k$ by the power method. In steps 2 to 5, a less expensive $\mathrm{SVT}_{\lambda/L}(\mathcal{Q}^\top\mathcal{Z}_t)$ is obtained from (10). Finally, $\mathrm{SVT}_{\lambda/L}(\mathcal{Z}_t)$ is recovered.

Hereafter, we denote the objective function in (4) by $\mathscr{F}_\lambda(\mathcal{W})$, that is $\mathscr{F}_\lambda(\mathcal{W}) = \mathscr{L}_{\mathcal{Y}}(\mathcal{W}) + \lambda\|\mathcal{W}\|_*$, for any $\mathcal{W} \in \mathscr{C}(\gamma)$. Recall that the gradient of the likelihood $\mathscr{L}_{\mathcal{Y}}$ is written as

$$\nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{W}) = -\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v(Y_{ij}^v - (G^v)'(W_{ij}^v))E_{ij}^v.$$

Algorithm 3: Approximate SVT: `Approx-SVT`$(\mathcal{Z}, \mathcal{R}, \lambda, \delta)$

1. **input:** $\mathcal{Z} \in \mathbb{R}^{d_u \times D}, \mathcal{R} \in \mathbb{R}^{D \times k}$, thresholds λ and δ;
2. $\mathcal{Q} = \mathtt{PowerMethod}(\mathcal{Z}, \mathcal{R}, \delta)$;
3. $[\mathcal{U}, \Sigma, \mathcal{V}] = \mathrm{SVD}(\mathcal{Q}^\top\mathcal{Z})$;
4. $\mathcal{U} = \{u_i | \sigma_i > \lambda\}$;
5. $\mathcal{V} = \{v_i | \sigma_i > \lambda\}$;
6. $\Sigma = \max(\Sigma - \lambda\mathcal{I}, \mathbf{0})$; // ($\mathcal{I}$ denotes the identity matrix)
7. **return** $\mathcal{Q}\mathcal{U}, \Sigma, \mathcal{V}$.

By Assumption 2, we have for any $\mathcal{W}, \mathcal{Q} \in \mathbb{R}^{d_u \times D}$

$$\begin{aligned}\|\nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{W}) - \nabla\mathscr{L}_{\mathcal{Y}}(\mathcal{Q})\|_F^2 &= \frac{1}{(d_u D)^2}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\{B_{ij}^v((G^v)'(W_{ij}^v) - (G^v)'(Q_{ij}^v))\}^2 \\ &\leq \frac{U_\gamma^2}{(d_u D)^2}\|\mathcal{W} - \mathcal{Q}\|_F^2.\end{aligned}$$

This yields that $\mathscr{L}_{\mathcal{Y}}$ has L-Lipschitz continuous gradient with $L = U_\gamma/(d_u D) \leq 1$. In the following algorithm and the experimental setup, we choose to work with $L = 1$.

Penalized Likelihood Accelerated Inexact Soft Impute (PLAIS-Impute). We present here the main algorithm in this paper, referred to as PLAIS-Impute, which is tailored to solving our collective matrix completion problem. The PLAIS-Impute is an adaption of the AIS-Impute algorithm in Yao and Kwok (2015) to the penalized likelihood completion problems. Note that AIS-Impute is an accelerated proximal gradient algorithm with further speed up based on approximate SVD. However, it is dedicated only to square-loss goodness-of-fitting. The PLAIS-Impute is summarized in Algorithm 4. The core steps are 10-12, where an approximate SVT is performed. Steps 10 and 11 use the column space of the last iterations ($\mathcal{V}_t$ and $\mathcal{V}_{t-1}$) to warm-start the power method. For further speed up, a continuation strategy is employed in which λ_t is initialized to a large value and then decreases gradually. The algorithm is restarted (at the step 14) if the objective function $\mathscr{F}_\lambda$ starts to increase. As AIS-Impute, PLAIS-Impute shares both low-iteration complexity and fast $\mathcal{O}(1/T^2)$ convergence rate (see Theorem 3.4 in Yao and Kwok (2015)).

5.2. Synthetic datasets

Software. The implementation of Algorithm 4 for the nuclear norm penalized estimator (4) was done in MATLAB R2017b on a desktop computer with macOS system, Intel i7 Core 3.5 GHz CPU and 16GB of RAM. For fast computation of SVD and sparse matrix computations, the experiments call an external package called PROPACK (Larsen, 1998) implemented in C and Fortran. The code that generates all figures given below is available from `https://github.com/mzalaya/collectivemc`.

Experimental setup. In our experiments we focus on square matrices. We set the number of the source matrices $V = 3$, then, for each $v \in \{1, 2, 3\}$, the low-rank ground truth parameter matrices $\boldsymbol{M}^v \in \mathbb{R}^{d \times d_v}$ are created with sizes $d \in \{3000, 6000, 9000\}$ and

Algorithm 4: PLAIS-Impute for Collective Matrix Completion

input: observed collective matrix $\boldsymbol{\mathcal{Y}}$, parameter λ, decay parameter $\nu \in (0,1)$, tolerance ε;
$[\boldsymbol{\mathcal{U}}_0, \lambda_0, \boldsymbol{\mathcal{V}}_0] = \text{rank-1 SVD}(\boldsymbol{\mathcal{Y}})$;
initialize $c = 1$, $\delta_0 = \|\boldsymbol{\mathcal{Y}}\|_F$, $\boldsymbol{\mathcal{W}}_0 = \boldsymbol{\mathcal{W}}_1 = \lambda_0 \boldsymbol{\mathcal{U}}_0 \boldsymbol{\mathcal{V}}_0^\top$;
for $t = 1, \ldots, T$ **do**
  $\delta_t = \nu^t \delta_0$;
  $\lambda_t = \nu^t(\lambda_0 - \lambda) + \lambda$;
  $\theta_t = (c-1)/(c+2)$;
  $\boldsymbol{\mathcal{Q}}_t = (1+\theta_t)\boldsymbol{\mathcal{W}}_t - \theta_t \boldsymbol{\mathcal{W}}_{t-1}$;
  $\boldsymbol{\mathcal{Z}}_t = \nabla \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Q}}_t))$;
  $\boldsymbol{\mathcal{V}}_{t-1} = \boldsymbol{\mathcal{V}}_{t-1} - \boldsymbol{\mathcal{V}}_t(\boldsymbol{\mathcal{V}}_t^\top \boldsymbol{\mathcal{V}}_{t-1})$;
  $\boldsymbol{\mathcal{R}}_t = \text{QR}([\boldsymbol{\mathcal{V}}_t, \boldsymbol{\mathcal{V}}_{t-1}])$;
  $[\boldsymbol{\mathcal{U}}_{t+1}, \boldsymbol{\Sigma}_{t+1}, \boldsymbol{\mathcal{V}}_{t+1}] = \texttt{Approx-SVT}(\boldsymbol{\mathcal{Z}}_t, \boldsymbol{\mathcal{R}}_t, \lambda_t, \delta_t)$;
  if $\mathscr{F}_\lambda(\boldsymbol{\mathcal{U}}_{t+1}\boldsymbol{\Sigma}_{t+1}\boldsymbol{\mathcal{V}}_{t+1}^\top) > \mathscr{F}_\lambda(\boldsymbol{\mathcal{U}}_t\boldsymbol{\Sigma}_t\boldsymbol{\mathcal{V}}_t^\top)$ **then**
    $c = 1$;
  else
    $c = c + 1$;
  if $|\mathscr{F}_\lambda(\boldsymbol{\mathcal{U}}_{t+1}\boldsymbol{\Sigma}_{t+1}\boldsymbol{\mathcal{V}}_{t+1}^\top) - \mathscr{F}_\lambda(\boldsymbol{\mathcal{U}}_t\boldsymbol{\Sigma}_t\boldsymbol{\mathcal{V}}_t^\top)| \leq \varepsilon$ **then**
    break;
return $\boldsymbol{\mathcal{W}}_{T+1}$.

$d_v \in \{1000, 2000, 3000\}$ (hence $d = D = \sum_{v=1}^{3} d_v$). Each source matrix $\boldsymbol{M}^v$ is constructed as $\boldsymbol{M}^v = \boldsymbol{L}^v \boldsymbol{R}^{v^\top}$ where $\boldsymbol{L}^v \in \mathbb{R}^{d \times r_v}$ and $\boldsymbol{R}^v \in \mathbb{R}^{d_v \times r_v}$. This gives a random matrix of rank at most r_v. The parameter r_v is set to $\{5, 10, 15\}$. A fraction of the entries of $\boldsymbol{M}^v$ is removed uniformly at random with probability $p \in [0,1]$. Then, the matrices $\boldsymbol{M}^v$ are scaled so that $\|\boldsymbol{M}^v\|_\infty = \gamma = 1$.

For $\boldsymbol{M}^1$, the elements of $\boldsymbol{L}^1$ and $\boldsymbol{R}^1$ are sampled i.i.d. from the normal distribution $\mathcal{N}(0.5, 1)$. For $\boldsymbol{M}^2$, the entries of $\boldsymbol{L}^2$ and $\boldsymbol{R}^2$ are i.i.d. according to Poisson distribution with parameter 0.5. Finally, for $\boldsymbol{M}^3$, the entries of $\boldsymbol{L}^3$ and $\boldsymbol{R}^3$ are i.i.d. sampled from Bernoulli distribution with parameter 0.5. The collective matrix $\boldsymbol{\mathcal{M}}$ is constructed by concatenation of the three sources $\boldsymbol{M}^1, \boldsymbol{M}^2$ and $\boldsymbol{M}^3$, namely $\boldsymbol{\mathcal{M}} = (\boldsymbol{M}^1, \boldsymbol{M}^2, \boldsymbol{M}^3)$. All the details of these experiments are given in Table 2.

The details of our experiments are summarized in Figures 1 and 2. In Figure 1, we plot the convergence of the objective function $\mathscr{F}_\lambda$ versus time in the three experiments. Note that PLAIS-Impute inherits the speed of AIS-Impute as it does not require performing SVD and it has both low per-iteration cost and fast convergence rate. In Figure 1, we plot also the convergence of the objective function $\mathscr{F}_\lambda$ versus $-\log(\lambda)$ in the three experiments. The regularization parameter in the PLAIS-Impute is initialized to a large value and decreased gradually. In Figure 2, we illustrate a learning rank curve obtained by PLAIS-Impute, where the green color corresponds to the input rank and the cyan color to the recovered rank of the collective matrix $\boldsymbol{\mathcal{M}}$.

		$\boldsymbol{M}^1$ (*Gaussian*)	$\boldsymbol{M}^2$ (*Poisson*)	$\boldsymbol{M}^3$ (*Bernoulli*)	$\boldsymbol{\mathcal{M}}$ (*Collective*)
`exp.1`	*dimension*	3000×1000	3000×1000	3000×1000	3000×3000
	rank	5	5	5	*unknown*
`exp.2`	*dimension*	6000×2000	6000×2000	6000×2000	6000×6000
	rank	10	10	10	*unknown*
`exp.3`	*dimension*	9000×3000	9000×3000	9000×3000	9000×9000
	rank	15	15	15	*unknown*

Table 2: Details of the synthetic data in the three experiments.

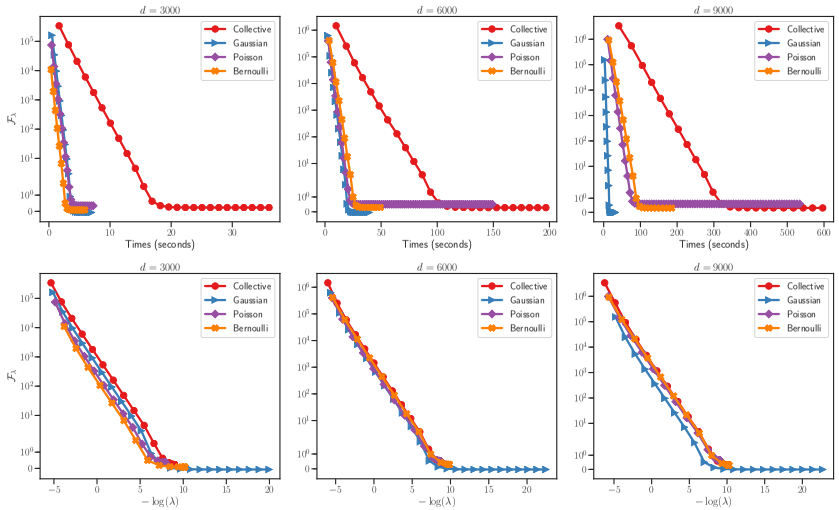


Figure 1: Convergence of the objective function $\mathscr{F}_\lambda$ in problem (4) versus time (top) and versus $-\log(\lambda)$ (bottom) in the three experiments with $p = 0.6$; left for `exp.1`; middle for `exp.2`; right for `exp.3`. Note that the objective functions for Gaussian, Poisson and Bernoulli distributions are calculated separately by the algorithm.

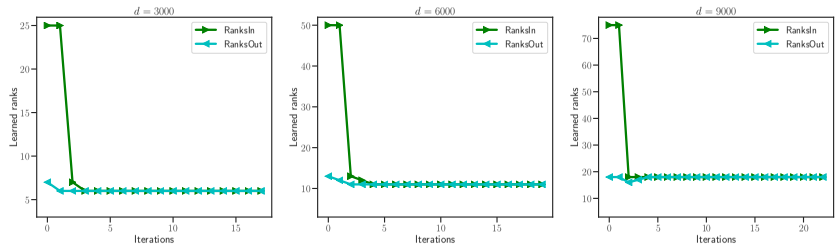


Figure 2: Learning ranks curve versus iterations in the three experiments with $p = 0.6$; left for `exp.1`; middle for `exp.2`; right for `exp.3`. We initialize the algorithm by setting a rank $r_0 = 5r$ where $r \in \{5, 10, 15\}$. The green color corresponds to the input rank while the cyan to the recovered rank of the collective matrix at each iteration. As can be seen, the two ranks gradually converge to the final recovered rank.

Evaluation. In our experiments, the PLAIS-Impute algorithm terminates when the absolute difference in the cost function values between two consecutive iterations is less than $\epsilon = 10^{-6}$. We set the regularization parameter $\lambda \propto \|\nabla \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}})\|$ as given by Theorem 3. Note that in step 12 of PLAIS-Impute, the threshold in SVT is given by λ_t (defined in step 6), which is decreasing from one iteration to another. This allows to tune the first regularization parameter λ in the program (4). We randomly sample 80% of the observed entries for training, and the rest for testing.

In order to measure the the accuracy of our estimator, we employ the relative error (as, e.g., in Cai. et al. (2010); Davenport et al. (2014); Cai and Zhou (2013)) which is widely used metric in matrix completion and is defined by

$$\mathrm{RE}(\widehat{\boldsymbol{W}}, \boldsymbol{W}) = \frac{\|\widehat{\boldsymbol{W}} - \boldsymbol{W}^o\|_F}{\|\boldsymbol{W}^o\|_F},$$

where $\widehat{\boldsymbol{W}}$ is the recovered matrix and $\boldsymbol{W}^o$ is the original full data matrix.

We run the PLAIS-Impute algorithm in each experiment by varying the percentage of known entries p from 0 to 1. In Figure 3, we plot the relative errors as a functions of p. We observe in Figure 3 that the relative errors are decaying with p. Note that for each $v \in \{1, 2, 3\}$, the estimator $\widehat{\boldsymbol{M}^v}$ is calculated separately using the same program (4). The results shown in Figure 3 confirm that collective matrix completion approach outperforms the approach that consists in estimating each component source independently.

Cold-Start problem. To simulate cold-start scenarios, we choose one of the source matrices $\boldsymbol{M}^v$ to be "cold" by increasing its sparsity. More precisely, we proceed in the following way: we extract vector of known entries of the chosen matrix and we set the first 1/5 fraction of its entries to be equal to 0. We denote the obtained matrix by $\boldsymbol{M}^v_{\text{cold}}$ and the collective matrix by $\boldsymbol{\mathcal{M}}^v_{\text{cold}}$. In `exp.1`, `exp.2` and `exp.3`, we increase the sparsity of $\boldsymbol{M}^1$, $\boldsymbol{M}^2$, and $\boldsymbol{M}^3$, respectively. Hence, we get the "cold" collective matrices $\boldsymbol{\mathcal{M}}^1_{\text{cold}} = (\boldsymbol{M}^1_{\text{cold}}, \boldsymbol{M}^2, \boldsymbol{M}^3)$, $\boldsymbol{\mathcal{M}}^2_{\text{cold}} = (\boldsymbol{M}^1, \boldsymbol{M}^2_{\text{cold}}, \boldsymbol{M}^3)$, and $\boldsymbol{\mathcal{M}}^3_{\text{cold}} = (\boldsymbol{M}^1, \boldsymbol{M}^2, \boldsymbol{M}^3_{\text{cold}})$.

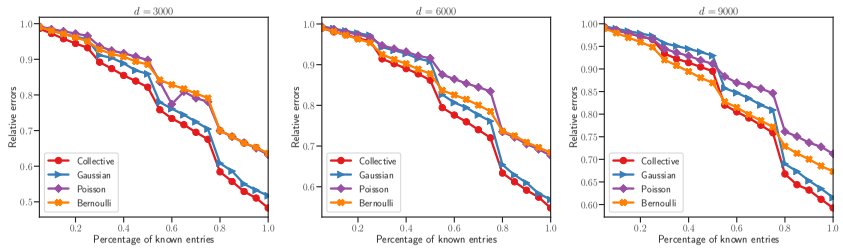


Figure 3: Performance on the synthetic data in terms of relative errors between the target and the estimator matrices as a function of the percentage of known entries p from 0 to 1.

We run 10 times the PLAIS-Impute algorithm for recovering the source $\boldsymbol{M}^v_{\text{cold}}$ and the collective $\boldsymbol{\mathcal{M}}^v_{\text{cold}}$ for each $v = 1, 2, 3$. We denote by $\widehat{\boldsymbol{M}^v_{\text{comp}}}$ the estimator of $\boldsymbol{M}^v_{\text{cold}}$ obtained by running the PLAIS-Impute algorithm only for this component. Analogously, we denote $\widehat{\boldsymbol{M}^v_{\text{collect}}}$ the estimator of $\boldsymbol{M}^v_{\text{cold}}$ obtained by extracting the v-th source of the collective estimator $\widehat{\boldsymbol{\mathcal{M}}^v_{\text{cold}}}$.

In Figure 4, we report the relative errors $\mathrm{RE}(\widehat{\boldsymbol{M}^v_{\text{comp}}}, \boldsymbol{M}^v_{\text{cold}})$ and $\mathrm{RE}(\widehat{\boldsymbol{M}^v_{\text{collect}}}, \boldsymbol{M}^v_{\text{cold}})$ in the three experiments. We see that, the collective matrix completion approach compensates the lack of informations in the "cold" source matrix. Therefore, this shared structure among the sources is useful to get better predictions.

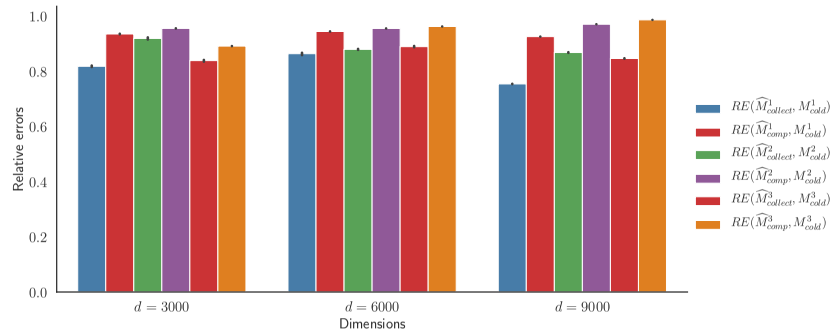


Figure 4: Relative errors over a set of 10 randomly generated datasets according to the cold-start scenarios (with the black lines representing ± the standard deviation) between the target and the estimator matrices.

6. Conclusion

This paper studies the problem of recovering a low-rank matrix when the data are collected from multiple and heterogeneous source matrices. We first consider the setting where, for each source, the matrix entries are sampled from an exponential family distribution. We then relax this assumption. The proposed estimators are based on minimizing the sum of a goodness-of-fit term and the nuclear norm penalization of the whole collective matrix. Allowing for non-uniform sampling, we establish upper bounds on the prediction risk of our estimator. As a by-product of our results, we provide exact minimax optimal rate of convergence for 1-bit matrix completion which previously was known upto a logarithmic factor. We present the proximal algorithm PLAIS-Impute to solve the corresponding convex programs. The empirical study provides evidence of the efficiency of the collective matrix completion approach in the case of joint low-rank structure compared to estimate each source matrices separately.

Acknowledgment

We would like to thank the Associated Editor and the two anonymous Referees for extremely valuable comments and remarks that helped us greatly to improve the paper. This work was supported by grants from DIM Math Innov Rgion Ile-de-France `https://www.dim-mathinnov.fr`

Appendix A. Proofs

We provide proofs of the main results, Theorems 3 and 11, in this section. The proofs of a few technical lemmas including Lemmas 5, 6 and 12 are also given. Before that, we recall some basic facts about matrices.

Basic facts about matrices. The singular value decomposition (SVD) of $\boldsymbol{A}$ has the form $\boldsymbol{A} = \sum_{l=1}^{\mathrm{rank}(\boldsymbol{A})} \sigma_l(\boldsymbol{A}) u_l(\boldsymbol{A}) v_l^\top(\boldsymbol{A})$ with orthonormal vectors $u_1(\boldsymbol{A}), \ldots, u_{\mathrm{rank}(\boldsymbol{A})}(\boldsymbol{A})$, orthonormal vectors $v_1(\boldsymbol{A}), \ldots, v_{\mathrm{rank}(\boldsymbol{A})}(\boldsymbol{A})$, and real numbers $\sigma_1(\boldsymbol{A}) \geq \cdots \geq \sigma_{\mathrm{rank}(\boldsymbol{A})}(\boldsymbol{A}) > 0$ (the singular values of $\boldsymbol{A}$). Let $(\mathcal{S}_1(\boldsymbol{A}), \mathcal{S}_2(\boldsymbol{A}))$ be the pair of linear vectors spaces, where $\mathcal{S}_1(\boldsymbol{A})$ is the linear span space of $\{u_1(\boldsymbol{A}), \ldots, u_{\mathrm{rank}(\boldsymbol{A})}(\boldsymbol{A})\}$, and $\mathcal{S}_2(\boldsymbol{\mathcal{A}})$ is the linear span space of $\{v_1(\boldsymbol{A}), \ldots, v_{\mathrm{rank}(\boldsymbol{A})}(\boldsymbol{A})\}$. We denote by $\mathcal{S}_j^\perp(\boldsymbol{A})$ the orthogonal complements of $\mathcal{S}_j(\boldsymbol{A})$, for $j = 1, 2$ and by $P_{\mathcal{S}}$ the projector on the linear subspace $\mathcal{S}$ of $\mathbb{R}^n$ or $\mathbb{R}^m$.

For two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we set $\mathscr{P}_{\boldsymbol{A}}^\perp(\boldsymbol{B}) = P_{\mathcal{S}_1^\perp(\boldsymbol{A})} \boldsymbol{B} P_{\mathcal{S}_2^\perp(\boldsymbol{A})}$ and $\mathscr{P}_{\boldsymbol{A}}(\boldsymbol{B}) = \boldsymbol{B} - \mathscr{P}_{\boldsymbol{A}}^\perp(\boldsymbol{B})$. Since $\mathscr{P}_{\boldsymbol{A}}(\boldsymbol{B}) = P_{\mathcal{S}_1(\boldsymbol{A})} \boldsymbol{B} + P_{\mathcal{S}_1^\perp(\boldsymbol{A})} \boldsymbol{B} P_{\mathcal{S}_2(\boldsymbol{A})}$, and $\mathrm{rank}(P_{\mathcal{S}_j(\boldsymbol{A})} \boldsymbol{B}) \leq \mathrm{rank}(\boldsymbol{A})$, we have that

$$\mathrm{rank}(\mathscr{P}_{\boldsymbol{A}}(\boldsymbol{B})) \leq 2\,\mathrm{rank}(\boldsymbol{A}). \tag{12}$$

It is easy to see that for two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ (Klopp, 2014)

$$\|\boldsymbol{A}\|_* - \|\boldsymbol{B}\|_* \leq \|\mathscr{P}_{\boldsymbol{A}}(\boldsymbol{A} - \boldsymbol{B})\|_* - \|\mathscr{P}_{\boldsymbol{A}}^\perp(\boldsymbol{A} - \boldsymbol{B})\|_*. \tag{13}$$

Finally, we recall the well-known trace duality property: for all $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times m}$, we have

$$|\langle \boldsymbol{A}, \boldsymbol{B} \rangle| \leq \|\boldsymbol{B}\| \|\boldsymbol{A}\|_*.$$

A.1. Proof of Theorem 3

First, noting that $\widehat{\boldsymbol{\mathcal{M}}}$ is optimal and $\boldsymbol{\mathcal{M}}$ is feasible for the convex optimization problem (4), we thus have the basic inequality that

$$\frac{1}{d_u D} \sum_{v \in [V]} \sum_{(i,j) \in [d_u] \times [d_v]} B_{ij}^v \big(G^v(\hat{M}_{ij}^v) - Y_{ij}^v \hat{M}_{ij}^v \big) + \lambda \|\widehat{\boldsymbol{\mathcal{M}}}\|_*$$
$$\leq \frac{1}{d_u D} \sum_{v \in [V]} \sum_{(i,j) \in [d_u] \times [d_v]} B_{ij}^v \big(G^v(M_{ij}^v) - Y_{ij}^v M_{ij}^v \big) + \lambda \|\boldsymbol{\mathcal{M}}\|_*.$$

It yields

$$\frac{1}{d_u D} \sum_{v \in [V]} \sum_{(i,j) \in [d_u] \times [d_v]} B_{ij}^v \Big(\big(G^v(\hat{M}_{ij}^v) - G^v(M_{ij}^v) \big) - Y_{ij}^v \big(\hat{M}_{ij}^v - M_{ij}^v \big) \Big) \leq \lambda (\|\boldsymbol{\mathcal{M}}\|_* - \|\widehat{\boldsymbol{\mathcal{M}}}\|_*).$$

Using the Bregman divergence associated to each G^v, we get

$$\frac{1}{d_u D} \sum_{v \in [V]} \sum_{(i,j) \in [d_u] \times [d_v]} B_{ij}^v d_{G^v}(\hat{M}_{ij}^v, M_{ij}^v)$$
$$\leq \lambda (\|\boldsymbol{\mathcal{M}}\|_* - \|\widehat{\boldsymbol{\mathcal{M}}}\|_*) - \frac{1}{d_u D} \sum_{v \in [V]} \sum_{(i,j) \in [d_u] \times [d_v]} B_{ij}^v \big((G^v)'(M_{ij}^v) - Y_{ij}^v \big) \big(\hat{M}_{ij}^v - M_{ij}^v \big).$$

Therefore, using the duality between $\|\cdot\|_*$ and $\|\cdot\|$, we arrive at

$$\frac{1}{d_uD}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v d_{G^v}(\hat{M}_{ij}^v, M_{ij}^v) \le \lambda(\|\boldsymbol{\mathcal{M}}\|_* - \|\widehat{\boldsymbol{\mathcal{M}}}\|_*) - \langle \nabla\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}}), \widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\rangle$$
$$\le \lambda(\|\boldsymbol{\mathcal{M}}\|_* - \|\widehat{\boldsymbol{\mathcal{M}}}\|_*) + \|\nabla\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}})\|\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_*.$$

Besides, using the assumption $\lambda \ge 2\|\nabla\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}})\|$ and inequality (13) lead to

$$\frac{1}{d_uD}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v d_{G^v}(\hat{M}_{ij}^v, M_{ij}^v) \le \frac{3\lambda}{2}\|\mathscr{P}_{\boldsymbol{\mathcal{M}}}(\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}})\|_*.$$

Since $\|\mathscr{P}_{\boldsymbol{\mathcal{A}}}(\boldsymbol{\mathcal{B}})\|_* \le \sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{A}})}\|\boldsymbol{\mathcal{B}}\|_F$ for any two matrices $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$, we obtain

$$\frac{1}{d_uD}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v d_{G^v}(\hat{M}_{ij}^v, M_{ij}^v) \le \frac{3\lambda}{2}\sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{M}})}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F. \tag{14}$$

Now, Assumption 2 implies that the Bregman divergence satisfies $L_\gamma^2(x-y)^2 \le 2d_G^v(x,y) \le U_\gamma^2(x-y)^2$, then we get

$$\Delta_{\boldsymbol{\mathcal{Y}}}^2(\widehat{\boldsymbol{\mathcal{M}}}, \boldsymbol{\mathcal{M}}) \le \frac{2}{L_\gamma^2}\frac{1}{d_uD}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v d_{G^v}(\hat{M}_{ij}^v, M_{ij}^v), \tag{15}$$

where

$$\Delta_{\boldsymbol{\mathcal{Y}}}^2(\widehat{\boldsymbol{\mathcal{M}}}, \boldsymbol{\mathcal{M}}) = \frac{1}{d_uD}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v(\hat{M}_{ij}^v - M_{ij}^v)^2.$$

Combining (14) and (15), we arrive at

$$\Delta_{\boldsymbol{\mathcal{Y}}}^2(\widehat{\boldsymbol{\mathcal{M}}}, \boldsymbol{\mathcal{M}}) \le \frac{3\lambda}{L_\gamma^2}\sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{M}})}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F. \tag{16}$$

Let us now define the threshold $\beta = \frac{946\gamma^2\log(d_u+D)}{pd_uD}$ and distinguish the two following cases that allows us to obtain an upper bound for the estimation error:
Case 1: if $(d_uD)^{-1}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2 < \beta$, then the statement of Theorem 3 is true.
Case 2: it remains to consider the case $(d_uD)^{-1}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2 \ge \beta$. Lemma 18 in Appendix B.1 implies $\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F \ge \frac{1}{4\sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{M}})}}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_*$, then we obtain

$$\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_* \le \sqrt{32\,\mathrm{rank}(\boldsymbol{\mathcal{M}})}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F.$$

This leads to $\widehat{\boldsymbol{\mathcal{M}}} \in \mathscr{C}\big(\beta, 32\,\mathrm{rank}(\boldsymbol{\mathcal{M}})\big)$, where the set

$$\mathscr{C}(\beta, r) = \Big\{\boldsymbol{\mathcal{W}} \in \mathscr{C}_\infty(\gamma) : \|\boldsymbol{\mathcal{M}} - \boldsymbol{\mathcal{W}}\|_* \le \sqrt{r}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|_F \text{ and } (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2 \ge \beta\Big\}. \tag{17}$$

Using Lemma 19 in Appendix B.1, we have

$$\Delta_{\boldsymbol{\mathcal{Y}}}^2(\widehat{\boldsymbol{\mathcal{M}}}, \boldsymbol{\mathcal{M}}) \ge \frac{\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2}{2d_uD} - 44536\,\mathrm{rank}(\boldsymbol{\mathcal{M}})\gamma^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2 - \frac{5567\gamma^2}{d_uDp}. \tag{18}$$

Together (18) and (16) imply

$$\frac{1}{2d_uD}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2 \le \frac{3\lambda}{L_\gamma^2}\sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{M}})}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_F$$
$$+ 44536\,\mathrm{rank}(\boldsymbol{\mathcal{M}})\gamma^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2 + \frac{5567\gamma^2}{pd_uD}$$
$$\le \frac{18\lambda^2 d_uD}{pL_\gamma^4}\mathrm{rank}(\boldsymbol{\mathcal{M}}) + \frac{1}{4d_uD}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2$$
$$+ 44536\,\mathrm{rank}(\boldsymbol{\mathcal{M}})\gamma^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2 + \frac{5567\gamma^2}{pd_uD}.$$

Then,

$$\frac{1}{4d_uD}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2 \le \frac{18\lambda^2 d_uD}{pL_\gamma^4}\mathrm{rank}(\boldsymbol{\mathcal{M}})$$
$$+ 44536p^{-1}d_uD\,\mathrm{rank}(\boldsymbol{\mathcal{M}})\gamma^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2 + \frac{5567\gamma^2}{d_uDp},$$

and,

$$\frac{1}{d_uD}\|\widehat{\boldsymbol{\mathcal{M}}} - \boldsymbol{\mathcal{M}}\|_{\Pi,F}^2 \le p^{-1}\max\left(d_uD\,\mathrm{rank}(\boldsymbol{\mathcal{M}})\left(\frac{c_1\lambda^2}{L_\gamma^4} + c_2\gamma^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2\right), \frac{c_3\gamma^2}{d_uD}\right),$$

where c_1, c_2 and c_3 are numerical constants. This concludes the proof of Theorem 3.

A.2. Proof of Lemma 5

We use the following result:

Proposition 16 *(Corollary 3.3 in Bandeira and van Handel (2016)) Let $\boldsymbol{W}$ be the $n\times m$ rectangular matrix whose entries W_{ij} are independent centered bounded random variables. Then there exists a universal constant $\boldsymbol{c}$ such that*

$$\mathbb{E}[\|\boldsymbol{W}\|] \le \boldsymbol{c}\Big(\kappa_1 \vee \kappa_2 + \kappa_*\sqrt{\log(n\wedge m)}\Big),$$

where we have defined

$$\kappa_1 = \max_{i\in[n]}\sqrt{\sum_{j\in[m]}\mathbb{E}[W_{i,j}^2]}, \quad \kappa_2 = \max_{j\in[m]}\sqrt{\sum_{i\in[n]}\mathbb{E}[W_{i,j}^2]}, \quad \textit{and} \quad \kappa_* = \max_{(i,j)\in[n]\times[m]}|W_{ij}|.$$

We apply Proposition 16 to $\boldsymbol{\Sigma}_R = \frac{1}{d_uD}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\varepsilon_{ij}^v B_{ij}^v E_{ij}^v$. We compute

$$\kappa_1 = \frac{1}{d_uD}\max_{i\in[d_u]}\sqrt{\sum_{v\in[V]}\sum_{j\in[d_v]}\mathbb{E}[(\varepsilon_{ij}^v)^2(B_{ij}^v)^2]} = \frac{1}{d_uD}\max_{i\in[d_u]}\sqrt{\sum_{v\in[V]}\sum_{j\in[d_v]}\pi_{ij}^v}$$
$$= \frac{1}{d_uD}\max_{i\in[d_u]}\sqrt{\pi_{i\bullet}},$$

$$\kappa_2 = \frac{1}{d_u D}\max_{v\in[V]}\max_{j\in[d_v]}\sqrt{\sum_{i\in[d_u]}\mathbb{E}[(\varepsilon_{ij}^v)^2(B_{ij}^v)^2]} = \frac{1}{d_u D}\max_{v\in[V]}\max_{j\in[d_v]}\sqrt{\sum_{i\in[d_u]}\pi_{ij}^v}$$
$$\leq \frac{1}{d_u D}\max_{j\in[d_v]}\sqrt{\max_{v\in[V]}\sum_{i\in[d_u]}\pi_{ij}^v}$$
$$\leq \frac{1}{d_u D}\max_{j\in[d_v]}\sqrt{\pi_{\bullet j}},$$

and $\kappa_* = \frac{1}{d_u D}\max_{v\in[V]}\max_{(i,j)\in[d_u]\times[d_v]}|\varepsilon_{ij}^v B_{ij}| \leq \frac{1}{d_u D}$. Using inequality (1), we have $\kappa_1 \leq \frac{\sqrt{\mu}}{d_u D}$ and $\kappa_2 \leq \frac{\sqrt{\mu}}{d_u D}$. Then, $\kappa_1 \vee \kappa_2 \leq \frac{\sqrt{\mu}}{d_u D}$, which establishes Lemma 5.

A.3. Proof of Lemma 6

We write $\nabla\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}}) = -\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} H_{ij}^v E_{ij}^v$, with $H_{ij}^v = B_{ij}^v\big(X_{ij}^v - (G^v)'(M_{ij}^v)\big)$. For a truncation level $T>0$ to be chosen, we decompose $\nabla\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}}) = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$, where

$$\boldsymbol{\Sigma}_1 = -\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\big(H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)} - \mathbb{E}\big[H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big]\big)E_{ij}^v,$$

and

$$\boldsymbol{\Sigma}_2 = -\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\big(H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])> T)} - \mathbb{E}\big[H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])> T)}\big]\big)E_{ij}^v,$$

then, the triangular inequality implies $\|\nabla\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}})\| \leq \|\boldsymbol{\Sigma}_1\| + \|\boldsymbol{\Sigma}_2\|$. Then, the proof is divided on two steps:
Step 1: control of $\|\boldsymbol{\Sigma}_1\|$. In order to control $\|\boldsymbol{\Sigma}_1\|$, we use the following bound on the spectral norms of random matrices. It is obtained by extension to rectangular matrices via self-adjoint dilation of Corollary 3.12 and Remark 3.13 in Bandeira and van Handel (2016).

Proposition 17 *(Bandeira and van Handel, 2016) Let $\boldsymbol{W}$ be the $n\times m$ rectangular matrix whose entries W_{ij} are independent centered bounded random variables. Then, for any $0\leq\epsilon\leq 1/2$ there exists a universal constant c_ϵ such that for every $x\geq 0$,*

$$\mathbb{P}\big[\|\boldsymbol{W}\| \geq 2\sqrt{2}(1+\epsilon)(\kappa_1\vee\kappa_2) + x\big] \leq (n\wedge m)\exp\Big(-\frac{x^2}{c_\epsilon\kappa_*^2}\Big),$$

where κ_1, κ_2, and κ_* are defined as in Proposition 16.

We apply Proposition 17 to $\boldsymbol{\Sigma}_1$. We compute

$$\kappa_1 = \frac{1}{d_u D}\max_{i\in[d_u]}\sqrt{\sum_{v\in[V]}\sum_{j\in[d_v]}\mathbb{E}\Big[\big(H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)} - \mathbb{E}\big[H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big]\big)^2\Big]}.$$

Besides, we have

$$\mathbb{E}\Big[\big(H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)} - \mathbb{E}\big[H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big]\big)^2\Big] \leq \mathbb{E}\big[(H_{ij}^v)^2\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big],$$

and

$$\mathbb{E}\big[(H_{ij}^v)^2\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big] = \mathbb{E}\big[(B_{ij}^v)^2\big(X_{ij}^v-\mathbb{E}[X_{ij}^v]\big)^2\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big]$$
$$\leq \pi_{ij}^v\mathbb{V}ar[X_{ij}^v]$$
$$= \pi_{ij}^v(G^v)''(M_{ij}^v).$$

By Assumption 2, we obtain $\mathbb{E}\big[(H_{ij}^v)^2\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big] \leq \pi_{ij}^v U_\gamma^2$ for all $v\in[V], (i,j)\in[d_u]\times[d_v]$. Then,

$$\kappa_1 \leq \frac{U_\gamma}{d_u D}\max_{i\in[d_u]}\sqrt{\sum_{v\in[V]}\sum_{j\in[d_v]}\pi_{ij}^v} \leq \frac{U_\gamma}{d_u D}\max_{i\in[d_u]}\sqrt{\pi_{i\bullet}^v} \leq \frac{U_\gamma\sqrt{\mu}}{d_u D},$$

and

$$\kappa_2 \leq \frac{U_\gamma}{d_u D}\max_{j\in[d_v]}\sqrt{\max_{v\in[V]}\sum_{i\in[d_u]}\pi_{ij}^v} \leq \frac{U_\gamma}{d_u D}\max_{j\in[d_v]}\sqrt{\pi_{\bullet j}} \leq \frac{U_\gamma\sqrt{\mu}}{d_u D}.$$

It yields, $\kappa_1\vee\kappa_2 \leq \frac{U_\gamma\sqrt{\mu}}{d_u D}$. Moreover, we have $\mathbb{E}\big[H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])\leq T)}\big] \leq T$, which entails $\kappa_* \leq \frac{2T}{d_u D}$. By choosing $\epsilon = 1/2$ in Proposition 17, we obtain, with probability at least $1-4(d_u\wedge D)e^{-x^2}$,

$$\|\boldsymbol{\Sigma}_1\| \leq \frac{3U_\gamma\sqrt{2\mu} + 2\sqrt{c_{1/2}}xT}{d_u D}.$$

Therefore, by setting $x = \sqrt{2\log(d_u+D)}$, we get with probability at least $1-4/(d_u+D)$,

$$\|\boldsymbol{\Sigma}_1\| \leq \frac{3U_\gamma\sqrt{2\mu} + 2\sqrt{c_{1/2}}\sqrt{2\log(d_u+D)}T}{d_u D}. \tag{19}$$

Step 2: control of $\|\boldsymbol{\Sigma}_2\|$. To control $\|\boldsymbol{\Sigma}_2\|$, we use Chebyshev's inequality, that is

$$\mathbb{P}\big[\|\boldsymbol{\Sigma}_2\| \geq \mathbb{E}[\|\boldsymbol{\Sigma}_2\|] + x\big] \leq \frac{\mathbb{V}ar[\|\boldsymbol{\Sigma}_2\|]}{x^2}, \text{ for all } x>0.$$

We start by estimating $\mathbb{E}[\|\boldsymbol{\Sigma}_2\|]$. We use the fact that $\mathbb{E}[\|\boldsymbol{\Sigma}_2\|] \leq \mathbb{E}[\|\boldsymbol{\Sigma}_2\|_F]$:

$$\mathbb{E}\big[\|\boldsymbol{\Sigma}_2\|_F^2\big] = \frac{1}{(d_u D)^2}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\mathbb{E}\Big[\big(H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])> T)} - \mathbb{E}\big[H_{ij}^v\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])> T)}\big]\big)^2\Big]$$
$$\leq \frac{1}{(d_u D)^2}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\mathbb{E}\big[(H_{ij}^v)^2\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])> T)}\big]$$
$$\leq \frac{1}{(d_u D)^2}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\pi_{ij}^v\mathbb{E}\big[(X_{ij}^v-\mathbb{E}[X_{ij}^v])^2\mathbb{1}_{((X_{ij}^v-\mathbb{E}[X_{ij}^v])> T)}\big]$$
$$\leq \frac{1}{(d_u D)^2}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]}\pi_{ij}^v\sqrt{\mathbb{E}\big[(X_{ij}^v-\mathbb{E}[X_{ij}^v])^4\big]}\sqrt{\mathbb{P}\big[((X_{ij}^v-\mathbb{E}[X_{ij}^v]) > T)\big]}.$$

By Lemma 28, we have that $X_{ij}^v - \mathbb{E}[X_{ij}^v]$ is an (U_γ, K)-sub-exponential random variable for every $v \in [V]$ and $(i,j) \in [d_u] \times [d_v]$. It yields, using (2) in Theorem 26, that

$$\mathbb{E}\big[(X_{ij}^v - \mathbb{E}[X_{ij}^v])^p\big] \le \mathcal{C} p^p \|X_{ij}^v\|_{\psi_1}^p, \text{ for every } p \ge 1,$$

and by (1) in Theorem 26

$$\mathbb{P}\big[|X_{ij}^v - \mathbb{E}[X_{ij}^v]| > T\big] \le \exp\Big(1 - \frac{T}{\mathcal{C}_{\text{se}} \|X_{ij}^v\|_{\psi_1}}\Big),$$

where $\mathcal{C}$ and $\mathcal{C}_{\text{se}}$ are absolute constants. Consequently,

$$\begin{aligned}\mathbb{E}\big[\|\mathbf{\Sigma}_2\|_F^2\big] &\le \frac{\mathcal{C}}{(d_u D)^2} \sum_{v\in[V]} \sum_{(i,j)\in[d_u]\times[d_v]} \pi_{ij}^v \sqrt{\|X_{ij}^v\|_{\psi_1}^4} \sqrt{\exp\Big(1 - \frac{T}{\mathcal{C}_{\text{se}}\|X_{ij}^v\|_{\psi_1}}\Big)} \\ &\le \frac{\mathcal{C}}{(d_u D)^2} \sum_{v\in[V]} \sum_{(i,j)\in[d_u]\times[d_v]} (U_\gamma \vee K)^2 \pi_{ij}^v \sqrt{\exp\Big(-\frac{T}{\mathcal{C}_{\text{se}} K}\Big)}.\end{aligned}$$

We choose $T = T_* := 4\mathcal{C}_{\text{se}}(U_\gamma \vee K)\log(d_u \vee D)$. It yields,

$$\begin{aligned}\mathbb{E}\big[\|\mathbf{\Sigma}_2\|_F^2\big] &\le \frac{\mathcal{C}}{(d_u D)^2} \frac{1}{(d_u \vee D)^2} \sum_{v\in[V]} \sum_{(i,j)\in[d_u]\times[d_v]} (U_\gamma \vee K)^2 \pi_{ij}^v \\ &\le \frac{\mathcal{C}(U_\gamma \vee K)^2}{(d_u D)^2} \frac{1}{(d_u \vee D)^2} \sum_{v\in[V]} \sum_{j\in[d_v]} \pi_{ij}^v \\ &\le \frac{\mathcal{C}(U_\gamma \vee K)^2}{(d_u D)^2} \frac{1}{(d_u \vee D)^2} (d_u \vee D)\mu \\ &\le \frac{\mathcal{C}(U_\gamma \vee K)^2 \mu}{(d_u D)^2 d_u \vee D}.\end{aligned}$$

Using the fact that $x \mapsto \sqrt{x}$ is concave, we obtain

$$\mathbb{E}[\|\mathbf{\Sigma}_2\|] \le \mathbb{E}[\|\mathbf{\Sigma}_2\|_F] \le \sqrt{\mathbb{E}\big[\|\mathbf{\Sigma}_2\|_F^2\big]} \le \sqrt{\frac{\mathcal{C}(U_\gamma \vee K)^2\mu}{(d_u D)^2 d_u \vee D}} \le \frac{\mathcal{C}(U_\gamma \vee K)\sqrt{\mu}}{d_u D\sqrt{d_u \vee D}}. \tag{20}$$

Let us now control the variance of $\|\mathbf{\Sigma}_2\|$. We have immediately, using (20),

$$\mathbb{V}ar[\|\mathbf{\Sigma}_2\|] \le \mathbb{E}[\|\mathbf{\Sigma}_2\|^2] \le \mathbb{E}\big[\|\mathbf{\Sigma}_2\|_F^2\big] \le \frac{\mathcal{C}(U_\gamma \vee K)^2\mu}{(d_u D)^2 d_u \vee D}.$$

By Chebyshev's inequality and using (20), we have, with probability at least $1 - 4/(d_u + D)$,

$$\|\mathbf{\Sigma}_2\| \le \frac{\mathcal{C}(U_\gamma \vee K)\sqrt{\mu}}{d_u D\sqrt{d_u \vee D}} + \frac{\mathcal{C}(U_\gamma \vee K)\sqrt{\mu}}{d_u D} \le \frac{\mathcal{C}(U_\gamma \vee K)\sqrt{\mu}}{d_u D}. \tag{21}$$

Finally, combining (19) and (21), we obtain, with probability at least $1 - 4/(d_u + D)$,

$$\|\nabla \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}})\| \le \frac{3U_\gamma\sqrt{2\mu} + 8(U_\gamma \vee K)\mathcal{C}_{\text{se}}\sqrt{2\mathcal{C}_{1/2}\log(d_u + D)}\log(d_u \vee D) + \mathcal{C}(U_\gamma \vee K)\sqrt{\mu}}{d_u D}$$

Then,

$$\|\nabla \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{M}})\| \le \mathcal{C}\Big(\frac{(U_\gamma \vee K)\big(\sqrt{\mu} + (\log(d_u \vee D))^{3/2}\big)}{d_u D}\Big),$$

where $\mathcal{C}$ is an absolute constant. This finishes the proof of Lemma 6.

A.4. Proof of Theorem 11

We start the proof with the following inequality using the fact that $\widehat{\boldsymbol{\mathcal{M}}}$ is the minimizer of the objective function in problem (8)

$$0 \le -(R_{\boldsymbol{\mathcal{Y}}}(\widehat{\boldsymbol{\mathcal{M}}}) + \Lambda\|\widehat{\boldsymbol{\mathcal{M}}}\|_*) + (R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}}) + \Lambda\|\overset{\star}{\boldsymbol{\mathcal{M}}}\|_*).$$

Then, by adding $R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) \ge 0$, we obtain

$$R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) \le -\big\{\big(R_{\boldsymbol{\mathcal{Y}}}(\widehat{\boldsymbol{\mathcal{M}}}) - R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\big) - \big(R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}})\big)\big\} + \Lambda\big(\|\overset{\star}{\boldsymbol{\mathcal{M}}}\|_* - \|\widehat{\boldsymbol{\mathcal{M}}}\|_*\big).$$

(13) implies $\|\boldsymbol{A}\|_* - \|\boldsymbol{B}\|_* \le \|\mathscr{P}_{\boldsymbol{A}}(\boldsymbol{A} - \boldsymbol{B})\|_*$ and we get

$$\begin{aligned}R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) &\le -\big\{\big(R_{\boldsymbol{\mathcal{Y}}}(\widehat{\boldsymbol{\mathcal{M}}}) - R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\big) - \big(R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}})\big)\big\} + \Lambda\|\mathscr{P}_{\overset{\star}{\boldsymbol{\mathcal{M}}}}(\overset{\star}{\boldsymbol{\mathcal{M}}} - \widehat{\boldsymbol{\mathcal{M}}})\|_* \\ &\le -\big(R_{\boldsymbol{\mathcal{Y}}}(\widehat{\boldsymbol{\mathcal{M}}}) - R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\big) + \big(R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}})\big) \\ &\quad + \Lambda\sqrt{2\,\text{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})}\|\widehat{\boldsymbol{\mathcal{M}}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_F.\end{aligned} \tag{22}$$

Let us now define the threshold $\nu = \frac{32\left(1+e\sqrt{3\rho/\varsigma\gamma}\right)\rho\gamma\log(d_u+D)}{3pd_uD}$ and distinguish the two following cases that allows us to obtain an upper bound for the prediction error:

Case 1: if $R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) < \nu$, then the statement of Theorem 11 is true.

Case 2: it remains to consider the case $R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) \ge \nu$. Lemma 21 implies

$$\|\widehat{\boldsymbol{\mathcal{M}}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_* \le \sqrt{32\,\text{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})}\|\widehat{\boldsymbol{\mathcal{M}}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_F,$$

then $\widehat{\boldsymbol{\mathcal{M}}} \in \mathscr{Q}(\nu, 32\,\text{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}}))$ where

$$\mathscr{Q}(\nu, r) = \Big\{\boldsymbol{\mathcal{Q}} \in \mathscr{C}_\infty(\gamma) : \|\boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_* \le \sqrt{r}\|\boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_F \text{ and } R(\boldsymbol{\mathcal{Q}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) \ge \nu\Big\}.$$

Using Lemma 22, we have

$$\begin{aligned}&R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) - \big(R_{\boldsymbol{\mathcal{Y}}}(\widehat{\boldsymbol{\mathcal{M}}}) - R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\big) \\ &\quad \le \frac{R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}})}{2} + \frac{\mathcal{C}\,\text{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})\rho^2\varsigma^{-1}(\mathbb{E}[\|\mathbf{\Sigma}_R\|])^2}{(1/4e) + (1 - 1/\sqrt{4e})\sqrt{3\rho/4\varsigma\gamma}}.\end{aligned} \tag{23}$$

Now, plugging (23) in (22), we get

$$R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) \leq \frac{\mathcal{C}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})\rho^2\varsigma^{-1}(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2}{(1/4e)+(1-1/\sqrt{4e})\sqrt{3\rho/4\varsigma\gamma}} + 2\Lambda\sqrt{2\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})}\|\widehat{\boldsymbol{\mathcal{M}}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_F,$$

where $\mathcal{C} = 1024$. Then using the fact that for any $a, b \in \mathbb{R}$, and $\epsilon > 0$, we have $2ab \leq a^2/(2\epsilon) + 2\epsilon b^2$, we get for $\epsilon = p\varsigma/4$

$$\begin{aligned}
R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) &\leq \frac{\mathcal{C}d_u D p^{-1}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})\rho^2\varsigma^{-1}(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2}{(1/4e)+(1-1/\sqrt{4e})\sqrt{3\rho/4\varsigma\gamma}} \\
&\quad + \Lambda^2 d_u D(p\varsigma/4)^{-1}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}}) + \frac{p\varsigma}{2d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_F^2 \\
&\leq \frac{\mathcal{C}d_u D p^{-1}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})\rho^2\varsigma^{-1}(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2}{(1/4e)+(1-1/\sqrt{4e})\sqrt{3\rho/4\varsigma\gamma}} \\
&\quad + \Lambda^2 d_u D(p\varsigma/4)^{-1}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}}) + \frac{\varsigma}{2d_u D}\|\widehat{\boldsymbol{\mathcal{M}}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_{\Pi,F}^2.
\end{aligned}$$

Using Assumption 4, we obtain

$$\begin{aligned}
R(\widehat{\boldsymbol{\mathcal{M}}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}}) &\leq \frac{2\mathcal{C}d_u D p^{-1}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})\rho^2\varsigma^{-1}(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2}{(1/4e)+(1-1/\sqrt{4e})\sqrt{3\rho/4\varsigma\gamma}} + 8\Lambda^2 d_u D(p\varsigma)^{-1}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}}) \\
&\leq (p\varsigma)^{-1}\operatorname{rank}(\overset{\star}{\boldsymbol{\mathcal{M}}})d_u D\Big(\frac{\rho^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2}{(1/4e)+(1-1/\sqrt{4e})\sqrt{3\rho/4\varsigma\gamma}} + 8\Lambda^2\Big).
\end{aligned}$$

This finishes the proof of Theorem 11.

A.5. Proof of Lemma 12

By the nonnegative factor and the sum properties of subdifferential calculus (Boyd and Vandenberghe, 2004), we write

$$\partial R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}}) = \Big\{\boldsymbol{\mathcal{G}} = \frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v G_{ij}^v E_{ij}^v : G_{ij}^v \in \partial\ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)\Big\}$$

Recall that the sudifferential of $\partial\ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)$ at the point $\overset{\star}{M}{}_{ij}^v$ is defined as

$$\partial\ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v) = \{G_{ij}^v : \ell^v(Y_{ij}^v, Q_{ij}^v) \geq \ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v) + G_{ij}^v(Q_{ij}^v - \overset{\star}{M}{}_{ij}^v)\}.$$

Thanks to Assumption 3, we have, for all $G_{ij}^v \in \partial\ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)$

$$|G_{ij}^v(Q_{ij}^v - \overset{\star}{M}{}_{ij}^v)| \leq |\ell^v(Y_{ij}^v, Q_{ij}^v) - \ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)| \leq \rho_v|Q_{ij}^v - \overset{\star}{M}{}_{ij}^v|,$$

In particular, with $Q_{ij}^v \neq \overset{\star}{M}{}_{ij}^v$ for all $v \in [V]$ and $(i,j) \in [d_u]\times[d_v]$, we get $|G_{ij}^v| \leq \rho_v$. Then, any subgradient $\boldsymbol{\mathcal{G}}$ of $R_{\boldsymbol{\mathcal{Y}}}$ has entries bounded by $\rho/(d_u D)$ (recall $\rho = \max_{v\in[V]}\rho_v$). By a

triangular inequality and the convexity of $\|\cdot\|$, we have

$$\begin{aligned}
\|\boldsymbol{\mathcal{G}}\| &\leq \|\boldsymbol{\mathcal{G}} - \mathbb{E}[\boldsymbol{\mathcal{G}}]\| + \|\mathbb{E}[\boldsymbol{\mathcal{G}}]\| \\
&\leq \|\boldsymbol{\mathcal{G}} - \mathbb{E}[\boldsymbol{\mathcal{G}}]\| + \mathbb{E}[\|\boldsymbol{\mathcal{G}}\|],
\end{aligned}$$

for any subgradient $\boldsymbol{\mathcal{G}}$ of $R_{\boldsymbol{\mathcal{Y}}}$. On the one hand, we use the fact that $\mathbb{E}[\|\boldsymbol{\mathcal{G}}\|] \leq \mathbb{E}[\|\boldsymbol{\mathcal{G}}\|_F] \leq \sqrt{\mathbb{E}[\|\boldsymbol{\mathcal{G}}\|_F^2]}$. Using (1), we have

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{\mathcal{G}}\|_F^2] &\leq \frac{1}{(d_u D)^2}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} \rho_v^2\mathbb{E}[B_{ij}^v] \\
&\leq \frac{\rho^2}{(d_u D)^2}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} \pi_{ij}^v \\
&\leq \frac{\rho^2\mu}{(d_u D)^2}.
\end{aligned}$$

Now we apply Proposition 17 to $\boldsymbol{\mathcal{G}} - \mathbb{E}[\boldsymbol{\mathcal{G}}]$. Taking into account (1), we upper bound the constants κ_1, κ_2 and κ_* as follows:

$$\begin{aligned}
\kappa_1 &= \frac{1}{d_u D}\max_{i\in[d_u]}\sqrt{\sum_{v\in[V]}\sum_{j\in[d_v]}\mathbb{E}[(B_{ij}^v G_{ij}^v - \mathbb{E}[B_{ij}^v G_{ij}^v])^2]} \\
&\leq \frac{2\rho}{d_u D}\max_{i\in[d_u]}\sqrt{\sum_{v\in[V]}\sum_{j\in[d_v]}\pi_{ij}^v} \\
&\leq \frac{2\rho\sqrt{\mu}}{d_u D},
\end{aligned}$$

$$\begin{aligned}
\kappa_2 &= \frac{1}{d_u D}\max_{v\in[V]}\max_{j\in[d_v]}\sqrt{\sum_{i\in[d_u]}\mathbb{E}[(B_{ij}^v G_{ij}^v - \mathbb{E}[B_{ij}^v G_{ij}^v])^2]} \\
&\leq \frac{2\rho}{d_u D}\max_{v\in[V]}\max_{j\in[d_v]}\sqrt{\sum_{i\in[d_u]}\pi_{ij}^v} \\
&\leq \frac{2\rho\sqrt{\mu}}{d_u D},
\end{aligned}$$

and $\kappa_* = \frac{1}{d_u D}\max_{v\in[V]}\max_{(i,j)\in[d_u]\times[d_v]}|B_{ij}^v G_{ij}^v - \mathbb{E}[B_{ij}^v G_{ij}^v]| \leq \frac{2\rho}{d_u D}$. Now, choose $\epsilon = 1/2$ in Proposition 17, then we obtain, with probability at least $1 - 4(d_u \wedge D)e^{-x^2}$,

$$\|\boldsymbol{\mathcal{G}} - \mathbb{E}[\boldsymbol{\mathcal{G}}]\| \leq \frac{6\rho\sqrt{2\mu} + 2\rho\sqrt{\mathcal{C}_{1/2}}x}{d_u D}. \tag{24}$$

Setting $x = \sqrt{2\log(d_u + D)}$ in (24), we get with probability at least $1 - 4/(d_u + D)$,

$$\|\boldsymbol{\mathcal{G}}\| \leq \frac{(1+6\sqrt{2})\rho\sqrt{\mu} + 2\rho\sqrt{\mathcal{C}_{1/2}}\sqrt{2\log(d_u + D)}}{d_u D}, \tag{25}$$

for any subgradient $\boldsymbol{\mathcal{G}}$ of $R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})$.

Appendix B. Technical Lemmas

In this section, we provide several technical lemmas, which are used for proving our main results.

B.1. Useful lemmas for the proof of Theorem 3

Lemma 18 *Let $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathscr{C}_\infty(\gamma)$. Assume that $\lambda \geq 2\|\nabla \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{B}})\|$, and $\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{A}}) + \lambda\|\boldsymbol{\mathcal{A}}\|_* \leq \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{B}}) + \lambda\|\boldsymbol{\mathcal{B}}\|_*$. Then,*

(i) $\|\mathscr{P}^{\perp}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_* \leq 3\|\mathscr{P}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_*$,

(ii) $\|\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\|_* \leq 4\sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{B}})}\|\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\|_F$.

Proof We have $\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{B}}) - \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{A}}) \geq \lambda(\|\boldsymbol{\mathcal{A}}\|_* - \|\boldsymbol{\mathcal{B}}\|_*)$. (13) implies

$$\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{B}}) - \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{A}}) \geq \lambda\big(\|\mathscr{P}^{\perp}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_* - \|\mathscr{P}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_*\big).$$

Moreover, by convexity of $\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\cdot)$ and the duality between $\|\cdot\|_*$ and $\|\cdot\|$ we obtain

$$\mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{B}}) - \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{A}}) \leq \langle \nabla \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{B}}), \boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{A}}\rangle \leq \|\nabla \mathscr{L}_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{B}})\|\|\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{A}}\|_* \leq \frac{\lambda}{2}\|\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{A}}\|_*.$$

Therefore,

$$\|\mathscr{P}^{\perp}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_* \leq \|\mathscr{P}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_* + \frac{1}{2}\|\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\|_* \tag{26}$$

Using the triangle inequality, we get

$$\|\mathscr{P}^{\perp}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_* \leq 3\|\mathscr{P}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_*,$$

which proves (i). To prove (ii), note that $\|\mathscr{P}_{\boldsymbol{\mathcal{B}}}(\boldsymbol{\mathcal{A}})\|_* \leq \sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{B}})}\|\boldsymbol{\mathcal{A}}\|_F$, and (i) imply

$$\|\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\|_* \leq 4\sqrt{2\,\mathrm{rank}(\boldsymbol{\mathcal{B}})}\|\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}})\|_F.$$

■

Lemma 19 *Let $\beta = \frac{946\gamma^2\log(d_u+D)}{pd_uD}$. Then, for all $\boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r)$,*

$$\Big|\Delta^2_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{M}}) - (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}]\Big| \leq \frac{(d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}}{2} + 1392r\gamma^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2 + \frac{5567\gamma^2}{d_uDp}$$

with probability at least $1 - 4/(d_u + D)$.

Proof We use a standard peeling argument. For any $\alpha > 1$ and $0 < \eta < 1/2\alpha$, we define

$$\boldsymbol{\kappa} = \frac{1}{1/(2\alpha) - \eta}\Big(128\gamma^2 r(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2 + \frac{512\gamma^2}{d_uDp}\Big)$$

and we consider the event

$$\mathscr{W} = \Big\{\exists\, \boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r) : \Big|\Delta^2_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{M}}) - (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}\Big| > \frac{(d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}}{2} + \boldsymbol{\kappa}\Big\}.$$

For $s \in \mathbb{N}^*$, set

$$\mathcal{R}_s = \Big\{\boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r) : \alpha^{s-1}\beta \leq (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F} \leq \alpha^s\beta\Big\}.$$

If the event $\mathscr{W}$ holds for some matrix $\boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r)$, then $\boldsymbol{\mathcal{W}}$ belongs to some $\mathcal{R}_s$ and

$$\begin{aligned}\Big|\Delta^2_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{M}}) - (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}\Big| &\geq \frac{(d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}}{2} + \boldsymbol{\kappa} \\ &\geq \frac{1}{2\alpha}\alpha^s\beta + \boldsymbol{\kappa}.\end{aligned}$$

For $\theta \geq \beta$ consider the following set of matrices

$$\mathscr{C}(\beta, r, \theta) = \Big\{\boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r) : (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F} \leq \theta\Big\},$$

and the following event

$$\mathscr{W}_s = \Big\{\exists\, \boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r, \theta) : \Big|\Delta^2_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{M}}) - (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}\Big| \geq \frac{1}{2\alpha}\alpha^s\beta + \boldsymbol{\kappa}\Big\}.$$

Note that $\boldsymbol{\mathcal{W}} \in \mathscr{W}_s$ implies that $\boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r, \alpha^s\beta)$. Then, we get $\mathscr{W} \subset \cup_s \mathscr{W}_s$. Thus, it is enough to estimate the probability of the simpler event $\mathscr{W}_s$ and then apply a the union bound. Such an estimation is given by the following lemma:

Lemma 20 *Let*

$$\boldsymbol{\mathcal{Z}}_\theta = \sup_{\boldsymbol{\mathcal{W}} \in \mathscr{C}(\beta, r, \theta)} \Big|\Delta^2_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{M}}) - (d_uD)^{-1}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{M}}\|^2_{\Pi,F}\Big|.$$

Then, we have

$$\mathbb{P}\big[\boldsymbol{\mathcal{Z}}_\theta > \frac{\theta}{2\alpha} + \boldsymbol{\kappa}\big] \leq 4\exp\Big(-\frac{pd_uD\eta^2\theta}{8\gamma^2}\Big).$$

The proof of Lemma 20 follows along the same lines of Lemma 10 in Klopp (2015). We now apply an union bound argument combined to Lemma 20, we get

$$\begin{aligned}\mathbb{P}[\mathscr{W}] \leq \mathbb{P}[\cup_{s=1}^{\infty}\mathscr{W}_s] &\leq 4\sum_{s=1}^{\infty}\exp\Big(-\frac{pd_uD\eta^2\alpha^s\beta}{8\gamma^2}\Big) \\ &\leq 4\sum_{s=1}^{\infty}\exp\Big(-\frac{pd_uD\eta^2\beta\log\alpha}{8\gamma^2}s\Big) \\ &\leq \frac{4\exp\Big(-\frac{pd_uD\eta^2\beta\log\alpha}{8\gamma^2}\Big)}{1 - \exp\Big(-\frac{pd_uD\eta^2\beta\log\alpha}{8\gamma^2}\Big)}.\end{aligned}$$

By choosing $\alpha = e, \eta = 1/4e$ and β as stated we get the desired result. ■

B.2. Useful lemmas for the proof of Theorem 11

Lemma 21 *Suppose* $\Lambda \geq 2\sup\{\|\mathcal{G}\| : \mathcal{G} \in \partial R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})\}$. *Then*

$$\|\widehat{\mathcal{M}} - \overset{\star}{\mathcal{M}}\|_* \leq 4\sqrt{2\,\mathrm{rank}(\overset{\star}{\mathcal{M}})}\|\widehat{\mathcal{M}} - \overset{\star}{\mathcal{M}}\|_F.$$

Proof For any subgradient $\mathcal{G}$ of $R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})$, we have $R_{\mathcal{Y}}(\widehat{\mathcal{M}}) \geq R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}}) + \langle \mathcal{G}, \widehat{\mathcal{M}} - \overset{\star}{\mathcal{M}}\rangle$. Then, the definition of the estimator $\widehat{\mathcal{M}}$, entails $R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}}) - R_{\mathcal{Y}}(\widehat{\mathcal{M}}) \geq \Lambda(\|\widehat{\mathcal{M}}\|_* - \|\overset{\star}{\mathcal{M}}\|_*)$, hence $\langle \mathcal{G}, \overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}}\rangle \geq \Lambda(\|\widehat{\mathcal{M}}\|_* - \|\overset{\star}{\mathcal{M}}\|_*)$. The duality between $\|\cdot\|_*$ and $\|\cdot\|$ yields

$$\Lambda(\|\widehat{\mathcal{M}}\|_* - \|\overset{\star}{\mathcal{M}}_*\|) \leq \|\mathcal{G}\|\|\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}}\|_* \leq \frac{\Lambda}{2}\|\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}}\|_*$$

then $\|\widehat{\mathcal{M}}\|_* - \|\overset{\star}{\mathcal{M}}_*\| \leq \frac{1}{2}\|\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}}\|_*$. Now, (13) implies

$$\|\mathscr{P}^{\perp}_{\overset{\star}{\mathcal{M}}}(\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}})\|_* \leq \|\mathscr{P}_{\overset{\star}{\mathcal{M}}}(\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}})\|_* + \frac{1}{2}\|\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}}\|_* \leq 3\|\mathscr{P}_{\overset{\star}{\mathcal{M}}}(\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}})\|_*.$$

Therefore $\|\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}}\|_* \leq 4\|\mathscr{P}_{\overset{\star}{\mathcal{M}}}(\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}})\|_*$. Since $\|\mathscr{P}_{\overset{\star}{\mathcal{M}}}(\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}})\|_* \leq \sqrt{2\,\mathrm{rank}(\overset{\star}{\mathcal{M}})}\|\overset{\star}{\mathcal{M}} - \widehat{\mathcal{M}}\|_F$, we establish the proof of Lemma 21. ■

Lemma 22 *Let*

$$\nu = \frac{32\big(1 + e\sqrt{3\rho/\varsigma\gamma}\big)\rho\gamma\log(d_u + D)}{3pd_uD},$$

then, with probability at least $1 - 4/(d_u + D)$, *the following holds uniformly over* $\mathcal{Q} \in \mathscr{Q}(\nu, r)$

$$\begin{aligned}&\Big|\big(R_{\mathcal{Y}}(\mathcal{Q}) - R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})\big) - \big(R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})\big)\Big| \\ &\qquad \leq \frac{R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})}{2} + \frac{16}{(1/4e) + (1 - 1/\sqrt{4e})\sqrt{3\rho/4\varsigma\gamma}} r\rho^2(p\varsigma)^{-1}(\mathbb{E}[\|\mathbf{\Sigma}_R\|])^2.\end{aligned}$$

Proof The proof is based on the peeling argument. For any $\delta > 1$ and $0 < \vartheta < 1/2\delta$, define

$$\boldsymbol{\zeta} = \frac{16r(p\varsigma)^{-1}\rho^2(\mathbb{E}[\|\mathbf{\Sigma}_R\|])^2}{(1/2\delta) + \sqrt{3\rho/4\varsigma\gamma} - \Big(\vartheta + \sqrt{3\rho/4\varsigma\gamma\vartheta}\Big)}, \tag{27}$$

and we consider the event

$$\mathscr{A} = \Big\{\exists\, \mathcal{Q} \in \mathscr{Q}(\nu, r) : \Big|\big(R_{\mathcal{Y}}(\mathcal{Q}) - R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})\big) - \big(R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})\big)\Big| > \frac{R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})}{2} + \boldsymbol{\zeta}\Big\}.$$

For $l \in \mathbb{N}^*$, we define the sequence of subsets

$$\mathscr{J}_l = \Big\{\mathcal{Q} \in \mathscr{Q}(\nu, r) : \delta^{l-1}\nu \leq R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}}) \leq \delta^l\nu\Big\}.$$

If the event $\mathscr{A}$ holds for some matrix $\mathcal{Q} \in \mathscr{Q}(\nu, r)$, then $\mathcal{Q}$ belongs to some $\mathscr{J}_l$ and

$$\begin{aligned}\Big|\big(R_{\mathcal{Y}}(\mathcal{Q}) - R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})\big) - \big(R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})\big)\Big| &> \frac{R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})}{2} + \boldsymbol{\zeta} \\ &\geq \frac{1}{2\delta}\delta^l\nu + \boldsymbol{\zeta}.\end{aligned}$$

For $\theta \geq \nu$, consider the following set of matrices

$$\mathscr{Q}(\nu, r, \theta) = \Big\{\mathcal{Q} \in \mathscr{Q}(\nu, r) : R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}}) \leq \theta\Big\},$$

and the following event

$$\mathscr{A}_l = \Big\{\exists\, \mathcal{Q} \in \mathscr{Q}(\nu, r, \theta) : \Big|\big(R_{\mathcal{Y}}(\mathcal{Q}) - R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})\big) - \big(R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})\big)\Big| \geq \frac{1}{2\delta}\delta^l\nu + \boldsymbol{\zeta}\Big\}.$$

Note that $\mathcal{Q} \in \mathscr{J}_l$ implies that $\mathcal{Q} \in \mathscr{Q}(\nu, r, \delta^l\nu)$. Then, we get $\mathscr{A} \subset \cup_l \mathscr{A}_l$. Thus, it is enough to estimate the probability of the simpler event $\mathscr{A}_l$ and then apply a the union bound. Such an estimation is given in Lemma 23, where we derive a concentration inequality for the following supremum of process:

$$\mathbf{\Xi}_\theta = \sup_{\mathcal{Q} \in \mathscr{Q}(\nu, r, \theta)} \Big|\big(R_{\mathcal{Y}}(\mathcal{Q}) - R_{\mathcal{Y}}(\overset{\star}{\mathcal{M}})\big) - \big(R(\mathcal{Q}) - R(\overset{\star}{\mathcal{M}})\big)\Big|$$

We now apply an union bound argument combined to Lemma 23, we get

$$\begin{aligned}\mathbb{P}[\mathscr{A}] \leq \mathbb{P}[\cup_{l=1}^{\infty}\mathscr{A}_l] &\leq \sum_{l=1}^{\infty}\exp\Big(-\frac{3d_uD\vartheta\delta^l\nu}{8\rho\gamma}\Big) \\ &\leq \sum_{l=1}^{\infty}\exp\Big(-\frac{3d_uD\vartheta\log(\delta)\nu}{8\rho\gamma}l\Big) \\ &\leq \frac{\exp\Big(-\frac{3d_uD\vartheta\log(\delta)\nu}{8\rho\gamma}\Big)}{1 - \exp\Big(-\frac{3d_uD\vartheta\log(\delta)\nu}{8\rho\gamma}\Big)},\end{aligned}$$

where se used the elementary inequality that $u^s = e^{s\log(u)} \geq s\log(u)$. By choosing $\delta = e, \vartheta = 1/4e$ and ν as stated we get the desired result. ■

Lemma 23 *One has*

$$\mathbb{P}\Big[\mathbf{\Xi}_\theta \geq \Big(1 + \delta\sqrt{\frac{3\rho}{\varsigma\gamma}}\Big)\frac{\theta}{2\delta} + \boldsymbol{\zeta}\Big] \leq \exp\Big(-\frac{3d_uD\vartheta\theta}{8\rho\gamma}\Big).$$

Proof The proof of this lemma is based on Bousquet's concentration theorem:

Theorem 24 *(Bousquet, 2002) (see also Corollary 16.1 in van de Geer (2016)) Let $\mathcal{F}$ be a class of real-valued functions. Let $T_1, \dots, T_N$ be independent random variables such that $\mathbb{E}[f(T_i)] = 0$ and $|f(T_i)| \le \xi$ for all $i = 1, \dots, N$ and for all $f \in \mathcal{F}$. Introduce $Z = \sup_{f\in\mathcal{F}} \left|\frac{1}{N}\sum_{i=1}^N \big(f(T_i) - \mathbb{E}[f(T_i)]\big)\right|$. Assume further that*

$$\frac{1}{N}\sum_{i=1}^N \sup_{f\in\mathcal{F}} \mathbb{E}\big[f^2(T_i)\big] \le M^2.$$

Then we have for all $t > 0$

$$\mathbb{P}\left[Z \ge 2\mathbb{E}[Z] + M\sqrt{\frac{2t}{N}} + \frac{4t\xi}{3N}\right] \le e^{-t}.$$

We start by bounding the expectation

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\Xi}_\theta] &= \mathbb{E}\Big[\sup_{\boldsymbol{\mathcal{Q}}\in\mathscr{Q}(\nu,r,\theta)} \Big|\big(R_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Q}}) - R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\big) - \big(R(\boldsymbol{\mathcal{Q}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}})\big)\Big|\Big] \\
&= \mathbb{E}\Big[\sup_{\boldsymbol{\mathcal{Q}}\in\mathscr{Q}(\nu,r,\theta)} \Big|\big(R_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Q}}) - R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\big) - \mathbb{E}\big[R_{\boldsymbol{\mathcal{Y}}}(\boldsymbol{\mathcal{Q}}) - R_{\boldsymbol{\mathcal{Y}}}(\overset{\star}{\boldsymbol{\mathcal{M}}})\big]\Big|\Big] \\
&= \mathbb{E}\Big[\sup_{\boldsymbol{\mathcal{Q}}\in\mathscr{Q}(\nu,r,\theta)} \Big|\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} B_{ij}^v\big(\ell^v(Y_{ij}^v, Q_{ij}^v) - \ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)\big) \\
&\qquad\qquad - \mathbb{E}\big[B_{ij}^v\big(\ell^v(Y_{ij}^v, Q_{ij}^v) - \ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)\big)\big]\Big|\Big] \\
&\le 2\mathbb{E}\Big[\sup_{\boldsymbol{\mathcal{Q}}\in\mathscr{Q}(\nu,r,\theta)} \Big|\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} \varepsilon_{ij}^v B_{ij}^v\big(\ell^v(Y_{ij}^v, Q_{ij}^v) - \ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)\big)\Big|\Big] \\
&\le 4\rho\mathbb{E}\Big[\sup_{\boldsymbol{\mathcal{Q}}\in\mathscr{Q}(\nu,r,\theta)} \Big|\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} \varepsilon_{ij}^v B_{ij}^v(Q_{ij}^v - \overset{\star}{M}{}_{ij}^v)\Big|\Big] \\
&\le 4\rho\mathbb{E}\Big[\sup_{\boldsymbol{\mathcal{Q}}\in\mathscr{Q}(\nu,r,\theta)} \Big|\langle \boldsymbol{\Sigma}_R, \boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\rangle\Big|\Big] \\
&\le 4\rho\mathbb{E}\Big[\|\boldsymbol{\Sigma}_R\| \sup_{\boldsymbol{\mathcal{Q}}\in\mathscr{Q}(\nu,r,\theta)} \|\boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_*\Big],
\end{aligned}$$

where the first inequality follows from symmetrization of expectations theorem of van der Vaart and Wellener, the second from contraction principle of Ledoux and Talagrand (see Theorems 14.3 and 14.4 in Bühlmann and van de Geer (2011)), and the third from duality between nuclear and operator norms. We have $\boldsymbol{\mathcal{Q}} \in \mathscr{Q}(\nu, r, \theta)$ then $\|\boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_* \le \sqrt{r}\|\boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_F$ and using Assumption 4, we have $\|\boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_* \le \sqrt{r(p\varsigma)^{-1}\big(R(\boldsymbol{\mathcal{Q}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}})\big)} \le \sqrt{r(p\varsigma)^{-1}\theta}$. Then,

$$\mathbb{E}[\boldsymbol{\Xi}_\theta] \le 4\sqrt{r(p\varsigma)^{-1}\theta}\rho\mathbb{E}[\|\boldsymbol{\Sigma}_R\|].$$

For the upper bound ξ in Theorem 24, we have that

$$\big|\ell^v(Y_{ij}^v, Q_{ij}^v) - \ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)\big| \le \rho_v|Q_{ij}^v - \overset{\star}{M}{}_{ij}^v| \le 2\rho_v\gamma \le 2\rho\gamma.$$

Now we compute M in Theorem 24. Thanks to Assumption 4, we have

$$\begin{aligned}
&\frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} \mathbb{E}\big[\big(B_{ij}^v\big(\ell^v(Y_{ij}^v, Q_{ij}^v) - \ell^v(Y_{ij}^v, \overset{\star}{M}{}_{ij}^v)\big)\big)^2\big] \\
&\le \frac{1}{d_u D}\sum_{v\in[V]}\sum_{(i,j)\in[d_u]\times[d_v]} (\rho_v)^2\mathbb{E}\big[B_{ij}^v(Q_{ij}^v - \overset{\star}{M}{}_{ij}^v)^2\big] \\
&\le \frac{\rho^2}{d_u D}\|\boldsymbol{\mathcal{Q}} - \overset{\star}{\boldsymbol{\mathcal{M}}}\|_{\Pi,F}^2 \\
&\le \frac{\rho^2}{\varsigma}(R(\boldsymbol{\mathcal{Q}}) - R(\overset{\star}{\boldsymbol{\mathcal{M}}})) \\
&\le \frac{\rho^2\theta}{\varsigma}.
\end{aligned}$$

Then, Bousquet's theorem implies that for all $t > 0$,

$$\mathbb{P}\Big[\boldsymbol{\Xi}_\theta \ge 2\mathbb{E}[\boldsymbol{\Xi}_\theta] + \sqrt{\frac{2\rho^2\theta t}{\varsigma d_u D}} + \frac{8\rho\gamma t}{3d_u D}\Big] \le e^{-t}.$$

Taking $t = \frac{3d_u D\vartheta\theta}{8\rho\gamma}$, we obtain

$$\mathbb{P}\Big[\boldsymbol{\Xi}_\theta \ge 8\gamma\sqrt{r(p\varsigma)^{-1}\theta}\rho\mathbb{E}[\|\boldsymbol{\Sigma}_R\|] + \Big(\sqrt{\frac{3\rho}{4\varsigma\gamma}}\vartheta + \vartheta\Big)\theta\Big] \le \exp\Big(-\frac{3d_u D\vartheta\theta}{8\rho\gamma}\Big). \tag{28}$$

Using the fact that for any $a, b \in \mathbb{R}$, and $\epsilon > 0$, $2ab \le a^2/\epsilon + \epsilon b^2$, we get (for $\epsilon = 1/2\delta + \sqrt{3\rho/4\varsigma\gamma} - \Big(\vartheta + \sqrt{3\rho\vartheta/4\varsigma\gamma}\Big)$), we get

$$\begin{aligned}
8\gamma\sqrt{r(p\varsigma)^{-1}\theta}\rho\mathbb{E}[\|\boldsymbol{\Sigma}_R\|] + \Big(\sqrt{\frac{3\vartheta\rho}{\varsigma\gamma}} + \vartheta\Big)\theta &\le \frac{16r(p\varsigma)^{-1}\rho^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2}{\frac{1}{2\delta} + \sqrt{\frac{3\rho}{4\varsigma\gamma}} - \vartheta - \sqrt{\frac{3\rho}{4\varsigma\gamma}}\vartheta} + \Big(\frac{1}{2\delta} + \sqrt{\frac{3\rho}{4\varsigma\gamma}}\Big)\theta \\
&\le \frac{16r(p\varsigma)^{-1}\rho^2(\mathbb{E}[\|\boldsymbol{\Sigma}_R\|])^2}{\frac{1}{2\delta} + \sqrt{\frac{3\rho}{4\varsigma\gamma}} - \vartheta - \sqrt{\frac{3\rho}{4\varsigma\gamma}}\vartheta} + \Big(1 + \delta\sqrt{\frac{3\rho}{\varsigma\gamma}}\Big)\frac{\theta}{2\delta}.
\end{aligned}$$

Using (28), we get $\mathbb{P}\Big[\boldsymbol{\Xi}_\theta \ge \Big(1 + \delta\sqrt{\frac{3\rho}{\varsigma\gamma}}\Big)\frac{\theta}{2\delta} + \boldsymbol{\zeta}\Big] \le \exp\Big(-\frac{3d_u D\vartheta\theta}{8\rho\gamma}\Big)$. This finishes the proof of Lemma 23. ■

Appendix C. Sub-exponential random variables

The material here is taken from R.Vershynin (2010).

Definition 25 *A random variable X is sub-exponential with parameters (ω, b) if for all t such that $|t| \le 1/b$,*

$$\mathbb{E}\big[\exp\big(t(X - \mathbb{E}[X])\big)\big] \le \exp\big(\frac{t^2\omega^2}{2}\big). \tag{29}$$

When $b = 0$, we interpret $1/0$ as being the same as ∞, it follows immediately from this definition that any sub-Gaussian random variable is also sub-exponential. There are also a variety of other conditions equivalent to sub-exponentiality, which we relate by defining the sub-exponential norm of random variable. In particular, we define the sub-exponential norm (sometimes known as the ψ_1-Orlicz in the literature) as

$$\|X\|_{\psi_1} := \sup_{q \geq 1} \frac{1}{q} (\mathbb{E}[|X^q|])^{1/q}.$$

Then we have the following lemma which provides several equivalent characterizations of sub-exponential random variables.

Theorem 26 *(Equivalence of sub-exponential properties (R. Vershynin, 2010))*
Let X be a random variable and $\omega > 0$ be a constant. Then, the following properties are all equivalent with suitable numerical constants $K_i > 0, i = 1, \ldots, 4$, that are different from each other by at most an absolute constant $\mathcal{C}$, meaning that if one statement (i) holds with parameter K_i, then the statement (j) holds with parameter $K_j \leq \mathcal{C} K_i$.

(1) sub-exponential tails: $\mathbb{P}[|X| > t] \leq \exp\left(1 - \frac{t}{\omega K_1}\right)$, for all $t \geq 0$.

(2) sub-exponential moments: $(\mathbb{E}[|X^q|])^{1/q} \leq K_2 \omega q$, for all $q \geq 1$.

(3) existence of moment generating function (Mgf): $\mathbb{E}\left[\exp\left(\frac{X}{\omega K_3}\right)\right] \leq e$.

Note that in each of the statements of Theorem 26, we may replace ω by $\|X\|_{\psi_1}$ and, up to absolute constant factors, $\|X\|_{\psi_1}$ is the smallest possible number in these inequalities.

Lemma 27 *(Mgf of sub-exponential random variables (R. Vershynin, 2010)) Let X be a centered sub- exponential random variable. Then, for t such that $|t| \leq c/\|X\|_{\psi_1}$, one has*

$$\mathbb{E}[\exp(tX)] \leq \exp(Ct^2 \|X\|_{\psi_1}^2)$$

where $C, c > 0$ are absolute constants.

Lemma 28 *For all $v \in [V]$ and $(i,j) \in [d_u] \times [d_v]$, the random variable $X_{i,j}^v$ is a sub-exponential with parameters (U_γ, K), where K is defined in Assumption 2. Moreover, we have that $\|X_{i,j}^v\|_{\psi_1} = \mathcal{C}(U_\gamma \vee K)$ for some absolute constant $\mathcal{C}$.*

Proof Let t such that $|t| \leq 1/K$, then

$$\begin{aligned}
&\mathbb{E}[\exp\left(t(X_{ij}^v - \mathbb{E}[X_{ij}^v])\right)] \\
&\quad = e^{-t(G^v)'(M_{ij}^v)} \int_{\mathbb{R}} h^v(x) \exp\left((t + M_{ij}^v)x - G^v(M_{ij}^v)\right) dx \\
&\quad = e^{G^v(t+M_{ij}^v) - G^v(M_{ij}^v) - t(G^v)'(M_{ij}^v)} \int_{\mathbb{R}} h^v(x) \exp\left((t + M_{ij}^v)x - G^v(t + M_{ij}^v)\right) dx \\
&\quad = e^{G^v(t+M_{ij}^v) - G^v(M_{ij}^v) - t(G^v)'(M_{ij}^v)},
\end{aligned}$$

where we used in the last inequality the fact that that $\int_{\mathbb{R}} h^v(x) \exp\left((t + M_{ij}^v)x - G^v(t + M_{ij}^v)\right) dx = \int_{\mathbb{R}} f_{h^v, G^v}(X_{i,j}^v | t + M_{ij}^v) dx = 1$. Therefore, an ordinary Taylor series expansion

of G^v implies that there exists $t_{\gamma,K} \in [-\gamma - \frac{1}{K}, \gamma + \frac{1}{K}]$ such that $G^v(t + M_{ij}^v) - G^v(M_{ij}^v) - t(G^v)'(M_{ij}^v) = (t^2/2)(G^v)''(t_{\gamma,K}^2)$. By Assumption 2, we obtain

$$\mathbb{E}[\exp\left(t(X_{ij}^v - \mathbb{E}[X_{ij}^v])\right)] \leq \exp\left(\frac{t^2 U_\gamma^2}{2}\right).$$

Using Lemma 27, we get $\|X_{i,j}^v\|_{\psi_1} = \mathcal{C}(U_\gamma \vee K)$ for some absolute constant $\mathcal{C}$. This proves Lemma 28. ■

References

D. Agarwal, L. Zhang, and R. Mazumder. Modeling itemitem similarities for personalized recommendations on yahoo! front page. *Ann. Appl. Stat.*, 5(3):1839–1875, 2011.

P. Alquier, V. Cottet, and G. Lecué. Estimation bounds and sharp oracle inequalities of regularized procedures with Lipschitz loss functions. *arXiv:1702.01402*, 2017.

M.G. Armentano, D. Godoy, and A. A. Amandi. Followee recommendation based on text analysis of micro-blogging activity. *Information Systems*, 38(8):1116 – 1127, 2013. ISSN 0306-4379.

A. S. Bandeira and R. van Handel. Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *Ann. Probab.*, 44(4):2479–2506, 2016.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, 2005.

P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe. Large margin classifiers: Convex loss, low noise, and convergence rates. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1173–1180. MIT Press, 2004.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Know.-Based Syst.*, 46:109–132, 2013.

G. Bouchard, D. Yin, and S. Guo. Convex collective matrix factorization. In *AISTATS*, volume 31 of *JMLR Workshop and Conference Proceedings*, pages 144–152. JMLR.org, 2013.

O. Bousquet. A bennett concentration inequality and its application to suprema of empirical processes. *Comptes Rendus Mathematique*, 334(6):495 – 500, 2002.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217, 1967.

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer Berlin Heidelberg, 2011.

J.-F. Cai., E. J. Cands, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

T. Cai and W. X. Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *J. Mach. Learn. Res.*, 14(1):3619–3647, 2013.

T. Tony Cai and Wen-Xin Zhou. Matrix completion via max-norm constrained optimization. *Electron. J. Statist.*, 10(1):1493–1525, 2016.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, 2009.

E. J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

I. Cantador, I. Fernández-Tobías, Shlomo Berkovsky, and P. Cremonesi. *Cross-Domain Recommender Systems*, pages 919–959. Springer US, Boston, MA, 2015.

Y. Censor and S.A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, USA, 1997.

K.-Y. Chiang, C.-J. Hsieh, and I. S. Dhillon. Matrix completion with noisy side information. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3447–3455, Cambridge, MA, USA, 2015. MIT Press.

K.-Y. Chiang, I. S. Dhillon, and C.-J. Hsieh. Using side information to reliably learn low-rank matrices from missing and corrupted observations. *Journal of Machine Learning Research*, 19(76):1–35, 2018.

M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters. 1-bit matrix completion. *Information and Inference: A Journal of the IMA*, 3(3):189, 2014.

P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangant, R. Virrankoski, and A. Savvides. Distance matrix reconstruction from incomplete distance information for sensor network localization. In *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, volume 2, pages 536–544, 2006.

A. Elsener and S. van de Geer. Robust low-rank matrix estimation. *Ann. Statist.*, 46(6B): 3481–3509, 12 2018.

M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 6, pages 4734–4739 vol.6, 2001.

W. Fithian and R. Mazumder. Flexible low-rank statistical modeling with missing data and side information. *Statist. Sci.*, 33(2):238–260, 2018.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

S. Gunasekar, P. Ravikumar, and J. Ghosh. Exponential family matrix completion under structural constraints. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1917–II–1925. JMLR.org, 2014.

S. Gunasekar, M. Yamada, D. Yin, and Y. Chang. Consistent collective matrix completion under joint low rank structure. In *AISTATS*, 2015.

S. Gunasekar, J. C. Ho, J. Ghosh, S. Kreml, A. N. Kho, J. C Denny, B. A. Malin, and J. Sun. Phenotyping using structured collective matrix factorization of multi–source ehr data. *preprint arXiv:1609.04466*, 2016.

N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2): 217–288, 2011.

J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 199–206, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-906-0.

S. Horii, T. Matsushima, and S. Hirasawa. A note on the correlated multiple matrix completion based on the convex optimization method. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1618–1623, 2014.

Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, 2013.

P. Jain and I. S. Dhillon. Provable inductive matrix completion. *CoRR*, abs/1306.0626, 2013.

M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 135–142, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-906-0.

S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 457–464, New York, NY, USA, 2009a. ACM.

S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 457–464, New York, NY, USA, 2009b. ACM.

O. Klopp. Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*, 20(1):282–303, 2014.

O. Klopp. Matrix completion by singular value thresholding: Sharp bounds. *Electron. J. Statist.*, 9(2):2348–2369, 2015.

O. Klopp, J. Lafond, E. Moulines, and J. Salmon. Adaptive multinomial matrix completion. *Electron. J. Statist.*, 9(2):2950–2975, 2015.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

J. Lafond. Low rank matrix completion with exponential family noise. In Peter Grnwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 1224–1243, Paris, France, 2015. PMLR.

X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2Nd International Conference on Ubiquitous Information Management and Communication*, ICUIMC '08, pages 208–211, New York, NY, USA, 2008. ACM.

R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization, 1998.

E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer-Verlag, New York, NY, USA, second edition, 1998.

Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.*, 31(3):1235–1256, 2009.

Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2010.

H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 287–296, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0493-1.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.*, 11:2287–2322, 2010.

S. Mendelson. Obtaining fast error rates in nonconvex situations. *Journal of Complexity*, 24(3):380 – 397, 2008.

N. Natarajan and I. S. Dhillon. Inductive matrix completion for predicting genedisease associations. *Bioinformatics*, 30(12):i60–i68, 06 2014. ISSN 1367-4803.

N. Natarajan, D. Shin, and I. S. Dhillon. Which app will you use next?: Collaborative filtering with interactional context. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 201–208, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2409-0.

S. Negahban and M. J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Ann. Statist.*, 39(2):1069–1097, 2011.

Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

S. Oh, A. Montanari, and A. Karbasi. Sensor network localization from local connectivity: Performance analysis for the mds-map algorithm. In *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, pages 1–5, 2010.

J. Pagès. *Multiple Factor Analysis by Example Using R*. Chapman & Hall/CRC The R Series. Taylor & Francis, 2014. ISBN 9781482205473.

N. Parikh and S. Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, 2014.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.

J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 713–719, New York, NY, USA, 2005. ACM.

R.Vershynin. Introduction to the non-asymptotic analysis of random matrices. *CoRR*, abs/1011.3027, 2010.

S. Shin, D.and Cetintas and I. S. Lee, K.-C.and Dhillon. Tumblr blog recommendation with boosted inductive matrix completion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 203–212, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6.

A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 650–658, New York, NY, USA, 2008. ACM.

A. P. Singh and G. J. Gordon. A bayesian matrix factorization model for relational data. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, pages 556–563, Arlington, Virginia, United States, 2010. AUAI Press.

A. M.-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 405–414, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization, 2005.

M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Found. Trends Mach. Learn.*, 9(1):1–118, June 2016. ISSN 1935-8237.

S. van de Geer. *Estimation and Testing Under Sparsity: École d'Été de Probabilités de Saint-Flour XLV – 2015*. Lecture Notes in Mathematics. Springer International Publishing, 2016.

L. Xu, Z. Chen, Q. Zhou, E. Chen, N. J. Yuan, and X. Xie. Aligned matrix completion: Integrating consistency and independency in multiple domains. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 529–538, 2016.

M. Xu, R. Jin, and Z.-H. Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2301–2309, USA, 2013. Curran Associates Inc.

Q. Yao and J. T. Kwok. Accelerated inexact soft-impute for fast large-scale matrix completion. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 4002–4008. AAAI Press, 2015.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.*, 32(1):56–85, 2004.

M. Zitnik and B. Zupan. Matrix factorization-based data fusion for gene function prediction in baker's yeast and slime mold. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 400–11, 2014.

M. Zitnik and B. Zupan. Data fusion by matrix factorization. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(1):41–53, 2015. ISSN 0162-8828.

Journal of Machine Learning Research 20 (2019) 1-24 Submitted 7/18; Revised 4/19; Published 9/19

On Asymptotic and Finite-Time Optimality of Bayesian Predictors

Daniil Ryabko DANIIL@RYABKO.NET

Editor: Csaba Szepesvari

Abstract

The problem is that of sequential probability forecasting for finite-valued time series. The data is generated by an unknown probability distribution over the space of all one-way infinite sequences. Two settings are considered: the realizable and the non-realizable one. Assume first that the probability measure generating the sequence belongs to a given set $\mathcal{C}$ (realizable case), but the latter is completely arbitrary (uncountably infinite, without any structure given). It is shown that the minimax asymptotic average loss—which may be positive—is always attainable, and it is attained by a Bayesian predictor whose prior is discrete and concentrated on $\mathcal{C}$. Moreover, the finite-time loss of the Bayesian predictor is also optimal up to an additive $\log n$ term (where n is the time step). This upper bound is complemented by a lower bound that goes to infinity but may do so arbitrarily slow.

Passing to the non-realizable setting, let the probability measure generating the data be arbitrary, and consider the given set $\mathcal{C}$ as a set of experts to compete with. The goal is to minimize the regret with respect to the experts. It is shown that in this setting it is possible that all Bayesian strategies are strictly suboptimal even asymptotically. In other words, a sublinear regret may be attainable but the regret of every Bayesian predictor is linear.

A very general recommendation for choosing a model can be made based on these results: it is better to take a model large enough to make sure it includes the process that generates the data, even if it entails positive asymptotic average loss, for otherwise any combination of predictors in the model class may be useless.

Keywords: sequence prediction, Bayesian prediction, complete-class theorems, minimax theorems

1. Introduction

[1] Given a sequence $x_1, \dots, x_n$ of observations $x_i \in \mathcal{X}$, where $\mathcal{X}$ is a finite set, it is required to predict the probabilities of observing $x_{n+1} = x$ for each $x \in \mathcal{X}$, before x_{n+1} is revealed, after which the process continues sequentially. The problem is considered in full generality; in particular, outcomes may exhibit arbitrary dependence. This and related problems arise in a variety of applications, where the data may be financial, such as a sequence of stock prices; human-generated, such as a written text or a behavioural sequence; biological (DNA sequences); physical measurements and so on.

1. Some of the results in this paper were reported at ALT'17 and ALT'16 (Ryabko, 2017, 2016). All the non-asymptotic results (upper and lower bounds) in this paper are new.

©2019 Daniil Ryabko.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-512.html.

The probabilities forecast by a predictor ρ having seen a sequence $x_1, \dots, x_n$ may be interpreted as conditional probabilities of x_{n+1} given $x_1, \dots, x_n$. Since the latter sequence is arbitrary, the predictor ρ defines a probability measure over the space of all one-way infinite sequences $\mathcal{X}^\infty$ (with the usual sigma-algebra generated by cylinder sets). Similarly, the mechanism that generates the data, that is, the observed sequence $x_1, \dots, x_n, \dots$, can be assumed probabilistic. Thus, predictors and the data-generating mechanism are objects of the same kind: probability measures of the space of one-way infinite sequences.

It is easy to see that without any model for the data nor for the predictor it is impossible to do meaningful inference for this task. A model can be seen as a set $\mathcal{C}$ of distributions over the space of one-way infinite sequences. There are two common approaches to use a model. The first one is to assume that the probability measure generating the sequence belongs to the model set $\mathcal{C}$. The goal is then to have a predictor whose (time-average) error, or *loss*, decreases (to zero), as fast as possible, as more and more data is revealed. The other approach is to suppose that the data may be an arbitrary individual sequence, and the model set $\mathcal{C}$ is considered a set of experts. The goal then is to have a predictor whose *regret* with respect to using the best expert in the set for the given sequence is as small as possible, and at least sublinear (average regret goes to zero as more data is revealed).

Here I would like to argue in favour of a third approach. It consists in making the model set $\mathcal{C}$ large enough to include the distribution that generates the data, even if it means that the best achievable asymptotic loss is strictly positive. In particular, whichever aspect of the data-generation process is completely unknown can be modelled by an arbitrary sequence.

Throughout the paper, the loss is chosen to be the Kullback-Leibler divergence (expected log-loss) mainly for the sake of convenience, but also reflecting its importance in applications. To fix ideas, consider a simple and familiar example (perhaps first considered by Willems 1996 in a somewhat different formulation) in which the best asymptotic average loss is non-zero: a piece-wise i.i.d. sequence. The sequence consists of segments, in each of which the data is i.i.d., but the distribution in each of the segments is otherwise arbitrary, and the sequence of points at which one segment ends and the next starts is completely arbitrary as well. Thus, one can learn the distribution within each segment but cannot predict when or to what the next change will be. If the frequency of these change points does not vanish then the minimax asymptotic average error is positive. This example is considered in more detail in Section 4.

Two results are presented to support the suggested approach to the general problem of sequential prediction.

The **first result** shows that, no matter how big a model class $\mathcal{C}$ is, if the distribution generating the data belongs to $\mathcal{C}$ then the optimal minimax asymptotic average performance, even if positive, is always achievable, and it can be attained by a Bayesian combination of countably many distributions in $\mathcal{C}$. One cannot, in general, say anything about the speed of convergence of the average loss, as it depends on $\mathcal{C}$ and may be arbitrarily slow for some classes $\mathcal{C}$. However, for any predictor ρ whatsoever, its performance can be matched by a Bayesian predictor up to an additive $O(\log n)$ term, with only rather small constants hidden inside the $O()$ term.

This means that, if the data-generating mechanism belongs to the model class, then no matter how big the latter is, the statistician knows where to start: it suffices to find the right prior distribution, and then the inference can be made by evaluating the posterior

with respect to the observed data. The performance guarantees would be pointwise (for every distribution in $\mathcal{C}$) rather than Bayesian (with prior probability 1 or in expectation), and it is possible to achieve nearly the best speed of decrease of the average loss. This adds a perspective to the classical results (Freedman, 1963, 1965; Diaconis and Freedman, 1986) that show that there are priors with which Bayesian inference is inconsistent even for problems on i.i.d. data (although, in that latter case, for infinite alphabets).

In some more detail, consider an arbitrary set $\mathcal{C}$ of probability distributions over one-way infinite sequences. Let $L_n(\mu,\nu)$ be the cumulative expected log loss (KL divergence) of ν used to predict μ, up to time n. A Bayesian predictor is a predictor of the form $\int \alpha dW(\alpha)$ where W is a probability distribution over $\mathcal{C}$ (a prior), that is, $W(\mathcal{C})=1$. The prediction is simply by evaluating the posterior over the given sequence $x_1,\dots,x_n$. If the prior is discrete (and so the integral is a sum) then we call such a predictor discrete Bayesian. While the general definition requires the structure of a probability space on $\mathcal{C}$ (or at least some subset of $\mathcal{C}$), for discrete Bayesian predictors no such structure is required. For the main result below, discrete Bayesian predictors are sufficient, so we do not need to worry about the measurability of $\mathcal{C}$. The main result is the following.

Theorem 1 *For every set $\mathcal{C}$ of probability measures and for every predictor ρ there is a discrete Bayesian predictor ν such that for every $\mu\in\mathcal{C}$ we have*

$$L_n(\mu,\nu)\le L_n(\mu,\rho)+8\log n+O(\log\log n).$$

There are no assumptions whatsoever needed for this result to hold. As mentioned above, the set $\mathcal{C}$ is not even required to be measurable, and the distributions in $\mathcal{C}$ may be completely arbitrary (no restrictions on dependence, memory, etc.). The constants in $O()$ are small and, besides absolute constants, only include linear dependence on the alphabet size $|\mathcal{X}|$.

To the author's knowledge, the upper bound presents the first non-asymptotic result in a setting of this generality.

A lower bound is also obtained, which takes the form of $\theta(n)$ where θ increases to infinity, but may do so arbitrarily slow. Thus, there exist a set $\mathcal{C}$ for which by choosing to be Bayesian a statistician would suffer a more-than-constant cumulative regret. This is made precise in Section 3.2.

It is also useful to put this result into the context of the **decision theory**; from this point of view, it can be seen (albeit with some important differences) as a *complete-class theorem* for asymptotic average performance: it shows that the class of Bayes strategies is always essentially complete. Decision theory typically deals with single-shot games and general losses (here we only consider expected average KL divergence), and Wald's complete class theorem and its generalizations require a number of conditions to hold. Other important differences, besides the absence of any conditions in Theorem 1, include the fact that in our case all strategies are inadmissible and one cannot speak about minimal complete classes. To further understand the game-theoretic side of the main result, it is useful to consider the minimax asymptotic average loss $V_{\mathcal{C}}$, defined as

$$\inf_{\rho}\sup_{\mu\in\mathcal{C}}\limsup_{n\to\infty}\frac{1}{n}L_n(\mu,\rho).$$

It can be shown that the theorem above implies that this value can be attained by a Bayesian predictor with a discrete prior. Besides providing a complete-class theorem for

(asymptotic) prediction, this result can also be seen as a partial minimax theorem: for every set of strategies of the opponent, the statistician has a minimax strategy. For the "classical" case of $V_{\mathcal{C}}=0$, that is, for the case when the average error of the best predictor vanishes, an analogous asymptotic result was obtained by Ryabko (2010). We shall see that the general case presents principled differences (not limited to the proofs). In particular, as mentioned above, in the case $V_{\mathcal{C}}>0$ all strategies are inadmissible and there are no minimal complete classes, while if $V_{\mathcal{C}}=0$ then every Bayes rule that achieves it is admissible and constitutes a minimal complete class. These results and implications are considered in Section 5.

The **second result** presented concerns regret minimization. That is, we turn the tables, and assume that the data-generating mechanism is unrestricted, in particular it can be an arbitrary (deterministic) sequence. The set of probability measures $\mathcal{C}$ is now the set of experts, and the goal of the statistician is to find a strategy that is as good as the best one in $\mathcal{C}$, for the given data. Now that the distribution generating the data may not be in $\mathcal{C}$, it may happen that every combination of the experts has strictly positive asymptotic average regret, even though 0 asymptotic average regret is achievable. The formal result concerns Bayesian combinations, but it is argued in Section 6 that it applies more generally.

Theorem 2 *There exists a measurable set $\mathcal{C}$ such that every Bayesian predictor with prior over $\mathcal{C}$ has a linear regret with respect to $\mathcal{C}$, while there exists a predictor ρ whose regret is sublinear.*

Putting the **two results together**, in the realizable case of the sequence prediction problem the best asymptotic performance can always be attained by a Bayesian strategy, while in the non-realizable case it is possible that all Bayesian strategies are strictly sub-optimal. We can therefore make the following fundamental recommendation for choosing a model for sequential data:

Better take a model large enough to make sure it includes the process that generates the data, even if it makes the worst-case asymptotic error larger than zero, for otherwise any combination of predictors in the model class may be useless.

Thus, the results presented invite a reconsideration of the familiar trade-off in model selection: the model must be large enough to describe the data but small enough to be learnable. In reality, the first part is often eschewed in favour of the second: the model does not include the data-generating mechanism, but allows for learning. In the context of sequence prediction, learnability of the model is usually understood as allowing for the average error or regret of the predictor to go to zero. In light of the presented results, it can be suggested to change the preference to the other side of the trade-off, namely, make the model large enough to include the data, since having a linearly increasing loss does not preclude one from finding the best predictor. On the other hand, not having the data-generating mechanism inside the model may mean that there is no optimal Bayesian predictor, or, as argued below in Section 6, any combination of predictors in the model class. Of course, this recommendation is not relevant for the specific cases where one knows how to find an optimal predictor that does not have such a form.

1.1. Related Work

The literature on sequential prediction is vast and spread between several fields: nonparametric statistics, machine learning, information theory and econometrics. While an ex-

haustive survey is beyond the scope of this work, some related results and some important differences in the settings are worth mentioning. Most of the literature deals with specific families of processes $\mathcal{C}$, and almost exclusively with the case $V_{\mathcal{C}} = 0$. For the latter case, an asymptotic analogue of Theorem 1 was established by Ryabko (2010). An important exception from the $V_{\mathcal{C}} = 0$ case is the problem of prediction of processes with abrupt changes mentioned above, even though the formulation considered previously does not introduce the value $V_{\mathcal{C}}$, since the loss of the predictor is measured with respect to the best predictor that knows the actual changes (expert-advise formulation); see, e.g., (Gyorgy et al., 2012) and references. Perhaps the most general specific classes of processes considered in the literature are those of stationary ergodic distributions (Ryabko, 1988; Morvai et al., 1997; Gyorfi and Ottucsak, 2007).

The problem of regret minimization in prediction is usually considered in a slightly different setting, called prediction with expert advice; an overview is provided by Cesa-Bianchi and Lugosi (2006). One of the key differences is that the loss is measured with respect to actual outcomes rather than probabilities, making the setting non-probabilistic even though the predictions are usually randomized. An attempt to put the two settings under the same formulation and thus making the results directly comparable has been taken by Ryabko (2011), which also poses the question that Theorem 2 answers.

The econometrics literature on the subject concerns the $V_{\mathcal{C}} = 0$ case (with a different loss) and studies sufficient conditions on the existence of consistent predictors (loss goes to 0 asymptotically), mostly in the Bayesian setting, which means that the results are with prior probability 1 and not pointwise; see (Kalai and Lehrer, 1994; Noguchi, 2015) and references.

Related decision-theoretic results concern the setting of the problem for "predicting" just one (the first) symbol of the sequence. For KL divergence (expected log loss) these results include (Ryabko, 1979; Gallager, 1976-1979; Haussler, 1997).

1.2. Organization

The rest of the paper is organized as follows. The next section introduces notation and definitions. Section 3 presents the main asymptotic and finite-time results on the existence of Bayesian prediction for minimizing loss, with Section 3.2 providing the lower bound. Section 4 provides examples of classes for which $V_{\mathcal{C}} > 0$, illustrating the main findings. Decision-theoretic interpretations of the asymptotic result are given in section 5. Section 6 presents the impossibility result for regret minimization, and Section 7 concludes.

2. Notation and Definitions

Let $\mathcal{X}$ be a finite set (the alphabet), and let

$$M := \log |\mathcal{X}|. \tag{1}$$

The notation $x_{1..n}$ is used for $x_1, \dots, x_n$. $\mathbb{N}$ is the set of naturals without 0; for a finite set A denote $|A|$ its cardinality, and $A^* := \cup_{k\in\mathbb{N}} A^k$. All logarithms are base 2. We use $\mathbf{E}_\mu$ for expectation with respect to a probability measure μ. We consider probability measures on $(\mathcal{X}^\infty, \mathcal{F})$, where $\mathcal{F}$ is the usual Borel sigma-field generated by cylinder sets (see below). The set of probability measures over $(\mathcal{X}^\infty, \mathcal{F})$ is denoted $\mathcal{P}(\mathcal{X}^\infty, \mathcal{F})$ or $\mathcal{P}$ for short.

A probability measure ρ is a *discrete Bayesian predictor* with a prior over $\mathcal{C}$ if $\rho = \sum_{k\in\mathbb{N}} w_k \mu_k$, for some measures $\mu_k \in \mathcal{C}$, $k \in \mathbb{N}$, where w_k are real weights that sum to 1. The latter weights can be seen as a prior distribution over (a subset of) $\mathcal{C}$.

In Section 6 we need a more general definition of Bayesian predictors, which allows for non-discrete priors, and thus require a structure of the probability space on $\mathcal{P}$. Here we shall define it in a standard way, following Gray (1988). Recall that the sigma-algebra $\mathcal{F}$ of the space of infinite sequences $(X^\infty, \mathcal{F})$ can be generated by the countable set $(B_i)_{i\in\mathbb{N}}$ of cylinders, $B_i := \{\mathbf{x} \in \mathcal{X}^\infty : x_{1..|b_i|} = b_i\}$ where the words b_i take all possible values in $|\mathcal{X}|^*$. Next, consider the countable set of sets $\{\nu \in \mathcal{P} : \nu(B_i) \in u\}$, where $u \subset [0, 1]$, obtained by taking all $B_i, i \in \mathbb{N}$ and all intervals u with rational endpoints. This set generates a sigma-algebra $\mathcal{F}'$. Denoting $\mathcal{P}'$ the set of all probability measures over $\mathcal{P}$ we obtain the measurable space $(\mathcal{P}', \mathcal{F}')$.

Associated with any probability measure $W \in \mathcal{P}'$ there is a probability measure $\rho_W \in \mathcal{P}$ defined by $\rho_W = \int_P \alpha dW(\alpha)$ (*barycentre*, in the terminology of Gray (1988); see the latter work for a detailed exposition). A measure $\rho \in \mathcal{P}$ is *Bayesian* for a set $\mathcal{C} \subset \mathcal{P}$ if $\rho = \rho_W$ for some $W \in \mathcal{P}'$ such that there exists a measurable set $\bar{\mathcal{C}} \subset \mathcal{C}$ with $W(\bar{\mathcal{C}}) = 1$. (The subset $\bar{\mathcal{C}}$ is necessary since $\mathcal{C}$ may not be measurable.) The reason behind the name is that W can be seen as a prior over $\mathcal{C}$, and ρ_W as a predictor whose predictions are simply by evaluating the posterior distribution $\rho_W(\cdot|x_1, \dots, x_n)$ where $x_1, \dots, x_n$ are observations up to time n.

2.1. Loss

For two probability measures μ and ρ introduce the *expected cumulative Kullback-Leibler divergence (KL divergence)* as

$$L_n(\mu, \rho) := \mathbf{E}_\mu \sum_{t=1}^{n} \sum_{a\in\mathcal{X}} \mu(x_t = a|x_{1..t-1}) \log \frac{\mu(x_t = a|x_{1..t-1})}{\rho(x_t = a|x_{1..t-1})} = \sum_{x_{1..n}\in\mathcal{X}^n} \mu(x_{1..n}) \log \frac{\mu(x_{1..n})}{\rho(x_{1..n})}. \tag{2}$$

In words, we take the μ-expected (over data) average (over time) KL divergence between μ- and ρ-conditional (on the past data) probability distributions of the next outcome; and this gives simply the μ-expected log-ratio of the likelihoods. Here μ will be interpreted as the distribution generating the data.

The asymptotic average expected loss is then defined as

$$\bar{L}(\nu, \rho) := \limsup_{n\to\infty} \frac{1}{n} L_n(\nu, \rho),$$

where the upper limit is chosen so as to reflect the worst performance over time. One can define the worst-case performance of a strategy ρ by

$$\bar{L}(\mathcal{C}, \rho) := \sup_{\mu\in\mathcal{C}} \bar{L}(\mu, \rho)$$

and the minimax value by

$$V_{\mathcal{C}} := \inf_{\rho\in\mathcal{P}} \bar{L}(\mathcal{C}, \rho). \tag{3}$$

Some examples of calculating the latter value for different sets $\mathcal{C}$ are considered in Section 4; the most common case in the literature is $V_{\mathcal{C}} = 0$, which is the case, for example, if $\mathcal{C}$ is the set of all i.i.d. or all stationary distributions.

2.2. Regret

Switching the roles, assume that the set of strategies of the opponent is unrestricted; the set of probability measures $\mathcal{C} \subset \mathcal{P}(X^\infty, \mathcal{F})$ is now the set of experts, and the goal of the statistician is to find a strategy that is as good as the best one in $\mathcal{C}$, for the given data. Thus, we are interested in the (asymptotic) *regret*

$$\bar{R}^\nu(\mu, \rho) := \limsup_{n\to\infty} \frac{1}{n} \left[L_n(\nu, \rho) - L_n(\nu, \mu) \right],$$

of using ρ as opposed to μ on the data generated by ν. The goal is to find ρ that minimizes the worst-case (over data) regret with respect to the best expert from the given set $\mathcal{C}$:

$$R(\mathcal{C}, \rho) := \sup_{\nu\in\mathcal{P}} \sup_{\mu\in\mathcal{C}} \bar{R}^\nu(\mu, \rho).$$

Note than in the expert-advice literature the regret is typically defined on finite sequences of length n, thus allowing both the experts and the algorithms to depend on n explicitly.

Similarly to $V_\mathcal{C}$, we can now define the value

$$U_\mathcal{C} := \inf_{\rho\in\mathcal{P}} R(\mathcal{C}, \rho),$$

which is the worst-case asymptotic average regret with respect to the set of experts $\mathcal{C}$.

In view of the (negative) result that is obtained for regret minimization, here we are mostly concerned with the case $U_\mathcal{C} = 0$.

3. Realizable Case: Minimizing Loss, Optimality of Bayes Rules

The main result (Theorem 1) shows that for any predictor ρ there is a Bayesian predictor that is as good as ρ, up to a $O(\log n)$ loss. It follows (Corollary 2) that the minimax loss is always achievable and is achieved by a discrete Bayesian predictor — without any assumptions on $\mathcal{C}$.

Theorem 1 (upper bound on the best Bayesian) *Let $\mathcal{C}$ be any set of probability measures on $(\mathcal{X}^\infty, \mathcal{F})$, and let ρ be another probability measure on this space, considered as a predictor. Then there is a discrete Bayesian predictor ν, that is, a predictor of the form $\sum_{k\in\mathbb{N}} w_k\mu_k$ where $\mu_k \in \mathcal{C}$ and $w_k \in [0, 1]$, such that for every $\mu \in \mathcal{C}$ we have*

$$L_n(\mu, \nu) - L_n(\mu, \rho) \le 8 \log n + O(\log\log n), \tag{4}$$

where the constants in $O(\cdot)$ are small and are given in (26) *using the notation defined in* (1), (6), (20) *and* (27). *The dependence on the alphabet size, M, is linear ($M \log\log n$) and the rest of the constants are universal.*

The proof is given after the corollary.

Corollary 2 (asymptotic optimality of Bayesian predictors) *For any set $\mathcal{C}$ of probability measures on $(\mathcal{X}^\infty, \mathcal{F})$, there exist a discrete Bayesian predictor φ such that*

$$\bar{L}(\mathcal{C}, \varphi) = V_\mathcal{C}.$$

Proof Note that the statement does not immediately follow from (4), because ρ in (4) may be such that $\sup_{\mu\in\mathcal{C}} \bar{L}(\mu, \rho) > V_\mathcal{C}$. Thus, let $\gamma_j > V_\mathcal{C}$, $j \in \mathbb{N}$ be a non-increasing sequence such that $\lim_{j\to\infty} \gamma_j = V_\mathcal{C}$. By the definition (3) of $V_\mathcal{C}$, it is possible to find a sequence $\rho_j \in \mathcal{P}$ such that $\bar{L}(\mathcal{C}, \rho_j) \le \gamma_j$ for all $j \in \mathbb{N}$. From Theorem 1 we conclude that for each ρ_j, $j \in \mathbb{N}$ there is a probability measure ν_j of the form $\sum_{k\in\mathbb{N}} w'_k\mu_k$, where $\mu_k \in \mathcal{C}$ such that $\bar{L}(\mathcal{C}, \nu_j) \le \bar{L}(\mathcal{C}, \rho_j)$. It remains to define $\varphi := \sum_{j\in\mathbb{N}} w_j\nu_j$, where w_j are positive and sum to 1. Clearly, φ is a discrete Bayesian predictor. Let us show that for every $j \in \mathbb{N}$ it satisfies

$$\bar{L}(\mathcal{C}, \varphi) \le \bar{L}(\mathcal{C}, \rho_j). \tag{5}$$

Indeed, for every $\mu \in \mathcal{C}$ and every $j \in \mathbb{N}$

$$L_n(\mu, \varphi) = E_\mu \log \frac{\mu(x_{1..n})}{\varphi(x_{1..n})} \le E_\mu \log \frac{\mu(x_{1..n})}{\nu_j(x_{1..n})} - \log w_j,$$

so that $\bar{L}(\mu, \varphi) \le \bar{L}(\mu, \nu_j) \le \bar{L}(\mu, \rho_j) \le \gamma_j$, establishing (5). Finally, recall that $\gamma_j \to V_\mathcal{C}$ to obtain the statement of the corollary. ■

3.1. Proof of Theorem 1

Before giving the proof of the theorem, let us briefly expose the main ideas behind it. Assume for a moment that, for each $\mu \in \mathcal{C}$, the limit $\lim_{n\to\infty} \frac{1}{n} \log \frac{\mu(x_{1..n})}{\rho(x_{1..n})}$ exists for μ-almost all $\mathbf{x} = x_1, \ldots, x_n, \cdots \in \mathcal{X}^\infty$, where ρ is the predictor given to compare to. Then we could define (μ-almost everywhere) the function $f_\mu(\mathbf{x})$ whose value at $\mathbf{x}$ equals this limit. Let us call it the "log-density" function. What we would be looking for thence is to find a countable dense subset of the set of log-densities of all probability measures from $\mathcal{C}$. The measures μ corresponding to each log-density in this countable set would then constitute the sequence whose existence the theorem asserts. To find such a dense countable subset we could employ a standard procedure: approximate all log-densities by step functions with finitely many steps. The main technical argument is then to show that, for each level of the step functions, there are not too many of these functions whose steps are concentrated on different sets of non-negligible probability, for otherwise the requirement that ρ attains $V_\mathcal{C}$ would be violated. Here "not too many" means exponentially many with the right exponent (the one corresponding to the step of the step-function with which we approximate the density), and "non-negligible probability" means a probability bounded away (in n) from 0. In reality, what we do instead in the proof is use the step-functions approximation at each time step n. Since there are only countably many time steps, the result is still a countable set of measures μ from $\mathcal{C}$. Before going further, note that constructing a predictor for each n does not mean constructing the best predictors up to this time step: in fact, taking a predictor that is minimax optimal up to n, for each n, and summing these predictors up (with weights) for all $n \in \mathbb{N}$ may result in the worst possible predictor overall, and in particular, a one much worse than the predictor ρ given. An example of this behaviour is given in the proof of Theorem 3 (the lower bound). The objective for each n is different, and it is to approximate the measure ρ up to this time step with measures from $\mathcal{C}$. For each n, we consider a covering of the set $\mathcal{X}^n$ with subsets, each of which is associated with a

measure μ from $\mathcal{C}$. These latter measures are then those the prior is concentrated on (that is, they are summed up with weights). The covering is constructed as follows. The log-ratio function $\log \frac{\mu(x_{1..n})}{\rho(x_{1..n})}$, where ρ is the predictor whose performance we are trying to match, is approximated with a step function for each μ, and for each size of the step. The cells of the resulting partition are then ordered with respect to their ρ probability. The main part of the proof is then to show that not too many cells are needed to cover the set $\mathcal{X}^n$ this way up to a small probability. Quantifying the "not too many" and "small" parts results in the final bound.

It is worth noting that the proof that Ryabko (2010) obtains for the special case $V_{\mathcal{C}} = 0$, does not directly generalize. In fact, tidying up the constants in that proof, one only obtains the asymptotic loss of $2V_{\mathcal{C}}$ for the mixture predictor presented there. It is not a problem for the case $V_{\mathcal{C}} = 0$, but of course is not what we want in the general case. The reason behind this problem is that for the construction in the proof of Ryabko (2010) one can only use the fact that each of the measures μ_k in the sequence is as good as the predictor ρ whose existence is assumed (the one that attains $V_{\mathcal{C}} = 0$). In contrast, in the proof below we are able to use the fact that each measure in the sequence is in fact much better than ρ on some subsets of $\mathcal{X}^n$.

Proof [of Theorem 1.] Define the weights w_k as follows: $w_1 := 1/2$, and, for $k > 1$

$$w_k := w/k \log^2 k, \tag{6}$$

where w is the normalizer such that $\sum_{k\in\mathbb{N}} w_k = 1$. Replacing ρ with $1/2(\rho+\delta)$ if necessary, where δ is the i.i.d. probability measure with equal probabilities of outcomes, i.e. $\delta(x_{1..n}) = M^{-1/n}$ for all $n \in \mathbb{N}, x_{1..n} \in \mathcal{X}^n$, we shall assume, without loss of generality,

$$-\log \rho(x_{1..n}) \le nM + 1 \text{ for all } n \in \mathbb{N} \text{ and } x_{1..n} \in \mathcal{X}^n. \tag{7}$$

The replacement is without loss of generality as it adds at most 1 to the final bound (to be accounted for). Thus, in particular,

$$L_n(\mu, \rho) \le nM + 1 \text{ for all } \mu. \tag{8}$$

The first part of the proof is the following covering construction.

For each $\mu \in \mathcal{C}$, $n \in \mathbb{N}$ define the sets

$$T^n_\mu := \left\{ x_{1..n} \in \mathcal{X}^n : \frac{\mu(x_{1..n})}{\rho(x_{1..n})} \ge \frac{1}{n} \right\}. \tag{9}$$

From Markov inequality, we obtain

$$\mu(\mathcal{X}^n \backslash T^n_\mu) \le 1/n. \tag{10}$$

For each $k > 1$ let U_k be the partition of $[-\frac{\log n}{n}, M + \frac{1}{n}]$ into k intervals defined as follows. $U_k := \{u^i_k : i = 1..k\}$, where

$$u^i_k = \begin{cases} \left[-\frac{\log n}{n}, \frac{iM}{k}\right] & i = 1, \\ \left(\frac{(i-1)M}{k}, \frac{iM}{k}\right] & 1 < i < k, \\ \left(\frac{(i-1)M}{k}, M + \frac{1}{n}\right] & i = k. \end{cases} \tag{11}$$

Thus, U_k is a partition of $[0, M]$ into k equal intervals but for some padding that we added to the leftmost and the rightmost intervals: on the left we added $[-\frac{\log n}{n}, 0)$ and on the right $(M, M + 1/n]$.

For each $\mu \in \mathcal{C}$, $n, k > 1$, $i = 1..k$ define the sets

$$T^n_{\mu,k,i} := \left\{ x_{1..n} \in \mathcal{X}^n : \frac{1}{n} \log \frac{\mu(x_{1..n})}{\rho(x_{1..n})} \in u^i_k \right\}. \tag{12}$$

Observe that, for every $\mu \in \mathcal{C}$, $k, n > 1$, these sets constitute a partition of T^n_μ into k disjoint sets: indeed, on the left we have $\frac{1}{n} \log \frac{\mu(x_{1..n})}{\rho(x_{1..n})} \ge -\frac{1}{n} \log n$ by definition (9) of T^n_μ, and on the right we have $\frac{1}{n} \log \frac{\mu(x_{1..n})}{\rho(x_{1..n})} \le M + 1/n$ from (7). In particular, from this definition, for all $x_{1..n} \in T^n_{\mu,k,i}$ we have

$$\mu(x_{1..n}) \le 2^{\frac{iM}{k} n + 1} \rho(x_{1..n}). \tag{13}$$

For every $n, k \in \mathbb{N}$ and $i \in \{1..k\}$ consider the following construction. Define

$$m_1 := \max_{\mu\in\mathcal{C}} \rho(T^n_{\mu,k,i})$$

(since $\mathcal{X}^n$ are finite all suprema are reached). Find any μ_1 such that $\rho(T^n_{\mu_1,k,i}) = m_1$ and let $T_1 := T^n_{\mu_1,k,i}$. For $l > 1$, let

$$m_l := \max_{\mu\in\mathcal{C}} \rho(T^n_{\mu,k,i} \backslash T_{l-1}).$$

If $m_l > 0$, let μ_l be any $\mu \in \mathcal{C}$ such that $\rho(T^n_{\mu_l,k,i} \backslash T_{l-1}) = m_l$, and let $T_l := T_{l-1} \cup T^n_{\mu_l,k,i}$; otherwise let $T_l := T_{l-1}$ and $\mu_l := \mu_{l-1}$. Note that, for each $x_{1..n} \in T_l$ there is $l' \le l$ such that $x_{1..n} \in T^n_{\mu_{l'},k,i}$ and thus from (12) we get

$$2^{\frac{(i-1)M}{k} n - \log n} \rho(x_{1..n}) \le \mu_{l'}(x_{1..n}). \tag{14}$$

Finally, define

$$\nu_{n,k,i} := \sum_{l=1}^{\infty} w_l \mu_l. \tag{15}$$

(Notice that for every n, k, i there is only a finite number of positive m_l, since the set $\mathcal{X}^n$ is finite; thus the sum in the last definition is effectively finite.) Finally, define the predictor ν as

$$\nu := \frac{1}{2} \sum_{n,k\in\mathbb{N}} w_n w_k \frac{1}{k} \sum_{i=1}^{k} \nu_{n,k,i} + \frac{1}{2} r, \tag{16}$$

where r is a regularizer defined so as to have for each $\mu' \in \mathcal{C}$ and $n \in \mathbb{N}$

$$\log \frac{\mu'(x_{1..n})}{\nu(x_{1..n})} \le nM - \log w_n + 1 \quad \text{for all } x_{1..n} \in \mathcal{X}^n; \tag{17}$$

this and the stronger statement (7) for ν can be obtained analogously to the latter inequality in the case the i.i.d. measure δ is in $\mathcal{C}$; otherwise (since we need to define ν as a combination of probability measures from $\mathcal{C}$ only), r can be defined the same way as is done in (Ryabko,

2010, Step r of the proof of Theorem 5); for the sake of completeness, this argument is given in the end of this proof.

Next, let us show that the measure ν is the predictor whose existence is claimed in the statement.

Introduce the notation

$$L_n|_A(\mu,\nu) := \sum_{x_{1..n}\in A} \mu(x_{1..n}) \log \frac{\mu(x_{1..n})}{\rho(x_{1..n})};$$

with this notation, for any set $A \subset \mathcal{X}^n$ we have

$$L_n(\mu,\nu) = L_n|_A(\mu,\nu) + L_n|_{\mathcal{X}^n\setminus A}(\mu,\nu).$$

First we want to show that, for each $\mu \in \mathcal{C}$, for each fixed k, i, the sets $T^n_{\mu,k,i}$ are covered by sufficiently few sets T_l, where "sufficiently few" is, in fact, exponentially many with the right exponent. By definition, for each n, i, k the sets $T_l \setminus T_{l-1}$ are disjoint (for different l) and have non-increasing (with l) ρ-probability. Therefore, $\rho(T_{l+1}\setminus T_l) \le 1/l$ for all $l \in \mathbb{N}$. Hence, from the definition of T_l, we must also have $\rho(T^n_{\mu,k,i}\setminus T_l) \le 1/l$ for all $l \in \mathbb{N}$. From the latter inequality and (13) we obtain

$$\mu(T^n_{\mu,k,i}\setminus T_l) \le \frac{1}{l} 2^{\frac{iM}{k}n+1}.$$

Take $l_i := \lceil kn 2^{\frac{iM}{k}n+1}\rceil$ to obtain

$$\mu(T^n_{\mu,k,i}\setminus T_{l_i}) \le \frac{1}{kn}. \tag{18}$$

Moreover, for every $i = 1..k$, for each $x_{1..n} \in T_{l_i}$, there is $l' \le l_i$ such that $x_{1..n} \in T^n_{\mu_{l'},k,i}$ and thus the following chain holds

$$\begin{aligned}
\nu(x_{1..n}) &\ge \frac{1}{2} w_n w_k \frac{1}{k} \nu_{n,k,i} \ge \frac{1}{2} w_n w_k \frac{1}{k} w_{kn\,2^{\frac{iM}{k}n+1}} \mu_{l'}(x_{1..n}) \\
&\ge \frac{w^3}{4n^2k^3 \log^2 n \log^2 k (\log k + \log n + 1 + nMi/k)^2} 2^{-\frac{iM}{k}n} \mu_{l'}(x_{1..n}) \\
&\ge \frac{w^3}{4(M+1)^2 n^4 k^3 \log^2 n \log^2 k} 2^{-\frac{iM}{k}n} \mu_{l'}(x_{1..n}) \\
&\ge \frac{w^3}{4(M+1)^2 n^5 k^3 \log^2 n \log^2 k} 2^{-\frac{M}{k}n} \rho(x_{1..n}) = B_n 2^{-\frac{M}{k}n} \rho(x_{1..n}),
\end{aligned} \tag{19}$$

where the first inequality is from (16), the second from (15) with $l = l_i$, the third is by definition of w_l, the fourth uses $i \le k$ for the exponential term, as well as $(\log n + \log k) \le n - 1$ for $n \ge 3$, which will be justified by the choice of k in the following (27), the fifth inequality uses (14), and the final equality introduces B_n defined as

$$B_n := \frac{w^3}{4(M+1)^2 n^5 k^3 \log^2 n \log^2 k}. \tag{20}$$

We have

$$L_n(\mu,\nu) = \left(\sum_{i=1}^k L_n|_{T_{l_i}}(\mu,\nu)\right) + L_n|_{\mathcal{X}^n\setminus\cup_{i=1}^k T_{l_i}}(\mu,\nu). \tag{21}$$

For the first term, from (19) we obtain

$$\begin{aligned}
\sum_{i=1}^k L_n|_{T_{l_i}}(\mu,\nu) &\le \sum_{i=1}^k L_n|_{T_{l_i}}(\mu,\rho) + Mn/k - \log B_n \\
&= L_n(\mu,\rho) - L_n|_{\mathcal{X}^n\setminus\cup_{i=1}^k T_{l_i}}(\mu,\rho) + Mn/k - \log B_n.
\end{aligned} \tag{22}$$

For the second term in (21), we recall that $T^n_{\mu,k,i}$, $i = 1..k$ is a partition of T^n_μ, and decompose

$$\mathcal{X}^n \setminus \cup_{i=1}^k T_{l_i} \subseteq \left(\cup_{i=1}^k (T^n_{\mu,k,i} \setminus T_{l_i})\right) \cup (\mathcal{X}^n \setminus T^n_\mu). \tag{23}$$

Next, using (17) and an upper-bound for the μ-probability of each of the two sets in (23), namely, (18) and (10), as well as $k \ge 1$, we obtain

$$L_n|_{\mathcal{X}^n\setminus\cup_{i=1}^k T_{l_i}}(\mu,\nu) \le (nM - \log w_n + 1)\frac{2}{n}. \tag{24}$$

Returning to (22), from Jensen's inequality one can show (see, e.g., Ryabko, 2010, equation 11) that, for any set $A \subset \mathcal{X}^n$,

$$-L_n|_A(\mu,\rho) \le \mu(A) \log \rho(A) + 1/2.$$

Therefore, using (8), similarly to (24) we obtain

$$-L_n|_{\mathcal{X}^n\setminus\cup_{i=1}^k T_{l_i}}(\mu,\rho) \le (nM+1)\frac{2}{n} + \frac{1}{2}. \tag{25}$$

Combining (21) with (22), (24) and (25) we derive

$$L_n(\mu,\nu) \le L_n(\mu,\rho) + Mn/k - \log B_n + 4M - \frac{2}{n}(\log w_n - 1) + 1/2; \tag{26}$$

setting

$$k := \lceil n/\log\log n\rceil \tag{27}$$

we obtain the statement of the theorem.

It remains to come back to (17) and define the regularizer r as a combination of measures from $\mathcal{C}$ for this inequality to hold. For each $n \in \mathbb{N}$, denote

$$A_n := \{x_{1..n} \in \mathcal{X}^n : \exists \mu \in \mathcal{C}\ \mu(x_{1..n}) \ne 0\},$$

and let, for each $x_{1..n} \in \mathcal{X}^n$, the probability measure $\mu_{x_{1..n}}$ be any probability measure from $\mathcal{C}$ such that $\mu_{x_{1..n}}(x_{1..n}) \ge \frac{1}{2}\sup_{\mu\in\mathcal{C}} \mu(x_{1..n})$. Define

$$r'_n := \frac{1}{|A_n|} \sum_{x_{1..n}\in A_n} \mu_{x_{1..n}}$$

for each $n \in \mathbb{N}$, and let $r := \sum_{n\in\mathbb{N}} w_n r'_n$. For every $\mu \in \mathcal{C}$ we have

$$r(x_{1..n}) \ge w_n |A_n|^{-1} \mu_{x_{1..n}}(x_{1..n}) \ge \frac{1}{2} w_n |\mathcal{X}|^{-n} \mu(x_{1..n})$$

for every $n \in \mathbb{N}$ and every $x_{1..n} \in A_n$, establishing (17). ■

3.2. Lower Bound

In this section we establish a lower bound on being a Bayesian, complementing the upper bound of Theorem 1. The bound leaves a significant gap with respect to the upper bound, but it shows that the regret of using the Bayesian predictor even with the *best* prior for the given set $\mathcal{C}$ cannot be upper-bounded by a constant.

Theorem 3 *There exists a measurable set of probability measures $\mathcal{C}$ and a probability measure ρ, such that for every Bayesian predictor ν whose prior is concentrated on $\mathcal{C}$, there exists a function $\theta(n)$ which is non-decreasing and goes to infinity with n, there exist infinitely many time steps n_i and measures $\mu_i \in \mathcal{C}$ such that $L_{n_i}(\mu_i, \nu) - L_{n_i}(\mu_i, \rho) \geq \theta(n_i)$ for all $i \in \mathbb{N}$.*

Thus, the lower bound goes to infinity with n but may do so arbitrarily slow. This leaves a gap with respect to the $O(\log n)$ upper bound of Theorem 1. Effectively, the theorem compares the regret of the (best) Bayesian with respect to using the best predictor for $\mathcal{C}$ — but not with using the best predictor for each $\mu \in \mathcal{C}$, which is always μ itself.

Note also that this formulation is good enough to be the opposite of Theorem 1, because the formulation of the latter is strong: Theorem 1 says that *for every μ and for every n* (the regret is upper bounded), so, in order to counter that, it is enough to say that *there exists n and there exists μ* (such that the regret is lower bounded); Theorem 3 is, in fact, a bit stronger, since it establishes that there are infinitely many such n. However, it does not preclude that for every fixed measure μ in $\mathcal{C}$ the loss of the Bayesian is upper-bounded by a constant independent of n (but dependent on μ), while the loss of ρ is linear in n. This is actually the case in the proof.

Proof Let $\mathcal{X} := \{0, 1\}$. Let $\mathcal{C}$ be the set of Dirac delta measures, that is, the probability measures each of which is concentrated on a single deterministic sequence, where the sequences are all the sequences that are 0 from some n on. In particular, introduce $S_n := \{x_{1,2,\ldots} \in \mathcal{X}^\infty : x_i = 0 \text{ for all } i > n\}$, $S := \cup_{n\in\mathbb{N}} S_n$. Let $\mathcal{C}_n$ be the set of all probability measures μ such that $\mu(x) = 1$ for some $x \in S_n$ and let $\mathcal{C} := \cup_{n\in\mathbb{N}} C_n$.

Observe that the set $\mathcal{C}$ is countable. It is, therefore, very easy to construct a (Bayesian) predictor for this set: enumerate it in any way, say $(\mu_k)_{k\in\mathbb{N}}$ spans all of $\mathcal{C}$, fix a sequence of positive weights w_k that sum to 1, and let

$$\nu := \sum_{k\in\mathbb{N}} w_k \mu_k. \tag{28}$$

Then $L_n(\mu_k, \nu) \leq -\log w_k$ for all $k \in \mathbb{N}$. That is, for every $\mu \in \mathcal{C}$ the loss of ν is upper-bounded by a constant: it depends on μ but not on the time index n. So, it is good for every μ for large n, but may be bad for some μ for (relatively) small n, which is what we shall exploit.

Observe that, since $\mathcal{C}$ is countable, every Bayesian ν with its prior over $\mathcal{C}$ must have, by definition, the form (28) for some weights $w_k \in [0, 1]$ and some measures $\mu_k \in \mathcal{C}$. Thus, we fix any Bayesian ν in this form.

Define ρ to be the Bernoulli i.i.d. measure with the parameter 1/2. Note that

$$L_n(\mu, \rho) = n \tag{29}$$

for every n. This is quite a useless predictor; its asymptotic average error is the worst possible, 1. However, it is minimax optimal for every single time step n:

$$\inf_{\rho'} \sup_{\mu\in\mathcal{C}} L_n(\mu, \rho') = n,$$

where the inf is over all possible probability measures. This is why ρ is hard to compete with— and, incidentally, why being minimax optimal for each n separately may be useless.

For each $s \in \mathbb{N}$, let W_s be the weight that ν spends on the measures in the sets $\mathcal{C}_k$ with $k < s$, and let M_s be the set of these measures:

$$W_s := \sum \{w_i : \exists k < s \text{ such that } \mu_i \in \mathcal{C}_k\},$$

and

$$M_s := \{\mu_i : \exists k < s \text{ such that } \mu_i \in \mathcal{C}_k\}.$$

By construction,

$$\lim_{s\to\infty} W_s = 1. \tag{30}$$

Next, for each $n \in \mathbb{N}$, let $U_n := S_{n+1} \setminus S_n$ (these are all the sequences in S_{n+1} with 1 on the nth position). Note that $\mu(U_n) = 0$ for each $\mu \in M_n$, while $|U_n| = 2^n$. From the latter equality, there exists $x_{1..n} \in \mathcal{X}^n$ and $\mu \in U_n \subset S_{n+1}$ such that

$$\mu(x_{1..n} = 1) \text{ but } \nu(x_{1..n}) \leq 2^{-n}(1 - W_s).$$

This, (30) and (29) imply the statement of the theorem. ■

4. Examples

The main object of interest here are sets $\mathcal{C}$ for which $V_\mathcal{C} > 0$, as well as corresponding Bayesian minimax predictors. Various examples of Bayesian predictors for sets $\mathcal{C}$ for which $V_\mathcal{C} = 0$ are analysed by Ryabko (2010), so we do not consider this case here.

4.1. Typical Bernoulli 1/3 Sequences

We start with an example which is somewhat artificial, but comes up as a component in more realistic cases. Take the binary $\mathcal{X}$ and consider all sequences $\mathbf{x} \in \mathcal{X}^\infty$ such that the limiting number of 1s in $\mathbf{x}$ equals 1/3. Denote the set of these sequences S and let the set $\mathcal{C}$ consist of all Dirac measures concentrated on sequences from S. Observe that the Bernoulli i.i.d. measure $\delta_{1/3}$ with probability 1/3 of 1 predicts measures in $\mathcal{C}$ relatively well: $\bar{L}(\mathcal{C}, \delta_{1/3}) = h(1/3)$, where h stands for the binary entropy, and this is also the minimax loss for this set, $V_\mathcal{C}$. It might then appear surprising that this loss is achievable by a combination of countably many measures from $\mathcal{C}$, which consists only of deterministic measures. Let us try to see what such a combination may look like. By definition, for any sequence $\mathbf{x} \in S$ and every ε we can find $n_\varepsilon(\mathbf{x}) \in \mathbb{N}$ such that, for all $n \geq n_\varepsilon(\mathbf{x})$, the average number of 1s in $x_{1..n}$ is within ε of 1/3. Fix the sequence of indices $n_j := 2^j$, $j \in \mathbb{N}$ and the sequence of thresholds $\varepsilon_l := 2^{-l}$. For each j let ${S'}_j^l \subset S$ be the set of all sequences $\mathbf{x} \in S$ such that

$n_{\varepsilon_l}(\mathbf{x}) < n_j$. Select then a finite subset S_j^l of S'^l_j such that for each $\mathbf{x}' \in S'^l_j$ there is $\mathbf{x} \in S$ such that $x'_{1..n_j} = x_{1..n_j}$. This is possible, since the set $\mathcal{X}^{n_j}$ is finite. Now for each $\mathbf{x} \in S_j^l$ take the corresponding measure $\mu_{\mathbf{x}} \in \mathcal{C}$ and attach to it the weight $w_l w_j / |S_j^l|$, where, as before, we are using the weights $w_k = w/k \log^2 k$. Taking these measures for all $j, l \in \mathbb{N}$, we obtain our convex combination. Of course, we did not enumerate all sequences in S (or measures in $\mathcal{C}$) this way; but for each sequence $\mathbf{x} \in S$ and for each n there is a sequence among those that we did enumerate that coincides with $\mathbf{x}$ up to the index n. One can then use the theory of types (Csiszar, 1998) to calculate the sizes of the sets S_j^l and to check that the weights we found give the optimal loss we are after; but for the illustrative purposes of this example this is already not necessary.

4.2. Processes with Abrupt Changes

Start with a family of distributions S, for which we have a good predictor: for example, take S to be the set B of all Bernoulli i.i.d. processes, or, more generally, a set for which $V_S = 0$. The family $\mathcal{C}_\alpha$ parametrized by $\alpha \in (0,1)$ and S is then the family of all processes constructed as follows: there is a sequence of indexes n_i such that $X_{n_i..n_{i+1}}$ is distributed according to μ_i for some $\mu_i \in S$. Take then all possible sequences μ_i and all sequences n_i whose limiting frequency $\limsup_{i\to\infty} \frac{1}{n}\{i : n_i < n\}$ is bounded by α, to obtain our set $\mathcal{C}_{S,\alpha}$. Thus, we have a family of processes with abrupt changes in distribution, where between changes the distribution is from S, the changes are assumed to have the frequency bounded by α but are otherwise arbitrary. This example was considered by Willems (1996) for the case $S = B$, with the goal of minimizing the regret with respect to the predictor that knows where the changes occur (the value $V_{\mathcal{C}}$ was not considered directly). The method proposed in the latter work, in fact, is not limited to the case $S = B$, but is general. The algorithm is based on a prior over all possible sequences n_i of changes; between the changes the optimal predictor for B is used, which is also a Bayesian predictor with a specific prior. The regret obtained is of order $\log n$. Since for Bernoulli processes themselves the best achievable average loss up to time n is $\frac{1}{n}(\frac{1}{2}\log n + 1)$, for the sequence $1..n_t$ it is $\frac{1}{n_t}\sum_{i=1}^{t}(\frac{1}{2}\log(n_i - n_{i-1}) + 1)$, where $n_0 := 1$. By Jensen's inequality, this sum is maximized when all the segments $n_i - n_{i-1}$ are of the same length, $1/\alpha$, so the total average loss is upper-bounded by $\alpha(1 - \frac{1}{2}\log\alpha)$. This value is also attainable, and thus gives $V_{\mathcal{C}_{B,\alpha}}$. A similar result can be obtained if we replace Bernoulli processes with Markov processes, but not with an arbitrary S for which $V_S = 0$. For example, if we take S to be all finite-memory distributions, then the resulting process may be completely unpredictable ($V_{\mathcal{C}} = 1$): indeed, if the memory of distributions μ_i grows (with i) faster than αn, then there is little one can do. For such sets S one can make the problem amenable by restricting the way the distributions μ_i are selected, for example, imposing an ergodicity-like condition that the average distribution has a limit. Another way (often considered in the literature in slightly different settings, see Gyorgy et al., 2012 and references) is to have $\alpha \to 0$, although in this case one recovers $V_{\mathcal{C}_S} = 0$ provided α goes to 0 slowly enough (and, of course, provided $V_S = 0$).

4.3. Predictable Aspects

The preceding example can be thought of as an instantiation of the general class of processes in which some aspects are predictable while others are not. Thus, in the considered example changes between the distributions were unpredictable, but between the changes the distributions were predictable. Another example of this kind is that of processes predictable on some scales but not on others. Imagine that it is possible to predict, for example, large fluctuations of the process but not small fluctuations (or the other way around). More formally, consider now an alphabet $\mathcal{X}$ with $|\mathcal{X}| > 2$, and let Y be a partition of $\mathcal{X}$. For any sequence $x_1, \dots, x_n, \dots$ there is an associated sequence $y_1, \dots, y_n, \dots$ where y_i is defined as $y \in Y$ such that $x_i \in y$. Here again we can obtain examples of sets $\mathcal{C}$ of processes with $V_{\mathcal{C}} \in (0,1)$ by restricting the distribution of $y_1, \dots, y_n, \dots$ to a set B with $V_B = 0$. The interpretation is that, again, we can model the y part (by processes in B) but not the rest, which we then allow to be arbitrary.

Yet another example is that of processes predictable only after certain kind of events: such as a price drop; or a rain. At other times, the process is unpredictable: it can, again, be an arbitrary sequence. More formally, let a set $A \subset \mathcal{X}^* := \cup_{k\in\mathbb{N}}\mathcal{X}^k$ be measurable. Consider for each sequence $\mathbf{x} = x_1, \dots, x_n, \dots$ another (possibly finite) sequence $\mathbf{x}' = x'_1, \dots, x'_n, \dots$ given by $x'_i := (x_{n_i+1})_{i\in\mathbb{N}}$ where n_i are all indexes such that $x_{1..n_i} \in A$. We now form the set $\mathcal{C}$ as the set of all processes μ such that $\mathbf{x}'$ belongs (μ-a.s.) to some pre-defined set B; for this set B we may have $V_B = 0$. This means that we can model what happens after events in A — by processes in B, but not the rest of the times, on which we say the process may be arbitrary. For different A and B we then obtain examples where $V_{\mathcal{C}} \in (0,1)$. In relation to this it is worth mentioning the work (Lattimore et al., 2011) which explores the possibility that a Bayesian predictor may fail to predict some subsequences.

5. Decision-Theoretic Interpretations of the Asymptotic Result

Classical decision theory is concerned with single-step games. Among its key results are the complete class and minimax theorems. The asymptotic formulation of the infinite-horizon problem considered here presents both differences and similarities which we attempt to summarize in this section. A distinction worth mentioning at this point is that the results presented here are obtained under no assumptions whatsoever, whereas the results in decision theory we refer to always have a number of conditions; on the other hand, here we are concerned with just one specific loss function (KL divergence) rather than general losses that are common in decision theory. The terminology in this section is mainly after Ferguson (1967).

Predictors $\rho \in \mathcal{P}$ are called *strategies of the statistician*. The probability measures $\mu \in \mathcal{C}$ are now the basic *strategies of the opponent* (a.k.a. Nature), and the first thing we need to do is to extend these to randomized strategies. To this end, denote $\mathcal{C}^*$ the set of all probability distributions over measurable subsets of $\mathcal{C}$. Thus, the opponent selects a randomized strategy $W \in \mathcal{C}^*$ and the statistician (predictor) ρ suffers the loss

$$E_{W(\mu)}\bar{L}(\mu, \rho), \tag{31}$$

where the notation $W(\mu)$ means that μ is drawn according to W. Note a distinction with the combinations we considered before. A combination of the kind $\nu = \int_{\mathcal{C}} \alpha dW(\alpha)$ is itself

a probability measure over the one-way infinite sequences, whereas a probability measure $W \in \mathcal{C}^*$ is a probability measure over $\mathcal{C}$.

5.1. Minimax

Generalizing the definition (3) of $V_{\mathcal{C}}$, we can now introduce the *upper value*

$$\bar{V}_{\mathcal{C}} := \inf_{\rho \in \mathcal{P}} \sup_{W \in \mathcal{C}^*} E_{W(\mu)} \bar{L}(\mu, \rho). \tag{32}$$

Furthermore, the *maximin* (the *lower value*) is defined as

$$\underline{V}_{\mathcal{C}} := \sup_{W \in \mathcal{C}^*} \inf_{\rho \in \mathcal{P}} E_{W(\mu)} \bar{L}(\mu, \rho). \tag{33}$$

The so-called minimax theorems in decision theory (e.g., Ferguson, 1967) for single-step games and general loss functions state that, under certain conditions, $\bar{V}_{\mathcal{C}} = \underline{V}_{\mathcal{C}}$ and the statistician has a minimax strategy, that is, there exists ρ on which $\bar{V}_{\mathcal{C}}$ is attained. Minimax theorems generalize the classical result of von Neumann (1928), and provide sufficient conditions of various generality for it to hold. A rather general sufficient condition is the existence of a topology with respect to which the set of all strategies of the statistician, $\mathcal{P}$ in our case, is compact, and the risk, which is $\bar{L}(\mu, \rho)$ in our case, is lower semicontinuous. Such a condition seems nontrivial to verify. For example, a (meaningful) topology with respect to which $\mathcal{P}$ is compact is that of the so-called distributional distance (Gray, 1988) (in our case it coincides with the topology of the weak* convergence), but $\bar{L}(\mu, \rho)$ is not (lower) semicontinuous with respect to it. Some other (including non-topological) sufficient conditions are given by Sion (1958); LeCam (1955). Other related results for KL divergence (expected log loss) include (Ryabko, 1979; Gallager, 1976-1979; Haussler, 1997).

In our setup, it is easy to see that, for every $\mathcal{C}$,

$$\bar{V}_{\mathcal{C}} = V_{\mathcal{C}}$$

and so Corollary 2 holds for $\bar{V}_{\mathcal{C}}$. Thus, using decision-theoretic terminology, we can state the following.

Corollary 4 (partial minimax theorem) *For every set $\mathcal{C}$ of strategies of the opponent, the statistician has a minimax strategy.*

However, the question of whether the upper and the lower values coincide remains open. That is, we are taking the worst possible distribution over $\mathcal{C}$, and ask what is the best possible predictor with the knowledge of this distribution ahead of time. The question is whether $\underline{V}_{\mathcal{C}} = V_{\mathcal{C}}$. A closely related question is whether there is a worst possible strategy for the opponent. This latter would be somehow a maximally spread-out (or maximal entropy) distribution over $\mathcal{C}$. In general, measurability issues seem to be very relevant here, especially for the maximal-entropy distribution part.

5.2. Complete Class

In this section we shall see that Corollary 2 can be interpreted as a complete-class theorem for asymptotic average loss, as well as some principled differences between the cases $V_{\mathcal{C}} > 0$ and $V_{\mathcal{C}} = 0$.

For a set of probability measures (strategies of the opponent) $\mathcal{C}$, a predictor ρ_1 is said to be *as good as* a predictor ρ_2 if $\bar{L}(\mu, \rho_1) \le \bar{L}(\mu, \rho_2)$ for all $\mu \in \mathcal{C}$. A predictor ρ_1 is *better (dominates)* ρ_2 if ρ_1 is as good as ρ_2 and $\bar{L}(\mu, \rho_1) < \bar{L}(\mu, \rho_2)$ for some $\mu \in \mathcal{C}$. A predictor ρ is *admissible* (also called *Pareto optimal*) if there is no predictor ρ' which is better than ρ; otherwise it is called *inadmissible*. Similarly, a set of predictors D is called a *complete class* if for every $\rho' \notin D$ there is $\rho \in D$ such that ρ is better than ρ'. A set of of predictors D is called an *essentially complete class* if for every $\rho' \notin D$ there is $\rho \in D$ such that ρ is as good as ρ'. An (essentially) complete class is called *minimal* if none of its proper subsets is (essentially) complete.

Furthermore, in decision-theoretic terminology, a predictor ρ is called a *Bayes rule* for a prior $W \in \mathcal{C}^*$ if it is optimal for W, that is, if it attains $\inf_{\rho \in \mathcal{P}} E_{W(\mu)} \bar{L}(\mu, \rho)$. Clearly, if W is concentrated on a finite or countable set then any mixture over this set with full support is a Bayes rule, and the value of the inf above is 0; so the use of this terminology is non-contradictory here.

In decision theory, the complete class theorem (Wald, 1950; LeCam, 1955), see also (Ferguson, 1967) states that, under certain conditions similar to those above for the minimax theorem, the set of Bayes rules is complete and the admissible Bayes rules form a minimal complete class.

An important difference in our set-up is that all strategies are inadmissible (unless $V_{\mathcal{C}} = 0$), and one cannot speak about minimal (essentially) complete classes. However, the set of all Bayes rules is still essentially complete, and an even stronger statement holds: it is enough to consider all Bayes rules with countable priors:

Corollary 5 (Complete class theorem) *For every set $\mathcal{C}$, the set of those Bayes rules whose priors are concentrated on at most countable sets is essentially complete. There is no admissible rule (predictor) and no minimal essentially complete class unless $V_{\mathcal{C}} = 0$. In the latter case, every predictor ρ that attains this value is admissible and the set $\{\rho\}$ is minimal essentially complete.*

Proof The first statement is a reformulation of Corollary 2. To prove the second statement, consider any $\mathcal{C}$ such that $V_{\mathcal{C}} > 0$, take a predictor ρ that attains this value (such a predictor exists by Theorem 1), and a probability measure μ such that $\bar{L}(\mu, \rho) > 0$. Then for a predictor $\rho' := 1/2(\rho + \mu)$ we have $\bar{L}(\mu, \rho') = 0$. Thus, ρ' is better than ρ: its loss is strictly smaller on one measure, μ, and is at least the same on all the rest of the measures in $\mathcal{C}$. Therefore, ρ is inadmissible. The statement about minimal essentially complete class is proven analogously: indeed, take any essentially complete class, D, and any predictor $\rho \in D$. Take then the predictor ρ' constructed as above. Since ρ' is better than ρ and D is essentially complete, there must be another predictor $\rho'' \in D$, such that ρ'' is as good as ρ'. Therefore, $D \setminus \{\rho\}$ is essentially complete and D is not minimal. The statements about the case $V_{\mathcal{C}} = 0$ are self-evident. ■

6. Non-Realizable Case: Regret Minimization, Suboptimality of Bayes

In the non-realizable case the situation is principally different: it may happen that every combination of distributions in the model is suboptimal — even asymptotically.

Theorem 6 *There exist a set $\mathcal{C}$ for which $U_{\mathcal{C}} = 0$ and this value is attainable, yet for some constant $c > 0$ every Bayesian predictor φ with a prior concentrated on $\mathcal{C}$ must have*

Intuitively, the reason why any Bayesian predictor does not work in the counterexample of the proof given below is as follows. The set $\mathcal{C}$ considered is so large that any Bayesian predictor has to attach an exponentially decreasing a-posteriori weight to each element in $\mathcal{C}$. At the same time, by construction, each measure in $\mathcal{C}$ already attaches too little weight to the part of the event space on which it is a good predictor. In other words, the likelihood of the observations with respect to each predictor in $\mathcal{C}$ is too small to allow for any added penalty. To combine predictors in $\mathcal{C}$ one has to *boost* the likelihood, rather than attach a penalty. While this result is stated for Bayesian predictors, from the argument above it is clear that the example used in the proof is applicable to any combination of predictors in $\mathcal{C}$ one might think of, including, for example, MDL (Rissanen, 1989) and expert-advice-style predictors (e.g., Cesa-Bianchi and Lugosi, 2006). Indeed, if one has to boost the likelihood for some classes of predictors, it clearly breaks the predictor for other classes. In other words, there is no way to combine the prediction of the experts, short of disregarding them and doing something else instead.

Remark 7 (Countable $\mathcal{C}$) *Note that any set $\mathcal{C}$ satisfying the theorem must necessarily be uncountable. Indeed, for any countable set $\mathcal{C} = (\mu_k)_{k\in\mathbb{N}}$, take the Bayesian predictor $\varphi := \sum_{k\in\mathbb{N}} w_k\mu_k$, where w_k can be, for example, $\frac{1}{k(k+1)}$. Then, for any ν and any n, from (2) we obtain*

$$L_n(\nu,\varphi) \le -\log w_k + L_n(\nu,\mu_k).$$

That is to say, the regret of φ with respect to any μ_k is a constant independent of n (though it does depend on k), and thus for every ν we have $\bar{R}^\nu(C,\varphi) = 0$. It is worth noting that the origins of the use of such countable mixtures for prediction trace back to Zvonkin and Levin (1970); Solomonoff (1978).

Proof [of Theorem 6.] Let the alphabet $\mathcal{X}$ be ternary $\mathcal{X} = \{0,1,2\}$. For $\alpha \in (0,1)$ denote $h(\alpha)$ the binary entropy $h(\alpha) := -\alpha\log\alpha - (1-\alpha)\log(1-\alpha)$. Fix an arbitrary $p \in (0,1/2)$ and let β_p be the Bernoulli i.i.d. measure (produces only 0s and 1s) with parameter p. Let S be the set of sequences in $\mathcal{X}^\infty$ that have no 2s and such that the frequency of 1 is close to p:

$$S := \left\{\mathbf{x} \in \mathcal{X}^\infty : x_i \ne 2\forall i, \text{ and } \left|\frac{1}{t}|\{i = 1..t : x_i = 1\}| - p\right| \le f(t) \text{ from some } t \text{ on}\right\},$$

where $f(t) = \log t/\sqrt{t}$. Clearly, $\beta_p(S) = 1$.

Define the set D_S as the set of all Dirac probability measures concentrated on a sequence from S, that is $D_S := \{\nu_{\mathbf{x}} : \nu_{\mathbf{x}}(\mathbf{x}) = 1,\ \mathbf{x} \in S\}$. Moreover, for each $\mathbf{x} \in S$ define the probability measure $\mu_{\mathbf{x}}$ as follows: $\mu_{\mathbf{x}}(X_{n+1}|X_{1..n}) = p$ coincides with β_p (that is, 1 w.p. p and 0 w.p. $1-p$) if $X_{1..n} = x_{1..n}$, and outputs 2 w.p. 1 otherwise: $\mu_{\mathbf{x}}(2|X_{1..n}) = 1$ if $X_{1..n} \ne x_{1..n}$. That is, $\mu_{\mathbf{x}}$ behaves as β_p only on the sequence $\mathbf{x}$, and on all other sequences it just outputs 2 deterministically. This means, in particular, that many sequences have probability 0, and some probabilities above are defined conditionally on zero-probability events, but this is not a problem; see the remark in the end of the proof.

Finally, let $\mathcal{C} := \{\mu_{\mathbf{x}} : \mathbf{x} \in S\}$. Next we define the predictor ρ that predicts well all measures in $\mathcal{C}$. First, introduce the probability measure δ that is going to take care of all the measures that output 2 w.p.1 from some time on. For each $a \in \mathcal{X}^*$ let δ_a be the probability measure that is concentrated on the sequence that starts with a and then consists of all 2s. Define $\delta := \sum_{a\in\mathcal{X}^*} w_a\delta_a$, where w_a are arbitrary positive numbers that sum to 1. Let also the probability measure β' be i.i.d. uniform over $\mathcal{X}$. Finally, define

$$\rho := 1/3(\beta_p + \beta' + \delta). \tag{34}$$

Next, let us show that, for every ν, the measure ρ predicts ν as well as any measure in $\mathcal{C}$: its loss is an additive constant factor. In fact, it is enough to see this for all $\nu \in D_S$, and for all measures that output all 2s w.p.1 from some n on. For each ν in the latter set, from (34) the loss of ρ is upper-bounded by $\log 3 - \log w_a$, where w_a is the corresponding weight. This is a constant (does not depend on n). For the former set, again from the definition (34) for every $\nu_{\mathbf{x}} \in D_S$ we have (see also Remark 7)

$$L_n(\nu_{\mathbf{x}},\rho) \le \log 3 + L_n(\nu_{\mathbf{x}},\beta_p) = nh(p) + o(n),$$

while

$$\inf_{\mu\in C} L_n(\nu_{\mathbf{x}},\mu) = L_n(\nu_{\mathbf{x}},\mu_{\mathbf{x}}) = nh(p) + o(n).$$

Therefore, for all ν we have

$$R_n^\nu(C,\rho) = o(n) \text{ and } \bar{R}^\nu(C,\rho) = 0.$$

Thus, we have shown that for every $\nu \in S$ there is a reasonably good predictor in $\mathcal{C}$ (here "reasonably good" means that its loss is linearly far from that of random guessing), and, moreover, there is a predictor ρ whose asymptotic regret is zero with respect to $\mathcal{C}$.

Next we need to show that any Bayes predictor has $2nh(p) + o(n)$ loss on at least some measure, which is double that of ρ, and which can be as bad as random guessing (or worse; depending on p). We show something stronger: any Bayes predictor has asymptotic average loss of $2nh(p)$ *on average* over all measures in S. So there will be many measures on which it is bad, not just one.

Let φ be any Bayesian predictor with its prior concentrated on $\mathcal{C}$. Since $\mathcal{C}$ is parametrized by S, for any $x_{1..n} \in \mathcal{X}^n, n \in \mathbb{N}$ we can write $\varphi(x_{1..n}) = \int_S \mu_{\mathbf{y}}(x_{1..n})dW(\mathbf{y})$ where W is some probability measure over S (the prior). Moreover, using the notation $W(x_{1..k})$ for the W-measure of all sequences in S that start with $x_{1..k}$, from the definition of the measures $\mu_{\mathbf{x}}$, for every $\mathbf{x} \in S$ we have

$$\int_S \mu_{\mathbf{y}}(x_{1..n})dW(\mathbf{y}) = \int_{\mathbf{y}\in S: y_{1..n}=x_{1..n}} \beta_p(x_{1..n})dW(\mathbf{y}) = \beta_p(x_{1..n})W(x_{1..n}). \tag{35}$$

Consider the average

$$E_U \limsup \frac{1}{n} L_n(\nu_x,\varphi)dU(\mathbf{x}),$$

where the expectation is taken with respect to the probability measure U defined as the measure β_p restricted to S; in other words, U is approximately uniform over this set. Fix

any $\nu_{\mathbf{x}} \in S$. Observe that $L_n(\nu_{\mathbf{x}}, \varphi) = -\log \varphi(x_{1..n})$. For the asymptotic regret, we can assume w.l.o.g. that the loss $L_n(\nu_{\mathbf{x}}, \varphi)$ is upper-bounded, say, by $n \log |\mathcal{X}|$ at least from some n on (for otherwise the statement already holds for φ). This allows us to use Fatou's lemma to bound

$$E_U \limsup \frac{1}{n} L_n(\nu_{\mathbf{x}}, \varphi) \geq \limsup \frac{1}{n} E_U L_n(\nu_{\mathbf{x}}, \varphi) = \limsup -\frac{1}{n} E_U \log \varphi(\mathbf{x})$$
$$= \limsup -\frac{1}{n} E_U \log \beta_p(x_{1..n}) W(x_{1..n}), \quad (36)$$

where in the last equality we used (35). Moreover,

$$- E_U \log \beta_p(x_{1..n}) W(x_{1..n})$$
$$= -E_U \log \beta_p(x_{1..n}) + E_U \log \frac{U(x_{1..n})}{W(x_{1..n})} - E_U \log U(x_{1..n}) \geq 2h(p)n + o(n), \quad (37)$$

where in the inequality we have used the fact that KL divergence is non-negative and the definition of U (that is, that $U = \beta_p|_S$). From this and (36) we obtain the statement of the theorem.

Finally, we remark that all the considered probability measures can be made non-zero everywhere by simply combining them with the uniform i.i.d. over $\mathcal{X}$ measure β', that is, taking for each measure ν the combination $\frac{1}{2}(\nu + \beta')$. This way all losses up to time n become bounded by $n \log |\mathcal{X}| + 1$, but the result still holds with a different constant. ■

7. Conclusion and Future Work

A statistician facing an unknown stochastic phenomenon has a large, nonparametric model class at hand that she has reasons to believe captures some aspects of the problem. Yet other aspects remain completely enigmatic, and there is little hope that the process generating the data indeed comes from the model class. For this reason, the statistician is content with having non-zero error no matter how much data may become available now or in the future, but she would still like to make some use of the model. There are now two rather distinct ways to proceed. One is to say that the data may come from an arbitrary sequence, and to try to construct a predictor that minimizes the regret with respect to every distribution in the model class, on every sequence. Thus, one would be treating the model class as a set of experts. The other way is to try to enlarge the model class, in particular, by allowing that all there is unknown in the process may be arbitrary (that is, an arbitrary sequence). Having done this, one may safely assume that the probability measure that generates the data belongs to the model class. This second way may be more difficult precisely on the modelling step. Yet, the conclusion of this work is that this is the way to follow, for in this case one can be sure that it is possible to make statistical inference by standard available tools, specifically, Bayesian forecasting. Indeed, even if the best achievable asymptotic error is non-zero, it is attained by a Bayesian forecaster with some prior. Finding such a prior is a separate problem, but it is a one with which Bayesians are familiar. Here, modelling that unknown part should not create much trouble: a good distribution over all

sequences is just the Bernoulli i.i.d. measure with equiprobable outcomes. (Note that it is not necessary to look for priors concentrated on countable sets.) On the other hand, for the regret-minimization route, the statistician cannot use an arbitrary model class; indeed, she would first need to make sure that regret minimization is viable at all for the model class at hand: it may happen that every combination of distributions in the model is suboptimal.

There are no criteria for checking this, only some (rather small) examples, such as finite or countable sets, or specific parametric families.

Finding such criteria for the viability of regret minimization is an interesting open problem. To make it more precise, the question is for which sets $\mathcal{C}$ of distributions the minimax regret (is attainable and) can be attained by a combination (either Bayesian or some other) of distributions in $\mathcal{C}$.

It is worth noting that the conclusions of the paper are not about Bayesian versus non-Bayesian inference. Rather, Bayesian inference is used as a generic approach to construct predictors for general (uncountable) model classes. At this level of generality, the choice of alternative approaches is very limited, but it would be interesting to see which predictors constructed for more specific settings can be generalized to arbitrary model classes, and whether the corresponding result holds for them.

Another interesting open question concerns different losses. While the proof does not seem to be hinged very specifically on the log loss, it does use some properties of it in an important way. In particular, the property that if μ predicts ν then also any convex combination $\alpha\mu + (1-\alpha)\rho$ predicts ν for any ρ. This does not hold for some other losses, in particular already for KL loss without Cesaro averaging; see (Ryabko and Hutter, 2008) for a discussion and some results on this property.

Some other interesting open questions are the decision-theoretic ones mentioned in Section 5; specifically, those concerning the minimax theorem and the existence of maximally spread distributions over $\mathcal{C}$. It would also be interesting to calculate the value $V_{\mathcal{C}}$ for different classes of distributions, similar to what is done for the i.i.d. example in Section 4.

Finally, several questions remain concerning the bounds presented in Section 3. The first question is how sharp is the upper bound in Theorem 1. So far, the lower bound (Theorem 3) only shows that, for every prior, the Bayesian may suffer more than constant regret. The question whether the $\log n$ term is necessary remains open. If it is necessary, then the constant in front of the log becomes important, in particular because the optimal loss is of order $\log n$ in some commonly studied special cases of $\mathcal{C}$, such as i.i.d. or Markov measures. (It is worth mentioning that the known optimal predictors in these cases (Krichevsky, 1993) are, in fact, Bayesian.) Moreover, it may be worth trying to improve the bounds specifically for the case $L_n(\mu, \rho) = O(\log n)$, since in the opposite case it is not important.

References

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

I. Csiszar. The method of types. *IEEE Transactions on Information Theory*, 44(6):2505–2523, 1998.

P. Diaconis and D. Freedman. On the consistency of Bayes estimates. *Annals of Statistics*, 14(1):1–26, 1986.

Th. S. Ferguson. *Mathematical statistics: A decision theoretic approach.* Academic press, 1967.

D. A.Freedman. On the asymptotic behavior of Bayes estimates in the discrete case I. *The Annals of Mathematical Statistics*, pages 1386–1403, 1963.

D. A. Freedman. On the asymptotic behavior of Bayes estimates in the discrete case II. *The Annals of Mathematical Statistics*, 36(2):454–456, 1965.

R. G. Gallager. Source coding with side information and universal coding. Technical Report LIDS-P-937, M.I.T., 1976-1979.

R. M. Gray. *Probability, Random Processes, and Ergodic Properties.* Springer Verlag, 1988.

L. Gyorfi and G. Ottucsak. Sequential prediction of unbounded stationary time series. *Information Theory, IEEE Transactions on*, 53(5):1866 –1872, May 2007. doi: 10.1109/TIT.2007.894660.

A. Gyorgy, T. Linder, and G. Lugosi. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, 2012.

D. Haussler. A general minimax result for relative entropy. *IEEE Trans. on Information Theory*, 43(4):1276–1280, 1997.

E. Kalai and E. Lehrer. Weak and strong merging of opinions. *Journal of Mathematical Economics*, 23:73–86, 1994.

R. Krichevsky. *Universal Compression and Retrival.* Kluwer Academic Publishers, 1993.

T. Lattimore, M. Hutter, and V. Gavane. Universal prediction of selected bits. In *Algorithmic Learning Theory*, pages 262–276. Springer, 2011.

L. LeCam. An extension of Wald's theory of statistical decision functions. *The Annals of Mathematical Statistics*, 26(1):69–81, 1955.

G. Morvai, S. J. Yakowitz, and P. Algoet. Weakly convergent nonparametric forecasting of stationary time series. *Information Theory, IEEE Transactions on*, 43(2):483 –498, March 1997. doi: 10.1109/18.556107.

Y. Noguchi. Merging with a set of probability measures: A characterization. *Theoretical Economics*, 10(2):411–444, 2015.

J. J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publ. Co., 1989.

B. Ryabko. Coding of a source with unknown but ordered probabilities. *Problems of Information Transmission*, 15(2):134–138, 1979.

B. Ryabko. Prediction of random sequences and universal coding. *Problems of Information Transmission*, 24:87–96, 1988.

D. Ryabko. On finding predictors for arbitrary families of processes. *Journal of Machine Learning Research*, 11:581–602, 2010.

D. Ryabko. On the relation between realizable and non-realizable cases of the sequence prediction problem. *Journal of Machine Learning Research*, 12:2161–2180, 2011.

D. Ryabko. Things Bayes can't do. In *Proceedings of the 27th International Conference on Algorithmic Learning Theory (ALT'16)*, volume 9925 of *LNCS*, pages 253–260, Bari, Italy, 2016. Springer.

D. Ryabko. Universality of Bayesian mixture predictors. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory (ALT'17)*, volume 76 of *PMLR*, pages 57–71, Kyoto, Japan, 2017. JMLR.

D. Ryabko and M. Hutter. Predicting non-stationary processes. *Applied Mathematics Letters*, 21(5):477–482, 2008.

M. Sion. On general minimax theorems. *Pacific J. Math*, 8(1):171–176, 1958.

R. J. Solomonoff. Complexity-based induction systems: comparisons and convergence theorems. *IEEE Trans. Information Theory*, IT-24:422–432, 1978.

J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

A. Wald. *Statistical decision functions.* John Wiley&Sons, New York, 1950.

F. M. J. Willems. Coding for a binary independent piecewise-identically-distributed source. *IEEE Transactions on Information Theory*, 42(6):2210–2217, 1996.

A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83–124, 1970.

Journal of Machine Learning Research 20 (2019) 1-75 Submitted 9/18; Revised 4/19; Published 6/19

Learning Optimized Risk Scores

Berk Ustun BERK@SEAS.HARVARD.EDU
Center for Research in Computation and Society
Harvard University

Cynthia Rudin CYNTHIA@CS.DUKE.EDU
Department of Computer Science
Department of Electrical and Computer Engineering
Department of Statistical Science
Duke University

Editor: Russ Greiner

Abstract

Risk scores are simple classification models that let users make quick risk predictions by adding and subtracting a few small numbers. These models are widely used in medicine and criminal justice, but are difficult to learn from data because they need to be calibrated, sparse, use small integer coefficients, and obey application-specific constraints. In this paper, we introduce a machine learning method to learn risk scores. We formulate the risk score problem as a mixed integer nonlinear program, and present a cutting plane algorithm to recover its optimal solution. We improve our algorithm with specialized techniques that generate feasible solutions, narrow the optimality gap, and reduce data-related computation. Our algorithm can train risk scores in a way that scales linearly in the number of samples in a dataset, and that allows practitioners to address application-specific constraints without parameter tuning or post-processing. We benchmark the performance of different methods to learn risk scores on publicly available datasets, comparing risk scores produced by our method to risk scores built using methods that are used in practice. We also discuss the practical benefits of our method through a real-world application where we build a customized risk score for ICU seizure prediction in collaboration with the Massachusetts General Hospital.

Keywords: scoring systems; classification; calibration; customization; interpretability; cutting plane methods; discrete optimization; mixed integer nonlinear programming.

1. Introduction

Risk scores are linear classification models that let users assess risk by adding, subtracting, and multiplying a few small numbers (see Figure 1). These models are widely used to support human decision-making in domains such as:

- *Medicine*: to assess the risk of mortality in intensive care (e.g., Moreno et al., 2005), critical physical conditions (e.g., adverse cardiac events, Six et al., 2008; Than et al., 2014) and mental illnesses (e.g., ADHD in Kessler et al., 2005; Ustun et al., 2017).
- *Criminal Justice*: to assess the risk of recidivism when setting bail, sentencing, and release on parole (see e.g., Latessa et al., 2009; Austin et al., 2010; Pennsylvania Bulletin, 2017).

©2019 Berk Ustun and Cynthia Rudin.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-615.html.

- *Finance*: to assess the risk of default on a loan (see e.g., credit scores in FICO, 2011; Siddiqi, 2017), and to guide investment decisions (Piotroski, 2000; Beneish et al., 2013).

The adoption of risk scores in these domains stems from the fact that decision-makers often find them easy to use and understand. In comparison to other kinds of models, risk scores let users make quick predictions by simple arithmetic, without a computer or calculator. Users can gauge the effect of changing multiple input variables on a prediction, and override predictions in an informed manner when needed. In comparison to scoring systems for decision-making, which predict a yes-or-no outcome at a single operating point (see e.g., the models considered in Ustun and Rudin, 2016; Zeng et al., 2017; Carrizosa et al., 2016; Van Belle et al., 2013; Billiet et al., 2018, 2017; Sokolovska et al., 2017, 2018), risk scores output risk estimates at multiple operating points. Thus, users can choose an operating point while the model is deployed. Further, they are given risk estimates that – when calibrated – can inform this choice and support decisions in other ways (see e.g., Shah et al., 2018). We provide more background on risk scores in Appendix C.

1.	**C**ongestive Heart Failure	1 point		...
2.	**H**ypertension	1 point	+	...
3.	**A**ge $\geq$ 75	1 point	+	...
4.	**D**iabetes Mellitus	1 point	+	...
5.	Prior **S**troke or Transient Ischemic Attack	**2** points	+	
		SCORE	=	

SCORE	0	1	2	3	4	5	6
RISK	1.9%	2.8%	4.0%	5.9%	8.5%	12.5%	18.2%

Figure 1: $CHADS_2$ risk score of Gage et al. (2001) to assess stroke risk (see www.mdcalc.com for other medical scoring systems). The variables and points of this model were determined by a panel of experts, and the risk estimates were computed empirically from data.

Although risk scores have existed for nearly a century (see e.g., Burgess, 1928), many of these models are still built *ad hoc*. This is partly because risk scores are often developed for applications where models must satisfy constraints on interpretability and usability (see e.g., requirements on "face validity" and "user friendliness" in Than et al., 2014). Handling such constraints necessitates precise control over multiple elements of a model, from its choice of features, to their relationship with the outcome (see e.g., Gupta et al., 2016), to the performance of the model on specific subgroups (Feldman et al., 2015; Pleiss et al., 2017). Since existing classification methods do not provide control over all these elements, risk scores are typically built using heuristics and expert judgment (e.g., preliminary feature selection, followed by logistic regression with the chosen features, scaling, and rounding as outlined by Antman et al., 2000). In some cases, risk scores are hand-crafted by a panel of experts (see e.g., the $CHADS_2$ score in Figure 1, or the National Early Warning Score of McGinley and Pearse, 2012). As we will show, such approaches may produce a model that violates important requirements, or that performs poorly relative to the best risk score that can be built using the same dataset. In such cases, the lack of a formal guarantee complicates model development: when a risk score performs poorly, one cannot tell if this is due to the use of heuristics, or due to overly restrictive constraints.

In this paper, we present a new machine learning method to learn risk scores from data. Our method learns risk scores by solving a mixed-integer nonlinear program (MINLP), which minimizes the logistic loss for calibration and AUC, penalizes the ℓ_0-norm for sparsity, and restricts coefficients to small integers. We refer to this optimization problem as the *risk score problem*, and refer to the risk score built from its solution as a *Risk-calibrated Supersparse Linear Integer Model* (RiskSLIM). We aim to recover a *certifiably optimal solution* – i.e., a global optimum along with a certificate of optimality. This requires solving a hard optimization problem, but has three major benefits for our setting:

(i) *Performance*: Since the MINLP directly penalizes and constrains discrete quantities, it can produce a risk score that is fully optimized for feature selection and small integer coefficients, and that obeys application-specific requirements. Thus, models will not suffer in training performance due to the use of heuristics or post-processing.

(ii) *Direct Customization*: Practitioners can address application-specific requirements by adding discrete constraints to the MINLP formulation, which can be solved with a generic solver (that is called by our algorithm as a subroutine). In this way, they can customize risk scores without parameter tuning, post-processing, or implementing a new algorithm.

(iii) *Evaluating the Impact of Constraints*: Our method pairs risk scores with a certificate of optimality. By definition, a certifiably optimal solution to the risk score problem attains the best performance among risk scores that satisfy a particular set of constraints. Once we recover a certifiably optimal solution, we therefore end up with a risk score with acceptable performance, or a risk score with unacceptable performance *and* a certificate proving that the constraints were too restrictive. By comparing certifiably optimal risk scores for different sets of constraints, we can make informed choices between models that obey different kinds of requirements.

Considering these potential benefits, a key goal of this paper is to recover certifiably optimal solutions to the risk score problem. As we will show, solving the risk score problem with a commercial MINLP solver is time-consuming even on small datasets, as generic MINLP algorithms struggle with excessive data-related computation. Accordingly, we aim to solve the risk score problem with a *cutting plane algorithm*, which reduces data-related computation by solving a surrogate problem with a linear approximation of the loss function that is much cheaper to evaluate. Cutting plane algorithms have an impressive track record on large supervised learning problems, as they scale linearly with the number of samples and provide control over data-related computation (see e.g., Teo et al., 2009; Franc and Sonnenburg, 2009; Joachims et al., 2009).

However, prior cutting plane algorithms were designed under the assumption that the surrogate problem can be solved to optimality at each iteration. This assumption leads cutting plane algorithms to *stall* on empirical risk minimization problems with non-convex regularizers and constraints, as the time to solve the surrogate to optimality increases exponentially with each iteration. We present a cutting plane algorithm to overcome this issue. We then improve its performance with specialized techniques to generate feasible solutions, narrow the optimality gap, and reduce data-related computation. Our approach extends the benefits of cutting plane algorithms to non-convex settings, allowing us to reliably train customized risk scores on many problems of interest.

Contributions The main contributions of this paper are as follows.

- We introduce a machine learning method to build risk scores. Our method can learn models that are fully optimized for feature selection and small integer coefficients, handle application-specific constraints without parameter tuning or post-processing, and pair models with a certificate of optimality.
- We present a new cutting plane algorithm – the *lattice cutting plane algorithm* (LCPA) – to solve empirical risk minimization problems with non-convex regularizers and constraints. LCPA can be easily implemented using a MIP solver (e.g., CPLEX). It can train customized risk scores in a way that scales linearly with the number of samples in a dataset.
- We design specialized techniques for LCPA to quickly find a risk score with good performance and a small optimality gap: rounding and polishing heuristics; bound-tightening and initialization procedures; and techniques to reduce data-related computation.
- We benchmark a large collection of methods to learn risk scores from data. Our results show that our method can consistently produce risk scores with best-in-class performance in minutes. We highlight pitfalls of heuristics that are often used in practice, and propose new heuristics to address these shortcomings.
- We present results from a collaboration with the Massachusetts General Hospital where we built a customized risk score for ICU seizure prediction. Our results highlight the performance benefits and the practical benefits of our method in applications where models must obey real-world constraints.
- We provide a software package to build optimized risk scores in Python, available online at http://github.com/ustunb/risk-slim.

Organization In the remainder of Section 1, we discuss related work. In Section 2, we formally define the risk score problem. In Section 3, we present our cutting plane algorithm. In Section 4, we describe techniques to improve it. In Section 5, we benchmark methods to build risk scores. In Section 6, we discuss an application to ICU seizure prediction.

The supplement to this paper contains: proofs of all theorems (Appendix A); a primer on how risk scores are developed in practice (Appendix C); additional algorithmic improvements (Appendix E); supporting material for the experiments in Sections 3 and 4 (Appendix D); the performance benchmark in Section 5 (Appendix F); and the seizure prediction application in Section 6 (Appendix G).

Prior Work Our paper extends work that was first published in KDD (Ustun and Rudin, 2017). Real-world applications of RiskSLIM include building a screening tool for adult ADHD from a short questionnaire (Ustun et al., 2017), and building a risk score for ICU seizure prediction (Struck et al., 2017). Applications of this work have been discussed in a paper that was a finalist for the 2017 INFORMS Daniel H. Wagner Prize (Rudin and Ustun, 2018). The application to seizure prediction (Ustun et al., 2017) was awarded the 2019 INFORMS Innovative Applications in Analytics Award.

1.1. Related Work

Scoring Systems While several methods have been proposed to learn scoring systems for *decision-making* (see, e.g., Ustun and Rudin, 2016; Carrizosa et al., 2016; Van Belle et al., 2013; Billiet et al., 2018, 2017; Sokolovska et al., 2017, 2018), this work focuses on learning scoring systems for *risk assessment* (i.e., *risk scores*). Risk scores represent the majority of scoring systems that are currently used in medicine and criminal justice. These models are built to output calibrated risk estimates (see e.g., Section 6 and Van Calster and Vickers, 2015; Alba et al., 2017, for a discussion on how miscalibrated risk estimates can lead to harmful decisions in medicine). As we will show in Section 5.2, building risk scores that output calibrated risk estimates is challenging. Common heuristics in risk score development (e.g., rounding or scaling) can undermine calibration in ways that are difficult to repair.

RiskSLIM risk scores are the risk assessment counterpart to SLIM scoring systems (Ustun et al., 2013; Ustun and Rudin, 2016), which have been applied to problems such as sleep apnea screening (Ustun et al., 2016), Alzheimer's diagnosis (Souillard-Mandar et al., 2016), and recidivism prediction (Zeng et al., 2017; Rudin et al., 2019). Both RiskSLIM and SLIM models are optimized for feature selection and small integer coefficients, and can be customized to obey application-specific constraints. RiskSLIM models are designed for risk assessment and minimize the logistic loss. In contrast, SLIM models are designed for decision-making and minimize the 0–1 loss. SLIM models do not output probability estimates, and the scores will not necessarily have high AUC. However, they will perform better at the operating point on the ROC curve for which they were optimized. Optimizing the 0–1 loss is also NP-hard, so training SLIM models may not scale to datasets with large sample sizes. In practice, RiskSLIM is better-suited for applications where models must output calibrated risk estimates and/or perform well at multiple operating points along the ROC curve.

Machine Learning The cutting-plane algorithm in this work can be adapted to empirical risk minimization problems with a convex loss function, a non-convex penalty, and non-convex constraints. Such problems can be solved to learn a large class of predictive models, including: scoring systems for decision-making (Carrizosa et al., 2016; Van Belle et al., 2013; Billiet et al., 2018, 2017; Sokolovska et al., 2017); sparse rule-based models such as decision lists (see e.g, Letham et al., 2015; Angelino et al., 2018), k-of-n rules (see e.g., Chevaleyre et al., 2013) and other Boolean functions (see e.g., Malioutov and Varshney, 2013; Wang et al., 2017; Lakkaraju et al., 2016); and other ℓ_0-regularized models (Sato et al., 2017, 2016; Bertsimas et al., 2016). For each of these model types, our cutting-plane algorithm can train models that optimize the same objective function and obey the same constraints, but in a way that recovers a globally optimal solution, handles application-specific constraints, and scales linearly with the number of samples.

Our work highlights an alternative approach to build models that obey constraints on, for example, *interpretability* (see e.g. Caruana et al., 2015; Gupta et al., 2016; Rudin, 2019; Chen et al., 2018; Li et al., 2018, where interpretability is addressed through constraints on model form) *safety* (Amodei et al., 2016), *credibility* (Wang et al., 2018), and *parity* (Kamishima et al., 2011; Zafar et al., 2017). Such qualities depend on multiple model properties, which vary significantly across applications and present unknown performance

trade-offs. Existing approaches often aim to address specific types of constraints for generic models by pre-processing or post-processing (see e.g., Goh et al., 2016; Calmon et al., 2017; Wang et al., 2019). In contrast, our approach aims to address such constraints directly for a specific model class. When these models belong to a simple hypothesis class (e.g., risk scores), we can expect model performance on training data to generalize, and we can evaluate this empirically (e.g., using cross-validation). In this way, one can assess the impact of constraints on predictive performance and make informed choices between models.

Our work is part of a broader stream of research on integer programming and other discrete optimization methods in supervised learning (e.g., Carrizosa et al., 2016; Liu and Wu, 2007; Goldberg and Eckstein, 2012; Guan et al., 2009; Nguyen and Franke, 2012; Sato et al., 2017, 2016; Rudin and Ertekin, 2018; Bertsimas et al., 2016; Lakkaraju et al., 2016; Angelino et al., 2018; Chen and Rudin, 2018; Chang et al., 2012; Verwer and Zhang, 2019; Hu et al., 2019; Rudin and Wang, 2018; Goh and Rudin, 2014; Ustun et al., 2019). A unique aspect of this work is that we recover models that are certifiably optimal or have small optimality gaps (see also Ustun and Rudin, 2016; Angelino et al., 2018). Our results suggest that certifiably optimal models perform better, especially in applications where models must satisfy constraints (see e.g., Section 6).

Optimization We train risk scores by solving a MINLP with three main components: (i) a convex loss function; (ii) a non-convex feasible region (i.e., small integer coefficients and application-specific constraints); (iii) a non-convex penalty function (i.e., the ℓ_0-penalty).

In Section 3.3, we show that this MINLP requires a specialized algorithm because off-the-shelf MINLP solvers fail to solve instances for small datasets. We propose solving the risk score problem with a cutting plane algorithm. Cutting planes have been extensively studied by the optimization community (see e.g., Kelley, 1960) and applied to solve *convex* empirical risk minimization problems (Teo et al., 2007, 2009; Franc and Sonnenburg, 2008, 2009; Joachims, 2006; Joachims et al., 2009).

Our cutting plane algorithm (the Lattice Cutting Plane Algorithm – LCPA) builds a cutting plane approximation while performing branch-and-bound search. It can be easily implemented using a MIP solver with *control callbacks* (see e.g., Bai and Rubin, 2009; Naoum-Sawaya and Elhedhli, 2010, for similar uses of control callbacks). LCPA retains the key benefits of existing cutting plane algorithms on empirical risk minimization problems, but does not stall on problems with non-convex regularizers or constraints. As we discuss in Section 3.1, stalling affects many cutting plane algorithms, including variants that are not considered in machine learning (see Boyd and Vandenberghe, 2004, for a list). LCPA is similar to recent outer-approximation algorithms that been developed for convex MINLP problems (see e.g., Lubin et al., 2018), which have also been shown to outperform generic MINLP algorithms (Kronqvist et al., 2019).

2. Risk Score Problem

In what follows, we formalize the problem of learning a risk score – i.e., a classification model with the same form as the one in Figure 1. We start with a dataset of n i.i.d. training examples $(\boldsymbol{x}_i, y_i)_{i=1}^n$ where $\boldsymbol{x}_i \subseteq \mathbb{R}^{d+1}$ denotes a vector of features $[1, x_{i,1}, \ldots, x_{i,d}]^\top$ and $y_i \in \{\pm 1\}$ denotes a class label. We represent the score as a linear function $s(\boldsymbol{x}) = \langle \boldsymbol{\lambda}, \boldsymbol{x} \rangle$ where $\boldsymbol{\lambda} \subseteq \mathbb{R}^{d+1}$ is a vector of $d+1$ coefficients $[\lambda_0, \lambda_1, \ldots, \lambda_d]^\top$, and λ_0 is an intercept. In this setup, coefficient λ_j represents the points that feature j contributes to the score. Given an example with features $\boldsymbol{x}_i$, a user tallies the points to compute a score $s_i = \langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle$, and then converts the score into an estimate of predicted risk. We estimate the *predicted risk* that example i is positive through the logistic link function[1] as:

$$p_i = \Pr\left(y_i = +1 \mid \boldsymbol{x}_i\right) = \frac{1}{1 + \exp(-\langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle)}.$$

Model Desiderata Our goal is to train a risk score that is sparse, has small integer coefficients, and performs well in terms of the following measures:

1. **Calibration**: A calibrated model outputs risk predictions that match their observed risks. We assess the calibration of a model using a *reliability diagram* (see DeGroot and Fienberg, 1983), which plots the *predicted risk* (x-axis) at each score against the *observed risk* (y-axis). We estimate the observed risk for a score of s as:

$$\bar{p}_s = \frac{1}{|\{i : s_i = s\}|} \sum_{i:s_i=s} \mathbb{1}\left[y_i = +1\right].$$

   We summarize the calibration of a model over the reliability diagram using the *expected calibration error* (Naeini et al., 2015):

$$\text{CAL} = \frac{1}{n} \sum_s \sum_{i:s_i=s} |p_i - \bar{p}_s|.$$

2. **Rank Accuracy**: A rank-accurate model outputs scores that can correctly rank examples in terms of their true risk. We assess the rank accuracy of a model using the *area under the ROC curve*:

$$\text{AUC} = \frac{1}{n^+ n^-} \sum_{[i:y_i=+1]} \sum_{[k:y_k=-1]} \mathbb{1}\left[s_i > s_k\right].$$

   Here, $n^+ = |\{i : y_i = +1\}|$ and $n^- = |\{i : y_i = -1\}|$.

As discussed in Section 1.1, calibration is the primary performance objective when building a risk score. Although good calibration should ensure good rank accuracy, it is important to report AUC because trivial risk scores (i.e., risk scores that assign the same score to all examples) can have low CAL on datasets with class imbalance (see Section 5.2 for an example).

We determine the values of the coefficients by solving a mixed integer nonlinear program (MINLP), which we refer to as the *risk score problem* or RiskSlimMINLP.

1. Other risk models can be used as well, so long as they produce a concave log-likelihood.

Definition 1 (Risk Score Problem, RiskSlimMINLP)
The risk score problem is a discrete optimization problem with the form:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & l(\boldsymbol{\lambda}) + C_0 \left\|\boldsymbol{\lambda}\right\|_0 \\ \text{s.t.} \quad & \boldsymbol{\lambda} \in \mathcal{L}, \end{aligned} \tag{1}$$

where:

- $l(\boldsymbol{\lambda}) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-\langle \boldsymbol{\lambda}, y_i \boldsymbol{x}_i \rangle))$ *is the normalized logistic loss function;*
- $\left\|\boldsymbol{\lambda}\right\|_0 = \sum_{j=1}^d \mathbb{1}\left[\lambda_j \neq 0\right]$ *is the ℓ_0-seminorm;*
- $\mathcal{L} \subset \mathbb{Z}^{d+1}$ *is a set of feasible coefficient vectors (user-provided);*
- $C_0 > 0$ *is a trade-off parameter to balance fit and sparsity (user-provided).*

RiskSlimMINLP captures what we desire in a risk score. The objective minimizes the *logistic loss* for calibration and AUC, and penalizes the ℓ_0-seminorm (the count of non-zero coefficients) for sparsity. The trade-off parameter C_0 controls the balance between these competing objectives, and represents the maximum log-likelihood that is sacrificed to remove a feature from the optimal model. The constraints restrict coefficients to a set of small integers such as $\mathcal{L} = \{-5, \ldots, 5\}^{d+1}$, and may be customized to encode other model requirements such as those in Table 1.

Model Requirement	**Example**
Feature Selection	Choose between 5 to 10 total features
Group Sparsity	Include either *male* or *female* in the model but not both
Optimal Thresholding	Use at most 3 thresholds for a set of indicator variables: $\sum_{k=1}^{100} \mathbb{1}\left[age \leq k\right] \leq 3$
Logical Structure	If *male* is in model, then include *hypertension* or $bmi \geq 30$
Side Information	Predict $\Pr\left(y = +1 \mid \boldsymbol{x}\right) \geq 0.90$ when *male* = TRUE and *hypertension* = TRUE

Table 1: Model requirements that can be addressed by adding operational constraints to RiskSlimMINLP.

A *Risk-calibrated Supersparse Linear Integer Model* (RiskSLIM) is a risk score that is an optimal solution to (1). By definition, the optimal solution to RiskSlimMINLP attains the lowest value of the logistic loss among feasible models on the training data, provided that C_0 is small enough (see Appendix B for a proof). Thus, a RiskSLIM risk score is a maximum likelihood logit model that satisfies all required constraints.

Our experiments in Section 5 show that models with lower loss typically attain better calibration and AUC on the training data (see also Caruana and Niculescu-Mizil, 2004), and that this generalizes to test data due to the simplicity of our hypothesis space. There are some theoretical results to explain why minimizing the logistic loss leads to good calibration and AUC. In particular, the logistic loss is a strictly proper loss (Reid and Williamson, 2010; Ertekin and Rudin, 2011) which yields calibrated risk estimates under the parametric assumption that the true risk can be modeled using a logistic link function (see Menon

et al., 2012). Further, the work of Kotlowski et al. (2011) shows that a "balanced" version of the logistic loss forms a lower bound on 1−AUC, so minimizing the logistic loss indirectly maximizes a surrogate of AUC.

Trade-off Parameter The trade-off parameter can be restricted to values between $C_0 \in [0, l(\mathbf{0})]$. Setting $C_0 > l(\mathbf{0})$ will produce a trivial model where $\boldsymbol{\lambda}^* = \mathbf{0}$. Using an exact formulation provides an alternative way to set the trade-off parameter C_0:

- If we are given a limit on model size (e.g., $\|\boldsymbol{\lambda}\|_0 \leq R$), we can add it as a constraint in the formulation, and set C_0 to a small value such as $C_0 = 10^{-8}$. In this case, the optimal solution to RiskSlimMINLP corresponds to the model minimizing the logistic loss that obeys the model size constraint, provided that C_0 is small enough (see Appendix B).
- If we wish to set the model size in a data-driven manner (e.g., to optimize a measure of cross-validated performance), we can solve several instances of RiskSlimMINLP with a model size constraint $\|\boldsymbol{\lambda}\|_0 \leq R$, where we fix C_0 to a small value and vary the model size limit from $R = 1$ to $R = d$. This approach produces the best models over the full ℓ_0-regularization path after solving d instances of RiskSlimMINLP. In comparison, a standard approach (i.e., where we treat C_0 as a hyperparameter and solve an instance of RiskSlimMINLP without a model size constraint for different values of C_0) requires solving at least d instances, since we cannot determine (in advance) d values of C_0 that produce the full range of risk scores.

Computational Complexity Optimizing RiskSlimMINLP is a difficult computational task given that ℓ_0-regularization, minimization over integers, and MINLP problems are all NP-hard (Bonami et al., 2012). These are worst-case complexity results that mean that finding an optimal solution to RiskSlimMINLP may be intractable for high dimensional datasets. As we will show, however, RiskSlimMINLP can be solved to optimality for many real-world datasets in minutes, and in a way that scales linearly in the sample size.

Notation, Assumptions, and Terminology We let $V(\boldsymbol{\lambda}) = l(\boldsymbol{\lambda}) + C_0 \|\boldsymbol{\lambda}\|_0$ denote the objective function of RiskSlimMINLP, and let $\boldsymbol{\lambda}^* \in \operatorname{argmin}_{\boldsymbol{\lambda} \in \mathcal{L}} V(\boldsymbol{\lambda})$ denote an optimal solution. We bound the optimal values of the objective, loss, and ℓ_0-seminorm as $V(\boldsymbol{\lambda}^*) \in [V^{\min}, V^{\max}]$, $l(\boldsymbol{\lambda}^*) \in [L^{\min}, L^{\max}]$, $\|\boldsymbol{\lambda}^*\|_0 \in [R^{\min}, R^{\max}]$, respectively. We denote the set of feasible coefficients for feature j as $\mathcal{L}_j$, and let $\Lambda_j^{\min} = \min_{\lambda_j \in \mathcal{L}_j} \lambda_j$ and $\Lambda_j^{\max} = \max_{\lambda_j \in \mathcal{L}_j} \lambda_j$.

For clarity of exposition, we assume that: (i) the coefficient set contains the null vector, $\mathbf{0} \in \mathcal{L}$, which ensures that RiskSlimMINLP is always feasible; (ii) the intercept is not regularized, which means that the more precise version of the RiskSlimMINLP objective function is $V(\boldsymbol{\lambda}) = l(\boldsymbol{\lambda}) + C_0 \left\|\boldsymbol{\lambda}_{[1,d]}\right\|_0$ where $\boldsymbol{\lambda} = [\lambda_0, \boldsymbol{\lambda}_{[1,d]}]$.

We measure the optimality of a feasible solution $\boldsymbol{\lambda}' \in \mathcal{L}$ in terms of its *optimality gap*, defined as $\frac{V(\boldsymbol{\lambda}') - V^{\min}}{V(\boldsymbol{\lambda}')}$. Given an algorithm to solve RiskSlimMINLP, we denote the best feasible solution that the algorithm returns in a fixed time as $\boldsymbol{\lambda}^{\text{best}} \in \mathcal{L}$. The optimality gap of $\boldsymbol{\lambda}^{\text{best}}$ is computed using an upper bound set as $V^{\max} = V(\boldsymbol{\lambda}^{\text{best}})$, and a lower bound $V^{\min}$ that is provided by the algorithm. We say that the algorithm has solved RiskSlimMINLP to *optimality* if $\boldsymbol{\lambda}^{\text{best}}$ has an optimality gap of $\varepsilon = 0.0\%$. This implies that it has found a best feasible solution to RiskSlimMINLP and produced a lower bound $V^{\min} = V(\boldsymbol{\lambda}^*)$.

3. Methodology

In this section, we present the cutting plane algorithm that we use to solve the risk score problem. In Section 3.1, we provide a brief introduction of cutting plane algorithms to discuss their practical benefits and to explain why existing algorithms stall on non-convex problems. In Section 3.2, we present a new cutting plane algorithm that does not stall. In Section 3.3, we compare the performance of cutting plane algorithms to a commercial MINLP solver on instances of the risk score problem.

3.1. Cutting Plane Algorithms

In Algorithm 1, we present a simple cutting plane algorithm to solve RiskSlimMINLP that we call CPA.

CPA recovers the optimal solution to RiskSlimMINLP by repeatedly solving a *surrogate problem* that optimizes a linear approximation of the loss function $l(\boldsymbol{\lambda})$. The approximation is built using *cutting planes* or *cuts*. Each cut is a supporting hyperplane to the loss function at a fixed point $\boldsymbol{\lambda}^t \in \mathcal{L}$:

$$l(\boldsymbol{\lambda}^t) + \langle \nabla l(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t \rangle.$$

Here, $l(\boldsymbol{\lambda}^t) \in \mathbb{R}_+$ and $\nabla l(\boldsymbol{\lambda}^t) \in \mathbb{R}^{d+1}$ are *cut parameters* that can be computed by evaluating the value and gradient of the loss at the point $\boldsymbol{\lambda}^t$:

$$l(\boldsymbol{\lambda}^t) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-\langle \boldsymbol{\lambda}^t, y_i \boldsymbol{x}_i \rangle)), \quad \nabla l(\boldsymbol{\lambda}^t) = \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i \boldsymbol{x}_i}{1 + \exp(-\langle \boldsymbol{\lambda}^t, y_i \boldsymbol{x}_i \rangle)}. \quad (3)$$

As shown in Figure 2, we can construct a *cutting plane approximation* of the loss function by taking the pointwise maximum of multiple cuts. In what follows, we denote the cutting plane approximation of the loss function built using k cuts as:

$$\hat{l}^k(\boldsymbol{\lambda}) = \max_{t=1\ldots k} \left[l(\boldsymbol{\lambda}^t) + \langle \nabla l(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t \rangle \right].$$

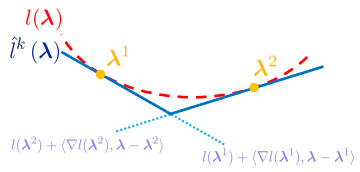


Figure 2: A convex loss function $l(\boldsymbol{\lambda})$ and its cutting plane approximation $\hat{l}^2(\boldsymbol{\lambda})$.

On iteration k, CPA solves a surrogate mixed-integer program (MIP) that minimizes the cutting plane approximation $\hat{l}^k$, namely RiskSlimMIP($\hat{l}^k$). CPA uses the optimal solution to the surrogate MIP $(L^k, \boldsymbol{\lambda}^k)$ in two ways: (i) it computes a new cut at $\boldsymbol{\lambda}^k$ to improve the cutting plane approximation; (ii) it computes bounds on optimal value of RiskSlimMINLP

Algorithm 1 Cutting Plane Algorithm (CPA)

Input

$(\boldsymbol{x}_i, y_i)_{i=1}^n$	training data
$\mathcal{L}$	coefficient set
C_0	ℓ_0 penalty parameter
$\varepsilon^{\text{stop}} \in [0,1]$	maximum optimality gap of acceptable solution

Initialize

$k \leftarrow 0$	iteration counter
$\hat{l}^0(\boldsymbol{\lambda}) \leftarrow \{0\}$	cutting plane approximation
$(V^{\min}, V^{\max}) \leftarrow (0, \infty)$	bounds on the optimal value of RiskSlimMINLP
$\varepsilon \leftarrow \infty$	optimality gap

1: **while** $\varepsilon > \varepsilon^{\text{stop}}$ **do**
2: $\quad (L^k, \boldsymbol{\lambda}^k) \leftarrow$ provably optimal solution to RiskSlimMIP($\hat{l}^k$)
3: $\quad$ compute cut parameters $l(\boldsymbol{\lambda}^k)$ and $\nabla l(\boldsymbol{\lambda}^k)$
4: $\quad \hat{l}^{k+1}(\boldsymbol{\lambda}) \leftarrow \max\{\hat{l}^k(\boldsymbol{\lambda}), l(\boldsymbol{\lambda}^k) + \langle \nabla l(\boldsymbol{\lambda}^k), \boldsymbol{\lambda} - \boldsymbol{\lambda}^k \rangle\}$ *update approximate loss function* $\hat{l}^k$
5: $\quad V^{\min} \leftarrow L^k + C_0 \left\| \boldsymbol{\lambda}^k \right\|_0$ *optimal value of* RiskSlimMIP *is lower bound*
6: $\quad$ **if** $V(\boldsymbol{\lambda}^k) < V^{\max}$ **then**
7: $\quad\quad V^{\max} \leftarrow V(\boldsymbol{\lambda}^k)$ *update upper bound*
8: $\quad\quad \boldsymbol{\lambda}^{\text{best}} \leftarrow \boldsymbol{\lambda}^k$ *update incumbent*
9: $\quad$ **end if**
10: $\quad \varepsilon \leftarrow 1 - V^{\min}/V^{\max}$
11: $\quad k \leftarrow k + 1$
12: **end while**

Output: $\boldsymbol{\lambda}^{\text{best}}$ — ε-optimal solution to RiskSlimMINLP

RiskSlimMIP($\hat{l}^k$) is a surrogate problem for RiskSlimMINLP that minimizes a cutting plane approximation $\hat{l}^k$ of the loss function l:

$$\begin{aligned} \min_{L,\boldsymbol{\lambda}} \quad & L + C_0 \left\| \boldsymbol{\lambda} \right\|_0 \\ \text{s.t.} \quad & L \geq \hat{l}^k(\boldsymbol{\lambda}) \\ & \boldsymbol{\lambda} \in \mathcal{L}. \end{aligned} \tag{2}$$

We present a MIP formulation for RiskSlimMIP($\hat{l}^k$) in Appendix D.2.

to check for convergence. Here, the upper bound is set as the objective value of the best solution across all iterations:

$$V^{\max} = \min_{t=1\ldots k} \left[l(\boldsymbol{\lambda}^t) + C_0 \|\boldsymbol{\lambda}^t\|_0 \right].$$

The lower bound is set as the optimal value of the surrogate problem at the current iteration:

$$V^{\min} = \hat{l}^k(\boldsymbol{\lambda}^k) + C_0 \|\boldsymbol{\lambda}^k\|_0.$$

CPA converges to an optimal solution of RiskSlimMINLP in a finite number of iterations (see e.g., Kelley, 1960, for a proof). In particular, the cutting plane approximation of a convex loss function improves monotonically with each cut:

$$\hat{l}^k(\boldsymbol{\lambda}) \leq \hat{l}^{k+m}(\boldsymbol{\lambda}) \leq l(\boldsymbol{\lambda}) \text{ for all } \boldsymbol{\lambda} \in \mathcal{L} \text{ and } k, m \in \mathbb{N}.$$

Since the cuts added at each iteration are not redundant, the lower bound improves monotonically with each iteration. Once the optimality gap ε is less than a stopping threshold $\varepsilon^{\text{stop}}$, CPA terminates and returns an ε-optimal solution $\boldsymbol{\lambda}^{\text{best}}$ to RiskSlimMINLP.

Key Benefits of Cutting Plane Algorithms CPA has three important properties that motivate why we want to use a cutting plane algorithm to solve the risk score problem:

(i) *Scalability in the Sample Size*: Cutting plane algorithms use the training data only when computing cut parameters, and not while solving RiskSlimMIP. Since the parameters in (3) can be computed using elementary matrix-vector operations in $\mathrm{O}(nd)$ time at each iteration, running time scales *linearly* in n for fixed d (see Figure 3).

(ii) *Control over Data-related Computation*: Cutting plane algorithms compute cut parameters in a single isolated step (e.g., Step 3 in Algorithm 1). Users can reduce data-related computation by customizing their implementation to compute these cut parameters efficiently (e.g., via distributed computing, or techniques that exploit structural properties of a specific model class as in Section E.2).

(iii) *Ability to use a MIP Solver*: Cutting plane algorithms have a special benefit in our setting since the surrogate problem can be solved with a MIP solver (rather than a MINLP solver). MIP solvers provide a fast implementation of branch-and-bound search and other features to speed up the search process (e.g., built-in heuristics, preprocessing and cut generation procedures, lazy evaluation of cut constraints, and control callbacks that let us customize the search with specialized techniques). As we show in Figure 6, using a MIP solver can substantially improve our ability to solve RiskSlimMINLP, despite the fact that one may have to solve multiple MIPs.

Stalling in Non-Convex Settings Cutting plane algorithms for empirical risk minimization (Joachims, 2006; Franc and Sonnenburg, 2008; Teo et al., 2009) are similar to CPA in that they solve a surrogate optimization problem at each iteration (e.g., Step 2 of Algorithm 1). When these algorithms are used to solve convex risk minimization problems, the surrogate is convex and therefore tractable. When the algorithms are used to solve risk minimization problems with non-convex regularizers or constraints, however, the surrogate

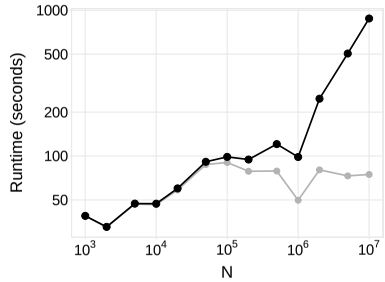


Figure 3: Runtime of CPA on synthetic datasets with $d = 10$ and $n \in [10^3, 10^7]$ (see Appendix D for details). As n increases, the runtime for the solver (grey) remains roughly constant. The total runtime (black) scales at $O(n)$, which reflects the scalability of matrix-vector operations used to compute cut parameters.

is non-convex. In these settings, cutting plane algorithms will typically *stall* as they eventually reach an iteration where the surrogate problem cannot be solved to optimality within a fixed time limit.

In Figure 4, we illustrate the stalling behavior of CPA on a difficult instance of RiskSlimMINLP for a synthetic dataset where $d = 20$ (see also Figure 6). As shown, the first iterations terminate quickly as the surrogate problem RiskSlimMIP contains a trivial approximation of the loss. Since the surrogate becomes increasingly difficult to optimize with each iteration, however, the time to solve RiskSlimMIP increases exponentially, leading CPA to stall at iteration $k = 86$. In this case, the solution returned by CPA after 6 hours has a large optimality gap and a highly suboptimal loss. This is unsurprising, as the solution was obtained by optimizing a low-fidelity approximation of the loss (i.e., an 85-cut approximation of a 20-dimensional function). Since the value of the loss is tied to the performance of the model (see Section 5), the solution corresponds to a risk score with poor performance.

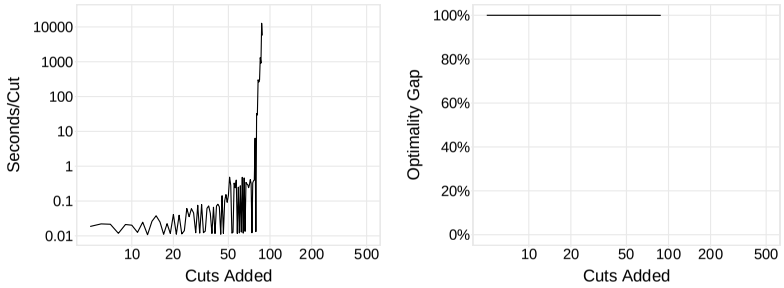


Figure 4: Performance profile of CPA on RiskSlimMINLP for a synthetic dataset with $n = 50{,}000$ and $d = 20$ (see Appendix D for details). We plot the time per iteration (left, in log-scale) and optimality gap (right) for each iteration over 6 hours. CPA stalls on iteration 86, at which point the time to solve RiskSlimMIP to optimality increases exponentially. The best solution obtained after 6 hours corresponds to a risk score with poor performance.

There is no simple fix to prevent standard cutting plane algorithms such as CPA from stalling on non-convex problems. This is because they need a globally optimal solution to a surrogate optimization problem at each iteration to compute a valid lower bound. In non-convex risk minimization problems, this requires finding the optimal solution of a non-convex surrogate problem, and *certifying* that there does not exist a better solution to the surrogate problem. If, for example, CPA only solved the surrogate until it found a feasible solution with a non-zero optimality gap, then it could produce a cutting plane that discards the true optimal solution. In this case, the lower bound computed in Step 5 would exceed the true optimal value, leading the algorithm to terminate prematurely and return a suboptimal solution with invalid bounds.

3.2. The Lattice Cutting Plane Algorithm

To avoid stalling in non-convex settings, we solve the risk score problem using the *lattice cutting plane algorithm* (LCPA) shown in Algorithm 2. LCPA has the same benefits as other cutting plane algorithms for the risk score problem, such as scalability in the sample size, control over data-related computation, and the ability to use a MIP solver. As shown in Figure 5, however, LCPA does not stall. This is because it can add cuts and compute a lower bound without having to optimize a non-convex surrogate.

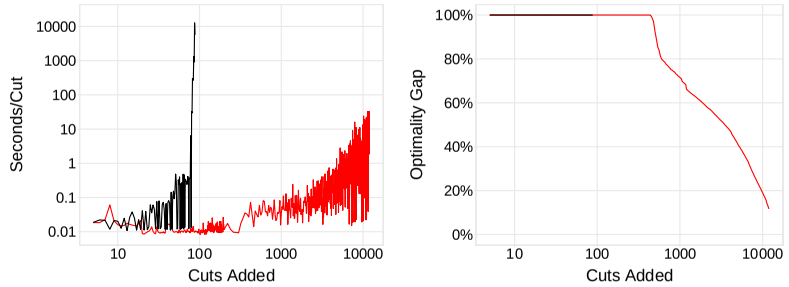


Figure 5: Performance profile of LCPA (red) and CPA (black) on the RiskSlimMINLP instance in Figure 4. Unlike CPA, LCPA does not stall. LCPA recovers a high-quality risk score (i.e., whose objective value is $\leq 10\%$ of the optimal value) in 9 minutes after adding 4,655 cuts. The remaining time is used to reduce the optimality gap.

LCPA recovers the optimal solution to RiskSlimMINLP via *branch-and-bound* (B&B) search. The search process recursively splits the feasible region of RiskSlimMINLP, discarding parts that are infeasible or provably suboptimal. LCPA solves a *surrogate linear program* (LP) over each region. It updates the cutting plane approximation when the surrogate LP yields an integer feasible solution. At that point, it sets the lower bound for the risk score problem as the smallest lower bound of the surrogate LP over unexplored regions.

Algorithm 2 Lattice Cutting Plane Algorithm (LCPA)

Input

$(\boldsymbol{x}_i, y_i)_{i=1}^n$	training data
$\mathcal{L}$	coefficient set
C_0	ℓ_0 penalty parameter
$\varepsilon^{\text{stop}} \in [0,1]$	optimality gap of acceptable solution
RemoveNode	procedure to remove node from a node set (provided by MIP solver)
SplitRegion	procedure to split region into disjoint subsets (provided by MIP solver)
RiskSlimLP$(\hat{l}, \mathcal{R})$	LP relaxation of RiskSlimMIP$(\hat{l})$ over the region $\mathcal{R} \subseteq \text{conv}(\mathcal{L})$ (see Definition 2)

Initialize

$k \leftarrow 0$	number of cuts
$\hat{l}^k(\boldsymbol{\lambda}) \leftarrow \{0\}$	cutting plane approximation
$(V^{\min}, V^{\max}) \leftarrow (0, \infty)$	bounds on the optimal value of RiskSlimMINLP
$\mathcal{R}^0 \leftarrow \text{conv}(\mathcal{L})$	initial region is convex hull of coefficient set
$v^0 \leftarrow 0$	lower bound of the objective value of the surrogate LP at $\mathcal{R}^0$
$\mathcal{N} \leftarrow \{(\mathcal{R}^0, v^0)\}$	node set
$\varepsilon \leftarrow \infty$	optimality gap

1: **while** $\varepsilon > \varepsilon^{\text{stop}}$ **do**
2: $\quad(\mathcal{R}^t, v^t) \leftarrow$ RemoveNode$(\mathcal{N})$ — *t is index of removed node*
3: $\quad$solve RiskSlimLP$(\hat{l}^k, \mathcal{R}^t)$
4: $\quad\boldsymbol{\lambda}^t \leftarrow$ coefficients from optimal solution to RiskSlimLP$(\hat{l}^k, \mathcal{R}^t)$
5: $\quad V^t \leftarrow$ optimal value of RiskSlimLP$(\hat{l}^k, \mathcal{R}^t)$
6: $\quad$**if** optimal solution is integer feasible **then**
7: $\quad\quad$compute cut parameters $l(\boldsymbol{\lambda}^t)$ and $\nabla l(\boldsymbol{\lambda}^t)$
8: $\quad\quad\hat{l}^{k+1}(\boldsymbol{\lambda}) \leftarrow \max\{\hat{l}^k(\boldsymbol{\lambda}), l(\boldsymbol{\lambda}^t) + \langle \nabla l(\boldsymbol{\lambda}^k), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t \rangle\}$ — *update approximate loss function $\hat{l}^k$*
9: $\quad\quad$**if** $V^t < V^{\max}$ **then**
10: $\quad\quad\quad V^{\max} \leftarrow V^t$ — *update lower bound*
11: $\quad\quad\quad\boldsymbol{\lambda}^{\text{best}} \leftarrow \boldsymbol{\lambda}^t$ — *update best solution*
12: $\quad\quad\quad\mathcal{N} \leftarrow \mathcal{N} \setminus \{(\mathcal{R}^s, v^s) \mid v^s \geq V^{\max}\}$ — *prune suboptimal nodes*
13: $\quad\quad$**end if**
14: $\quad\quad k \leftarrow k + 1$
15: $\quad$**else if** optimal solution is not integer feasible **then**
16: $\quad\quad(\mathcal{R}', \mathcal{R}'') \leftarrow$ SplitRegion$(\mathcal{R}^t, \boldsymbol{\lambda}^t)$ — *$\mathcal{R}'$, $\mathcal{R}''$ are disjoint subsets of $\mathcal{R}^t$*
17: $\quad\quad\mathcal{N} \leftarrow \mathcal{N} \cup \{(\mathcal{R}', V^t), (\mathcal{R}'', V^t)\}$ — *V^t is lower bound of* RiskSlimLP *for child regions $\mathcal{R}', \mathcal{R}''$*
18: $\quad$**end if**
19: $\quad V^{\min} \leftarrow \min_{s=1..|\mathcal{N}|} v^s$ — *$V^{\min}$ is smallest lower bound among nodes in $\mathcal{N}$*
20: $\quad\varepsilon \leftarrow 1 - V^{\min}/V^{\max}$ — *update optimality gap*
21: **end while**

Output: $\boldsymbol{\lambda}^{\text{best}}$ — ε-optimal solution to RiskSlimMINLP

Definition 2 (RiskSlimLP)
Given a bounded convex region $\mathcal{R} \subseteq \text{conv}(\mathcal{L})$, trade-off parameter $C_0 > 0$, cutting plane approximation $\hat{l}^k : \mathbb{R}^{d+1} \to \mathbb{R}_+$ with cut parameters $\{l(\boldsymbol{\lambda}^t), \nabla l(\boldsymbol{\lambda}^t)\}_{t=1}^k$, and bounds $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, $R^{\min}$, $R^{\max}$, the surrogate optimization problem RiskSlimLP$(\hat{l}^k, \mathcal{R})$ *can be formulated as the linear program:*

$$
\begin{array}{llcll}
\min\limits_{L,\boldsymbol{\lambda},\alpha} & V & & & \\
\text{s.t.} & V & = & L + C_0 R & & \textit{objective value} \\
 & R & = & \sum\limits_{j=1}^{d} \alpha_j & & \textit{relaxed } \ell_0\textit{-norm} \\
 & L & \geq & l(\boldsymbol{\lambda}^t) + \langle \nabla l(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t \rangle & t = 1,\ldots,k & \textit{cut constraints} \\
 & \lambda_j & \leq & \Lambda_j^{\max} \alpha_j & j = 1,\ldots,d & \ell_0\textit{-indicator constraints} \\
 & \lambda_j & \geq & -\Lambda_j^{\min} \alpha_j & j = 1,\ldots,d & \ell_0\textit{-indicator constraints} \\
 & & & & & \\
 & \boldsymbol{\lambda} & \in & \mathcal{R} & & \textit{feasible region} \\
 & V & \in & [V^{\min}, V^{\max}] & & \textit{objective bounds} \\
 & L & \in & [L^{\min}, L^{\max}] & & \textit{loss bounds} \\
 & R & \in & [R^{\min}, R^{\max}] & & \textit{relaxed } \ell_0\textit{-bounds} \\
 & \alpha_j & \in & [0,1] & j = 1,\ldots,d & \textit{relaxed } \ell_0\textit{-indicators}
\end{array}
$$

Branch-and-Bound Search In Algorithm 2, we represent the state of the B&B search process using a B&B tree. We refer to each leaf of the tree as a *node*, and denote the set of all nodes as $\mathcal{N}$. Each *node* $(\mathcal{R}^t, v^t) \in \mathcal{N}$ consists of: a *region* of the convex hull of the coefficient set $\mathcal{R}^t \subseteq \text{conv}(\mathcal{L})$; and a lower bound on the objective value of the surrogate LP over this region v^t.

Each iteration of LCPA removes a node from the node set $(\mathcal{R}^t, v^t) \in \mathcal{N}$, then solves the surrogate LP for the corresponding region, that is, RiskSlimLP$(\hat{l}^k, \mathcal{R}^t)$. Subsequent steps of the algorithm are determined by the solution status of the surrogate LP:

- If RiskSlimLP$(\hat{l}^k, \mathcal{R}^t)$ has an integer solution, LCPA updates the cutting plane approximation $\hat{l}^k$ with a new cut at $\boldsymbol{\lambda}^t$ in Step 8.
- If RiskSlimLP$(\hat{l}^k, \mathcal{R}^t)$ has a real-valued solution, LCPA adds two child nodes $(\mathcal{R}', v^t)$ and $(\mathcal{R}'', v^t)$ to the node set $\mathcal{N}$ in Step 17. The child nodes are produced by applying a splitting rule, which splits $\mathcal{R}^t$ into disjoint regions $\mathcal{R}'$ and $\mathcal{R}''$. The lower bound for each child node is set as the optimal value of the surrogate LP v^t.
- If RiskSlimLP$(\hat{l}^k, \mathcal{R}^t)$ is infeasible, then LCPA discards the node from the node set.

The B&B search is governed by two procedures that are implemented in a MIP solver:

- RemoveNode, which removes a node $(\mathcal{R}^t, v^t)$ from the node set $\mathcal{N}$ (e.g., the node with the smallest lower bound v^t).
- SplitRegion, which splits $\mathcal{R}^t$ into disjoint subsets of $\mathcal{R}^t$ (e.g., split on a fractional component of $\boldsymbol{\lambda}^t$, which returns $\mathcal{R}' = \{\boldsymbol{\lambda} \in \mathcal{R}^t \mid \lambda_j^t \geq \lceil \lambda_j^t \rceil\}$ and $\mathcal{R}'' = \{\boldsymbol{\lambda} \in \mathcal{R}^t \mid \lambda_j^t \leq \lfloor \lambda_j^t \rfloor\}$). The output conditions for SplitRegion must ensure that the regions at each node remain disjoint, the total number of nodes remains finite, and the total search region shrinks even when the surrogate LP has a real-valued solution.

LCPA evaluates the optimality of solutions to the risk score problem by using bounds on the objective value of RiskSlimMINLP. The upper bound $V^{\max}$ is set as the objective value of the best integer feasible solution in Step 10. The lower bound $V^{\min}$ is set as the smallest objective value among all nodes in Step 19. The value of $V^{\min}$ can be viewed as a lower bound on the objective value of the surrogate LP over the *remaining search region* $\bigcup_t \mathcal{R}^t$ (i.e., $V^{\min}$ is a lower bound on the objective value of $\text{RiskSlimLP}(\hat{l}^k, \bigcup_t \mathcal{R}^t)$). Thus, $V^{\min}$ will increase when we reduce the remaining search region or add cuts.

Each iteration of LCPA reduces the remaining search region by either finding an integer feasible solution, identifying an infeasible region, or splitting a region into disjoint subsets. Thus, $V^{\min}$ increases monotonically as the search region becomes smaller, and cuts are added at integer feasible solutions. Likewise, $V^{\max}$ decreases monotonically as it is set as the objective value of the best integer feasible solution. Since there are a finite number of nodes, even in the worst-case, LCPA terminates after a finite number of iterations, returning an optimal solution to the risk score problem.

Remark 3 (Worst-Case Data-Related Computation for LCPA)
Given any training dataset $(\boldsymbol{x}_i, y_i)_{i=1}^n$, any trade-off parameter $C_0 > 0$, and any finite coefficient set $\mathcal{L} \subset \mathbb{Z}^{d+1}$, LCPA *returns an optimal solution to the risk score problem after computing at most $|\mathcal{L}|$ cutting planes, and processing at most $2^{|\mathcal{L}|} - 1$ nodes.*

Implementation with a MIP Solver with Lazy Cut Evaluation LCPA can easily be implemented using a MIP solver (e.g., CPLEX, Gurobi, GLPK) with *control callbacks*. In this approach, the solver handles the B&B related steps of Algorithm 2, and one needs only to write a few lines of code to update the cutting plane approximation when the algorithm finds an integer feasible solution. In a basic implementation, the solver would call the control callback when it finds an integer feasible solution (i.e., Step 6). The code would retrieve the integer feasible solution, compute the cut parameters, and add a cut to the surrogate LP, handing control back to the solver at Step 9.

A key benefit of using a MIP solver is the ability to add cuts as *lazy constraints*. In practice, if we were to add cuts as generic constraints to the surrogate LP, the time to solve the surrogate LP would increase with each cut, which would progressively slow down LCPA. When we add cuts as lazy constraints, the solver branches using a surrogate LP that contains a subset of relevant cuts, and only evaluates the complete set of cuts when LCPA finds an integer feasible solution. In this case, LCPA still returns the optimal solution. However, computation is significantly reduced as the surrogate LP is much faster to solve for the vast majority of cases where it is infeasible or yields a real-valued solution. From a design perspective, lazy cut evaluation reduces the marginal computational cost of adding cuts, which allows us to add cuts liberally (i.e., without having to worry about slowing down LCPA by adding too many cuts).

3.3. Performance Comparison with MINLP Algorithms

In what follows, we benchmark CPA and LCPA against three MINLP algorithms as implemented in a commercial MINLP solver (Artelsys Knitro 9.0, which is an updated version of the solver in Byrd et al., 2006).

In Figure 6, we show the performance of algorithms on difficult instances of the risk score problem for synthetic datasets with d dimensions and n samples (see Appendix D for details). We consider the following performance metrics: (i) the time to find a near-optimal solution; (ii) the optimality gap of the best solution at termination; and (iii) the proportion of time spent on data-related computation. Since all three MINLP algorithms behave similarly, we only show the best one in Figure 6 (i.e., ActiveSetMINLP), and include results for the others in Appendix D.3.

As shown, LCPA finds an optimal or near-optimal solution for almost all instances of the risk score problem, and pairs the solution with a small optimality gap. CPA performs similarly to LCPA on low-dimensional instances. On instances with $d \geq 15$, however, CPA stalls after a few iterations and returns a highly suboptimal solution (i.e., a risk score with poor performance). In comparison, the MINLP algorithms can only handle instances with small n or d. On larger instances, the solver is slowed down by operations that involve data-related computation, fails to converge within the 6-hour time limit and fails to recover a high-quality solution. Seeing how MINLP solvers are designed to solve a diverse set of optimization problems, we do not believe that they can identify and exploit the structure of the risk score problem in the same way as a cutting plane algorithm.

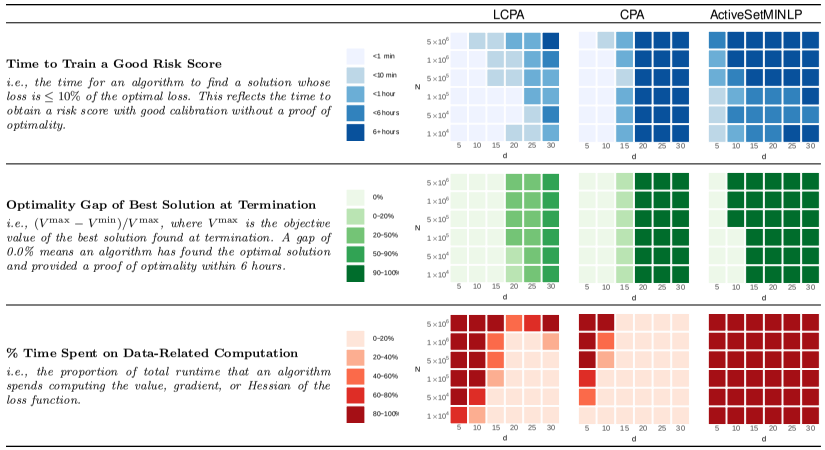


Figure 6: Performance of LCPA, CPA, and a commercial MINLP solver on difficult instances of RiskSlimMINLP for synthetic datasets with d dimensions and n samples (see Appendix D for details). ActiveSetMINLP fails to produce good risk scores on instances with large n or d as it struggles with data-related computation. CPA and LCPA scale linearly in n when d is fixed: if they solve an instance for a given d, then they can solve instances for larger n in $O(n)$ additional time. CPA stalls when $d \geq 15$ and returns a low-quality risk score when $d \geq 20$. In contrast, LCPA consistently recovers a good model without stalling. Results reflect the performance for a basic LCPA implementation without the improvements in Section 4. We show results for two other MINLP algorithms in Appendix D.

4. Algorithmic Improvements

In this section, we describe specialized techniques to improve the performance of the lattice cutting plane algorithm (LCPA) on the risk score problem. They include:

- *Polishing Heuristic.* We present a technique called discrete coordinate descent (DCD; Section 4.1), which we use to polish integer solutions found by LCPA (solutions satisfying the condition in Step 6). DCD aims to improve the objective value of all integer solutions, which produces stronger upper bounds over the course of LCPA, and reduces the time to recover a good risk score.
- *Rounding Heuristic.* We present a rounding technique called SequentialRounding (Section 4.2) to generate integer solutions. We use SequentialRounding to round real-valued solutions of the surrogate LP (which are solutions that satisfy the condition in Step 15) and then polish the rounded solution with DCD. Rounded solutions may improve the best solution found by LCPA, producing stronger upper bounds and reducing the time to recover a good risk score.
- *Bound Tightening Procedure.* We design a fast procedure to strengthen bounds on the optimal values of the objective, loss, and number of non-zero coefficients called ChainedUpdates (Section 4.3). We call ChainedUpdates whenever the solver updates the upper bound in Step 10 or the lower bound in Step 19. ChainedUpdates improves the lower bound, and reduces the optimality gap of the final risk score.

We present additional techniques to improve LCPA in Appendix E such as an initialization procedure and techniques to reduce data-related computation.

4.1. Discrete Coordinate Descent

Discrete coordinate descent (DCD) is a technique to polish an integer solution (Algorithm 3). It takes as input an integer solution $\boldsymbol{\lambda} = [\lambda_0, \ldots, \lambda_d]^\top \in \mathcal{L}$ and iteratively changes a single coordinate j to attain an integer solution with a better objective value. The coordinate at each iteration is set to minimize the objective value, i.e., $j \in \operatorname{argmin}_{j'} V\left(\boldsymbol{\lambda} + \delta_{j'} e_{j'}\right)$.

DCD terminates once it can no longer strictly improve the objective value along any coordinate. This eliminates the potential of cycling, and thereby guarantees that the procedure will terminate in a finite number of iterations. The polished solution returned by DCD satisfies a type of local optimality guarantee for discrete optimization problems. Formally, it is *1-opt* with respect to the objective value, meaning that one cannot improve the objective value by changing any single coefficient (see e.g., Park and Boyd, 2018, for a technique to find a 1-opt point for a different optimization problem).

In practice, the most expensive computation in DCD is finding a step-size $\delta_j \in \Delta_j$ that minimizes the objective along coordinate j (Step 5 of Algorithm 3). We can significantly reduce this computation by using golden section search. This approach requires $nd \log_2 |\mathcal{L}_j|$ flops per iteration compared to $nd|\mathcal{L}_j|$ flops per iteration required by a brute force approach (i.e., which evaluates the loss for all $\lambda_j \in \mathcal{L}_j$).

In Figure 7, we show how DCD improves the performance of LCPA when we use it to polish feasible solutions found by the MIP solver (i.e., the polishing is placed just after Step 6 of Algorithm 2).

Algorithm 3 Discrete Coordinate Descent (DCD)

Input

$(\boldsymbol{x}_i, y_i)_{i=1}^n$	training data
$\mathcal{L}$	coefficient set
C_0	ℓ_0 penalty parameter
$\boldsymbol{\lambda} \in \mathcal{L}$	integer solution to RiskSlimMINLP
$\mathcal{J} \subseteq \{0, \ldots, d\}$	valid descent directions

1: **repeat**
2: $\quad V \leftarrow V(\boldsymbol{\lambda})$ — *objective value at current solution*
3: $\quad$ **for** $j \in \mathcal{J}$ **do**
4: $\quad\quad \Delta_j \leftarrow \{\delta \in \mathbb{Z} \mid \boldsymbol{\lambda} + \delta e_j \in \mathcal{L}\}$ — *list feasible moves along dim j*
5: $\quad\quad \delta_j \leftarrow \operatorname{argmin}_{\delta \in \Delta_j} V(\boldsymbol{\lambda} + \delta)$ — *find best move in dim j*
6: $\quad\quad v_j \leftarrow V(\boldsymbol{\lambda} + \delta_j e_j)$ — *store objective value for best move in dim j*
7: $\quad$ **end for**
8: $\quad m \leftarrow \operatorname{argmin}_{j \in \mathcal{J}} v_j$ — *descend along dim that minimizes objective*
9: $\quad \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \delta_m e_m$
10: **until** $v_m \geq V$

Output: $\boldsymbol{\lambda}$ — solution that is 1-opt with respect to the objective of RiskSlimMINLP

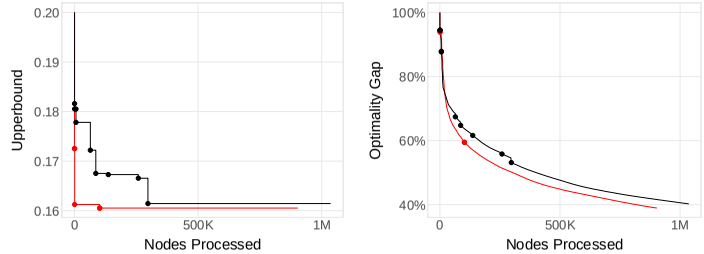


Figure 7: Performance profile of LCPA in a basic implementation (black) and with DCD (red). We use DCD to polish every integer solution found by the MIP solver whose objective value is within 10% of the current upper bound. We plot the number of total nodes processed of LCPA (x-axis) against the upper bound (y-axis; left) and the optimality gap (y-axis; right). We mark iterations where LCPA updates the incumbent solution. Results reflect performance on RiskSlimMINLP for a synthetic dataset with $d = 30$ and $n = 50{,}000$ (see Appendix D for details).

4.2. Sequential Rounding

SequentialRounding (Algorithm 4) is a rounding heuristic to generate integer solutions for the risk score problem. In comparison to naïve rounding, which returns the closest rounding from a set of 2^{d+1} possible roundings, SequentialRounding returns a rounding that iteratively finds a local optimizer of the risk score problem.

Given a real-valued solution $\boldsymbol{\lambda}^{\text{real}} \in \operatorname{conv}(\mathcal{L})$, the procedure iteratively rounds one component (up or down) in a way that reduces the objective of RiskSlimMINLP. On iteration k, it has already rounded k components of $\boldsymbol{\lambda}^{\text{real}}$, and must round one of the remaining

Algorithm 4 SequentialRounding

Input

$(\boldsymbol{x}_i, y_i)_{i=1}^n$ — training data

$\mathcal{L}$ — coefficient set

C_0 — ℓ_0 penalty parameter

$\boldsymbol{\lambda} \in \text{conv}(\mathcal{L})$ — non-integer infeasible solution from RiskSlimLP

$\mathcal{J}^{\text{real}} \leftarrow \{j : \lambda_j \neq \lceil \lambda_j \rfloor\}$ — *index set of non-integer coefficients*

repeat

  $\boldsymbol{\lambda}^{j,\text{up}} \leftarrow (\lambda_1, \ldots, \lceil \lambda_j \rceil, \ldots, \lambda_d)$ for all $j \in \mathcal{J}^{\text{real}}$

  $\boldsymbol{\lambda}^{j,\text{down}} \leftarrow (\lambda_1, \ldots, \lfloor \lambda_j \rfloor, \ldots, \lambda_d)$ for all $j \in \mathcal{J}^{\text{real}}$

  $v^{\text{up}} \leftarrow \min_{j \in \mathcal{J}^{\text{real}}} V(\boldsymbol{\lambda}^{j,\text{up}})$

  $v^{\text{down}} \leftarrow \min_{j \in \mathcal{J}^{\text{real}}} V(\boldsymbol{\lambda}^{j,\text{down}})$

  if $v^{\text{up}} < v^{\text{down}}$ **then**

    $k \leftarrow \text{argmin}_{j \in \mathcal{J}^{\text{real}}} V(\boldsymbol{\lambda}^{j,\text{up}})$ and $\lambda_k \leftarrow \lceil \lambda_k \rceil$

  else

    $k \leftarrow \text{argmin}_{j \in \mathcal{J}^{\text{real}}} V(\boldsymbol{\lambda}^{j,\text{down}})$ and $\lambda_k \leftarrow \lfloor \lambda_k \rfloor$

  end if

  $\mathcal{J}^{\text{real}} \leftarrow \mathcal{J}^{\text{real}} \setminus \{k\}$

until $\mathcal{J}^{\text{real}} = \emptyset$

Output: $\boldsymbol{\lambda} \in \mathcal{L}$ — integer solution

$d+1-k$ components to $\lceil \lambda_j^{\text{real}} \rceil$ or $\lfloor \lambda_j^{\text{real}} \rfloor$. To this end, it computes the objective of all feasible (component, direction)-pairs and chooses the best one. Formally, the minimization on iteration k requires $2 \cdot (d+1-k)$ evaluations of the loss function. Thus, given that there are $d+1$ iterations, SequentialRounding terminates after $2 \cdot \sum_{k=1}^{d} k = d(d+1)$ evaluations of the loss function.

In Figure 8, we show the impact of using SequentialRounding in LCPA. Here, we apply SequentialRounding to the non-integer solution of RiskSlimLP when the lower bound changes (i.e., just after Step 3 of Algorithm 2), then polish the rounded solution using DCD. As shown, this strategy can reduce the time required for LCPA to find a high-quality risk score, and attain a lower optimality gap.

4.3. Chained Updates

We describe a fast bound tightening technique called ChainedUpdates (Algorithm 5). This technique iteratively bounds the optimal values of the objective, loss, and ℓ_0-norm by iteratively setting the values of $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, and $R^{\max}$ in RiskSlimLP. Bounding these quantities over the course of B&B restricts the search region without discarding the optimal solution, thereby improving the lower bound and reducing the optimality gap.

Initial Bounds on Objective Terms We initialize ChainedUpdates with values of $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, and $R^{\max}$ that can be computed using only the training data $(\boldsymbol{x}_i, y_i)_{i=1}^n$ and the coefficient set $\mathcal{L}$. We start with Proposition 4, which provides initial values for $L^{\min}$ and $L^{\max}$ using the fact that the coefficient set $\mathcal{L}$ is bounded.

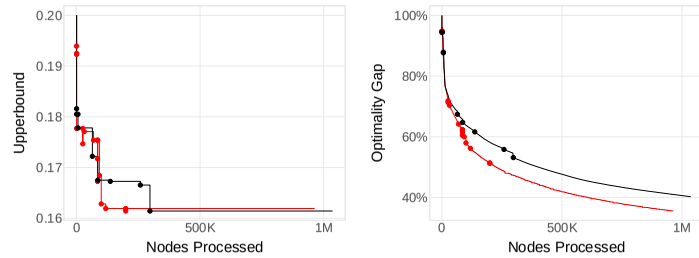


Figure 8: Performance profile of LCPA in a basic implementation (black) and with SequentialRounding and DCD polishing (red). We call SequentialRounding to round non-integer solutions to RiskSlimLP in Step 15, and then polish the integer solution with DCD. We plot large points to show when LCPA updates the incumbent solution. Results reflect performance on RiskSlimMINLP for a synthetic dataset with $d = 30$ and $n = 50{,}000$ (see Appendix D for details). Here, SequentialRounding and DCD reduce the upper bound and optimality gap of LCPA compared to a basic implementation.

Proposition 4 (Bounds on Logistic Loss over a Bounded Coefficient Set)
Given a training dataset $(\boldsymbol{x}_i, y_i)_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$ for $i = 1, \ldots, n$, consider the normalized logistic loss of a linear classifier with coefficients $\boldsymbol{\lambda}$:

$$l(\boldsymbol{\lambda}) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-\langle \boldsymbol{\lambda}, y_i \boldsymbol{x}_i \rangle)).$$

If the coefficients belong to a bounded set $\mathcal{L}$, then the value of the normalized logistic loss must obey $l(\boldsymbol{\lambda}) \in [L^{\min}, L^{\max}]$ for all $\boldsymbol{\lambda} \in \mathcal{L}$, where:

$$L^{\min} = \frac{1}{n} \sum_{i:y_i=+1} \log\left(1 + \exp(-s_i^{\max})\right) + \frac{1}{n} \sum_{i:y_i=-1} \log\left(1 + \exp(s_i^{\min})\right),$$

$$L^{\max} = \frac{1}{n} \sum_{i:y_i=+1} \log\left(1 + \exp(-s_i^{\min})\right) + \frac{1}{n} \sum_{i:y_i=-1} \log\left(1 + \exp(s_i^{\max})\right),$$

$$s_i^{\min} = \min_{\boldsymbol{\lambda} \in \mathcal{L}} \langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle \text{ for } i = 1, \ldots, n,$$

$$s_i^{\max} = \max_{\boldsymbol{\lambda} \in \mathcal{L}} \langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle \text{ for } i = 1, \ldots, n.$$

The value of $L^{\min}$ in Proposition 4 represents the "best-case" loss in a separable setting where we assign each positive example its maximal score $s_i^{\max}$, and each negative example its minimal score $s_i^{\min}$. Conversely, $L^{\max}$ represents the "worst-case" loss when we assign each positive example its minimal score $s_i^{\min}$ and each negative example its maximal score $s_i^{\max}$. We initialize the bounds on the number of non-zero coefficients R to $\in \{0, \ldots, d\}$, trivially. In some cases, these bounds may be stronger due to operational constraints (e.g., we can set $R \in \{0, \ldots, 5\}$ if models are required to use ≤ 5 features). Having initialized $L^{\min}$, $L^{\max}$, $R^{\min}$ and $R^{\max}$, we set the bounds on the optimal objective value as $V^{\min} = L^{\min} + C_0 R^{\min}$ and $V^{\max} = L^{\max} + C_0 R^{\max}$, respectively.

Algorithm 5 ChainedUpdates

Input

C_0 — ℓ_0 penalty parameter

$V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, $R^{\min}$, $R^{\max}$ — initial bounds on $V(\boldsymbol{\lambda}^*)$, $l(\boldsymbol{\lambda}^*)$ and $\|\boldsymbol{\lambda}^*\|_0$

1: **repeat**

2: $V^{\min} \leftarrow \max\left(V^{\min},\ L^{\min} + C_0 R^{\min}\right)$ — *update lower bound on* $V(\boldsymbol{\lambda}^*)$

3: $V^{\max} \leftarrow \min\left(V^{\max},\ L^{\max} + C_0 R^{\max}\right)$ — *update upper bound on* $V(\boldsymbol{\lambda}^*)$

4: $L^{\min} \leftarrow \max\left(L^{\min},\ V^{\min} - C_0 R^{\max}\right)$ — *update lower bound on* $l(\boldsymbol{\lambda}^*)$

5: $L^{\max} \leftarrow \min\left(L^{\max},\ V^{\max} - C_0 R^{\min}\right)$ — *update upper bound on* $l(\boldsymbol{\lambda}^*)$

6: $R^{\max} \leftarrow \min\left(R^{\max},\ \left\lfloor \frac{V^{\max} - L^{\min}}{C_0} \right\rfloor\right)$ — *update upper bound on* $\|\boldsymbol{\lambda}^*\|_0$

7: **until** there are no more bound updates due to Steps 2 to 6.

Output: $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, $R^{\min}$, $R^{\max}$

Dynamic Bounds on Objective Terms In Propositions 5 to 7, we present bounds that can use information from the solver in LCPA to strengthen the values of $L^{\min}$, $L^{\max}$, $R^{\max}$, $V^{\min}$ and $V^{\max}$ (see Appendix A for proofs).

Proposition 5 (Upper Bound on Optimal Number of Non-Zero Coefficients)
Given an upper bound on the optimal value $V^{\max} \geq V(\boldsymbol{\lambda}^*)$, and a lower bound on the optimal loss $L^{\min} \leq l(\boldsymbol{\lambda}^*)$, the optimal number of non-zero coefficients is at most

$$R^{\max} = \left\lfloor \frac{V^{\max} - L^{\min}}{C_0} \right\rfloor.$$

Proposition 6 (Upper Bound on Optimal Loss)
Given an upper bound on the optimal value $V^{\max} \geq V(\boldsymbol{\lambda}^*)$, and a lower bound on the optimal number of non-zero coefficients $R^{\min} \leq \|\boldsymbol{\lambda}^*\|_0$, the optimal loss is at most

$$L^{\max} = V^{\max} - C_0 R^{\min}.$$

Proposition 7 (Lower Bound on Optimal Loss)
Given a lower bound on the optimal value $V^{\min} \leq V(\boldsymbol{\lambda}^*)$, and an upper bound on the optimal number of non-zero coefficients $R^{\max} \geq \|\boldsymbol{\lambda}^*\|_0$, the optimal loss is at least

$$L^{\min} = V^{\min} - C_0 R^{\max}.$$

Implementation In Algorithm 5, we present a bound-tightening procedure that uses the results of Propositions 5 to 7 to strengthen the values of $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, and $R^{\max}$ in RiskSlimLP.

Propositions 5 to 7 impose dependencies between $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, $R^{\min}$ and $R^{\max}$ that may produce a complex "chain" of updates. As shown in Figure 9, ChainedUpdates can update multiple terms, and may update the same term more than once. Consider

a case where we call ChainedUpdates after LCPA improves $V^{\min}$. Say the procedure updates $L^{\min}$ in Step 4. If ChainedUpdates updates $R^{\max}$ in Step 6, then it will also update $V^{\max}$, $L^{\min}$, $L^{\max}$, and $V^{\min}$. However, if ChainedUpdates does not update $R^{\max}$ in Step 6, then it will not update $V^{\max}$, $L^{\min}$, $L^{\max}$, $V^{\min}$ and terminate.

Considering these dependencies, Algorithm 5 applies Propositions 5 to 7 until it can no longer improve $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$ or $R^{\max}$. This ensures that ChainedUpdates will return its strongest possible bounds regardless of the term that was first updated.

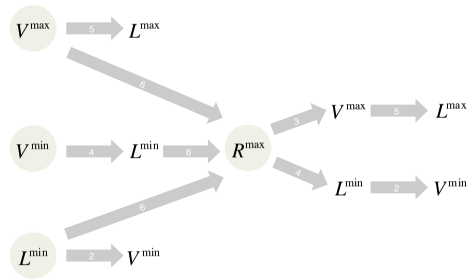


Figure 9: All possible "chains" of updates in ChainedUpdates. Circles represent "source" terms that can be updated by LCPA to trigger ChainedUpdates. The path from each source term shows all bounds that can be updated by the procedure. The number in each arrow references the update step in Algorithm 5.

In our implementation, we call ChainedUpdates whenever LCPA improves $V^{\max}$ or $V^{\min}$ (i.e., Step 10 or Step 19 of Algorithm 2). If ChainedUpdates improves any bounds, we pass this information back to the solver by updating the bounds on the auxiliary variables in the RiskSlimLP (Definition 2). As shown in Figure 10, this technique can considerably improve the lower bound and optimality gap over the course of LCPA.

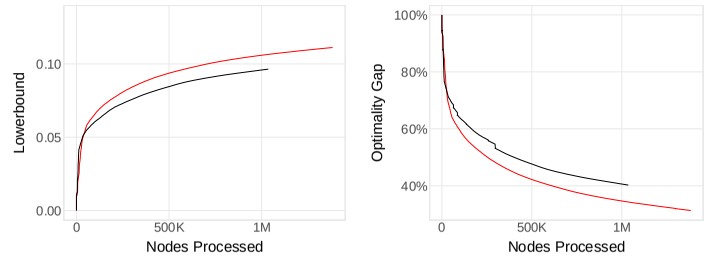


Figure 10: Performance profile of LCPA in a basic implementation (black) and with ChainedUpdates (red). Results reflect performance on a RiskSlimMINLP instance for a synthetic dataset with $d = 30$ and $n = 50{,}000$ (see Appendix D).

5. Experiments

In this section, we compare the performance of methods to create risk scores. We have three goals: (i) to benchmark the performance and computation of our approach on real-world datasets; (ii) to highlight pitfalls of traditional approaches used in practice; and (iii) to present new approaches that address the pitfalls of traditional approaches.

5.1. Setup

We considered 6 publicly available datasets shown in Table 2. We chose these datasets to see how methods are affected by factors such as class imbalance, the number of features, and feature encoding. For each dataset, we fit risk scores using RiskSLIM and 6 baseline methods that post-processed the coefficients of the best logistic regression model built using Lasso, Ridge or Elastic Net. We used each method to fit a risk score with small integer coefficients $\lambda_j \in \{-5, \ldots, 5\}$ that obeys the model size constraint $\|\boldsymbol{\lambda}\|_0 \leq R^{\max}$. We benchmarked each method for target model sizes $R^{\max} \in \{2, \ldots, 10\}$.

Dataset	n	d	$\Pr(y_i = 1)$	**Conditions for** $y_i = 1$	**Reference**
`income`	32,561	36	24.1%	person in 1994 US census earns over $50,000	Kohavi (1996)
`mammo`	961	14	46.3%	person has breast cancer	Elter et al. (2007)
`mushroom`	8,124	113	48.2%	mushroom is poisonous	Schlimmer (1987)
`rearrest`	22,530	48	59.0%	person is arrested after release from prison	Zeng et al. (2017)
`spambase`	4,601	57	39.4%	e-mail is spam	Cranor and LaMacchia (1998)
`telemarketing`	41,188	57	11.3%	person opens bank account after marketing call	Moro et al. (2014)

Table 2: Datasets used in Section 5. All datasets are available on the UCI repository (Bache and Lichman, 2013), other than `rearrest` which must be requested from ICPSR. We processed each dataset by dropping examples with missing values, and by binarizing categorical variables and some real-valued variables. We provide processed datasets and the code to process `rearrest` at http://github.com/ustunb/risk-slim.

RiskSLIM We formulated an instance of RiskSlimMINLP with the constraints: $\lambda_0 \in \{-100, \ldots, 100\}$, $\lambda_j \in \{-5, \ldots, 5\}$, and $\|\boldsymbol{\lambda}\|_0 \leq R^{\max}$. We set the trade-off parameter to a small value $C_0 = 10^{-6}$ to recover the sparsest model among equally accurate models (see Appendix B). We solved each instance for at most 20 minutes on a 3.33 GHz CPU with 16 GB RAM using CPLEX 12.6.3 (ILOG, 2017).

Penalized Logistic Regression PLR is the best logistic regression model produced over the full regularization path using a weighted combination of the ℓ_1 and ℓ_2 penalties (i.e., the best model produced by Lasso, Ridge or Elastic Net). We train PLR models using the glmnet package of Friedman et al. (2010). The coefficients of each model are the solution to the optimization problem:

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^{d+1}} \quad \frac{1}{2n} \sum_{i=1}^{n} \log(1 + \exp(-\langle \boldsymbol{\lambda}, y_i \boldsymbol{x}_i \rangle)) + \gamma \cdot \left(\alpha \|\boldsymbol{\lambda}\|_1 + (1-\alpha) \|\boldsymbol{\lambda}\|_2^2 \right)$$

where $\alpha \in [0, 1]$ is the elastic-net mixing parameter and $\gamma \geq 0$ is a regularization penalty. We trained 1,100 PLR models by choosing 1,100 combinations of (α, γ): 11 values of $\alpha \in \{0.0, 0.1, \ldots, 0.9, 1.0\} \times 100$ values of γ (chosen automatically by glmnet for each α).

This free parameter grid produces 1,100 PLR models that include models obtained by: (i) Lasso (ℓ_1-penalty), which corresponds to PLR when $\alpha = 1.0$; (ii) Ridge (ℓ_2-penalty), which corresponds to PLR when $\alpha = 0.0$; (iii) standard logistic regression, which corresponds to PLR when $\alpha = 0.0$ and γ is small.

Traditional Approaches While there is considerable variation in how risk scores are developed in practice, many researchers follow a two-step approach: (i) fit a sparse logistic regression model with real-valued coefficients; (ii) convert it into a risk score with integer coefficients. We consider three methods that adopt this approach. Each method first trains a PLR model (i.e., the one that maximizes the 5-CV AUC and obeys the model size constraint), and then converts it into a risk score by applying a common rounding heuristic:

- PLR$\gg$Rd (Rounding): We round each coefficient to the nearest integer in $\{-5 \ldots 5\}$ by setting $\lambda_j \leftarrow \lceil \min(\max(\lambda_j, -5), 5) \rfloor$, and round the intercept as $\lambda_0 \leftarrow \lceil \lambda_0 \rfloor$.
- PLR$\gg$Unit (Unit Weighting): We round each coefficient to ± 1 as $\lambda_j \leftarrow \text{sign}(\lambda_j) \mathbb{1}[\lambda_j \neq 0]$. Unit weighting is a common heuristic in medicine and criminal justice (see e.g., Antman et al., 2000; Kessler et al., 2005; U.S. Department of Justice, 2005; Duwe and Kim, 2016), and sometimes called the *Burgess method* (as it was first proposed by Burgess, 1928).
- PLR$\gg$RsRd (Rescaled Rounding) We first rescale coefficients so that the largest coefficient is ± 5, then round each coefficient to the nearest integer (i.e., $\lambda_j \rightarrow \lceil \gamma \lambda_j \rfloor$ where $\gamma = 5 / \max_j |\lambda_j|$). Rescaling is often used to avoid rounding small coefficients to zero, which happens when $|\lambda_j| < 0.5$ (see e.g., Le Gall et al., 1993).

Pooled Approaches We also propose three new methods that use a *pooling* strategy and the loss-minimizing heuristics from Section 4. Each method generates a pool of PLR models with real-valued coefficients, applies the same post-processing procedure to each model in the pool, then selects the best risk score among feasible risk scores. The methods include:

- PooledRd (Pooled PLR + Rounding): We fit a pool of 1,100 models using PLR. For each model in the pool, we round each coefficient to the nearest integer in $\{-5, \ldots, 5\}$ by setting $\lambda_j \leftarrow \lceil \min(\max(\lambda_j, -5), 5) \rfloor$, and round the intercept as $\lambda_0 \leftarrow \lceil \lambda_0 \rfloor$.
- PooledRd* (Pooled PLR + Rounding + Polishing): We fit a pool of 1,100 models using PooledRd. For each model in the pool, we polish the rounded coefficients using DCD.
- PooledSeqRd* (Pooled PLR + Sequential Rounding + Polishing): We fit a pool of 1,100 models using PLR. For each model in the pool, we round the coefficients using SequentialRounding and then polish the rounded coefficients using DCD.

To ensure that the polishing step in PooledRd*and PooledSeqRd* does not increase the number of non-zero coefficients (which would violate the model size constraint), we run DCD only on the set $\{j \mid \lambda_j \neq 0\}$ (i.e., by fixing the set of zeros coefficients).

Performance Evaluation We evaluate the calibration of each risk score by plotting a *reliability diagram*, which shows how the predicted risk (x-axis) matches the observed risk (y-axis) for each distinct score (DeGroot and Fienberg, 1983). The *observed* risk at a score of s is defined as

$$\bar{p}_s = \frac{1}{|\{i : s_i = s\}|} \sum_{i : s_i = s} \mathbb{1}[y_i = +1].$$

If a model has over 30 distinct scores, we group them into 10 bins before plotting the reliability diagram. A model with perfect calibration should output predictions that are perfectly aligned with observed risk, as shown by a reliability diagram where all points lie on the $x = y$ line.

We report the following summary statistics for each model:

- *Calibration Error*, computed as $\text{CAL} = \frac{1}{n}\sum_s \sum_{i:s_i=s} |p_i - \bar{p}_s|$ where p_i is the *predicted risk* of example i, and $\bar{p}_s$ is the observed risk for all examples with a score of s. CAL is the expected calibration error over the reliability diagram (see, e.g., Naeini et al., 2015).
- *Area under the ROC curve*, computed as $\text{AUC} = \frac{1}{n^+ n^-}\sum_{i:y_i=+1}\sum_{k:y_k=-1} \mathbb{1}\left[s_i > s_k\right]$, where $n^+ = |\{i : y_i = +1\}|$, $n^- = |\{i : y_i = -1\}|$. Note that trivial models (i.e., models that predict one class) achieve the best possible CAL (0.0%) but poor AUC (0.5).
- *Logistic Loss*, computed as $\text{Loss} = \frac{1}{n}\sum_{i=1}^n \log(1 + \exp(-y_i s_i))$. The loss reflects the objective values of the risk score problem when C_0 is small. We report the loss to see if minimizing the objective value of the risk score problem improves CAL and AUC.
- *Model Size*: the number of non-zero coefficients excluding the intercept $\sum_{j=1}^d \mathbb{1}\left[\lambda_j \neq 0\right]$.

Parameter Tuning We use *nested* 5-fold cross-validation (5-CV) to choose the free parameters of a final risk score (see Cawley and Talbot, 2010). The final risk score is fit using the entire dataset for an instance of the free parameters that satisfies the model size constraint and maximizes the 5-CV mean test AUC.

5.2. Discussion

On the Performance of Risk Scores We compare the performance of RiskSLIM to traditional approaches in Figure 11, and to pooled approaches in Figure 12. These results show that RiskSLIM models consistently attain better calibration and AUC than alternatives. We present these results in greater detail for risk scores with a target model size of $R^{\max} = 5$ in Table 3. Here, RiskSLIM has the best 5-CV mean test CAL on 5/6 datasets, the best 5-CV mean test AUC on 5/6 datasets, and no method has better test CAL *and* test AUC than RiskSLIM.

We make two observations to explain the empirical performance of risk scores.

(i) Models that attain low values of the logistic loss have good calibration (see Figures 11 and 12 and the empirical results of e.g., Caruana and Niculescu-Mizil, 2004, 2006).

(ii) Since we are fitting from a simple class of models, risk scores tend to generalize (see the test CAL/AUC and training CAL/AUC of RiskSLIM models in Figures 15 to 20, and other risk scores in Table 7 in Appendix F).

Since RiskSLIM models optimize the loss over exact constraints on model form, they attain minimal or near-minimal values of the loss. Thus, they perform well in terms of training CAL/AUC as per (i) and test CAL/AUC as per (ii). These observations also explain why methods that use loss-minimizing heuristics produce risk scores with better CAL and AUC than those that do not (e.g., PooledRd* has better test CAL/AUC than PooledRd since DCD polishing can only reduce the loss).

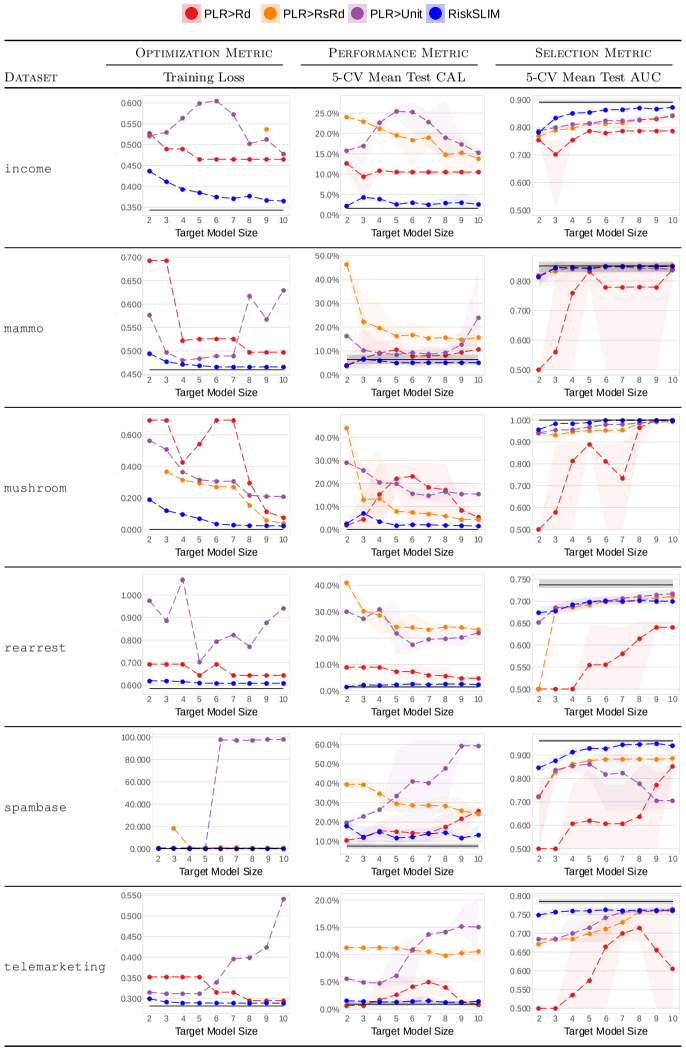


Figure 11: Summary statistics for risk scores built using RiskSLIM and traditional approaches. Each point represents the best risk score with integer coefficients $\lambda_j \in \{-5, \ldots, 5\}$ and model size $\|\boldsymbol{\lambda}\|_0 \leq R^{\max}$ for $R^{\max} \in \{2, \ldots, 10\}$. We show the variation in 5-CV mean test CAL and AUC for each method by shading the range between the 5-CV minimum and maximum. The black line in each plot is a baseline, which shows the performance of a single PLR model with real-valued coefficients and no model size constraint.

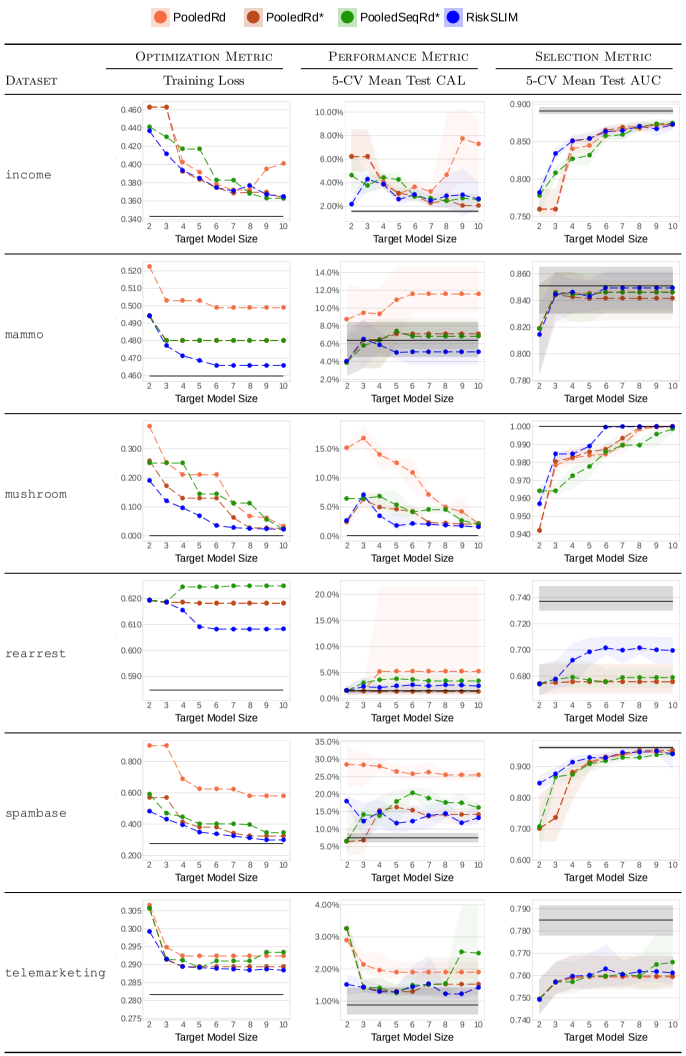


Figure 12: Summary statistics for risk scores built using RiskSLIM and pooled approaches. Each point represents the best risk score with integer coefficients $\lambda_j \in \{-5, \ldots, 5\}$ and model size $\|\boldsymbol{\lambda}\|_0 \leq R^{\max}$ for $R^{\max} \in \{2, \ldots, 10\}$. We show the variation in 5-CV mean test CAL and AUC for each method by shading the range between the 5-CV minimum and maximum. The black line in each plot is a baseline, which shows the performance of a single PLR model with real-valued coefficients and no model size constraint.

		Traditional Approaches			Pooled Approaches			
Dataset	**Metric**	PLR≫Rd	PLR≫RsRd	PLR≫Unit	PooledRd	PooledRd*	PooledSeqRd*	RiskSLIM
	test cal							2.6%
income	**test auc**	10.5%	19.5%	25.4%	3.0%	3.1%	4.2%	0.854
$n = 32561$	**loss value**	0.787	0.813	0.814	0.845	0.854	0.832	0.385
$d = 36$	**model size**	0.465	0.777	0.599	0.392	0.383	0.417	5
	opt. gap	2	3	5	5	5	4	9.7%
		-	-	-	-	-	-	
	test cal							5.0%
mammo	**test auc**	10.5%	16.2%	8.5%	10.9%	7.1%	7.4%	0.843
$n = 961$	**loss value**	0.832	0.846	0.842	0.845	0.841	0.845	0.469
$d = 14$	**model size**	0.526	0.745	0.484	0.503	0.480	0.480	5
	opt. gap	3	5	5	3	3	3	0.0%
		-	-	-	-	-	-	
	test cal							
mushroom	**test auc**	22.1%	8.0%	19.9%	12.6%	4.6%	5.4%	1.8%
$n = 8124$	**loss value**	0.890	0.951	0.969	0.984	0.986	0.978	0.989
$d = 113$	**model size**	0.543	0.293	0.314	0.211	0.130	0.144	0.069
	opt. gap	1	2	5	4	4	5	5
		-	-	-	-	-	-	0.0%
	test cal							
rearrest	**test auc**	7.3%	24.2%	21.8%	5.2%	1.4%	3.8%	2.4%
$n = 22530$	**loss value**	0.555	0.692	0.698	0.676	0.676	0.677	0.699
$d = 48$	**model size**	0.643	1.437	0.703	0.618	0.618	0.624	0.609
	opt. gap	1	5	5	4	4	4	5
		-	-	-	-	-	-	3.9%
	test cal							11.7%
spambase	**test auc**	15.0%	29.5%	33.4%	26.5%	16.3%	17.9%	0.928
$n = 4601$	**loss value**	0.620	0.875	0.861	0.910	0.913	0.908	0.349
$d = 57$	**model size**	0.666	1.090	0.515	0.624	0.381	0.402	5
	opt. gap	1	4	5	5	5	5	27.8%
		-	-	-	-	-	-	
	test cal							1.3%
telemarketing	**test auc**	2.6%	11.2%	6.2%	1.9%	1.3%	1.3%	0.760
$n = 41188$	**loss value**	0.574	0.700	0.715	0.759	0.760	0.760	0.289
$d = 57$	**model size**	0.352	11.923	0.312	0.292	0.289	0.289	5
	opt. gap	0	3	3	4	5	5	3.5%
		-	-	-	-	-	-	

Table 3: Summary statistics for risk scores with integer coefficients $\lambda_j \in \{-5, \ldots, 5\}$ for a model size constraint $\|\boldsymbol{\lambda}\|_0 \leq 5$. Here: *test cal* is the 5-CV mean test CAL; *test auc* is the 5-CV mean test AUC; *model size* and *loss value* pertain to a final model trained using the entire dataset. For each dataset, we highlight the method that attains the best test cal, auc, and loss value in green. We also highlight methods that produce trivial models in red.

On the Caveats of CAL The PLR≫Rd risk score for `telemarketing` in Table 3 highlights a key shortcoming of CAL that illustrates why we report AUC: trivial and near-trivial models can have misleadingly low CAL. Here, PLR≫Rd rounds all coefficients other than the intercept to zero, producing a model that trivially assigns a constant score to all examples $s_i = \langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle = \lambda_0 = 2$. Since there is only one score, the predicted risk for all points is $p_i = 11.9\%$,and the observed risk is the proportion of positive examples $\bar{p} = \Pr(y_i = +1) = 11.3\%$. Thus, a trivial model has a training CAL of 0.7%, the lowest among all methods, which (misleadingly) suggests that it has the best performance on training data (see Table 7 in Appendix F for values of training CAL). In this case, one could instead determine that the model is trivial by its training AUC, which is 0.500. This result also shows why we choose free parameters that maximize the 5-CV mean test AUC rather

than CAL: choosing free parameters to minimize the 5-CV mean test CAL can result in a trivial model.

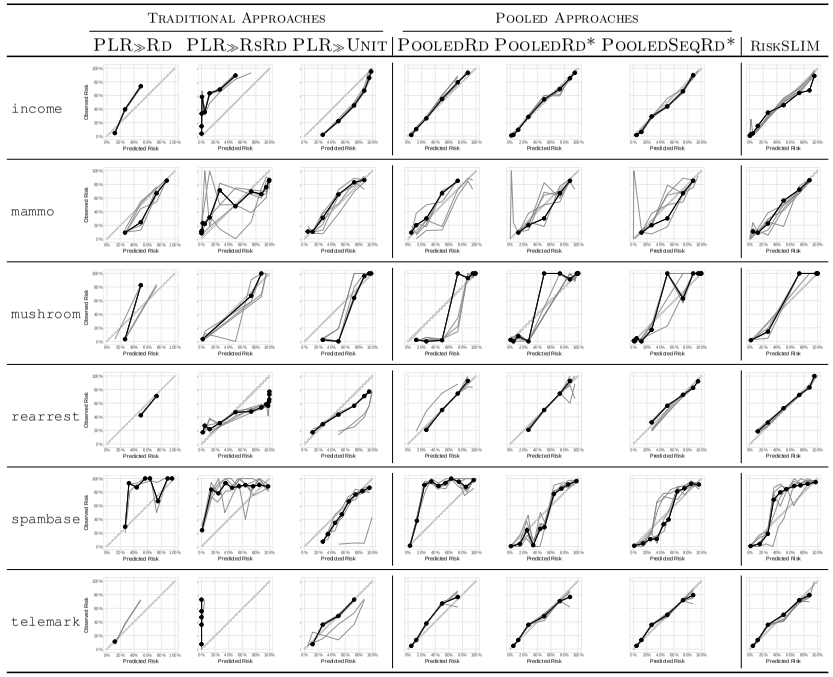


Figure 13: Reliability diagrams for risk scores with integer coefficients $\lambda_j \in \{-5, \ldots, 5\}$ for a model size constraint $\|\boldsymbol{\lambda}\|_0 \leq 5$. We plot results for models from each fold on test data in grey, and for the final model on training data in black.

On Calibration Issues of Risk Scores The reliability diagrams in Figure 13 highlight two issues with respect to the calibration of risk scores that are difficult to capture using a summary statistic:

- Monotonicity violations in observed risk. For example, the reliability diagrams for PLR≫Rd on `spambase`, or PooledRd and PooledSeqRd* on `mushroom` show that the observed risk does not increase monotonically with predicted risk. There is not a way to calibrate the risk scores to remove this, and it is non-intuitive to have an increase in risk score correspond to a decrease in actual risk.
- Irregular spacing and coverage of predicted risk. For example, the PLR≫Rd risk score for `income` outputs risk predictions that range between only 20% to 60%, and the PLR≫RsRd risk score for `rearrest` produces risk predictions that are clustered at end points.

The results in Figure 13 suggest that such issues can be mitigated by optimizing the logistic loss (see, e.g., the calibration of risk scores built using RiskSLIM, PooledRd*, PooledSeqRd* where integer coefficients are determined by directly optimizing the logistic loss). In contrast, these issues are difficult to address by post-processing. Consider, for example, using Platt scaling (Platt, 1999) to improve the calibration of the PLR≫Rd and PLR≫RsRd risk scores in Figure 13. As shown in Figure 14, Platt scaling improves calibration by centering and spreading risk estimates over the reliability diagram. However, it does not resolve issues that were introduced by earlier heuristics, such as monotonicity violations, a lack of coverage in risk predictions, or low AUC.

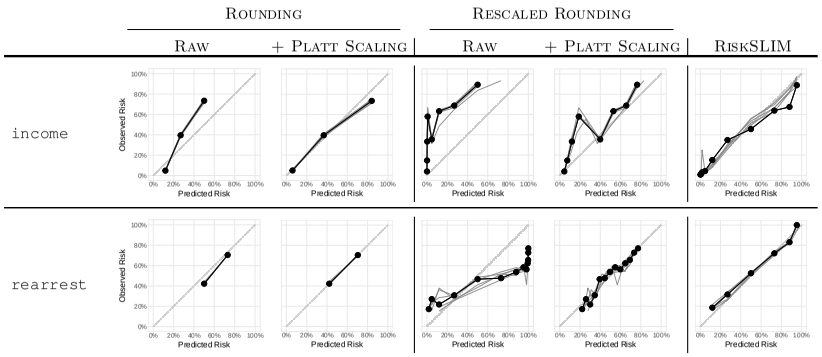


Figure 14: Reliability diagrams for PLR≫Rd and PLR≫RsRd risk scores with and without Platt scaling, and for RiskSLIM. Platt scaling improves calibration by centering and spreading risk predictions. However, it cannot overcome calibration issues introduced by rounding heuristics such a lack of decision points (e.g. PLR≫Rd on `income` and `rearrest`, left) or monotonicity violations (e.g, PLR≫RsRd on `income`, top middle).

On the Pitfalls of Traditional Approaches Our results in Figure 11 and Table 3 show that risk scores built using traditional approaches perform poorly in terms of CAL and AUC. In particular:

- Rounding (PLR≫Rd) produces risk scores with low AUC when it eliminates features from the model by rounding small coefficients $\lambda_j < |0.5|$ to zero.
- Rescaled rounding (PLR≫RsRd) hurts calibration since the logistic loss is not scale-invariant (see e.g., the reliability diagram for `rearrest` PLR≫RsRd in Figure 13).
- Unit weighting (PLR≫Unit) results in poor calibration and unpredictable behavior (e.g., risk scores with more features can perform worse as seen in PLR≫Unit models for `income` and `telemarketing` in Figure 11).

The effects of rescaled rounding and unit weighting on calibration are reflected by highly suboptimal values of the loss in Table 3 and Figure 11. These issues are often overlooked, perhaps because their effect on AUC is far less severe (e.g. the rescaling is recommended by U.S. Department of Justice, 2005; Pennsylvania Commission on Sentencing, 2012).

Our baseline methods may not match the exact methods used in practice as they do not reflect the significant human input used in risk score development (e.g., domain experts perform preliminary feature selection, round coefficients, or choose a scaling factor before rounding, manually or without validation, as shown in Appendix C). Nevertheless, these results highlight two major pitfalls of traditional approaches, namely:

- Traditional approaches heuristically post-process a single model. This means that they fail whenever a heuristic dramatically changes CAL or AUC.
- Traditional approaches use heuristics that are oblivious to the value of the loss function, which tends to result in poor calibration.

On Pooled Approaches Our results suggest that risk scores built using our pooled approaches attain considerably better calibration and rank accuracy than those built using traditional approaches. These methods aim to overcome the pitfalls of traditional approaches using two strategies:

- *Pooling*, which generates a pool of PLR models, post-processes each model to produce a pool of risk scores, and selects the best risk score within the pool. Pooling provides some robustness against failure modes of heuristics that dramatically alter performance (e.g., for rounding, it is very unlikely that the coefficients of all models in the pool will be rounded to zero). The performance gain due to pooling can be seen by comparing the results for PLR$\gg$Rd to PooledRd in Table 3.
- *Loss-Sensitive Heuristics*, such as SequentialRounding and DCD, which produce a pool of risk scores that attain lower values of the loss, and thereby let us select a risk score with better CAL and AUC. The performance gain due to loss-sensitive heuristics can be seen by comparing the results for PooledRd to PooledRd* in Table 3.

The fact that RiskSLIM risk scores have lower loss compared to pooled methods shows that direct optimization can efficiently find solutions that may not be found by exhaustive post-processing (e.g., where we fit all possible ℓ_1 and ℓ_2 penalized logistic regression models, and convert them to risk scores with specially-designed heuristics). Here, we have shown that exhaustive post-processing strategies can often produce risk scores that perform well. In Section 6, however, we will see that the performance gap can be significant in the presence of non-trivial constraints.

On Computation Although the risk score problem is NP-hard, we trained RiskSLIM models that were certifiably optimal or had small optimality gaps for all datasets in under 20 minutes using an LCPA implementation with the improvements in Section 4. Even when LCPA did not recover a certifiably optimal solution, it produced a risk score that performed well and did not exhibit the calibration issues of models built heuristically.

In general, the time spent computing cutting planes is a small portion of the overall runtime for LCPA ($< 1\%$, for all datasets). Given that LCPA scales linearly with sample size, we expect to obtain similar results even if the datasets had far more samples. Factors that affected the time to obtain a certifiably optimal solution include:

- *Highly Correlated Features*: Subsets of redundant features produce multiple optima, which increases the size of the B&B tree.

- *Feature Encoding*: In particular, the problem is harder when the dataset includes real-valued variables, like those in the `spambase` dataset.
- *Difficulty of the Learning Problem*: On separable problems such as `mushroom`, it is easy to recover a certifiably optimal solution since many solutions perform well and produce a near-optimal lower bound.

1.	Prior Arrests ≥ 2	1 point		...
2.	Prior Arrests ≥ 5	1 point	+	...
3.	Prior Arrests for Local Ordinance	1 point	+	...
4.	Age at Release between 18 to 24	1 point	+	...
5.	Age at Release ≥ 40	-1 point	+	
		SCORE	=	

SCORE	-1	0	1	2	3	4
RISK	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

Figure 15: RiskSLIM model for `rearrest`. RISK is the predicted probability that a prisoner is arrested within 3 years of release from prison. This model has a 5-CV mean test CAL/AUC of 1.7%/0.697 and training CAL/AUC of 2.6%/0.701.

1.	Married	3 points		...
2.	Reported Capital Gains	2 points	+	...
3.	Age 22 to 29	-1 point	+	...
4.	Highest Level of Education is High School Diploma	-2 points	+	...
5.	No High School Diploma	-3 points	+	
		SCORE	=	

SCORE	-4 to -1	0	1	2	3	4	5
RISK	$< 5.0\%$	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

Figure 16: RiskSLIM model for `income`. RISK is the predicted probability that a US resident earns over \$50 000. This model has a 5-CV mean test CAL/AUC of 2.4%/0.854 and training CAL/AUC of 4.1%/0.860.

1.	Call between January and March	1 point		...
2.	Called Previously	1 point	+	...
3.	Previous Call was Successful	1 point	+	...
4.	Employment Indicator < 5100	1 point	+	...
5.	3 Month Euribor Rate ≥ 100	-1 point	+	
		SCORE	=	

SCORE	-1	0	1	2	3	4
RISK	4.7%	11.9%	26.9%	50.0%	73.1%	88.1%

Figure 17: RiskSLIM model for `telemarketing`. RISK is the predicted probability that a client opens a new bank account after a marketing call. This model has a 5-CV mean test CAL/AUC of 1.3%/0.760 and a training CAL/AUC of 1.1%/0.760.

1.	odor = foul	5 points		...
2.	gill size = broad	-3 points	+	...
3.	odor = almond	-5 points	+	...
4.	odor = anise	-5 points	+	...
5.	odor = none	-5 points	+	
		SCORE	=	

SCORE	-8	-5	-3	2 to 5
RISK	1.8%	26.9%	73.1%	> 95.0%

Figure 18: RiskSLIM model for `mushroom`. RISK is the predicted probability that a mushroom is poisonous. This model has a 5-CV mean test CAL/AUC of 1.8%/0.989 and a training CAL/AUC of 1.0%/0.990.

1.	IrregularShape	1 point		...
2.	Age $\geq$ 60	1 point	+	...
3.	OvalShape	-1 point	+	...
4.	ObscuredMargin	-1 point	+	...
5.	CircumscribedMargin	-2 points	+	
		SCORE	=	

SCORE	-3	-2	-1	0	1	2
RISK	4.7%	11.9%	26.9%	50.0%	73.1%	88.1%

Figure 19: RiskSLIM model for `mammo`. RISK is the predicted probability that a mammogram pertains to a patient with breast cancer. This model has a 5-CV mean test CAL/AUC of 5.0%/0.843 and a training CAL/AUC of 3.1%/0.849.

1.	CharacterFrequency_DollarSign	$\times$ 5 points		...
2.	WordFrequency_Remove	$\times$ 4 points	+	...
3.	WordFrequency_Free	$\times$ 2 points	+	...
4.	WordFrequency_HP	$\times$ -2 points	+	...
5.	WordFrequency_George	$\times$ -5 points	+	
		SCORE	=	

SCORE	$\leq$ -1.2	-1.2 to -0.4	-0.4 to 0.2	0.2 to 0.6	0.6 to 1.0	1.0 to 1.4	1.4 to 1.8	1.9 to 2.4	2.4 to 3.2	$\geq$ 3.2
RISK	1.4%	14.8%	26.8%	35.3%	45.1%	54.1%	65.3%	74.7%	85.6%	97.2%

Figure 20: RiskSLIM model for `spambase`. RISK is the predicted probability that an e-mail is spam. This model has a 5-CV mean test CAL/AUC of 11.7%/0.928 and a training CAL/AUC of 12.3%/0.935.

6. ICU Seizure Prediction

In this section, we describe a collaboration with the Massachusetts General Hospital and the University of Wisconsin Hospital where we built a customized risk score for ICU seizure prediction (Struck et al., 2017). Our goal is to discuss practical aspects of our method on a real-world problem with non-trivial constraints.

6.1. Problem Description

Patients who suffer from traumatic brain injury (e.g., due to a ruptured brain aneurysm) often experience seizures that may lead to irreversible brain damage. Since these seizures are not outwardly visible, patients who are brought into an intensive care unit are monitored via *continuous electroencephalography* (cEEG). Based on current clinical standards, neurologists are trained to recognize a large set of patterns in cEEG output (see e.g., Figure 21 and Hirsch et al., 2013). They consider the presence and characteristics of cEEG patterns along with other medical information to evaluate a patient's risk of seizure. These risk estimates are used to decide if a patient can be dismissed from the ICU, kept for further monitoring, or prescribed a medical intervention to avert potential brain injury. In practice, hospitals have a limited number of cEEG monitors that may be poorly allocated among patients in the ICU. Given reliable estimates of seizure risk, patients with a low risk of seizure can be taken off monitoring to free up monitors for new patients.

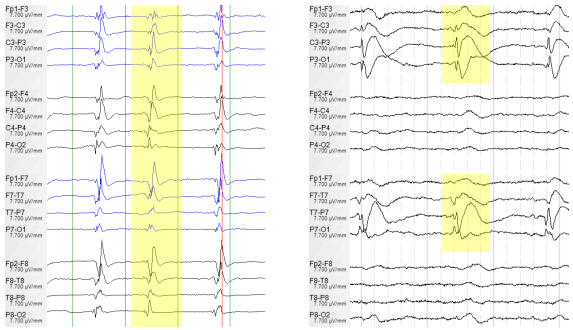


Figure 21: cEEG displays electrical activity at 16 standardized locations in a patient's brain using electrodes placed on the scalp. We show two cEEG patterns: a Generalized Periodic Discharge (GPD), which occurs on both sides of the brain (left); and a Lateralized Periodic Discharge (LPD), which occurs on one side of the brain (right). These figures were reproduced from a presentation at a training module by the American Clinical Neurophysiology Society (2012).

Data Our dataset was derived from cEEG recordings from 41 hospitals, curated by the Critical Care EEG Monitoring Research Consortium. It contains $n = 5,427$ recordings and $d = 87$ input variables (see Appendix G for a list). The outcome is defined as $y_i = +1$ if patient i who has been in the ICU for the past 24 hours will have a seizure in the next 24 hours. There is significant class imbalance as $\Pr(y_i = +1) = 12.5\%$. The input variables include information on patient medical history, secondary neurological symptoms,

and the presence and characteristics of 5 standard cEEG patterns: *Lateralized Periodic Discharges* (LPD); *Lateralized Rhythmic Delta* (LRDA); *Generalized Periodic Discharges* (GPD); *Generalized Rhythmic Delta* (GRDA); and *Bilateral Periodic Discharges* (BiPD).

Model Requirements Our collaborators wanted a risk score to help them reliably predict seizure risk by checking the presence and characteristics of cEEG patterns. It was critical for the model to output calibrated risk predictions since physicians would use the predicted risk to choose between multiple treatment options (i.e., patients may be prescribed different medication based on their predicted risk; see also Van Calster and Vickers, 2015; Shah et al., 2018, for a discussion on how miscalibrated risk predictions can lead to harmful decisions). To be adopted by physicians, it was also important to build a model that could be validated by domain experts and that was aligned with domain expertise.

In Figure 22, we present a RiskSLIM risk score for this problem that satisfies all of these requirements (see Struck et al., 2017, for details on model development). This risk score outputs calibrated risk estimates at several operating points. It obeys a model size constraint ($\|\boldsymbol{\lambda}\|_0 \leq 6$) to let physicians easily validate the model, and monotonicity constraints to ensure that the signs of some coefficients are aligned with domain knowledge.

1.	Any cEEG Pattern with Frequency > **2 Hz**	1 point		...
2.	**E**pileptiform Discharges	1 point	+	...
3.	Patterns include **L**PD or LRDA or BIPD	1 point	+	...
4.	**P**atterns Superimposed with Fast, or Sharp Activity	1 point	+	...
5.	Prior **S**eizures	1 point	+	...
6.	**B**rief Rhythmic Discharges	2 points	+	
		SCORE	=	

SCORE	0	1	2	3	4	5	6+
RISK	<5%	12%	27%	50%	73%	88 %	>95%

Figure 22: 2HELPS2B risk score built by RiskSLIM (see Struck et al., 2017, for details). This model has a 5-CV mean test CAL/AUC of 2.7%/0.819.

As a follow up to our work in Struck et al. (2017), our collaborators wanted to see if we could improve the usability of the risk score in Figure 22 without sacrificing too much calibration or rank accuracy. Seeing how the risk score in Figure 22 includes features that could require a physician to check a large number of patterns (e.g., *MaxFrequencyOfAnyPattern*≥ 2 *Hz* or *PatternsIncludeLPD or LRDA or BIPD*), our collaborators sought to improve usability by specifying *operational constraints* on feature composition and feature encoding. In turn, our goal was to produce the best risk score that satisfied these constraints, so that our collaborators could correctly evaluate the loss in predictive performance due to their requirements, and make an informed choice between competing models. We present a complete list of their constraints in Appendix G. They can be grouped as follows:

- *Limited Model Size*: The model had to use at most 4 input variables, so that it would be easy to validate and use in an ICU.
- *Monotonicity*: The model had to obey monotonicity constraints for well-known risk factors for seizures (e.g., it could not suggest that having prior seizures lowers seizure risk).

- *No Redundancy Between Categorical Variables*: The model had to include variables that were linearly independent (e.g., it could include *Male* or *Female* but not both).
- *Specific cEEG Patterns or Any cEEG Pattern*: The dataset had variables for specific cEEG patterns (e.g., *MaxFrequencyLPD*) and variables for any cEEG pattern (e.g., *MaxFrequencyAnyPattern*). The model had to use variables for specific patterns or any pattern, but not both.
- *Frequency in Continuous or Thresholded Form*: The dataset had two kinds of variables related to the frequency of a cEEG pattern: (i) a real-valued variable (e.g., *MaxFrequencyLPD* $\in [0, 3.0]$); and (ii) 7 binary threshold variables (e.g., *MaxFrequencyLPD* ≤ 0.5 *Hz*). Models had to use the real-valued variable or the binary variables, not both.
- *Limited # of Thresholds for Thresholded Encoding*: To prevent clinicians from having to check multiple thresholds, the model could include at most 2 binary threshold variables for a given cEEG pattern.

Training Setup We used the training setup in Section 5.1, which we adapted to address constraints as follows. We trained a RiskSLIM model by solving a customized instance of RiskSlimMINLP with 20 additional constraints and 2 additional variables, which we solved to optimality in $\leq$ 20 minutes. The baseline methods had built-in mechanisms to handle monotonicity constraints, but required tuning to handle other constraints. For each method, we trained a final model using all of the training data for the instance of the free parameters that obeyed all constraints and maximized the mean 5-CV test AUC.

6.2. Discussion

On Performance and Usability in a Constrained Setting The results in Table 4 show the potential performance benefits of training an optimized risk score for problems with non-trivial constraints. Here, RiskSLIM has a 5-CV mean test CAL/AUC of 2.5%/0.801 while the best risk score built using a heuristic method has a 5-CV mean test CAL/AUC of 2.8%/0.745

In contrast to the experiments in Section 5, only RiskSLIM and our pooled methods were able to find a feasible risk score under the constraints. Traditional methods (e.g., PLR$\gg$Rd, PLR$\gg$RsRd, and PLR$\gg$Unit) violate one or more constraints after rounding. In fact, these methods cannot produce a risk score with comparable performance to the RiskSLIM risk score even when these constraints are relaxed. If we consider only simple constraints on model size and monotonicity, then risk scores produced by these methods have a test AUC of at most 0.761 (PLR$\gg$RsRd) and a test CAL of at least 7.0% (PLR$\gg$Rd).

As shown in Figures 23 to 26, risk scores with similar test CAL can still exhibit important differences in terms of calibration. Here, the risk predictions of the RiskSLIM model are monotonic and within the boundaries of the risk predictions of the fold-based models. In comparison, the risk predictions of other models do not monotonically increase with observed risk and vary significantly across test folds (e.g., PooledSeqRd*). As noted by our collaborators, such issues affect model adoption: the monotonicity violations of the PooledSeqRd model suggest that patients with a score of 3.5 may have more seizures compared to patients with a score of 4.0, eroding trust in the model's risk predictions.

Method	Constraints Violated	Test CAL	Test AUC	Model Size	Loss Value	Optimality Gap	Train CAL	Train AUC
RiskSLIM	–	2.5% 1.9 - 3.4%	0.801 0.758 - 0.841	4 4 - 4	0.293	0.0%	2.0%	0.806
PooledRd	–	5.3% 3.1 - 7.1%	0.740 0.712 - 0.757	2 1 - 3	0.350	–	6.0%	0.752
PooledRd*	–	3.0% 1.4 - 3.6%	0.745 0.712 - 0.776	2 1 - 3	0.308	–	1.9%	0.754
PooledSeqRd*	–	2.8% 2.4 - 3.1%	0.745 0.713 - 0.805	3 2 - 4	0.313	–	1.9%	0.767
PLR	All	2.6% 1.7 - 3.6%	0.844 0.829 - 0.869	29 20 - 35	0.272	–	2.0%	0.850
PLR	Integrality Operational	4.4% 3.3 - 6.5%	0.742 0.712 - 0.774	4 3 - 4	0.325	–	3.9%	0.771
PLR≫Rd	Operational	7.0% 5.7 - 9.2%	0.743 0.705 - 0.786	2 2 - 3	0.329	–	7.0%	0.735
PLR≫RsRd	Operational	12.4% 11.2 - 13.6%	0.761 0.733 - 0.815	4 4 - 4	2.109	–	12.5%	0.760
PLR≫Unit	Operational	24.6% 23.6 - 25.7%	0.759 0.732 - 0.813	4 4 - 4	0.520	–	24.8%	0.759

Table 4: Performance of risk scores for seizure prediction, and feasibility with respect to constraints. We report the 5-CV mean test CAL and 5-CV mean test AUC. The ranges in each cell represent the 5-CV minimum and maximum. We present the risk scores built using each method in Figures 23 to 26.

Although the risk scores in Figures 23 to 26 obey all of the constraints specified by our collaborators, they exhibit differences in terms of usability. The RiskSLIM model requires physicians to check cEEG output for a single cEEG pattern (LPD). In comparison, other risk scores include the feature *PatternsInclude BiPD or LRDA or LPD*, which can require physicians to check cEEG output for 3 patterns in the worst case. The PooledSeqRd* risk score also uses *MaxFrequencyLPD*, which requires estimating the frequency of LPD and thereby requires more time.

On the Value of the Optimality Gap in Practice The results in Table 4 illustrate how heuristics may lead practitioners to overestimate the true impact of real-world constraints on model performance. Here: three traditional methods could not output a feasible risk score; six pooled methods produced feasible risk scores with suboptimal AUC and calibration issues; and a baseline PLR model with real-valued coefficients has an AUC of 0.742. Based on these results, a practitioner might inadvertently conclude that no feasible risk score could achieve a test AUC of 0.801.

In contrast, RiskSLIM models are paired with an optimality gap. In practice, a small optimality gap suggests that we have trained the best possible risk score that satisfies a specific set of constraints. Thus, if a risk score with a small optimality gap performs poorly on training data, and the model generalizes (e.g., its training performance is similar to its

K-CV performance), then one can attribute the performance deficit of the model to overly restrictive constraints and improve performance by relaxing them.

This provides a general mechanism to evaluate the effect of constraints on performance. If, for example, that our collaborators were not satisfied with the performance or usability of our model, then we could train certifiably optimal risk scores for different sets of constraints. By comparing the performance of certifiably optimal models, our collaborators could evaluate the true impact of their requirements on predictive performance, and navigate trade-offs in an informed manner. This approach helped our collaborators decide between a model with 4 features or 5 features. Here, we trained a RiskSLIM risk score with 5 features, which has a 5-CV test CAL/AUC of 3.4%/0.816. However, the slight improvement in test AUC did not outweigh the fact that the larger model included the feature $MaxFreqFactorAnyPattern \geq 2$ which could increase the model evaluation time by doctors.

On the Challenges of Handling Operational Constraints One of the practical benefits of our method is that it can address constraints without parameter tuning or post-processing. Since our method can directly incorporate constraints into the MINLP formulation, all RiskSLIM models are feasible. Thus we can produce a feasible risk score and estimate its predictive performance by training 6 models: 1 final model fit on the full dataset for deployment, and 5 models trained on subsets of the training data to produce an unbiased performance estimate for the algorithm via 5-fold CV.

In contrast, the pooled methods produce a feasible model by post-processing a large pool of models and discarding those that are infeasible. Since we must then choose between feasible models on the basis of 5-CV performance, we must use a nested CV setup to pair any model with an unbiased performance estimate. This requires fitting a total of 33,000 models.[2] In general settings, there is no guarantee that pooled methods will produce a feasible risk score. In this case, for instance, only 12% of the instances for the pooled methods satisfied all constraints (see Table 9 in Appendix G). This kind of massive testing can become computationally burdensome.

Our results highlight other issues with methods that aim to address constraints by parameter tuning. Let us say we would use a standard K-fold CV setup to tune parameters. In this case, we would train models on K validation folds for each instance of the free parameters, choose the instance of the free parameters that maximizes the mean K-CV test AUC without violating any constraints, and then train a "final model" for this instance. Unfortunately, there is no guarantee that the final model will obey all constraints.

On the Benefits of Risk Scores with Small Integer Coefficients Figures 23 to 26 illustrate some of the practical benefits of risk scores with small integer coefficients. When input variables belong to a small discrete set, the scores also belong to a small discrete set. This reduces the number of operating points on the ROC curve and reliability diagram, which makes it easier to pick an operating point. Further, when input variables are binary, the decision rules at each operating point can be represented as a Boolean rule. For the RiskSLIM model in Figure 23, for example, the decision rule:

$$\hat{y}_i = +1 \text{ if score} \geq 2$$

2. A nested CV setup with 5 outer folds, 5 inner folds, and 1,100 free parameter instances requires fitting $1,100 \times 5 \times (5+1) = 33,000$ models.

is equivalent to the Boolean function:

$$\hat{y}_i = +1 \text{ if } \mathit{BriefRhythmicDischarge} \\ \mathit{OR}\ \mathit{PatternsIncludeLPD} \\ \mathit{OR}\ (\mathit{PriorSeizure}\ \mathit{AND}\ \mathit{EpiletiformDischarge})$$

Small integer coefficients let users extract such rules by listing conditions when the score exceeds the threshold. This is more challenging when a model uses real-valued coefficients, as shown by the score function of the PLR model from Table 4:

$$\text{score} = -2.35 + 0.91 \cdot \mathit{PatternsIncludeBiPD\ or\ LRDA\ or\ LPD} + 0.03 \cdot \mathit{PriorSeizure} \\ + 0.61 \cdot \mathit{MaxFrequencyLPD}.$$

In this case, extracting a Boolean function is difficult as computing the score involves arithmetic with real-valued coefficients and the real-valued variable $\mathit{MaxFrequencyLPD}$.

1.	BriefRhythmicDischarge	2 points		...
2.	PatternsInclude LPD	2 points	+	...
3.	PriorSeizure	1 point	+	...
4.	EpiletiformDischarge	1 point	+	
		SCORE	=	

SCORE	0	1	2	3	4	5	6
RISK	4.7%	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

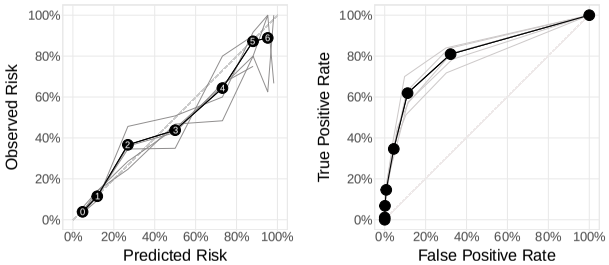


Figure 23: RiskSLIM risk score (top), reliability diagram (bottom left), and ROC curve (bottom right) for the `seizure` dataset. We plot results for the final model on training data in black, and the 5 fold models on test data in grey. This model has a 5-CV mean test CAL/AUC of 2.5%/0.801.

1.	BriefRhythmicDischarge	1 point		...
2.	PatternsIncludeBiPD or LRDA or LPD	1 point	+	
		SCORE	=	

SCORE	0	1	2
RISK	4.7%	11.9%	26.9%

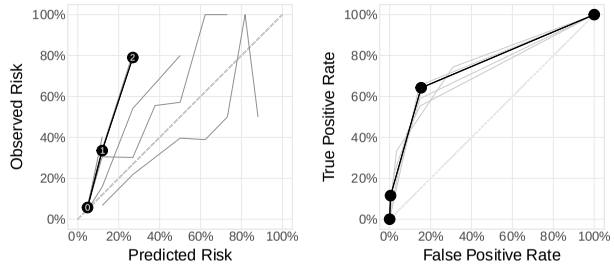


Figure 24: PooledRd risk score (top), reliability diagram (bottom left), and ROC curve (bottom right) for the `seizure` dataset. We plot results for the final model on training data in black, and results for the fold models on test data in grey. This model has a 5-CV mean test CAL/AUC of 5.3/0.740%.

1.	BriefRhythmicDischarge	3 points		...
2.	PatternsIncludeBiPD or LRDA or LPD	2 points	+	
		SCORE	=	

SCORE	0	2	3	5
RISK	4.7%	26.9%	50.0%	88.1%

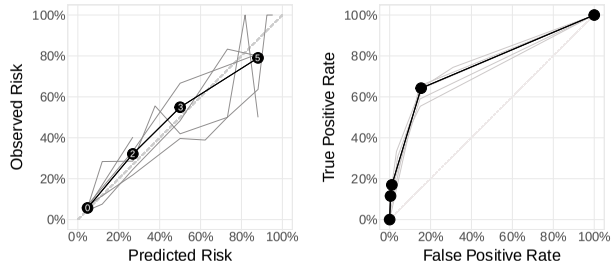


Figure 25: PooledRd* risk score (top), reliability diagram (bottom left), and ROC curve (bottom right) for the `seizure` dataset. We plot results for the final model on training data in black, and results for the fold models on test data in grey. This model has a 5-CV mean test CAL/AUC of 3.0/0.745%.

1.	PriorSeizure	1 point		...
2.	PatternsIncludeBiPD or LRDA or LPD	1 point	+	...
3.	MaxFreqFactorLPD	× 1 point per Hz	+	
		SCORE	=	

SCORE	0	1	2	2.5	3	3.5	4	4.5	5
RISK	4.7%	11.9%	26.9%	37.8%	50.0%	62.2%	73.1%	81.8%	88.1%

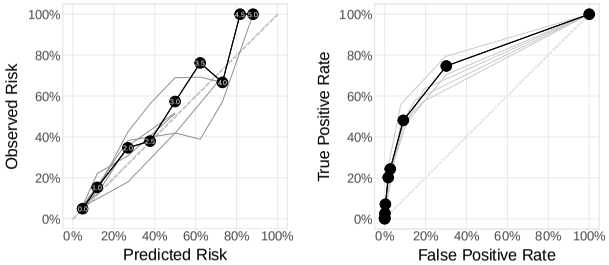


Figure 26: PooledSeqRd* model (top), reliability diagram (bottom left), and ROC curve (bottom right) for the `seizure` dataset. We plot results for fold models on test data in grey, and for the final model on training data in black. This model has a 5-CV mean test CAL/AUC of 2.8%/0.745.

7. Concluding Remarks

Risk scores are simple models that are often used to support important decisions in tasks such as mortality prediction and loan approval. Despite the fact that risk scores have been used in such tasks for nearly a century, many models are still developed *ad hoc* – i.e., by combining machine learning methods with heuristics (see e.g., the pipeline for the TIMI risk score in Antman et al., 2000, in Appendix C). Ad hoc approaches do not produce risk scores with performance guarantees, and may result in the deployment of risk score with poor calibration or rank accuracy. In practice, ad hoc approaches can produce risk scores that violate important requirements, leading practitioners to adjust risk scores manually, or to specify models without data (e.g., via a panel of domain experts as in Gage et al., 2001; McGinley and Pearse, 2012).

Our goal in this paper was to develop a machine learning method to build risk scores that would streamline model development. Our method – RiskSLIM – trains risk scores by solving an empirical risk minimization problem that performs exact feature selection, restricts coefficients to small integers, and enforces application-specific constraints. Since commercial solvers could not solve this problem reliably, we solved it with a new cutting plane algorithm – LCPA –which allows training to scale linearly with the number of samples, and can be used to solve other empirical risk minimization problems with non-convex regularizers or constraints. As shown in Sections 5 and 6, LCPA can train certifiably optimal risk scores for real-world datasets in minutes. These models attain best-in-class calibration and rank accuracy and avoid pitfalls of heuristics used in practice. Our method also simplifies model development, by allowing practitioners to customize risk scores without parameter tuning or post-processing, and by pairing models with a certificate of optimality that helps them understand how application-specific constraints affects performance.

Acknowledgments

We gratefully acknowledge support from Siemens, Phillips, and Wistron. We would like to thank Paul Rubin for helpful discussions. We would also like to thank our collaborators Aaron Struck and Brandon Westover for their guidance on the seizure prediction application.

References

Alba, Ana Carolina, Thomas Agoritsas, Michael Walsh, Steven Hanna, Alfonso Iorio, PJ Devereaux, Thomas McGinn, and Gordon Guyatt. Discrimination and calibration of clinical prediction models: Users' guides to the medical literature. *Journal of the American Medical Association*, 318 (14):1377–1384, 2017.

American Clinical Neurophysiology Society. Standardized Critical Care EEG Terminology Training Module, 2012.

Amodei, Dario, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Angelino, Elaine, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18(234): 1–78, 2018.

Antman, Elliott M, Marc Cohen, Peter JLM Bernink, Carolyn H McCabe, Thomas Horacek, Gary Papuchis, Branco Mautner, Ramon Corbalan, David Radley, and Eugene Braunwald. The TIMI risk score for unstable angina/non–ST elevation MI. *Journal of the American Medical Association*, 284(7):835–842, 2000.

Austin, James, Roger Ocker, and Avi Bhati. Kentucky Pretrial Risk Assessment Instrument Validation. *Bureau of Justice Statistics*, 2010.

Bache, K. and M. Lichman. UCI Machine Learning Repository, 2013.

Bai, Lihui and Paul A Rubin. Combinatorial Benders Cuts for the Minimum Tollbooth Problem. *Operations Research*, 57(6):1510–1522, 2009.

Bardenet, Rémi and Odalric-Ambrym Maillard. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, 2015.

Beneish, Messod D, Charles MC Lee, and D Craig Nichols. Earnings manipulation and expected returns. *Financial Analysts Journal*, 69(2):57–82, 2013.

Bertsimas, Dimitris, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

Billiet, Lieven, Sabine Van Huffel, and Vanya Van Belle. Interval coded scoring extensions for larger problems. In *Proceedings of the IEEE Symposium on Computers and Communications*, pages 198–203. IEEE, 2017.

Billiet, Lieven, Sabine Van Huffel, and Vanya Van Belle. Interval Coded Scoring: A toolbox for interpretable scoring systems. *PeerJ Computer Science*, 4:e150, 04 2018.

Bobko, Philip, Philip L Roth, and Maury A Buster. The usefulness of unit weights in creating composite scores. A literature review, application to content validity, and meta-analysis. *Organizational Research Methods*, 10(4):689–709, 2007.

Bonami, Pierre, Mustafa Kilinç, and Jeff Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In *Mixed Integer Nonlinear Programming*, pages 1–39. Springer, 2012.

Boyd, Stephen P and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Burgess, Ernest W. Factors determining success or failure on parole. *The workings of the indeterminate sentence law and the parole system in Illinois*, pages 221–234, 1928.

Byrd, Richard H, Jorge Nocedal, and Richard A Waltz. KNITRO: An Integrated Package for Nonlinear Optimization. In *Large-scale Nonlinear Optimization*, pages 35–59. Springer, 2006.

Calmon, Flavio, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pages 3995–4004, 2017.

Carrizosa, Emilio, Amaya Nogales-Gómez, and Dolores Romero Morales. Strongly agree or strongly disagree?: Rating features in support vector machines. *Information Sciences*, 329:256–273, 2016.

Caruana, Rich and Alexandru Niculescu-Mizil. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 69–78. ACM, 2004.

Caruana, Rich and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine Learning*, pages 161–168. ACM, 2006.

Caruana, Rich, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

Cawley, Gavin C and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107, 2010.

Chang, Allison, Cynthia Rudin, Michael Cavaretta, Robert Thomas, and Gloria Chou. How to reverse-engineer quality rankings. *Machine Learning*, 88:369–398, September 2012.

Chen, Chaofan and Cynthia Rudin. An optimization approach to learning falling rule lists. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 604–612. PMLR, 09–11 Apr 2018.

Chen, Chaofan, Kancheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. An interpretable model with globally consistent explanations for credit risk. In *Proceedings of NeurIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy*, 2018.

Chevaleyre, Yann, Frederic Koriche, and Jean-Daniel Zucker. Rounding methods for discrete linear classification. In *Proceedings of the 30th International Conference on Machine Learning*, pages 651–659, 2013.

Cranor, Lorrie Faith and Brian A LaMacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

Dawes, Robyn M. The robust beauty of improper linear models in decision making. *American Psychologist*, 34(7):571–582, 1979.

DeGroot, Morris H and Stephen E Fienberg. The comparison and evaluation of forecasters. *The Statistician*, pages 12–22, 1983.

Duwe, Grant and KiDeuk Kim. Sacrificing accuracy for transparency in recidivism risk assessment: The impact of classification method on predictive performance. *Corrections*, pages 1–22, 2016.

Einhorn, Hillel J and Robin M Hogarth. Unit weighting schemes for decision making. *Organizational Behavior and Human Performance*, 13(2):171–192, 1975.

Elter, M, R Schulz-Wendtland, and T Wittenberg. The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. *Medical Physics*, 34:4164, 2007.

Ertekin, Şeyda and Cynthia Rudin. On equivalence relationships between classification and ranking algorithms. *Journal of Machine Learning Research*, 12:2905–2929, 2011.

Feldman, Michael, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

FICO. Introduction to Scorecard for FICO Model Builder. http://www.fico.com/en/node/8140?file=7900, 2011.

Finlay, Steven. *Credit scoring, response modeling, and insurance rating: a practical guide to forecasting consumer behavior*. Palgrave Macmillan, 2012.

Franc, Vojtěch and Soeren Sonnenburg. Optimized cutting plane algorithm for support vector machines. In *Proceedings of the 25th International Conference on Machine Learning*, pages 320–327. ACM, 2008.

Franc, Vojtěch and Sören Sonnenburg. Optimized cutting plane algorithm for large-scale risk minimization. *Journal of Machine Learning Research*, 10:2157–2192, 2009.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

Gage, Brian F, Amy D Waterman, William Shannon, Michael Boechler, Michael W Rich, and Martha J Radford. Validation of clinical classification schemes for predicting stroke. *Journal of the American Medical Association*, 285(22):2864–2870, 2001.

Goel, Sharad, Justin M Rao, and Ravi Shroff. Precinct or Prejudice? Understanding Racial Disparities in New York City's Stop-and-Frisk Policy. *Annals of Applied Statistics*, 10(1):365–394, 2016.

Goh, Gabriel, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems*, pages 2415–2423, 2016.

Goh, Siong Thye and Cynthia Rudin. Box drawings for learning with imbalanced data. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 333–342. ACM, 2014.

Goldberg, Noam and Jonathan Eckstein. Sparse weighted voting classifier selection and its linear programming relaxations. *Information Processing Letters*, 112:481–486, 2012.

Gottfredson, Don M and Howard N Snyder. The mathematics of risk classification: Changing data into valid instruments for juvenile courts. NCJ 209158. *Office of Juvenile Justice and Delinquency Prevention Washington, D.C.*, 2005.

Guan, Wei, Alex Gray, and Sven Leyffer. Mixed-integer support vector machine. In *NIPS Workshop on Optimization for Machine Learning*, 2009.

Gupta, Maya, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. *Journal of Machine Learning Research*, 17(1):3790–3836, 2016.

Hirsch, LJ, SM LaRoche, N Gaspard, E Gerard, A Svoronos, ST Herman, R Mani, H Arif, N Jette, Y Minazad, et al. American clinical neurophysiology society's standardized critical care EEG terminology: 2012 version. *Journal of Clinical Neurophysiology*, 30(1):1–27, 2013.

Holte, Robert C. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.

Holte, Robert C. Elaboration on Two Points Raised in "Classifier Technology and the Illusion of Progress". *Statistical Science*, 21(1):24–26, February 2006.

Hu, Xiyang, Cynthia Rudin, and Margo Seltzer. Optimal sparse decision trees. In *Proc. Neural Information Processing Systems*, 2019.

ILOG, IBM. CPLEX Optimizer 12.6. https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, 2017.

Joachims, Thorsten. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International conference on Knowledge Discovery and Data Mining*, pages 217–226. ACM, 2006.

Joachims, Thorsten, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

Kamishima, Toshihiro, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

Kelley, James E, Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

Kessler, Ronald C, Lenard Adler, Minnie Ames, Olga Demler, Steve Faraone, EVA Hiripi, Mary J Howes, Robert Jin, Kristina Secnik, Thomas Spencer, et al. The World Health Organization Adult ADHD Self-Report Scale (ASRS): a short screening scale for use in the general population. *Psychological Medicine*, 35(02):245–256, 2005.

Kohavi, Ron. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 202–207. AAAI Press, 1996.

Kotlowski, Wojciech, Krzysztof J Dembczynski, and Eyke Huellermeier. Bipartite ranking through minimization of univariate loss. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1113–1120, 2011.

Kronqvist, Jan, David E Bernal, Andreas Lundell, and Ignacio E Grossmann. A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, 20(2):397–455, 2019.

Lakkaraju, Himabindu, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684. ACM, 2016.

Latessa, Edward, Paula Smith, Richard Lemke, Matthew Makarios, and Christopher Lowenkamp. Creation and validation of the Ohio risk assessment system: Final report, 2009.

Le Gall, Jean-Roger, Stanley Lemeshow, and Fabienne Saulnier. A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study. *Journal of the American Medical Association*, 270(24):2957–2963, 1993.

Letham, Benjamin, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 9(3):1350–1371, 2015.

Li, Oscar, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *32nd AAAI Conference on Artificial Intelligence*, 2018.

Liu, Yufeng and Yichao Wu. Variable selection via a combination of the L0 and L1 penalties. *Journal of Computational and Graphical Statistics*, 16(4), 2007.

Lubin, Miles, Emre Yamangil, Russell Bent, and Juan Pablo Vielma. Polyhedral approximation in mixed-integer convex optimization. *Mathematical Programming*, 172(1-2):139–168, 2018.

Malioutov, Dmitry and Kush Varshney. Exact rule learning via boolean compressed sensing. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 765–773. PMLR, 17–19 Jun 2013.

Mangasarian, Olvi L, W Nick Street, and William H Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, 1995.

McGinley, Ann and Rupert M Pearse. A National Early Warning Score for Acutely Ill Patients, 2012.

Menon, Aditya Krishna, Xiaoqian J Jiang, Shankar Vembu, Charles Elkan, and Lucila Ohno-Machado. Predicting accurate probabilities with a ranking loss. In *Proceedings of the International Conference on Machine Learning*, volume 2012, page 703, 2012.

Moreno, Rui P, Philipp GH Metnitz, Eduardo Almeida, Barbara Jordan, Peter Bauer, Ricardo Abizanda Campos, Gaetano Iapichino, David Edbrooke, Maurizia Capuzzo, and Jean-Roger Le Gall. SAPS 3 - From evaluation of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at ICU admission. *Intensive Care Medicine*, 31(10):1345–1355, 2005.

Moro, Sérgio, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Naeini, Mahdi Pakdaman, Gregory F Cooper, and Milos Hauskrecht. Binary classifier calibration: A bayesian non-parametric approach. In *Proc. SIAM Int Conf Data Mining (SDM)*, pages 208–216, 2015.

Naoum-Sawaya, Joe and Samir Elhedhli. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210(1):33–55, November 2010.

Nguyen, Hai Thanh and Katrin Franke. A general Lp-norm support vector machine via mixed 0-1 programming. In *Machine Learning and Data Mining in Pattern Recognition*, pages 40–49. Springer, 2012.

Park, Jaehyun and Stephen Boyd. A semidefinite programming method for integer convex quadratic minimization. *Optimization Letters*, 12(3):499–518, 2018.

Pennsylvania Bulletin. Sentence Risk Assessment Instrument, April 2017.

Pennsylvania Commission on Sentencing. Interim Report 4: Development of Risk Assessment Scale, June 2012.

Piotroski, Joseph D. Value investing: The use of historical financial statement information to separate winners from losers. *Journal of Accounting Research*, pages 1–41, 2000.

Platt, John C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

Pleiss, Geoff, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5684–5693, 2017.

Reid, Mark D and Robert C Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11:2387–2422, 2010.

Reilly, Brendan M and Arthur T Evans. Translating clinical research into clinical practice: Impact of using prediction rules to make decisions. *Annals of Internal Medicine*, 144(3):201–209, 2006.

Rudin, Cynthia. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, May 2019.

Rudin, Cynthia and Şeyda Ertekin. Learning customized and optimized lists of rules with mathematical programming. *Mathematical Programming C (Computation)*, 10:659–702, 2018.

Rudin, Cynthia and Berk Ustun. Optimized Scoring Systems: Toward Trust in Machine Learning for Healthcare and Criminal Justice. *INFORMS Journal on Applied Analytics*, 48:399–486, 2018. Special Issue: 2017 Daniel H. Wagner Prize for Excellence in Operations Research Practice.

Rudin, Cynthia and Yining Wang. Direct Learning to Rank And Rerank. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 775–783. PMLR, 09–11 Apr 2018.

Rudin, Cynthia, Caroline Wang, and Beau Coker. The *age* of secrecy and unfairness in recidivism prediction. *Harvard Data Science Review*, 2019. Forthcoming.

Sato, Toshiki, Yuichi Takano, Ryuhei Miyashiro, and Akiko Yoshise. Feature subset selection for logistic regression via mixed integer optimization. *Computational Optimization and Applications*, 64(3):865–880, July 2016.

Sato, Toshiki, Yuichi Takano, and Ryuhei Miyashiro. Piecewise-linear approximation for feature subset selection in a sequential logit model. *Journal of the Operations Research Society of Japan*, 60(1):1–14, March 2017.

Schlimmer, Jeffrey C. *Concept acquisition through representational adjustment*. PhD thesis, University of California, Irvine, 1987. AAI8724747.

Shah, Nilay D, Ewout W Steyerberg, and David M Kent. Big Data and Predictive Analytics: Recalibrating Expectations. *Journal of the American Medical Assocation*, 2018.

Siddiqi, Naeem. *Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards*. John Wiley & Sons, second edition, January 2017. ISBN 978-1-119-27915-0.

Six, A. J., B. E. Backus, and J. C. Kelder. Chest pain in the emergency room: value of the HEART score. *Netherlands Heart Journal*, 16(6):191–196, 2008.

Sokolovska, Nataliya, Yann Chevaleyre, Karine Clément, and Jean-Daniel Zucker. The fused lasso penalty for learning interpretable medical scoring systems. In *Proceedings of the International Joint Conference on Neural Networks*, pages 4504–4511. IEEE, May 2017.

Sokolovska, Nataliya, Yann Chevaleyre, and Jean-Daniel Zucker. A provable algorithm for learning interpretable scoring systems. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 566–574. PMLR, 09–11 Apr 2018.

Souillard-Mandar, William, Randall Davis, Cynthia Rudin, Rhoda Au, David J. Libon, Rodney Swenson, Catherine C. Price, Melissa Lamar, and Dana L. Penney. Learning classification models of cognitive conditions from subtle behaviors in the digital clock drawing test. *Machine Learning*, 102(3):393–441, 2016.

Struck, Aaron F., Berk Ustun, Andres Rodriguez Ruiz, Jong Woo Lee, Suzette M. LaRoche, Lawrence J. Hirsch, Emily J. Gilmore, Jan Vlachy, Hiba Arif Haider, Cynthia Rudin, and M. Brandon Westover. Association of an electroencephalography-based risk score with seizure probability in hospitalized patients. *JAMA Neurology*, 74(12):1419–1424, 12 2017.

Teo, Choon Hui, Alex Smola, SVN Vishwanathan, and Quoc Viet Le. A scalable modular convex solver for regularized risk minimization. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727–736. ACM, 2007.

Teo, Choon Hui, S Vishwanathan, Alex Smola, and Quoc V Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2009.

Than, Martin, Dylan Flaws, Sharon Sanders, Jenny Doust, Paul Glasziou, Jeffery Kline, Sally Aldous, Richard Troughton, Christopher Reid, and William A Parsonage. Development and validation of the Emergency Department Assessment of Chest pain Score and 2h accelerated diagnostic protocol. *Emergency Medicine Australasia*, 26(1):34–44, 2014.

U.S. Department of Justice. The Mathematics of Risk Classification: Changing Data into Valid Instruments for Juvenile Courts, 2005.

Ustun, Berk and Cynthia Rudin. Supersparse Linear Integer Models for Optimized Medical Scoring Systems. *Machine Learning*, 102(3):349–391, 2016.

Ustun, Berk and Cynthia Rudin. Optimized Risk Scores. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1125–1134. ACM, 2017.

Ustun, Berk, Stefano Tracà, and Cynthia Rudin. Supersparse Linear Integer Models for Predictive Scoring Systems. In *AAAI Late-Breaking Developments*, 2013.

Ustun, Berk, M.B. Westover, Cynthia Rudin, and Matt T. Bianchi. Clinical prediction models for sleep apnea: The importance of medical history over symptoms. *Journal of Clinical Sleep Medicine*, 12(2):161–168, 2016.

Ustun, Berk, Lenard A Adler, Cynthia Rudin, Stephen V Faraone, Thomas J Spencer, Patricia Berglund, Michael J Gruber, and Ronald C Kessler. The World Health Organization Adult Attention-Deficit / Hyperactivity Disorder Self-Report Screening Scale for DSM-5. *JAMA Psychiatry*, 74(5):520–526, 2017.

Ustun, Berk, Yang Liu, and David Parkes. Fairness without harm: Decoupled classifiers with preference guarantees. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6373–6382. PMLR, 09–15 Jun 2019.

Van Belle, Vanya, Patrick Neven, Vernon Harvey, Sabine Van Huffel, Johan AK Suykens, and Stephen Boyd. Risk group detection and survival function estimation for interval coded survival methods. *Neurocomputing*, 112:200–210, 2013.

Van Calster, Ben and Andrew J Vickers. Calibration of risk prediction models: impact on decision-analytic performance. *Medical Decision Making*, 35(2):162–169, 2015.

Verwer, Sicco and Yingqian Zhang. Learning optimal classification trees using a binary linear program formulation. In *33rd AAAI Conference on Artificial Intelligence*, 2019.

Wang, Hao, Berk Ustun, and Flavio Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6618–6627. PMLR, 09–15 Jun 2019.

Wang, Jiaxuan, Jeeheh Oh, Haozhu Wang, and Jenna Wiens. Learning credible models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2417–2426. ACM, 2018.

Wang, Tong, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A Bayesian Framework for Learning Rule Sets for Interpretable Classification. *Journal of Machine Learning Research*, 18(70):1–37, 2017.

Zafar, Muhammad Bilal, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P. Gummadi. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 962–970. PMLR, 20–22 Apr 2017.

Zeng, Jiaming, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

Appendix A. Omitted Proofs

Proof (Remark 3). *We first explain why* LCPA *attains the optimal objective value, and then justify the upper bounds on the number of cuts and number of nodes.*

Observe that LCPA *finds the optimal solution to* RiskSlimMINLP *through an exhaustive search over the feasible region $\mathcal{L}$. Thus,* LCPA *is bound to encounter the optimal solution, since it only discards a node $(v^t, \mathcal{R}^t)$ if: (i) the surrogate problem is infeasible over $\mathcal{R}^t$ (in which case* RiskSlimMINLP *is also infeasible over $\mathcal{R}^t$); or (ii) the surrogate problem has an objective value that exceeds than $V^{\max}$ (in which case, any integer feasible solution $\mathcal{R}^t$ is also suboptimal).*

The bound on the number of cuts follows from the fact that Algorithm 2 only adds cuts at integer feasible solutions, of which there are at most $|\mathcal{L}|$. The bound on the number of processed nodes represents a worst-case limit produced by bounding the depth of the branch-and-bound tree. To do this, we exploit the fact that the splitting rule SplitRegion *splits a partition into two mutually exclusive subsets by adding integer-valued bounds such $\lambda_j \geq \lceil \lambda_j \rceil$ and $\lambda_j \leq \lfloor \lambda_j \rfloor - 1$ on a single coefficient to the feasible solution. Consider applying* SplitRegion *a total of $\Lambda_j^{\max} - \Lambda_j^{\min} + 1$ times in succession on a fixed dimension j. This results in a total of $\Lambda_j^{\max} - \Lambda_j^{\min} + 1$ nodes where each node fixes the coefficient in dimension j to an integer value $\lambda_j \in \{\Lambda_j^{\min}, \ldots, \Lambda_j^{\max}\}$. Pick any node and repeat this process for coefficients in the remaining dimensions. The resulting B&B tree will have at most $2^{|\mathcal{L}|} - 1$ leaf nodes where $\boldsymbol{\lambda}$ is restricted to integer feasible solutions.* ■

Proof (Proposition 4). *Since the coefficient set $\mathcal{L}$ is bounded, the data are $(\boldsymbol{x}_i, y_i)_{i=1}^n$ discrete, and the normalized logistic loss function $l(\boldsymbol{\lambda})$ is continuous, it follows that the value of $l(\boldsymbol{\lambda})$ is bounded:*

$$l(\boldsymbol{\lambda}) \in [\min_{\boldsymbol{\lambda} \in \mathcal{L}} l(\boldsymbol{\lambda}), \max_{\boldsymbol{\lambda} \in \mathcal{L}}, l(\boldsymbol{\lambda})] \text{ for all } \boldsymbol{\lambda} \in \mathcal{L}.$$

Thus we need only to show that $L^{\min} \leq \min_{\boldsymbol{\lambda} \in \mathcal{L}} l(\boldsymbol{\lambda})$, and $L^{\max} \geq \max_{\boldsymbol{\lambda} \in \mathcal{L}} l(\boldsymbol{\lambda})$. For the lower bound, we observe that:

$$\begin{aligned}
\min_{\boldsymbol{\lambda} \in \mathcal{L}} l(\boldsymbol{\lambda}) &= \min_{\boldsymbol{\lambda} \in \mathcal{L}} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-\langle \boldsymbol{\lambda}, y_i \boldsymbol{x}_i \rangle)) \\
&= \min_{\boldsymbol{\lambda} \in \mathcal{L}} \frac{1}{n} \sum_{i:y_i=+1} \log(1 + \exp(-\langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle)) + \frac{1}{n} \sum_{i:y_i=-1} \log(1 + \exp(\langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle)) \\
&\geq \frac{1}{n} \sum_{i:y_i=+1} \min_{\boldsymbol{\lambda} \in \mathcal{L}} \log(1 + \exp(-\langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle)) + \frac{1}{n} \sum_{i:y_i=-1} \min_{\boldsymbol{\lambda} \in \mathcal{L}} \log(1 + \exp(\langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle)) \\
&= \frac{1}{n} \sum_{i:y_i=+1} \log(1 + \exp(-\max_{\boldsymbol{\lambda} \in \mathcal{L}} \langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle)) + \frac{1}{n} \sum_{i:y_i=-1} \log(1 + \exp(\min_{\boldsymbol{\lambda} \in \mathcal{L}} \langle \boldsymbol{\lambda}, \boldsymbol{x}_i \rangle)) \\
&= \frac{1}{n} \sum_{i:y_i=+1} \log(1 + \exp(-s_i^{\max}) + \frac{1}{n} \sum_{i:y_i=-1} \log(1 + \exp(s_i^{\min})) \\
&= L^{\min}.
\end{aligned}$$

The upper bound can be derived in a similar manner. ■

Proof (Proposition 5). *We are given that $V^{\max} \geq V(\boldsymbol{\lambda}^*)$ where $V(\boldsymbol{\lambda}^*) := l(\boldsymbol{\lambda}^*) + C_0 \|\boldsymbol{\lambda}^*\|_0$ by definition. Thus, we can recover the upper bound from Proposition 5 as follows:*

$$\begin{aligned} l(\boldsymbol{\lambda}^*) + C_0 \|\boldsymbol{\lambda}^*\|_0 &\leq V^{\max}, \\ \|\boldsymbol{\lambda}^*\|_0 &\leq \frac{V^{\max} - l(\boldsymbol{\lambda}^*)}{C_0}, \\ \|\boldsymbol{\lambda}^*\|_0 &\leq \frac{V^{\max} - L^{\min}}{C_0}, \qquad (5) \\ \|\boldsymbol{\lambda}^*\|_0 &\leq \left\lfloor \frac{V^{\max} - L^{\min}}{C_0} \right\rfloor. \qquad (6) \end{aligned}$$

Here, (5) follows from the fact that $L^{\min} \leq l(\boldsymbol{\lambda}^*)$ by definition, and (6) follows from the fact that the number of non-zero coefficients is a natural number. ■

Proof (Proposition 6). *We are given that $V^{\max} \geq V(\boldsymbol{\lambda}^*)$ where $V(\boldsymbol{\lambda}^*) := l(\boldsymbol{\lambda}^*) + C_0 \|\boldsymbol{\lambda}^*\|_0$ by definition. Thus, we can recover the upper bound from Proposition 6 as follows:*

$$\begin{aligned} l(\boldsymbol{\lambda}^*) + C_0 \|\boldsymbol{\lambda}^*\|_0 &\leq V^{\max}, \\ l(\boldsymbol{\lambda}^*) &\leq V^{\max} - C_0 \|\boldsymbol{\lambda}^*\|_0, \\ l(\boldsymbol{\lambda}^*) &\leq V^{\max} - C_0 R^{\min}. \end{aligned}$$

Here, the last line follows from the fact that $R^{\min} \leq \|\boldsymbol{\lambda}^*\|_0$ by definition. ■

Proof (Proposition 7). *We are given that $V^{\min} \leq V(\boldsymbol{\lambda}^*)$ where $V(\boldsymbol{\lambda}^*) := l(\boldsymbol{\lambda}^*) + C_0 \|\boldsymbol{\lambda}^*\|_0$ by definition. Thus, we can recover the lower bound from Proposition 7 as follows:*

$$\begin{aligned} l(\boldsymbol{\lambda}^*) + C_0 \|\boldsymbol{\lambda}^*\|_0 &\geq V^{\min}, \\ l(\boldsymbol{\lambda}^*) &\geq V^{\min} - C_0 \|\boldsymbol{\lambda}^*\|_0, \\ l(\boldsymbol{\lambda}^*) &\geq V^{\min} - C_0 R^{\max}. \end{aligned}$$

Here, the last line follows from the fact that $R^{\max} \geq \|\boldsymbol{\lambda}^*\|_0$ by definition. ■

Appendix B. Small Regularization Parameters do not Influence Accuracy

In Section 2, we state that if the trade-off parameter C_0 in the objective of the risk score problem is sufficiently small, then its optimal solution will attain the best possible trade-off between logistic loss and sparsity. In what follows, we formalize this statement. In what follows, we will omit the intercept term for clarity and explicitly show the regularization parameter C_0 in the RiskSLIM objective so that $V(\boldsymbol{\lambda}; C_0) := l(\boldsymbol{\lambda}) + C_0 \|\boldsymbol{\lambda}\|_0$. We use some new notation shown in Table 5.

Notation	Description
$\mathcal{M} = \operatorname{argmin}_{\boldsymbol{\lambda} \in \mathcal{L}} l(\boldsymbol{\lambda})$	minimizers of the logistic loss
$\mathcal{L}(k) = \{\boldsymbol{\lambda} \in \mathcal{L} \mid \|\boldsymbol{\lambda}\|_0 \leq k\}$	feasible coefficients of models with model size $\leq k$
$\mathcal{M}(k) = \operatorname{argmin}_{\boldsymbol{\lambda} \in \mathcal{L}(k)} l(\boldsymbol{\lambda})$	minimizers of the logistic loss among models of size $\leq k$
$L(k) = \min_{\boldsymbol{\lambda} \in \mathcal{L}(k)} l(\boldsymbol{\lambda})$	logistic loss of minimizers with model size $\leq k$
$\boldsymbol{\lambda}^{\text{opt}} \in \operatorname{argmin}_{\boldsymbol{\lambda} \in \mathcal{M}} \|\boldsymbol{\lambda}\|_0$	sparsest minimizers among all minimizers of the logistic loss
$k^{\text{opt}} = \|\boldsymbol{\lambda}^{\text{opt}}\|_0$	model size of sparsest minimizer

Table 5: Notation used in Remarks 8 and 9

Remark 8 (Minimizers of the Risk Score Problem)
Any optimal solution to the risk score problem will achieve a logistic loss of $L(k)$ for some $k \geq 0$:

$$\min_{\boldsymbol{\lambda} \in \mathcal{L}} V(\boldsymbol{\lambda}; C_0) = \min_{k \in \{0,1,\ldots,k^{\text{opt}}\}} L(k) + C_0 k.$$

Remark 9 (Small Trade-Off Parameters Do Not Influence Accuracy)
There exists an integer $z \geq 1$ such that if

$$C_0 < \frac{1}{z} \left[L(k^{\text{opt}} - z) - L(k^{\text{opt}}) \right],$$

then

$$\boldsymbol{\lambda}^{\text{opt}} \in \operatorname*{argmin}_{\boldsymbol{\lambda} \in \mathcal{L}} V(\boldsymbol{\lambda}; C_0).$$

Remark 9 states that as long as C_0 is sufficiently small, the regularized objective is minimized by any model $\boldsymbol{\lambda}^{\text{opt}}$ that has the smallest size among the most accurate feasible models. In other words, there exists a small enough value of C_0 to guarantee that we will obtain the best possible solution. Since we do not know z in advance and since it is just as difficult to compute L as it is to solve the risk score problem, we will not know in advance how small C_0 must be to avoid sacrificing sparsity. However, it does not matter which value of C_0 we choose as long as it is sufficiently small. In practice, we set $C_0 = 10^{-8}$, which is the smallest value that we can use without running into numerical issues with the MIP solver.

Proof (Remark 8). *Since $L(k)$ is the minimal value of the logistic loss for all models with at most k non-zero coefficients, we have:*

$$L(k) + C_0 k \leq l(\boldsymbol{\lambda}) + C_0 k \text{ for any } \boldsymbol{\lambda} \in \mathcal{L}(k) \tag{7}$$

Denote a feasible minimizer of $V(\boldsymbol{\lambda}; C_0)$ as $\boldsymbol{\lambda}' \in \operatorname{argmin}_{\lambda \in \mathcal{L}} V(\boldsymbol{\lambda}; C_0)$, and let $k' = \|\boldsymbol{\lambda}'\|_0$. Since $\boldsymbol{\lambda}' \in \mathcal{L}(k')$, we have that:

$$L(k') + C_0 k' \leq V(\boldsymbol{\lambda}'; C_0). \tag{8}$$

Taking the minimum of the left hand side of (8):

$$\min_{k \in \{0,1,2,\ldots,k^{\text{opt}}\}} L(k) + C_0 k \leq L(k') + C_0 k'. \tag{9}$$

Combining (8) and (9), we get:

$$\min_{k \in \{0,1,2,\ldots,k^{\text{opt}}\}} L(k) + C_0 k \leq L(k') + C_0 k' \leq \min_{\boldsymbol{\lambda} \in \mathcal{L}} V(\boldsymbol{\lambda}; C_0).$$

If these inequalities are not all equalities, we have a contradiction with the definition of $\boldsymbol{\lambda}'$ and k' as the minimizer of $V(\boldsymbol{\lambda}; C_0)$ and its size. So all must be equality. This proves the statement. ■

Proof (Remark 9). *Note that if $C_0 = 0$, then any minimizer of $V(\boldsymbol{\lambda}; C_0)$ has k^{opt} non-zero coefficients. Consider increasing the value of C_0 starting from zero until a threshold value $C_0^{\min}$, defined as the smallest value such that the minimizer can sacrifice some loss to remove at least one non-zero coefficient. Let $z \geq 1$ be the number of non-zero coefficients removed. For the threshold value $C_0^{\min}$ at which we choose the smaller model rather than the one with size k^{opt}, we have*

$$L(k^{\text{opt}}) + C_0^{\min} k^{\text{opt}} \geq L(k^{\text{opt}} - z) + C_0^{\min}(k^{\text{opt}} - z).$$

Simplifying, we obtain:

$$\frac{1}{z}\left[L(k^{\text{opt}} - z) - L(k^{\text{opt}})\right] \leq C_0^{\min}.$$

Thus, RiskSLIM *does not sacrifice sparsity for logistic loss when:*

$$\frac{1}{z}\left[L(k^{\text{opt}} - z) - L(k^{\text{opt}})\right] > C_0.$$

Here, we know that the value on the left hand side is greater than 0 because $L(\cdot)$ is decreasing in its argument, and if there was no strict decrease, there would be a contradiction with the definition of k^{opt} as the smallest number of terms of an optimal model. In particular, $L(k^{\text{opt}} - z) = L(k^{\text{opt}})$ implies that:

$$\begin{aligned}
\min_{\boldsymbol{\lambda}} V(\boldsymbol{\lambda}; 0) &= L(k^{\text{opt}}) + C_0^{\min} k^{\text{opt}} \\
&= L(k^{\text{opt}} - z) + C_0^{\min}(k^{\text{opt}}) \\
&\geq L(k^{\text{opt}} - z) + C_0^{\min}(k^{\text{opt}} - z) \\
&\geq \min_{\boldsymbol{\lambda}} V(\boldsymbol{\lambda}; 0).
\end{aligned}$$

Thus all inequalities are equalities, which implies that $z = 0$ and contradicts $z \geq 1$. ■

Appendix C. Background on Risk Scores

In Table 6, we present quotes from authors in medicine, criminal justice, and finance to support the claim that risk scores are used because they are easy to use, understand, and validate.

Domain	Reference	Quote
Medicine	Than et al. (2014)	*"Ease of use might be facilitated by presenting a rule developed from logistic regression as a score, where the original predictor weights have been converted to integers that are easy to add together... Though less precise than the original regression formula, such presentations are less complex, easier to apply by memory and usable without electronic assistance."*
Criminal Justice	Duwe and Kim (2016)	*"It is commonplace... for fine-tuned regression coefficients to be replaced with a simple-point system... to promote the easy implementation, transparency, and interpretability of risk-assessment instruments."*
Finance	Finlay (2012)	*"presenting a linear model in the form of a scorecard is attractive because it's so easy to explain and use. In particular, the score can be calculated using just addition to add up the relevant points that someone receives"*

Table 6: Quotes on why risk scores with small integer coefficients are used in different domains.

Importance of Operational Constraints

The approaches used to create risk scores vary significantly for each problem. There is no standard approach within a given domain, (see e.g., the different techniques proposed in criminal justice by Gottfredson and Snyder, 2005; Bobko et al., 2007; Duwe and Kim, 2016), or a given application (see, e.g., the different approaches used to create risk scores for cardiac illness, Six et al., 2008; Antman et al., 2000; Than et al., 2014).

A key reason for the lack of a standardized approach is because risk scores in domains such as medicine and criminal justice need to obey additional *operational constraints* to be used and accepted. In some cases, these constraints can be explicitly stated. Reilly and Evans (2006), for example, describe the requirements put forth by physicians when building a model to detect major cardiac complications for patients with chest pain:

> "*Our physicians... insisted that a new left bundle-branch block be considered as an electrocardiographic predictor of acute ischemia. In addition, they argued that patients who are stratified as low risk by the prediction rule could inappropriately include patients presenting with acute pulmonary edema, ongoing ischemic pain despite maximal medical therapy, or unstable angina after recent coronary revascularization (52). They insisted that such emergent clinical presentations be recommended for coronary care unit admission, not telemetry unit admission.*"

In other cases, however, operational constraints may depend on qualities that are difficult to define a priori. Consider for example, the following statement of Than et al. (2014), that describes the importance of sensibility for deployment:

> "*An important consideration during development is the clinical sensibility of the resulting prediction rule [...] Evaluation of sensibility requires judgment rather than statistical methods. A sensible rule is easy to use, and has content and face validities. Prediction rules are unlikely to be applied in practice if they are not considered sensible by the end-user, even if they are accurate.*"

Approaches to Model Development

Common heuristics used in model development include:

- *Heuristic Feature Selection*: Many approaches use heuristic feature selection to reduce the number of variables in the model. Model development pipelines can often involve multiple rounds of feature selection, and may use different heuristics at each stage (e.g., Antman et al. 2000 use a significance test to remove weak predictors, then use approximate feature selection via forward stepwise regression).
- *Heuristic Rounding*: Many approaches use rounding heuristics to produce models with integer coefficients. In the simplest case, this involves scaling and rounding the coefficients from a logistic regression model (e.g., Goel et al., 2016) or a linear probability model (e.g., U.S. Department of Justice, 2005). The SAPS II score (Le Gall et al., 1993), for example, was built in this way ("*the general rule was to multiply the β for each range by 10 and round off to the nearest integer.*")
- *Expert Judgement*: A common approach to model development involves having a panel experts build a model by hand, and using data to validate the model after it is built (e.g., for the CHADS_2 score for stroke prediction of Gage et al. 2001, and the National Early Warning Score to assess acute illness in the ICU of McGinley and Pearse 2012). Expert judgement can also be used in data-driven approaches. In developing the EDACS score (Than et al., 2014), for example, expert judgement was used to: (i) determine a scaling factor for model coefficients ("*The beta coefficients were multiplied by eight, which was the smallest common multiplication factor possible to obtain a sensible score that used whole numbers and facilitated clinical ease of use.*") (ii) convert a continuous variable into a binary variables ("*Age was the only continuous variable to be included in the final score. It was converted to a categorical variable, using 5-year age bands with increasing increments of +2 points.*")
- *Unit Weighting*: This technique aims to produce a score by simply adding together all variables that are significantly correlated with the outcome of interest. Unit weighting is prevalent in criminal justice (see, e.g., Bobko et al., 2007; Duwe and Kim, 2016), where it is referred to as the *Burgess* method (as it was first proposed by Burgess, 1928). The use of this technique is frequently motivated by empirical work showing that linear models with unit weights may perform surprisingly well (see, e.g., Einhorn and Hogarth, 1975; Dawes, 1979; Holte, 1993, 2006; Bobko et al., 2007).

Critical Analysis of a Real-World Training Pipeline

Many risk scores are built using sequential *training pipelines* that combine traditional statistical techniques, heuristics, and expert judgement. The TIMI Risk Score of Antman et al. (2000), for example, was built as follows:

1. *"A total of 12 baseline characteristics arranged in a dichotomous fashion were screened as candidate predictor variables of risk of developing an end-point event"*

2. *"After each factor was tested independently in a univariate logistic regression model, those that achieved a significance level of $p < .20$ were [retained]."*

3. *"[The remaining factors]... selected for testing in a multivariate step-wise (backward elimination) logistic regression model. Variables associated with $p < .05$ were retained in the final model."*

4. *"After development of the multivariate model, the [risk predictions were determined]... for the test cohort using those variables that had been found to be statistically significant predictors of events in the multivariate analysis."*

5. *"The score was then constructed by a simple arithmetic sum of the number of variables present."*

Although this pipeline uses several established statistical techniques (e.g. stepwise regression, significance testing), it is unlikely to output a risk score that attains the best possible performance because:

- Decisions involving feature selection and rounding are made sequentially rather than globally (e.g., Steps 1-3 involve feature selection, Step 5 involves rounding).
- The objective function that is optimized at each step differs from the global performance metric of interest.
- Some steps optimize conflicting objective functions (e.g., backward elimination typically optimizes the AIC or BIC, while the final model is fit to optimize the logistic loss).
- Some steps do not fully optimize their own objective function (i.e., backward elimination does not return a globally optimal feature set).
- Some steps depend free parameters that are set without validation (e.g., the threshold significance level of $p < .20$ used in Step 2)
- The final model is not trained using all the available training data. Here, Steps 4 and 5 use data from a "test cohort" that could have been used to improve the fit of the final model.
- The coefficients of the final model are set to $+1$ and not optimized (that is, the model uses unit weights).

Appendix D. Details on Computational Experiments

In this appendix, we provide additional details on the computational experiments in Sections 3.3 and 4.

D.1. Simulation Procedure for Synthetic Datasets

We ran the computational experiments in Sections 3.3 and 4 using a collection of synthetic datasets that we generated from the `breastcancer` dataset of Mangasarian et al. (1995), which contains $n = 683$ samples and $d = 9$ features $x_{ij} \in \{0, \ldots, 10\}$. This choice was based off the fact that the `breastcancer` dataset produces a RiskSlimMINLP instance that can be solved using many algorithms, the data have been extensively studied in the literature, and can be obtained from the UCI ML repository (Bache and Lichman, 2013).

We show the simulation procedure in Algorithm 6. Given the original dataset, this procedure generates a collection of nested synthetic datasets in two steps. First, it generates the largest dataset (with $n^{\max} = 5 \times 10^6$ and $d^{\max} = 30$) by replicating features and samples from the original dataset and adding normally distributed noise. Second, it produces smaller datasets by taking nested subsets of the samples and features. This ensures that a synthetic dataset with d features and n samples contains the same features and examples as a smaller synthetic dataset (i.e., with $d' < d$ features and $n' < n$ samples).

The resulting collection of synthetic datasets has two properties that are useful for benchmarking the performance of algorithms for RiskSlimMINLP:

1. They produce difficult instances of the risk score problem. Here, the RiskSlimMINLP instances for synthetic datasets with $d > 9$ are challenging because the dataset contains replicates of the original 9 features. In particular, feature selection becomes exponentially harder when the dataset contains several copies of highly correlated features, as it means that there are an exponentially larger number of slightly suboptimal solutions.

2. They can be used to make inferences about the optimal objective value of RiskSlimMINLP instances we may not be able to solve. Say, for example, that we could not solve an instance of the risk score problem for a synthetic dataset with $(d, n) = (20, 10^6)$, but could solve an instance for a smaller synthetic dataset with $(d, n) = (10, 10^6)$. In this case, we know that the optimal value of an RiskSlimMINLP instance where $(d, n) = (20, 10^6)$ must be less than or equal to the optimal value of an RiskSlimMINLP instance where $(d, n) = (10, 10^6)$. This is because the $(d, n) = (20, 10^6)$ dataset contains all of the features as the $(d, n) = (10, 10^6)$ dataset.

Algorithm 6 Simulation Procedure to Generate Nested Synthetic Datasets

Input

$X^{\text{original}} \leftarrow [x_{ij}]_{i=1\ldots n^{\text{original}}, j=1\ldots d^{\text{original}}}$ — feature matrix of original dataset
$Y^{\text{original}} \leftarrow [y_i]_{i=1\ldots n^{\text{original}}}$ — label vector of original dataset
$d^1 \ldots d^{\max}$ — dimensions for synthetic datasets (increasing order)
$n^1 \ldots n^{\max}$ — sample sizes for synthetic datasets (increasing order)

Initialize

$\mathcal{J}^{\text{original}} \leftarrow [1, \ldots, d^{\text{original}}]$ — index array for original features
$\mathcal{J}^{\max} \leftarrow []$ — index array of features for largest synthetic dataset
$m^{\text{full}} \leftarrow \lfloor d^{\max}/d^{\text{original}} \rfloor$
$m^{\text{remainder}} \leftarrow d^{\max} - m^{\text{full}} \cdot d^{\text{original}}$

Step I: Generate Largest Dataset

for $m = 1, \ldots, m^{\text{full}}$ **do**
  $\mathcal{J}^{\max} \leftarrow [\mathcal{J}^{\max}, \text{RandomPermute}(\mathcal{J}^{\text{original}})]$
end for
$\mathcal{J}^{\max} \leftarrow [\mathcal{J}^{\max}, \text{RandomSampleWithoutReplacement}(\mathcal{J}^{\text{original}}, m^{\text{remainder}})]$
for $i = 1, \ldots, n^{\max}$ **do**
  sample l with replacement from $1, \ldots, n^{\text{original}}$
  $y_i^{\max} \leftarrow y_i$
  for $j = 1, \ldots, d^{\max}$ **do**
    $k \leftarrow \mathcal{J}^{\max}[j]$
    sample ε from Normal$(0, 0.5)$
    $x_{ij}^{\max} \leftarrow \lceil x_{l,k} + \varepsilon \rfloor$ — *new features are noisy versions of original features*
    $x_{ij}^{\max} \leftarrow \min(10, \max(0, x_{ij}^{\max}))$ — *new features have same bounds as old features*
  end for
end for
$X^{\max} \leftarrow [x_{ij}^{\max}]_{i=1\ldots n^{\max}, j=1\ldots d^{\max}}$
$Y^{\max} \leftarrow [y_i^{max}]_{i=1\ldots n^{\max}}$

Step II: Generate Smaller Datasets

for $d = [d^1, \ldots, d^{\max}]$ **do**
  for $n = [n^1, \ldots, n^{\max}]$ **do**
    $X^{(n,d)} = X^{\max}[1:n, 1:d]$
    $Y^n = Y^{\max}[1:n]$
  end for
end for

Output: synthetic datasets $(X^{(n,d)}, Y^n)$ for all $n^1 \ldots n^{\max}$ and $d^1 \ldots d^{\max}$.

D.2. Setup on the Performance Comparison in Section 3.3

We consider an instance of RiskSlimMINLP where $C_0 = 10^{-8}$, $\lambda_0 \in \{-100, 100\}$, and $\lambda_j = \{-10, \ldots, 10\}$ for $j = 1, \ldots, d$. We solved this instance on a 3.33 GHz Intel Xeon CPU with 16GB of RAM for up to 6 hours using the following algorithms:

(i) CPA, as described in Algorithm 1;

(ii) LCPA, as described in Algorithm 2;

(iii) ActiveSetMINLP, an active set MINLP algorithm;

(iv) InteriorMINLP, an interior point MINLP algorithm;

(v) InteriorCGMINLP, an interior point MINLP algorithm where the primal-dual KKT system is solved with a conjugate gradient method;

All MINLP algorithms were implemented in a state-of-the-art commercial solver (i.e., Artelsys Knitro 9.0, which is an updated version of the solver described in Byrd et al. 2006). If an algorithm did not return a certifiably optimal solution for a particular instance within the 6 hour time limit, we reported results for the best feasible solution. Both CPA and LCPA were implemented using CPLEX 12.6.3.

Our CPA implementation uses the solves the following formulation of RiskSlimMIP.

Definition 10 (RiskSlimMIP)
Given a finite coefficient set $\mathcal{L} \subset \mathbb{Z}^{d+1}$, trade-off parameter $C_0 > 0$, and cutting plane approximation $\hat{l}^k : \mathbb{R}^{d+1} \to \mathbb{R}_+$ with cut parameters $\{l(\boldsymbol{\lambda}^t), \nabla l(\boldsymbol{\lambda}^t)\}_{t=1}^k$, the surrogate optimization problem RiskSlimMIP$(\hat{l}^k)$ *can be formulated as the mixed integer program:*

$$
\begin{array}{llllll}
\min\limits_{L,\boldsymbol{\lambda},\alpha} & V & & & & \\
\text{s.t.} & V & = & L + C_0 R & & \textit{objective value} \quad (10a) \\
 & L & \geq & l(\boldsymbol{\lambda}^t) + \langle \nabla l(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t \rangle & t = 1,\ldots,k & \textit{cut constraints} \quad (10b) \\
 & R & = & \sum\limits_{j=1}^{d} \alpha_j & & \ell_0\textit{-norm} \quad (10c) \\
 & \lambda_j & \leq & \Lambda_j^{\max} \alpha_j & j = 1,\ldots,d & \ell_0 \textit{ indicator constraints} \quad (10d) \\
 & \lambda_j & \geq & -\Lambda_j^{\min} \alpha_j & j = 1,\ldots,d & \ell_0 \textit{ indicator constraints} \quad (10e) \\
 & V & \in & [V^{\min}, V^{\max}] & & \textit{bounds on objective value} \quad (10f) \\
 & L & \in & [L^{\min}, L^{\max}] & & \textit{bounds on loss value} \quad (10g) \\
 & R & \in & \{R^{\min}, \ldots, R^{\max}\} & & \textit{bounds on } \ell_0\textit{-norm} \quad (10h) \\
 & \lambda_j & \in & \{\Lambda_j^{\min}, \ldots, \Lambda_j^{\max}\} & j = 1,\ldots,d & \textit{coefficient bounds} \quad (10i) \\
 & \alpha_j & \in & \{0, 1\} & j = 1,\ldots,d & \ell_0 \textit{ indicator variables}
\end{array}
$$

The formulation in Definition 10 contains $2d + 3$ variables and $k + 2d + 2$ constraints (excluding bounds on the variables). Here, the cutting plane approximation is represented via the cut constraints in (10b) and the auxiliary variable $L \in \mathbb{R}_+$. The ℓ_0–norm is computed using the indicator variables $\alpha_j = \mathbb{1}[\lambda_j \neq 0]$ set in constraints (10d) and (10e). The λ_j are restricted to a bounded set of integers in constraints (10i). The formulation includes two additional auxiliary variables: V, defined as the objective value in (10a); and R, defined as the ℓ_0-norm in (10c).

D.3. Results for MINLP Algorithms

In Figure 27, we plot results for all MINLP algorithms that we benchmarked against CPA and LCPA: ActiveSetMINLP, InteriorMINLP, and InteriorCGMINLP. We reported results for ActiveSetMINLP in Section 3.3 because it solved the largest number of instances to optimality.

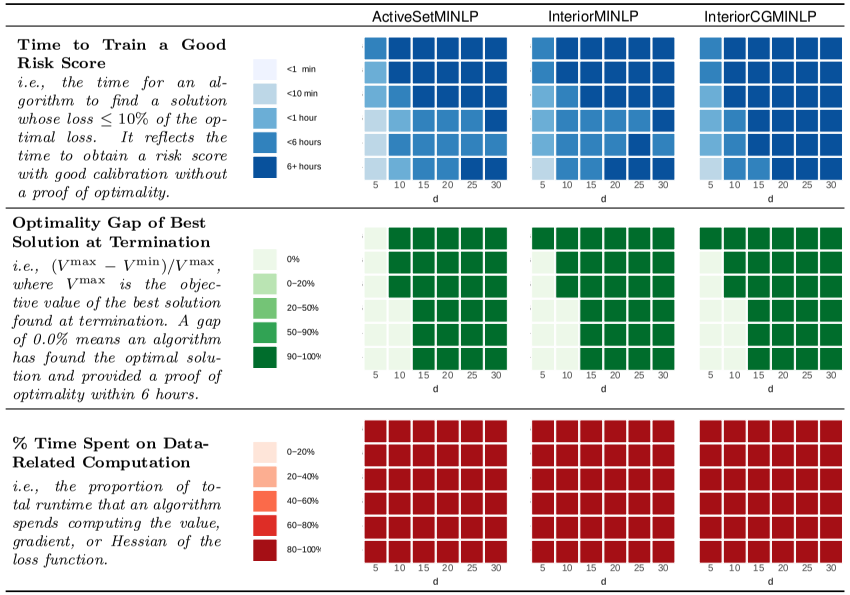


Figure 27: Performance of MINLP algorithms on difficult instances of RiskSlimMINLP for synthetic datasets with varying dimensions d and sample sizes n. All algorithms perform similarly. We report results for ActiveSetMINLP in Section 3.3 because it solves the most instances to optimality.

Appendix E. Additional Details on Algorithmic Improvements

E.1. Initialization Procedure

In Algorithm 7, we present an initialization procedure for LCPA. The procedure aims to speed up LCPA by generating: a collection of cutting planes; a good integer solution; and non-trivial bounds on the values of the objective, loss, and number of non-zero coefficients. It combines all of the techniques from this section, as follows:

1. *Run* CPA *using* RiskSlimLP: We apply traditional CPA to produce a cutting-plane approximation of the loss function using a convex relaxation of RiskSlimMINLP. We can achieve this by replacing RiskSlimMIP(k) in Step 2 of CPA with RiskSlimLP$(k, \text{conv}(\mathcal{L}))$, and running CPA until a user-specified stopping condition is met. We store: (i) the cuts from RiskSlimLP (which will be used as an initial cutting plane approximation for LCPA); (ii) the lower bound from CPA on the objective value of RiskSlimLP (which represents a lower bound $V^{\min}$ on the optimal value of RiskSlimMINLP).

2. *Sequential Rounding and Polishing*: We collect the solutions produced at each iteration of CPA. For each solution, we run SequentialRounding to obtain an integer solution for RiskSlimMINLP, and then polish it using DCD. We use the best solution found so far to update the upper bound $V^{\max}$ on the optimal value to RiskSlimMINLP.

3. *Chained Updates*: Having obtained non-trivial bounds on $V^{\min}$ and $V^{\max}$, we update the bounds on key quantities using ChainedUpdates.

In Figure 28, we show how the initialization procedure in Algorithm 7 improves the lower bound and the optimality gap over the course of LCPA.

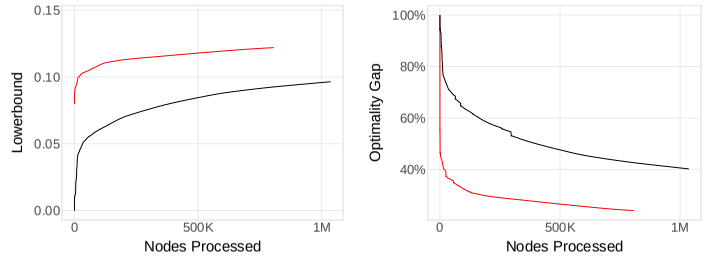


Figure 28: Performance profile of LCPA in a basic implementation (black) and with the initialization procedure in Algorithm 7 (red). Results reflect performance on a RiskSlimMINLP instance for a synthetic dataset with $d = 30$ and $n = 50{,}000$ (see Appendix D for details).

E.2. Reducing Data-Related Computation

In this section, we present techniques to reduce data-related computation for the risk score problem.

Algorithm 7 Initialization Procedure for LCPA

Input

$(\boldsymbol{x}_i, y_i)_{i=1}^n$	training data
$\mathcal{L}$	coefficient set
C_0	ℓ_0 penalty parameter
$V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, $R^{\min}$, $R^{\max}$	initial bounds on $V(\boldsymbol{\lambda}^*)$, $l(\boldsymbol{\lambda}^*)$ and $\|\boldsymbol{\lambda}^*\|_0$
$T^{\max}$	time limit for CPA on RiskSlimLP

Initialize

$\hat{l}^0(\boldsymbol{\lambda}) \leftarrow \{0\}$	initial approximation of loss function
$\mathcal{P}^{\text{int}} \leftarrow \emptyset$	initial collection of integer solutions

Step I: Run CPA using subroutine RiskSlimLP

Run CPA	*Algorithm 1 using* RiskSlimLP$(\hat{l}^0, \text{conv}(\mathcal{L}))$
$k \leftarrow$ number of cuts added by CPA within time limit $T^{\max}$	
$\hat{l}^{\text{initial}} \leftarrow \hat{l}^k$	*store cuts from each iteration*
$\mathcal{P}^{\text{real}} \leftarrow \{\boldsymbol{\lambda}^t\}_{t=1}^k$	*store solutions from each iteration*
$V^{\min} \leftarrow$ lower bound from CPA	*lower bound for* RiskSlimLP *is lower bound for* RiskSlimMINLP

Step II: Round and Polish Non-Integer Solutions from CPA

for each $\boldsymbol{\lambda}^{\text{real}} \in \mathcal{P}^{\text{real}}$ **do**	
$\quad\boldsymbol{\lambda}^{\text{sr}} \leftarrow$ SequentialRounding $(\boldsymbol{\lambda}^{\text{real}}, \mathcal{L}, C_0)$	*Algorithm 4*
$\quad\boldsymbol{\lambda}^{\text{dcd}} \leftarrow$ DCD $(\boldsymbol{\lambda}^{\text{sr}}, \mathcal{L}, C_0)$	*Algorithm 3*
$\quad\mathcal{P}^{\text{int}} \leftarrow \mathcal{P}^{\text{int}} \cup \{\boldsymbol{\lambda}^{\text{dcd}}\}$	*store polished integer solutions*
end for	
$\boldsymbol{\lambda}^{\text{best}} \leftarrow \text{argmin}_{\boldsymbol{\lambda} \in \mathcal{P}^{\text{int}}} V(\boldsymbol{\lambda})$	
$V^{\max} \leftarrow V(\boldsymbol{\lambda}^{\text{best}})$	*best integer solution produces upper bound for* $V(\boldsymbol{\lambda}^*)$

Step III: Update Bounds on Objective Terms

$(V^{\min}, \ldots, R^{\max}) \leftarrow$ ChainedUpdates $(V^{\min}, \ldots, R^{\max}, C_0)$	*Algorithm 5*

Output: $\boldsymbol{\lambda}^{\text{best}}$, $\hat{l}^{\text{initial}}(\boldsymbol{\lambda})$, $V^{\min}$, $V^{\max}$, $L^{\min}$, $L^{\max}$, $R^{\min}$, $R^{\max}$

E.2.1. Fast Loss Evaluation via a Lookup Table

The first technique is designed to speed up the evaluation of the loss function and its gradient, which reduces runtime when we compute cut parameters (3) and call the rounding and polishing procedures in Section 4. The technique requires that the features $\boldsymbol{x}_i$ and coefficients $\boldsymbol{\lambda}$ belong to sets that are bounded, discrete, and regularly spaced, such as $\boldsymbol{x}_i \in \mathcal{X} \subseteq \{0,1\}^d$ and $\boldsymbol{\lambda} \in \mathcal{L} \subseteq \{-10, \ldots, 10\}^{d+1}$.

Evaluating the logistic loss, $\log(1 + \exp(-\langle \boldsymbol{\lambda}, y_i \boldsymbol{x}_i \rangle))$, is a relatively expensive operation because it involves exponentiation and must be carried out in multiple steps to avoid numerical overflow/underflow when the *scores* $s_i = \langle \boldsymbol{\lambda}, \boldsymbol{x}_i y_i \rangle$ are too small or large[3]. When the training data and coefficients belong to discrete bounded sets, the scores $s_i = \langle \boldsymbol{\lambda}, \boldsymbol{x}_i y_i \rangle$ belong to a discrete and bounded set

$$\mathcal{S} = \left\{ \langle \boldsymbol{\lambda}, \boldsymbol{x}_i y_i \rangle \;\middle|\; i = 1, \ldots, n \text{ and } \boldsymbol{\lambda} \in \mathcal{L} \right\}.$$

If the elements of the feature set $\mathcal{X}$ and the coefficient set $\mathcal{L}$ are regularly spaced, then the scores belong to the set of integers $\mathcal{S} \subseteq \mathbb{Z} \cap [s^{\min}, s^{\max}]$ where:

$$s^{\min} = \min_{i,\boldsymbol{\lambda}} \left\{ \langle \boldsymbol{\lambda}, \boldsymbol{x}_i y_i \rangle \text{ for all } (\boldsymbol{x}_i, y_i) \in \mathcal{D} \text{ and } \boldsymbol{\lambda} \in \mathcal{L} \right\},$$
$$s^{\max} = \max_{i,\boldsymbol{\lambda}} \left\{ \langle \boldsymbol{\lambda}, \boldsymbol{x}_i y_i \rangle \text{ for all } (\boldsymbol{x}_i, y_i) \in \mathcal{D} \text{ and } \boldsymbol{\lambda} \in \mathcal{L} \right\}.$$

Thus, we can precompute and store all possible values of the loss function in a lookup table with $s^{\max} - s^{\min} + 1$ rows, where row m contains the value of $[\log(1 + \exp(-(m + s^{\min} - 1)))]$.

This strategy can reduce the time to evaluate the loss as it replaces a computationally expensive operation with a fast lookup. In practice, the lookup table is small enough to be cached in memory, which yields a substantial speedup. In addition, when $R^{\max}$ is updated over the course of LCPA, the lookup table can be further reduced by recomputing $s^{\min}$ and $s^{\max}$, and limiting the entries to values between $s^{\min}$ and $s^{\max}$. The values of $s^{\min}$ and $s^{\max}$ can be computed in $\mathrm{O}(n)$ time, so the update is not expensive.

E.2.2. Faster Rounding Heuristics via Subsampling

We now describe a subsampling technique to reduce computation for rounding heuristics that require multiple evaluations of the loss function (e.g., SequentialRounding).

Ideally, we would want to run such heuristics frequently as possible because each run may output a solution that updates the incumbent solution (i.e., the current best solution to the risk score problem). In practice, however, this may slow down LCPA since each run requires multiple evaluations of the loss, and runs that fail to update the incumbent amount to wasted computation. If, for example, we ran SequentialRounding each time the MIP solver found a set of non-integer coefficients in LCPA (i.e., Step 15 of Algorithm 2), many rounded solutions would not update the incumbent, and we would have wasted too much time rounding, without necessarily finding a better solution.

Our technique aims to reduce the overhead of calling heuristics by running them on a smaller dataset $\mathcal{D}_m$ built by sampling m points without replacement from the training

3. The value of $\exp(s)$ can be computed reliably using IEEE 754 double precision floating point numbers for $s \in [-700, 700]$. The term will overflow to ∞ when $s < -700$, and underflow to 0 when when $s > 700$.

dataset $\mathcal{D}_n$. In what follows, we present probabilistic guarantees to choose the number of samples m so that an incumbent update using $\mathcal{D}_m$ guarantees an incumbent update using $\mathcal{D}_n$. To clarify when the loss and objective are computed using $\mathcal{D}_m$ or $\mathcal{D}_n$, we let $l_i(\boldsymbol{\lambda}) = \log(1 + \exp(\langle \boldsymbol{\lambda}, y_i \boldsymbol{x}_i \rangle))$ and define:

$$l_m(\boldsymbol{\lambda}) = \frac{1}{m}\sum_{i=1}^{m} l_i(\boldsymbol{\lambda}), \qquad V_m(\boldsymbol{\lambda}) = l_m(\boldsymbol{\lambda}) + C_0 \|\boldsymbol{\lambda}\|_0,$$

$$l_n(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n} l_i(\boldsymbol{\lambda}), \qquad V_n(\boldsymbol{\lambda}) = l_n(\boldsymbol{\lambda}) + C_0 \|\boldsymbol{\lambda}\|_0.$$

Consider a case where a heuristic returns a promising solution $\boldsymbol{\lambda}^{\text{hr}}$ such that:

$$V_m(\boldsymbol{\lambda}^{\text{hr}}) < V^{\max}. \tag{11}$$

In this case, we compute the objective on the full training dataset $\mathcal{D}_n$ by evaluating the loss for each of the $n - m$ points that were not included in $\mathcal{D}_m$. We then update the incumbent solution if $\boldsymbol{\lambda}^{\text{hr}}$ attains an objective value that is less than the current upper bound:

$$V_n(\boldsymbol{\lambda}^{\text{hr}}) < V^{\max}. \tag{12}$$

Although this strategy requires evaluating the loss for the full training dataset to validate an incumbent update, it still reduces data-related computation since rounding heuristics typically require multiple evaluations of the loss (e.g., SequentialRounding, which requires $\frac{d}{3}(d^2 + 3d + 2)$ evaluations).

To guarantee that any solution that updates the incumbent when the objective is evaluated with $\mathcal{D}_m$ will also update the incumbent when the objective is evaluated with $\mathcal{D}_n$ (i.e., that any solution that satisfies (11) will also satisfy (12)), we can use the generalization bound from Theorem 11.

Theorem 11 (Generalization of Sampled Loss on Finite Coefficient Set)
Let $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$ denote a training dataset with $n > 1$ points, $\mathcal{D}_m = (\boldsymbol{x}_i, y_i)_{i=1}^m$ denote a sample of m points drawn without replacement from $\mathcal{D}_n$, and $\boldsymbol{\lambda}$ denote the coefficients of a linear classifier from a finite set $\mathcal{L}$. For all $\varepsilon > 0$, it holds that

$$\Pr\left(\max_{\boldsymbol{\lambda}\in\mathcal{L}} \left(l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda})\right) \geq \varepsilon\right) \leq |\mathcal{L}| \exp\left(-\frac{2\varepsilon^2}{(\frac{1}{m})(1 - \frac{m^2}{n^2})\Delta^{\max}(\mathcal{L}, \mathcal{D}_n)^2}\right),$$

where $\Delta^{\max}(\mathcal{L}, \mathcal{D}_n) = \max_{\boldsymbol{\lambda}\in\mathcal{L}} \left(\max_{i=1,\ldots,n} l_i(\boldsymbol{\lambda}) - \min_{i=1,\ldots,n} l_i(\boldsymbol{\lambda})\right).$

Proof (Theorem 11). *For a fixed set coefficient vector $\boldsymbol{\lambda} \in \mathcal{L}$, consider a sample of n points composed of the values for the loss function $l_i(\boldsymbol{\lambda})$ for each example in the full training dataset $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$. Let $l_n(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^n l_i(\boldsymbol{\lambda})$ and $l_m(\boldsymbol{\lambda}) = \frac{1}{m}\sum_{i=1}^m l_i(\boldsymbol{\lambda})$. Then, the Hoeffding-Serfling inequality (see e.g., Theorem 2.4 in Bardenet and Maillard, 2015) guarantees the following for all $\varepsilon > 0$:*

$$\Pr\left(l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda}) \geq \varepsilon\right) \leq \exp\left(-\frac{2\varepsilon^2}{(\frac{1}{m})(1 - \frac{m}{n})(1 + \frac{m}{n})\Delta(\boldsymbol{\lambda}, \mathcal{D}_n)^2}\right),$$

where

$$\Delta(\boldsymbol{\lambda}, \mathcal{D}_n) = \max_{i=1,\ldots,n} l_i(\boldsymbol{\lambda}) - \min_{i=1,\ldots,n} l_i(\boldsymbol{\lambda}).$$

We recover the desired inequality by generalizing this bound to hold for all $\boldsymbol{\lambda} \in \mathcal{L}$ as follows.

$$\begin{aligned}
\Pr\left(\max_{\boldsymbol{\lambda}\in\mathcal{L}} \left(l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda})\right) \geq \varepsilon\right) &= \Pr\left(\bigcup_{\boldsymbol{\lambda}\in\mathcal{L}} \left(l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda}) \geq \varepsilon\right)\right), \\
&\leq \sum_{\boldsymbol{\lambda}\in\mathcal{L}} \Pr\left(l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda}) \geq \varepsilon\right), && (13) \\
&\leq \sum_{\boldsymbol{\lambda}\in\mathcal{L}} \exp\left(-\frac{2\varepsilon^2}{(\frac{1}{m})(1 - \frac{m}{n})(1 + \frac{m}{n})\Delta(\boldsymbol{\lambda}, \mathcal{D}_n)^2}\right), && (14) \\
&\leq |\mathcal{L}| \exp\left(-\frac{2\varepsilon^2}{(\frac{1}{m})(1 - \frac{m}{n})(1 + \frac{m}{n})\Delta^{\max}(\mathcal{L}, \mathcal{D}_n)^2}\right). && (15)
\end{aligned}$$

Here, (13) *follows from the union bound,* (14) *follows from the Hoeffding Serling inequality,* (15) *follows from the fact that $\Delta(\boldsymbol{\lambda}, \mathcal{D}_n) \leq \Delta^{\max}(\mathcal{L}, \mathcal{D}_n)$ given that $\boldsymbol{\lambda} \in \mathcal{L}$.* ■

Theorem 11 is derived from a concentration inequality for sampling without replacement, called the Hoeffding-Serfling inequality (see Bardenet and Maillard, 2015). The Hoeffding-Serfling inequality is tighter than the classical Hoeffding inequality as it ensures that $\Pr\left(l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda}) \geq \epsilon\right) \to 0$ as $m \to n$ for all $\epsilon > 0$. Here, $\Delta^{\max}(\mathcal{L}, \mathcal{D}_n)$ is a normalization term that represents the maximum range of the loss and can be computed quickly using the training dataset $\mathcal{D}_n$ and coefficient set $\mathcal{L}$ as shown in Proposition 4 in Section 4.3.

In general machine learning settings, the $|\mathcal{L}|$ term in Theorem 11 would yield a vacuous bound. In this setting, however, rounding ensures that $\mathcal{L}$ contains at most 2^d elements, which produces a well-defined bound on the difference between $l_n(\boldsymbol{\lambda})$ and $l_m(\boldsymbol{\lambda})$. As a result, Theorem 11 can be used to assess the probability that a proposed incumbent update leads to an actual incumbent update (see Corollary 12). Alternatively, it can be used to set the sample size m so that an incumbent update on $\mathcal{D}_m$ is likely to yield an incumbent update on $\mathcal{D}_n$. In practice, the bound in Theorem 11 can be tightened by recomputing the normalization term $\Delta^{\max}(\mathcal{L}, \mathcal{D}_n)$ over the course of LCPA. This can be done for each real-valued solution ρ, or when the MIP solver restricts the set of feasible coefficients $\mathcal{L}$.

Corollary 12 (Update Probabilities of Rounding Heuristics on Subsampled Data) *Consider a rounding heuristic that takes as input a vector of real-valued coefficients* $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_d) \in \operatorname{conv}(\mathcal{L})$ *and outputs a vector of rounded coefficients* $\boldsymbol{\lambda} \in \mathcal{L}_{|\boldsymbol{\rho}}$ *where*

$$\mathcal{L}_{\boldsymbol{\rho}} = \left(\boldsymbol{\lambda} \in \mathcal{L} \mid \lambda_j \in \{\lceil \rho_j \rceil, \lfloor \rho_j \rfloor\} \text{ for } j = 1, \ldots, d\right).$$

If we evaluate the rounding heuristic using m *points* $\mathcal{D}_m = (\boldsymbol{x}_i, y_i)_{i=1}^m$ *drawn without replacement from* $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$ *and the rounded coefficients* $\boldsymbol{\lambda} \in \mathcal{L}_{\boldsymbol{\rho}}$ *attain an objective value* $V_m(\boldsymbol{\lambda})$*, then for any* δ*, with probability at least* $1-\delta$*, we have that*

$$V_m(\boldsymbol{\lambda}) < V^{\max} - \varepsilon_\delta \implies V_n(\boldsymbol{\lambda}) \leq V^{\max},$$

where

$$\varepsilon_\delta = \Delta^{\max}(\mathcal{L}_{\boldsymbol{\rho}}, \mathcal{D}_n)\sqrt{\frac{\log(1/\delta) + d\log(2)}{2}\left(\frac{1}{m}\right)\left(1 - \frac{m^2}{n^2}\right)}.$$

Proof (Corollary 12). *We will first show that for any tolerance* $\delta > 0$ *that we pick, the prescribed choice of* ε_δ *will ensure that* $V_n(\boldsymbol{\lambda}) - V_m(\boldsymbol{\lambda}) \leq \varepsilon_\delta$ *w.p. at least* $1-\delta$*. Restating the result of Theorem 11, we have that for any* $\varepsilon > 0$*:*

$$\Pr\left(\max_{\boldsymbol{\lambda} \in \mathcal{L}} \left(l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda})\right) \geq \varepsilon\right) \leq |\mathcal{L}| \exp\left(-\frac{2\varepsilon^2}{(\frac{1}{m})(1-\frac{m}{n})(1+\frac{m}{n})\Delta^{\max}(\mathcal{L}, \mathcal{D}_n)^2}\right). \quad (16)$$

Note that $l_n(\boldsymbol{\lambda}) - l_m(\boldsymbol{\lambda}) = V_n(\boldsymbol{\lambda}) - V_m(\boldsymbol{\lambda})$ *for any fixed* $\boldsymbol{\lambda}$*. In addition, note that the set of rounded coefficients* $\mathcal{L}(\boldsymbol{\rho})$ *contains at most* $|\mathcal{L}(\boldsymbol{\rho})| \leq 2^d$ *coefficient vectors. Therefore, in this setting,* (16) *implies that for any* $\varepsilon > 0$*,*

$$\Pr\left(V_n(\boldsymbol{\lambda}) - V_m(\boldsymbol{\lambda}) \geq \varepsilon\right) \leq 2^d \exp\left(-\frac{2\varepsilon^2}{(\frac{1}{m})(1-\frac{m}{n})(1+\frac{m}{n})\Delta(\mathcal{L}(\boldsymbol{\rho}), \mathcal{D}_n)^2}\right). \quad (17)$$

By setting $\varepsilon = \varepsilon_\delta$ *and simplifying the terms on the right hand side in* (17)*, we see that*

$$\Pr\left(V_n(\boldsymbol{\lambda}) - V_m(\boldsymbol{\lambda}) \geq \varepsilon_\delta\right) \leq \delta.$$

Thus, the prescribed value of ε_δ *ensures that* $V_n(\boldsymbol{\lambda}) - V_m(\boldsymbol{\lambda}) \leq \varepsilon_\delta$ *w.p. at least* $1-\delta$*.*

Since we have set ε_δ *so that* $V_n(\boldsymbol{\lambda}) - V_m(\boldsymbol{\lambda}) \leq \varepsilon_\delta$ *w.p. at least* $1-\delta$*, we now need only to show that any* $\boldsymbol{\lambda}$ *satisfying* $V_m(\boldsymbol{\lambda}) < V^{\max} - \varepsilon_\delta$ *will also satisfy* $V_n(\boldsymbol{\lambda}) \leq V^{\max}$ *to complete the proof. To see this, observe that:*

$$\begin{aligned} V_n(\boldsymbol{\lambda}) - V_m(\boldsymbol{\lambda}) &\leq \varepsilon_\delta, \\ V_n(\boldsymbol{\lambda}) &\leq V_m(\boldsymbol{\lambda}) + \varepsilon_\delta, \\ V_n(\boldsymbol{\lambda}) &< V^{\max}. \end{aligned} \quad (18)$$

Here, (18) *follows from the fact that* $V_m(\boldsymbol{\lambda}) < V^{\max} - \varepsilon_\delta \implies V_m(\boldsymbol{\lambda}) + \varepsilon_\delta < V^{\max}$. ■

Appendix F. Additional Experimental Results

		Traditional Approaches			Pooled Approaches			
Dataset	**Metric**	PLR≫Rd	PLR≫RsRd	PLR≫Unit	PooledRd	PooledRd*	PooledSeqRd*	RiskSLIM
income	**test cal**	10.5%	19.5%	25.4%	3.0%	3.1%	4.2%	2.6%
$n = 32561$	**train cal**	10.5%	19.8%	25.8%	2.6%	2.5%	4.4%	4.2%
$d = 36$	**test auc**	0.787	0.813	0.814	0.845	0.854	0.832	0.854
	train auc	0.787	0.811	0.815	0.848	0.857	0.827	0.860
mammo	**test cal**	10.5%	16.2%	8.5%	10.9%	7.1%	7.4%	5.0%
$n = 961$	**train cal**	12.2%	14.2%	7.2%	10.1%	5.4%	5.4%	3.1%
$d = 14$	**test auc**	0.832	0.846	0.842	0.845	0.841	0.845	0.843
	train auc	0.846	0.852	0.850	0.847	0.847	0.847	0.849
mushroom	**test cal**	22.1%	8.0%	19.9%	12.6%	4.6%	5.4%	1.8%
$n = 8124$	**train cal**	28.6%	5.6%	18.3%	11.9%	4.2%	3.3%	1.0%
$d = 113$	**test auc**	0.890	0.951	0.969	0.984	0.986	0.978	0.989
	train auc	0.890	0.942	0.978	0.984	0.984	0.983	0.990
rearrest	**test cal**	7.3%	24.2%	21.8%	5.2%	1.4%	3.8%	2.4%
$n = 22530$	**train cal**	4.8%	24.3%	14.1%	1.1%	1.1%	3.7%	2.6%
$d = 48$	**test auc**	0.555	0.692	0.698	0.676	0.676	0.677	0.699
	train auc	0.640	0.700	0.698	0.676	0.676	0.682	0.701
spambase	**test cal**	15.0%	29.5%	33.4%	26.5%	16.3%	17.9%	11.7%
$n = 4601$	**train cal**	18.7%	26.8%	23.8%	25.1%	14.6%	19.3%	12.3%
$d = 57$	**test auc**	0.620	0.875	0.861	0.910	0.913	0.908	0.928
	train auc	0.772	0.872	0.876	0.917	0.921	0.926	0.935
telemarketing	**test cal**	2.6%	11.2%	6.2%	1.9%	1.3%	1.3%	1.3%
$n = 41188$	**train cal**	0.7%	11.3%	4.9%	1.7%	1.1%	1.1%	1.1%
$d = 57$	**test auc**	0.574	0.700	0.715	0.759	0.760	0.760	0.760
	train auc	0.500	0.685	0.685	0.759	0.760	0.760	0.760

Table 7: Summary statistics of risk scores with integer coefficients $\lambda_j \in \{-5, \ldots, 5\}$ with model size of $\|\boldsymbol{\lambda}\|_0 \leq 5$. Here: *test cal* and *test auc*, which are the 5-CV mean test CAL / AUC; *train cal* and *train auc* which are the CAL and AUC of the final model fit using the entire dataset.

Appendix G. Supporting Material for Seizure Prediction

In this appendix, we provide supporting material for the seizure prediction application in Section 6.

G.1. List of Input Variables

In Table 8, we list all input variables in the training dataset.

Input Variable	Values	Sign
Male	$\{0,1\}$	
Female	$\{0,1\}$	
PriorSeizure	$\{0,1\}$	+
PosteriorDominantRhythmPresent	$\{0,1\}$	−
BriefRhythmicDischarge	$\{0,1\}$	+
NoReactivityToStimulation	$\{0,1\}$	
EpileptiformDischarges	$\{0,1\}$	+
SecondaryDXIncludesMentalStatusFirst	$\{0,1\}$	
SecondaryDXIncludesCNSInfection	$\{0,1\}$	
SecondaryDXIncludesCNSInflammatoryDisease	$\{0,1\}$	
SecondaryDXIncludesCNSNeoplasm	$\{0,1\}$	
SecondaryDXIncludesHypoxisIschemicEncephalopathy	$\{0,1\}$	
SecondaryDXIncludesIntracerebralHemorrhage	$\{0,1\}$	
SecondaryDXIncludesIntraventricularHemorrhage	$\{0,1\}$	
SecondaryDXIncludesMetabolicEncephalopathy	$\{0,1\}$	
SecondaryDXIncludesIschemicStroke	$\{0,1\}$	
SecondaryDXIncludesSubarachnoidHemmorage	$\{0,1\}$	
SecondaryDXIncludesSubduralHematoma	$\{0,1\}$	
SecondaryDXIncludesTraumaticBrainInjury	$\{0,1\}$	
SecondaryDXIncludesHydrocephalus	$\{0,1\}$	
PatternIsStimulusInducedAny	$\{0,1\}$	
PatternIsStimulusInducedBiPD	$\{0,1\}$	
PatternIsStimulusInducedGPD	$\{0,1\}$	
PatternIsStimulusInducedGRDA	$\{0,1\}$	
PatternIsStimulusInducedLPD	$\{0,1\}$	
PatternIsStimulusInducedLRDA	$\{0,1\}$	
PatternIsSuperImposedAny	$\{0,1\}$	+
PatternIsSuperImposedBiPD	$\{0,1\}$	+
PatternIsSuperImposedGPD	$\{0,1\}$	+
PatternIsSuperImposedGRDA	$\{0,1\}$	+
PatternIsSuperImposedLPD	$\{0,1\}$	+
PatternIsSuperImposedLRDA	$\{0,1\}$	+
PatternsInclude_BiPD	$\{0,1\}$	+
PatternsInclude_GPD	$\{0,1\}$	+
PatternsInclude_GRDA	$\{0,1\}$	+
PatternsInclude_LPD	$\{0,1\}$	+
PatternsInclude_LRDA	$\{0,1\}$	+
PatternsInclude_GRDA_or_GPD	$\{0,1\}$	+
PatternsInclude_BiPD_or_LRDA_or_LPD	$\{0,1\}$	+
MaxFrequencyAnyPattern	$\{0.0, 0.5, \ldots, 3.0\}$	+
MaxFrequencyBiPD	$\{0.0, 0.5, \ldots, 3.0\}$	+
MaxFrequencyGPD	$\{0.0, 0.5, \ldots, 3.0\}$	+
MaxFrequencyLPD	$\{0.0, 0.5, \ldots, 3.0\}$	+
MaxFrequencyLRDA	$\{0.0, 0.5, \ldots, 3.0\}$	+

Input Variable	Values	Sign
MaxFrequencyAnyPattern = 0.0Hz	$\{0,1\}$	
MaxFrequencyAnyPattern ≥ 0.5Hz	$\{0,1\}$	+
MaxFrequencyAnyPattern ≥ 1.0Hz	$\{0,1\}$	+
MaxFrequencyAnyPattern ≥ 1.5Hz	$\{0,1\}$	+
MaxFrequencyAnyPattern ≥ 2.0Hz	$\{0,1\}$	+
MaxFrequencyAnyPattern ≥ 2.5Hz	$\{0,1\}$	+
MaxFrequencyAnyPattern ≥ 3.0Hz	$\{0,1\}$	+
MaxFrequencyBiPD = 0.0	$\{0,1\}$	
MaxFrequencyBiPD ≥ 0.5Hz	$\{0,1\}$	+
MaxFrequencyBiPD ≥ 1.0Hz	$\{0,1\}$	+
MaxFrequencyBiPD ≥ 1.5Hz	$\{0,1\}$	+
MaxFrequencyBiPD ≥ 2.0Hz	$\{0,1\}$	+
MaxFrequencyBiPD ≥ 2.5Hz	$\{0,1\}$	+
MaxFrequencyBiPD ≥ 3.0Hz	$\{0,1\}$	+
MaxFrequencyGPD = 0.0	$\{0,1\}$	
MaxFrequencyGPD ≥ 0.5Hz	$\{0,1\}$	+
MaxFrequencyGPD ≥ 1.0Hz	$\{0,1\}$	+
MaxFrequencyGPD ≥ 1.5Hz	$\{0,1\}$	+
MaxFrequencyGPD ≥ 2.0Hz	$\{0,1\}$	+
MaxFrequencyGPD ≥ 2.5Hz	$\{0,1\}$	+
MaxFrequencyGPD ≥ 3.0Hz	$\{0,1\}$	+
MaxFrequencyGRDA = 0.0	$\{0,1\}$	
MaxFrequencyGRDA ≥ 0.5Hz	$\{0,1\}$	+
MaxFrequencyGRDA ≥ 1.0Hz	$\{0,1\}$	+
MaxFrequencyGRDA ≥ 1.5Hz	$\{0,1\}$	+
MaxFrequencyGRDA ≥ 2.0Hz	$\{0,1\}$	+
MaxFrequencyGRDA ≥ 2.5Hz	$\{0,1\}$	+
MaxFrequencyGRDA ≥ 3.0Hz	$\{0,1\}$	+
MaxFrequencyLPD = 0.0	$\{0,1\}$	
MaxFrequencyLPD ≥ 0.5Hz	$\{0,1\}$	+
MaxFrequencyLPD ≥ 1.0Hz	$\{0,1\}$	+
MaxFrequencyLPD ≥ 1.5Hz	$\{0,1\}$	+
MaxFrequencyLPD ≥ 2.0Hz	$\{0,1\}$	+
MaxFrequencyLPD ≥ 2.5Hz	$\{0,1\}$	+
MaxFrequencyLPD ≥ 3.0Hz	$\{0,1\}$	+
MaxFrequencyLRDA = 0.0	$\{0,1\}$	
MaxFrequencyLRDA ≥ 0.5Hz	$\{0,1\}$	+
MaxFrequencyLRDA ≥ 1.0Hz	$\{0,1\}$	+
MaxFrequencyLRDA ≥ 1.5Hz	$\{0,1\}$	+
MaxFrequencyLRDA ≥ 2.0Hz	$\{0,1\}$	+
MaxFrequencyLRDA ≥ 2.5Hz	$\{0,1\}$	+
MaxFrequencyLRDA ≥ 3.0Hz	$\{0,1\}$	+

Table 8: Names, values, and sign constraints for input variables in the `seizure` dataset.

G.2. List of Operational Constraints

No Redundant Categorical Variables

1. Use either *Male* or *Female*.
2. Use either *PatternsInclude_GRDA_or_GPD* or any one of (*PatternsInclude_GRDA*, *PatternsInclude_GPD*).
3. Use either *PatternsInclude_BiPD_or_LRDA_or_LPD* or any one of (*PatternsInclude_BiPD*, *PatternsInclude_LRDA*, *PatternsInclude_LPD*).
4. Use either *MaxFrequencyAnyPattern*$=$ 0.0 or *MaxFrequencyAnyPattern*$\geq$ 0.5 or neither.
5. Use either *MaxFrequencyLPD*$=$ 0.0 or *MaxFrequencyLPD*$\geq$ 0.5 or neither.
6. Use either *MaxFrequencyGPD*$=$ 0.0 or *MaxFrequencyGPD*$\geq$ 0.5 or neither.
7. Use either *MaxFrequencyGRDA* $=$ 0.0 or *MaxFrequencyGRDA* $\geq$ 0.5 or neither.
8. Use either *MaxFrequencyBiPD*$=$ 0.0 or *MaxFrequencyBiPD*$\geq$ 0.5 or neither.
9. Use either *MaxFrequencyLRDA*$=$ 0.0 or *MaxFrequencyLRDA*$\geq$ 0.5 or neither.

Frequency in Continuous Encoding or Thresholded Encoding

10. Choose between *MaxFrequencyAnyPattern* or (*MaxFrequencyAnyPattern*$=$ 0.0 $\ldots$ *MaxFrequencyAnyPattern*$\geq$ 3.0).
11. Choose between *MaxFrequencyGPD* or (*MaxFrequencyGPD*$=$ 0.0 $\ldots$ *MaxFrequencyGPD*$\geq$ 3.0).
12. Choose between *MaxFrequencyLPD* or (*MaxFrequencyLPD*$=$ 0.0 $\ldots$ *MaxFrequencyLPD*$\geq$ 3.0).
13. Choose between *MaxFrequencyGRDA* or (*MaxFrequencyGRDA*$=$ 0.0 $\ldots$ *MaxFrequencyGRDA*$\geq$ 3.0).
14. Choose between *MaxFrequencyBiPD* or (*MaxFrequencyBiPD*$=$ 0.0 $\ldots$ *MaxFrequencyBiPD*$\geq$ 3.0).
15. Choose between *MaxFrequencyLRDA* or (*MaxFrequencyLRDA*$=$ 0.0 $\ldots$ *MaxFrequencyLRDA*$\geq$ 3.0).

Limited # of Thresholds for Thresholded Variables

16. Use at most 2 of: *MaxFrequencyAnyPattern*$=$ 0.0, *MaxFrequencyAnyPattern*$\geq$ 0.5 $\ldots$ *MaxFrequencyAnyPattern*$\geq$ 3.0.
17. Use at most 2 of: *MaxFrequencyLPD*$=$ 0.0, *MaxFrequencyLPD*$\geq$ 0.5$\ldots$ *MaxFrequencyLPD*$\geq$ 3.0.
18. Use at most 2 of: *MaxFrequencyGPD*$=$ 0.0, *MaxFrequencyGPD*$\geq$ 0.5$\ldots$ *MaxFrequencyGPD*$\geq$ 3.0.

19. Use at most 2 of: $MaxFrequencyGRDA = 0.0$, $MaxFrequencyGRDA \geq 0.5 \ldots MaxFrequencyGRDA \geq 3.0$.
20. Use at most 2 of: $MaxFrequencyBiPD = 0.0$, $MaxFrequencyBiPD \geq 0.5 \ldots MaxFrequencyBiPD \geq 3.0$.
21. Use at most 2 of: $MaxFrequencyLRDA = 0.0$, $MaxFrequencyLRDA \geq 0.5 \ldots MaxFrequencyLRDA \geq 3.0$.

Any cEEG Pattern or Specific cEEG Patterns

22. Use either *PatternIsStimulusInducedAny* or any of (*PatternIsStimulusInducedBiPD*, *PatternIsStimulusInducedGRDA*, *PatternIsStimulusInducedGPD*, *PatternIsStimulusInducedLPD*, *PatternIsStimulusInducedLRDA*) or none of the variables.
23. Use either *PatternIsSuperImposed* or any of (*PatternIsSuperImposedBiPD*, *PatternIsSuperImposedGPD PatternIsSuperImposedGRDA*, *PatternIsSuperImposedLPD*, *PatternIsSuperImposedLRDA*), or none of the variables.
24. Use either *MaxFrequencyAnyPattern* (or its thresholded versions) or any of *MaxFrequencyBiPD*, *MaxFrequencyGRDA*, *MaxFrequencyGPD*, *MaxFrequencyLPD*, *MaxFrequencyLRDA*, (or their thresholded versions), or none of the variables.

G.3. Additional Experimental Results

	Training Requirements		% of Instances That Satisfy Constraints on			
Method	**# Instances**	**# Models**	**Monotonicity**	**Model Size**	**Operational**	**All Constraints**
RiskSLIM	1	6	100%	100%	100%	100%
PooledRd	1,100	33,000	100%	22%	20%	20%
PooledRd*	1,100	33,000	100%	22%	20%	20%
PooledRsRd	1,100	33,000	100%	9%	5%	2%
PooledRsRd*	1,100	33,000	23%	9%	5%	5%
PooledSeqRd	1,100	33,000	100%	10%	7%	4%
PooledSeqRd*	1,100	33,000	98%	10%	8%	4%

Table 9: Training requirements and constraint violations for the methods in Table 4. Each instance is a unique combination of free parameters. # models represents the total number of models that we must train to (1) choose parameters of the final risk score *and* (2) pair this model an unbiased estimate of performance. We need to train 33K for other methods since they require nested cross validation.

Journal of Machine Learning Research 20 (2019) 1-49 Submitted 9/18; Revised 4/19; Published 7/19

Nonparametric Estimation of Probability Density Functions of Random Persistence Diagrams

Vasileios Maroulas VMAROULA@UTK.EDU
Department of Mathematics
University of Tennessee
Knoxville, TN 37996, USA

Joshua L Mike MIKEJOSH@MSU.EDU
Computational Mathematics, Science, and Engineering Department
Michigan State University
East Lansing, MI 48823, USA

Christopher Oballe COBALLE@VOLS.UTK.EDU
Department of Mathematics
University of Tennessee
Knoxville, TN 37996, USA

Editor: Boaz Nadler

Abstract

Topological data analysis refers to a broad set of techniques that are used to make inferences about the shape of data. A popular topological summary is the persistence diagram. Through the language of random sets, we describe a notion of global probability density function for persistence diagrams that fully characterizes their behavior and in part provides a noise likelihood model. Our approach encapsulates the number of topological features and considers the appearance or disappearance of those near the diagonal in a stable fashion. In particular, the structure of our kernel individually tracks long persistence features, while considering those near the diagonal as a collective unit. The choice to describe short persistence features as a group reduces computation time while simultaneously retaining accuracy. Indeed, we prove that the associated kernel density estimate converges to the true distribution as the number of persistence diagrams increases and the bandwidth shrinks accordingly. We also establish the convergence of the mean absolute deviation estimate, defined according to the bottleneck metric. Lastly, examples of kernel density estimation are presented for typical underlying datasets as well as for virtual electroencephalographic data related to cognition.

Keywords: Topological Data Analysis; Persistence Homology; Finite Set Statistics; Global Distribution of Persistence Diagrams; Kernel Density Estimation; EEG Signals

1. Introduction

Topological data analysis (TDA) encapsulates a range of data analysis methods that investigate the topological structure of a dataset (Edelsbrunner and Harer, 2010). One such method, persistent homology, describes the geometric structure of a given dataset and summarizes this information as a persistence diagram. TDA, and in particular persistence diagrams, have been employed in several studies with topics ranging from classification and clustering (Venkataraman et al., 2016; Adcock et al., 2016; Pereira and de Mello, 2015; Marchese and Maroulas, 2018) to the analysis of dynamical

©2019 Vasileios Maroulas, Joshua L Mike, and Christopher Oballe.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-618.html.

systems (Perea and Harer, 2015; Sgouralis et al., 2017; Guillemard and Iske, 2011; Seversky et al., 2016) and complex systems such as sensor networks (De Silva and Ghrist, 2007; Xia et al., 2015; Bendich et al., 2016). In this work, we establish the probability density function (pdf) for a random persistence diagram.

Persistence diagrams offer a topological summary for a collection of $\mathbb{d}$-dimensional data, say $\{x_i\} \subset \mathbb{R}^{\mathbb{d}}$, which focuses on the global geometric structure of the data. A persistence diagram is a multiset of homological features $\{(b_i, d_i, k_i)\}$, each representing a k_i-dimensional hole which appears at scale $b_i \in \mathbb{R}^+$ and is filled at scale $d_i \in (b_i, \infty)$. In general, the dataset arises from any metric space, though restricting to $\{x_i\} \subset \mathbb{R}^{\mathbb{d}}$ guarantees $k_i \in \{0, ..., \mathbb{d}-1\}$. For example, if the data form a time series trajectory $x_i = f(t_i)$, the associated persistence diagram describes multistability through a corresponding number of persistent 0-dimensional features or periodicity through a single persistent 1-dimensional feature. In a typical persistence diagram, few features exhibit long persistence (range of scales $d_i - b_i$), and such features describe important topological characteristics of the underlying dataset. Moreover, persistent features are stable under perturbation of the underlying dataset (Cohen-Steiner et al., 2010).

Persistence diagrams have recently seen intense active research, including significant successful efforts toward facilitating previously challenging computations with them; these efforts impact evaluation of Wasserstein distance in (Kerber et al., 2016) and the creation of persistence diagrams with packages such as Dionysus (Fasy et al., 2015) and Ripser (Bauer, 2015) which take advantage of certain properties of simplicial complexes (Chen and Kerber, 2011). Recently, various approaches have defined specific summary statistics such as center and variance (Bobrowski et al., 2014; Mileyko et al., 2011; Turner et al., 2014; Marchese and Maroulas, 2018), birth and death estimates (Emmett et al., 2014), and confidence sets (Fasy et al., 2014). While the aforementioned studies focus on valuable specific summaries, here we view a distribution of persistence diagrams in a global sense via a nonparametric method to estimate its density function.

We naturally think of a (random) persistence diagram as a random element which depends upon a stochastic procedure which is used to generate the underlying dataset that it summarizes. Given that geometric complexes are the typical paradigms for application of persistent homology to data analysis, see for example the partial list (De Silva and Ghrist, 2007; Emmett et al., 2014; Guillemard and Iske, 2011; Marchese and Maroulas, 2016; Perea and Harer, 2015; Seversky et al., 2016; Xia et al., 2015; Venkataraman et al., 2016; Edelsbrunner, 2013; Emrani et al., 2014)), we consider persistence diagrams which arise from a dataset and its associated Čech filtration. Thus, sample datasets yield sample persistence diagrams without direct access to the distribution of persistence diagrams. In this sense, a distribution of persistence diagrams is defined by transforming the distribution of underlying data under the process used to create a persistence diagram, as discussed in (Mileyko et al., 2011). The diagrams are created through a highly nonlinear process which relies on the global arrangement of datapoints (see Section 2); thus, the structure of a persistence diagram distribution remains unclear even for underlying data with a well-understood distribution. Indeed, results for the persistent homology of noise alone, such as (Adler et al., 2014), primarily concern the asymptotics of feature cardinality at coarse scale. More recently, it was proved that under mild conditions there exists a density with respect to Lebesgue measure for the intensity of persistence diagrams induced by filter functions defined on random variables having support on manifolds; in addition it was shown that kernel density estimation of this intensity is possible through persistence surfaces (Chazal and Divol, 2018). Another approach is to study these distributions through non-

parametric means. Kernel density estimation is a well known nonparametric technique for random vectors in $\mathbb{R}^{\mathrm{d}}$ (Scott, 2015) which relies on convolution with a smooth kernel.

There has been extensive work to devise various maps from persistence diagrams into Hilbert spaces, especially Reproducing Kernel Hilbert Spaces (RKHS). For example, (Adams et al., 2017) discretizes persistence diagrams via bins, yielding vectors in a high dimensional Euclidean space; the authors in (Chazal and Divol, 2018) introduce a novel bandwidth selection procedure for this particular vectorization based on cross-validation. The persistence landscapes of (Bubenik, 2015) reinterpret the space of persistence diagrams within a function-based vector space. The works (Reininghaus et al., 2014) and (Kusano et al., 2016) define kernels between persistence diagrams in a RKHS. By mapping into a Hilbert space, these studies allow the application of machine learning methods such as principal component analysis, random forest, support vector machine, and more. The universality of such a kernel is investigated in (Kwitt et al., 2015); this property induces a metric on distributions of persistence diagrams (by comparing means in the RKHS), as (Kwitt et al., 2015) demonstrates with a two-sample hypothesis test. In a similar vein, (Adler et al., 2017) uses Gibbs distributions in order to replicate similar persistence diagrams, e.g. for use in MCMC type sampling.

By mapping into a Hilbert space, the preceding approaches kernelize persistence diagrams for express application in typical machine learning methodology. In a similar vein, the studies (Bobrowski et al., 2014) and (Fasy et al., 2014) work with kernel density estimation on the underlying data to estimate a target diagram as the number of underlying datapoints goes to infinity. In both cases, the target diagram is directly associated to the probability density function (pdf) of the underlying data via the superlevel sets of the pdf. The first work constructs an estimator for the target diagram, while the second defines a confidence set. In either case, kernel density estimation is used to approximate the pdf of the underlying datapoints, assuming the data are independent and identically distributed (i.i.d.). In complementary fashion, our work considers a new kernel density defined on a completely different space: the space of persistence diagrams.

Since persistence diagrams lack a vector space structure, we must smooth without convolution. Instead, we treat persistence diagrams as random multisets and provide a noise model for persistence diagrams which ascribes a great level of uncertainty near the diagonal. The theory of random sets is an interpretation of point processes which provides the tools necessary to describe the global pdf of the noise model as a kernel density centered at a particular persistence diagram. Instead of a transformed collection or a center diagram, the output of our method is an estimate of a probability density function (pdf) of a random persistence diagram. Access to a persistence diagram pdf facilitates definition and application of new statistical techniques in this context such as hypothesis testing, utilization of Bayesian priors, or likelihood based methods, e.g. see (Maroulas et al., 2019). The proposed kernel density is centered at a persistence diagram and describes each feature as having either short or long persistence; by treating each long-persistence point individually and short persistence points collectively, the kernel density strikes a careful balance between accuracy and computation time. Our method also enables expedient sampling of new persistence diagrams from the kernel density estimate. In contrast to previous methodologies, our kernel density estimate has the potential to describe high probability features in a random persistence diagram, even if these features have *brief* persistence. Such features are typically indicative of the geometric structure, e.g., curvature, of the dataset rather than its topology.

The homological features (b_i, d_i, k_i) in a persistence diagram come without an ordering and their cardinality is variable, being bounded but not defined by the cardinality of the underlying

dataset. Thus, any notion of density must be (i) invariant to the ordering of features and (ii) account for variability in their cardinality. Indeed, the approach used to analyze a collection of persistence diagrams in (Bendich et al., 2016) is a good step toward understanding a random persistence diagram, but requires a choice of order and considers only a fixed number of features and is therefore unsuitable for creating probability densities. In this work, we offer a kernel density with the desirable properties (i) and (ii), which also calls attention to the persistence of each feature. A typical persistence diagram has many features with brief persistence and few with moderate or longer persistence; consequently, our kernel density groups features with short persistence together in order to combat the curse of dimensionality. Indeed, the kernel density still considers features of short persistence, but simplifies their treatment in order to facilitate computation. The kernel density is defined on a pertinent space of finite random sets which is equipped to describe pdfs for random persistence diagrams generated from associated data with bounded cardinality of topological features. In this sense, our kernel density provides estimation of the distribution of persistence diagrams which in turn describes the geometry of the random underlying dataset. The requirement of bounded feature cardinality is trivially satisfied for datasets with bounded cardinality, which is reasonable for application and theory. Indeed, the creation of a persistence diagram from an infinite collection of data is often nonsensical (e.g., for anything with unbounded noise), and a scaling limit should be considered instead; For example, this problem can be approached via the studies (Bobrowski et al., 2014) and (Fasy et al., 2014).

The overall *contribution* of this article is listed below.

1. A probabilistic framework for defining distributions of random persistence diagrams.

2. A novel kernel density estimator centered at persistence diagrams, which in part provides a noise likelihood model, and its convergence to the true distribution as the number of persistence diagrams goes to infinity.

3. A new dispersion statistic for persistence diagrams, the mean absolute bottleneck deviation (MAD) and the convergence of the sample MAD computed with our kernel density estimator.

4. Applications of the kernel density estimator to analyze data data arising from a neuroscience problem related to cognition.

Organization: We establish the kernel density estimation problem through the lens of finite set statistics and we consequently begin with relevant background in topological data analysis in Section 2. The reader may refer to (Edelsbrunner and Harer, 2010) for a more rigorous treatment of the subject. Section 3 contains our framework for random persistence diagrams and their probability distributions. Our main theoretical results are in Section 4. In Subsection 4.1, we construct the kernel density associated to a center persistence diagram and kernel bandwidth parameter. This consists of decomposing the center persistence diagram into lower and upper halves, finding pdfs associated to each half, and lastly determining the pdf for their union. Convergence of this estimator to the true distribution of random persistence diagrams is proved in Subsection 4.2. In Subsection 4.3, we introduce a new measure of dispersion for persistence diagrams, the MAD, and show that the sample MAD computed with our kernel converges (with mild assumptions). Next, Section 5 contains practical examples of our kernel density. Namely, an example of persistence diagram kernel density estimation and its convergence are demonstrated for persistence diagrams associated to underlying data with annular distribution and we apply our kernel to a neuroscience

problem (Example 4). Finally, we end with conclusions and discussion in Section 6. Further examples of KDE convergence and the proofs of auxiliary propositions and lemmas are given in the supplementary materials.

2. Topological Data Analysis Background

The topological background discussed here builds toward the definition of persistence diagrams, the pertinent objects in this work. We begin by briefly discussing simplicial complexes and homology, an algebraic descriptor for coarse shape in topological spaces. In turn, persistent homology, and its summary, persistence diagrams, are techniques for bringing the power and convenience of homology to describe subspace filtrations of topological spaces. The reader should refer to (Edelsbrunner and Harer, 2010), for example, for a rigorous treatment of persistent homology. We first consider topological spaces of discernible dimension, called manifolds.

Definition 1 *A topological space X is called a k-dimensional manifold if every point $x \in X$ has a neighborhood which is homeomorphic to an open neighborhood in k-dimensional Euclidean space.*

We generalize the fixed-dimension notion of a manifold in order to define simplicial homology for simplicial complexes. We then discuss the Čech construction which is used to associate simplicial complexes to datasets.

Definition 2 *A k-simplex is a collection of $k+1$ linearly independent vertices along with all convex combinations of these vertices: $(v_0, ..., v_k) = \left\{\sum_{i=0}^{k} \alpha_i v_i : \sum_{i=0}^{k} \alpha_i = 1 \text{ and } \alpha_i \geq 0 \, \forall i\right\}$. Topologically, a k-simplex is treated as a k-dimensional manifold (with boundary). An oriented simplex is typically described by a list of its vertices, such as (v_0, v_1, v_2). The faces of a simplex consist of all the simplices built from a subset of its vertex set; for example, the edge (v_1, v_2) and vertex (v_2) are both faces of the triangle (v_0, v_1, v_2).*

Definition 3 *A simplicial complex $\mathcal{K}$ is a collection of simplices wherein (i) if $\sigma \in \mathcal{K}$, then all its faces are also in $\mathcal{K}$, and (ii) the intersection of any pair of simplices in $\mathcal{K}$ is another simplex in $\mathcal{K}$.*

A simplicial complex is realized by the union of all its simplices; some examples are shown in Fig. 1. Conditions (i) and (ii) in Def. 3 establish a unique topology on the realization of a simplicial complex which restricts to the subspace topology on each open simplex. For finite simplicial complexes realized in $\mathbb{R}^{\mathrm{d}}$, this topology is also consistent with the Euclidean subspace topology.

Given a simplicial complex, we are interested in describing its global topology and local geometry through homological features. For our purposes, it suffices to define $k-$dimensional homological features of simplicial complexes as $k-$dimensional holes, so that, for example, $0-$dimensional homological features are connected components, $1-$dimensional homological features are loops, and $2-$dimensional homological features are voids.

We wish to extend the notion of homology for a discrete set of data $\boldsymbol{x} = \{x_i\}_{i=1}^{N}$ within a metric space (X, d_X). Treating the set itself as a simplicial complex, its homology yields only the cardinality of the data points. So, we use the metric to obtain more information. Here we denote by $B(x_0, r_0)$ a metric ball centered at x_0 of radius r_0. Fix a radius $r > 0$ and consider the collection of neighborhoods $U = \{U_i\} = \{B(x_i, r)\}$ along with its union $\mathcal{U}_r = \cup_i B(x_i, r)$. The filtration of sets $\{\mathcal{U}_r\}_{r \in \mathbb{R}^+}$ naturally yields information about the arrangement within X of the dataset $\boldsymbol{x}$ at various

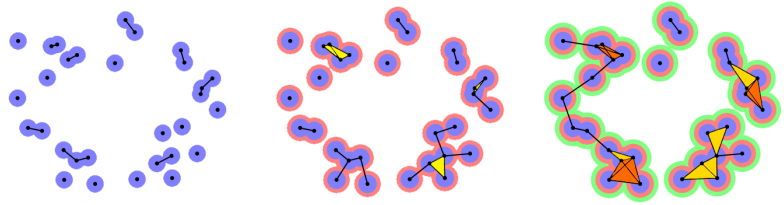


Figure 1: The neighborhood space and Čech complex of matching radius plotted at three different radii. Yellow indicates a triangle while orange indicates a tetrahedron. This family of simplicial complexes is the filtration used to compute and define persistent homology.

scales. To make homology computations more tractable for $\mathcal{U}_r$, we instead consider the associated nerve complexes.

Definition 4 *The nerve $\mathcal{N}(U)$ of a collection of open sets U is the simplicial complex where a k-simplex $(v_{i_0}, ..., v_{i_k})$ is in $\mathcal{N}(U)$ if and only if $\cap_{j=0}^{k} U_{i_j} \neq \emptyset$. The nerve of the neighborhoods $U = \{B(x_i, r)\}$ is called the Čech complex on the data $\{x_i\}$ at radius r and is denoted by Čech$(\boldsymbol{x}, r)$.*

Examples of the Čech complex for the same data at different radii are depicted in Fig. 1, where they are superimposed with the associated neighborhood space. Any nerve complex trivially satisfies the requirements for a simplicial complex (Edelsbrunner and Harer, 2010). Moreover, the nerve theorem states that the nerve and union of a collection of convex sets have similar topology (they are homotopy equivalent) (Hatcher, 2002); specifically, the Čech complex and neighborhood space $\mathcal{U}$ have identical homology for any given radius.

A priori, it is unclear which choice of scale (radius), best describes the data; and oftentimes different scales reveal different information. Thus, to investigate the topology of our data, we consider the appearance and disappearance of homological features at growing scale. This multiscale viewpoint, called persistent homology, is introduced in (Edelsbrunner et al., 2002) and yields a topological summary of the data called a persistence diagram. This is possible because we have a growing filtration of complexes, so each complex is included in the next (see Fig. 1). These inclusion maps induce maps at the level of homology groups. These induced maps are referred to here as the persistence maps, and take features to features or to zero (Cohen-Steiner et al., 2007). Thus, each feature is tracked by how far the persistence maps preserve it. In turn, tracking features is boiled down to a very specific algorithm for obtaining the birth and death radii for each homological feature (e.g., see (Edelsbrunner and Harer, 2010)). Features which persist over a large range of scale are typically considered more important, and their presence is stable under small perturbations of the underlying data (Cohen-Steiner et al., 2010).

Persistent homology yields a multiset of homological features, each born at a scale b_i, lasting until its death scale d_i, with degree of homology k_i; in short, it yields a persistence diagram $\mathscr{D} = \{\xi_i\}_{i=1}^{M} = \{(b_i, d_i, k_i)\}_{i=1}^{M}$. We interpret the birth-death values as coordinate points with degree of homology as labels. For clarity and simplicity, we ignore any features with death value $d_i = \infty$, since these features are generally a characteristic of the ambient space. In particular, one homological feature with $(b, d, k) = (0, \infty, 0)$ is expected from any Čech filtration.

Specifically, for data in $\mathbb{R}^{\mathbb{d}}$, we consider each feature as an element of

$$\mathcal{W}_{0:\mathbb{d}-1} = W \times \{0, ..., \mathbb{d}-1\}, \tag{2.1}$$

where $W = \left\{(b,d) \in \mathbb{R}^2 : d > b \geq 0\right\}$ is the infinite wedge. As a topological space, the $\mathbb{d}$-fold multiwedge $\mathcal{W}_{0:\mathbb{d}-1}$ is treated as $\mathbb{d}$-disconnected copies of W, where W has the Euclidean metric and topology.

It is desirable to define a metric between persistence diagrams with which to measure topological similarity. In TDA, Hausdorff distance is typically used to compare underlying datasets, while the bottleneck distance (Def. 5) is used to compare their associated persistence diagrams (Fasy et al., 2014; Munch, 2017). A distance that accounts for cardinality differences between persistence diagrams was introduced in (Marchese and Maroulas, 2018) and its stability with respect to perturbations in the underlying point cloud was proved in (Maroulas et al., 2018).

Definition 5 *The bottleneck distance between two persistence diagrams D_1 and D_2 is given by*

$$W_\infty(D_1, D_2) = \min_{\gamma} \max_{x \in D_1} \|x - \gamma(x)\|_\infty . \tag{2.2}$$

where γ ranges over all possible bijections between D_1 and D_2 which match in degree of homology. The diagonal $\{b = d\}$ is included in both persistence diagrams with infinite multiplicity so that any feature may be matched to the diagonal.

Remark 6 *Due to the unstable presence of features near the diagonal, typical metrics on persistence diagrams such as the bottleneck distance treat the diagonal as part of every persistence diagram (Mileyko et al., 2011) in order to achieve stability with respect to Hausdorff perturbations of the underlying dataset (Cohen-Steiner et al., 2007). Morally, one considers the diagonal as representing vacuous features which are born and die simultaneously. For convenient computation, the definition of bottleneck distance can be applied to each degree of homology separately.*

3. Random Persistence Diagrams

In this section we establish background to make the notion of probability density for a random persistence diagram explicit and well-defined. A persistence diagram changes its feature cardinality under small perturbation of the underlying dataset, and these features have no intrinsic order. Consequently, we cannot treat persistence diagrams as elements of a vector space. Instead, we consider a random persistence diagram D as a random multiset of features $D = \{\xi_i\} \subset \mathcal{W}_{0:\mathbb{d}-1}$ in the multiwedge defined in Eq. (2.1). For underlying datasets sampled from $\mathbb{R}^{\mathbb{d}}$ with bounded cardinality, the affiliated Čech persistence diagrams also have bounded feature cardinality and degree of homology. Thus, we assume that the cardinality of a random persistence diagram is bounded above by some value $|D| \leq M \in \mathbb{N}$, and so consider the space $\mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathbb{d}-1}) = \{D \text{ multiset in } \mathcal{W}_{0:\mathbb{d}-1} : |D| \leq M\}$. We view $\mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathbb{d}-1})$ through a list of functions h_N which each map the appropriate dimension of Euclidean space into its corresponding cardinality component, $\mathcal{C}_N(\mathcal{W}_{0:\mathbb{d}-1})$. This viewpoint facilitates the definition of probability densities.

Definition 7 *For each $N \in \{0, ..., M\}$, consider the space of N topological features, denoted $\mathcal{C}_N(\mathcal{W}_{0:\mathbb{d}-1}) = \{D \text{ multiset in } \mathcal{W}_{0:\mathbb{d}-1} : |D| = N\}$, and the associated map $h_N : \mathcal{W}^N_{0:\mathbb{d}-1} \to \mathcal{C}_N(\mathcal{W}_{0:\mathbb{d}-1})$ defined by*

$$h_N(\xi_1, ..., \xi_N) = \{\xi_1, ..., \xi_N\} . \tag{3.1}$$

The map h_N creates equivalence classes on $\mathcal{W}^N_{0:\mathbb{d}-1}$ according to the action of the permutations Π_N; specifically, $[Z] = [(\xi_1, ..., \xi_N)]_{h_N} = \left\{\left(\xi_{\pi(1)}, ..., \xi_{\pi(N)}\right) : \pi \in \Pi_N\right\}$ for each $Z = (\xi_1, ..., \xi_N) \in \mathcal{W}^N_{0:\mathbb{d}-1}$. These equivalence classes yield the space

$$\mathcal{W}^N_{0:\mathbb{d}-1}/\Pi_N = \left\{[\boldsymbol{\xi}]_{h_N} : \boldsymbol{\xi} \in \mathcal{W}^N_{0:\mathbb{d}-1}\right\}, \tag{3.2}$$

equipped with the quotient topology. The topology on $\mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathbb{d}-1})$ is defined so that each h_N lifts to a homeomorphism between $\mathcal{W}^N_{0:\mathbb{d}-1}/\Pi_N$ and $\mathcal{C}_N(\mathcal{W}_{0:\mathbb{d}-1})$, and we write $\mathcal{W}^N_{0:\mathbb{d}-1}/\Pi_N \cong \mathcal{C}_N(\mathcal{W}_{0:\mathbb{d}-1})$.

With a topology in hand, one can define probability measures on the associated Borel σ-algebra. Thus, we define a random persistence diagram D to be a random element distributed according to some probability measure on $\mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathbb{d}-1})$ for a fixed maximal cardinality $M \in \mathbb{N}$. We denote associated probabilities by $\mathbb{P}[\cdot]$ and expected values by $\mathbb{E}[\cdot]$. Since $\mathcal{W}^N_{0:\mathbb{d}-1}/\Pi_N \cong \mathcal{C}_N(\mathcal{W}_{0:\mathbb{d}-1})$, we work toward defining probability densities on the collection of Euclidean spaces $\cup_{N=0}^M \mathcal{W}^N_{0:\mathbb{d}-1}$.

Definition 8 *For a given random persistence diagram D and any Borel subset A of $\mathcal{W}_{0:\mathbb{d}-1}$, the belief function β_D is defined as*

$$\beta_D(A) = \mathbb{P}[D \subset A] . \tag{3.3}$$

Since A is a Borel subset of $\mathcal{W}_{0:\mathbb{d}-1}$, the collection $O_A = \{D \in \mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathbb{d}-1}) : D \subset A\}$ is the quotient of $\cup_{N=0}^M A^N \subset \cup_{N=0}^M \mathcal{W}^N_{0:\mathbb{d}-1}$ under h_N; moreover, A^N is clearly Borel in the Euclidean topology of $\cup_{N=0}^M \mathcal{W}^N_{0:\mathbb{d}-1}$. Therefore, since h_N induces a homeomorphism (see definition 7), O_A is a Borel subset of $\mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathbb{d}-1})$. The belief function of a random persistence diagram is similar to the joint cumulative distribution function for a random vector, in particular by yielding a probability density function through Radon-Nikodým type derivatives.

Definition 9 *Fix ϕ defined on Borel subsets of $\mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathbb{d}-1})$ into $\mathbb{R}$. For an element $\xi \in \mathcal{W}_{0:\mathbb{d}-1}$ or a multiset $Z \subset \mathcal{W}_{0:\mathbb{d}-1}$ with $Z = \{\xi_1, ..., \xi_N\}$, the set derivative (evaluated at the empty set $\emptyset$) is respectively given by*

$$\begin{aligned} \frac{\delta\phi}{\delta\xi}(\emptyset) &= \lim_{n\to\infty} \frac{\phi(B(\xi, 1/n))}{\lambda(B(\xi, 1/n))}, \\ \frac{\delta\phi}{\delta Z}(\emptyset) &= \frac{\delta^N \phi}{\delta\xi_1 ... \delta\xi_N} = \left[\frac{\delta}{\delta\xi_1} \cdots \frac{\delta}{\delta\xi_N}\phi\right](\emptyset), \end{aligned} \tag{3.4}$$

where $B(\xi, 1/n)$ are Euclidean balls and λ indicates Lebesgue measure on $\mathcal{W}_{0:\mathbb{d}-1}$.

Remark 10 *Def. 9 for set derivatives at the empty set closely mirrors the Radon-Nikodým derivative with respect to Lebesgue measure. The definition of a set derivative evaluated on a nonempty set is more involved, and is found in (Matheron, 1975). Here we are primarily concerned with evaluation at $\emptyset$, since this suffices for the definition of a probability density function. Also, note that set derivatives satisfy the product rule.*

Remark 11 *Restricting to a particular cardinality N, consider $\phi_N = \phi \circ h_N$, a function on Euclidean space which is invariant under the action of Π_N. The viewpoint of ϕ_N elucidates the relationship between set derivatives and Radon-Nikodým derivatives with respect to Lebesgue measure. This viewpoint also shows that the iterated derivative given in Eq. (3.4) is independent of order and thus is well-defined for a multiset Z.*

As with typical derivatives, there is a complementary set integration operation for set derivatives. Set derivatives (at $\emptyset$) are essentially Radon-Nikodým derivatives with order tied to cardinality, and so the corresponding set integral acts like Lebesgue integration summed over each cardinality.

Definition 12 *Consider a Borel subset A of $\mathcal{W}_{0:\mathrm{cl}-1}$ and a Borel subset O of $\mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathrm{cl}-1})$. For a set function $f : \mathcal{C}_{\leq M}(\mathcal{W}_{0:\mathrm{cl}-1}) \to \mathbb{R}$, its set integrals over A and O are respectively defined according to the following sums of Lebesgue integrals:*

$$\int_A f(Z)\delta Z = \sum_{N=0}^{M} \frac{1}{N!} \int_{A^N} f(h_N(\xi_1, ..., \xi_N)) d\xi_1 ... d\xi_N, \tag{3.5a}$$

$$\int_O f(Z)\delta Z = \sum_{N=0}^{M} \frac{1}{N!} \int_{h_N^{-1}(O)} f(h_N(\xi_1, ..., \xi_N)) d\xi_1 ... d\xi_N, \tag{3.5b}$$

where $Z = \{\xi_1, ..., \xi_N\} \subset \mathcal{W}_{0:\mathrm{cl}-1}$ is a persistence diagram.

Dividing by $N!$ in Eqs. (3.5a) and (3.5b) accounts for integrating over $\mathcal{W}^N_{0:\mathrm{cl}-1}$ instead of $\mathcal{W}^N_{0:\mathrm{cl}-1}/\Pi_N \cong \mathcal{C}_N(\mathcal{W}_{0:\mathrm{cl}-1})$. It has been shown that set derivatives and integrals are inverse operations (Matheron, 1975); specifically, the set derivative of a belief function yields a probability density for a random diagram D such that

$$\beta_D(A) = \int_A \frac{\delta \beta_D}{\delta Z}(\emptyset)\delta Z. \tag{3.6}$$

Indeed, $A^N = h_N^{-1}(\{D \subset A\})$ so that Eq. (3.5a) also holds as an integral over the set $O_A = \{D \in \mathcal{C}_{\leq M} : D \subset A\}$ in the sense of Eq. (3.5b).

Definition 13 *For a random persistence diagram D, a global probability density function (global pdf) $f_D : \cup_{N\in\mathbb{N}} \mathcal{W}^N_{0:\mathrm{cl}-1} \to \mathbb{R}$ must satisfy*

$$\sum_{\pi \in \Pi_N} f_D(\xi_{\pi(1)}, ..., \xi_{\pi(N)}) = \frac{\delta^N \beta_D}{\delta\xi_1 \cdot ... \cdot \delta\xi_N}(\emptyset). \tag{3.7}$$

and is described by its layered restrictions $f_N = f_D\big|_{\mathcal{W}^N_{0:\mathrm{cl}-1}} : \mathcal{W}^N_{0:\mathrm{cl}-1} \to \mathbb{R}$ for each N.

Remark 14 *It is necessary to make a distinction between local and global densities because the global pdf is not defined on a single Euclidean space, and is instead expressed as a collection of densities over a range of dimensions. Specifically, while each local density f_N (for input cardinality N) is defined on $\mathcal{W}^N_{0:\mathrm{cl}-1}$, the global pdf f_D is defined on $\cup_{N=1}^M \mathcal{W}^N_{0:\mathrm{cl}-1}$ and restricts to a local density on each input dimension. Each local density $f_N(Z) = f_D\big|_{\mathcal{W}^N_{0:\mathrm{cl}-1}}(Z)$ decomposes into the product of the conditional density $f_D(Z\big||Z| = N)$ and the cardinality probability $\mathbb{P}[|Z| = N]$ (this follows from Proposition 16). Thus, each local density does not integrate to one, but instead to the associated probability $\mathbb{P}[|Z| = N]$. Also, the global pdf is not a set function and does not require division by $N!$, leading to the following relation: $\int_{A^N} f_D(\xi_1, ..., \xi_N) d\xi_1 ... d\xi_N = \frac{1}{N!}\int_{A^N} \frac{\delta^N \beta_D}{\delta^N Z}(\emptyset) d\xi_1 ... d\xi_N$.*

Remark 15 *While the global pdf and its local constituents need not be symmetric with respect to Π_N, there is a unique choice of global pdf (up to sets of Lebesgue measure 0) which satisfies Eq. (3.7) and is symmetric under the action of Π_N. In this case, we safely abuse notation by denoting $f_D(\{\xi_1, ..., \xi_N\}) := N! f_D(\xi_1, ..., \xi_N)$ and often write $f_D(Z)$ and allow context to determine whether Z denotes a set or a vector.*

The following proposition is critical to determine the global pdf for (i) the union of independent singleton diagrams (i.e., $\left|D^j\right| \leq 1$), (ii) a randomly chosen cardinality, N, followed by N i.i.d. draws from a fixed distribution, and (iii) a random persistence diagram kernel density function. The proof of this proposition follows similar arguments to (Mahler, 1995) (Theorem 17, pp. 155–156).

Proposition 16 *Let D be a random persistence diagram with cardinality bounded by M and let $\beta_D(S) = \mathbb{P}(D \subset S)$ be the belief function for D. Then β_D expands as $\beta_D(S) = a_0 + \sum_{m=1}^M a_m q_m(S)$, where $a_m = \mathbb{P}(|D| = m)$ and $q_m(S) = \mathbb{P}[D \subset S\big||D| = m]$.*

Remark 17 *The decomposition in Proposition 16 is often applied as a first step toward finding the local density constituents of the global pdf. In particular, $f_N = f_D\big|_{\mathcal{W}^N_{0:\mathrm{cl}-1}} = 0$ for $N > M$.*

Lastly, we encounter a computationally convenient summary for a random persistence diagram called the probability hypothesis density (PHD). The integral of the PHD over a subset U in $\mathcal{W}_{0:\mathrm{cl}-1}$ gives the expected number of points in the region U; moreover, any other function on $\mathcal{W}_{0:\mathrm{cl}-1}$ with this property is a.e. equal to the PHD (Goodman et al., 2013).

Definition 18 *(Matheron, 1975) The probability hypothesis density (PHD) for a random persistence diagram D is defined as the set function $F_D(a) = \frac{\delta\beta_D}{\delta Z}(\{a\})$ and is expressed as a set integral as*

$$F_D(a) = \int_{\{Z:\{a\}\subset Z\}} \frac{\delta\beta}{\delta Z}(\emptyset)\delta Z. \tag{3.8}$$

In particular, $\mathbb{E}(|D \cap U|) = \int_U F_D(u)\, du$ for any region U.

Remark 19 *Def. 18 is equivalent to an intensity function of a point process. In general, the intensity function induced by a given global pdf may be undefinied, but under mild conditions Eq. (3.8) is finite (we discuss this further in Section 4.2). Since D is a random persistence diagram, the PHD is always defined as a distribution and can always be integrated to obtain the identity $\mathbb{E}(|D \cap U|) = \int_U F_D(u)\, du$ for any region U.*

Proposition 16 leads to the following lemma which is crucial for determining the kernel density. We refer to a random persistence diagram D with $|D| \leq 1$ as a singleton diagram, and such singletons are indexed by superscripts.

Lemma 20 *Consider a multiset of independent singleton random persistence diagrams $\{D^j\}_{j=1}^M$. If each singleton D^j is described by the value $q^{(j)} = \mathbb{P}[D^j \neq \emptyset]$ and the subsequent conditional pdf, $p^{(j)}(\xi)$, given $\left|D^j\right| = 1$, then the global pdf for $D = \cup_{j=1}^M D^j$ is given by*

$$f_D(\xi_1, ..., \xi_N) = \sum_{\gamma \in I(N,M)} \mathcal{Q}(\gamma) \prod_{k=1}^N p^{(\gamma(k))}(\xi_k), \tag{3.9}$$

for each $N \in \{0, ..., M\}$ where

$$\mathcal{Q}(\gamma) = \mathcal{Q}^*(\gamma) \prod_{k=1}^N q^{(\gamma(k))}, \tag{3.10}$$

$I(N,M)$ consists of all (strictly) increasing injections $\gamma : \{1,...,N\} \to \{1,...,M\}$, which enumerate (unordered) correspondences between the input features $(\xi_1,\ldots,\xi_N)$ and a subset of the M random singletons, and

$$\mathcal{Q}^*(\gamma) = \frac{\prod_{j=1}^{M}(1-q^{(j)})}{\prod_{k=1}^{N}(1-q^{(\gamma(k))})}. \tag{3.11}$$

Proof Since the singleton events D^j are independent, the belief function for $D = \cup_j D^j$ decomposes into $\beta_D(S) = \prod_{j=1}^{M}\beta_{D^j}(S)$. Next, we employ the product rule for the set derivative (see Def. 9) to obtain the global pdf for D in terms of the singleton belief functions and their first derivatives. Higher derivatives of β_{D^j} are zero since D^j are singletons (see Remark 17). Thus, the product rule yields first derivatives on all (ordered) subsets of the singleton belief functions:

$$\frac{\delta^N\beta_D}{\delta\xi_1...\delta\xi_N}(\emptyset) = \sum_{1\le j_1\neq,...,\neq j_N\le M} \frac{\beta_{D^1}(\emptyset)\cdots\beta_{D^M}(\emptyset)}{\beta_{D^{j_1}}(\emptyset)\cdots\beta_{D^{j_N}}(\emptyset)}\left[\frac{\delta\beta_{D^{j_1}}}{\delta\xi_1}(\emptyset)\cdots\frac{\delta\beta_{D^{j_N}}}{\delta\xi_N}(\emptyset)\right].$$

By Proposition 16, we have that $\beta_{D^j}(\emptyset) = (1-q^{(j)})$ and $\frac{\delta\beta_{D^{j_i}}}{\delta\xi_i}(\emptyset) = q_{j_i}p^{(j_i)}(\xi_i)$ and so

$$\frac{\delta^N\beta_D}{\delta\xi_1...\delta\xi_N}(\emptyset) = \sum_{1\le j_1\neq,...,\neq j_N\le M}\left[\frac{\prod_{j=1}^{M}(1-q^{(j)})}{\prod_{j=1}^{N}(1-q^{(j_k)})}\prod_{k=1}^{N}q^{(j_k)}\right]\prod_{k=1}^{N}p^{(j_k)}(\xi_k),$$

which nearly resembles Eq. (3.9). To bridge the gap, we describe the choice of indices j_i by an injective function from $\{1,...,N\}$ into $\{1,...,M\}$. In turn, each such injective function is uniquely determined by the composition of an increasing injection $\gamma \in I(N,M)$ which decides the range of the function and permutations on the domain, Π_N. These permutations take into account the order of the range. The value of $\mathcal{Q}$ is independent of order, and thus is determined by γ as in Eq. (3.10). We reorder the product in order to shift these permutations onto the input variables, obtaining

$$\frac{\delta^N\beta_D}{\delta\xi_1...\delta\xi_N}(\emptyset) = \sum_{\pi\in\Pi_N}\sum_{\gamma\in I(N,M)}\mathcal{Q}(\gamma)\prod_{k=1}^{N}p^{(\gamma(k))}(\xi_{\pi(k)}). \tag{3.12}$$

Finally, the global pdf in Eq. (3.9) follows directly from applying Eq. (3.7) to Eq.(3.12). ■

Remark 21 *The global pdf in Eq. (3.9), and in particular the sum over $\gamma \in I(N,M)$, accounts for each possible combination of singleton presence. Moreover, summing over permutations as in Eq. (3.12) and dividing by $N!$ yields a symmetric pdf with terms for every possible assignment between singletons and inputs. The weights $\mathcal{Q}(\gamma)$ indicate the probability of each assignment occurring, and is the product of the appropriate probability for each singleton to be either present, $q^{(j)}$, or absent, $1-q^{(j)}$, for each j.*

Example 1 Consider two 1-dimensional singleton diagrams, D^1 and D^2, with probabilities of being nonempty $q^{(1)} = 0.6$ and $q^{(2)} = 0.8$, respectively. The corresponding local densities when nonempty are given by $p^{(1)}(x) = \frac{1}{\sqrt{2\pi}}e^{-(x+1)^2/2}$ and $p^{(2)}(x) = \frac{1}{\sqrt{2\pi}}e^{-(x-1)^2/2}$. Lemma 20 yields the global pdf for $D = D^1 \cup D^2$ through a set of local densities $\{f_0, f_1(x), f_2(x,y)\}$ such that

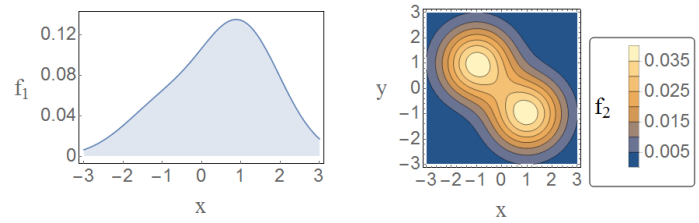


Figure 2: *Left:* Plot of the local density $f_1(x)$ in Eq. (3.13a). *Right:* Contour plot of the local density $f_2(x,y)$ in Eq. (3.13b). These pdfs cover the different possible input dimensions and are symmetric under permutations of the input.

$f_0 = \mathbb{P}[|D| = 0] = (1-q^{(1)})(1-q^{(2)}) = 0.08$, $f_1 = f_D|_{\mathbb{R}}$, and $f_2 = f_D|_{\mathbb{R}^2}$. We sum over permutations and divide by $N!$ ($N = 1,2$ is the input cardinality) to obtain a symmetric global pdf.

$$\begin{aligned} f_1(x) &= (1-q^{(2)})q^{(1)}p^{(1)}(x) + (1-q^{(1)})q^{(2)}p^{(2)}(x) \\ &= \frac{0.12}{\sqrt{2\pi}}e^{-(x+1)^2/2} + \frac{0.32}{\sqrt{2\pi}}e^{-(x-1)^2/2}, \end{aligned} \tag{3.13a}$$

$$\begin{aligned} f_2(x,y) &= \frac{q^{(1)}q^{(2)}}{2}\left[p^{(1)}(x)p^{(2)}(y) + p^{(1)}(y)p^{(2)}(x)\right] \\ &= \frac{0.24}{2\pi}\left(e^{-((x-1)^2+(y+1)^2)/2} + e^{-((x+1)^2+(y-1)^2)/2}\right). \end{aligned} \tag{3.13b}$$

Accounting for each cardinality and following Eq. (3.13a) and Eq. (3.13b), the total probability adds up to

$$\begin{aligned} \mathbb{P}[|D|=0] + \mathbb{P}[|D|=1] + \mathbb{P}[|D|=2] &= f_0 + \int_{\mathbb{R}} f_1(x)dx + \int_{\mathbb{R}^2} f_2(x,y)dxdy \\ &= (0.08) + (0.12+0.32) + (0.24+0.24) = 1, \end{aligned}$$

as desired. The local densities in Eq. (3.13a) and Eq. (3.13b) are plotted in Fig. 2. Though $f_1(x)$ is the sum of two Gaussians, in Fig. 2 (Left) we see that the Gaussian centered at $x = 1$ dominates, while the Gaussian centered at $x = -1$ is only indicated by a heavy left tail. This behavior occurs because $q^{(2)} = 0.8$ is very close to 1.

4. Kernel Density Estimation

Lemma 3.2 yields a definition of global pdf for a random persistence diagram that considers all features individually; however, as seen in Example 1, the computation of Eq. (3.9) can be rather formidable if one considers persistence diagrams with more than two points. To that end, our goal is the construction of a kernel density centered at a persistence diagram $\mathscr{D}$ with a bandwidth $\sigma > 0$ that reduces computational burden by treating some features individually and others collectively.

Generically, persistence diagrams have the majority of their points concentrated close to the diagonal. Consequently, the bandwidth σ is responsible for splitting a persistence diagram into upper and lower portions; see Eq. (4.1) and Fig. 3 (*Left*). The upper portion models the most topologically prominent points, which encompass topological information about the data, and its distribution reflects uncertainty in the precise location for prominent topological features in a persistence diagram. The lower portion models the majority of points in a persistence diagram. These points arise as a result of local noise in the underlying data, and in this fashion its distribution prescribes a noise likelihood model. Moreover, one can evaluate diagrams of any cardinality in the kernel (in this sense, the kernel is a global density). On the other hand, if one fixes the cardinality, one obtains the local kernel.

The construction of the kernel density proceeds by treating the upper and lower parts as independent, which necessitates the establishment of two density functions, one for each portion. The density for the upper part follows the recipe of Lemma 3.2 with a modified Gaussian chosen for $p^j(\xi)$ in Eq. (3.9). To construct the kernel for the lower portion, we use (i) the number of points in $\mathscr{D}^\ell$ to create a pertinent counting measure, and (ii) a modified Gaussian mixture with mean the projection of each point in $\mathscr{D}^\ell$ to the diagonal. When evaluating a persistence diagram in the composite kernel, some of the points are evaluated in the density for the lower while others are evaluated in the density for the upper part. For a particular allocation of points to the upper and lower portions and by independence, the total evaluation follows from multiplying the results of these two evaluations together. However, since it is unknown a priori which input points should be used in each kernel, one must account for every possible partitioning of input points.

Section 4.1 gives a precise construction of of our kernel density estimator. In Section 4.2, we prove that our kernel density estimator converges to the true distribution as the number of persistence diagrams used to create it goes to infinity. Finally, Section 4.3 proposes a new statistic for persistence diagrams, the mean absolute bottleneck deviation (MAD), and establishes convergence of the sample MAD computed with our kernel density estimator.

4.1. Construction

We first define a random persistence diagram as a union of simpler constituents, and then determine its global pdf by combination in a fashion similar to Lemma 20. Indeed, we define the desired kernel density as the global pdf for this composite random diagram. To start, we fix a degree of homology k and consider a center diagram $\mathscr{D} \subset \mathcal{W}_k = W \times \{k\}$ (see Eq. (2.1)). Since k is fixed, we treat $\mathscr{D} = \{\xi_i\}_{i=1}^M = \{(b_i, d_i)\}_{i=1}^M$ within $W = \{(b,d) \in \mathbb{R}^2 : d > b \geq 0\}$.

Long persistence points in a persistence diagram represent prominent topological features which are stable under perturbation of underlying data, and so it is important to track each independently. In contrast, we leverage the point of view that the small persistence features near the diagonal are considered together as a single geometric signature as opposed to individually important topological signatures. Toward this end, features with short persistence are grouped together and interpreted through i.i.d. draws near the diagonal. Since features cluster near the diagonal in a typical persistence diagram (see, e.g., Fig. 10 (g), (h) or Fig. 14 (Right)), treating short persistence features collectively simplifies our kernel density and thus speeds up its evaluation. It is imperative that these short persistence features are not ignored, because they still capture crucial geometric information for applications such as classification (Marchese and Maroulas, 2016, 2018; De Silva and Ghrist, 2007; Xia et al., 2015; Donato et al., 2016; Atienza et al., 2016). Thus, we split $\mathscr{D}$ into

upper and lower portions according to a bandwidth σ as

$$\mathscr{D}^u = \{(b_i, d_i, k) \in \mathscr{D} : d_i - b_i \geq \sigma\} \text{ and } \mathscr{D}^\ell = \{(b_i, d_i, k) \in \mathscr{D} : d_i - b_i < \sigma\}. \tag{4.1}$$

Now define random diagrams D^u centered at $\mathscr{D}^u$ and D^ℓ centered at $\mathscr{D}^\ell$ such that $D = D^u \cup D^\ell$. Ultimately, the global pdf of D centered at $\mathscr{D}$ is our kernel density.

Definition 22 *Each feature $\xi_j = (b_j, d_j) \in \mathscr{D}^u$ yields an independent random singleton diagram D^j defined by its chance to be nonempty $q^{(j)}$ (via Eq. (4.3)) along with its potential position (b,d) sampled according to a modified Gaussian distribution, denoted by $N^*((b_j, d_j), \sigma I)$. The global pdf for D^u is then determined by Lemma 20, where each $p^{(j)}$ is given by the pdf associated with $N^*((b_j, d_j), \sigma I)$, which is given by*

$$p^{(j)}(b,d) = \frac{\varphi_j(b,d)}{\int_W \varphi_j(u,v)du\,dv} \mathbb{1}_W(b,d), \tag{4.2}$$

where φ_j is the pdf of the (unmodified) normal $N((b_j, d_j), \sigma I)$, and $\mathbb{1}_W(\cdot)$ is the indicator function for the wedge.

The global pdf for each D^j is readily obtained by a pair of restrictions. First, we restrict the usual Gaussian distribution to the halfspace $T = \{(b,d) \in \mathbb{R}^2 : b < d\}$. Features sampled below the diagonal are considered to disappear from the diagram and thus we define the chance to be nonempty by

$$q^{(j)} = \mathbb{P}(D^j \neq \emptyset) = \int_{\{v>u\}} \varphi_j(u,v)\,du\,dv. \tag{4.3}$$

Afterward, the Gaussian restricted to T is further restricted to W and renormalized to obtain a probability measure as in Eq. (4.2). This double restriction to both T and W is necessary for proper restriction of the Gaussian pdf and definition of $q^{(j)} = \mathbb{P}(D^j \neq \emptyset)$. Indeed, restriction to W alone causes points with small birth time to have an artificially high chance to disappear; while restriction to T alone yields nonsensical features with negative radius (with $b < 0$). In kernel density estimation, the effects of this distinction become negligible as the bandwidth goes to zero. In practice, this distinction is important for features with small birth time relative to the bandwidth.

Remark 23 *In the Čech construction of a persistence diagram, a feature lies on the line $b = 0$ if and only if it has degree of homology $k = 0$. Consequently, for a feature $(0, d_j)$ with $k = 0$, we instead take*

$$p^{(j)}(d) = \frac{\phi_j(d)}{\int_{\mathbb{R}^+} \phi_j(u)\,du} \mathbb{1}_{\mathbb{R}^+}(d) \text{ and } q^{(j)} = \int_{\mathbb{R}^+} \phi_j(u)\,du$$

where ϕ_j is the 1-dimensional Gaussian centered at d_j with standard deviation σ.

Whereas the large persistence features in D^u have small chance to fall below the diagonal and disappear, the existence of the small persistence features in D^ℓ is volatile: these features disappear and appear fluidly under small changes in the underlying data. The distribution of D^ℓ is described by a probability mass function (pmf) ν and lower density p^ℓ.

Definition 24 *The lower random diagram D^ℓ is defined by choosing a cardinality N according to a pmf ν followed by N i.i.d. draws according to a fixed density p^ℓ. First, take $N_\ell = |\mathscr{D}^\ell|$ and define $\nu(\cdot)$ with mean N_ℓ and so that $\nu(n) = 0$ for $n > mN_\ell$ for some $m > 0$ independent of N_ℓ. The subsequent density $p^\ell(b,d)$ is given by projecting the lower features $\mathscr{D}^\ell$ of the center diagram $\mathscr{D}$ onto the diagonal $b = d$, then creating a restricted Gaussian kernel density estimation for these features; specifically,*

$$p^\ell(b,d) = \frac{1}{N_\ell} \sum_{(b_i,d_i)\in\mathscr{D}^\ell} \frac{1}{\pi\sigma^2} e^{-\left(\left(b-\frac{b_i+d_i}{2}\right)^2+\left(d-\frac{b_i+d_i}{2}\right)^2\right)/2\sigma^2}. \tag{4.4}$$

Projecting the lower features $\mathscr{D}^\ell$ of the center diagram $\mathscr{D}$ onto the diagonal simplifies later analysis and evaluation of p^ℓ; without projecting, a unique normalization factor, similar to $q^{(j)}$ in Def. 22, would be required for each Gaussian summand in Eq. (4.4). By Proposition 16 and Eq. (3.7), global pdfs of random persistence diagrams are described by a random vector pdf for each cardinality layer, resulting in the following global pdf for D^ℓ:

$$f_{D^\ell}(\xi_1,...,\xi_N) = \nu(N)\prod_{j=1}^{N} p^\ell(\xi_j). \tag{4.5}$$

Eq. (4.5) provides a noise model for the short-lived features near the diagonal. Combining the expressions for D^ℓ and D^u, we arrive at the following proposition.

Proposition 25 *Fix a center persistence diagram $\mathscr{D}$ and bandwidth $\sigma > 0$. Split $\mathscr{D}$ into $\mathscr{D}^\ell$ and $\mathscr{D}^u$ according to Eq. (4.1). Define D^ℓ with global pdf from Eq. (4.5), and D^u with global pdf from Eq. (3.9). Treating the random persistence diagrams D^u and D^ℓ as independent, define their union D. The following kernel density satisfies Def. 13 as the global pdf of D:*

$$K_\sigma(Z,\mathscr{D}) = \sum_{j=0}^{N_u} \nu(N-j) \sum_{\gamma\in I(j,N_u)} \mathcal{Q}(\gamma) \prod_{k=1}^{j} p^{(\gamma(k))}(\xi_k) \prod_{k=j+1}^{N} p^\ell(\xi_k), \tag{4.6}$$

where $Z = (\xi_1,...,\xi_N)$ is the input, $\xi_i = (b_i,d_i)$ for $i = 1,...,N$ are the features, and $N_u = |\mathscr{D}^u|$ depends on both $\mathscr{D}$ and σ. Here $\mathcal{Q}(\gamma)$ is given by Eq. (3.10), each $p^{(j)}$ refers to the modified Gaussian pdf as shown in Eq. (4.2) for its matching feature ξ_j in D^u, and p^ℓ is given by Eq. (4.4).

Proof Since D^u and D^ℓ are independent random persistence diagrams, the belief function decomposes into $\beta_D(S) = \beta_{D^u}(S)\beta_{D^\ell}(S)$. Moreover, since derivatives above order N_u vanish for β_{D^u} (see Remark 17), the product rule and binomial-type counting yield

$$\begin{aligned}\frac{\delta^N\beta_D}{\delta\xi_1...\delta\xi_N}(\emptyset) &= \sum_{j=0}^{N_u} \sum_{1\le i_1\neq...\neq i_j\le N} \frac{\delta^j\beta_{D^u}}{\delta\xi_{i_1}...\delta\xi_{i_j}}(\emptyset) \frac{\delta^{N-j}\beta_{D^\ell}}{\delta\xi_1...\delta\hat{\xi}_{i_1}...\delta\hat{\xi}_{i_j}...\delta\xi_N}(\emptyset) \\ &= \sum_{\pi\in\Pi_N}\sum_{j=0}^{N_u} \frac{1}{j!(N-j)!} \frac{\delta^j\beta_{D^u}}{\delta\xi_{\pi(1)}...\delta\xi_{\pi(j)}}(\emptyset) \frac{\delta^{N-j}\beta_{D^\ell}}{\delta\xi_{\pi(j+1)}...\delta\xi_{\pi(N)}}(\emptyset)\end{aligned} \tag{4.7}$$

where $\delta\hat{\xi}_i$ indicates that the given index is skipped in the set derivative (having been allocated to the other factor). Similar to the proof of Lemma 20, the choice of indices i_j is replaced with a

permutation $\pi \in \Pi_N$; however, the ordering within each derivative is unrelated the choice of i_j, leading to $j!$-fold and $(N-j)!$-fold redundancy within each term.

Taking Eq. (4.5) together with Eq. (3.7) yields

$$\frac{\delta\beta_{D^\ell}}{\delta\xi_{\pi(j+1)}...\delta\xi_{\pi(N)}}(\emptyset) = (N-j)!\nu(N-j)\prod_{j=1}^{N-j} p^\ell(\xi_j).$$

Also, Eq. (3.9) and Eq. (3.7) yield

$$\frac{\delta\beta_{D^u}}{\delta\xi_{\pi(1)}...\delta\xi_{\pi(j)}}(\emptyset) = \sum_{\pi^*\in\Pi_j}\sum_{\gamma\in I(j,N_u)} \mathcal{Q}(\gamma)\prod_{k=1}^{j} p^{(\gamma(k))}(\xi_{\pi^*(k)}).$$

We substitute these relations into the final expression of Eq. (4.7). The first of these substitutions is straightforward, while the second has $j!$-fold redundant permutations overtop the existing permutations in Π_N. These substitutions yield that $\frac{\delta^N\beta_D}{\delta\xi_1...\delta\xi_N}(\emptyset) = \sum_{\pi\in\Pi_N} K_\sigma(Z,\mathscr{D})$ as described in Eq. (4.6) and shows that the kernel $K_\sigma(Z,\mathscr{D})$ satisfies the definition of a global pdf for D (Def. 13). Finally, the sum over permutations is removed according to Eq. (3.7) to obtain the expression for $f_D(Z) = K_\sigma(Z,\mathscr{D})$. ■

Remark 26 *A specific example of the component distributions provided for the kernel in Proposition 25 is presented in Fig. 3. Since the kernel density K_σ of Eq. (4.6) is a probability density according to Def. 13, it is a function on $\cup_{N=0}^{M}\mathcal{W}_{0:\mathfrak{d}-1}^{N}$, and so the sum of several such kernels is defined by adding each local pdf layer separately.*

Remark 27 *In the definition of our kernel, a single parameter σ has been chosen for both the split of center diagrams, as well as the standard deviation used in the Gaussians which build our kernel. Without loss of generality, this choice simplifies the presentation of the kernel density and the proof of kernel density estimate (KDE) convergence (Theorem 31). In general, the bandwidth parameter σ_2 which refers to the standard deviation used to define the Gaussians (as σ appears in Defs. 22 and 24) need not be equal to the splitting parameter σ_1 which determines which points are in $\mathscr{D}^u$ or $\mathscr{D}^\ell$ (as σ appears in Eq. (4.1)). Still, it is certainly desirable that $\sigma_1 = C\sigma_2$ when taking a limit of KDEs as the number of persistence diagrams grows to infinity (Theorem 31). For a fixed kernel bandwidth σ_2, increasing C (and thus σ_1) moves more features into the lower portion of the diagram. This choice may be useful in practice when underlying data are known to be noisy and more noise-related features are expected near the diagonal. By the same token, for $\sigma_1 >> \sigma_2$, projecting the lower features onto the diagonal may lead to significant error in the approximation. On the other hand, taking $\sigma_1 << \sigma_2$ eliminates the computational benefit of splitting the diagram and is probably not useful in practice. For most cases, taking $\sigma_1 = \sigma_2$, is a reasonable balance between KDE accuracy and evaluation computation.*

Since the kernel density is a probability density function for a random persistence diagram, it has an associated probability hypothesis density (See Def. 18).

Corollary 28 *Fix a center persistence diagram $\mathscr{D}$ and bandwidth $\sigma > 0$. Split $\mathscr{D}$ into $\mathscr{D}^\ell$ and $\mathscr{D}^u$ according to Eq. (4.1). Define D^ℓ with global pdf from Eq. (4.5), and D^u with global pdf from*

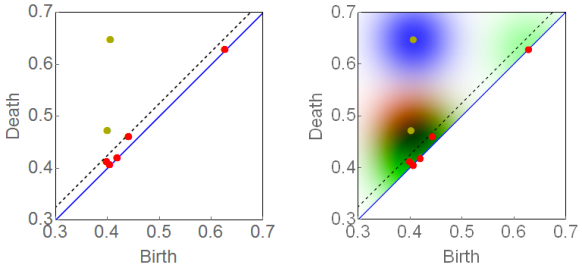


Figure 3: *Left:* A persistence diagram split according to Eq. (4.1). The dashed black line, $d = b+\sigma$, separates the diagram into the red upper points of $\mathscr{D}^u$ and the yellow lower points of $\mathscr{D}^\ell$. *Right:* The red and black gradients represent the upper singleton densities $p^{(1)}$ and $p^{(2)}$ given by Eq. (4.2). The green gradient represents the lower density p^ℓ defined in Eq 4.4. While each of these densities is defined on the wedge $W \subset \mathbb{R}^2$, the global kernel in Eq. (4.6) is defined on $\bigcup_N W^N$ for each input-cardinality N.

Eq. (3.9). *Treating the random persistence diagrams D^u and D^ℓ as independent, the probability hypothesis density (PHD) associated with the kernel density centered at $\mathscr{D}$ with bandwidth σ of Theorem 25 is given by*

$$K_{\sigma,PHD}(\xi, \mathscr{D}) = N_\ell p^\ell(\xi) + \sum_{j=1}^{N_u} q^{(j)} p^{(j)}(\xi), \tag{4.8}$$

where the feature ξ is the input and $N_u = |\mathscr{D}^u|$ and $N_\ell = |\mathscr{D}^\ell|$ depend on both $\mathscr{D}$ and σ. Here each $p^{(j)}$ refers to the modified Gaussian pdf as shown in Eq. (4.2) *for its matching singleton feature ξ_j in D^u, $q^{(j)}$ given by* (4.3) *is the probability each singleton is present, and the lower density p^ℓ is given by Eq.* (4.4).

Proof The PHD is uniquely defined by its integral over a region U, which yields the expected number of points in the region. Consequently, the independent upper and lower random draws which build the kernel contribute additively to the PHD. Within the sum, each singleton density $p^{(j)}$ is weighted by the chance for D^j to be present, $q^{(j)}$ and the lower density p^ℓ is weighted according to the mean draw cardinality, which was chosen to be $|\mathscr{D}^\ell|$. ■

Remark 29 (Computational cost) *The kernel density presented in Eq.* (4.6) *of Proposition 25 has approxmiately $N!2^{N_u}$ terms, necessitating shrewd computational strategies for real world usage. In practice, one may choose to consider only terms that correspond to high probability matchings. Such an implementation may be carried out with a linear assignment algorithm, like Munkres, resulting in a computational complexity of $O(N_u N^3)$. Another approximation for the full kernel density is to consider input features as independent draws from the PHD given in* (4.8) *of Corollary 28. Evaluation of a diagram in Eq.* (4.8) *has time complexity $O(N(N_u + N_\ell))$. Finally, sampling from the kernel in Proposition* 25 *has cost $O(N_u + N_\ell)$.*

Notice that in Corollary 28, the input for the PHD is a single feature ξ as opposed to a list of features $Z = \{\xi_1, ..., \xi_N\}$ for the global kernel in Proposition 25. Furthermore, Proposition 25 extends to the analogue result for a center persistence diagram with features of varied degree of homology.

Corollary 30 *Consider a persistence diagram $\mathscr{D} = \bigcup_{k=0}^{\mathrm{d}-1} \mathscr{D}_k \times \{k\}$ split according to the degrees of homology with associated random persistence diagrams D_k defined according to Eq.* (4.6) *for each center diagram $\mathscr{D}_k$. Treating each D_k as independent, the full global pdf for $D = \bigcup D_k$ centered at $\mathscr{D}$ with bandwidth σ is given by*

$$K_\sigma(Z, \mathscr{D}) = \Lambda(N) \prod_{k=0}^{\mathrm{d}-1} K_\sigma(Z_k, \mathscr{D}_k), \tag{4.9}$$

where $Z = \bigcup_{k=0}^{\mathrm{d}-1} Z_k \times \{k\} \subset \mathcal{W}_{0:\mathrm{d}-1}$ with each $Z_k \subset W$ of cardinality $|Z_k| = N_k$ within the multi-index $N = (N_0, ..., N_{\mathrm{d}-1})$ and

$$\Lambda(N) = \frac{N!}{|N|!} := \frac{\prod N_k!}{(\sum N_k)!}.$$

Proof The result follows immediately from taking set derivatives of the full belief function $\beta_D(S) = \prod_k \beta_{D_k}(S)$. In particular, the set derivatives $\frac{\delta \beta_{D_k}}{\delta Z}(\emptyset)$ are zero unless $Z \subset \mathcal{W}_k$. Thus, the product rule leaves only the single term $\frac{\delta \beta_D}{\delta Z}(\emptyset) = \prod_{k=0}^{\mathrm{d}-1} \frac{\delta \beta_{D_k}}{\delta Z_k}(\emptyset)$. In turn, each kernel global pdf $K_\sigma(Z_k, \mathscr{D}_k)$ is related to the associated belief function derivative by a sum over permutations Π_{N_k} (see Eq. (3.7)). Compositions of these permutations are $N_k!$-fold redundant against the $|N|!$ permutations in $\Pi_{|N|}$, yielding the coefficient $\Lambda(N)$. ■

4.2. Convergence of the Kernel Density Estimator

In this section, to prove the convergence (to the target distribution) of the kernel density estimate defined via the kernel established in Proposition 25, we consider persistence diagrams $\{\mathscr{D}_i\}_{i=1}^n$ which are i.i.d. sampled from a target distribution with global pdf f. Toward this end, we require the following assumptions on f:

- $(A1)$ $f(Z) = 0$ for $|Z| > M \in \mathbb{N}$ (bounded cardinality).
- $(A2)$ The local density $f_N : \mathcal{W}_k^N \to \mathbb{R}$ is bounded for each $N \in \{1, ..., M\}$.
- $(A3)$ There exists $C_N > 0$ so that $f(\xi_1, ..., \xi_N) \leq C_N \|(\xi_1, ..., \xi_N)\|^{-2N}$ for each $N \in \{1, ..., M\}$.

The assumptions (A1), (A2), and (A3) describe conditions on the target random persistence diagram pdf. It is important that these assumptions also hold for a random persistence diagram associated with typical (random) underlying datasets. For example, $(A1)$ trivially holds for underlying data in $\mathbb{R}^{\mathrm{d}}$ of bounded cardinality. The conditions $(A2)$ and $(A3)$ hold for underlying data sampled from a compact set $E \subset \mathbb{R}^{\mathrm{d}}$ perturbed by Gaussian noise. The work (Adler et al., 2014)(see Corollary 2.3 and Thm 2.6 therein) describes the persistent homology of noise, and describes a 'core' neighborhood. Specifically for Gaussian noise, features are retained in the 'core', but then extreme decay occurs for features of arbitrary degree outside the 'core'. Intuitively, by

bounding death values by the diameter of the underlying dataset, one expects that the decay will be at worst a polynomial times Gaussian decay, which is sufficient for (A3).

The following theorem shows that the kernel density estimate converges to the true global pdf of a random persistence diagram as the number of persistence diagrams increases. The pdf tracks not only the birth and death of features, but also their prevalence. In particular, the persistence diagram pdf tied to a random dataset can determine which geometric features are stable regardless of their persistence.

Theorem 31 *Consider a random persistence diagram global pdf f satisfying assumptions (A1)-(A3). Define the kernel $K_\sigma(Z, \mathscr{D})$ according to Theorem 25 and consider the kernel density estimate $\hat{f}(Z) = \frac{1}{n}\sum_{i=1}^{n} K_\sigma(Z, \mathscr{D}_i)$, with centers $\mathscr{D}_i$ sampled i.i.d. according to global pdf f and bandwidth $\sigma = O(n^{-\alpha})$ chosen with $0 < \alpha < \alpha_{2M}$. Then, as $n \to \infty$, $\hat{f} \to f$ uniformly on compact subsets of W.*

Remark 32 *The value of α_{2M} is inherited from bandwidth selection for $2M$-dimensional kernel density estimates (Scott, 2015). While the scaling of the bandwidth in the limit is determined by the maximum cardinality M (and thus, the largest dimension of the local pdfs), choosing a bandwidth for a specific sample is an important step in applying kernel density estimation. If the bandwidth is too narrow, the estimate is overfitted and potentially biased; if the bandwidth is too large, the estimate will be oversmoothed, resulting in accuracy loss. Several methods for bandwidth selection in multivariate kernel estimation are discussed in (Silverman, 1986). As a general rule of thumb, (Silverman, 1986) recommends choosing the bandwidth as $\sigma_{opt} = A(K)n^{-1/(\mathrm{d}+4)}$, where n is the sample size (i.e., the number of persistence diagrams), d is the dimension, and $A(K)$ is a constant depending on the kernel, K. In particular, one may choose $\alpha \approxeq 1/(2M+4)$ as an unbiased estimator for all local pdfs with cardinalities $m \leq M$ (Scott, 2015). Silverman's rule of thumb works best for distributions which are nearly Gaussian; for more general distributions, the bandwidth may be chosen empirically.*

Eq.(4.8) of Corollary 28 could be used as an approximation to the full kernel. The following argument verifies its convergence.

Corollary 33 *Let F_g denote the PHD (Def. 18) of a random persistence diagram with global pdf g. Define $\hat{f}$ as in Thm 31. For a random persistence diagram whose global pdf f satisfies assumptions (A1)-(A3) of Thm 31, one has $F_{\hat{f}} \to F_f$ as $n \to \infty$ almost everywhere.*

Proof Let $C \subset W$ be compact. Define the counting function $\kappa_C(Z) = |Z \cap C|$. Notice by Def. 18 that $|\int_C F_{\hat{f}}(u)du - \int_C F_f(u)du| \leq \int_W \kappa_C(Z)|\hat{f}(Z) - f(Z)|\delta Z \leq M(2^{M+1}-1)\int_C |\hat{f}(Z) - f(Z)|\delta Z$, where the last inequality follows because we assume that random persistence diagrams are bounded by M and κ_C vanishes outside of C. By Thm 31 and boundedness of C, we can choose n sufficiently large to ensure $\int_C |\hat{f}(Z) - f(Z)|\delta Z$ is arbitrarily small. Hence, $\int_C F_{\hat{f}}(u)du \to \int_C F_f(u)du$ as $n \to \infty$ on arbitrary compact subsets C. The result follows immediately by standard results in measure theory. ■

The proof of Thm. 31 presented in this section describes the case for degree of homology $k > 0$. The case for $k = 0$ is obtained by a slight modification and the full result follows by an application of Corollary 30.

Throughout the proof we use ξ_i to denote input features and $Z = \{\xi_1, ..., \xi_N\}$ or $Z = (\xi_1, ..., \xi_N)$ to denote an input persistence diagram as a set or vector of features. Several preliminary lemmas are presented before the main body of the proof. We begin with a critical lemma which controls the number of features sampled in the band diagonal $\Delta_\alpha^\beta = \{(b, d) \in W : \alpha < d - b < \beta\}$.

Lemma 34 *Consider a random persistence diagram D distributed according to f satisfying assumptions (A1)-(A3). Then there exists $C > 0$ so that $\mathbb{E}^f(|\Delta_0^\sigma \cap D|) \leq C\sigma$.*

Proof Consider a region $A \subset W$ and a counting function $\kappa_A(Z) = |Z \cap A|$ such that $\kappa_A(\{\xi_1, ..., \xi_N\}) = \sum_{i=1}^{N} \mathbb{1}_A(\xi_i)$. It is clear that this set function is well defined and measurable if A is measurable. Using set integration (Def. 12),

$$\mathbb{E}(|\Delta_0^\sigma \cap D|) = \int_W \kappa_{\Delta_0^\sigma}(Z)f(Z)\delta Z = \sum_{N=0}^{M} \frac{N}{N!}\int_W \mathbb{1}_{\Delta_0^\sigma}(\xi_1)\left[\int f(\xi_1, ...\xi_N)d\xi_2...d\xi_N\right]d\xi_1 \tag{4.10}$$

The expressions in Eq. (4.10) can be phrased in terms of the probability hypothesis density from Eq. (3.8), and for any choice of $L > 0$ are bounded by

$$\begin{aligned}\int_{\Delta_0^\sigma} F_D(\xi)d\xi &\leq \int_0^L \int_{y-\sigma}^{y} F_D(x, y)\,dx\,dy + \int_L^\infty \int_{y-\sigma}^{y} C_3 y^{-2}\,dx\,dy \\ &\leq LC_2\sigma + 3C_3\sigma/L = (LC_2 + C_3/L)\sigma\end{aligned}$$

where assumptions (A2) and (A3) respectively yield the bounds C_2 and $C_3 y^{-2}$ on the probability hypothesis density, F_D. ■

Lemma 34 yields control over the counting measure ν_i defined in Def. 24 and the coefficients $\mathcal{Q}_i^*(\cdot)$ of Eq. (3.11) which respectively determine the distribution of lower and upper cardinalities for a persistence diagram sampled according to the kernel density $K_\sigma(Z, \mathscr{D}_i)$.

Corollary 35 *Consider a random persistence diagram D distributed according to f satisfying assumptions (A1)-(A3). Take ν to be the lower cardinality probability mass function for the kernel density $K_\sigma(Z, D)$ shown in Eq. (4.6). Then, there exists $C > 0$ so that $\mathbb{E}^f\nu(j_0) \leq C\sigma$ whenever $j_0 \neq 0$.*

Proof Since D is random with respect to f, ν is random with respect to f as well. Recall that ν is defined so that $\mathbb{E}^\nu(\boldsymbol{a}) = |D^\ell|$ for $\boldsymbol{a}$ distributed according to ν and thus $\mathbb{E}^f[\mathbb{E}^\nu(\boldsymbol{a})] \leq C\sigma$ for some $C > 0$ by Lemma 34. Subsequently, the value $\mathbb{E}^f\nu(j_0)$ is controlled by this double expectation so long as $j_0 \neq 0$. Indeed,

$$\mathbb{E}(\boldsymbol{a}) = \sum_{j=0}^{\infty} j\nu(j) = \sum_{j=1}^{\infty} j\nu(j) \geq \sum_{j=1}^{\infty} \nu(j) \geq \nu(j_0)$$

for any $j_0 > 0$ and $\nu_i(j_0) = 0$ for $j_0 < 0$ since it represents a cardinality distribution. ■

In the following lemma, the result of Lemma 34 is used to control the expressions $\mathcal{Q}(\gamma)$ or $\mathcal{Q}^*(\gamma)$, of Eq. (3.10) and Eq. (3.11) respectively, in the kernel density estimate.

Lemma 36 *Consider a random persistence diagram D distributed according to f satisfying assumptions (A1)-(A3). Take $\mathcal{Q}$ of Eq. (3.10) and $\mathcal{Q}^*$ of Eq. (3.11) to be the upper singleton probabilities for the kernel density $K_\sigma(Z,D)$ shown in Eq. (4.6). Then, there exists $C>0$ so that $\mathbb{E}^f[\mathcal{Q}(\gamma)] \le \mathbb{E}^f[\mathcal{Q}^*(\gamma)] \le C\sigma$ for any $\gamma \in I(j,N)$ with $j<N$.*

Proof Since every $q^{(k)} \in (0,1)$, we have that $\mathcal{Q}(\gamma) \le \mathcal{Q}^*(\gamma)$; and furthermore, since $\gamma \in I(j,N)$ are not onto when $j<N$, each product $\mathcal{Q}^*$ is bounded by one of the terms of the $(1-q_i^{(k)})$ type. By construction, these terms depend monotonically upon a feature's persistence, and the maximum (over all indices $j<N$ and functions γ) is tied to the least persistent feature of $\mathscr{D}_i^u$.

For a feature (b,d) of persistence $p=d-b$, we define $q(p) := \int_{-p/(\sqrt{2}\sigma)}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}dx$ in concordance with Eq. (4.3); or in terms of the error function Φ, $q(p) = \frac{1}{2}\left(1+\Phi\left(\frac{p}{2\sigma}\right)\right)$. Define the minimal persistence as $p_{\min}(Z) = \sup\{p : |\Delta_0^p \cap Z| = \emptyset\}$ which satisfies $p_{\min}(Z) \ge p$ if and only if $|\Delta_0^p \cap Z| = \emptyset$. In turn, we may bound $\mathcal{Q}^*(\gamma) \le (1-q(p_{\min}(D))$ independently of γ. By Lemma 34, there is $C>0$ such that $\mathbb{P}^f[|\Delta_0^\sigma \cap D| \ne \emptyset] \le \mathbb{E}^f[|\Delta_0^\sigma \cap D|] \le C\sigma$, which controls the distribution of the minimal persistence.

In particular, $q'(p) = \frac{1}{2\sigma\sqrt{\pi}} e^{-p^2/4\sigma^2}$ by the fundamental theorem of calculus. The control of Lemma 34 and the fact that $p_{\min}(Z) \ge 0$ also allows us to use integration via the probability of sublevel sets. Take $g(p) = 1-q(p)$ so that $\lim_{p\to\infty} g(p) = 0$. Specifically, since $\mathcal{Q}^*(\gamma) \le (1-q(p_{\min}(D))$, and using the fundamental theorem of calculus then Fubini's theorem, we have:

$$\begin{aligned}\mathbb{E}^f[\mathcal{Q}^*(\gamma)] &\le \int_{\mathcal{W}_{0:\mathrm{d}-1}} g(p_{\min}(Z))f(Z)\delta Z = \int_{\mathcal{W}_{0:\mathrm{d}-1}} \left(\int_{\infty}^{p_{\min}(Z)} g'(p)dp\right) f(Z)\delta Z \\ &= \int_{\infty}^{0}\left(\int_{\{Z:p_{\min}(Z)<p\}} f(Z)\delta Z\right) g'(p)dp = \int_0^\infty \left(\mathbb{P}^f[p_{\min}<p]\right) q'(p)dp.\end{aligned} \tag{4.11}$$

We now further bound the expectation in Eq. (4.11). Replacing terms with their definitions and using the bound control from Lemma 34 we obtain:

$$\begin{aligned}\mathbb{E}^f[\mathcal{Q}^*(\gamma)] &\le \int_0^\infty \mathbb{P}^f(\Delta_0^p \cap D \ne \emptyset)\frac{1}{2\sigma\sqrt{\pi}} e^{-p^2/4\sigma^2}dp \\ &\le \frac{C}{2\sigma\sqrt{\pi}}\int_0^\infty pe^{-(p/2\sigma)^2}dp = \frac{C}{2\sigma\sqrt{\pi}}\left[-2\sigma^2 e^{-p^2/4\sigma^2}\right]_{p=0}^{\infty} = \frac{C}{\sqrt{\pi}}\sigma.\end{aligned}$$

■

Proof of Theorem 31. For convenience, we denote the upper cardinalities by $N_i = |\mathscr{D}_i^u|$ and total cardinalities by $M_i = |\mathscr{D}_i|$ for the sample persistence diagrams. Denote the set of strictly increasing functions from $\{1,...,j\}$ into $\{1,...,N_i\}$ by $I(j,N_i)$. Here we use 'id' to denote the identity map, where $I(N_i,N_i) = \{\mathrm{id}\}$. The proof is organized by splitting the kernel densities into several pieces and then controlling each piece separately.

First, we separate the kernel $K_\sigma(Z,\mathscr{D}_i)$, defined in Eq. (4.6), into three portions, A_i, B_i, and C_i, according to the upper cardinality j:

$$\begin{aligned}K_\sigma(Z,\mathscr{D}_i) &= \sum_{j=0}^{N_i}\nu_i(N-j)\sum_{\gamma\in I(j,N_i)}\mathcal{Q}_i(\gamma)\prod_{k=1}^{j}p_i^{(\gamma(k))}(\xi_k)\prod_{k=j+1}^{N}p_i^{\ell}(\xi_k) \\ &= \nu_i(N-N_i)\mathcal{Q}_i(\mathrm{id})\prod_{k=1}^{N_i}p_i^{(k)}(\xi_k)\prod_{k=N_i+1}^{N}p_i^{\ell}(\xi_k) \\ &+ \sum_{j=0,j\ne N}^{N_i-1}\nu_i(N-j)\sum_{\gamma\in I(j,N_i)}\mathcal{Q}_i(\gamma)\prod_{k=1}^{j}p_i^{(\gamma(k))}(\xi_k)\prod_{k=j+1}^{N}p_i^{\ell}(\xi_k) \\ &+ \mathbb{1}_{\{n\in\mathbb{N}:n<N_i\}}(N)\nu_i(0)\sum_{\gamma\in I(N,N_i)}\mathcal{Q}_i(\gamma)\prod_{k=1}^{N}p_i^{(\gamma(k))}(\xi_k) \\ &= A_i + B_i + C_i,\end{aligned} \tag{4.12}$$

where A_i follows from $j=N_i$, C_i follows from $j=N$ ($C_i=0$ if $N_i \le N$), and B_i consists of all remaining terms.

The terms B_i in Eq. (4.12) are controlled by the lower product $\left[\prod_{k=j+1}^{N}p_i^{\ell}(\xi_k)\right]$. Since $(1-q_i^{(j)}) \le 1$ and $\nu_i(N-j) \le 1$ for any choice of γ and j, we have that B_i is bounded above by

$$\sum_{j=0,j\ne N}^{N_i-1}\sum_{\gamma\in I(j,N_i)}\left[\prod_{k=1}^{j}q_i^{(\gamma(k))}p_i^{(\gamma(k))}(\xi_k)\prod_{k=j+1}^{N}p_i^{\ell}(\xi_k)\right]. \tag{4.13}$$

The bounding sum of Eq. (4.13) consists of restricted $2N$-dimensional Gaussians, with the weights $q_i^{(j)}$ dominating the restriction rescaling in Eq. (4.2). Fix $\pi \in \Pi_N$ and $j \in \{0,...,M-1\}\setminus\{N\}$. Without loss of generality, we treat the case when the permutation π is the identity. Since our ultimate goal is to control the kernel density estimate $\hat{f}$, consider the portion of $\sum_{i=1}^n \frac{1}{n}B_i$ for which the cardinalities $M_i = |\mathscr{D}_i|$ are fixed at level $M_i = m \in \{0,...,M\}$. Now, $m = |\mathscr{D}_i| \ge N_i > j$, so there is some extension for every γ within the sum, $\gamma^* \in \Pi_m$. Recall that this collection is random because each $\mathscr{D}_i$ is randomly distributed according to f, therefore we consider the expectation with respect to this randomness:

$$\mathbb{E}^f\left[\sum_{\{i:M_i=m\}}\frac{1}{|\{i:M_i=m\}|}\prod_{k=1}^{M_i}q_i^{(\gamma^*(k))}p_i^{(\gamma^*(k))}(\xi_k)\right] \to f(\xi_1,...,\xi_m),$$

for any point $(\xi_1,...,\xi_m)$ as a $2m$-dimensional Gaussian kernel density estimate with a proper choice of $\sigma = O(n^{-\alpha})$ appropriate for $2M$ (and hence $2m$) dimensions (Scott, 2015). Integrating both sides against the extra coordinates, Assumptions (A2) and (A3) along with the dominated convergence theorem yield

$$\mathbb{E}^f\left[\sum_{\{i:M_i=m\}}\frac{1}{|\{i:M_i=m\}|}\prod_{k=1}^{j}q_i^{(\gamma(k))}p_i^{(\gamma(k))}(\xi_k)\right] \to \int_{W^{m-j}} f(\xi_1,...,\xi_m)d\xi_{j+1}...d\xi_m, \tag{4.14}$$

which is again bounded via (A2) and (A3). Of course, $|\{i : M_i = m\}| \leq n$, so taking Eq. (4.14) into account for every m bounds the averaging sum of the upper product: $\frac{1}{n}\sum_{i=1}^{n}\prod_{k=1}^{j} q_i^{(\gamma(k))} p_i^{(\gamma(k))}(\xi_k)$.

Relying on Eq. (4.13), we must also consider the lower product $\prod_{k=j+1}^{N} p_i^{\ell}(\xi_k)$. Since the points ξ_i are fixed, we focus on their minimal persistence $p_{\min} = \min_i(d_i - b_i)$. Thus,

$$p_i^{\ell}(\xi_i) \leq \frac{1}{2\pi\sigma^2} e^{-(b-d)^2/4\sigma^2} \leq \frac{1}{2\pi\sigma^2} e^{-p_{\min}^2/4\sigma^2},$$

and subsequently,

$$\left[\prod_{k=j+1}^{N} p_i^{\ell}(\xi_k)\right] \leq \frac{1}{(2\pi\sigma^2)^N} e^{-Np_{\min}^2/4\sigma^2} \to 0, \tag{4.15}$$

as $\sigma \to 0$, uniformly on any compact subset of W (or $\mathcal{W}_{0:\mathrm{d}-1}$). Altogether, Eqs. (4.14) and (4.15) guarantee that the term $\sum_{i=1}^{n}\frac{1}{n}B_i \to 0$ as $n \to \infty$ in the kernel density estimation.

Next we focus on the terms A_i in Eq. (4.12). We split the sum $\frac{1}{n}\sum_{i=1}^{n} A_i$ according to the cardinality of $\mathscr{D}_i$. Specifically, separate A_i into the cases where $M_i \neq N_i$ or $M_i = N_i$. First consider the associated set of indices $\{i : M_i \neq N_i\}$ and define the mismatch number $\mathrm{MM}(n)$ to be its cardinality. Critical to our argument, the mismatch number is random with respect to f because it is defined according to the features in $\mathscr{D}_i$. We obtain the following mismatched term:

$$\frac{1}{n}\sum_{\{i:N_i\neq M_i\}} A_i \leq \left(\frac{MM(n)}{n}\right)\frac{1}{MM(n)}\sum_{\{i:N_i\neq M_i\}}\left[\mathcal{Q}_i(\mathrm{id})\prod_{k=1}^{N_i} p_i^{(k)}(\xi_k)\prod_{k=N_i+1}^{N} p_i^{\ell}(\xi_k)\right] \tag{4.16}$$

The bounding sum in Eq. (4.16) is split into pieces where $M_i = m$ for each m between 0 and M. Using the same strategy yielding Eq. (4.14), with $MM(n)$ in place of n, the sum of the upper product converges to layered integrals of f for each level m and each $N_i < m$ by extending $\gamma = \mathrm{id}$. Using the same approach leading to Eq. (4.15), the lower product vanishes in the limit if $N_i \neq N$, or is an empty product if $N_i = N$; in either case, this factor is bounded. Now, according to Lemma 34, $\mathbb{P}^f(M_i \neq N_i) = \mathbb{P}^f(\mathscr{D}_i \cap \Delta_0^{\epsilon\sigma} \neq \emptyset) \leq C_5\sigma$; consequently, $\mathbb{E}^f[MM(n)/n] \to 0$ and the mismatch terms on left hand side of Eq. (4.16) follow.

Now consider the indices for which $N_i = M_i$. In this case, since $\mathscr{D}_i^{\ell}$ are empty, $\nu_i = \delta_0$, and the only values which contribute to the sum are for $N_i = N$. The remaining portion of the kernel density estimate is given by

$$\frac{1}{n}\mathbb{E}^f\sum_{\{i:N_i=M_i\}} A_i = \frac{1}{n}\mathbb{E}^f\left[\sum_{\{i:N_i=M_i\}}\left(\mathcal{Q}_i(\mathrm{id})\prod_{k=1}^{N} p_i^{(k)}(\xi_k)\right)\right] = \frac{1}{n}\mathbb{E}^f\left[\sum_{\{i:N_i=M_i\}}\left(\prod_{k=1}^{N} q_i^{(k)} p_i^{(k)}(\xi_k)\right)\right]. \tag{4.17}$$

As shown, the terms in Eq. (4.17) are restricted $2N$ dimensional Gaussians. It is known (Scott, 2015) that restricted Gaussian kernel density estimates like $\left[\prod_{k=1}^{N} q_i^{(k)} p_i^{(k)}(\xi_k)\right]$ converge (uniformly on compactly contained sets) to the true value of the chosen draws $\mathscr{D}_i$ for a suitable choice of α in $\sigma = O(n^{-\alpha})$ as restricted by $N \leq M$. After correcting for the samples with $N_i < M_i = N$, the samples $\mathscr{D}_i$ are treated as random draws from $f(D||D| = N)$. Consequently, we may conclude

that the target distribution associated with $\left[\prod_{k=1}^{N} q_i^{(k)} p_i^{(k)}(\xi_k)\right]$ is the rescaled $\frac{1}{f(N)} f(\xi_1, ..., \xi_N)$, where $f(N) := \mathbb{P}^f(|D| = N)$. This rescaling for the conditional pdf $f(D||D| = N)$ is necessary to reweight according to Proposition 16.

Application of classical kernel density estimate results require division by the cardinality of the draw, when in context n is generally larger than this cardinality. Thus, we must again consider the cases wherein $N_i \neq M_i$. Consequently, we find that the expectation for the ratio between the true draw cardinality and n is given by $\mathbb{P}^f(|D| = N) + O(\sigma)$ according to Lemma 34. Indeed, this ratio converges to $f(N) := \mathbb{P}^f(|D| = N)$. After this final correction, we have shown that $\frac{1}{n}\sum_{i=1}^{n} A_i$ approach the true pdf $f(\xi_1, ..., \xi_N)$.

Lastly, we need only to control the terms C_i from Eq. (4.12). We begin by bounding the probability mass functions ν_i by 1 and considering only terms for which the characteristic function is nonzero:

$$\frac{1}{n}\sum_{i=1}^{n} C_i = \frac{1}{n}\sum_{\{i:N<N\}}\nu_i(0)\sum_{\gamma\in I(N,N_i)}\mathcal{Q}_i(\gamma)\prod_{k=1}^{N} p_i^{(\gamma(k))}(\xi_k) \leq \frac{1}{n}\sum_{\{i:N<N_i\}}\sum_{\gamma\in I(N,N_i)}\mathcal{Q}_i(\gamma)\prod_{k=1}^{N} p_i^{(\gamma(k))}(\xi_k). \tag{4.18}$$

Next, we split the term $\mathcal{Q}(\gamma)$ according to Eq. (3.10) and apply Lemma 36 to the upper bound in Eq. (4.18) to obtain the larger upper bound

$$\frac{1}{n}\sum_{\{i:N<N_i\}}\sum_{\gamma\in I(N,N_i)}\mathcal{Q}^*(\gamma)\prod_{k=1}^{N} q_i^{(\gamma(k))} p_i^{(\gamma(k))}(\xi_k) \leq C\left[\frac{1}{n}\sum_{\{i:N<N_i\}}\sum_{\gamma\in I(N,N_i)}\prod_{k=1}^{N} q_i^{(\gamma(k))} p_i^{(\gamma(k))}(\xi_k)\right]\sigma. \tag{4.19}$$

The expectation of the bracketed terms in Eq. (4.19) converges in a fashion identical to the terms $\frac{1}{n}\sum_{i=1}^{n} A_i$. Since these terms are multiplied by σ, altogether $\left[\frac{1}{n}\sum_{i=1}^{n} C_i\right]$ vanishes in the limit as $n \to \infty$. Putting together the limits of each portion built from $K_\sigma(Z, \mathscr{D}_i) = A_i + B_i + C_i$, the theorem follows. ■

4.3. A Measure of Dispersion

Theorem 31 has established the convergence of a kernel density estimator. Along with density function estimation, one would like to verify the convergence of properties such as spread. In the absence of vector space structure on the space of persistence diagrams, we turn to the bottleneck metric (Def. 5) to define a notion of spread. Specifically, we measure dispersion with respect to a distribution of persistence diagrams through its mean absolute deviation in this metric.

Definition 37 *The mean absolute bottleneck deviation (MAD) from origin diagram $\mathscr{D}$ with respect to a global pdf f is given by*

$$MAD_f(\mathscr{D}) = \int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}, Z) f(Z)\delta Z \tag{4.20}$$

The following proposition and lemma aid in proving convergence of MAD kernel estimates. Their proofs are delegated to the supplementary materials.

Proposition 38 *Consider D distributed according to the kernel density $K_\sigma(\cdot, \mathscr{D})$ with center diagram $\mathscr{D}$ and bandwidth σ. Fix $\delta \geq 1$. Then,*

$$\mathbb{P}\left[W_\infty(D, \mathscr{D}) < \delta\sigma\right] \geq \left(\int_{B(\mathbf{0},\delta)} \frac{1}{2\pi} e^{-(x^2+y^2)/2}\, dx\, dy\right)^M \tag{4.21}$$

where M is the maximal cardinality of D (a multiple of $|\mathscr{D}|$). Here $B(x, r)$ refers to a ball with respect to the infinity metric (as is used in bottleneck distance).

Next, we relax assumption $(A2)$ by considering the entire multi-wedge $\mathcal{W}_{0:\mathrm{d}-1}$ and tighten the decay control from assumption $(A3)$. Formally,

$(A2)^*$ The local density $f_N : \mathcal{W}_{0:\mathrm{d}-1}^N \to \mathbb{R}$ is bounded for each $N \in \{1, ..., M\}$.

$(A3)^*$ There exists $C > 0$ so that $f(\xi_1, ..., \xi_N) \leq C \left\|(\xi_1, ..., \xi_N)\right\|^{-2N-2}$ for $N \in \{1, ..., M\}$.

These assumptions (and $(A1)$) are required for the subsequent lemma, which ensures that the mean absolute bottleneck deviation (MAD) is finite.

Lemma 39 *Consider a random persistence diagram D distributed according to a global pdf f satisfying assumptions $(A1)$, $(A2)^*$, and $(A3)^*$. Then D has finite MAD for any choice of origin diagram $\mathscr{D}$.*

Similar to assumption $(A3)$ (given prior to Theorem 31), $(A3)^*$ holds for a random persistence diagram associated with underlying data sampled from a compact set perturbed by Gaussian noise. One may also replace Lemma 39 and its assumptions by directly assuming that the maximal persistence moment is bounded; with this, the results of Lemma 39 follow immediately from Eq. (A.3) in the supplementary. This direct assumption is weaker (implied by $(A1)$, $(A2)^*$, and $(A3)^*$), but may be difficult to show directly in practice.

Theorem 40 *Consider a distribution of persistence diagrams with bounded global pdf, f, satisfying assumptions $(A1)$, $(A2)^*$, and $(A3)^*$. Let $\hat{f}(Z) = \frac{1}{n}\sum_{i=1}^n K_\sigma(Z, \mathscr{D}_i)$ be a kernel density estimate with centers $\mathscr{D}_i$ sampled i.i.d. according to global pdf f and bandwidth $\sigma = O(n^{-\alpha})$ chosen with $0 < \alpha < \alpha_{2M}$. Then, the mean absolute bottleneck deviation estimate converges; in other words,*

$$\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z)\hat{f}(Z)\delta Z \to \int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z) f(Z)\delta Z \tag{4.22}$$

as $n \to \infty$ for any origin diagram $\mathscr{D}_0$.

Proof The MAD of f with origin $\mathscr{D}_0$ is finite by Lemma 39. To show convergence of the estimate, we begin by adding and subtracting the integral of the sample estimator for the MAD. Then, we split the sum into $n+1$ terms via the triangle inequality to obtain

$$\begin{aligned}
&\left|\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z) f(Z)\delta Z - \int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z)\hat{f}(Z)\delta Z\right| \\
&\quad \leq \left|\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z) f(Z)\delta Z - \frac{1}{n}\sum_{i=1}^n \int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, \mathscr{D}_i) K_\sigma(Z, \mathscr{D}_i)\delta Z\right| \\
&\quad + \frac{1}{n}\sum_{i=1}^n \left|\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z) K_\sigma(Z, \mathscr{D}_i)\delta Z - \int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, \mathscr{D}_i) K_\sigma(Z, \mathscr{D}_i)\delta Z\right|.
\end{aligned} \tag{4.23}$$

The term of the upper bound in Eq. (4.23) trivially simplfies to obtain the sample estimator for the MAD:

$$\begin{aligned}
&\left|\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z) f(Z)\delta Z - \sum_{i=1}^n \frac{1}{n}\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, \mathscr{D}_i) K_\sigma(Z, \mathscr{D}_i)\delta Z\right| \\
&\quad = \left|\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z) f(Z)\delta Z - \frac{1}{n}\sum_{i=1}^n W_\infty(\mathscr{D}_0, \mathscr{D}_i)\right|.
\end{aligned} \tag{4.24}$$

The MAD sample estimator converges since the MAD is finite, and thus this term vanishes as $n \to \infty$. The remaining term of the upper bound in Eq. (4.23) is further bounded via the reverse triangle inequality; specifically,

$$\begin{aligned}
&\sum_{i=1}^n \frac{1}{n}\left|\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, Z) K_\sigma(Z, \mathscr{D}_i)\delta Z - \int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_0, \mathscr{D}_i) K_\sigma(Z, \mathscr{D}_i)\delta Z\right| \\
&\quad \leq \sum_{i=1}^n \frac{1}{n}\left|\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_i, Z) K_\sigma(Z, \mathscr{D}_i)\delta Z\right|.
\end{aligned} \tag{4.25}$$

Toward bounding Eq. (4.25), choose a threshold parameter $a = O(\sigma^\beta)$ for some $\beta \in (0, 1)$, so that $a \to 0$ but $a/\sigma \to \infty$ in the sample size (and bandwidth) limit. Next, take $A_i = \{Z \subset W : W_\infty(Z, \mathscr{D}_i) \leq a\}$ and split the integral between A_i and its complement as

$$\int_{\mathcal{W}_{0:\mathrm{d}-1}} W_\infty(\mathscr{D}_i, Z) K_\sigma(Z, \mathscr{D}_i)\delta Z = \int_{A_i} W_\infty(\mathscr{D}_i, Z) K_\sigma(Z, \mathscr{D}_i)\delta Z + \int_{A_i^c} W_\infty(\mathscr{D}_i, Z) K_\sigma(Z, \mathscr{D}_i)\delta Z.$$

The integral over A_i is trivially bounded by a. Integration over the complementary events is controlled via layered integration along with Proposition 38. For $a/\sigma > 1$, which occurs when n is large enough, we obtain

$$\begin{aligned}
\int_{A_i^c} W_\infty(\mathscr{D}_i, Z) K_\sigma(Z, \mathscr{D}_i)\delta Z &= a\,\mathbb{P}^i\left[W_\infty(\mathscr{D}_i, Z) > a\right] + \int_a^\infty \mathbb{P}^i\left[W_\infty(\mathscr{D}_i, Z) > b\right] db \\
&\leq a\left(\mathbb{P}[|z| > a/\sigma]^M\right) + \int_a^\infty \left(\mathbb{P}[|z| > b/\sigma]^M\right) db,
\end{aligned} \tag{4.26}$$

where $z = (x, y)$ is distributed as a pair of independent standard normals. We chose $a/\sigma = O(\sigma^{\beta-1}) \to \infty$ and so $\mathbb{P}(|z| < a/\sigma) \to 0$ exponentially fast and the last term vanishes quickly as $\sigma \to 0$.

Indeed, let $g(Z) = W_\infty(\mathscr{D}_i, Z)$, then by the fundamental theorem of calculus and Fubini's theorem:

$$\begin{aligned}
\int_{A_i^c} g(Z) K_\sigma(Z, \mathscr{D}_i)\delta Z &= \int_{\{Z : g(Z) > a\}} \left(\int_0^{g(Z)} db\right) K_\sigma(Z, \mathscr{D}_i)\delta Z \\
&= \int_0^\infty \int_{\{Z : g(Z) > a \text{ and } g(Z) > b\}} K_\sigma(Z, \mathscr{D}_i)\delta Z\, db \\
&= \int_0^\infty \mathbb{P}^f[g(Z) > \max\{a, b\}] db \\
&= a\mathbb{P}^f[g(Z) > a] + \int_a^\infty \mathbb{P}^f[g(Z) > b] db.
\end{aligned}$$

Applying Proposition 38 changes the probabilities on $g(Z)$ to normal tail probabilities. Thus, both bounding terms in Eq. (4.23) converge to zero and thus the kernel estimate converges to the true mean absolute deviation. ■

5. Examples

Here we provide detailed examples of the kernel density and kernel density estimation of an unknown pdf. For simplicity, we restrict to a single degree of homology, say $k = 1$. Due to the intrinsic high dimension of the kernel, we present contour plots for slices of the kernel density. Specifically, for inputs $((b_1, d_1), ..., (b_N, d_N))$, we consider the kernel density evaluated at $(b_1, d_1) \in W$ with (b_i, d_i) fixed for $i \geq 2$. For clarity, the unique symmetric pdf $f_{sym}(\xi_1, ..., \xi_N) = \frac{1}{N!}\sum_{\pi\in\Pi_N} f(\xi_{\pi(1)}, ..., \xi_{\pi(N)})$ is used in the contour plots (see Remark 15). For explicit computation, we choose the probability mass function

$$\nu(N) = \max\left\{\frac{N_\ell + 1 - |N_\ell - N|}{(N_\ell + 1)^2}, 0\right\} \tag{5.1}$$

when evaluating the lower density in Eq. (4.5), where $N_\ell = \left|\mathscr{D}^\ell\right|$ is the lower cardinality of the center diagram. This probability mass function is chosen to satisfy the requirements of Def. 24, and specifically has the property that $\nu(N) > 0$ for $0 \leq N \leq 2\left|\mathscr{D}^\ell\right|$.

Example 2 Consider the center persistence diagram $\mathscr{D} = \{(1, 3), (2, 4), (1, 1.3), (3, 3.2)\} \subset W$ and bandwidth $\sigma = 1/2$. We construct the associated kernel density $K_\sigma(Z, \mathscr{D})$ according to Theorem 25 and follow with some plots and analysis of the kernel density. The random persistence diagram D associated with the kernel density $K_\sigma(Z, \mathscr{D})$ has a variable number of features $N = |D|$; consequently, the input diagram $Z = \{\xi_1, ..., \xi_N\}$ must have variable length and therefore the kernel density has local definitions (see Rmk. 14) on W^N for each possible input cardinality N.

Since each modified Gaussian $p^{(j)}$ (Def. 22) and the lower density p^ℓ (Def. 24) integrate to 1 over the wedge W, an expression for the probability mass function (pmf) $\mathbb{P}[|D| = N]$ can be expressed solely in terms of ν and $q^{(j)}$:

$$\begin{aligned}\mathbb{P}[|D| = N] = &\left[q^{(1)}q^{(2)}\right]\nu(N-2)\\ &+\left[q^{(1)}\left(1-q^{(2)}\right)+q^{(2)}\left(1-q^{(1)}\right)\right]\nu(N-1)\\ &+\left[\left(1-q^{(1)}\right)\left(1-q^{(2)}\right)\right]\nu(N)\end{aligned} \tag{5.2}$$

The plot of this pmf is shown in Fig. 4. Recall that $D = D^u \cup D^\ell$, so that $|D| = |D^u| + \left|D^\ell\right|$; since $q^{(j)} \approx 1$ for $j = 1, 2$, $|D^u| = 2$ with high probability and the pmf $\mathbb{P}[|D| = N]$ is nearly the pmf for $\left|D^\ell\right|$, ν, shifted up by 2 units. Fig. 4 suggests that understanding the kernel density requires investigation into higher cardinality inputs. In general, it is important to consider input diagrams Z with $|Z| \geq |\mathscr{D}^u|$.

First, we describe the random diagram associated to the lower features $\mathscr{D}^\ell = \{(1, 1.3), (3, 3.2)\}$ of the center diagram $\mathscr{D}$. The lower random diagram D^ℓ is described in Def. 24 according to a probability mass function (pmf) ν for the cardinality of D^ℓ and a single probability density $p^\ell(b, d)$ for the subsequent features' locations in the wedge W. The pmf ν is defined according to Eq. (5.1) with $N_\ell = 2$; that is, $\nu(\{0, 1, 2, 3, 4\}) = \{1/9, 2/9, 3/9, 2/9, 1/9\}$ respectively, and zero otherwise.

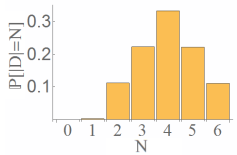


Figure 4: Cardinality probabilities $\mathbb{P}[|D| = N]$ for random diagram D distributed according to global pdf $K_\sigma(\cdot, \mathscr{D})$ in Ex. 2. In general, we have that $0 \leq |D^u| \leq |\mathscr{D}^u|$ and according to Eq. (5.1), $\nu(N) \neq 0$ for $0 \leq N \leq 2\left|\mathscr{D}^\ell\right|$. Thus, the cardinality $|D| = |D^u| + \left|D^\ell\right|$ takes on values between 0 and $6 = |\mathscr{D}^u| + 2\left|\mathscr{D}^\ell\right|$.

Following Def. 24, we project the features of $\mathscr{D}^\ell$ onto the diagonal to obtain $\{(1.15, 1.15), (3.1, 3.1)\}$. Relying on Eq. (4.4), the resulting lower density is given by

$$p^\ell(b, d) = \frac{2}{\pi}\left[e^{-\left((b-1.15)^2+(d-1.15)^2\right)} + e^{-\left((b-3.1)^2+(d-3.1)^2\right)}\right]. \tag{5.3}$$

restricted to the wedge W. The coefficient $\frac{2}{\pi}$ is obtained by a direct substitution into Eq. (4.4).

Due to the flexible input cardinality, the kernel will be expressed and plotted separately for different input cardinalities. For brevity, we present the local kernels on $W^N \subset \mathbb{R}^{2N}$ for cardinalities $N = 1, 2, 3$. First, we consider the probability hypothesis density (or PHD, as defined in Eq. (3.8)) along with the kernel density evaluated at a single input feature in Fig. 5. Recall that the integral of the PHD over a region U yields the expected number of features in U (see definition 18). The kernel's corresponding PHD is a sum of Gaussians as described in Corollary 28.

$$\begin{aligned}K_{\sigma,PHD}((b, d), \mathscr{D}) &= 2p^\ell(b, d) + q^{(1)}p^{(1)}(b, d) + q^{(2)}p^{(2)}(b, d)\\ &= 1.273\left(e^{-2\left((b-3.1)^2+(d-3.1)^2\right)} + e^{-2\left((b-1.15)^2+(d-1.15)^2\right)}\right)\\ &\quad + 0.635e^{-2\left((b-2)^2+(d-4)^2\right)} + 0.635e^{-2\left((b-1)^2+(d-3)^2\right)}.\end{aligned} \tag{5.4}$$

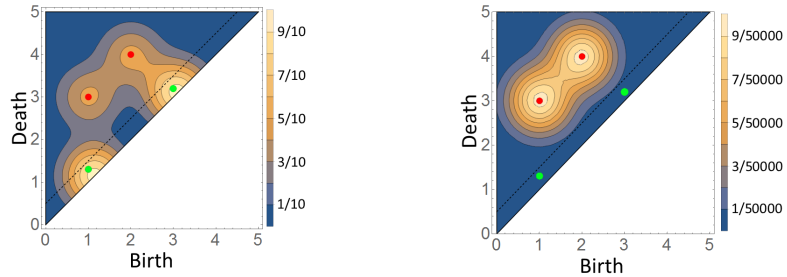
Next, for input of cardinality $|Z| = 1$, we obtain an easily viewable 2-dimensional distribution. Theorem 25 yields the following expression:

$$\begin{aligned}K_\sigma((b_1, d_1), \mathscr{D}) &= \nu(0)\left[(1-q^{(2)})q^{(1)}p^{(1)}(b_1, d_1) + (1-q^{(1)})q^{(2)}p^{(2)}(b_1, d_1)\right]\\ &\quad + \nu(1)\left[(1-q^{(1)})(1-q^{(2)})p^\ell(b_1, d_1)\right].\\ &= 7.74\times 10^{-2}\left(e^{-2\left((b_1-2)^2+(d_1-4)^2\right)} + e^{-2\left((b_1-1)^2+(d_1-3)^2\right)}\right).\\ &\quad + 1.65\times 10^{-4}p^\ell(b_1, d_1).\end{aligned} \tag{5.5}$$

The kernel is treated as a global pdf as in Proposition 13 and Rmk. 14; thus, this 2-D density is only a local density for the whole kernel. Each term is a weighted product of the combination of upper features considered (In order: $(2, 4)$, $(1, 3)$, or none.). Since the values of $q^{(j)}$ are very close to 1, terms which include the upper pdfs $p^{(j)}$ have much larger total mass.

Contour plots of the densities expressed in Eqs. (5.4) and (5.5) (restricted to W) are respectively shown in Figs. 5(a) and 5(b). In Fig. 5(a), the PHD indicates that in general, as many features

will appear near the diagonal as will appear near the upper features. According to the local kernel shown in Fig. 5(b), if only a single feature is present, this feature is far more likely to have long persistence. Indeed, the kernel density is defined (see Eq. (4.6)) so that the number of points near the diagonal is fluid (by our choice of ν), whereas the probability of each feature in the upper diagram is nearly 1. In essence, this demonstrates that the kernel density naturally considers features with long persistence to be stable or prominent in density estimation.



(a) (b)

Figure 5: Contour maps for (a) the probability hypothesis density associated to the kernel density (Eq. (5.4)) and (b) the kernel density restricted to a single input feature (Eq. (5.5)). The center diagram is indicated by red (upper) and green (lower) points. Scale bars at the right of each plot indicate the range of probability density in each shaded region.

Taking $Z = (\xi_1, \xi_2) = ((b_1, d_2), (b_2, d_2))$, we arrive at a more complex expression for the kernel density when considering 2 input features. From Eq. (4.6), we obtain:

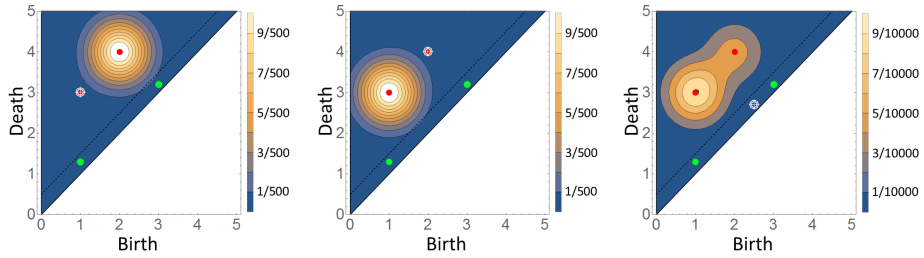
$$\begin{aligned} K_\sigma((\xi_1,\xi_2),\mathscr{D}) &= \nu(0)q^{(1)}q^{(2)}p^{(1)}(b_1,d_1)p^{(2)}(b_2,d_2) \\ &\quad + \nu(1)\left[(1-q^{(2)})q^{(1)}p^{(1)}(b_1,d_1) + (1-q^{(1)})q^{(2)}p^{(2)}(b_1,d_1)\right]p^\ell(b_2,d_2) \\ &\quad + \nu(2)(1-q^{(1)})(1-q^{(2)})p^\ell(b_1,d_1)p^\ell(b_2,d_2) \\ &= 4.5\times 10^{-2}e^{-2\left((b_1-2)^2+(d_1-4)^2\right)}e^{-2\left((b_1-1)^2+(d_1-3)^2\right)} \\ &\quad + 2.11\times 10^{-4}\left[e^{-2\left((b_1-2)^2+(d_1-4)^2\right)} + e^{-2\left((b_1-1)^2+(d_1-3)^2\right)}\right]p^\ell(b_2,d_2) \\ &\quad + 7.39\times 10^{-7}p^\ell(b_1,d_1)p^\ell(b_2,d_2). \end{aligned} \tag{5.6}$$

Notice that this local kernel also decomposes into terms which describe presence of upper features: one term for both, one term for each of the two upper features, and the last term has no upper features. Contour plots of slices of this local kernel are shown in Fig. 6; a general description of slicing is given in Rmk. 41.

Remark 41 *Slices are used to view local pdfs defined on a high dimensional space $W^N \subset \mathbb{R}^{2N}$ for $N > 1$. To obtain these slices, one fixes features $(b_j, d_j) = (b'_j, d'_j)$ for $j = 2, ..., N$, and views the density on the corresponding hyperplane $W \times \{(b'_2, d'_2)\} \times ... \times \{(b'_N, d'_N)\} \subset W^N$. In practice, the fixed features are chosen as modes of earlier (smaller N) slices in order to view important parts of*

the distribution. We also sum over possible permutations in order to view a slice of the symmetric pdf, as was done for Ex. 1.

If we consider the density evaluated along slices as $K_\sigma\left(((b,d),(1,3)),\mathscr{D}\right)$ or $K_\sigma\left(((b,d),(2,4)),\mathscr{D}\right)$ (Fig. 6 (a) or (b), respectively), the restricted plot is a Gaussian centered at the other upper feature. If the fixed feature is instead close to the diagonal, as in Fig. 6 (c), the density slice is close to a mixture between the two upper Gaussians $p^{(1)}$ and $p^{(2)}$.



(a) (b) (c)

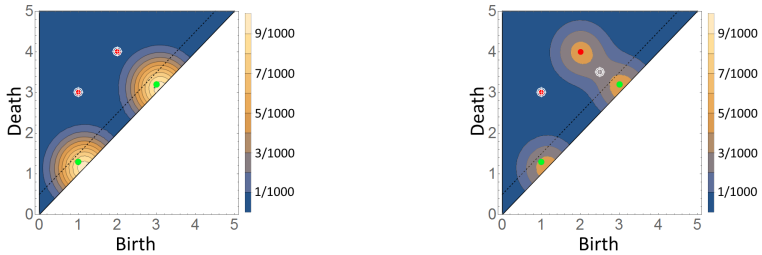
Figure 6: Contour maps for slices of the kernel density $K_\sigma((\xi,\xi'_2),\mathscr{D})$ with input cardinality 2. A single feature ξ'_2, indicated by white crosshairs, is fixed to restrict to a 2D subspace as follows: (a) $\xi'_2 = (1,3)$ (b) $\xi'_2 = (2,4)$ and (c) $\xi'_2 = (2.5, 2.7)$. The center diagram is indicated by red (upper) and green (lower) points. Scale bars at the right of each plot indicate the range of probability density in each shaded region.

In a similar fashion, we also express the kernel density with input cardinality $|Z| = 3$. Since there are only 2 upper features in $\mathscr{D}$, this and further expressions are not markedly more complicated than Eq. (5.6). From Eq. (4.6), we obtain:

$$\begin{aligned} K_\sigma((\xi_1,\xi_2,\xi_3),\mathscr{D}) &= \nu(1)\left[q^{(1)}q^{(2)}p^{(1)}(b_1,d_1)p^{(2)}(b_2,d_2)\right]p^\ell(b_3,d_3) \\ &\quad + \nu(2)(1-q^{(2)})q^{(1)}p^{(1)}(b_1,d_1)p^\ell(b_2,d_2)p^\ell(b_3,d_3) \\ &\quad + \nu(2)(1-q^{(1)})q^{(2)}p^{(2)}(b_1,d_1)p^\ell(b_2,d_2)p^\ell(b_3,d_3) \\ &\quad + \nu(3)(1-q^{(1)})(1-q^{(2)})p^\ell(b_1,d_1)p^\ell(b_2,d_2)p^\ell(b_3,d_3). \\ &= 9.01\times 10^{-2}p^\ell(b_3,d_3)e^{-2\left((b_1-1)^2+(d_1-3)^2\right)}e^{-2\left((b_2-2)^2+(d_2-4)^2\right)} \\ &\quad + 4.96\times 10^{-4}p^\ell(b_2,d_2)p^\ell(b_3,d_3)e^{-2\left((b_1-2)^2+(d_1-4)^2\right)} \\ &\quad + 4.96\times 10^{-4}p^\ell(b_2,d_2)p^\ell(b_3,d_3)e^{-2\left((b_1-1)^2+(d_1-3)^2\right)} \\ &\quad + 1.22\times 10^{-6}p^\ell(b_1,d_1)p^\ell(b_2,d_2)p^\ell(b_3,d_3). \end{aligned} \tag{5.7}$$

One may notice that Eq. (5.7) has the same 4 terms as Eq. (5.6), but with another factor of p^ℓ in each term. Indeed, the local kernels for input cardinality $N = 4, 5, 6$ appear very similar as well, and with progressively more factors of p^ℓ. Contour plot slices of this local kernel are shown in Fig. 7, following Rmk. 41. In this case, since the local pdf is defined in W^3, we must fix a pair of features in order to view a slice in $W \times \{(b'_2, d'_2)\} \times \{(b'_3, d'_3)\}$. In Eq. (5.7), the heaviest

weighted term consists of both upper features' densities as well as the lower density $p^{\ell}(b_3, d_3)$. Indeed, Fig. 7(a) shows the slice $K_\sigma(((b,d),(1,3),(2,4)),\mathscr{D})$, which leaves both upper features fixed, and the resulting slice is nearly proportional to the lower density p^{ℓ}. Fig. 7 (b) shows the slice $K_\sigma(((b,d),(1,3),(2.5,3.5)),\mathscr{D})$, which fixes one of the upper features of $\mathscr{D}$ as well as a feature of moderate persistence. This slice does not go through a mode of the local kernel, and so the geometry of the dataspace W^3/Π_3 makes the slice look multi-modal, depending on whether $(2.5, 3.5)$ is assigned to $p^{(2)}$ or p^{ℓ}. Other assignments have negligible mass. Thus, Fig. 7 (b) resembles a mixture of these two densities.



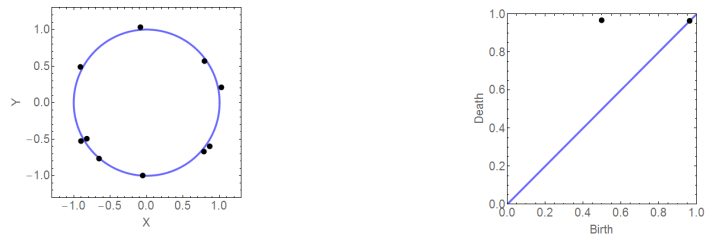
(a) (b)

Figure 7: Contour maps for slices of the kernel density $K_\sigma((\xi,\xi_2',\xi_3'),\mathscr{D})$ with input cardinality 3. A pair of features ξ_2' and ξ_3', indicated by white crosshairs, are fixed to restrict to a 2D subspace as follows: (a) $(\xi_2',\xi_3') = ((1,3),(2,4))$ and (b) $(\xi_2',\xi_3') = ((1,3),(2.5,3.5))$.
Since the symmetric version of the density is used, the order of these features is irrelevant. The center diagram is indicated by red (upper) and green (lower) points. Scale bars at the right of each plot indicate the range of probability density in each shaded region.

The terms $(1 - q^{(k)})$ within the $\mathcal{Q}^*$ expression (see Eq. (3.11)) are very small and appear in terms for which the corresponding upper feature is unassigned. These terms are so small because both upper features have very long persistence in this example (four times the bandwidth), and so the terms in Eqs. (5.5), (5.6), and (5.7) which do not include one or both upper Guassians $p^{(1)}$ and $p^{(2)}$ have progressively smaller contribution to the overall local kernel. Consequently, the kernel places much higher probability density near input diagrams with features nearby each upper feature in the center diagram. This behavior is seen in Fig. 5, 6, 7, and their respective analyses, and is directly correlated to the ratio of persistence to bandwidth for each feature.

Example 3 Here we consider the random persistence diagram generated from a specific random dataset in $\mathbb{R}^2$. Our goal in this example is to build and demonstrate convergence of the kernel density estimate for the pdf of the associated random persistence diagram. Specifically, we generate sample datasets which each consist of 10 points sampled uniformly from the unit circle with additive Gaussian noise, $N((0,0),\left(\frac{1}{50}\right)^2 I_2)$. This toy dataset is prototypical for signal analysis (corresponding to the circular dynamics of a noisy sine curve), wherein the high dimensional point cloud is obtained through delay-embedding of the signal. An in-depth analysis of using delay embedding alongside persistent homology is found in (Perea and Harer, 2015).

These datasets each yield a Čech persistence diagram as described in Section 2 for degree of homology $k = 1$. A sample dataset and its associated $k = 1$ persistence diagram are shown in Fig.



(a) (b)

Figure 8: An example underlying dataset and its associated persistence diagram. The persistence diagrams are used as the centers for the kernel density estimate. For this example, persistence diagrams with more than one feature are relatively rare.

KDE	(1)	(2)	(3)	(4)
n	100	300	1000	5000
σ	0.03	0.025	0.020	0.015

Table 1: Choices of sample size n (number of persistence diagrams) and bandwidth σ for each kernel density estimate $\hat{f}_{n,\sigma}(Z)$ shown in Fig. 9.

8. Since these datasets are sampled from the unit circle perturbed by relatively small noise, one expects the associated 1-homology to have a single persistent feature with $d \approx 1$ with possible brief features caused by noise.

We consider several KDEs as we simultaneously increase the number of persistence diagrams (n) and narrow the bandwidth (σ) as shown in Table 1). The bandwidth was chosen to scale according to Silverman's rule of thumb (Silverman, 1986) (see Rmk. 32).

Since the KDEs $\hat{f}_{n,\sigma}(Z)$ are defined on $\bigcup_N W^N$ for several input cardinalities N, we present them in multiple slices by fixing a cardinality and then fixing all but one input feature as described in Rmk 41. For example, $g(\xi) = \hat{f}_{n,\sigma}(\xi,\xi_2',...,\xi_N')$ for fixed ξ_j' $(j = 2,...,N)$ is a function on W and represents a slice of the local KDE on W^N. The progression of KDE slices can be seen in Fig. 9, wherein the same slices (i.e., the same features are fixed) are viewed for each choice of (n,σ). These plots demonstrate in practice the convergence of the kernel density estimator shown in Theorem 1. Because the sample points for the underlying dataset lie so close to the unit circle, one expects the topological feature to die near scale $d = 1$, as is reflected in the KDEs shown in Fig. 9 (left); however, the distribution of points along the circle allows its birth scale to vary quite a lot. Additional features with brief persistence are concentrated very close to the diagonal due to small noise. These features tend to be either spurious holes near the edge (smaller b and d) or a short split of the main topological loop in two (larger b and d); this behavior is reflected in the two peaks for slices of the KDEs shown in Fig. 9 (right). Indeed, the persistence diagram shown in Fig. 8 is typical for this example. Overall, by scanning from top to bottom, Fig. 9 demonstrates the convergence of the KDEs as n increases and σ decreases. The location and mass of each mode is as expected from underlying data sampled from the unit circle. Moreover, very small spread in

the limiting density arises from the small noise in the underlying data. The shape and spread of each mode converges, and the densities for $n = 1000$ and $n = 5000$ are nearly the same.

Example 4 Electroencephalography (EEG) monitors electrical activity in the brain by measuring changes in voltage over time at particular locations on the scalp. To obtain data, researchers place arrays of electrodes on subjects' heads that record fluctuations in voltage as time series. It is known that EEG in the 1-100 Hz range is heavily involved in cognition (Boothe et al., 2017); moreover, specific bands in the 1-100 Hz range are hypothesized to be associated with certain tasks or brain states. For example, alpha range EEG, which has a spectral peak in the 8-12 Hz range, is thought to be important in inhibition and excitation during decision problems (Kilmesch, 2012). Collecting EEG is a noninvasive procedure, however, measurements are often obscured by noise arising from electrical activity in the environment, movement, or other physiological processes like heartbeat. Consequently, a critical problem is the need to detect EEG with the same underlying dynamics, e.g. EEG that is predominantly composed of 8-12 Hz oscillations, in the presence of varying levels of noise (Repovs, 2010).

In this example, we consider the widely–used autoregressive EEG model introduced in (Franaszczuk and Bilnowaska, 1985), and we employ the KDE established in Proposition 25 to statistically analyze EEG. The authors in (Franaszczuk and Bilnowaska, 1985) model an EEG time series $(x_{t_i})_{i=0}^L$ of time length L as a convolution of white noise with a linear filter function given in Eq. (5.8),

$$h(t_i) = \sum_{j=1}^{p} e^{-\beta_j t_i} \cos(\omega_j t_i), \tag{5.8}$$

where ω_j correspond to centers of peaks in the power spectral density of $(x_{t_i})_{i=0}^L$ while the parameters β_j are approximately equal to 1/2 of their respective widths (both ω_j and β_j are given in Hz). Recall that the power spectral density describes the contribution of each frequency to the total power of $(x_{t_i})_{i=0}^L$ after decomposing $(x_{t_i})_{i=0}^L$ into a series of oscillatory functions. For example, a power spectral density with a narrow peak at 10 Hz corresponds to a time series that heavily resembles a function oscillating at 10 Hz. A broader peak at 10 Hz in essence means that the time series has a greater contribution from more frequencies surrounding 10 Hz, diminishing the resemblance (for comparison, the power spectral density of white noise is completely flat). This view of the power spectral density means one can effectively simulate EEG comprised of oscillations in a desired band of frequencies by selecting appropriate parameters in Eq. (5.8).

We focus on alpha range (8-12 Hz) EEG. Specifically, we simulate 200 EEG signals in the alpha range by first generating 200 white noise vectors of length $L = 1,024$ through independent draws from $\mathcal{N}(0,1)$ then convolving them with the linear filter described by Eq. (5.8) with $p = 1, \beta_1 = 3.7$, and $\omega_1 = 10.5$. We corrupt 100 signals by additive noise $\mathcal{N}(0, 10^{-1/20})$, while the rest are corrupted by $\mathcal{N}(0, 10^{-5/20})$. This yields two collections of EEG signals with signal-to-noise ratios (SNRs) of 1 and 5, denoted by SNR_1 and SNR_5, respectively.

Next, we convert each EEG signal into a persistence diagram using the methodology of (Perea and Harer, 2015). Namely, we transform EEG signals to point clouds in $\mathbb{R}^2$ using delay embeddings where the delay parameter was determined by the sampling rate (100 Hz) along with the dominant underlying frequency of the signals (10 Hz); we then center and scale the point clouds by their variances along the vertical and horizontal axes; see Fig. 10 (e) and (f). Once we obtain point clouds, we compute persistence diagrams for 1-dimensional homological features using Rips

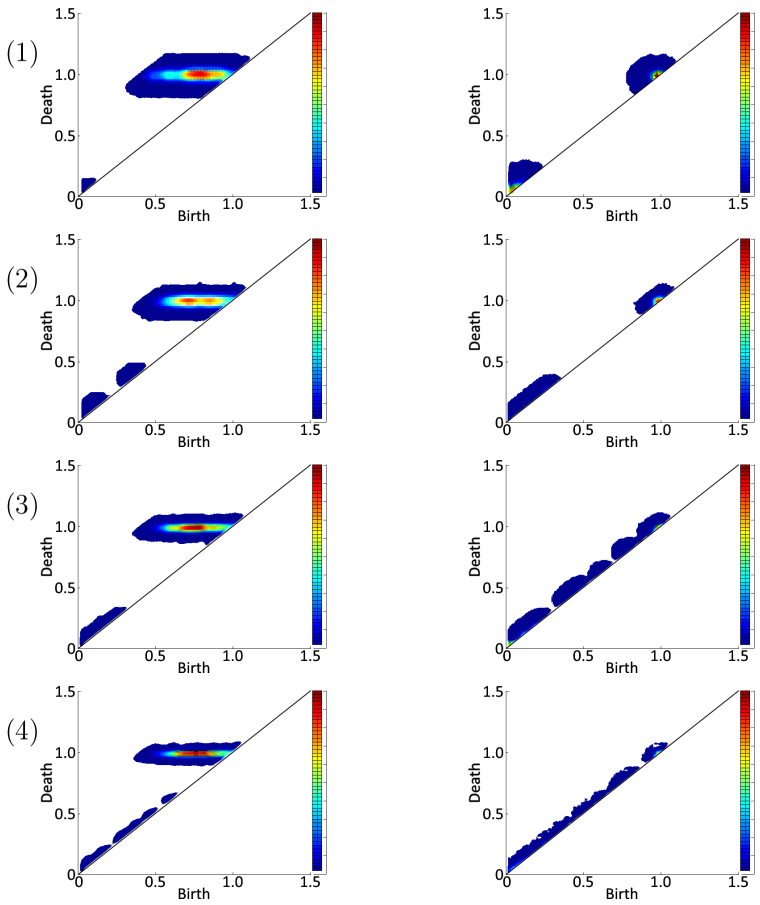


Figure 9: Plots of persistence diagram KDEs for Ex. 3. Each plot is presented as a heat map where color indicates the probability density. White regions above the diagonal indicate portions of very low probability density. Each column is a particular slice, while each row is a particular global KDE with n and σ as indicated in Table 1. The left column are the local KDEs $\hat{f}_{n,\sigma}((b,d))$ evaluated at a diagram with only one feature. The mode of the converged density is approximately $(b_2', d_2') = (0.77, 0.98)$. The right column are the local KDEs $\hat{f}_{n,\sigma}((b,d),(0.77,0.98))$ evaluated at a diagram with two features and one feature fixed. These slices have two modes which are very close to the diagonal at $(0,0)$ and $(1,1)$. Overall, this figure demonstrates KDE convergence.

filtrations; see Section 2. We choose to focus solely on 1-dimensional homological features since they relate to periodicity in the underlying time series, which is the defining characteristic of our signals.

Denote the family of persistence diagrams created from SNR_i for $i = 1, 5$ by D_{SNR_i}, and let f_{SNR_i} be their global probability densities. Our goal is to verify that SNR_1 and SNR_5 EEG have the same underlying dynamics. A sensible strategy is to select a quantity created from persistence diagrams that is robust to noise, approximate its distribution for SNR_i using D_{SNR_i}, and then compare the two empirical distributions. To this end, we start by approximating f_{SNR_i} with the kernel density estimators $\hat{f}_{SNR_i}(Z) := 10^{-2} \sum_{\mathscr{D} \in D_{SNR_i}} K_\sigma(Z, \mathscr{D})$. For each fixed $i = 1, 5$ the persistence diagrams $\mathscr{D} \in D_{SNR_i}$ are the 100 diagrams of each SNR_i case. For the noise likelihood model related to the lower part of a persistence diagram, $\mathscr{D}^\ell$, we use Eq. (5.1) as the cardinality distribution. Given that features with higher persistence generally describe global topology that is more resilient to noise, and relying on these kernels $\hat{f}_{SNR_i}$, we take $S = 1,000$ sample persistence diagrams and compute their bottleneck distance $W_\infty(\emptyset, S_i^j) = \max_{(b,d) \in S_i^j} d - b$, where S_i^j is the jth sample persistence ($j = 1, \ldots, S$) diagram distributed according to $\hat{f}_{SNR_i}$, $i = 1, 5$. These distances create empirical distributions, one for each SNR_i EEG denoted by F_{SNR_i}. We formally proceed with hypothesis testing

$$H_0 : F_{SNR_1} = F_{SNR_5} \text{ vs } H_1 : F_{SNR_1} \neq F_{SNR_5}.$$

Failure to reject H_0 in this case is evidence that D_{SNR_1} and D_{SNR_5} have similar behavior for the features less affected by noise, which in turn implies that SNR_1 and SNR_5 have similar underlying dynamics. Finally, we compare these distributions with a two-sided Kolmogorov-Smirnov (KS) Test (Simard, 2011) that yields a $p-$value=0.72.

	KS-Test P-value	Time (s)
KDE MP	0.72	0.047
PI L_∞	6.15×10^{-9}	0.042
PL L_∞	0.79	0.048

Table 2: The p-values and run times for each method (KDE, PI, and PL) used for the hypothesis test of Eq. (4).

	Sample MAD
SNR_1	1.040
SNR_5	1.035

Table 3: The sample MADs for SNR_1 and SNR_5 computed by taking the means of the distributions in Fig 11(e) and Fig 11(f), repsectively.

For the sake of comparison to other TDA methods, we also compute persistence images (PIs) with resolution 50×50 and spread 0.2 using the ramp function to produce weights, (Adams et al., 2017), and persistence landscapes (PLs) from D_{SNR_i}, (Bubenik, 2015). We examine the L_∞-norm as a summary for each of these vectorizations (the L_∞-norm of the first landscape in particular for PLs) since this measurement is also associated with high persistence features. After computing L_∞-norms for each of the PIs and PLs obtained from D_{SNR_i}, we resample each L_∞ empirical distribution 1,000 times to create bootstrapped distributions with size matching those of the W_∞

distributions obtained from the kernel density estimators; see Fig. 11 (c),(d),(e), and (f). In the end, we also compare the bootstrapped distributions with a two-sided KS-test. Table 2 shows the KS-test p-values and a standardized run time for each method.

Notice the kernel density max persistence and landscape L_∞ correctly fail to reject H_0 at the most commonly used significance levels ($p-$value = 0.79). In particular, our method is competitive with landscapes (with a slight edge on computational time). On the other hand, the persistence image L_∞ incorrectly rejects H_0 ($p-$value close to 0). Failure of PIs to recognize different dynamics may be a result of the fact that in addition to accounting for the max persistence (through the use of the ramp function for weights), the PI L_∞ also considers the cardinality of each diagram, although the contribution of cardinality diminishes for higher resolutions and smaller spreads.

Finally, we report estimates for $MAD_{f_{SNR_1}}(\emptyset)$ and $MAD_{f_{SNR_5}}(\emptyset)$ by taking the means of our empirical distributions for the max persistences; see Table 3. Notice the estimates are very close numerically and by appealing to Theorem 40, one could argue they are close to their true values. Hence, the MAD offers more evidence that SNR_1 and SNR_5 are statistically indistinguishable.

6. Discussion and Conclusions

A nonparametric approach to approximating density functions of finite random persistence diagrams has been presented. This includes the introduction of a kernel density function, as well as proof that the kernel density itself and its mean absolute deviation converge to those of the target distribution. Our kernel density function arises by creating a noise model for persistence diagrams which simplifies treatment of features near the diagonal. Consequently, the kernel density can also be used in measuring likelihood when observing small perturbations of persistence diagrams. Future work will investigate the convergence of powers of the absolute deviation (e.g., bottleneck variance) and deviations involving the Wasserstein metric (an L^p generalization of bottleneck metric, see (Edelsbrunner and Harer, 2010)). Our framework is presented through the lens of geometric simplicial complexes, and in particular Čech complexes. The resulting persistence diagrams are based on underlying datasets in a metric space. In general, one may define persistent homology for a function f defined on a topological space (Edelsbrunner and Harer, 2010), and therefore random functions may also give rise to random persistence diagrams, see (Adler et al., 2010) for an example. A similar kernel density estimate approach can be formulated in this case, but perhaps different assumptions may be needed on the target pdf.

Our approach is fully data-driven, a necessary step since distributions of persistence diagrams were previously poorly understood. The assumptions $(A1)$-$(A3)$, $(A2)^*$, and $(A3)^*$ are typical for kernel density estimators (Scott, 2015). Similar assumptions on the underlying data are inherited by the random persistence diagram, because variation in Čech persistent homology is controlled by interpoint distances. In particular, probability density decay follows the same trends as noise in the underlying data; this is seen in Fig. 9 (a) for Gaussian noise. Thus, the kernel density estimates defined here can be reliably used for data analysis, adding a detailed tool to the methods used in topological data analysis. In particular, this is the first result yielding probability density functions which directly analyze the full distribution information of a random persistence diagram. For applications in machine learning such as classification, the kernel density estimates carry information for generating more sophisticated features than previously available; e.g., the value of the global pdf at a specific input or list of inputs or the integral of the global pdf over a specified region.

Access to a pdf also provides a tool with which one can check for classification robustness in terms of likelihood or Bayes factors, providing a measure of the confidence in a particular outcome.

Lending credence to applicability in data analysis, examples of kernel density estimation are presented in Section 5. In one example, underlying datasets are generated to lie on the unit circle with additive noise, a prototypical example for topological data analysis. Our analysis yields detailed information about the distribution of diagrams, even though only two $2-$dimensional slices of the kernel density estimate are shown. This example demonstrates the convergence of the kernel density estimator in practice for large enough sample size (number of persistence diagrams). This example along with the supplementary examples also demonstrate the detailed information contained in a persistence diagram KDE. Moreover, our example with EEG data verifies that our kernel density estimator is accessible to data in a realistic setting while remaining competitive with or outperforming pre-existing TDA techniques.

In the context of Fig. 3, it is clear that sampling from the kernel density is straightforward, and in fact computation time scales linearly in the number of features in the center diagram $\mathscr{D}$. In contrast, precise evaluation of the kernel global pdf at a diagram requires the more thorough computations shown in Eq. (4.6). This evaluation is made tractable due to the separation of the center diagram into upper and lower portions: $\mathscr{D} = \mathscr{D}^u \cup \mathscr{D}^\ell$ as described in Eq. (4.1). In practice, diagrams should split so that $|\mathscr{D}^u|$ is small while $|\mathscr{D}^\ell|$ is large. Evaluation of individual feature pdfs on the multi-wedge $\mathcal{W}_{0:\mathrm{cl}-1}$ only scales quadratically on the cardinality $|\mathscr{D}|$ and higher degree calculations are required only for combinatorics on the large persistence features in the upper diagram $\mathscr{D}^u$. Consequently, these calculations are tractable so long as $\mathscr{D}^u$ does not grow too much in cardinality, while an increased cardinality for $\mathscr{D}^\ell$ has a lesser effect on computation time.

The kernel density presented here treats the small persistent features in D^ℓ as a single group. Since convergence (Theorem 31) requires very little structure in the lower random diagram, it may be helpful in practice to cluster the lower portion of the center diagram, followed by defining a random diagram centered at each cluster. This approach somewhat complicates the expression and evaluation of the kernel density, but does not complicate sampling from the kernel density. The goal of this approach is to more carefully capture the geometric features of the underlying random dataset, since such geometric features often correspond to briefly persistent homological features. For example, geometric features are of paramount importance for classifying periodic signals through their delay embeddings, wherein the large persistent feature indicates periodicity and thus is expected to appear in every class.

Acknowledgements

The authors would like to thank 3 anonymous reviewers for their comments, which improved the manuscript significantly. Moreover, we would like to thank Drs. David Boothe, Piotr Franasczcuk, and Jason Sherwin for their useful insight and discussions about EEG data. Research has been partially supported by the Army Research Office (ARO) Grant # W911NF-17-1-0313, the National Science Foundation (NSF) Grant # DMS-1821241, and the Thor Industries/Army Research Lab (ARL) Grant # W911NF-17-2-0141.

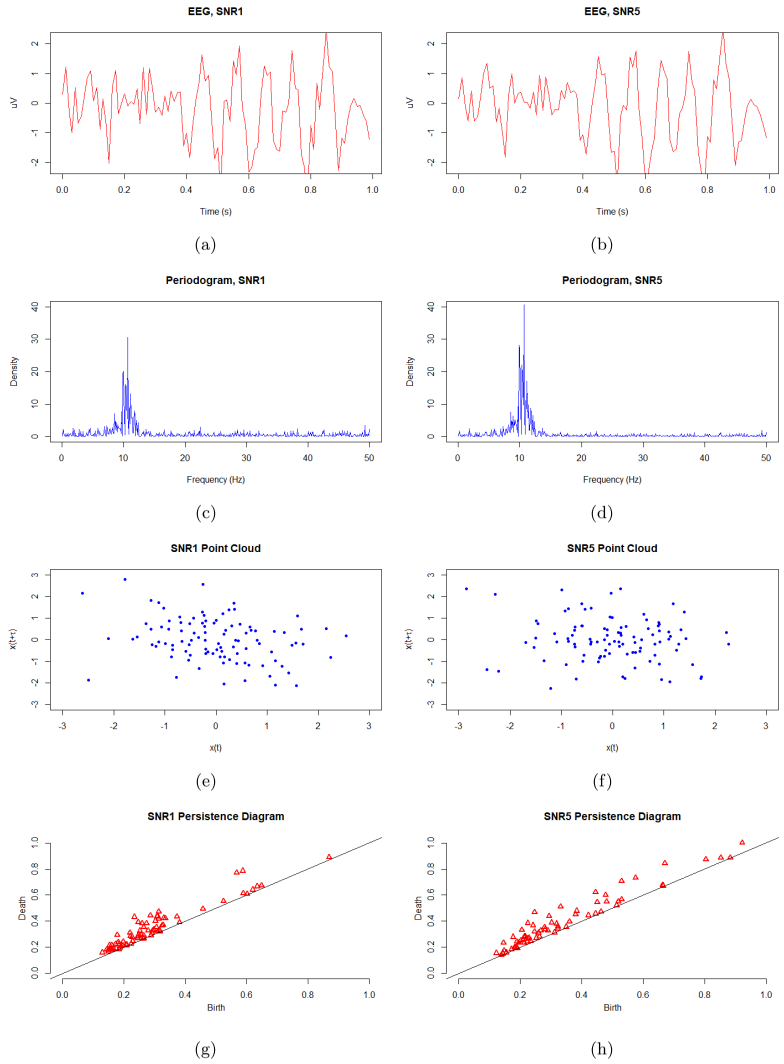


Figure 10: A segment of an EEG signal with (a) SNR_1, and (b) SNR_5, respectively, along with (c) and (d): their corresponding periodograms (estimates of the power spectral densities). The associated point clouds are given in (e) and (f), respectively, and (g) and (h) their resulting persistence diagrams.

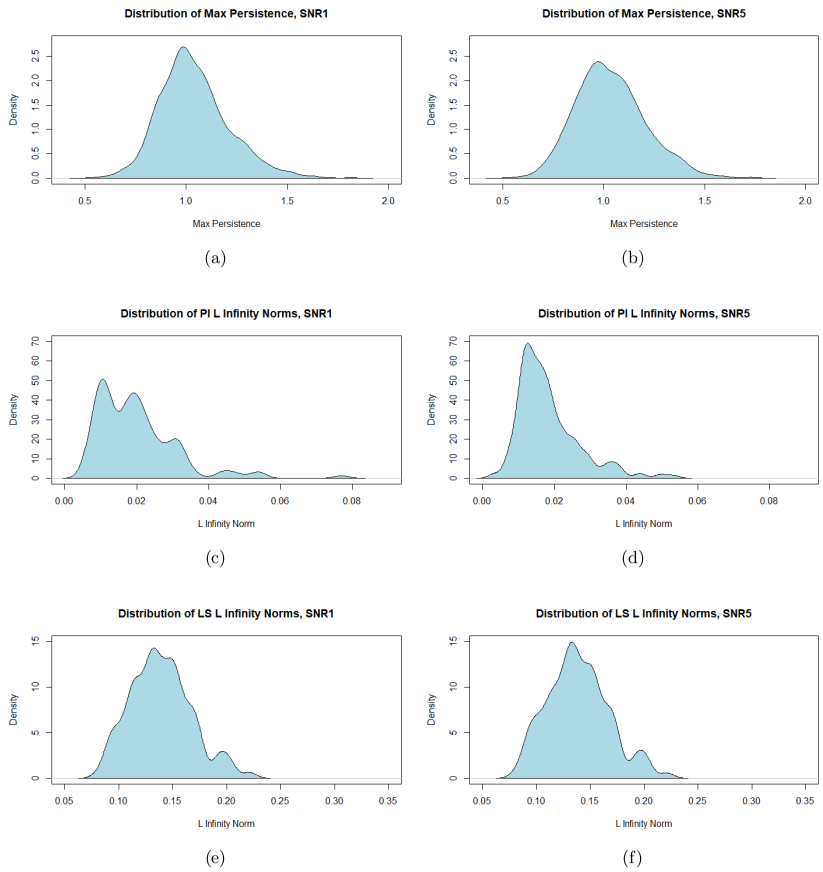


Figure 11: This figure shows the distribution for each statistic we considered when comparing SNR_1 to SNR_5. Each column represents a class, either SNR_1 or SNR_5, and each row a particular statistic.

References

ADAMS, H., EMERSON, T., KIRBY, M., NEVILLE, R., PETERSON, C., SHIPMAN, P., CHEPUSHTANOVA, S., HANSON, E., MOTTA, F., AND ZIEGELMEIER, L.(2015). Persistence images: A stable vector representation of persistent homology *The Journal of Machine Learning Research* 18(1):218–252.

ADCOCK, A., CARLSSON, E., AND CARLSSON, G. (2016). The ring of algebraic functions on persistence bar codes. *Homology, Homotopy and Applications*, 18(1):381–402.

ADLER, R. J., BOBROWSKI, O., BORMAN, M. S., SUBAG, E., AND WEINBERGER, S. (2010). Persistent homology for random fields and complexes. In *Borrowing strength: theory powering applications–a Festschrift for Lawrence D. Brown* (pp. 124–143). Institute of Mathematical Sciences.

ADLER, R. J., BOBROWSKI, O., AND WEINBERGER, S. (2014). Crackle: The homology of noise. *Discrete & Computational Geometry*, 52(4):680–704.

ADLER, R. J., AGAMI, S., AND PRANAV, P. (2017). Modeling and replicating statistical topology, and evidence for CMB non-homogeneity. *arXiv:1704.08248*.

ATIENZA, N., GONZALEZ-DIAZ, R., AND RUCCO, M. (2016). Separating topological noise from features using persistent entropy. In *Federation of International Conferences on Software Technologies: Applications and Foundations* (pp. 3–12). Springer International Publishing.

BAUER, U. (2015). Ripser. https://github.com/Ripser/ripser.

BENDICH, P., MARRON, J. S., MILLER, E., PIELOCH, A., AND SKWERER, S. (2016). Persistent homology analysis of brain artery trees. *The Annals of Applied Statistics*, 10(1):198–218.

BOBROWSKI, O., MUKHERJEE, S., AND TAYLOR, J. E. (2014). Topological consistency via kernel estimation. *arXiv:1407.5272*.

BOOTHE, D., YU, A., KUDELA, P., ANDERSON, W., VETTEL, J., AND FRANASZCZUK, P.(2017). Impact of neuronal membrane damage on local field potential in large-scale simulation of cerebral cortex. *Frontiers in Neurology*, volume 8, pages 236

BUBENIK, P. (2015). Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research* 16(1):77-102.

CHAZAL, F., FASY, B., LECCI, F., MICHEL, B., RINALDO, A., AND WASSERMAN, L. (2015). Subsampling methods for persistent homology. In *International Conference on Machine Learning* (pp. 2143–2151).

CHAZAL, F. AND DIVOL, V. (2018). The density of expected persistence diagrams and its kernel based estimation. In *The Symposium of Computational Geometry 2018*

CHEN, C. AND KERBER, M. (2011). Persistent homology computation with a twist. In *Proceedings 27th European Workshop on Computational Geometry*, volume 11.

Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. (2007). Stability of persistence diagrams. *Discrete Comput. Geom*, 37:103–120.

Cohen-Steiner, D., Edelsbrunner, H., Harer, J., and Mileyko, Y. (2010). Lipschitz functions have l p-stable persistence. *Foundations of computational mathematics*, 10(2):127–139.

De Silva, V. and Ghrist, R. (2007). Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358.

Donato, I., Gori, M., Pettini, M., Petri, G., De Nigris, S., Franzosi, R., and Vaccarino, F. (2016). Persistent homology analysis of phase transitions. *Physical Review E*, 93(5), 052138.

Edelsbrunner, H. and Harer, J. (2010). *Computational topology: an introduction*. American Mathematical Society.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2002). Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533.

Edelsbrunner, H. (2013). Persistent homology in image processing. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 182–183. Springer, Berlin, Heidelberg.

Emmett, K., Rosenbloom, D., Camara, P., and Rabadan, R. (2014). Parametric inference using persistence diagrams: A case study in population genetics. *arXiv:1406.4582*.

Emrani, S., Gentimis, T., and Krim, H. (2014). Persistent homology of delay embeddings and its application to wheeze detection. *IEEE Signal Processing Letters*, 21(4):459–463.

Fasy, B. T., Kim, J., Lecci, F., Maria, C., Rouvreau., V., The included GUDHI is authored by Clement Maria, Dionysus by Dmitriy Morozov, P. b. U. B. M. K., and Reininghaus., J. (2015). Tda: Statistical tools for topological data analysis r package version 1.4.1.

Fasy, B. T., Lecci, F., Rinaldo, A., Wasserman, L., Balakrishnan, S., and Singh, A. (2014). Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339.

Franaszczuk, P. and Bilnowaska, K. Linear model of brainactivity-EEG as a superposition of damped oscillatory modes *Biological Cybernetics* 53:19-25 Spring-Verlag

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA.

Goodman, I. R., Mahler, R. P., and Nguyen, H. T. (2013). *Mathematics of data fusion*, volume 37. Springer Science & Business Media.

Guillemard, M. and Iske, A. (2011). Signal filtering and persistent homology: an illustrative example. *Proc. Sampling Theory and Applications (SampTA'11)*.

Hatcher, A. (2002). Algebraic topology. 2002. *Cambridge UP, Cambridge*, 606(9).

Kerber, M., Morozov, D., and Nigmetov, A. (2016). Geometry helps to compare persistence diagrams. *Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments*, pages 103–112.

Kilmesch, W.(2012). Alpha band oscillations, attention, and controlled access to stored information. *Trends in Cognitive Sciences*, 16(12):606-617

Kusano, G., Fukumizu, K., and Hiraoka, Y. (2016). Persistence weighted Gaussian kernel for topological data analysis. In *Proceedings of the* 33rd *International Conference on Machine Learning*.

Kwitt, R., Huber, S., Niethammer, M., Lin, W., and Bauer, U. Statistical topological data analysis- a kernel perspective. In *Advances in neural information processing systems*, pages 3070–3078.

Mahler, R. P. (1995). Unified nonparametric data fusion. In *SPIE's 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics*, pages 66–74. International Society for Optics and Photonics.

Mahler, R. P. (2003). Multi-target Bayes filtering via first-order multi-target moments. In *IEEE Trans. AES*, volume 37, pages 1152-1178.

Marchese, A. and Maroulas, V. (2016). Topological learning for acoustic signal identification. In *Information Fusion (FUSION), 2016 19th International Conference on*, pages 1377–1381.

Marchese, A. and Maroulas, V. (2018). Signal classification with a point process distance on the space of persistence diagrams. *Advances in Data Analysis and Classification*, 12 (3), pp. 657–682.

Marchese, A., Maroulas, V., and Mike, J. (2017). K-means clustering on the space of persistence diagrams. In *Wavelets and Sparsity XVII* (Vol. 10394, p. 103940W). International Society for Optics and Photonics.

Maroulas, V., Micucci, C., and Spannaus, A. (2018). A stable cardinality distance for topological classification. https://arxiv.org/abs/1812.01664.pdf

Maroulas, V., Nasrin, F., and Oballe, C. (2019). A Bayesian Framework for Persistent Homology https://arxiv.org/pdf/1901.02034.pdf

Matheron, G. (1975). *Random Sets and Integral Geometry*. John Wiley & Sons.

Mileyko, Y., Mukherjee, S., and Harer, J. (2011). Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12).

Munch, E. (2017). A user's guide to topological data analysis *Journal of Learning Analytics*, 4(2):47–61.

Perea, J. A. and Harer, J. (2015). Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838.

Pereira, C. M. and de Mello, R. F. (2015). Persistent homology for time series and spatial data clustering. *Expert Systems with Applications*, 42(15):6026–6038.

Reininghaus, J., Huber, S., Bauer, S., and Kwitt, R. (2014). A stable multi-scale kernel for topological machine learning. *arXiv:1412.6821*

Repovs, G.(2010). Dealing with noise in EEG recording and data analysis. *Informatica Medica Slovenica*, 15(1):18-25

Scott, D. W. (2015). *Multivariate density estimation: theory, practice, and visualization.* John Wiley & Sons.

Seversky, L. M., Davis, S., and Berger, M. (2016). On time-series topological data analysis: New data and opportunities. *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 59–67.

Sgouralis, I., Nebenführ, A., and Maroulas, V. (2017). A Bayesian topological framework for the identification and reconstruction of subcellular motion. *SIAM Journal on Imaging Sciences*, 10(2):871–899.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis (Vol. 26).*. CRC press, New York.

Simard, R. and L'Ecuyer, P.(2011). *Computing the two-sided Kolmogorov Smirnov distribution.* Journal of Statistical Software 39(11):1-18

Turner, K., Mileyko, Y., Mukherjee, S., and Harer, J. (2014). Fréchet means for distribution of persistence diagrams. *Discrete & Computational Geometry*, 52:44–70.

Venkataraman, V., Ramamurthy, K. N., and Turaga, P. (2016). Persistent homology of attractors for action recognition. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 4150–4154.

Xia, K., Feng, X., Tong, Y., and Wei, G. W. (2015). Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry*, 36(6):408–422.

Appendix A. Proofs from Section 4.3

A.1. Proof of Proposition 38

Note that the lower bound integral is the probability for a pair $z = (x, y)$ of independent standard normal variables to lie in $B((0,0),\delta)$. In order to bound the bottleneck distance $W_\infty(D, \mathscr{D}) < \delta\sigma$, it is sufficient that each constituent feature does not stray too far from either its corresponding center or the diagonal (see Fig. 3 for reference). Specifically, we follow Def. 5 to build a correspondence between D and $\mathscr{D}$ so that the maximal distance undercuts $\delta\sigma$, and thus the (potentially smaller) bottleneck distance is also bounded by $\delta\sigma$. For clarity, features in D are denoted using ζ while features in $\mathscr{D}$ are denoted using ξ.

Consider each feature $\xi^j \in \mathscr{D}^u = \mathscr{D} \cap \{d - b \geq \sigma\}$ and its associated random singleton diagram $D^j = \{\zeta^j\}$ or $\emptyset$ as in Def. 22. Assuming the disc neighborhood $B_j = B(\xi^j, \delta\sigma)$ is contained in the

wedge $W = \{(b,d) \in \mathbb{R}^2 : d > b \geq 0\}$, the density of $z^j = \frac{\zeta^j - \xi^j}{\sigma}$ is a multiple (> 1) of the density of the Gaussian random variable $z \sim N((0,0), I_2)$ in the region where $\zeta^j \in B_j$ (or equivalently $z^j \in B((0,0),\delta)$). Thus, we obtain $\mathbb{P}\left[\zeta^j \in B(\xi^j, \delta\sigma)\right] \geq \mathbb{P}\left[|z| \leq \delta\right]$ for the probability that ζ^j can be mapped to ξ^j in a bounding correspondence. If $B_j \not\subseteq W$, this probability is even higher because ξ^j can be mapped to the diagonal and thus the case $D^j = \emptyset$ is included.

Now take into account the features in $\mathscr{D}^\ell$ and the associated random features D^ℓ as in Def. 24. Although the features in D^ℓ are not necessarily independent, we may assume without loss of generality the worst case, in which the maximal cardinality is drawn. Given a fixed cardinality, the draws of D^ℓ are independent. Since any feature may be mapped to the diagonal in the bottleneck distance, a bounding correspondence can be obtained whenever the draws in D^ℓ and features in $\mathscr{D}^\ell$ are close enough to the diagonal (within $\delta\sigma$). Indeed, the features in $\mathscr{D}^\ell$ are by definition distance $\sigma \leq \delta\sigma$ from the diagonal. Restricting to W, the pdf for the draws of $D^\ell = \{(b_j, d_j)\}_{j=1}^{|N_\ell|}$ is given by $p^\ell(b,d) = \frac{1}{\pi N_\ell \sigma^2} \sum_{j=1}^{N_\ell} e^{-\left(\left(x - \frac{b_j+d_j}{2}\right)^2 + \left(y - \frac{b_j+d_j}{2}\right)^2\right)/2\sigma^2}$. Consider the sets $U_j = B\left(\left(\frac{b_j+d_j}{2}, \frac{b_j+d_j}{2}\right), \delta\sigma\right)$ and $\mathcal{U} = \bigcup_{j=1}^{N_\ell} U_j$. For each lower feature $(b,d) \in D^\ell$, mapping to the diagonal yields a bounding correspondence and the associated probability is bounded below by $\mathbb{P}[d - b \leq \delta\sigma] = \int_{\Delta_0^{\delta\sigma}} p^\ell(x,y)\,dx\,dy \geq \int_{W \cap \mathcal{U}} p^\ell(x,y)\,dx\,dy$ since $W \cap U \subset \Delta_0^{\delta\sigma} = \{(b,d) \in W : d - b \leq \delta\sigma\}$. Next, we restrict the lower bounding integral for each term of p^ℓ to its matching subset U_j and change variables to attain the desired form:

$$\begin{aligned}\int_{W\cap\mathcal{U}} p^\ell(x,y)\,dx\,dy &\geq \sum_{j=1}^{N_\ell} \int_{U_j} \frac{1}{2\pi N_\ell \sigma^2} e^{-\left(\left(x - \frac{b_j+d_j}{2}\right)^2 + \left(y - \frac{b_j+d_j}{2}\right)^2\right)/2\sigma^2}\,dx\,dy \\ &= \int_{B((0,0),\delta)} \frac{1}{2\pi} e^{-(x^2+y^2)/2}\,dx\,dy.\end{aligned}$$

Overall, this argument shows that with probability at least $\mathbb{P}(|\boldsymbol{z}| \leq \delta)^M$ there is a correspondence which bounds the bottleneck distance by $\delta\sigma$ and the result follows.

A.2. Proof of Lemma 39

Choose an arbitrary persistence diagram $\mathscr{D}$. Since bottleneck distance is defined according to the sup-norm (see Eq. (2.2)), the bottleneck distance to the null persistence diagram (i.e., without any features) is precisely half the maximal persistence. Thus, we begin by showing that the maximal persistence moment is finite. Taking $Z = \{\xi_1, ..., \xi_N\}$ with $\xi_i = (b_i, d_i, k_i)$, we have:

$$\int_{\mathcal{W}_{0:\mathrm{cl}-1}} \max(d_i - b_i)\delta Z \leq \int_{\mathcal{W}_{0:\mathrm{cl}-1}} \|Z\|\, f(Z)\delta Z \tag{A.1}$$

since $\max(d_i - b_i) \leq \max(\|(b_i, d_i)\|) \leq \|Z\|$. Consider a compact set $K \subset \mathcal{W}_{0:\mathrm{cl}-1}$ which contains a neighborhood of the origin. Given assumptions $(A2)^*$ and $(A3)^*$, Eq. (A.1) is bounded by the following finite expression.

$$\int_{\mathcal{W}_{0:\mathrm{cl}-1}} \|Z\|\, f(Z)\delta Z \leq \int_K C_2 \|Z\|\, \delta Z + \sum_{N=1}^{M} \int_{h_N^{-1}(h_N(K)^c)} C_3 \|Z\|^{-2N-1}\, d\xi_1 ... d\xi_N. \tag{A.2}$$

Lastly, we take advantage of the Minkowski inequality, which holds trivially for set integration since it is a linear combination of Lebesgue integrals. Indeed, the MAD centered at $\mathscr{D}_0$ is bounded

as follows.

$$\int_{\mathcal{W}_{0:\mathrm{cl}-1}} W_\infty(\mathscr{D}_0, Z) f(Z)\delta Z \leq \int_{\mathcal{W}_{0:\mathrm{cl}-1}} W_\infty(\mathscr{D}_0, \emptyset) f(Z)\delta Z + \int_{\mathcal{W}_{0:\mathrm{cl}-1}} W_\infty(\emptyset, Z) f(Z)\delta Z \qquad \text{(A.3)}$$

where $\emptyset$ represents the null persistence diagram and the distance to the null persistence diagram is precisely half the maximal persistence. Since f integrates to 1, the first integral simplifies to the finite distance $W_\infty(\mathscr{D}_0, \emptyset)$, while the second integral is finite according to Eq. (A.2).

Appendix B. Extra Examples

Here we present two more examples of constructing a kernel density estimator (KDE) according to the kernel given in Eq. (4.6). In these examples, we view slices of the KDE at various sample sizes and bandwidths. In the first example, the underlying dataset consists of points sampled from a circle with relatively large noise, in contrast to Ex. 3 in Subsection 5. This example demonstrates how, despite the symmetry of the unit circle and Gaussian noise of the underlying data, the resulting persistence diagram KDE and eventually its limiting behavior lacks Gaussian structure. In the second example, the underlying dataset consists of points sampled from a pinched circle. The underlying dataset has only one loop, but the persistence diagrams typically have a feature of long persistence and another feature of moderate persistence. Both features are captured by the KDE, and are clearly separable into distinct features despite their adjacency. To keep the presentation relatively simple to interpret, the same slices will be presented for each KDE (see Rmk. 41). This allows one to track the convergence of the KDE as the sample size of persistence diagrams, n, increases and the bandwidth, σ, decreases.

Example 5 Consider random underlying datasets each consisting of 25 points sampled uniformly from the unit circle, which are then perturbed by Gaussian noise with variance $(1/6)^2 I_2$, and their associated Čech persistence diagrams for degree of homology $k = 1$. An example dataset and its associated Čech persistence diagram for $k = 1$ are shown in Fig. 12.

Since the underlying datasets are sampled from a perfect circle perturbed by large noise, one expects the associated 1-homology to have a single persistent feature with several smaller features caused by noise. We consider several KDEs as we simultaneously increase the number of persistence diagrams and narrow the bandwidth. The bandwidth was chosen to vary according to Silverman's rule of thumb (Silverman, 1986). Since the KDEs are defined on $\bigcup_N W^N$ for several input cardinalities N, we present $\hat{f}_{n,\sigma}(Z)$ in multiple slices by fixing a cardinality and then fixing all but one input feature, as explained in Rmk. 41. For example, $g(\xi) = \hat{f}_{n,\sigma}(\xi, \xi'_2, ..., \xi'_N)$ for fixed ξ'_j is a function on W and represents a slice of the local KDE on W^N. This progression of KDEs can be seen in Fig. 13, wherein the same slices are viewed for each choice of n and σ. Modes of each slice are used as fixed features in the slices of higher cardinality inputs; consequently, the presented slices capture portions of the KDE with high probability density.

Fig. 13 demonstrates slower convergence of the KDEs than in Ex. 3, which is expected due to larger noise. Though the tail behavior of the KDEs remains Gaussian in nature, the limiting density is not Gaussian. In fact, the KDEs $\hat{f}(n, \sigma)$ are neither symmetric nor unimodal, even for a single input. Much like the kernel densities themselves, each KDE separates into upper and lower densities on W; however, the lower density varies depending on which upper mode is fixed in $\hat{f}(\xi, \xi'_j)$.

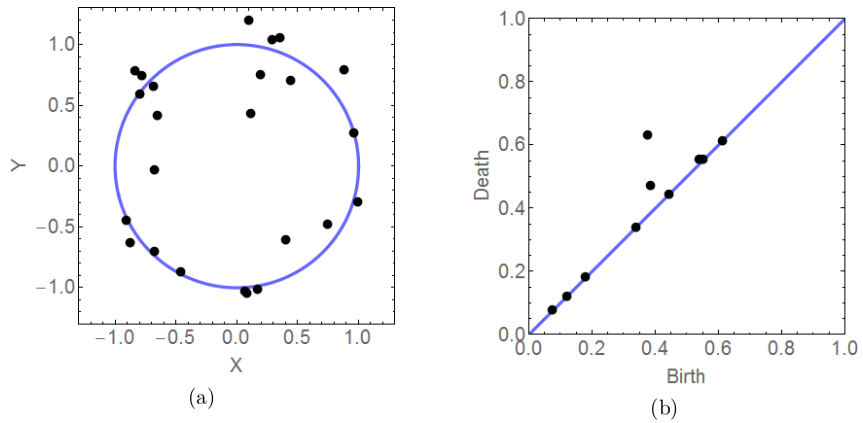


Figure 12: (a) An example of the underlying datasets generated for Ex. 5. Each dataset consists of 25 points sampled uniformly on the unit circle which are then perturbed by i.i.d. Gaussian noise with variance $(1/6)^2 I_2$. (b) The persistence diagram associated to the Čech filtration of the dataset

While the underlying dataspace is the unit circle in both Ex. 3 and Ex. 5, the precise presentation of the underlying data effects the pdf of the associated random persistence diagram. Precisely, two primary parameters for the underlying dataset are involved: (i) the scale of Gaussian noise and (ii) the sample size of the underlying dataset. The persistence diagram (for degree of homology $k = 1$) associated with the ''true' unit circle is not random and has a single feature at $(b, d) = (0, 1)$. The random, discrete nature of these examples creates persistence diagrams which deviate from this 'truth.'

As described for Ex. 3, with very little noise all the sample points lie close to the unit circle, and so the Čech complex becomes contractible at a radius $r \approx 1$. Consequently, the death value of the main topological feature is near the 'true' value (e.g., the mode in Fig. 9 is $d = 0.98 \approx 1$). However, since we are working with discrete points, this feature does not appear immediately: the gaps in the circle need to be filled in (this is even true without noise). In Ex. 3, the sample size is only 10, so the birth value is typically much larger than the 'true' value (e.g., the mode in Fig. 9 is $b = 0.77 >> 0$).

In comparison to Ex. 3, Ex. 5 has relatively more noise; this results in a random persistence diagram with smaller death values for the main feature (e.g., the mode in Fig. 13 is $d = 0.8 < 0.98$). It is evident from Fig. 13 that while the noise is additive on the underlying data, its precise effect on the random persistence diagram is nonlinear. Moreover, Ex. 5 has a larger sample size (25 as opposed to 10), resulting in more consistent and smaller birth times for the main feature (e.g., the mode in Fig. 13 is $b = 0.4 < 0.77$). In addition, larger noise and sample size both result in more features near the diagonal in Ex. 5 as compared to Ex. 3.

Example 6 While Ex. 5 demonstrates the effect of noise on a persistence diagram pdf, this example will look into the effect of geometry. Consider random underlying datasets each consisting

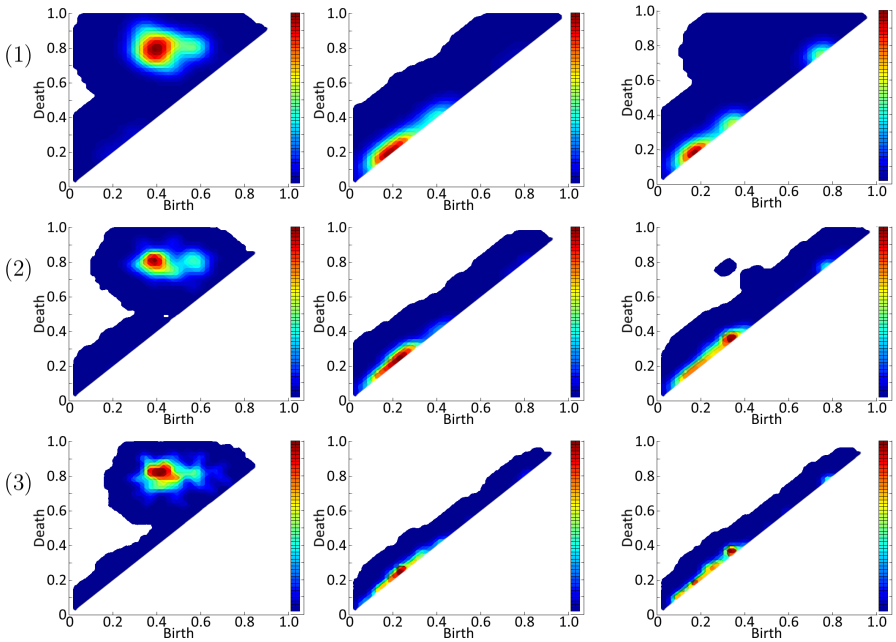


Figure 13: Key slices of persistence diagram KDEs for Ex. 5. Each column is a particular slice, while each row is a particular KDE: (1) $n = 20$ and $\sigma = 0.05$, (2) $n = 100$ and $\sigma = 0.03$, and (3) $n = 300$ and $\sigma = 0.02$. The first column are the local KDEs $\hat{f}_{n,\sigma}((b,d))$ evaluated at a diagram with only one feature. The second column are the local KDEs $\hat{f}_{n,\sigma}((b,d),(0.4,0.8))$ evaluated at a diagram with two features but one feature fixed. The third column are the local KDEs $\hat{f}_{n,\sigma}((b,d),(0.56,0.8))$ evaluated at a diagram with two features but with a different feature fixed. Overall, this figure demonstrates convergence of the KDE as the number of persistence diagrams increases and the bandwidth decreases. Indeed, the two modes on the left already stabilize after $n = 300$, and the spread is no longer determined by the kernel bandwidth.

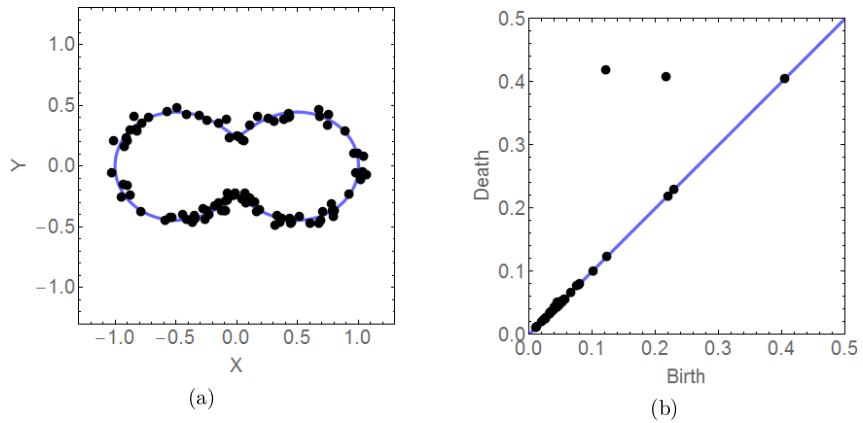


Figure 14: (a) An example of the underlying datasets generated for Ex. 5. Each dataset consists of 100 points sampled uniformly (according to angle) on the two-lobed polar curve which are then perturbed by i.i.d. Gaussian noise with variance $(1/30)^2 I_2$. (b) The persistence diagram associated to the Čech filtration of the underlying dataset.

of 100 points sampled from a two-lobed polar curve, which are then perturbed by Gaussian noise with variance $(1/30)^2 I_2$, and their associated Čech persistence diagrams for degree of homology $k = 1$. An example dataset and its associated persistence diagram for $k = 1$ are shown in Fig. 14.

We consider several KDEs as we simultaneously increase the number of persistence diagrams and narrow the bandwidth. The bandwidth was chosen to vary according to Silverman's rule of thumb (Silverman, 1986). Since the KDEs are defined on $\bigcup_N W^N$ for several input cardinalities N, we present $\hat{f}_{n,\sigma}(Z)$ in multiple slices by fixing a cardinality and then fixing all but one input feature, as explained in Rmk. 41. For example, $g(\xi) = \hat{f}_{n,\sigma}(\xi, \xi'_2, ..., \xi'_N)$ for fixed ξ'_j is a function on W and represents a slice of the local KDE on W^N. This progression of KDEs can be seen in Fig. 15, wherein the same slices are viewed for each choice of n and σ. Modes of each slice are used as fixed features in the slices of higher cardinality inputs; consequently, the presented slices capture portions of the KDE with high probability density. Moreover, Fig. 15 demonstrates that these slices tend to capture specific topological or geometric features of the underlying dataspace.

The two-lobed curve in this example has a Čech persistence diagram consisting of two features, a topological feature of very long persistence and a geometric feature of moderate persistence. The moderate persistence feature describes the pinching of the curve. These two features are captured as separate points by the KDEs, and are thus viewed in completely separate slices of the KDE. By observing the KDE in the last row of Fig. 15, the geometric feature with moderate persistence has considerably less variance. Indeed, while the birth time of the topological feature relies on bridging gaps around the entire shape, which can all vary, the larger birth time of the geometric feature has less variance since it relies solely only on the short circuit between the lobes. As a result of this small variance, the geometric feature is emphasized for the local KDEs with a single input feature; also, the density takes longer to converge near this feature.

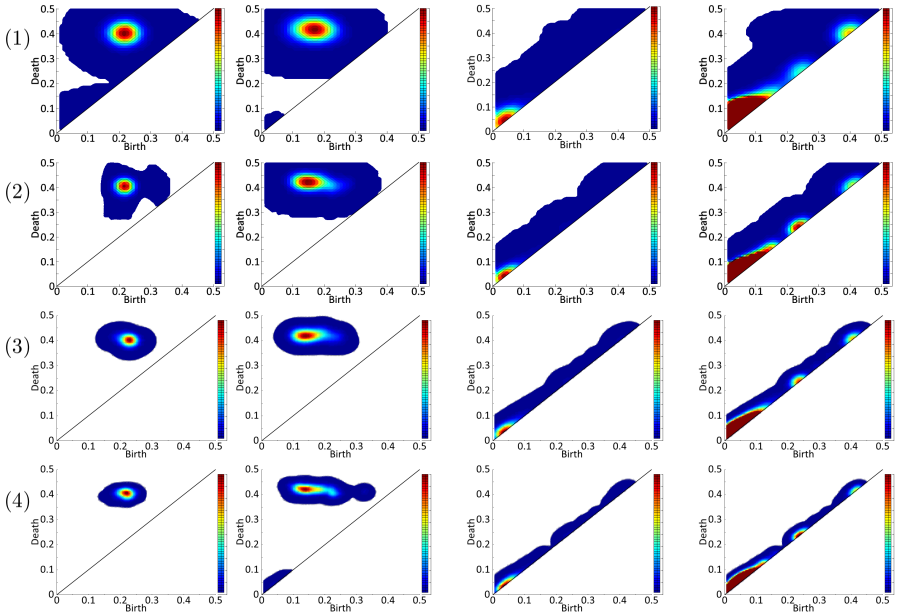


Figure 15: Key slices of persistence diagram KDEs for Ex. 6. Each column is a particular slice, while each row is a particular KDE (1) $n = 20$ and $\sigma = 0.03$, (2) $n = 100$ and $\sigma = 0.02$, (3) $n = 300$ and $\sigma = 0.015$, and (4) $n = 1000$ and $\sigma = 0.01$. The first column are the local KDEs $\hat{f}_{n,\sigma}((b,d))$ evaluated at a diagram with only one feature; the mode is $\xi_1' = (0.2, 0.4)$. The second column are the local KDEs $\hat{f}_{n,\sigma}((b,d),(0.2,0.4))$ evaluated at a diagram with two features, but with one feature fixed; the mode is $\xi_2' = (0.14, 0.42)$ The third column are the local KDEs $\hat{f}_{n,\sigma}((b,d),(0.2,0.4),(0.14,0.42))$ evaluated at a diagram with three features, but with two features fixed. The fourth column shows the same slices as the third, but with the colormap shifted down to show the smaller modes. The variance of certain features effects the rate of convergence nearby, similar to Gaussian KDE in Euclidean space for a distribution with modes of different variance.

The lower portion of the KDE shows three separate modes. Features which build the largest mode consists of small loops, caused by local noise and gaps along the curve. The two modes which appear at larger scale indicate short circuiting of the pinch (smaller) or one of the lobes (larger, like the second mode in the circle example); These two lower modes are separate from noise-based features and are indicative of geometry in the underlying data.

Journal of Machine Learning Research 20 (2019) 1–44 Submitted 10/18; Revised 5/19; Published 10/19

High-dimensional Varying Index Coefficient Models via Stein's Identity

Sen Na SENNA@UCHICAGO.EDU
Department of Statistics
University of Chicago
Chicago, IL 60637, USA

Zhuoran Yang ZY6@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Zhaoran Wang ZHAORAN.WANG@NORTHWESTERN.EDU
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, USA

Mladen Kolar MKOLAR@CHICAGOBOOTH.EDU
The University of Chicago Booth School of Business
Chicago, IL 60637, USA

Editor: Kenji Fukumizu

Abstract

We study the parameter estimation problem for a varying index coefficient model in high dimensions. Unlike the most existing works that iteratively estimate the parameters and link functions, based on the generalized Stein's identity, we propose computationally efficient estimators for the high-dimensional parameters without estimating the link functions. We consider two different setups where we either estimate each sparse parameter vector individually or estimate the parameters simultaneously as a sparse or low-rank matrix. For all these cases, our estimators are shown to achieve optimal statistical rates of convergence (up to logarithmic terms in the low-rank setting). Moreover, throughout our analysis, we only require the covariate to satisfy certain moment conditions, which is significantly weaker than the Gaussian or elliptically symmetric assumptions that are commonly made in the existing literature. Finally, we conduct extensive numerical experiments to corroborate the theoretical results.

Keywords: high-dimensional estimation, semiparametric modeling, Stein's identity, varying index coefficient model

1. Introduction

We consider the problem of estimating parameters in a high-dimensional varying index coefficient model with following form

$$y = \sum_{j=1}^{d_2} z_j \cdot f_j(\langle \boldsymbol{x}, \boldsymbol{\beta}_j^\star \rangle) + \epsilon, \tag{1}$$

©2019 Sen Na, Zhuoran Yang, Zhaoran Wang, and Mladen Kolar.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-705.html.

where y is response variable, $\boldsymbol{x} = (x_1, \ldots, x_{d_1})^\top \in \mathbb{R}^{d_1}$ and $\boldsymbol{z} = (z_1, \ldots, z_{d_2})^\top \in \mathbb{R}^{d_2}$ are given covariates, and ϵ is random noise with $\mathbb{E}[\epsilon \mid \boldsymbol{x}, \boldsymbol{z}] = 0$. For $j \in [d_2]$[1], $\boldsymbol{\beta}_j^\star = (\beta_{j1}^\star, \ldots, \beta_{jd_1}^\star)^\top$ are the coefficient vectors, that is parameters, which vary with different covariates z_j, and $f_j(\cdot)$ are unknown nonparametric link functions. For identification purposes, we can always permute $\boldsymbol{\beta}_j^\star$ and multiply by a scalar such that

$$\boldsymbol{\beta}_j^\star \in \{\boldsymbol{\beta} \in \mathbb{R}^{d_1} : \|\boldsymbol{\beta}\|_2 = 1 \text{ and } \beta_1 > 0\}, \quad j = 1, \ldots, d_2. \tag{2}$$

All further restrictions on parameters will only be considered under (2).

Model (1) has been introduced by Ma and Song (2015) as a flexible generalization of a number of well studied semiparametric statistical models (see also Xue and Wang, 2012). When $z_j = 1$ for all $j \in [d_2]$, the model reduces to the additive single-index model (Chen, 1991; Carroll et al., 1997), which can also be viewed as a two-layer neural network with d_2 hidden nodes. When $d_1 = 1$ and $\boldsymbol{\beta}_j^\star = 1$ for $j = 1, \ldots, d_2$, the model (1) reduces to the varying coefficient model proposed in Cleveland et al. (1991) and Hastie and Tibshirani (1993), with wide applications in scientific areas such as economics and medical science (Fan and Zhang, 2008). Varying coefficient models allow the coefficients of $\boldsymbol{z}$ to be smooth functions of $\boldsymbol{x}$, thus incorporating nonlinear interactions between $\boldsymbol{x}$ and $\boldsymbol{z}$. Model (1) is also easily interpreted in real applications because it inherits features from both single-index model and varying coefficient model, while being able to capture complex multivariate nonlinear structure.

Our focus is on the case when the dimension of $\boldsymbol{x}$ is high, which makes estimation of the coefficients difficult. Existing procedures estimate the unknown functions and coefficients iteratively. First, with the signal parameters $\{\boldsymbol{\beta}_j^\star\}_{j\in[d_2]}$ fixed, one estimates the functions $\{f_j(\cdot)\}_{j\in[d_2]}$ using a nonparametric method, such as local polynomial estimator. Next, using the estimated link functions, one re-estimates the coefficients. While the global minimizer has desirable properties (see Xue and Wang (2012) and Ma and Song (2015) and the references therein), the loss function is usually nonconvex and it is computationally intractable to obtain the global optima. For high-dimensional single-index models ($d_2 = 1$ and $\boldsymbol{z} = 1$), when the distribution of $\boldsymbol{x}$ is known, the signal parameter can be estimated directly by fitting Lasso (Tibshirani, 1996). Such an estimator is shown to achieve minimax-optimal statistical rate of convergence (Plan and Vershynin, 2016; Plan et al., 2017). Thus, the following question naturally arises:

Is it possible to estimate signal parameters $\{\boldsymbol{\beta}_j^\star\}_{j\in[d_2]}$ in (1) *with both statistical accuracy and computational efficiency?*

In this work, we provide a positive answer to above question. Specifically, we focus on the problem of estimating the parameter matrix $\boldsymbol{B}^\star = (\boldsymbol{\beta}_1^\star, \ldots, \boldsymbol{\beta}_{d_2}^\star) \in \mathbb{R}^{d_1 \times d_2}$ in the high-dimensional setting where the sample size is much smaller than $d_1 \times d_2$ and $\boldsymbol{B}^\star$ is either sparse or low-rank. We utilize the score functions and the generalized Stein's identity (Stein, 1972; Stein et al., 2004) to estimate the unknown coefficients through a regularized least-square regression problem, without learning the unknown functions $\{f_j(\cdot)\}_{j\in[d_2]}$. We prove that the estimators achieve (near) optimal statistical rates of convergence under weak moment conditions, which make our procedure suitable for heavy-tailed data, using a careful truncation argument. Finally, our estimator can be computed as a solution to a convex optimization problem.

1. For any integer d, we denote $[d] = \{1, 2, ..., d\}$.

Main Contributions. Our contributions are three-fold. First, we propose a computationally efficient estimation procedure for the varying index coefficient model in high dimensions. Different from existing work, our approach does not need to estimate the unknown functions $\{f_j\}_{j\in[d_2]}$. Second, when $\boldsymbol{B}^\star$ is sparse, we prove that the proposed estimator achieves the optimal statistical rate of convergence, while when $\boldsymbol{B}^\star$ is low-rank, our estimator is shown to be near-optimal (up to a logarithmic factor). Finally, we provide thorough numerical experiments with both synthetic and real data to back up our theories.

Related Work. There is a plethora of literature on the varying coefficient model, first proposed in Cleveland et al. (1991) and Hastie and Tibshirani (1993), where the coefficients are modeled as nonparametric functions of $\boldsymbol{x}$. See Fan and Zhang (2008) for a detailed review. Xia and Li (1999), Fan et al. (2003), and Xue and Wang (2012) considered model in (1) with $\boldsymbol{\beta}_j^\star = \boldsymbol{\beta}^\star$ for all $j \in [d_2]$ and estimated it with standard nonparametric techniques. Ma and Song (2015) proposed model (1) and developed a profile least-square approach to estimate the coefficients. Unfortunately, the estimator is defined as a solution to a constrained optimization problem with nonconvex objective function that can be hard to globally optimize in practice. This should be contrasted to estimators that are based on solving convex optimization problems.

Another related line of research is on the high-dimensional single-index model (SIM) with sparse coefficient vector, which is a special case of model (1). Most of the existing results require either knowing the distribution of $\boldsymbol{x}$ or strong assumptions on the link functions. Specifically, Thrampoulidis et al. (2015), Neykov et al. (2016), Plan and Vershynin (2016), and Plan et al. (2017) all showed that when $\boldsymbol{x}$ is a standard Gaussian and the link function satisfies certain conditions, Lasso estimators could also work for SIM with the same theoretical guarantee as if the link function is not present. To relax the Gaussian assumption, Goldstein et al. (2018) proposed a modified Lasso-type estimator when $\boldsymbol{x}$ has an elliptically symmetric distribution. Moreover, using the generalized Stein's identity, Yang et al. (2017) proposed a soft-thresholding estimator for SIM when the distribution of $\boldsymbol{x}$ is known, and Na and Kolar (2018) proposed estimators for single-index volatility model. Our work can be viewed as the extension of this work from two aspects. First, as mentioned before, our model in (1) includes the single-index model considered in Yang et al. (2017) as a special case, which can be recovered by letting $d_2 = 1$ and $\boldsymbol{z} = 1$. In comparison, our model contains d_2 unknown signal vectors $\{\boldsymbol{\beta}_j^\star\}_{j\in[d_2]}$, where d_2 itself can be large compared to the sample size. As we will show later, when the parameter matrix $\boldsymbol{B}^\star$ has a certain low-dimensional structure, such as being sparse or low rank, we construct efficient estimators that enjoy sharp statistical rates of convergence even when d_2 is large and $\{z_j\}_{j\in[d_2]}$ are dependent, heavy-tailed. Second, from the methodology perspective, our estimators are based on combining the first-order Stein's identity and the dependence structure of the covariate $\boldsymbol{z}$, which is a novel observation in index models and generalizes the method in Yang et al. (2017) for SIM. In particular, we extract the information of the parameter matrix $\boldsymbol{B}^\star$ via multiplying the response y by the score function of $\boldsymbol{x}$ and the covariate $\boldsymbol{z}$ adjusted by its precision matrix (see (9)). When $\boldsymbol{z}$ has independent components, our procedure for estimating each single sparse vector $\boldsymbol{\beta}_j^\star$ is equivalent to fitting d_2 SIMs using the procedure proposed in Yang et al. (2017). However, even having independent $\{z_j\}_{j\in[d_2]}$, their method fails when estimating $\boldsymbol{B}^\star$ as a whole with either low-rank or sparse structure, since they only have one coefficient and cannot deal with varying coefficient models without accounting for $\boldsymbol{z}$ in Steins identity. Furthermore, to handle dependence structure of $\boldsymbol{z}$ in general, we construct suitable precision matrix estimators for heavy-tailed variable in either low or high dimensions based on the CLIME procedure (Cai et al., 2011) and the soft truncation technique, which is of independent interests. Besides aforementioned related literature, a sequence of work (Zhu et al., 2006; Jiang and Liu, 2014; Zhang et al., 2017; Lin et al., 2017, 2018) applied the sliced inverse regression (SIR) technique on high-dimensional SIM, which is generalized from Li (1991). But all these works require the distribution of $\boldsymbol{x}$ to be Gaussian or elliptical. To resolve this limitation, Babichev and Bach (2018) incorporated SIR with both first-order and second-order score function when fitting a low-dimensional index model, while the high-dimensional analysis is not included.

Furthermore, our work is also related to the study of additive index model, which is more challenging than (1), and there is very much work in this direction. Most existing work focuses on estimating the signal parameters and the link functions together in the low-dimensional setting. See, for example, Yuan (2011), Wang et al. (2015), and Chen and Samworth (2016). When the covariate is Gaussian and the link functions are known, Sedghi et al. (2016) proposed to estimate the signal parameters via tensor decomposition. These works are not comparable with ours as we consider a different model and our goal is to efficiently estimate the high-dimensional parameters.

Lastly, our estimation methodology utilizes the generalized Stein's identity (Stein et al., 2004), which extends the well-known Stein's identity for Gaussian distribution (Stein, 1972) to general distributions whose density satisfies certain regularity conditions. This identity is widely applied in probability, statistics, and machine learning. We point the reader to Chen et al. (2011), Chwialkowski et al. (2016), Liu et al. (2016), Liu and Wang (2016), and Liu et al. (2018) for recent applications.

Notations: Throughout the paper, we use boldface letters, $\boldsymbol{v}, \boldsymbol{V}$, to denote vector or matrix and their elements will be denoted as v_i, V_{ij}. For any vector $\boldsymbol{v}$ and $p \geqslant 1$, $\|\boldsymbol{v}\|_p$ is the vector ℓ_p-norm, with the usual extension $\|\boldsymbol{v}\|_0 = |\mathrm{supp}(\boldsymbol{v})| = |\{i : v_i \neq 0\}|$. Given a matrix $\boldsymbol{V} \in \mathbb{R}^{m\times n}$, we let $\|\boldsymbol{V}\|_p$ be the induced p-norm. We denote the nuclear norm and Frobenius norm as $\|\boldsymbol{V}\|_*$ and $\|\boldsymbol{V}\|_F$, respectively. We also define $\|\boldsymbol{V}\|_{p,q} = \big(\sum_{j=1}^n (\sum_{i=1}^m |V_{ij}|^p)^{q/p}\big)^{1/q}$, which is basically computing the vector ℓ_p-norm for each column and then computing the ℓ_q norm for those n numbers. We also define $\|\boldsymbol{V}\|_{\max} = \|\boldsymbol{V}\|_{\infty,\infty}$ and $\mathrm{supp}(\boldsymbol{V}) = \{(i,j) : V_{ij} \neq 0\}$. For two matrices $\boldsymbol{V}, \boldsymbol{U}$ with the same dimension, we let $\langle \boldsymbol{V}, \boldsymbol{U}\rangle = \mathrm{trace}(\boldsymbol{V}^T\boldsymbol{U}) = \sum_{i,j=1}^n V_{ij}U_{ij}$. When presenting the result, we use $a \lesssim b$ ($\gtrsim$) to denote $a \leqslant c \cdot b$ ($\geqslant$) for some constant c that we are less interested in. Also, we have $a \asymp b \Leftrightarrow a \lesssim b$ and $a \gtrsim b$. Given a threshold λ, we define the soft-thresholding function $\mathcal{T}_\lambda(\cdot)$ as follows: (i) when $\boldsymbol{a} \in \mathbb{R}^d$, we let $\mathcal{T}_\lambda(\boldsymbol{a}) \in \mathbb{R}^d$ with $[\mathcal{T}_\lambda(\boldsymbol{a})]_i = (1 - \lambda/|a_i|)_+ a_i$; (ii) when $\boldsymbol{A} \in \mathbb{R}^{d_1\times d_2}$, suppose its singular value decomposition can be written as $\boldsymbol{A} = \boldsymbol{U}\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{V}^T$, then we let $\mathcal{T}_\lambda(\boldsymbol{A}) = \boldsymbol{U}\mathrm{diag}(\hat{\boldsymbol{\sigma}})\boldsymbol{V}^T$ where $\hat{\sigma}_i = (\sigma_i - \lambda)_+$.

2. Estimation via the Generalized Stein's Identity

In this section, we present the main idea for estimating coefficients in model (1). Our estimator relies on the generalized Stein's identity (Stein et al., 2004), which we state next.

Theorem 1 (Generalized Stein's identity, Stein et al. (2004)) *Suppose $\boldsymbol{v} \in \mathbb{R}^d$ is a random vector with differentiable positive density $p_{\boldsymbol{v}} : \mathbb{R}^d \to \mathbb{R}$. We define its score function as $S_{\boldsymbol{v}} : \mathbb{R}^d \to \mathbb{R}^d$, $S_{\boldsymbol{v}}(\mathbf{v}) = -\nabla \log p_{\boldsymbol{v}}(\mathbf{v})$. If a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ together with $\boldsymbol{v}$ satisfies regularity condition: $|p_{\boldsymbol{v}}(\mathbf{v})| \to 0$ as $\|\mathbf{v}\| \to \infty$ and $\mathbb{E}[|f(\boldsymbol{v})S_{\boldsymbol{v}}(\boldsymbol{v})|] \vee \mathbb{E}[|\nabla f(\boldsymbol{v})|] < \infty$, then we have*

$$\mathbb{E}[f(\boldsymbol{v})S_{\boldsymbol{v}}(\boldsymbol{v})] = \mathbb{E}[\nabla f(\boldsymbol{v})]. \tag{3}$$

In particular, when $\boldsymbol{v} \sim N(\mathbf{0}, \boldsymbol{I}_d)$, we have

$$\mathbb{E}[g(\boldsymbol{v})\boldsymbol{v}] = \mathbb{E}[\nabla g(\boldsymbol{v})].$$

We drop off the subscript of density and score function to make notation concise. In order to use Theorem 1 for estimation of coefficients in model (1), we require the following regularity condition.

Assumption 2 (Regularity) *We assume that $\boldsymbol{x}$, $\boldsymbol{z}$ in (1) are independent and the density function $p(\cdot)$ of $\boldsymbol{x}$ is positive and differentiable. For any $j \in [d_2]$, we assume function $\tilde{f}_j : \mathbb{R}^{d_1} \to \mathbb{R}$, defined to be $\tilde{f}_j(\mathbf{x}) = f_j(\langle \mathbf{x}, \boldsymbol{\beta}_j^\star \rangle)$, together with variable $\boldsymbol{x}$ satisfies the regularity condition of Theorem 1. Further, let $\mu_j := \mathbb{E}[f_j'(\langle \boldsymbol{x}, \boldsymbol{\beta}_j^\star \rangle)]$ and we assume $\mu_j \neq 0$. In addition, we assume covariate $\boldsymbol{z}$ are standardized with $\mathbb{E}[z_j] = 0$ and $\mathbb{E}[z_j^2] = 1$, $\forall j \in [d_2]$.*

Note that the standardization of $\boldsymbol{z}$ is made only to simplify our presentation. It is easy to extend to a general $\boldsymbol{z}$ using the fact that $\mathbb{E}[\boldsymbol{z}(\boldsymbol{z} - \mathbb{E}[\boldsymbol{z}])^T] = \mathrm{Var}(\boldsymbol{z})$, where diagonal entries of $\mathrm{Var}(\boldsymbol{z})$ can be assumed to be one without loss of generality, as the variance of each z_j can be absorbed into $f_j(\cdot)$. Since $\mathbb{E}[\boldsymbol{z}]$ is easy to estimate at a fast enough rate, we can replace $\boldsymbol{z}$ by $\boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]$ whenever necessary in analysis for the general $\boldsymbol{z}$. See equation (9) for example. Under Assumption 2, Stein's identity will allow us to extract the unknown coefficient parameter, which is proportional to the derivative of the corresponding unknown function in the index model. To clarify, note that

$$\mathbb{E}[f_j(\langle \boldsymbol{x}, \boldsymbol{\beta}_j^\star \rangle)S(\boldsymbol{x})] = \mathbb{E}[\tilde{f}_j(\boldsymbol{x})S(\boldsymbol{x})] \overset{(3)}{=} \mathbb{E}[\nabla \tilde{f}_j(\boldsymbol{x})] = \mu_j \boldsymbol{\beta}_j^\star := \tilde{\boldsymbol{\beta}}_j. \tag{4}$$

The condition $\mu_j \neq 0$ ensures that the above expectation will not vanish and further $\boldsymbol{\beta}_j^\star$ can be fully identifiable from $\tilde{\boldsymbol{\beta}}_j$ due to (2).

With this setup, we start with a warm-up example and illustrate how to estimate coefficients $\{\boldsymbol{\beta}_j^\star\}_{j \in [d_2]}$ when $\boldsymbol{x} \sim N(\mathbf{0}, \boldsymbol{I}_{d_1})$, $\boldsymbol{z} \sim N(\mathbf{0}, \boldsymbol{I}_{d_2})$ and $\boldsymbol{x}$ and $\boldsymbol{z}$ are independent. The extension to heavy-tailed distributions is presented in the next section. Similar to (4), for any $k \in [d_2]$, Stein's identity gives us

$$\mathbb{E}[y \cdot z_k \cdot \boldsymbol{x}] = \sum_{j=1}^{d_2} \mathbb{E}[z_j z_k f_j(\langle \boldsymbol{\beta}_j^\star, \boldsymbol{x} \rangle)\boldsymbol{x}] = \mathbb{E}[f_k(\langle \boldsymbol{\beta}_k^\star, \boldsymbol{x} \rangle)\boldsymbol{x}] \overset{(4)}{=} \mu_k \boldsymbol{\beta}_k^\star = \tilde{\boldsymbol{\beta}}_k. \tag{5}$$

Under Assumption 2 and identifiability condition (2), the above equation allows us to form an estimator for $\boldsymbol{\beta}_k^\star$ by minimizing the following population loss

$$\tilde{\boldsymbol{\beta}}_k = \arg\min_{\boldsymbol{\beta}_k} L_k(\boldsymbol{\beta}_k) = \arg\min_{\boldsymbol{\beta}_k} \left\{ \|\boldsymbol{\beta}_k\|_2^2 - 2\mathbb{E}[y \cdot z_k \cdot \langle \boldsymbol{\beta}_k, \boldsymbol{x} \rangle] \right\}. \tag{6}$$

Given n i.i.d. copies of $(y, \boldsymbol{x}, \boldsymbol{z})$, $\{y_i, \boldsymbol{X}_i, \boldsymbol{Z}_i\}_{i=1}^n$, we obtain an estimator of $\tilde{\boldsymbol{\beta}}_k$ by replacing the expectation in (6) with a sample mean

$$\hat{\boldsymbol{\beta}}_k = \arg\min_{\boldsymbol{\beta}_k} \hat{L}_k(\boldsymbol{\beta}_k) + R_k(\boldsymbol{\beta}_k) = \arg\min_{\boldsymbol{\beta}_k} \left\{ \|\boldsymbol{\beta}_k\|^2 - \frac{2}{n}\sum_{i=1}^n y_i Z_{ik} \langle \boldsymbol{\beta}_k, \boldsymbol{X}_i \rangle + \lambda_k \|\boldsymbol{\beta}_k\|_1 \right\}, \tag{7}$$

where $R_k(\boldsymbol{\beta}_k)$ is a penalty function that imposes desired structural assumptions on the estimate. In a high-dimensional setting, it is common to assume that $\tilde{\boldsymbol{\beta}}_k$ is sparse, so here we use the ℓ_1-norm penalty with $R_k(\boldsymbol{\beta}_k) = \lambda_k \|\boldsymbol{\beta}_k\|_1$. Note that the loss function in (6) can also be written as

$$L(\boldsymbol{\beta}_k) = \mathbb{E}[(y - z_k \langle \boldsymbol{x}, \boldsymbol{\beta}_k \rangle)^2],$$

which leads to an alternative form for the estimator with a design matrix

$$\arg\min_{\boldsymbol{\beta}_k} \left\{ \frac{1}{n}\sum_{i=1}^n (y_i - Z_{ik}\boldsymbol{X}_i^T \boldsymbol{\beta}_k)^2 + \lambda_k \|\boldsymbol{\beta}_k\|_1 \right\}.$$

Finally, we note that the estimator in (7) can be obtained in a closed form

$$\hat{\boldsymbol{\beta}}_k = \mathcal{T}_{\lambda_k/2}\left(\frac{1}{n}\sum_{i=1}^n y_i Z_{ik} \boldsymbol{X}_i \right),$$

where $\mathcal{T}(\cdot)$ is the soft-thresholding operator defined at the end of Section 1.

Our first result establishes the convergence rate for the estimator in (7). We present the result for a slightly more general setting where $\boldsymbol{z}$ has independent sub-Gaussian components with $\|z_j\|_{\psi_2} = \Upsilon_{z_j}$, $\forall j \in [d_2]$.[2]

Theorem 3 (Warm-up) *Consider model (1) with $\|\boldsymbol{\beta}_k^\star\|_0 \leqslant s$ for $k \in [d_2]$, $\boldsymbol{x} \sim N(\mathbf{0}, \boldsymbol{I}_{d_1})$, components of $\boldsymbol{z}$ are independent with $\|z_k\|_{\psi_2} = \Upsilon_{z_k} \leqslant \Upsilon_{\boldsymbol{z}}$ for $k \in [d_2]$ and independent of $\boldsymbol{x}$, and y is sub-exponential with $\|y\|_{\psi_1} \leqslant \Upsilon_y$. Furthermore assume that Assumption 2 holds. The estimator in (7) with $\lambda_k = 4\Upsilon\sqrt{\log n/n}$, for a constant Υ that depends on Υ_y and $\Upsilon_{\boldsymbol{z}}$ only, satisfies*

$$\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \leqslant \frac{3}{2}\sqrt{s}\lambda_k \quad \text{and} \quad \|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_1 \leqslant 6s\lambda_k, \quad \forall k \in [d_2]$$

with probability at least $1 - d_2 d_1/n^2$.

Theorem 3 established rate of convergence for the estimator of $\tilde{\boldsymbol{\beta}}_k$. In particular, it suggests that with high probability we have $\forall k \in [d_2]$,

$$\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \lesssim \sqrt{\frac{s \log n}{n}} \quad \text{and} \quad \|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_1 \lesssim s\sqrt{\frac{\log n}{n}},$$

2. For centered random variable x, we define $\|x\|_{\psi_1} = \sup_{p \geqslant 1} p^{-1}(\mathbb{E}|x|^p)^{1/p}$, $\|x\|_{\psi_2} = \sup_{p \geqslant 1} p^{-1/2}(\mathbb{E}|x|^p)^{1/p}$. We call x a sub-exponential random variable if $\|x\|_{\psi_1} < \infty$. We call x a sub-Gaussian random variable with proxy variance $\|x\|_{\psi_2}^2$ if $\|x\|_{\psi_2} < \infty$. See Vershynin (2012) for detailed properties.

which matches the optimal rate of convergence for sparse vectors recovery under the setting when $n \ll d_1 \ll n^2$ (Lin et al., 2017). It is useful to note that sub-exponential assumption on y is mild. For example, it is satisfied when $\{f_j\}_{j\in[d_2]}$ can be dominated by a linear function and ϵ is sub-exponential. The result follows from a bound on $\|\nabla \hat{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty$, which is presented in the following lemma.

Lemma 4 *Under the conditions of Theorem 3, we have* $\forall k \in [d_2]$,

$$P\left(\|\nabla \hat{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty > 2\Upsilon\sqrt{\frac{\log n}{n}}\right) < \frac{d_1}{n^2}.$$

Under the identifiability condition (2), we have the following corollary for the normalized estimator.

Corollary 5 *Suppose the conditions of Theorem 3 are satisfied, then for sufficiently large* n *(threshold depends on* s *and* $\min_{j\in[d_2]}(|\mu_j| \wedge \beta^\star_{j1})$*), we have*

$$\left\|sign(\hat{\beta}_{k1})\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \boldsymbol{\beta}^\star_k\right\|_2 \lesssim \sqrt{\frac{s\log n}{n}} \quad \text{and} \quad \left\|sign(\hat{\beta}_{k1})\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \boldsymbol{\beta}^\star_k\right\|_1 \lesssim s\sqrt{\frac{\log n}{n}}, \quad \forall k \in [d_2]$$

with probability $1 - d_2 d_1/n^2$. *Further we can get* $\|\hat{\boldsymbol{B}} - \boldsymbol{B}^\star\|_F \lesssim \sqrt{\frac{d_2 s\log n}{n}}$ *where* $\hat{\boldsymbol{B}}$ *saves normalized estimators by column.*

Note that the order of $\|\hat{\boldsymbol{B}} - \boldsymbol{B}^\star\|_F$ in Corollary 5 is proven under the column-wise sparsity. Details under the assumption that $\boldsymbol{B}^\star$ is fully sparse will be discussed later.

In the next few sections, we will build on the illustrative example studied in this section and generalize our results to a heavy-tailed setting, which will improve the applicability of the estimator. Furthermore, we will consider estimation of all coefficients $\{\boldsymbol{\beta}^\star_j\}_{j\in[d_2]}$ simultaneously by imposing structural assumptions on the coefficient matrix $\boldsymbol{B}^\star$. Let $\tilde{\boldsymbol{B}} = (\tilde{\boldsymbol{\beta}}_1, ..., \tilde{\boldsymbol{\beta}}_{d_2})$, we usually focus on establishing statistical guarantee for estimating $\tilde{\boldsymbol{B}}$ as it keeps the same structure as $\boldsymbol{B}^\star$, and conversely, $\boldsymbol{B}^\star$ is fully identifiable from $\tilde{\boldsymbol{B}}$ under (2).

3. Overview of Results

In this section, we introduce weak moment assumption and provide an overview of the proposed estimators and their statistical convergence rates. Our theoretical analysis is separated into two cases: (i) estimating a single sparse coefficient vector $\boldsymbol{\beta}^\star_k$; (ii) estimating the coefficient matrix $\boldsymbol{B}^\star$. In the former case, we assume covariate $\boldsymbol{z}$ has independent entries so that we can extract one specific parameter, while in the latter case, we impose either low-rank or sparse structure on $\boldsymbol{B}^\star$ and relax the requirement for independence of $\boldsymbol{z}$ by modifying the estimation procedure to include the estimator of the precision matrix. We build our theoretical results on following weak moment condition.

Assumption 6 (Finite p-th moment) *We say finite p-th moment holds if there exists a constant $M_p > 0$ such that*

$$\mathbb{E}[y^p] \vee \mathbb{E}[S(\boldsymbol{x})^p_j] \vee \mathbb{E}[z^p_k] \leqslant M_p, \quad \forall j \in [d_1], k \in [d_2].$$

Single sparse vector recovery			
Type	Moment condition	Dimension	Rate
Warm-up	$\boldsymbol{x}, \boldsymbol{z} \sim N(0, \boldsymbol{I})$, indep; $y \sim$ subE	$s \ll n \ll d_1 \ll n^2$	$\sqrt{\frac{s\log n}{n}}$
General	$p = 6$	$s \ll n \ll d_1$	$\sqrt{\frac{s\log d_1 d_2}{n}}$
Low-rank matrix recovery			
Sparse precision	$p = 4$	$(r, w) \ll (d_1, d_2) \ll n$	$\sqrt{\frac{r(d_1+d_2)\log(d_1+d_2)}{n}} \vee w\sqrt{\frac{r\log d_2}{n}}$
General precision	$p = 4$	$r \ll (d_1, d_2) \ll n$	$\sqrt{\frac{r(d_1+d_2)\log(d_1+d_2)}{n}}$
Indep. entries in $\boldsymbol{z}$	$p = 4$	$r \ll (d_1, d_2) \ll n$	$\sqrt{\frac{r(d_1+d_2)\log(d_1+d_2)}{n}}$
Sparse matrix recovery			
Column sparse & sparse precision	$p = 6$	$(s, d_2) \ll n \ll d_1$	$\sqrt{\frac{sd_2\log d_1 d_2}{n}}$
Column sparse & general precision	$p = 6$	$(s, d_2) \ll n \ll d_1$	$\sqrt{\frac{sd_2\log d_1 d_2}{n}} \vee \frac{d_2\sqrt{s\log d_2}}{\sqrt{n}}$
Fully sparse & sparse precision	$p = 6$	$(s, w) \ll n \ll (d_1, d_2)$	$\sqrt{\frac{s\log d_1 d_2}{n}}$
Fully sparse & indep. entries in $\boldsymbol{z}$	$p = 6$	$s \ll n \ll (d_1, d_2)$	$\sqrt{\frac{s\log d_1 d_2}{n}}$

Table 1: Convergence Rate for $\|\cdot\|_2$ or $\|\cdot\|_F$.

This condition is used throughout all of our theoretical analysis. In sparse vector recovery, we require finite 6-th moment, while in low-rank matrix recovery, we only require finite 4-th moment. Note that though we cannot assume $S(\boldsymbol{x})$ is sub-Gaussian, it turns out assuming $S(\boldsymbol{x})$ to have finite moment is still reasonable in the sense that even for some heavy-tailed distributions, such as t-distribution and Gamma distribution, their score variables still satisfy the finite moment assumption for some p. On the other hand, assumptions for applying Stein's identity always boil down to finite moment conditions. For example, Yang et al. (2017) required finite 4-th moment when estimating SIM. In order to estimate varying indices, we require additional two moments to be finite.

Our results are summarized in Table 1. In summary, we achieve $\sqrt{s\log d_1/n}$ rate for estimating a single sparse vector, while $\sqrt{s\log d_1 d_2/n}$ for estimating a fully sparse parameter matrix. Both of them attain the minimax rate considering the case where all unknown link functions $f_j(\cdot)$ are identity functions. For low-rank estimation, we achieve $\sqrt{r(d_1+d_2)\log(d_1+d_2)/n}$ rate, which is also comparable with results in Plan and Vershynin (2016); Goldstein et al. (2018), though it only attains near-optimal rate up to the logarithmic factor. Note that estimating precision matrix of $\boldsymbol{z}$ can be conducted independently from our main procedure and any advanced, suitable estimators can be plugged into our approach. To make the paper compact but self-contained, we only consider estimating a general low-dimensional precision matrix for heavy-tailed $\boldsymbol{z}$ as an illustration, and discuss the high-dimensional sparse precision matrix estimation in appendix. Basically, if the precision matrix of $\boldsymbol{z}$ is sparse, we can estimate it by CLIME procedure (Cai et al., 2011) with a careful truncation of the sample covariance to attain optimal rate of convergence,

even though $\boldsymbol{z}$ only has finite certain moment. The proposed precision matrix estimator for heavy-tailed variable is used as a plug-in estimator whenever $\boldsymbol{z}$ has non-diagonal covariance matrix. Detailed estimation procedures and corresponding error rates are showed in Section 5 and Appendix A, respectively.

4. Sparse Vector Recovery

In this section, we present an extension of the estimator discussed in Section 2 to heavy-tailed data. Applying Theorem 1 after replacing $\boldsymbol{x}$ by $S(\boldsymbol{x})$ in (5) leads to

$$\mathbb{E}[y \cdot z_k \cdot S(\boldsymbol{x})] = \sum_{j=1}^{d_2} \mathbb{E}[z_j z_k f_j(\langle \boldsymbol{\beta}_j^\star, \boldsymbol{x}\rangle) S(\boldsymbol{x})] = \mathbb{E}[f_k(\langle \boldsymbol{\beta}_k^\star, \boldsymbol{x}\rangle) S(\boldsymbol{x})] \overset{(4)}{=} \mu_k \boldsymbol{\beta}_k^\star = \tilde{\boldsymbol{\beta}}_k,$$

under the independence condition that $\mathbb{E}[z_j z_k] = 0$ for $j \neq k$, which we maintain throughout the section but relax it in Section 5 and 6. The above identity allows us to estimate the direction of $\boldsymbol{\beta}_k^\star$ by estimating the left hand side even in the setting with heavy-tailed data. However, in order to get a fast rate of convergence we will require the covariates and the response to be appropriately truncated.

Given a threshold $\tau > 0$, we define the truncation of a vector $\boldsymbol{v} \in \mathbb{R}^d$ as $\breve{\boldsymbol{v}} \in \mathbb{R}^d$ whose coordinates are defined by $[\breve{\boldsymbol{v}}]_i = v_i$ if $|v_i| \leqslant \tau$ and 0 otherwise. Our estimator for $\tilde{\boldsymbol{\beta}}_k$ is given as

$$\hat{\boldsymbol{\beta}}_k = \arg\min_{\boldsymbol{\beta}_k} \bar{L}_k(\boldsymbol{\beta}_k) + R_k(\boldsymbol{\beta}_k) = \arg\min_{\boldsymbol{\beta}_k} \left\{ \|\boldsymbol{\beta}_k\|^2 - \frac{2}{n}\sum_{i=1}^n \breve{y}_i \breve{Z}_{ik} \langle \boldsymbol{\beta}_k, \widetilde{S(\boldsymbol{X}_i)}\rangle + \lambda_k \|\boldsymbol{\beta}_k\|_1 \right\}, \tag{8}$$

which can be obtained in a closed form as

$$\hat{\boldsymbol{\beta}}_k = \mathcal{T}_{\lambda_k/2}\left(n^{-1} \sum_{i=1}^n \breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)} \right).$$

Compared to the estimator in (7), we have replaced $\boldsymbol{X}_i$ by $S(\boldsymbol{X}_i)$ and have carefully truncated the data to obtain the following result.

Theorem 7 (Single sparse vector recovery) *Consider the model* (1) *with* $\|\boldsymbol{\beta}_k^\star\|_0 \leqslant s$, $\forall k \in [d_2]$. *Suppose Assumption 2, Assumption 6* ($p = 6$) *hold and* $\mathbb{E}[z_j z_k] = 0$ *for* $j \neq k$, *then the estimator defined in* (8) *with* $\lambda_k = 76\sqrt{M_6 \log d_1 d_2 / n}$ *and* $\tau = (M_6 n / \log d_1 d_2)^{1/6}/2$ *satisfies*

$$\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \leqslant \frac{3}{2}\sqrt{s}\lambda_k \quad \textit{and} \quad \|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_1 \leqslant 6s\lambda_k, \quad \forall k \in [d_2],$$

with probability at least $1 - 2/d_1^2 d_2^2$.

The theorem establishes that

$$\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \lesssim \sqrt{\frac{s \log d_1 d_2}{n}} \quad \text{and} \quad \|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_1 \lesssim s\sqrt{\frac{\log d_1 d_2}{n}}$$

with high probability. When $d_2 = 1$ the rate matches the minimax rate established in Lin et al. (2017). Our proof technique requires finite 6-th moment, which ensures that the truncated variables do not lose too much information. This assumption can be compared to boundedness of the 4-th moment in estimation of a single-index model (Yang et al., 2017). We require a stronger assumption due to estimation in a more general model. Theorem 7 follows from a bound on $\|\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty$ given in the following lemma.

Lemma 8 *Under the conditions of Theorem 7,*

$$P\left(\|\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty \leqslant 38\sqrt{\frac{M_6 \log d_1 d_2}{n}}, \quad \forall k \in [d_2] \right) \geqslant 1 - \frac{2}{d_1^2 d_2^2}.$$

From the standard analysis of the ℓ_1-penalized methods in Bühlmann and van de Geer (2011), we know that the penalty parameter λ_k should be set as $c\|\nabla \hat{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty$ for some $c > 0$. Moreover, we see threshold τ has the order $\tau \asymp 1/\lambda_k^{2/p}$, where p is the number of moments variables have. Thus, the more moments the variables have, the smaller the threshold level τ would be, which is also consistent with our intuition.

We note that the estimator in (8) crucially depends on the independence among coordinates of $\boldsymbol{z}$. Without this assumption the estimator is not valid. In what follows, we study estimators of the matrix $\boldsymbol{B}^\star$ as a whole, by imposing either low-rank or sparse structure, instead of estimating the matrix column by column.

5. Low-rank Matrix Recovery

In this section, we propose an estimator for $\boldsymbol{B}^\star$ in model (1), which has near optimal rate of convergence under an assumption that $\boldsymbol{B}^\star$ is low-rank. We relax the condition that $\mathbb{E}[z_j z_k] = 0$ as assumed earlier, by estimating the inverse of the covariance of $\boldsymbol{z}$, also called the precision matrix. Let $\boldsymbol{\Sigma}^\star = \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top] \in \mathbb{R}^{d_2 \times d_2}$ and $\boldsymbol{\Omega}^\star = (\boldsymbol{\Sigma}^\star)^{-1}$. We consider two cases: (i) when d_2 is in low dimensions, we have no structural assumptions on the precision matrix $\boldsymbol{\Omega}^\star$; (ii) when d_2 is in high dimensions, we assume that the precision matrix is in the set $\mathcal{F}_w^K$ for some w and K, where

$$\mathcal{F}_w^K = \left\{ \boldsymbol{\Omega} \succeq \boldsymbol{0} : \|\boldsymbol{\Omega}\|_{0,\infty} \leqslant w, \|\boldsymbol{\Omega}\|_2 \leqslant K, \|\boldsymbol{\Omega}^{-1}\|_2 \leqslant K \right\}.$$

The set above is borrowed from Cai et al. (2011) and it controls upper and lower bounds on eigenvalues of $\boldsymbol{\Omega}^\star$, as well as the maximal sparsity over columns. Since estimating the precision matrix itself is a well studied topic and can be conducted independently from estimating model (1), we only take the former case as an example. For the latter case, the sparsity structure on precision matrix can allow us to study the model with both d_1, d_2 in high dimensions. We will discuss how to make use of the CLIME procedure (Cai et al., 2011) with truncated sample covariance to estimate the precision matrix for heavy-tailed variable in Appendix A.

We start by writing down the identifiability relationship. Under Assumption 2, we have

$$\begin{aligned}\mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T]\boldsymbol{\Omega}^\star &= \sum_{j=1}^{d_2} \mathbb{E}[f_j(\langle \boldsymbol{\beta}_j^\star, \boldsymbol{x}\rangle)S(\boldsymbol{x})]\mathbb{E}[z_j \cdot \boldsymbol{z}^T]\boldsymbol{\Omega}^\star \\ &= \sum_{j=1}^{d_2} \tilde{\boldsymbol{\beta}}_j \boldsymbol{e}_j^T \boldsymbol{\Sigma}^\star\boldsymbol{\Omega}^\star = (\tilde{\boldsymbol{\beta}}_1, ..., \tilde{\boldsymbol{\beta}}_{d_2}) = \tilde{\boldsymbol{B}}, \end{aligned} \tag{9}$$

where $\boldsymbol{e}_j \in \mathbb{R}^{d_2}$ is the j-th canonical basis vector. This relationship allows us to estimate the $\tilde{\boldsymbol{B}}$ as a minimizer of the population loss,

$$\tilde{\boldsymbol{B}} = \arg\min_{\boldsymbol{B}} \left\{ \|\boldsymbol{B}\|_F^2 - 2\mathbb{E}[y \cdot \langle S(\boldsymbol{x})\boldsymbol{z}^T\boldsymbol{\Omega}^\star, \boldsymbol{B}\rangle] \right\}. \tag{10}$$

In order to use the above relationship, we will separately estimate $\mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T]$ and $\boldsymbol{\Omega}^\star$.

Let

$$\phi(x) = \begin{cases} -\log(1 - x + x^2/2) & \text{if } x \leqslant 0, \\ \log(1 + x + x^2/2) & \text{if } x > 0 \end{cases} \tag{11}$$

be the soft truncation function. When $|x|$ is small $\phi(x) \approx x$; when $|x|$ is relatively large $\phi(x)$ dramatically shrinks x, while still being monotonically increasing. This function has been widely used in robust mean estimation, especially for variables that only have finite certain moments. For the univariate case, suppose $v_1, \ldots, v_n$ to be a sequence of i.i.d. samples distributed as $v_i \sim v$ that only has finite 2nd moment. Catoni (2012) proposed a simple mean estimator defined as $(n\kappa)^{-1}\sum_{i=1}^n \phi(\kappa v_i)$ with a properly chosen κ and showed sub-Gaussian tail bound. Minsker (2018) generalized to the multivariate case using estimator $(n\kappa)^{-1}\sum_{i=1}^n \phi(\kappa \boldsymbol{V}_i)$, where $\boldsymbol{V}_i$ are independent random Hermitian matrices with finite 2nd moments and $\phi(\cdot)$ is applied to the eigenvalues only. Similarly, when $\boldsymbol{V}_i$ are not Hermitian, we can define a dimension-free matrix soft truncation function $\Phi(\cdot)$ using Hermitian dilation as follows: for a matrix $\boldsymbol{V}$, let $\begin{pmatrix} \boldsymbol{0} & \boldsymbol{V} \\ \boldsymbol{V}^T & \boldsymbol{0} \end{pmatrix} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ be the eigenvalue decomposition of the Hermitian dilation of $\boldsymbol{V}$. Let $\tilde{\boldsymbol{U}} = \boldsymbol{Q}\phi(\boldsymbol{\Lambda})\boldsymbol{Q}^T$, where $\phi(\boldsymbol{\Lambda})$ is applied entrywise. Then $\Phi(\boldsymbol{V})$ is defined to be the upper right corner matrix of $\tilde{\boldsymbol{U}}$ with the same dimension as $\boldsymbol{V}$. Using the truncation function $\Phi(\cdot)$, our estimator for $\mathbb{E}[yS(\boldsymbol{x})\boldsymbol{z}^T]$ is further defined as

$$\frac{1}{n\kappa_1}\sum_{i=1}^n \Phi(\kappa_1 y_i \cdot S(\boldsymbol{X}_i)\boldsymbol{Z}_i^T), \tag{12}$$

where $\kappa_1 > 0$ is a user-specified parameter. In the later section, we will also apply the function $\Phi(\cdot)$ when estimating precision matrix for heavy-tailed variables in low dimensions. With the matrix in (12), our estimator of $\tilde{\boldsymbol{B}}$ is given as

$$\hat{\boldsymbol{B}} = \arg\min_{\boldsymbol{B}} \left\{ \|\boldsymbol{B}\|_F^2 - \frac{2}{n\kappa_1}\sum_{i=1}^n \langle \Phi(\kappa_1 y_i \cdot S(\boldsymbol{X}_i)\boldsymbol{Z}_i^T)\hat{\boldsymbol{\Omega}}, \boldsymbol{B}\rangle + \lambda\|\boldsymbol{B}\|_* \right\}, \tag{13}$$

where $\hat{\boldsymbol{\Omega}}$ is an estimator of $\boldsymbol{\Omega}^\star$. The penalty function $\lambda\|\boldsymbol{B}\|_*$ biases the estimated matrix $\hat{\boldsymbol{B}}$ to be low-rank. Note that the estimator $\hat{\boldsymbol{B}}$ can be obtained in a closed form as

$$\hat{\boldsymbol{B}} = \mathcal{T}_{\lambda/2}\left(\frac{1}{n\kappa_1}\sum_{i=1}^n \Phi(\kappa_1 y_i \cdot S(\boldsymbol{X}_i)\boldsymbol{Z}_i^T)\hat{\boldsymbol{\Omega}} \right).$$

We characterize convergence rate for the estimator (13) in the next theorem and discuss estimation of a general low-dimensional $\boldsymbol{\Omega}^\star$ for heavy-tailed covariate later.

Theorem 9 (Low-rank matrix recovery) *Consider the model (1) with rank$(\boldsymbol{B}^\star) \leqslant r$. Suppose Assumption 2, Assumption 6 ($p = 4$) hold and furthermore suppose a precision matrix estimator $\hat{\boldsymbol{\Omega}}$ satisfies*

$$P\big(\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_2 \leqslant \mathcal{H}(n, d_2)\big) \geqslant 1 - \mathcal{P}(n, d_2).$$

Denote $K = \|\boldsymbol{\Sigma}^\star\|_2 \vee \|\boldsymbol{\Omega}^\star\|_2$. If we set $\kappa_1 = \sqrt{\frac{2\log(d_1+d_2)}{n(d_1+d_2)M_4^{3/2}}}$ and

$$\lambda \geqslant 16KM_4^{3/4}\sqrt{\frac{(d_1+d_2)\log(d_1+d_2)}{n}} + 4K\max_{j\in[d_2]}|\mu_j| \cdot \|\boldsymbol{B}^\star\|_2 \cdot \mathcal{H}(n, d_2),$$

then the estimator (13) satisfies

$$\|\hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}\|_F \leqslant 3\sqrt{r}\lambda \quad \textit{and} \quad \|\hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}\|_* \leqslant 24r\lambda.$$

with probability at least $1 - 2/(d_1+d_2)^2 - \mathcal{P}(n, d_2)$.

The theorem follows from the following concentration result.

Lemma 10 *Under the conditions in Theorem 9, we have*

$$\left\| \frac{1}{n\kappa_1}\sum_{i=1}^n \Phi(\kappa_1 y_i \cdot S(\boldsymbol{X}_i)\boldsymbol{Z}_i^T) - \mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T] \right\|_2 \leqslant 4M_4^{3/4}\sqrt{\frac{(d_1+d_2)\log(d_1+d_2)}{n}},$$

with probability at least $1 - \frac{2}{(d_1+d_2)^2}$.

Different from Theorem 7, the penalty parameter λ depends also on a term that comes from estimating $\boldsymbol{\Omega}^\star$. Specifically, if it holds that $\mathbb{E}[z_j z_k] = 0$, we can get the following corollary immediately.

Corollary 11 *Suppose the conditions of Theorem 9 are satisfied. In addition, suppose that $\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top] = \boldsymbol{I}_{d_2}$. If we set $\kappa_1 = \sqrt{\frac{2\log(d_1+d_2)}{n(d_1+d_2)M_4^{3/2}}}$ and $\lambda = 16M_4^{3/4}\sqrt{\frac{(d_1+d_2)\log(d_1+d_2)}{n}}$, then*

$$\|\hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}\|_F \leqslant 3\sqrt{r}\lambda \quad \textit{and} \quad \|\hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}\|_* \leqslant 24r\lambda,$$

with probability at least $1 - 2/(d_1+d_2)^2$.

Next, we briefly discuss how to estimate the precision matrix $\boldsymbol{\Omega}^\star$, noting that any suitable estimator for heavy-tailed data can be used. In a general case, when no additional structural assumptions are available, we can invert the soft truncated empirical covariance matrix as

$$\hat{\boldsymbol{\Omega}} = \hat{\boldsymbol{\Sigma}}^{-1} \quad \text{where} \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{n\kappa_2}\sum_{i=1}^{n}\Phi(\kappa_2 \boldsymbol{Z}_i\boldsymbol{Z}_i^T). \tag{14}$$

We will show that $\hat{\boldsymbol{\Sigma}}$ is invertible for sufficiently large n. In particular, we prove $\|\hat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}^\star\|_2 \lesssim \sqrt{d_2\log d_2/n}$. Thus, $\hat{\boldsymbol{\Sigma}}$ is invertible when $\sqrt{d_2\log d_2/n} < \lambda_{\min}(\boldsymbol{\Sigma}^\star)$, where $\lambda_{\min}(\boldsymbol{\Sigma}^\star)$ denotes the minimum eigenvalue of $\boldsymbol{\Sigma}^\star$. The following lemma characterizes the rate of convergence.

Lemma 12 *Set* $\kappa_2 = \sqrt{\frac{2\log d_2}{nd_2M_4^{1/2}}}$. *If* $n \geqslant 64\sqrt{M_4}K^2 d_2\log d_2$, *the estimator* (14) *satisfies*

$$P\left(\|\hat{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^\star\|_2 \leqslant 8K^2M_4^{1/4}\sqrt{\frac{d_2\log d_2}{n}}\right) \geqslant 1-\frac{2}{d_2^2}.$$

In fact, we only need finite 2nd moment for $\boldsymbol{z}$ to make $\hat{\boldsymbol{\Omega}}$ in (14) consistent. Combining the rate obtained in Lemma 12 with that of Theorem 9, we observe that

$$\|\hat{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\|_F \lesssim \sqrt{\frac{r(d_1+d_2)\log(d_1+d_2)}{n}}.$$

with high probability. In particular, the rate of convergence is governed by the rate obtained in Lemma 10 and the estimation of the precision matrix contributes to the higher order terms. Furthermore, we note that the rate is optimal up to logarithmic terms (Rohde and Tsybakov, 2011). Similar rate is shown in estimating the single-index model (Plan and Vershynin, 2016; Goldstein et al., 2018; Yang et al., 2017). Estimation of a high-dimensional sparse precision matrix is presented in Appendix A.

6. Sparse Matrix Recovery

In this section, we consider the setting as in Section 5, but with the parameter matrix $\boldsymbol{B}^\star$ being sparse rather than low-rank. Different from (12), here we can simply estimate $\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T]$ by

$$\frac{1}{n}\sum_{i=1}^{n}\breve{y}_i\cdot\widetilde{S(\boldsymbol{X}_i)}\breve{\boldsymbol{Z}}_i^T \tag{15}$$

for some truncation threshold $\tau > 0$. We apply hard truncation instead of soft truncation (based on $\phi(\cdot)$) in (15), since we are going to bound the max norm of the error, instead of the operator norm as we did in Section 5. The hard truncation can give a fast rate when bounding the max norm, while a slow rate when bounding the operator norm. The soft truncation works in the opposite way. Finally, our estimator is defined as

$$\hat{\boldsymbol{B}} = \arg\min_{\boldsymbol{B}}\left\{\|\boldsymbol{B}\|_F^2 - \frac{2}{n}\sum_{i=1}^{n}\langle\breve{y}_i\cdot\widetilde{S(\boldsymbol{X}_i)}\breve{\boldsymbol{Z}}_i^T\hat{\boldsymbol{\Omega}},\boldsymbol{B}\rangle + \lambda\|\boldsymbol{B}\|_{1,1}\right\}. \tag{16}$$

We obtain the following rate of convergence for $\hat{\boldsymbol{B}}$.

Theorem 13 (Sparse matrix recovery (column-wise sparse)) *Consider the model (1) with* $\|\boldsymbol{\beta}_k^\star\|_0 \leqslant s$ *for all* $k\in[d_2]$. *Suppose Assumption 2 and Assumption 6* ($p=6$) *hold and furthermore suppose that the precision matrix estimator* $\hat{\boldsymbol{\Omega}}$ *satisfies*

$$P(\|\hat{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^\star\|_{\max} \leqslant \tilde{\mathcal{H}}(n,d_2)) \geqslant 1-\tilde{\mathcal{P}}(n,d_2).$$

If $\tau = (M_6 n/\log d_1d_2)^{1/6}/2$ *in* (15) *and*

$$\lambda \geqslant 76\|\boldsymbol{\Omega}^\star\|_1\sqrt{\frac{M_6\log d_1d_2}{n}} + 4\max_{j\in[d_2]}|\mu_j|\cdot\|\boldsymbol{B}^\star\boldsymbol{\Sigma}^\star\|_\infty\tilde{\mathcal{H}}(n,d_2),$$

then

$$\|\hat{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\|_F \leqslant 2\sqrt{sd_2}\lambda \quad \text{and} \quad \|\hat{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\|_{1,1} \leqslant 8sd_2\lambda,$$

with probability at least $1-2/d_1^2d_2^2-\tilde{\mathcal{P}}(n,d_2)$.

Different from Theorem 9, we require the bound $\|\hat{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^\star\|_{\max}$ with high probability here because $\|\cdot\|_{\max}$ is the dual norm of $\|\cdot\|_{1,1}$. Note that $\|\hat{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^\star\|_{\max} \leqslant \|\hat{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^\star\|_2$, so we can simply have $\tilde{\mathcal{H}}(n,d_2) = \mathcal{H}(n,d_2)$ and $\tilde{\mathcal{P}}(n,d_2) = \mathcal{P}(n,d_2)$ for estimation in low dimensions where $\mathcal{H}(n,d_2)$ and $\mathcal{P}(n,d_2)$ come from Lemma 12. However, this bound is not sharp for CLIME procedure in high dimensions. Above theorem follows from the following lemma.

Lemma 14 *Under the conditions in Theorem 13,*

$$\left\|\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T] - \frac{1}{n}\sum_{i=1}^{n}\breve{y}_i\cdot\widetilde{S(\boldsymbol{X}_i)}\breve{\boldsymbol{Z}}_i^T\right\|_{\max} \leqslant 19\sqrt{\frac{M_6\log d_1d_2}{n}}$$

with probability at least $1-2/d_1^2d_2^2$.

Note that the rate obtained in Theorem 13 is the same as the one obtained in Theorem 7, which required the assumption that $\mathbb{E}[z_jz_k]=0$. Furthermore, we observe that the same proof provided for Theorem 13 can be used under the setting that $n \ll d_1\wedge d_2$ and $\boldsymbol{B}^\star$ is fully sparse with $\|\boldsymbol{B}^\star\|_{0,1} \leqslant s$. Estimation of a high-dimensional precision matrix is discussed in Appendix A, but we note that the error in estimating the precision matrix in any case only contributes higher order terms and our final rate is

$$\|\hat{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\|_F \lesssim \sqrt{\frac{s\log d_1d_2}{n}} \quad \text{and} \quad \|\hat{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\|_{1,1} \lesssim s\sqrt{\frac{\log d_1d_2}{n}}$$

with probability at least $1-2/d_1d_2-2/d_2^2$. It follows by combining Theorem 13 with Lemma 15. Finally, similar to Corollary 11, when $\boldsymbol{\Sigma}^\star = \boldsymbol{I}_{d_2}$, we can set $\tilde{\mathcal{H}}(n,d_2) = \tilde{\mathcal{P}}(n,d_2) = 0$ in Theorem 13 to derive the same optimal rate.

Until now, we have shown a comprehensive theoretical analysis for the model (1). When estimating a single sparse vector, we assume that $\boldsymbol{z}$ has independent entries. When estimating a parameter matrix $\boldsymbol{B}^\star$, we relax this assumption by incorporating an estimate of the precision matrix. Based on our analysis, we see the error occurred in estimating $\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T]$ will always be the dominant term, while estimation of the precision matrix only contributes higher order terms.

7. Numerical Experiment

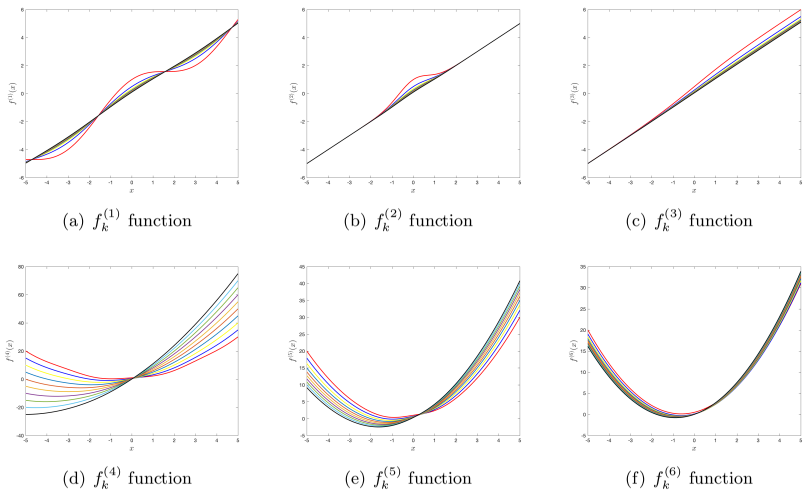
In this section, we illustrate the performance of our proposed estimators in different simulation settings and carry out a real data application in the end. In simulation studies, the link function is set to one of the following six forms:

$$f_k^{(1)}(x) = x + \frac{1}{k}\cos(x); \qquad f_k^{(2)}(x) = x + \frac{1}{k}\exp(-x^2); \qquad f_k^{(3)}(x) = x + \frac{1}{k}\frac{\exp(x)}{1+\exp(x)};$$

$$f_k^{(4)}(x) = x^2 + kx + \frac{1}{k}\cos^2(x); \quad f_k^{(5)}(x) = x^2 + \sqrt{k}x + \frac{1}{\sqrt{k}}\exp(-x^2);$$

$$f_k^{(6)}(x) = x^2 + k^{\frac{1}{4}}x + \frac{1}{k^2}\frac{\exp(x)}{1+\exp(x)}.$$

Their plots are shown in Figure 1. For all simulations we let $\epsilon \sim N(0, 0.01)$. To measure the estimation accuracy we use the cosine distance defined by $\cos(\hat{\boldsymbol{\beta}}, \boldsymbol{\beta}^\star) = 1 - |\hat{\boldsymbol{\beta}}^T\boldsymbol{\beta}^\star|/\|\hat{\boldsymbol{\beta}}\|_2$. Note that we do not normalize $\hat{\boldsymbol{\beta}}$ and change its direction according to the sign of its first entry because cosine distance is more suitable for verifying our matrix results and it allows us to generate $\boldsymbol{\beta}^\star$ without restricting the first entry to be positive. We can easily see $\cos(\hat{\boldsymbol{\beta}}_k, \boldsymbol{\beta}_k^\star) \asymp \|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2^2$, $\forall k \in [d_2]$. For a matrix estimator, we will sum up cosine distance



(a) $f_k^{(1)}$ function (b) $f_k^{(2)}$ function (c) $f_k^{(3)}$ function

(d) $f_k^{(4)}$ function (e) $f_k^{(5)}$ function (f) $f_k^{(6)}$ function

Figure 1: The link functions used in simulations. When k varies from 1 to 10, the line moves from red to black. The link functions in the first row are essentially linear functions combined with different fluctuations, while in the second row they are quadratic. As k increases, the fluctuation is more and more moderate.

Distribution	parameter	score function
Gaussian	$\mu = 0$; $\sigma = 1$	$s(x) = x$
Beta	$\alpha = 8$; $\beta = 8$	$s(x) = \frac{14x-7}{(1-x)x}$
Gamma	$k = 8$; $\theta = 0.1$	$s(x) = 10 - \frac{7}{x}$
Student's t	$\nu = 13$	$s(x) = \frac{14x}{13+x^2}$
Rayleigh	$\sigma = 1$	$s(x) = x - \frac{1}{x}$
Weibull	$k = 7$; $\lambda = 1$	$s(x) = 7x^6 - \frac{6}{x}$

Table 2: *Distribution of* $\boldsymbol{x}$

over all columns, which is a rough surrogate of $\|\hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}\|_F^2$. Our results are averaged of 40 independent runs.

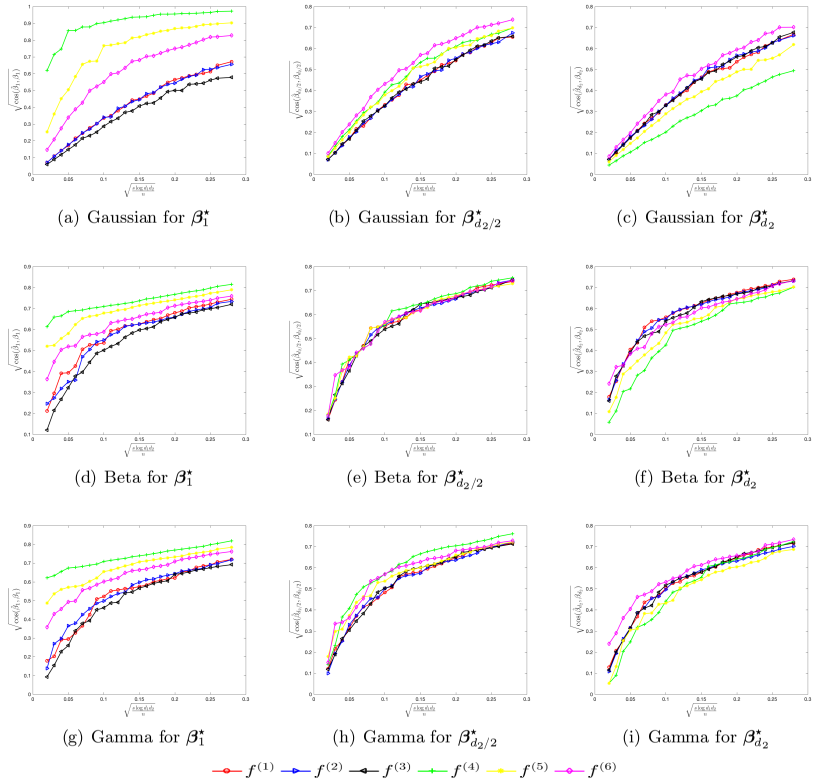
7.1. Single Sparse Vector

We set $d_1 = 100$, $d_2 = 20$, $s = 10$, and vary n only. For $i \in [n]$ and $k \in [d_2]$, we let $\boldsymbol{X}_i$ be independent and identically generated from one of distributions listed in Table 2, and $Z_{ik} \in \{-1, 1\}$ with equal probability and independent from other coordinates. To generate d_1-dimensional parameter $\boldsymbol{\beta}_k^\star$, we first generate the support of nonzero coefficients S_k uniformly at random and then let $\{[\boldsymbol{\beta}_k^\star]_l\}_{l \in S_k} \overset{iid}{\sim} \frac{1}{\sqrt{s}} \cdot \mathrm{Unif}(\{-1, 1\})$. According to Theorem 3 and 7, we set $\lambda_k = 30\sqrt{\log d_1 d_2 / n}$ and $\tau = 2(n/\log d_1 d_2)^{1/6}$.

Figure 2 illustrates the error trend for estimating $\boldsymbol{\beta}_1^\star$, $\boldsymbol{\beta}_{d_2/2}^\star$ and $\boldsymbol{\beta}_{d_2}^\star$ under different link functions and $\boldsymbol{x}$ following one of the following distributions: Gaussian, Beta or Gamma. Additional results are shown in Figure 9 in Appendix D for $\boldsymbol{x}$ following t_{13}, Rayleigh or Weibull distributions. From the plots, we see the error for using linear link functions ($f^{(1)}$ to $f^{(3)}$) is generally smaller than the error for using quadratic link functions ($f^{(4)}$ to $f^{(6)}$), no matter which distribution we use for design. But in all settings the scaled error plots have a linear trend when $n \gg s \log d_1 d_2$, which is consistent with Theorem 7. Another interesting observation is that, as k increases, the estimation will be more and more precise. This can also be seen from our theoretical results. Note that quadratic function $f(x) = x^2$ brings the singularity to the estimation problem even for the Gaussian design since $\mathbb{E}[f'(\langle \boldsymbol{x}, \boldsymbol{\beta}^\star \rangle)] = 0$. Thus, when k is small, the scalar μ_k, appeared in (5), will be close to zero, which implies that the estimation problem is almost singular. Fortunately, when k is large, the linear term in quadratic functions will have more contributions and $\mu_k \approx k$. From Corollary 5 and identity (5), we see that μ_k amplifies the magnitude of the signal $\boldsymbol{\beta}_k^\star$ that characterizes the difficulty of estimation problem. Therefore a larger μ_k implies an easier problem. Figure 10—12 in Appendix D provide additional simulation results where we plot the error trend for all d_2 parameters, which make this observation more clear.

7.2. Low-rank Matrix

We study the estimation of $\boldsymbol{B}^\star$ under the low-rank assumption. We let $d_1 = d_2 = 25$, $r = 5$. The distribution of $\boldsymbol{x}$ is set as described in Table 2. For generating $\boldsymbol{B}^\star$, we first generate two random orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{d_1 \times d_1}$ and $\boldsymbol{V} \in \mathbb{R}^{d_2 \times d_2}$, then generate $d_1 \times d_2$ diagonal matrix $\boldsymbol{\Lambda}$ by first sampling r locations from $[\min\{d_1, d_2\}]$ and further



(a) Gaussian for $\boldsymbol{\beta}_1^\star$ (b) Gaussian for $\boldsymbol{\beta}_{d_2/2}^\star$ (c) Gaussian for $\boldsymbol{\beta}_{d_2}^\star$

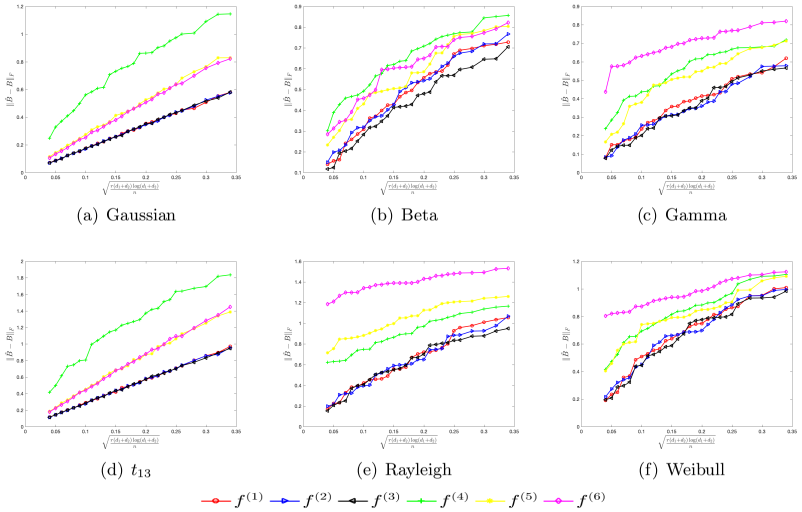
(d) Beta for $\boldsymbol{\beta}_1^\star$ (e) Beta for $\boldsymbol{\beta}_{d_2/2}^\star$ (f) Beta for $\boldsymbol{\beta}_{d_2}^\star$

(g) Gamma for $\boldsymbol{\beta}_1^\star$ (h) Gamma for $\boldsymbol{\beta}_{d_2/2}^\star$ (i) Gamma for $\boldsymbol{\beta}_{d_2}^\star$

$f^{(1)}$ $f^{(2)}$ $f^{(3)}$ $f^{(4)}$ $f^{(5)}$ $f^{(6)}$

Figure 2: Sparse vector estimation plot (I). This figure shows cosine distance trend for error of estimating single sparse parameter in model (1). Six lines indicates six different types of link functions. Each row represents one type of distribution for design.

setting each entry to be $\{-\frac{1}{\sqrt{r}}, \frac{1}{\sqrt{r}}\}$ with equal probability. Finally we let $\boldsymbol{B}^\star = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T$. We consider both independent and dependent $\boldsymbol{z}$. For the independent case, $\boldsymbol{z}$ is generated as in the previous section. For the dependent case, we generate $\boldsymbol{z}$ as follows. Using the Gaussian copula with the correlation matrix $\boldsymbol{\Sigma} = 0.2 \cdot \mathbf{1}_{d_2}\mathbf{1}_{d_2}^T + 0.8\boldsymbol{I}_{d_2}$, we generate d_2-dimensional random vectors with support in $[0,1]^{d_2}$. Then we apply inverse transformation on each coordinate to make it marginally distributed as t_7. Note that under our setup $\boldsymbol{z}$ has dependent coordinates with each coordinate being t_7 distribution. However, the true covariance matrix of $\boldsymbol{z}$ is not $\boldsymbol{\Sigma}$ anymore and is even unknown. According to Theorem 9, we



(a) Gaussian (b) Beta (c) Gamma

(d) t_{13} (e) Rayleigh (f) Weibull

$f^{(1)}$ $f^{(2)}$ $f^{(3)}$ $f^{(4)}$ $f^{(5)}$ $f^{(6)}$

Figure 3: Low-rank matrix estimation plot. This figure shows $\|\hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}\|_F$ error of estimating low-rank parameter matrix in model (1). Six lines indicates six different types of link functions. Components of $\boldsymbol{z}$ are independent.

set $\kappa_1 = 2\sqrt{\log(d_1+d_2)/n(d_1+d_2)}$ and $\lambda = 12\sqrt{(d_1+d_2)\log(d_1+d_2)/n}$. The precision matrix estimator we use is defined in (14) with $\kappa_2 = 2\sqrt{\log d_2/nd_2}$, suggested by Lemma 12.

Figure 3 and 4 summarize our results. We observe a linear trend for all link functions, though the quadratic link functions result in a larger error overall. Also, for each distribution, dependent or independent $\boldsymbol{z}$ have very similar error trends among all six types of link functions. This observation verifies our theoretical results in Theorem 9 and Lemma 12 that estimating precision matrix only contributes higher order error.

7.3. Sparse Matrix

For the sparse matrix recovery, we only consider estimating fully sparse $\boldsymbol{B}^\star$ with dependent covariate $\boldsymbol{z}$. We let $d_1 = 100$, $d_2 = 50$, $s = 10$. The considered distributions for design $\boldsymbol{x}$ are in Table 2. $\boldsymbol{B}^\star$ is generated by first generating the support of nonzero entries and then setting each entry on the support to be $\{-\frac{1}{\sqrt{s}}, \frac{1}{\sqrt{s}}\}$ with equal probability. For generating dependent $\boldsymbol{z}$, we define the sparse precision matrix $\Theta = (\theta_{ij})_{i,j=1}^{d_2}$ as follows

$$\theta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0.2 & \text{if } |i-j| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

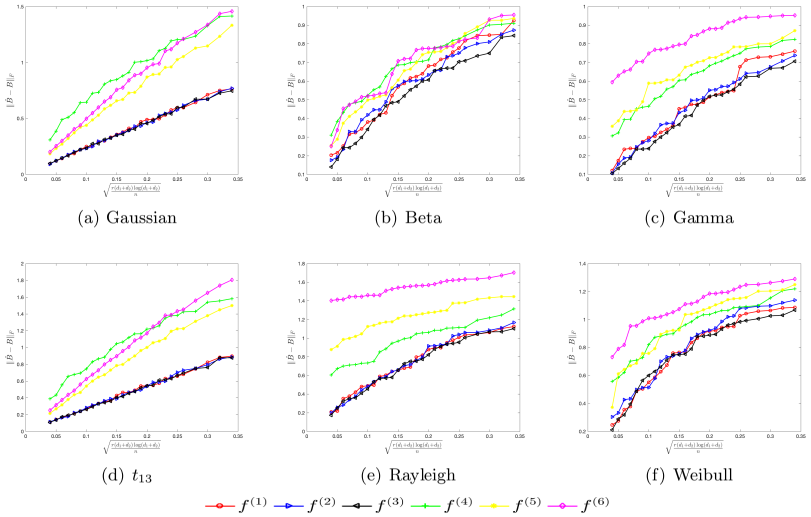


Figure 4: Low-rank matrix estimation plot. This figure shows $\|\hat{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\|_F$ error of estimating low-rank parameter matrix in model (1). Six lines indicates six different types of link functions. Components of $\boldsymbol{z}$ are dependent.

Then we calculate the corresponding covariance matrix Θ^{-1} and convert it to correlation matrix. Following the same steps as in the previous section, we use Gaussian copula to derive dependent covariates $\boldsymbol{z}$, whose marginal distribution for each coordinate is t_7. The precision matrix of $\boldsymbol{z}$ has the same sparsity structure as Θ, even though it is not equal to Θ. According to Theorem 13, we set $\tau = 2(n/\log d_1 d_2)^{1/6}$ and $\lambda = 10\sqrt{\log d_1 d_2/n}$. The sparse precision matrix estimator is defined in (A.1) and (A.2) with truncation threshold $2(n/\log d_2)^{1/4}$ and $\gamma = 10\sqrt{\log d_2/n}$. We use default settings in CVX package (Grant and Boyd, 2008, 2012) to solve (A.1) efficiently. The error plot is shown in Figure 5 and linear trends for errors appear again, as explained by Theorem 13.

7.4. Real Data Application

In this subsection, we will further illustrate the proposed method by analyzing a *Coffea canephora* genetic data set.[3] The data set collects three important traits (phenotypes): production of coffee beans (y_1), leaf rust incidence (y_2) and yield of green beans (y_3), from two recurrent selection populations (x_1) of *Coffea canephora* and each one of traits is evaluated at two locations (x_2). For each individual sample, the single nucleotide polymor-

3. The data set was obtained from Dryad at https://datadryad.org/resource/doi:10.5061/dryad.1139fm7.

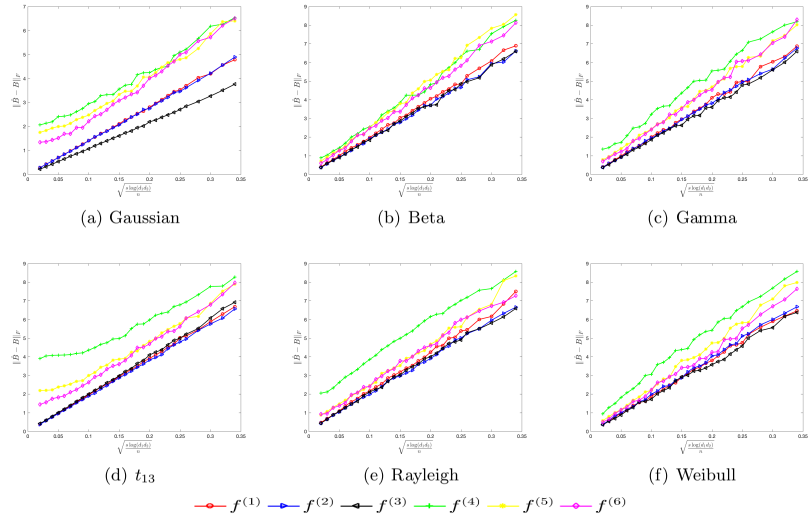


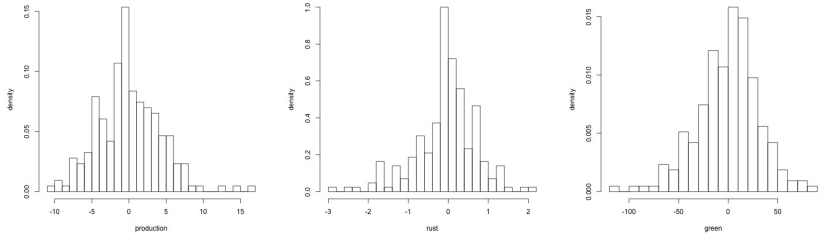
Figure 5: Sparse matrix estimation plot. This figure shows $\|\hat{\boldsymbol{B}}-\tilde{\boldsymbol{B}}\|_F$ error of estimating sparse parameter matrix in model (1) with dependent $\boldsymbol{z}$. Six lines indicates six different types of link functions. All above simulation results are consistent with Theorem 13.

phisms (SNPs)—genotype $\{z_j\}$—are identified by Genotyping-by-Sequencing. In particular, the population group (x_1) and the evaluation location (x_2) are two confounders which may modify the effect of each SNP (z_j) on traits (y_1, y_2, y_3). To study this higher-level effect from confounders, we consider the varying index coefficient model in (1), which in this case is written as

$$y_i = \sum_{j=1}^{d_2} z_j \cdot f_j\big((\beta_j^{1,i})^\star x_1 + (\beta_j^{2,i})^\star x_2\big) + \epsilon, \quad \forall i = 1,2,3, \tag{17}$$

and focus on estimating parameter matrices $\boldsymbol{B}_i^\star = \big((\boldsymbol{\beta}^{1,i})^\star, (\boldsymbol{\beta}^{2,i})^\star\big)^T \in \mathbb{R}^{2\times d_2}$.

We follow the preprocessing steps described in Ferrão et al. (2018). The data set has $n = 215$ samples evaluated at two locations in total, where $n_1 = 119$ of them are collected from the first population group with 45748 SNPs measured, while $n_2 = 96$ of them are collected from the second population group with 59332 SNPs measured. There are 38106 SNPs in common and we select $d_2 = 250$ from them uniformly at random. To make individuals independent from each other, in each group we only use the data evaluated at the first location for the first half of individuals and the data evaluated at the second location for the second half of individuals. Continuous distributions are used for the confounders x_1



(a) Production (b) Leaf rust (c) Green beans

Figure 6: Histograms for three traits: production of coffee beans, leaf rust incidence and yield of green beans.

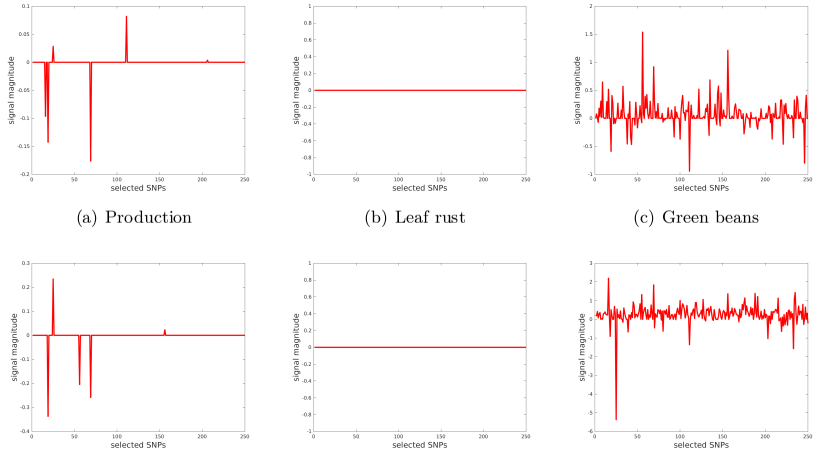
and x_2. In particular, we let $x_1 \sim N(0,1)$, if the observation is from the first population group and $x_1 \sim N(50,1)$, if it is from the second population group. Analogously, we have $x_2 \sim t_{13}$, if the observation is evaluated at the first location and $x_2 \sim 50 + t_{13}$, if it is evaluated at the second location. Here t_{13} refers to the t distribution with degree of freedom 13. Under our setup, the confounders come from the following mixture distribution:

$$x_1 \sim \frac{n_1}{n} N(0,1) + \frac{n_2}{n} N(50,1), \quad x_2 \sim \frac{1}{2} t_{13} + \frac{1}{2}(50 + t_{13}). \tag{18}$$

The corresponding score functions are computed based on the above distributions. We note that the data set is a high-dimensional one, since $n < d_2$, and $\boldsymbol{B}_i^\star$ cannot be estimated using many of the related works, where typically $d_2 < 5$ (Huang and Zhang, 2013; Guo et al., 2016; Zhao et al., 2017). The histograms for three traits are provided in Figure 6. Note that they have been normalized as in Ferrão et al. (2018). We also refer to Ferrão et al. (2018) for other basic statistical analysis and detailed preprocessing steps on this data set.

For each $i = 1, 2, 3$, we assume that $\boldsymbol{B}_i^\star$ is fully sparse. We compute the sparse matrix estimator via (16) with $\tau = \left(n/\log d_1 d_2\right)^6$ and $\lambda = \sqrt{\log d_1 d_2 / n}$. The sparse precision matrix is estimated by conducting CLIME procedure with $\gamma = 5\sqrt{\log d_2 / n}$. Throughout the experiment, we apply the hard truncation on confounders only since other variables are light-tailed already (for example, SNPs take value in $\{-1, 0, 1\}$). The estimated signal trajectories are drawn in Figure 7. From the plots, we observe that when fitting y_1 under the model (17), both confounders may cause different effect sizes for 5 out of 250 selected SNPs, and 3 of them affect common SNPs. However, confounders have no effects on selected SNPs when fitting y_2, while have dense effects when fitting y_3. In the latter case, we should mention the signals with large magnitude are still sparse.

In addition, we provide the 95% confidence interval for each trajectory using a non-parametric bootstrap: we sample the data set with replacement for 100 times and for each data set we estimate $\boldsymbol{B}_i^\star$ using the same method with the same parameters, then compute 0.025 and 0.975 quantiles to construct the confidence interval. The results are shown



(a) Production (b) Leaf rust (c) Green beans

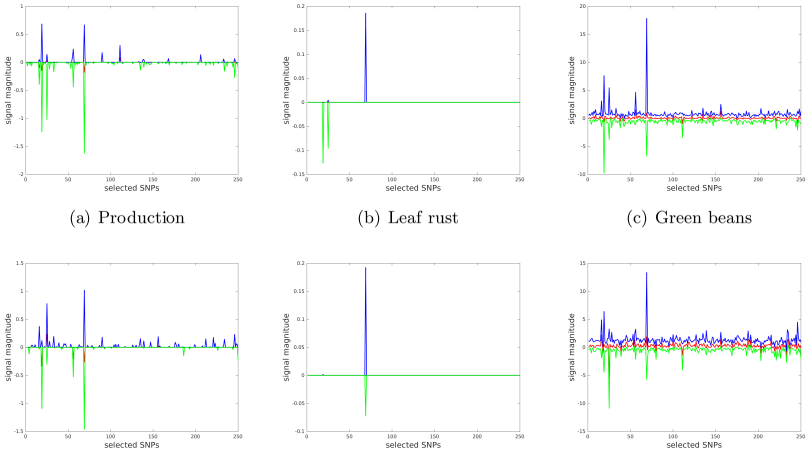
(d) Production (e) Leaf rust (f) Green beans

Figure 7: Signal trajectories. Each plot corresponds to a pair of confounders and responses, i.e. (l, i) plot shows the trajectory of $(\boldsymbol{\beta}^{l,i})^\star$ where $l = 1, 2$ indexes the confounder and $i = 1, 2, 3$ indexes the response.

in Figure 8 where the width of confidence interval indicates the variance of our estimators. We see some of identified signals have large variance but all unidentified signals have small variance. As a complement of analysis, we also conduct parametric bootstrapping: we generate new confounders x_1, x_2 from distribution in (18) for 10000 times, and for each new data set we estimate $\boldsymbol{B}_i^\star$ with the same setup, then use quantiles to construct confidence interval. The plots for parametric bootstrapping are provided in Figure 13 in Appendix D and similar observations appear again. Our code is available for download at: `https://github.com/senna1128/Varying-Index-Coefficient-Models`.

8. Conclusion

In this paper, we proposed new estimators based on Stein's identity for varying index coefficient models. By utilizing score function, we can either estimate a single sparse vector or a low-rank/sparse parameter matrix. The estimator can handle dependent covariates $\boldsymbol{z}$ through estimation of the precision matrix and can achieve optimal convergence rate in sparse estimation and near optimal rate in low-rank estimation. In all cases, the estimators we proposed have closed form and are easy to implement. Instead of assuming that covariates $\boldsymbol{x}$ follow an elliptical distribution, we only require certain finite moment assumption



(a) Production (b) Leaf rust (c) Green beans

(d) Production (e) Leaf rust (f) Green beans

Figure 8: Signal trajectories for 95% confidence interval (nonparametric bootstrapping). The (l, i) plot shows the confidence interval trajectory of $(\boldsymbol{\beta}^{l,i})^\star$ where $l = 1, 2$ indexes the confounder and $i = 1, 2, 3$ indexes the response. The blue line indicates the upper bound, the red line indicates the estimator, and the green line indicates the lower bound.

on response y, covariates $\boldsymbol{z}$, and score variable $S(\boldsymbol{x})$. We also conduct extensive numerical experiments to illustrate our result.

There are still many open problems worth exploring. One of future work is about finite moment assumption. Under the general sparsity assumption, we believe that finite 6-th moment condition is mild enough, however whether it is possible to relax it further is not clear. Furthermore, we note that all first-order Stein's estimators suffer from $\mu_k = \mathbb{E}[f_k'(\langle \boldsymbol{x}, \boldsymbol{\beta}_k^\star \rangle)] = 0$. Therefore developing a second-order Stein's estimator is of practical interest and will be explored elsewhere.

Acknowledgments

This work was completed in part with resources provided by the University of Chicago Research Computing Center. We thank Kenji Fukumizu and two anonymous reviewers for helpful comments.

Appendix A. Estimate of Sparse Precision Matrix

We propose an approach to estimate a high-dimensional sparse precision matrix for heavy-tailed variable. Suppose $\boldsymbol{z}$ has finite 4th moment and $\boldsymbol{\Omega}^\star = (\boldsymbol{\Sigma}^\star)^{-1}$, with $\boldsymbol{\Sigma}^\star = \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T]$, is column sparse. In particular, we assume that $\boldsymbol{\Omega}^\star \in \mathcal{F}_w^{K4}$ for some w and K. In this setting, we estimate the precision matrix using the CLIME procedure (Cai et al., 2011)

$$\begin{aligned} &\min \|\boldsymbol{\Omega}\|_{1,1}, \\ &\text{s.t.} \quad \|\hat{\boldsymbol{\Sigma}}\boldsymbol{\Omega} - \boldsymbol{I}_{d_2}\|_{\max} \leqslant \gamma \end{aligned} \tag{A.1}$$

but with

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} \breve{\boldsymbol{Z}}_i \breve{\boldsymbol{Z}}_i^T \tag{A.2}$$

being a thresholded estimator of the covariance matrix for some threshold $\tau > 0$, and γ is a tuning parameter. The linear program in (A.1) is the same as in Cai et al. (2011), with the difference that we use an estimator of $\boldsymbol{\Sigma}^\star$ that is more suitable for heavy-tailed data.

Lemma 15 *If $\tau = (M_4 n / \log d_2)^{1/4}/2$ and $\gamma = 12\|\boldsymbol{\Omega}^\star\|_1 \sqrt{M_4 \log d_2 / n}$, the estimator* (A.1) *satisfies*

$$P\left(\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_2 \leqslant 96 \|\boldsymbol{\Omega}^\star\|_1^2 w \sqrt{M_4 \log d_2 / n} \right) \geqslant 1 - \frac{2}{d_2^2},$$

and

$$P\left(\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_{\max} \leqslant 48 \|\boldsymbol{\Omega}^\star\|_1^2 \sqrt{M_4 \log d_2 / n} \right) \geqslant 1 - \frac{2}{d_2^2}.$$

From above lemma, we see the setting for γ in (A.1) is oracle in the sense that $\|\boldsymbol{\Omega}^\star\|_1$ is unknown. Cai et al. (2011) showed a detailed discussion on this aspect and this dependence could be removed by using a self-calibrated estimator, similar to scaled lasso (Sun and Zhang, 2013). We should also mention that (A.1) achieves the optimal rate (Cai et al., 2016).

Appendix B. Proofs of Lemmas

Throughout the proof, we frequently utilize the Bernstein's inequality presented in Corollary 2.11 in Boucheron et al. (2013). To simplify subsequent presentation, we define a function to denote the common upper bound:

$$\varphi(t, a, b) = \exp\left(-\frac{t^2/2}{a + b \cdot t/3}\right).$$

As shown in Bernstein's inequality, usually a measures the total variance and b is bound for a single variable. We also use M as the substitute of M_p (p is certain moment) for simplicity. We summarize all structures we used in the paper for future reference.

4. See definition in Section 5.

Assumption 16 (Column-wise sparse) *We assume* $\|\boldsymbol{\beta}_k^\star\|_0 \leqslant s$, $\forall k \in [d_2]$.

Assumption 17 (Fully sparse) *We assume* $\boldsymbol{B}^\star$ *is* s*-sparse:* $\|\boldsymbol{B}^\star\|_{0,1} = |supp(\boldsymbol{B}^\star)| \leqslant s$.

Assumption 18 (Low-rank) *We assume* $\boldsymbol{B}^\star$ *satisfies* $rank(\boldsymbol{B}^\star) \leqslant r$.

Assumption 19 (Independence) *We assume* $\boldsymbol{z}$ *satisfies* $\mathbb{E}[z_i z_j] = 0$, $\forall i \neq j \in [d_2]$.

Assumption 20 (Precision matrix restriction) *Define* $\boldsymbol{\Sigma}^\star = \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T]$ *and let* $\boldsymbol{\Omega}^\star = (\boldsymbol{\Sigma}^\star)^{-1}$, *we assume*

$$\boldsymbol{\Omega}^\star \in \mathcal{F}_w^K = \left\{ \boldsymbol{\Omega} \in \mathbb{R}^{d_2 \times d_2} : \|\boldsymbol{\Omega}\|_{0,\infty} \leqslant w, \|\boldsymbol{\Omega}\|_2 \leqslant K, \|\boldsymbol{\Omega}^{-1}\|_2 \leqslant K \right\}$$

for some w *and* K.

B.1. Proof of Lemma 4

Under Assumption 2, we can get from (7) that

$$\nabla \hat{L}_k(\tilde{\boldsymbol{\beta}}_k) = 2\tilde{\boldsymbol{\beta}}_k - \frac{2}{n}\sum_{i=1}^n y_i Z_{ik} \boldsymbol{X}_i \overset{(5)}{=} 2(\mathbb{E}[y z_k \cdot \boldsymbol{x}] - \frac{1}{n}\sum_{i=1}^n y_i Z_{ik} \boldsymbol{X}_i).$$

So, for any fixed $j \in [d_1]$, we have

$$[\nabla \hat{L}_k(\tilde{\boldsymbol{\beta}}_k)]_j = 2(\mathbb{E}[y z_k \cdot x_j] - \frac{1}{n}\sum_{i=1}^n y_i Z_{ik} X_{ij}). \tag{B.1}$$

Note that $z_k x_j$ is a sub-exponential random variable with

$$\|z_k x_j\|_{\psi_1} \leqslant \|z_k\|_{\psi_2} \|x_j\|_{\psi_2} \leqslant \Upsilon_{\boldsymbol{z}} \Upsilon_{\boldsymbol{x}}, \tag{B.2}$$

where $\Upsilon_{\boldsymbol{x}}$ is ψ_2-norm of a standard Gaussian variable. So we see $\{y_i, Z_{ik} X_{ij}\}_{i \in [n]}$ are n independent copies of y and $z_k x_j$. Based on Lemma C.4 in Yang et al. (2019) and (B.2), let $\gamma = \max(\Upsilon_y, \Upsilon_{\boldsymbol{x}} \Upsilon_{\boldsymbol{z}})$ and we get

$$P(|\frac{1}{n}\sum_{i=1}^n y_i Z_{ik} X_{ij} - \mathbb{E}[y z_k \cdot x_j]| > \Upsilon_\gamma \sqrt{\frac{\log n}{n}}) < \frac{1}{n^2}$$

where $\Upsilon_\gamma > 0$ only depends on γ. According to equation (B.1), we can take union bound and further have

$$P(\|\nabla \hat{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty > 2\Upsilon_\gamma \sqrt{\frac{\log n}{n}}) < \frac{d_1}{n^2},$$

which concludes the proof.

B.2. Proof of Lemma 8

Based on equation (8), we know

$$\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k) = 2\tilde{\boldsymbol{\beta}}_k - \frac{2}{n}\sum_{i=1}^n \breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}.$$

Under Assumption 2 and 19, we know $\tilde{\boldsymbol{\beta}}_k = \mathbb{E}[y z_k \cdot S(\boldsymbol{x})]$. So we can separate above gradient into two parts

$$\begin{aligned}
&\|\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty = 2\|\tilde{\boldsymbol{\beta}}_k - \frac{1}{n}\sum_{i=1}^n \breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}\|_\infty \\
\leqslant & 2\underbrace{\|\mathbb{E}[y z_k \cdot S(\boldsymbol{x})] - \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}]\|_\infty}_{\mathcal{I}_1} + 2\underbrace{\|\frac{1}{n}\sum_{i=1}^n \mathbb{E}[\breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}] - \frac{1}{n}\sum_{i=1}^n \breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}\|_\infty}_{\mathcal{I}_2}.
\end{aligned} \tag{B.3}$$

We will provide a deterministic bound for $\mathcal{I}_1$ and probabilistic bound for $\mathcal{I}_2$. Let's deal with $\mathcal{I}_1$ first. For any $j \in [d_1]$, we know

$$\begin{aligned}
\mathcal{I}_{1j} =& \mathbb{E}[y z_k \cdot S(\boldsymbol{x})_j] - \mathbb{E}[\breve{y}\breve{z}_k \cdot \widetilde{S(\boldsymbol{x})}_j] \\
=& \mathbb{E}\big[y z_k \cdot S(\boldsymbol{x})_j \cdot \mathbf{1}_{|y|>\tau \text{ or } |z_k|>\tau \text{ or } |S(\boldsymbol{x})_j|>\tau}\big] \\
\leqslant& \sqrt{\mathbb{E}[y^2 \boldsymbol{z}_k^2 S(\boldsymbol{x})_j^2] \cdot \big(P(|y|>\tau) + P(|\boldsymbol{z}_k|>\tau) + P(|S(\boldsymbol{x})_j|>\tau)\big)} \\
\leqslant& \sqrt[4]{\mathbb{E}[y^4]\mathbb{E}[z_k^4]\mathbb{E}[S(\boldsymbol{x})_j^4]} \frac{\sqrt{3}M^{1/2}}{\tau^3} \\
\leqslant& \frac{2M}{\tau^3}.
\end{aligned} \tag{B.4}$$

Here, the third inequality is from Cauchy-Schwarz inequality; the fourth inequality is Chebyshev inequality; the last inequality is due to Assumption 6 ($p = 6$). So from (B.4), we know

$$\|\mathcal{I}_1\|_\infty \leqslant 2M/\tau^3. \tag{B.5}$$

For the $\mathcal{I}_2$ term in equation (B.3), we apply Bernstein's inequality. We have $\forall j \in [d_1]$,

$$\begin{aligned}
&-\tau^3 \leqslant \breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}_j \leqslant \tau^3 \Longrightarrow C = 2\tau^3, \\
&V_n = \sum_{i=1}^n \mathrm{Var}(\breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}_j) \leqslant \sum_{i=1}^n \mathbb{E}[\breve{y}_i^2 \breve{Z}_{ik}^2 \widetilde{S(\boldsymbol{X}_i)}_j^2] \leqslant nM.
\end{aligned} \tag{B.6}$$

Thus, based on (B.6), we have $\forall t > 0$,

$$P(\big|\mathbb{E}[\breve{y}\breve{z}_k \cdot \widetilde{S(\boldsymbol{x})}_j] - \frac{1}{n}\sum_{i=1}^n \breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}_j\big| > t) \leqslant 2\varphi(nt, nM, 2\tau^3). \tag{B.7}$$

Then we take union bound for (B.7) and get

$$P(\|\mathcal{I}_2\|_\infty > t) \leqslant 2d_1\varphi(nt, nM, 2\tau^3) \tag{B.8}$$

Combine (B.5) and (B.8) together, and take union bound over k, we have $\forall t, \tau > 0$,

$$P(\|\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty \leqslant \frac{4M}{\tau^3} + 2t, \quad \forall k \in [d_2]) \geqslant 1 - 2d_1d_2\exp(-\frac{nt^2}{2M + 2\tau^3 t}). \tag{B.9}$$

Suppose, for some positive constant c_1, c_2, we let

$$t = c_1\sqrt{\log d_1d_2/n} \quad \text{and} \quad \tau = c_2^{1/3}(n/\log d_1d_2)^{1/6}. \tag{B.10}$$

Then plug (B.10) into the right hand side of (B.9) and get

$$2d_1d_2\exp(-\frac{nt^2}{2M + 2\tau^3 t}) = 2d_1d_2\exp(-\frac{c_1^2\log d_1d_2}{2M + 2c_1c_2}) \leqslant 2/d_1^2d_2^2, \tag{B.11}$$

if

$$\frac{c_1^2}{2M + 2c_1c_2} \geqslant 3 \Longrightarrow c_1^2 - 6c_1c_2 - 6M \geqslant 0. \tag{B.12}$$

We can set $c_1 = 3\sqrt{M}$ and $c_2 = \sqrt{M}/8$, which satisfies condition in (B.12) naturally. Further we know (B.11) holds. Plug this setting in (B.10) and (B.9) and we get

$$\|\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty \leqslant 38\sqrt{M\log d_1d_2/n}, \quad \forall k \in [d_2] \tag{B.13}$$

with probability at least $1 - 2/d_1^2d_2^2$. This finishes the proof.

B.3. Proof of Lemma 10

Define $\mathcal{I}_3 = \frac{1}{n\kappa_1}\sum_{i=1}^n \Phi(\kappa_1 y_i \cdot S(\boldsymbol{X}_i)\boldsymbol{Z}_i^T) - \mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T]$, we will apply Corollary 3.1 in Minsker (2018). Let's first bound the variance. Under Assumption 2, we know $\mathbb{E}[S(\boldsymbol{x})_j] = 0, \forall j \in [d_1]$. So for any unit vector $\boldsymbol{v} \in \mathbb{R}^{d_1}$, we have

$$\begin{aligned}
&\mathbb{E}[y^2 \cdot \boldsymbol{v}^T S(\boldsymbol{x})\boldsymbol{z}^T\boldsymbol{z}S(\boldsymbol{x})^T\boldsymbol{v}] = \mathbb{E}[y^2 \cdot \boldsymbol{z}^T\boldsymbol{z} \cdot (S(\boldsymbol{x})^T\boldsymbol{v})^2] \leqslant \sqrt{\mathbb{E}[y^4]\mathbb{E}[(\boldsymbol{z}^T\boldsymbol{z})^2]\mathbb{E}[(S(\boldsymbol{x})^T\boldsymbol{v})^4]} \\
&\leqslant M^{1/2}\sqrt{\mathbb{E}[d_2(\boldsymbol{z}_1^4 + ... + \boldsymbol{z}_{d_2}^4)]}\sqrt{\mathbb{E}[\sum_{i_1=1}^{d_1}\sum_{i_2=1}^{d_1} S(\boldsymbol{x})_{i_1}^2 S(\boldsymbol{x})_{i_2}^2 \boldsymbol{v}_{i_1}^2\boldsymbol{v}_{i_2}^2]} \\
&\leqslant d_2M\sqrt{\sum_{i_1=1}^{d_1}\sum_{i_2=1}^{d_1}\mathbb{E}[S(\boldsymbol{x})_{i_1}^2 S(\boldsymbol{x})_{i_2}^2]\boldsymbol{v}_{i_1}^2\boldsymbol{v}_{i_2}^2} \leqslant d_2M\sqrt{\sum_{i_1=1}^{d_1}\sum_{i_2=1}^{d_1}\sqrt{\mathbb{E}[S(\boldsymbol{x})_{i_1}^4]}\sqrt{\mathbb{E}[S(\boldsymbol{x})_{i_2}^4]}\boldsymbol{v}_{i_1}^2\boldsymbol{v}_{i_2}^2} \\
&\leqslant d_2M^{3/2}.
\end{aligned} \tag{B.14}$$

The second inequality uses Cauchy-Schwarz inequality; the third inequality uses Assumption 2. From (B.14) we can get

$$\|\mathbb{E}[y^2 \cdot S(\boldsymbol{x})\boldsymbol{z}^T\boldsymbol{z}S(\boldsymbol{x})^T]\|_2 \leqslant d_2M^{3/2}. \tag{B.15}$$

Follow the exactly same derivation in (B.14) we can also have

$$\|\mathbb{E}[y^2 \cdot \boldsymbol{z}S(\boldsymbol{x})^T S(\boldsymbol{x})\boldsymbol{z}^T]\|_2 \leqslant d_1M^{3/2}. \tag{B.16}$$

Thus, combine (B.15) and (B.16) together, we have $\forall t > 0$

$$P(\|\mathcal{I}_3\|_2 \geqslant t) \leqslant 2(d_1 + d_2)\exp(-n\kappa_1 t + \frac{n(d_1 + d_2)M^{3/2}\kappa_1^2}{2}). \tag{B.17}$$

In above tail bound (B.17), we let $t = 2M^{3/4}\sqrt{\frac{2(d_1+d_2)\log(d_1+d_2)}{n}}$ and $\kappa_1 = \sqrt{\frac{2\log(d_1+d_2)}{n(d_1+d_2)M^{3/2}}}$ and have

$$P\left(\|\mathcal{I}_3\| \leqslant 2M^{3/4}\sqrt{\frac{2(d_1 + d_2)\log(d_1 + d_2)}{n}}\right) \geqslant 1 - \frac{2}{(d_1 + d_2)^2}. \tag{B.18}$$

This is consistent with argument of lemma.

B.4. Proof of Lemma 12

Let's first get concentration rate for $\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_2 = \|\frac{1}{n\kappa_2}\Phi(\kappa_2 Z_i Z_i^T) - \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T]\|_2$. We have $\forall \boldsymbol{v} \in \mathbb{R}^{d_2}$ such that $\|\boldsymbol{v}\|_2 = 1$,

$$\mathbb{E}[\boldsymbol{v}^T\boldsymbol{z}\boldsymbol{z}^T\boldsymbol{v}] = \mathbb{E}[(\boldsymbol{v}^T\boldsymbol{z})^2] \leqslant \mathbb{E}[\|\boldsymbol{z}\|_2^2] \leqslant d_2\sqrt{M}.$$

Based on Corollary 3.1 in Minsker (2018), we know $\forall t > 0$,

$$P(\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_2 \geqslant t) \leqslant 2d_2\exp(-n\kappa_2 t + \frac{nd_2\sqrt{M}\kappa_2^2}{2}). \tag{B.19}$$

In above (B.19), we let $t = 2M^{1/4}\sqrt{\frac{2d_2\log d_2}{n}}$ and $\kappa_2 = \sqrt{\frac{2\log d_2}{nd_2M^{1/2}}}$, and have

$$P\left(\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_2 \leqslant 2M^{1/4}\sqrt{\frac{2d_2\log d_2}{n}}\right) \geqslant 1 - \frac{2}{d_2^2}. \tag{B.20}$$

We use matrix perturbation analysis to give bound for $\hat{\boldsymbol{\Omega}}$. As shown in Chapter III Theorem 2.5 in Stewart and Sun (1990), when

$$\|\boldsymbol{\Omega}^\star(\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star)\|_2 \leqslant \|\boldsymbol{\Omega}^\star\|_2\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_2 \leqslant \|\boldsymbol{\Omega}^\star\|_2 4M^{1/4}\sqrt{d_2\log d_2/n} \leqslant 1/2,$$

we know $\hat{\boldsymbol{\Omega}}$ is perforce invertible and satisfies

$$\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_2 \leqslant 2\|\boldsymbol{\Omega}^\star\|_2^2\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_2 \leqslant 8\|\boldsymbol{\Omega}^\star\|_2^2 M^{1/4}\sqrt{d_2\log d_2/n}, \tag{B.21}$$

with probability at least $1 - 2/d_2^2$. Therefore we finish the proof.

B.5. Proof of Lemma 14

We define $\mathcal{I}_7 = \mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T] - \frac{1}{n}\sum_{i=1}^n \breve{y}_i \cdot \widetilde{S(\boldsymbol{X}_i)}\breve{\boldsymbol{Z}}_i^T$. For any $j \in [d_1], k \in [d_2]$, we know

$$|(\mathcal{I}_7)_{jk}| \leqslant |\frac{1}{n}\sum_{i=1}^n \breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}_j - \mathbb{E}[\breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}_j]| + |\frac{1}{n}\sum_{i=1}^n \mathbb{E}[\breve{y}_i \breve{Z}_{ik} \widetilde{S(\boldsymbol{X}_i)}_j] - \mathbb{E}[y \cdot z_k \cdot S(\boldsymbol{x})_j]|.$$

Use the same technique as in (B.4)-(B.8), we can get that $\forall t, \tau > 0$

$$P(\|\mathcal{I}_7\|_{\max} > t + \frac{2M}{\tau^3}) \leqslant 2d_1 d_2 \exp(-\frac{nt^2}{2M + 2\tau^3 t}).$$

We let $t = 3\sqrt{M \log d_1 d_2 / n}$, $\tau = (Mn/\log d_1 d_2)^{1/6}/2$ and have

$$P\left(\|\mathcal{I}_7\|_{\max} \leqslant 19\sqrt{\frac{M \log d_1 d_2}{n}}\right) \geqslant 1 - 2/d_1^2 d_2^2. \tag{B.22}$$

So we finish the proof of lemma.

B.6. Proof of Lemma 15

We first prove a concentration bound for truncated empirical covariance $\hat{\boldsymbol{\Sigma}}$. $\forall j, k \in [d_2]$,

$$\begin{aligned}|\hat{\Sigma}_{jk} - \Sigma^\star_{jk}| &= |\frac{1}{n}\sum_{i=1}^n \breve{Z}_{ij}\breve{Z}_{ik} - \mathbb{E}[z_j z_k]| \\ &\leqslant \underbrace{|\frac{1}{n}\sum_{i=1}^n (\breve{Z}_{ij}\breve{Z}_{ik} - \mathbb{E}[\breve{Z}_{ij}\breve{Z}_{ik}])|}_{\mathcal{I}_5} + \underbrace{|\frac{1}{n}\sum_{i=1}^n \mathbb{E}[\breve{Z}_{ij}\breve{Z}_{ik}] - \mathbb{E}[z_j z_k]|}_{\mathcal{I}_6}.\end{aligned} \tag{B.23}$$

Use Bernstein's inequality for $\mathcal{I}_5$, we have

$$\begin{aligned}&-\tau^2 \leqslant \breve{Z}_{ij}\breve{Z}_{ik} \leqslant \tau^2, \\ &V_n = \sum_{i=1}^n \mathrm{Var}(\breve{Z}_{ij}\breve{Z}_{ik}) \leqslant \sum_{i=1}^n \mathbb{E}[\breve{Z}_{ij}^2 \breve{Z}_{ik}^2] \leqslant nM.\end{aligned}$$

Note that the above last inequality holds no matter whether $j = k$ or not. Thus, $\forall t > 0$, we have

$$P(|\mathcal{I}_5| > t) \leqslant 2\varphi(nt, nM, 2\tau^2). \tag{B.24}$$

For the term $\mathcal{I}_6$, we know

$$|\mathcal{I}_6| = \mathbb{E}[z_j z_k \cdot \mathbf{1}_{\{|z_j|>\tau \text{ or } |z_k|>\tau\}}] \leqslant \sqrt{\mathbb{E}[z_j^2 z_k^2] \cdot (P(|z_j| > \tau) + P(|z_k| > \tau))} \leqslant \frac{2M}{\tau^2}. \tag{B.25}$$

Combine (B.23), (B.24) and (B.25) together, we get

$$|\hat{\Sigma}_{jk} - \Sigma^\star_{jk}| \leqslant t + \frac{2M}{\tau^2}, \tag{B.26}$$

with probability at least $1 - 2\exp(-\frac{nt^2}{2M+2\tau^2 t})$. Take union bound for (B.26) we have

$$P(\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_{\max} \leqslant t + \frac{2M}{\tau^2}) \geqslant 1 - 2d_2^2\varphi(nt, nM, 2\tau^2).$$

Let $t = 4\sqrt{M \log d_2 / n}$, $\tau = (Mn/\log d_2)^{1/4}/2$, we have

$$P\left(\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_{\max} \leqslant 12\sqrt{\frac{M \log d_2}{n}}\right) \geqslant 1 - \frac{2}{d_2^2}. \tag{B.27}$$

Based on this bound, we deal with convex problem (A.1). Suppose $\hat{\boldsymbol{\Omega}} = (\hat{\boldsymbol{\omega}}_1, ..., \hat{\boldsymbol{\omega}}_{d_2})$, we will show each $\hat{\boldsymbol{\omega}}_j$ is also a solution to following problem:

$$\begin{aligned}&\min_{\boldsymbol{l}_j} \|\boldsymbol{l}_j\|_1, \\ &\text{s.t. } \|\hat{\boldsymbol{\Sigma}}\boldsymbol{l}_j - \boldsymbol{e}_j\|_\infty \leqslant \gamma.\end{aligned} \tag{B.28}$$

In fact, it's easy to see $\hat{\boldsymbol{\omega}}_j$ is a feasible point for problem (B.28), so $\|\hat{\boldsymbol{l}}_j\|_1 \leqslant \|\hat{\boldsymbol{\omega}}_j\|_1$. Further we know $\|(\hat{\boldsymbol{l}}_1, ..., \hat{\boldsymbol{l}}_{d_2})\|_{1,1} \leqslant \|\hat{\boldsymbol{\Omega}}\|_{1,1}$. On the other hand, $\|\hat{\boldsymbol{\omega}}_j\|_1 \leqslant \|\hat{\boldsymbol{l}}_j\|_1$ for sure. Otherwise $(\hat{\boldsymbol{\omega}}_1, ..., \hat{\boldsymbol{l}}_j, ..., \hat{\boldsymbol{\omega}}_{d_2})$ satisfies condition of (A.1) but with smaller objective value. In this case, we know each $\hat{\boldsymbol{\omega}}_j$ can also be solved from (B.28). Note for $\boldsymbol{\Omega}^\star = (\boldsymbol{\omega}_1^\star, ..., \boldsymbol{\omega}_{d_2}^\star) \in \mathbb{R}^{d_2 \times d_2}$, we have

$$\|\hat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^\star - \boldsymbol{I}_{d_2}\|_{\max} = \|(\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star + \boldsymbol{\Sigma}^\star)\boldsymbol{\Omega}^\star - \boldsymbol{I}_{d_2}\|_{\max} = \|(\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star)\boldsymbol{\Omega}^\star\|_{\max} \leqslant \|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_{\max}\|\boldsymbol{\Omega}^\star\|_1.$$

So when $\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_{\max}\|\boldsymbol{\Omega}^\star\|_1 \leqslant \gamma$, we know $\boldsymbol{\Omega}^\star$ is feasible for problem (A.1) and $\boldsymbol{\omega}_j^\star$ is feasible for problem (B.28). So we know

$$\|\hat{\boldsymbol{\Omega}}\|_{1,1} \leqslant \|\boldsymbol{\Omega}^\star\|_{1,1} \quad \text{and} \quad \|\hat{\boldsymbol{\omega}}_j\|_1 \leqslant \|\boldsymbol{\omega}_j^\star\|_1. \tag{B.29}$$

Based on (B.29), we know $\|\hat{\boldsymbol{\Omega}}\|_1 \leqslant \|\boldsymbol{\Omega}^\star\|_1$. Further, we have

$$\begin{aligned}\|\boldsymbol{\Sigma}^\star(\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star)\|_{\max} \leqslant& \|(\boldsymbol{\Sigma}^\star - \hat{\boldsymbol{\Sigma}})(\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star)\|_{\max} + \|\hat{\boldsymbol{\Sigma}}(\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star)\|_{\max} \\ \leqslant& \|\boldsymbol{\Sigma}^\star - \hat{\boldsymbol{\Sigma}}\|_{\max}\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_1 + \|\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Omega}} - \boldsymbol{I}_{d_2}\|_{\max} + \|\hat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^\star - \boldsymbol{I}_{d_2}\|_{\max} \\ \leqslant& \|\boldsymbol{\Sigma}^\star - \hat{\boldsymbol{\Sigma}}\|_{\max}(\|\hat{\boldsymbol{\Omega}}\|_1 + \|\boldsymbol{\Omega}^\star\|_1) + 2\gamma \\ \leqslant& 2\|\boldsymbol{\Sigma}^\star - \hat{\boldsymbol{\Sigma}}\|_{\max}\|\boldsymbol{\Omega}^\star\|_1 + 2\gamma \\ \leqslant& 4\gamma.\end{aligned} \tag{B.30}$$

Based on the bound (B.30), we have

$$\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_{\max} = \|\boldsymbol{\Omega}^\star\boldsymbol{\Sigma}^\star(\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star)\|_{\max} \leqslant \|\boldsymbol{\Omega}^\star\|_\infty\|\boldsymbol{\Sigma}^\star(\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star)\|_{\max} \leqslant 4\gamma\|\boldsymbol{\Omega}^\star\|_1. \tag{B.31}$$

For the last inequality in (B.31), we use $\|\boldsymbol{\Omega}^\star\|_1 = \|\boldsymbol{\Omega}^\star\|_\infty$ because $\boldsymbol{\Omega}^\star$ is symmetric matrix. To proceed, let's derive the cone condition. Define $\Delta_j = \hat{\boldsymbol{\omega}}_j - \boldsymbol{\omega}_j^\star$ and $s_j = \mathrm{supp}(\boldsymbol{\omega}_j^\star)$. From equation (B.29) we know

$$\|\boldsymbol{\omega}_j^\star\|_1 \geqslant \|\hat{\boldsymbol{\omega}}_j\|_1 = \|\Delta_j + \boldsymbol{\omega}_j^\star\|_1 = \|(\Delta_j + \boldsymbol{\omega}_j^\star)_{s_j}\|_1 + \|(\Delta_j)_{s_j^c}\|_1 \Longrightarrow \|(\Delta_j)_{s_j}\|_1 \geqslant \|(\Delta_j)_{s_j^c}\|_1. \tag{B.32}$$

Combine the cone condition in (B.32) with (B.31) and Assumption 20, we know

$$\|\Delta_j\|_1 \leqslant 2\|(\Delta_j)_{s_j}\|_1 \leqslant 2w\|\Delta_j\|_\infty = 2w\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_{\max} \leqslant 8\|\boldsymbol{\Omega}^\star\|_1 w\gamma. \tag{B.33}$$

Therefore, if $\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^\star\|_{\max}\|\boldsymbol{\Omega}^\star\|_1 \leqslant \gamma$, then

$$\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_2 \leqslant \sqrt{\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_1\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_\infty} = \|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_1 \leqslant 8\|\boldsymbol{\Omega}^\star\|_1 w\gamma. \tag{B.34}$$

Based on (B.27) and (B.34), we can choose $\gamma = 12\|\boldsymbol{\Omega}^\star\|_1\sqrt{M\log d_2/n}$, then with probability at least $1 - 2/d_2^2$, we have

$$\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_2 \leqslant 96\|\boldsymbol{\Omega}^\star\|_1^2 w\sqrt{M\log d_2/n}. \tag{B.35}$$

Further, from (B.31), we know with probability at least $1 - 2/d_2^2$

$$\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star\|_{\max} \leqslant 4\|\boldsymbol{\Omega}^\star\|_1\gamma \leqslant 48\|\boldsymbol{\Omega}^\star\|_1^2\sqrt{M\log d_2/n}. \tag{B.36}$$

This concludes the proof.

Appendix C. Proofs of Theorems and Corollaries

C.1. Proof of Theorem 3

Let's fix $k \in [d_2]$ first. Based on the definition of $\hat{\boldsymbol{\beta}}_k$ in (7), we have following basic inequality

$$\hat{L}_k(\hat{\boldsymbol{\beta}}_k) + \lambda_k\|\hat{\boldsymbol{\beta}}_k\|_1 \leqslant \hat{L}_k(\tilde{\boldsymbol{\beta}}_k) + \lambda_k\|\tilde{\boldsymbol{\beta}}_k\|_1. \tag{C.1}$$

We define $\boldsymbol{\theta}_k = \hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k$ and have

$$\begin{aligned}\hat{L}_k(\hat{\boldsymbol{\beta}}_k) - \hat{L}_k(\tilde{\boldsymbol{\beta}}_k) &= \|\boldsymbol{\theta}_k\|_2^2 + 2\langle\tilde{\boldsymbol{\beta}}_k, \boldsymbol{\theta}_k\rangle - \frac{2}{n}\sum_{i=1}^n y_i Z_{ik}\langle\boldsymbol{X}_i, \boldsymbol{\theta}_k\rangle \\ &= \|\boldsymbol{\theta}_k\|_2^2 + \langle\nabla\hat{L}_k(\tilde{\boldsymbol{\beta}}_k), \boldsymbol{\theta}_k\rangle. \end{aligned}\tag{C.2}$$

Given a vector $\boldsymbol{v} \in \mathbb{R}^d$ and an index set $\mathcal{I} \subset [d]$, we define $\boldsymbol{v}_{\mathcal{I}} \in \mathbb{R}^d$ to be $\boldsymbol{v}$ restricted on $\mathcal{I}$ as $[\boldsymbol{v}_{\mathcal{I}}]_i = \boldsymbol{v}_i$ if $i \in \mathcal{I}$ and 0 otherwise. Suppose S_k is the support of $\tilde{\boldsymbol{\beta}}_k$, which is the same as $\boldsymbol{\beta}_k^\star$, combine (C.1) and (C.2) together and we get

$$\begin{aligned}\|\boldsymbol{\theta}_k\|_2^2 &\leqslant -\langle\nabla\hat{L}_k(\tilde{\boldsymbol{\beta}}_k), \boldsymbol{\theta}_k\rangle + \lambda_k\|\tilde{\boldsymbol{\beta}}_k\|_1 - \lambda_k\|\hat{\boldsymbol{\beta}}_k\|_1 \\ &= -\langle\nabla\hat{L}_k(\tilde{\boldsymbol{\beta}}_k), \boldsymbol{\theta}_k\rangle + \lambda_k\|(\tilde{\boldsymbol{\beta}}_k)_{S_k}\|_1 - \lambda_k\|(\hat{\boldsymbol{\beta}}_k)_{S_k}\|_1 - \lambda_k\|(\hat{\boldsymbol{\beta}}_k)_{S_k^C}\|_1 \\ &\leqslant -\langle\nabla\hat{L}_k(\tilde{\boldsymbol{\beta}}_k), \boldsymbol{\theta}_k\rangle + \lambda_k\|(\boldsymbol{\theta}_k)_{S_k}\|_1 - \lambda_k\|(\boldsymbol{\theta}_k)_{S_k^C}\|_1 \\ &\leqslant \|\nabla\hat{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty\|\boldsymbol{\theta}_k\|_1 + \lambda_k\|(\boldsymbol{\theta}_k)_{S_k}\|_1 - \lambda_k\|(\boldsymbol{\theta}_k)_{S_k^C}\|_1, \end{aligned}\tag{C.3}$$

where the third inequality is from triangle inequality and the last is based on Hölder's inequality. If we set $\lambda_k = 4\Upsilon_\gamma\sqrt{\log n/n}$, based on Lemma 4, we have

$$\|\nabla\hat{L}(\tilde{\boldsymbol{\beta}}_k)\|_\infty \leqslant \frac{\lambda_k}{2} \tag{C.4}$$

with probability at least $1 - d_1/n^2$. Combine (C.3) and (C.4), we know with probability at least $1 - d_1/n^2$,

$$\|\boldsymbol{\theta}_k\|_2^2 \leqslant \frac{3\lambda_k}{2}\|(\boldsymbol{\theta}_k)_{S_k}\|_1 - \frac{\lambda_k}{2}\|(\boldsymbol{\theta}_k)_{S_k^C}\|_1. \tag{C.5}$$

From (C.5) we further get the following cone condition:

$$\|(\boldsymbol{\theta}_k)_{S_k^C}\|_1 \leqslant 3\|(\boldsymbol{\theta}_k)_{S_k}\|_1. \tag{C.6}$$

Also from equation (C.5) and sparsity condition we know

$$\|\boldsymbol{\theta}_k\|_2^2 \leqslant \frac{3}{2}\lambda_k\|(\boldsymbol{\theta}_k)_{S_k}\|_1 \leqslant \frac{3}{2}\lambda_k\sqrt{s}\|(\boldsymbol{\theta}_k)_{S_k}\|_2 \leqslant \frac{3}{2}\lambda_k\sqrt{s}\|\boldsymbol{\theta}_k\|_2.$$

So we have with probability at least $1 - d_1/n^2$,

$$\|\boldsymbol{\theta}_k\|_2 \leqslant \frac{3}{2}\sqrt{s}\lambda_k.$$

Further by cone condition in (C.6) we can get l_1-norm convergence rate as

$$\|\boldsymbol{\theta}_k\|_1 = \|(\boldsymbol{\theta}_k)_{S_k}\|_1 + \|(\boldsymbol{\theta}_k)_{S_k^C}\|_1 \leqslant 4\|(\boldsymbol{\theta}_k)_{S_k}\|_1 \leqslant 4\sqrt{s}\|(\boldsymbol{\theta}_k)_{S_k}\|_2 \leqslant 6s\lambda_k.$$

By taking the union bound, it's easy to have

$$P(\|\boldsymbol{\theta}_k\|_2 \leqslant \frac{3}{2}\sqrt{s}\lambda_k \text{ and } \|\boldsymbol{\theta}_k\|_2 \leqslant 6s\lambda_k, \forall k) \geqslant 1 - \frac{d_2 d_1}{n^2}.$$

C.2. Proof of Corollary 5

We still fix $k \in [d_2]$ first and then take union bound. Denote $\mu = \min_{j\in[d_2]}|\mu_j|$, from Theorem 3, we know there exists $N(s, \mu)$ such that whenever $n \geqslant N$, we have

$$\|\hat{\boldsymbol{\beta}}_k\|_2 \geqslant \mu - \|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \geqslant \mu - 6\Upsilon\sqrt{s\log n/n} \geqslant \mu/2. \tag{C.7}$$

with probability at least $1 - d_1/n^2$. For either l_2-norm or l_1-norm, use (C.7) and we can get

$$\begin{aligned}\|\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \frac{\tilde{\boldsymbol{\beta}}_k}{|\mu_k|}\| &= \frac{\|\hat{\boldsymbol{\beta}}_k - \|\hat{\boldsymbol{\beta}}_k\|_2/|\mu_k|\cdot\tilde{\boldsymbol{\beta}}_k\|}{\|\hat{\boldsymbol{\beta}}_k\|_2} \leqslant \frac{\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\| + \big||\mu_k| - \|\hat{\boldsymbol{\beta}}_k\|_2\big|\|\boldsymbol{\beta}_k^\star\|}{\|\hat{\boldsymbol{\beta}}_k\|_2} \\ &\leqslant \frac{2}{\mu}\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\| + \frac{2}{\mu}\|\boldsymbol{\beta}_k^\star\|\cdot\big|\|\tilde{\boldsymbol{\beta}}_k\|_2 - \|\hat{\boldsymbol{\beta}}_k\|_2\big| \\ &\leqslant \frac{2}{\mu}\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\| + \frac{2}{\mu}\|\boldsymbol{\beta}_k^\star\|\cdot\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2. \end{aligned}\tag{C.8}$$

Thus, combine (C.8) with Theorem 3, we know with probability $1 - d_1/n^2$

$$\begin{aligned}\|\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \frac{\tilde{\boldsymbol{\beta}}_k}{|\mu_k|}\|_2 &\leqslant \frac{4}{\mu}\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \lesssim \sqrt{s\log n/n}/\mu, \\ \|\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \frac{\tilde{\boldsymbol{\beta}}_k}{|\mu_k|}\|_1 &\leqslant \frac{2}{\mu}\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_1 + \frac{2\sqrt{s}}{\mu}\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \lesssim s\sqrt{\log n/n}/\mu. \end{aligned}$$

Note that under identifiability condition (2) we have $\boldsymbol{\beta}_k^\star = \text{sign}(\tilde{\beta}_{k1}) \cdot \frac{\tilde{\boldsymbol{\beta}}_k}{|\mu_k|}$, hence there exists $M(s, N, \min_{j\in[d_2]} \beta_{j1}^\star)$, such that $n \geqslant M$, $\text{sign}(\hat{\beta}_{k1}) = \text{sign}(\tilde{\beta}_{k1})$. So we can get for either l_2-norm or l_1-norm

$$\|\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \frac{\tilde{\boldsymbol{\beta}}_k}{|\mu_k|}\| = \|\text{sign}(\hat{\beta}_{k1})\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \text{sign}(\hat{\beta}_{k1})\frac{\tilde{\boldsymbol{\beta}}_k}{|\mu_k|}\| = \|\text{sign}(\hat{\beta}_{k1})\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \boldsymbol{\beta}_k^\star\|.$$

By taking the union bound, we can get the conclusion. Particularly, in the worst case, we have

$$\|\hat{\boldsymbol{B}} - \boldsymbol{B}^\star\|_F^2 = \sum_{j=1}^{d_2} \|\text{sign}(\hat{\beta}_{k1})\frac{\hat{\boldsymbol{\beta}}_k}{\|\hat{\boldsymbol{\beta}}_k\|_2} - \boldsymbol{\beta}_k^\star\|_2^2 \lesssim d_2 s \log n / n\mu^2.$$

So, we know $\|\hat{\boldsymbol{B}} - \boldsymbol{B}^\star\|_F \lesssim \frac{1}{\mu}\sqrt{\frac{d_2 s \log n}{n}}$. This concludes the proof.

C.3. Proof of Theorem 7

We still fix $k \in [d_2]$ first. Start from the definition of $\hat{\boldsymbol{\beta}}_k$ in (8) and basic inequality, then follow the same steps as in (C.1), (C.2), and (C.3), we finally get

$$\|\boldsymbol{\theta}_k\|_2^2 \leqslant \|\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty \|\boldsymbol{\theta}_k\|_1 + \lambda_k \|(\boldsymbol{\theta}_k)_{\mathcal{S}_k}\|_1 - \lambda_k \|(\boldsymbol{\theta}_k)_{\mathcal{S}_k^C}\|_1. \tag{C.9}$$

Based on Lemma 8, we can let $\lambda_k = 76\sqrt{M \log d_1 d_2 / n}$ and have $\|\nabla \bar{L}_k(\tilde{\boldsymbol{\beta}}_k)\|_\infty \leqslant \lambda_k/2$. Plug into (C.9) and follow the derivation in equation (C.5) and (C.6), we can finally get

$$\|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_2 \leqslant \frac{3}{2}\sqrt{s}\lambda_k \quad \text{and} \quad \|\hat{\boldsymbol{\beta}}_k - \tilde{\boldsymbol{\beta}}_k\|_1 \leqslant 6s\lambda_k.$$

Note that above error bound holds uniformly over $k \in [d_2]$ and we finish the proof.

C.4. Proof of Theorem 9

To make notation consistent, let's denote the loss function without penalty defined in (13) by $\hat{L}(\boldsymbol{B})$ and define $\mathcal{I}_4 = \hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star$, then we have

$$\begin{aligned}\nabla \hat{L}(\tilde{\boldsymbol{B}}) =& 2\tilde{\boldsymbol{B}} - \frac{2}{n\kappa_1}\sum_{i=1}^n \Phi(\kappa_1 y_i \cdot S(X_i) Z_i^T)\hat{\boldsymbol{\Omega}} \\ \overset{(9)}{=}& 2\left(\mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T]\boldsymbol{\Omega}^\star - \frac{1}{n\kappa_1}\sum_{i=1}^n \Phi(\kappa_1 y_i \cdot S(X_i) Z_i^T)\hat{\boldsymbol{\Omega}}\right).\end{aligned} \tag{C.10}$$

Based on equation (C.10), we use triangle inequality and get

$$\|\nabla \hat{L}(\tilde{\boldsymbol{B}})\|_2 \tag{C.11}$$

$$\leqslant 2\|\underbrace{\mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T] - \frac{1}{n\kappa_1}\sum_{i=1}^n \Phi(\kappa_1 y_i \cdot S(X_i) Z_i^T)}_{\mathcal{I}_3}\|_2 \|\hat{\boldsymbol{\Omega}}\|_2 + 2\|\mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T]\|_2 \|\underbrace{\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star}_{\mathcal{I}_4}\|_2$$

$$\leqslant 2\|\mathcal{I}_3\|_2\|\mathcal{I}_4\|_2 + \underbrace{2\|\boldsymbol{\Omega}^\star\|_2\|\mathcal{I}_3\|_2 + 2\|\mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T]\|_2\|\mathcal{I}_4\|_2}_{\text{dominant term}}. \tag{C.12}$$

Note that

$$\|\mathbb{E}[y \cdot S(\boldsymbol{x})\boldsymbol{z}^T]\|_2 = \|\tilde{\boldsymbol{B}}\boldsymbol{\Sigma}^\star\|_2 \leqslant \max_{j\in[d_2]} |\mu_j| \cdot \|\boldsymbol{B}^\star\|_2 \|\boldsymbol{\Sigma}^\star\|_2. \tag{C.13}$$

So combine (B.18), (C.11), (C.13) together and drop off smaller order term, we can get

$$\begin{aligned}P\Big(\|\nabla \hat{L}(\tilde{\boldsymbol{B}})\|_2 \leqslant& 8KM^{3/4}\sqrt{\frac{(d_1+d_2)\log(d_1+d_2)}{n}} + 2K\max_{j\in[d_2]}|\mu_j| \cdot \|\boldsymbol{B}^\star\|_2 \mathcal{H}(n, d_2)\Big) \\ \geqslant& 1 - \frac{2}{(d_1+d_2)^2} - \mathcal{P}(n, d_2).\end{aligned} \tag{C.14}$$

So we know under the setup of λ as in theorem, we have $\|\nabla \hat{L}(\tilde{\boldsymbol{B}})\|_2 \leqslant \lambda/2$ with probability at least $1 - 2/(d_1+d_2)^2 - \mathcal{P}(n, d_2)$. On the other side, start from the definition of $\hat{\boldsymbol{B}}$ in (13), we have following basic inequality

$$\hat{L}(\hat{\boldsymbol{B}}) + \lambda\|\hat{\boldsymbol{B}}\|_* \leqslant \hat{L}(\tilde{\boldsymbol{B}}) + \lambda\|\tilde{\boldsymbol{B}}\|_*. \tag{C.15}$$

Define $\boldsymbol{\Theta} = \hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}$, we have

$$\begin{aligned}\hat{L}(\hat{\boldsymbol{B}}) - \hat{L}(\tilde{\boldsymbol{B}}) =& \|\hat{\boldsymbol{B}}\|_F^2 - \|\tilde{\boldsymbol{B}}\|_F^2 - \frac{2}{n\kappa_1}\sum_{i=1}^n \langle \Phi(\kappa_1 y_i \cdot S(\boldsymbol{X}_i)\boldsymbol{Z}_i^T)\hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}\rangle \\ =& \|\boldsymbol{\Theta}\|_F^2 + 2\langle \tilde{\boldsymbol{B}}, \boldsymbol{\Theta}\rangle - \frac{2}{n\kappa_1}\sum_{i=1}^n \langle \Phi(\kappa_1 y_i \cdot S(\boldsymbol{X}_i)\boldsymbol{Z}_i^T)\hat{\boldsymbol{\Omega}}, \boldsymbol{\Theta}\rangle \\ =& \langle \nabla \hat{L}(\tilde{\boldsymbol{B}}), \boldsymbol{\Theta}\rangle + \|\boldsymbol{\Theta}\|_F^2.\end{aligned} \tag{C.16}$$

Combine (C.15) and (C.16) together, we have

$$\begin{aligned}\|\boldsymbol{\Theta}\|_F^2 =& -\langle \nabla \hat{L}(\tilde{\boldsymbol{B}}), \boldsymbol{\Theta}\rangle + \hat{L}(\hat{\boldsymbol{B}}) - \hat{L}(\tilde{\boldsymbol{B}}) \\ \leqslant& -\langle \nabla \hat{L}(\tilde{\boldsymbol{B}}), \boldsymbol{\Theta}\rangle + \lambda\|\tilde{\boldsymbol{B}}\|_* - \lambda\|\hat{\boldsymbol{B}}\|_* \\ \leqslant& \|\nabla \hat{L}(\tilde{\boldsymbol{B}})\|_2\|\boldsymbol{\Theta}\|_* + \lambda\|\tilde{\boldsymbol{B}}\|_* - \lambda\|\hat{\boldsymbol{B}}\|_*.\end{aligned} \tag{C.17}$$

Under Assumption 2 and 18, we know $r = \text{rank}(\boldsymbol{B}^\star) = \text{rank}(\tilde{\boldsymbol{B}})$. We let $\tilde{\boldsymbol{B}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T$ be its singular value decomposition where diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d_1 \times d_2}$ can be expressed as $\begin{pmatrix}\boldsymbol{\Lambda}_{11} & 0 \\ 0 & 0\end{pmatrix}$ for $\boldsymbol{\Lambda}_{11} \in \mathbb{R}^{r\times r}$. We define

$$\boldsymbol{T} = \boldsymbol{U}^T\boldsymbol{\Theta}\boldsymbol{V} = \boldsymbol{T}^{(1)} + \boldsymbol{T}^{(2)}$$

where $\boldsymbol{T}^{(1)} = \begin{pmatrix}0 & 0 \\ 0 & \boldsymbol{T}_{22}\end{pmatrix}$ and $\boldsymbol{T}^{(2)} = \begin{pmatrix}\boldsymbol{T}_{11} & \boldsymbol{T}_{12} \\ \boldsymbol{T}_{21} & 0\end{pmatrix}$ have the same corresponding block size as $\boldsymbol{\Lambda}$. Then we get

$$\begin{aligned}\|\hat{\boldsymbol{B}}\|_* =& \|\tilde{\boldsymbol{B}} + \boldsymbol{\Theta}\|_* = \|\boldsymbol{U}(\boldsymbol{\Lambda} + \boldsymbol{T})\boldsymbol{V}^T\|_* = \|\boldsymbol{\Lambda} + \boldsymbol{T}\|_* \\ \geqslant& \|\boldsymbol{\Lambda} + \boldsymbol{T}^{(1)}\|_* - \|\boldsymbol{T}^{(2)}\|_* = \|\tilde{\boldsymbol{B}}\|_* + \|\boldsymbol{T}^{(1)}\|_* - \|\boldsymbol{T}^{(2)}\|_*.\end{aligned} \tag{C.18}$$

The last equality is because of the block diagonal structure of $\boldsymbol{\Lambda}$ and $\boldsymbol{T}^{(1)}$ and $\|\tilde{\boldsymbol{B}}\|_* = \|\boldsymbol{\Lambda}\|_*$. Combine (C.18) and (C.17), we have

$$\|\boldsymbol{\Theta}\|_F^2 \leqslant \frac{3\lambda}{2}\|\boldsymbol{T}^{(2)}\|_* - \frac{\lambda}{2}\|\boldsymbol{T}^{(1)}\|_*. \tag{C.19}$$

According to (C.19), we have following cone condition

$$\|\boldsymbol{T}^{(1)}\|_* \leqslant 3\|\boldsymbol{T}^{(2)}\|_*. \tag{C.20}$$

Also form (C.19) and using Assumption 18, we get

$$\|\boldsymbol{\Theta}\|_F^2 \leqslant \frac{3\lambda}{2}\|\boldsymbol{T}^{(2)}\|_* \leqslant \frac{3\lambda}{2}\sqrt{\operatorname{rank}(\boldsymbol{T}^{(2)})}\|\boldsymbol{T}^{(2)}\|_F \leqslant 3\lambda\sqrt{r}\|\boldsymbol{T}^{(2)}\|_F \leqslant 3\lambda\sqrt{r}\|\boldsymbol{\Theta}\|_F.$$

Combining with (C.20), we know with probability at least $1 - 2/(d_1+d_2)^2 - \mathcal{P}(n, d_2)$,

$$\begin{aligned}\|\boldsymbol{\Theta}\|_F \leqslant& 3\sqrt{r}\lambda,\\ \|\boldsymbol{\Theta}\|_* \leqslant& \|\boldsymbol{T}^{(1)}\|_* + \|\boldsymbol{T}^{(2)}\|_* \leqslant 4\|\boldsymbol{T}^{(2)}\|_* \leqslant 24r\lambda.\end{aligned}$$

This concludes the theorem.

C.5. Proof of Theorem 13

Let's denote the loss function without penalty defined in (16) by $\hat{L}(\boldsymbol{B})$ and let $\mathcal{I}_4 = \hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^\star$. We know

$$\begin{aligned}\nabla\hat{L}(\tilde{\boldsymbol{B}}) =& 2\tilde{\boldsymbol{B}} - \frac{2}{n}\sum_{i=1}^n \breve{y}_i \cdot \widetilde{S(\boldsymbol{X}_i)}\breve{\boldsymbol{Z}}_i^T\hat{\boldsymbol{\Omega}} \overset{(9)}{=} 2\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T]\boldsymbol{\Omega}^\star - \frac{2}{n}\sum_{i=1}^n \breve{y}_i \cdot \widetilde{S(\boldsymbol{X}_i)}\breve{\boldsymbol{Z}}_i^T\hat{\boldsymbol{\Omega}}\\ =& 2\left(\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T](\boldsymbol{\Omega}^\star - \hat{\boldsymbol{\Omega}}) + (\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T] - \frac{2}{n}\sum_{i=1}^n \breve{y}_i \cdot \widetilde{S(\boldsymbol{X}_i)}\breve{\boldsymbol{Z}}_i^T)\hat{\boldsymbol{\Omega}}\right).\end{aligned} \tag{C.21}$$

From (C.21), we have

$$\|\nabla\hat{L}(\tilde{\boldsymbol{B}})\|_{\max} \leqslant 2\|\mathcal{I}_7\|_{\max}\|\mathcal{I}_4\|_1 + \underbrace{2\|\mathcal{I}_7\|_{\max}\|\boldsymbol{\Omega}^\star\|_1 + 2\|\mathcal{I}_4\|_{\max}\|\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T]\|_\infty}_{\text{dominant term}}.$$

Note that

$$\|\mathbb{E}[y\cdot S(\boldsymbol{x})\boldsymbol{z}^T]\|_\infty = \|\tilde{\boldsymbol{B}}\boldsymbol{\Sigma}^\star\|_\infty \leqslant \max_{j\in[d_2]}|\mu_j|\cdot\|\boldsymbol{B}^\star\boldsymbol{\Sigma}^\star\|_\infty. \tag{C.22}$$

Combine (C.22) with (B.22) and drop off the intersection term (smaller order), we know

$$\begin{aligned}P\left(\|\nabla\hat{L}(\tilde{\boldsymbol{B}})\|_{\max} > 38\|\boldsymbol{\Omega}^\star\|_1\sqrt{\frac{M\log d_1 d_2}{n}} + 2\max_{j\in[d_2]}|\mu_j|\cdot\|\boldsymbol{B}^\star\boldsymbol{\Sigma}^\star\|_\infty\tilde{\mathcal{H}}(n,d_2)\right)\\ \leqslant \frac{2}{d_1^2 d_2^2} + \tilde{\mathcal{P}}(n,d_2).\end{aligned}$$

So under the setup in theorem we have $\|\nabla\hat{L}(\tilde{\boldsymbol{B}})\|_{\max} \leqslant \lambda/2$. On the other side, based on definition of $\hat{\boldsymbol{B}}$ in (16), we have following basic inequality

$$\hat{L}(\hat{\boldsymbol{B}}) + \lambda\|\hat{\boldsymbol{B}}\|_{1,1} \leqslant \hat{L}(\tilde{\boldsymbol{B}}) + \lambda\|\tilde{\boldsymbol{B}}\|_{1,1}. \tag{C.23}$$

Define $\boldsymbol{\Theta} = \hat{\boldsymbol{B}} - \tilde{\boldsymbol{B}}$, same with (C.16) we have

$$\hat{L}(\hat{\boldsymbol{B}}) - \hat{L}(\tilde{\boldsymbol{B}}) = \langle\nabla\hat{L}(\tilde{\boldsymbol{B}}), \boldsymbol{\Theta}\rangle + \|\boldsymbol{\Theta}\|_F^2. \tag{C.24}$$

Combine (C.23) and (C.24), and define $S = \operatorname{supp}(\tilde{\boldsymbol{B}})$, we have

$$\begin{aligned}\|\boldsymbol{\Theta}\|_F^2 \leqslant& -\langle\nabla\hat{L}(\tilde{\boldsymbol{B}}), \boldsymbol{\Theta}\rangle + \lambda\|\tilde{\boldsymbol{B}}\|_{1,1} - \lambda\|\hat{\boldsymbol{B}}\|_{1,1}\\ \leqslant& \|\nabla\hat{L}(\tilde{\boldsymbol{B}})\|_{\max}\|\boldsymbol{\Theta}\|_{1,1} + \lambda\|\tilde{\boldsymbol{B}}_S\|_{1,1} - \lambda\|\hat{\boldsymbol{B}}_S\|_{1,1} - \lambda\|\hat{\boldsymbol{B}}_{S^C}\|_{1,1}\\ \leqslant& \|\nabla\hat{L}(\tilde{\boldsymbol{B}})\|_{\max}\|\boldsymbol{\Theta}\|_{1,1} + \lambda\|\boldsymbol{\Theta}_S\|_{1,1} - \lambda\|\boldsymbol{\Theta}_{S^C}\|_{1,1}.\end{aligned} \tag{C.25}$$

So, based on (C.25), we know, with probability at least $1 - 2/d_1d_2 - \tilde{\mathcal{P}}(n,d_2)$,

$$\|\boldsymbol{\Theta}\|_F^2 \leqslant \frac{3\lambda}{2}\|\boldsymbol{\Theta}_S\|_{1,1} \leqslant \frac{3\lambda}{2}\sqrt{sd_2}\|\boldsymbol{\Theta}_S\|_F \Longrightarrow \|\boldsymbol{\Theta}\|_F \leqslant 2\sqrt{sd_2}\lambda. \tag{C.26}$$

Similarly, we have

$$\|\boldsymbol{\Theta}\|_{1,1} \leqslant 4\|\boldsymbol{\Theta}_S\|_{1,1} \leqslant 8sd_2\lambda.$$

This concludes the proof.

Appendix D. Additional simulation results and plots

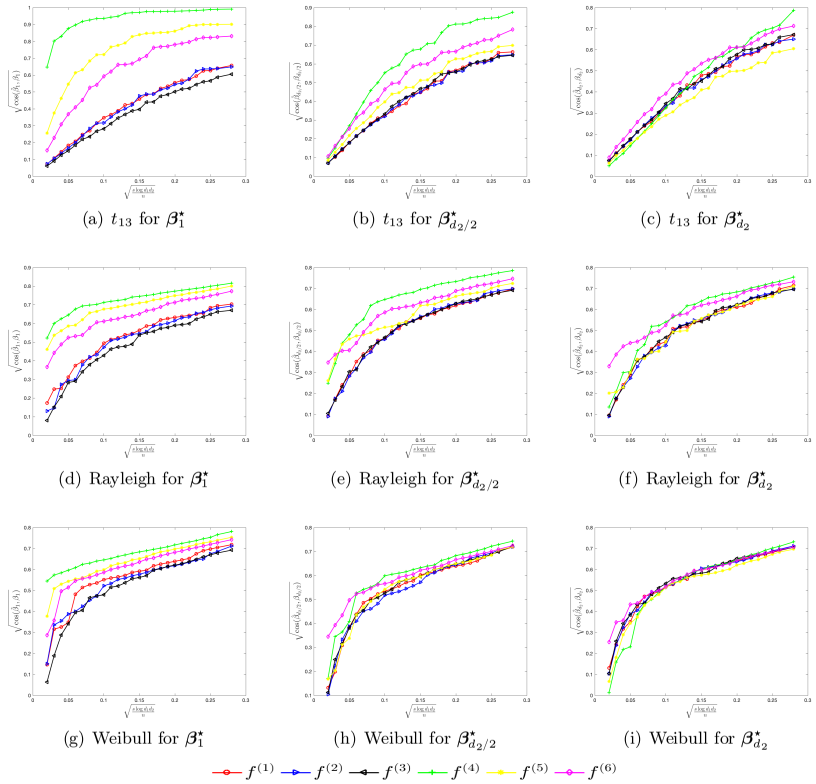
We present additional simulations results for the setting discussed in Section 7.1 and 7.4. Figure 9 illustrates the error trend for estimating $\boldsymbol{\beta}_1^\star$, $\boldsymbol{\beta}_{d_2/2}^\star$ and $\boldsymbol{\beta}_{d_2}^\star$ under different link functions and $\boldsymbol{x}$ following one of the following distributions: t_{13}, Rayleigh or Weibull. Figure 10–12 plot the error trend for all d_2 parameters. From the plots we observe that the linear trend will be more and more obvious when k varies from 1 to d_2, especially for quadratic link functions. Note also that the number of samples needed to get comparable results decrease dramatically as k increases. Figure 13 shows 95% confidence interval for the signal trajectory of the real dataset we analyzed in Section 7.4, which is obtained by doing parametric bootstrapping.

References

D. Babichev and F. Bach. Slice inverse regression with score functions. *Electron. J. Stat.*, 12(1):1507–1543, 2018.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities*. Oxford University Press, Oxford, 2013. A nonasymptotic theory of independence, With a foreword by Michel Ledoux.

P. Bühlmann and S. A. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011. Methods, theory and applications.

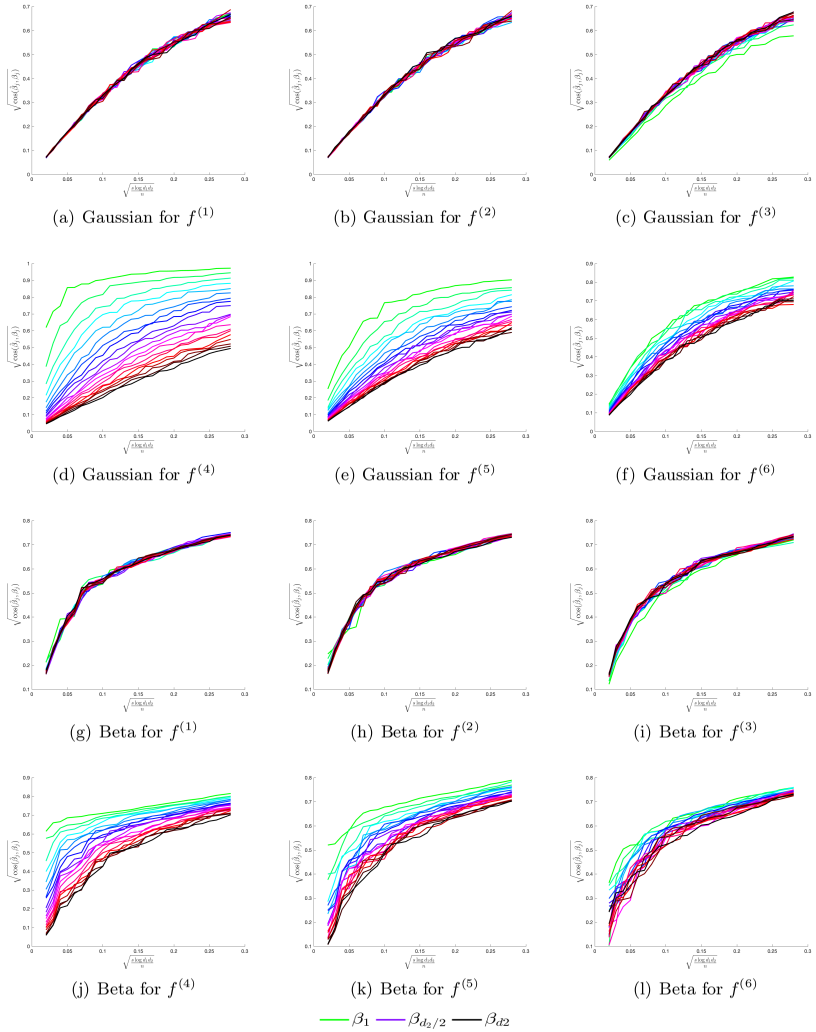


(a) t_{13} for $\boldsymbol{\beta}_1^\star$ (b) t_{13} for $\boldsymbol{\beta}_{d_2/2}^\star$ (c) t_{13} for $\boldsymbol{\beta}_{d_2}^\star$

(d) Rayleigh for $\boldsymbol{\beta}_1^\star$ (e) Rayleigh for $\boldsymbol{\beta}_{d_2/2}^\star$ (f) Rayleigh for $\boldsymbol{\beta}_{d_2}^\star$

(g) Weibull for $\boldsymbol{\beta}_1^\star$ (h) Weibull for $\boldsymbol{\beta}_{d_2/2}^\star$ (i) Weibull for $\boldsymbol{\beta}_{d_2}^\star$

Figure 9: Sparse vector estimation plot (II). This figure shows cosine distance trend for error of estimating single sparse parameter in model (1). Six lines indicates six different types of link functions. Each row represents one type of distribution for design. We only plot the error trend for the first, the middle, the last parameter. All above simulation results are consistent with Theorem 7.



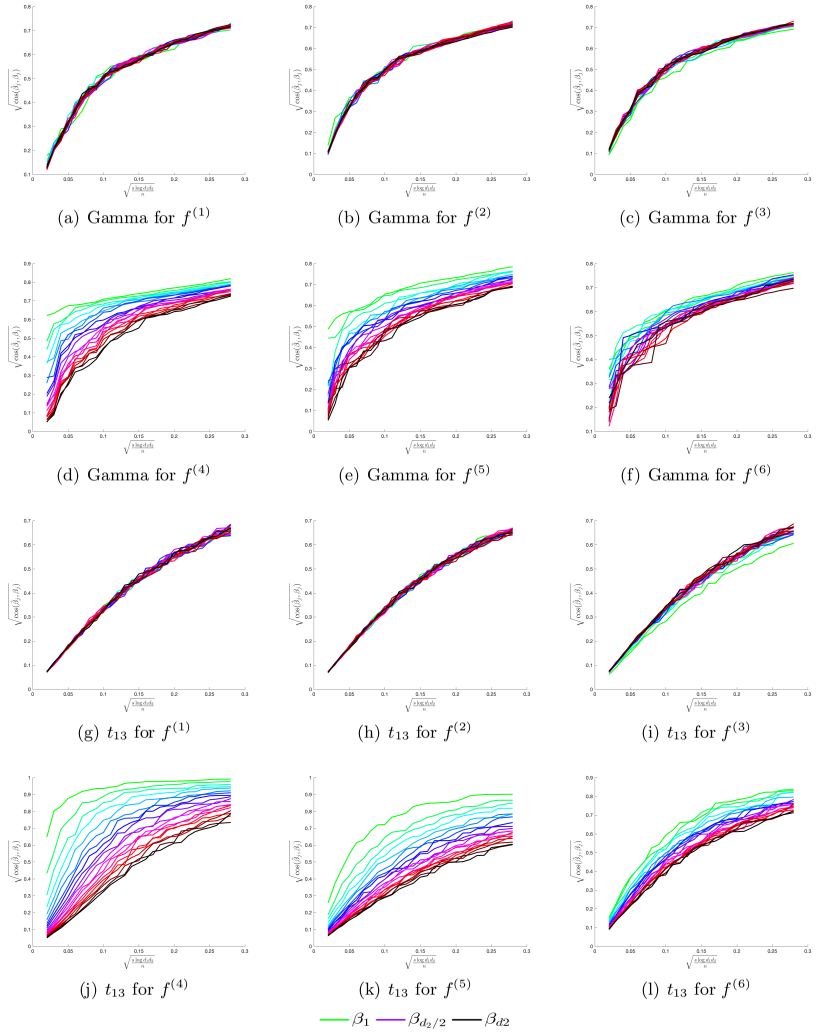
(a) Gaussian for $f^{(1)}$ (b) Gaussian for $f^{(2)}$ (c) Gaussian for $f^{(3)}$

(d) Gaussian for $f^{(4)}$ (e) Gaussian for $f^{(5)}$ (f) Gaussian for $f^{(6)}$

(g) Beta for $f^{(1)}$ (h) Beta for $f^{(2)}$ (i) Beta for $f^{(3)}$

(j) Beta for $f^{(4)}$ (k) Beta for $f^{(5)}$ (l) Beta for $f^{(6)}$

Figure 10: Sparse vector estimation for Gaussian and Beta design. This figure shows cosine distance trend for error of estimating single sparse parameter in model (1). The color varies from green to black when k varies from 1 to d_2. The first and third row correspond to the linear link function, while the second and fourth row correspond to the quadratic link function.



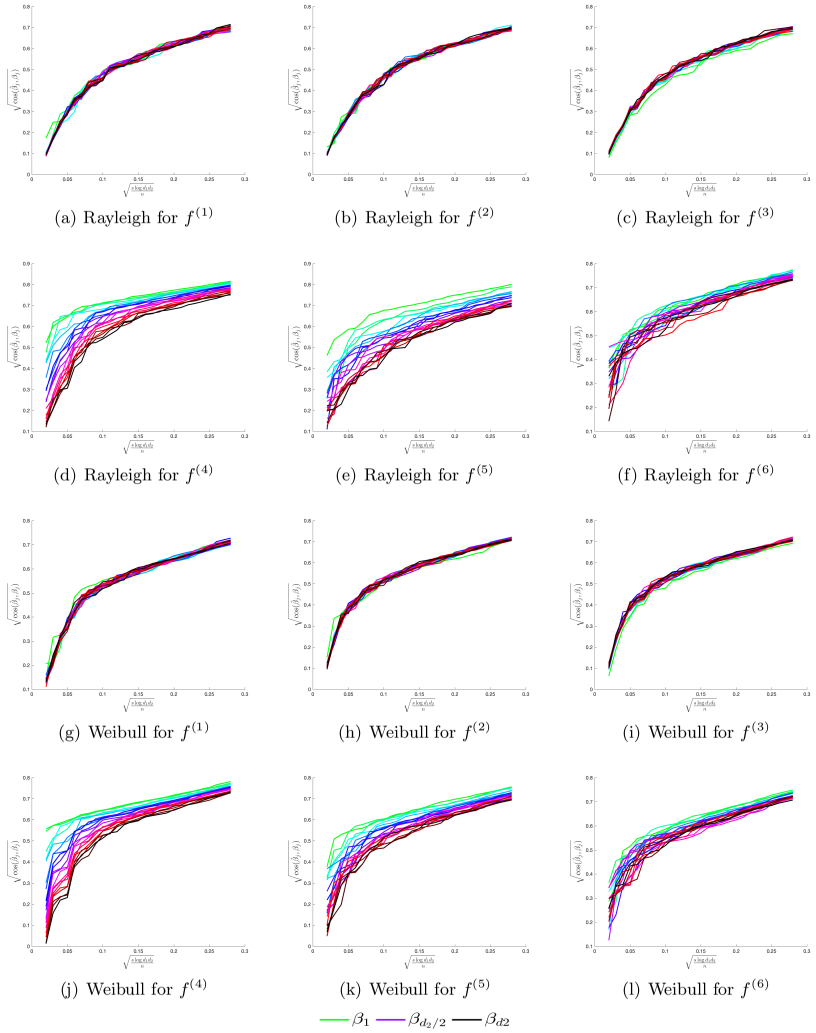
(a) Gamma for $f^{(1)}$ (b) Gamma for $f^{(2)}$ (c) Gamma for $f^{(3)}$

(d) Gamma for $f^{(4)}$ (e) Gamma for $f^{(5)}$ (f) Gamma for $f^{(6)}$

(g) t_{13} for $f^{(1)}$ (h) t_{13} for $f^{(2)}$ (i) t_{13} for $f^{(3)}$

(j) t_{13} for $f^{(4)}$ (k) t_{13} for $f^{(5)}$ (l) t_{13} for $f^{(6)}$

Figure 11: Sparse vector estimation for Gamma and t design. This figure shows cosine distance trend for error of estimating single sparse parameter in model (1). The color varies from green to black when k varies from 1 to d_2. The first and third row correspond to the linear link function, while the second and fourth row correspond to the quadratic link function.



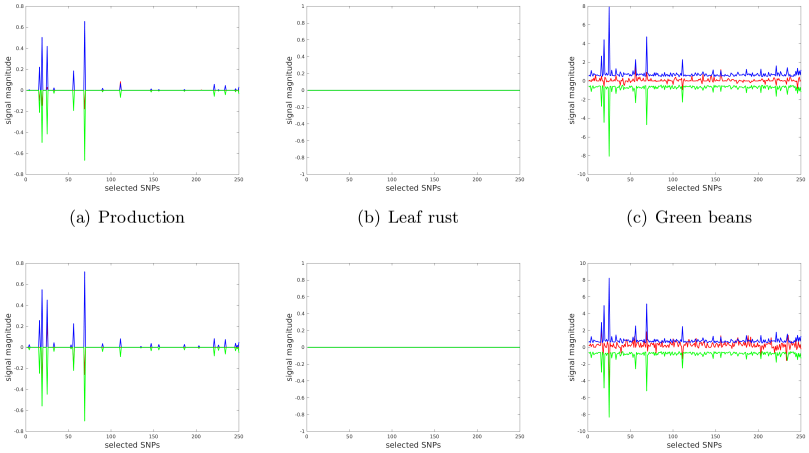
(a) Rayleigh for $f^{(1)}$ (b) Rayleigh for $f^{(2)}$ (c) Rayleigh for $f^{(3)}$

(d) Rayleigh for $f^{(4)}$ (e) Rayleigh for $f^{(5)}$ (f) Rayleigh for $f^{(6)}$

(g) Weibull for $f^{(1)}$ (h) Weibull for $f^{(2)}$ (i) Weibull for $f^{(3)}$

(j) Weibull for $f^{(4)}$ (k) Weibull for $f^{(5)}$ (l) Weibull for $f^{(6)}$

Figure 12: Sparse vector estimation for Rayleigh and Weibull design. This figure shows cosine distance trend for error of estimating single sparse parameter in model (1). The color varies from green to black when k varies from 1 to d_2. The first and third row correspond to the linear link function, while the second and fourth row correspond to the quadratic link function.



(a) Production (b) Leaf rust (c) Green beans

(d) Production (e) Leaf rust (f) Green beans

Figure 13: Signal trajectories for 95% confidence interval (parametric bootstrapping). The (l, i) plot shows the confidence interval trajectory of $(\boldsymbol{\beta}^{l,i})^\star$ where $l = 1, 2$ indexes the confounder and $i = 1, 2, 3$ indexes the response. The blue line indicates the upper bound, the red line indicates the estimator, and the green line indicates the lower bound.

T. T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *J. Am. Stat. Assoc.*, 106(494):594–607, 2011.

T. T. Cai, W. Liu, and H. H. Zhou. Estimating sparse precision matrix: optimal rates of convergence and adaptive estimation. *Ann. Statist.*, 44(2):455–488, 2016.

R. J. Carroll, J. Fan, I. Gijbels, and M. P. Wand. Generalized partially linear single-index models. *J. Amer. Statist. Assoc.*, 92(438):477–489, 1997.

O. Catoni. Challenging the empirical mean and empirical variance: a deviation study. *Ann. Inst. Henri Poincaré Probab. Stat.*, 48(4):1148–1185, 2012.

H. Chen. Estimation of a projection-pursuit type regression model. *Ann. Statist.*, 19(1): 142–157, 1991.

L. H. Y. Chen, L. Goldstein, and Q.-M. Shao. *Normal approximation by Stein's method*. Probability and its Applications (New York). Springer, Heidelberg, 2011.

Y. Chen and R. J. Samworth. Generalized additive and index models with shape constraints. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 78(4):729–754, 2016.

K. Chwialkowski, H. Strathmann, and A. Gretton. A kernel test of goodness of fit. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2606–2615, New York, New York, USA, 2016. PMLR.

W. S. Cleveland, E. Grosse, and W. M. Shyu. Local regression models. In J. M. Chambers and T. J. Hastie, editors, *Statistical Models in S*, pages 309–376, 1991.

J. Fan and W. Zhang. Statistical methods with varying coefficient models. *Statistics and its Interface*, 1(1):179–195, 2008.

J. Fan, Q. Yao, and Z. Cai. Adaptive varying-coefficient linear models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 65(1):57–80, 2003.

L. F. V. Ferrão, R. G. Ferrão, M. A. G. Ferrão, A. Fonseca, P. Carbonetto, M. Stephens, and A. A. F. Garcia. Accurate genomic prediction of coffea canephora in multiple environments using whole-genome statistical models. *Heredity*, pages 1–15, 2018.

L. Goldstein, S. Minsker, and X. Wei. Structured signal recovery from non-linear and heavy-tailed measurements. *IEEE Trans. Inform. Theory*, 64(8):5513–5530, 2018.

M. C. Grant and S. P. Boyd. cvx: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`, 2012.

M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

C. Guo, H. Yang, and J. Lv. Generalized varying index coefficient models. *J. Comput. Appl. Math.*, 300:1–17, 2016.

T. J. Hastie and R. J. Tibshirani. Varying-coefficient models. *J. R. Stat. Soc. B*, 55(4): 757–796, 1993.

Z. Huang and R. Zhang. Profile empirical-likelihood inferences for the single-index-coefficient regression model. *Stat. Comput.*, 23(4):455–465, 2013.

B. Jiang and J. S. Liu. Variable selection for general index models via sliced inverse regression. *Ann. Statist.*, 42(5):1751–1786, 2014.

K.-C. Li. Sliced inverse regression for dimension reduction. *J. Amer. Statist. Assoc.*, 86 (414):316–342, 1991. With discussion and a rejoinder by the author.

Q. Lin, X. Li, D. Huang, and J. S. Liu. On the optimality of sliced inverse regression in high dimensions. *ArXiv e-prints: 1701.06009*, 2017.

Q. Lin, Z. Zhao, and J. S. Liu. On consistency and sparsity for sliced inverse regression in high dimensions. *Ann. Statist.*, 46(2):580–610, 2018.

H. Liu, Y. Feng, Y. Mao, D. Zhou, J. Peng, and Q. Liu. Action-dependent control variates for policy optimization via stein identity. In *International Conference on Learning Representations*, 2018.

Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2378–2386. Curran Associates, Inc., 2016.

Q. Liu, J. Lee, and M. Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 276–284, New York, New York, USA, 2016. PMLR.

S. Ma and P. X.-K. Song. Varying index coefficient models. *J. Amer. Statist. Assoc.*, 110 (509):341–356, 2015.

S. Minsker. Sub-Gaussian estimators of the mean of a random matrix with heavy-tailed entries. *Ann. Statist.*, 46(6A):2871–2903, 2018.

S. Na and M. Kolar. High-dimensional index volatility models via stein's identity. *arXiv preprint arXiv:1811.10790*, 2018.

M. Neykov, J. S. Liu, and T. Cai. L1-regularized least squares for support recovery of high dimensional single index models with gaussian designs. *Journal of Machine Learning Research*, 17(87):1–37, 2016.

Y. Plan, R. Vershynin, and E. Yudovina. High-dimensional estimation with geometric constraints. *Inf. Inference*, 6(1):1–40, 2017.

Y. Plan and R. Vershynin. The generalized Lasso with non-linear observations. *IEEE Trans. Inform. Theory*, 62(3):1528–1537, 2016.

A. Rohde and A. B. Tsybakov. Estimation of high-dimensional low-rank matrices. *Ann. Statist.*, 39(2):887–930, 2011.

H. Sedghi, M. Janzamin, and A. Anandkumar. Provable tensor methods for learning mixtures of generalized linear models. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1223–1231, Cadiz, Spain, 2016. PMLR.

C. Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. pages 583–602, 1972.

C. Stein, P. Diaconis, S. Holmes, and G. Reinert. Use of exchangeable pairs in the analysis of simulations. In *Stein's method: expository lectures and applications*, volume 46 of *IMS Lecture Notes Monogr. Ser.*, pages 1–26. Inst. Math. Statist., Beachwood, OH, 2004.

G. W. Stewart and J. G. Sun. *Matrix Perturbation Theory*. Computer Science and Scientific Computing. Academic Press Inc., Boston, MA, 1990.

T. Sun and C.-H. Zhang. Sparse matrix inversion with scaled lasso. *J. Mach. Learn. Res.*, 14:3385–3418, 2013.

C. Thrampoulidis, E. Abbasi, and B. Hassibi. Lasso with non-linear measurements is equivalent to one with linear measurements. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3420–3428. Curran Associates, Inc., 2015.

R. J. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B*, 58 (1):267–288, 1996.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed sensing*, pages 210–268. Cambridge Univ. Press, Cambridge, 2012.

T. Wang, J. Zhang, H. Liang, and L. Zhu. Estimation of a groupwise additive multiple-index model and its applications. *Statist. Sinica*, 25(2):551–566, 2015.

Y. Xia and W. K. Li. On single-index coefficient regression models. *J. Amer. Statist. Assoc.*, 94(448):1275–1285, 1999.

L. Xue and Q. Wang. Empirical likelihood for single-index varying-coefficient models. *Bernoulli*, 18(3):836–856, 2012.

Z. Yang, K. Balasubramanian, and H. Liu. High-dimensional non-Gaussian single index models via thresholded score function estimation. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3851–3860, International Convention Centre, Sydney, Australia, 2017. PMLR.

Z. Yang, L. F. Yang, E. X. Fang, T. Zhao, Z. Wang, and M. Neykov. Misspecified nonconvex statistical optimization for sparse phase retrieval. *Math. Program.*, 176(1-2, Ser. B):545–571, 2019.

M. Yuan. On the identifiability of additive index models. *Statist. Sinica*, 21(4):1901–1911, 2011.

J. Zhang, X. Chen, and W. Zhou. High dimensional elliptical sliced inverse regression in non-gaussian distributions. *ArXiv e-prints 1801.01950*, 2017.

W. Zhao, H. Lian, and H. Liang. Quantile regression for the single-index coefficient model. *Bernoulli*, 23(3):1997–2027, 2017.

L. Zhu, B. Miao, and H. Peng. On sliced inverse regression with high-dimensional covariates. *J. Amer. Statist. Assoc.*, 101(474):630–643, 2006.

Journal of Machine Learning Research 20 (2019) 1-33 Submitted 10/18; Revised 6/19; Published 8/19

Approximation algorithms for stochastic clustering

David G. Harris DAVIDGHARRIS29@GMAIL.COM
Department of Computer Science, University of Maryland
College Park, MD 20742

Shi Li SHIL@BUFFALO.EDU
University at Buffalo, Buffalo, NY

Thomas Pensyl TPENSYL@BANDWIDTH.COM
Bandwidth, Inc. Raleigh, NC

Aravind Srinivasan SRIN@CS.UMD.EDU
Department of Computer Science and Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742

Khoa Trinh KHOATRINH@GOOGLE.COM
Google, Mountain View, CA 94043

Editor: Maya Gupta

Abstract

We consider stochastic settings for clustering, and develop provably-good approximation algorithms for a number of these notions. These algorithms yield better approximation ratios compared to the usual deterministic clustering setting. Additionally, they offer a number of advantages including clustering which is fairer and has better long-term behavior for each user. In particular, they ensure that *every user* is guaranteed to get good service (on average). We also complement some of these with impossibility results.

Keywords: clustering, k-center, k-median, lottery, approximation algorithms

This is an extended version of a paper that appeared in the *Conference on Neural Information Processing Systems* (NeurIPS) 2018. Research supported in part by NSF Awards CNS-1010789, CCF-1422569, CCF-1749864, CCF-1566356, CCF-1717134, a gift from Google, Inc., and by research awards from Adobe, Inc. and Amazon, Inc.

1. Introduction

Clustering is a fundamental problem in machine learning and data science. A general clustering task is to partition the given datapoints such that points inside the same cluster are "similar" to each other. More formally, consider a set of datapoints $\mathcal{C}$ and a set of "potential cluster centers" $\mathcal{F}$, with a metric d on $\mathcal{C} \cup \mathcal{F}$. We define $n := |\mathcal{C} \cup \mathcal{F}|$. Given any set $\mathcal{S} \subseteq \mathcal{F}$, each $j \in \mathcal{C}$ is associated with the key statistic $d(j, \mathcal{S}) = \min_{i \in \mathcal{S}} d(i, j)$. The typical clustering task is to select a set $\mathcal{S} \subseteq \mathcal{F}$ which has a small size and which minimizes the values of $d(j, \mathcal{S})$.

©2019 David G. Harris, Shi Li, Thomas Pensyl, Aravind Srinivasan.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-716.html.

The size of the set $\mathcal{S}$ is often fixed to a value k, and we typically "boil down" the large collection of values $d(j, \mathcal{S})$ into a single overall objective function. A variety of objective functions and assumptions on sets $\mathcal{C}$ and $\mathcal{F}$ are used. The most popular problems include [1]

- the k-center problem: minimize the value $\max_{j \in \mathcal{C}} d(j, \mathcal{S})$ given that $\mathcal{F} = \mathcal{C}$.
- the k-supplier problem: minimize the value $\max_{j \in \mathcal{C}} d(j, \mathcal{S})$ (where $\mathcal{F}$ and $\mathcal{C}$ may be unrelated);
- the k-median problem: minimize the summed value $\sum_{j \in \mathcal{C}} d(j, \mathcal{S})$; and
- the k-means problem: minimize the summed square value $\sum_{j \in \mathcal{C}} d(j, \mathcal{S})^2$.

An important special case is when $\mathcal{C} = \mathcal{F}$ (e.g. the k-center problem); since this often occurs in the context of data clustering, we refer to this as the **self-contained clustering (SCC)** setting.

These classic NP-hard problems have been studied intensively for the past few decades. There is an alternative interpretation from the viewpoint of operations research: the sets $\mathcal{F}$ and $\mathcal{C}$ can be thought of as "facilities" and "clients", respectively. We say that $i \in \mathcal{F}$ is *open* if i is placed into the solution set $\mathcal{S}$. For a set $\mathcal{S} \subseteq \mathcal{F}$ of open facilities, $d(j, \mathcal{S})$ can then be interpreted as the connection cost of client j. This terminology has historically been used for clustering problems, and we adopt it throughout for consistency. However, our focus is on the case in which $\mathcal{C}$ and $\mathcal{F}$ are arbitrary abstract sets in the data-clustering setting.

Since these problems are NP-hard, much effort has been paid on algorithms with "small" provable *approximation ratios/guarantees*: i.e., polynomial-time algorithms that produce solutions of cost at most α times the optimal. The current best-known approximation ratio for k-median is 2.675 by Byrka et al. (2017) and it is NP-hard to approximate this problem to within a factor of $1 + 2/e \approx 1.735$ (Jain et al., 2002). The recent breakthrough by Ahmadian et al. (2017) gives a 6.357-approximation algorithm for k-means, improving on the previous approximation guarantee of $9 + \epsilon$ based on local search (Kanungo et al., 2004). Finally, the k-supplier problem is "easier" than both k-median and k-means in the sense that a simple 3-approximation algorithm (Hochbaum and Shmoys, 1986) is known, as is a 2-approximation for k-center problem: we cannot do better than these approximation ratios unless P = NP (Hochbaum and Shmoys, 1986).

While optimal approximation algorithms for the center-type problems are well-known, one can easily demonstrate instances where such algorithms return a worst-possible solution: (i) all clusters have the same worst-possible radius ($2T$ for k-center and $3T$ for k-supplier where T is the optimal radius) and (ii) almost all data points are on the circumference of the resulting clusters. Although it is NP-hard to improve these approximation ratios, our new randomized algorithms provide significantly better "per-point" guarantees. For example, we achieve a new "per-point" guarantee $\mathbf{E}[d(j, \mathcal{S})] \leq (1 + 2/e)T \approx 1.736T$, while respecting the usual guarantee $d(j, \mathcal{S}) \leq 3T$ with probability one. *Thus, while maintaining good global quality with probability one, we also provide superior stochastic guarantees for each user.*

1. In the original version of the k-means problem, $\mathcal{C}$ is a subset of $\mathbb{R}^\ell$ and $\mathcal{F} = \mathbb{R}^\ell$ and d is the Euclidean metric. By standard discretization techniques (see, e.g., Matoušek (2000); Feldman et al. (2007)), the set $\mathcal{F}$ can be reduced to a polynomially-bounded set with only a small loss in the value of the objective function.

The general problem we study in this paper is to develop approximation algorithms for center-type problems where $\mathcal{S}$ is drawn from a probability distribution over k-element subsets of $\mathcal{F}$; we refer to these as *k-lotteries.* We aim to construct a k-lottery Ω achieving certain guarantees on the distributional properties of $d(j, \mathcal{S})$. The classical k-center problem can be viewed as the special case where the distribution Ω is deterministic, that is, it is supported on a single point. Our goal is to find an *approximating distribution* $\tilde{\Omega}$ which matches the *target distribution* Ω as closely as possible for each client j.

Stochastic solutions can circumvent the approximation hardness of a number of classical center-type problems. There are a number of additional applications where stochasticity can be beneficial. We summarize three here: smoothing the integrality constraints of clustering, solving repeated problem instances, and achieving fair solutions.

Stochasticity as interpolation. In practice, *robustness* of the solution is often more important than achieving the absolute optimal value for the objective function. One potential problem with the (deterministic) center measure is that it can be highly non-robust. As an extreme example, consider k-center with k points, each at distance 1 from each other. This clearly has value 0 (choosing $\mathcal{S} = \mathcal{C}$). However, if a single new point at distance 1 to all other points is added, then the solution jumps to 1. Stochasticity alleviates this discontinuity: by choosing k facilities uniformly at random among the full set of $k+1$, we can ensure that $\mathbf{E}[d(j, \mathcal{S})] = \frac{1}{k+1}$ for every point j, a much smoother transition.

Repeated clustering problems. Consider clustering problems where the choice of $\mathcal{S}$ can be changed periodically: e.g., $\mathcal{S}$ could be the set of k locations in the cloud chosen by a service-provider. This set $\mathcal{S}$ can be shuffled periodically in a manner transparent to end-users. For any user $j \in \mathcal{C}$, the statistic $d(j, \mathcal{S})$ represents the latency of the service j receives (from its closest service-point in $\mathcal{S}$). If we aim for a fair or minmax service allocation, then our k-center stochastic approximation results ensure that, with high probability, *every client j* gets long-term average service of at most around $1.736T$. The average here is taken over the periodic re-provisioning of $\mathcal{S}$. (Furthermore, we have the risk-avoidance guarantee that in no individual provisioning of $\mathcal{S}$ will any client have service greater than $3T$.)

Fairness in clustering. The classical clustering problems combine the needs of many different points (elements of $\mathcal{C}$) into one metric. However, clustering (and indeed many other ML problems) are increasingly driven by inputs from parties with diverse interests. Fairness in these contexts has taken on greater importance in the current environment where decisions are increasingly made by algorithms and machine learning. Some examples of recent concerns include the accusations of, and fixes for, possible racial bias in Airbnb rentals (Badger, 2016) and the finding that setting the gender to "female" in Google's *Ad Settings* resulted in getting fewer ads for high-paying jobs (Datta et al., 2015). Starting with older work such as Schulman et al. (1999), there have been highly-publicized works on bias in allocating scarce resources – e.g., racial discrimination in hiring applicants who have very similar resumés (Bertrand and Mullainathan, 2004). Additional work discusses the possibility of bias in electronic marketplaces, whether human-mediated or not (Ayres et al., 2015; Badger, 2016).

A fair allocation should provide good service guarantees *to each user individually.* In data clustering settings where a user corresponds to a datapoint, this means that every point $j \in \mathcal{C}$ should be guaranteed a good value of $d(j, \mathcal{S})$. This is essentially the goal of

k-center type problems, but the stochastic setting broadens the meaning of good per-user service.

Consider the following scenarios. Each user, either explicitly or implicitly, submits their data (corresponding to a point in $\mathcal{C}$) to an aggregator such as an e-commerce site. A small number k of users are then chosen as "influencer" nodes; for instance, the aggregator may give them a free product sample to influence the whole population in aggregate, as in Kempe et al. (2015), or the aggregator may use them as a sparse "sketch", so that each user gets relevant recommendations from a influencer which is similar to them. Each point j would like to be in a cluster that is "high quality" *from its perspective*, with $d(j, \mathcal{S})$ being a good proxy for such quality. Indeed, there is increasing emphasis on the fact that organizations monetize their user data, and that users need to be compensated for this (see, e.g., Lanier (2014); Ibarra et al. (2018)). This is a transition from viewing data as capital to viewing *data as labor.* A concrete way for users (i.e., the data points $j \in \mathcal{C}$) to be compensated in our context is for each user to get a guarantee on their solution quality: i.e., bounds on $d(j, \mathcal{S})$.

1.1. Our contributions and overview

In Section 2, we encounter the first clustering problem which we refer to as *chance k-coverage*: namely, every client j has a distance demand r_j and probability demand p_j, and we wish to find a distribution satisfying $\Pr[d(j, \mathcal{S}) \leq r_j] \geq p_j$. We show how to obtain an approximation algorithm to find an approximating distribution $\tilde{\Omega}$ with[2]

$$\Pr_{\mathcal{S} \sim \tilde{\Omega}}[d(j, \mathcal{S}) \leq 9r_j] \geq p_j.$$

In a number of special cases, such as when all the values of p_j or r_j are the same, the distance factor 9 can be improved to 3, *which is optimal*; it is an interesting question to determine whether this factor can also be improved in the general case.

In Section 3, we consider a special case of chance k-coverage, in which $p_j = 1$ for all clients j. This is equivalent to the classical (deterministic) k-supplier problem. Allowing the approximating distribution $\tilde{\Omega}$ to be stochastic yields significantly better distance guarantees than are possible for k-supplier or k-center. For instance, we find an approximating distribution $\tilde{\Omega}$ with

$$\forall j \in \mathcal{C} \qquad \mathbf{E}_{\mathcal{S} \sim \tilde{\Omega}}[d(j, \mathcal{S})] \leq 1.592T \text{ and } \Pr[d(j, \mathcal{S}) \leq 3T] = 1$$

where T is the optimal solution to the (deterministic) k-center problem. By contrast, deterministic polynomial-time algorithms cannot guarantee $d(j, \mathcal{S}) < 2T$ for all j, unless $\mathrm{P} = \mathrm{NP}$ (Hochbaum and Shmoys, 1986).

In Section 4, we show a variety of lower bounds on the approximation factors achievable by efficient algorithms (assuming $\mathrm{P} \neq \mathrm{NP}$). For instance, we show that our approximation algorithm for chance k-coverage with equal p_j or r_j has the optimal distance approximation factor 3, that our approximation algorithm for k-supplier has optimal approximation factor $1 + 2/e$, and that the approximation factor 1.592 for k-center cannot be improved below $1 + 1/e$.

2. Notation such as "$S \sim \tilde{\Omega}$" indicates that the random set S is drawn from the distribution $\tilde{\Omega}$.

In Section 5, we consider a different type of stochastic approximation problem based on expected distances: namely, every client has a demand t_j, and we seek a k-lottery Ω with $\mathbf{E}[d(j,\mathcal{S})] \leq t_j$. We show that we can leverage *any given* α-approximation algorithm for k-median to produce a k-lottery $\tilde{\Omega}$ with $\mathbf{E}[d(j,\mathcal{S})] \leq \alpha t_j$. (Recall that the current-best α here is 2.675 as shown in Byrka et al. (2017).)

In Section 6, we consider the converse problem to Section 3: if we are given a k-lottery Ω with $\mathbf{E}[d(j,\mathcal{S})] \leq t_j$, can we produce a single deterministic set $\mathcal{S}$ so that $d(j,\mathcal{S}) \approx t_j$ and $|\mathcal{S}| \approx k$? We refer to this as a *determinization* of Ω. We show a variety of determinization algorithms. For instance, we are able to find a set $\mathcal{S}$ with $|\mathcal{S}| \leq 3k$ and $d(j,\mathcal{S}) \leq 3t_j$. We also show a number of nearly-matching lower bounds.

1.2. Related Work

With algorithms increasingly running our world, there has been substantial recent interest on incorporating fairness systematically into algorithms and machine learning. One important notion is *disparate impact*: in addition to requiring that *protected attributes* such as gender or race not be used (explicitly) in decisions, this asks that decisions not be disproportionately different for diverse protected classes (Feldman et al., 2015). This is developed further in the context of clustering in the work of Chierichetti et al. (2017). Such notions of *group fairness* are considered along with *individual fairness* – treating similar individuals similarly – in Zemel et al. (2013). See Dwork et al. (2012) for earlier work that developed foundations and connections for several such notions of fairness.

In the context of the location and sizing of services, there have been several studies indicating that proactive on-site provision of healthcare improves health outcomes significantly: e.g., mobile mammography for older women (Reuben et al., 2002), mobile healthcare for reproductive health in immigrant women (Guruge et al., 2010), and the use of a community mobile-health van for increased access to prenatal care (Edgerley et al., 2007). Studies also indicate the impact of distance to the closest facility on health outcomes: see, e.g., McCarthy et al. (2007); Mooney et al. (2000); Schmitt et al. (2003). Such works naturally suggest tradeoffs between resource allocation (provision of such services, including sizing – e.g., the number k of centers) and expected health outcomes.

While much analysis for facility-location problems has focused on the static case, other works have examined a similar lottery model for center-type problems. Harris et al. (2019, 2017) analyzed models similar to chance k-coverage and minimization of $\mathbf{E}[d(j,\mathcal{S})]$, but applied to knapsack center and matroid center problems; they also considered robust versions (in which a small subset of clients may be denied service). While the overall model was similar to the ones we explore here, the techniques are somewhat different. Furthermore, these works focus only on the case where the target distribution is itself deterministic.

Similar stochastic approximation guarantees have appeared in the context of approximation algorithms for static problems, particularly k-median problems. Charikar and Li (2012) discussed a randomized procedure for converting a linear-programming relaxation in which a client has *fractional* distance t_j, into a distribution Ω satisfying $\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})] \leq 3.25t_j$. This property can be used, among other things, to achieve a 3.25-approximation for k-median. However, many other randomized rounding algorithms for k-median only seek to preserve the *aggregate* value $\sum_j \mathbf{E}[d(j,\mathcal{S})]$, without our type of per-point guarantee.

We also contrast our approach with a different stochastic k-center problem considered in works such as Huang and Li (2017); Alipour and Jafari (2018). These consider a model with a fixed, deterministic set $\mathcal{S}$ of open facilities, while the client set is determined stochastically; this model is almost precisely opposite to ours.

1.3. Publicly Verifying the Distributions

Our approximation algorithms will have the following structure: given some target distribution Ω, we construct a randomized procedure $\mathcal{A}$ which returns some random set $\mathcal{S}$ with good probabilistic guarantees matching Ω. Thus the algorithm $\mathcal{A}$ is *itself* the approximating distribution $\tilde{\Omega}$.

In a number of cases, we can convert the randomized algorithm $\mathcal{A}$ into a distribution $\tilde{\Omega}$ which has a sparse support (set of points to which it assigns nonzero probability), and which can be enumerated directly. This may cause a small loss in approximation ratio. The distribution $\tilde{\Omega}$ can be publicly verified, and the users can then draw from $\tilde{\Omega}$ as desired.

Recall that one of our main motivations is fairness in clustering; the ability for the users to verify that they are being treated fairly in a stochastic sense (although not necessarily in any one particular run of the algorithm) is particularly important.

1.4. Notation

We define $\binom{\mathcal{F}}{k}$ to be the collection of k-element subsets of $\mathcal{F}$. We assume throughout that $\mathcal{F}$ can be made arbitrarily large by duplicating its elements; thus, whenever we have an expression like $\binom{\mathcal{F}}{k}$, we assume without loss of generality that $|\mathcal{F}| \geq k$.

We will let $[t]$ denote the set $\{1, 2, \ldots, t\}$. For any vector $a = (a_1, \ldots, a_t)$ and a subset $X \subseteq [t]$, we write $a(X)$ as shorthand for $\sum_{i\in X} a_i$.

We use the Iverson notation throughout, so that for any Boolean predicate $\mathcal{P}$ we let $[[\mathcal{P}]]$ be equal to one if $\mathcal{P}$ is true and zero otherwise.

For a real number $q \in [0,1]$, we define $\overline{q} = 1 - q$.

Given any $j \in \mathcal{C}$ and any real number $r \geq 0$, we define the ball $B(j,r) = \{i \in \mathcal{F} \mid d(i,j) \leq r\}$. We let $\theta(j)$ be the distance from j to the nearest facility, and V_j be the facility closest to j, i.e. $d(j,V_j) = d(j,\mathcal{F}) = \theta(j)$. Note that in the SCC setting we have $V_j = j$ and $\theta(j) = 0$.

For a solution set $\mathcal{S}$, we say that $j \in \mathcal{C}$ is *matched* to $i \in \mathcal{S}$ if i is the closest facility of $\mathcal{S}$ to j; if there are multiple closest facilities, we take i to be one with least index.

1.5. Some Useful Subroutines

We will use two basic subroutines repeatedly: *dependent rounding* and *greedy clustering*.

In dependent rounding, we aim to preserve certain marginal distributions and negative correlation properties while satisfying some constraints with probability one. Our algorithms use a dependent-rounding algorithm from Srinivasan (2001), which we summarize as follows:

Proposition 1 *There exists a randomized polynomial-time algorithm* DepRound(y) *which takes as input a vector* $y \in [0,1]^n$*, and outputs a random set* $Y \subseteq [n]$ *with the following properties:*

(P1) $\Pr[i \in Y] = y_i$, *for all* $i \in [n]$,

(P2) $\lfloor \sum_{i=1}^n y_i \rfloor \le |Y| \le \lceil \sum_{i=1}^n y_i \rceil$ *with probability one,*

(P3) $\Pr[Y \cap S = \emptyset] \le \prod_{i \in S}(1 - y_i)$ *for all* $S \subseteq [n]$.

We adopt the following additional convention: suppose $(y_1, \ldots, y_n) \in [0,1]^n$ and $S \subseteq [n]$; we then define $\textsc{DepRound}(y, S) \subseteq S$ to be $\textsc{DepRound}(x)$, for the vector x defined by $x_i = y_i[[i \in S]]$.

The greedy clustering procedure takes an input a set of weights w_j and sets $F_j \subseteq \mathcal{F}$ for every client $j \in \mathcal{C}$, and executes the following procedure:

Algorithm 1 $\textsc{GreedyCluster}(F, w)$

Sort $\mathcal{C}$ as $\mathcal{C} = \{j_1, j_2, \ldots, j_\ell\}$ where $w_{j_1} \le w_{j_2} \le \cdots \le w_{j_\ell}$.
Initialize $C' = \emptyset$
for $t = 1, \ldots, \ell$ **do**
  if $F_{j_t} \cap F_{j'} = \emptyset$ for all $j' \in C'$ **then** update $C' \leftarrow C' \cup \{j_t\}$
Return C'

Observation 2 *If* $C' = \textsc{GreedyCluster}(F, w)$ *then for any* $j \in \mathcal{C}$ *there is* $z \in C'$ *with* $w_z \le w_j$ *and* $F_z \cap F_j \ne \emptyset$.

2. The Chance k-coverage Problem

In this section, we consider a scenario we refer to as the *chance k-coverage problem*: every point $j \in \mathcal{C}$ has demand parameters p_j, r_j, and we wish to find a k-lottery Ω such that

$$\Pr_{\mathcal{S} \sim \Omega}[d(j, \mathcal{S}) \le r_j] \ge p_j. \tag{1}$$

If a k-lottery satisfying (1) exists, we say that parameters p_j, r_j are *feasible*. We refer to the special case wherein every client j has a common value $p_j = p$ and a common value $r_j = r$, as *homogeneous*. Homogeneous instances naturally correspond to fair allocations, for example, k-supplier is a special case of the homogeneous chance k-coverage problem, in which $p_j = 1$ and r_j is equal to the optimal k-supplier radius.

Our approximation algorithms for this problem will be based on a linear programming (LP) relaxation which we denote $\mathcal{P}_{\text{chance}}$. It has fractional variables b_i, where i ranges over $\mathcal{F}$ (b_i represents the probability of opening facility i), and is defined by the following constraints:

(B1) $\sum_{i \in B(j, r_j)} b_i \ge p_j$ for all $j \in \mathcal{C}$,

(B2) $b(\mathcal{F}) = k$,

(B3) $b_i \in [0, 1]$ for all $i \in \mathcal{F}$.

Proposition 3 *If parameters* p, r *are feasible, then* $\mathcal{P}_{chance}$ *is nonempty.*

Proof Consider a distribution Ω satisfying (1). For each $i \in \mathcal{F}$, set $b_i = \Pr_{\mathcal{S} \sim \Omega}[i \in \mathcal{S}]$. For $j \in \mathcal{C}$ we have $p_j = \Pr\left[\bigvee_{i \in B(j, r_j)} i \in \mathcal{S}\right] \le \sum_{i \in B(j, r_j)} \Pr[i \in S] = \sum_{i \in B(j, r_j)} b_i$ and thus (B1) is satisfied. We have $k = \mathbf{E}[|\mathcal{S}|] = \sum_{i \in \mathcal{F}} \Pr[i \in S] = b(\mathcal{F})$ and so (B2) is satisfied. (B3) is clear, so we have demonstrated a point in $\mathcal{P}_{\text{chance}}$. ■

For the remainder of this section, we assume we have a vector $b \in \mathcal{P}_{\text{chance}}$ and focus on how to round it to an integral solution. By a standard facility-splitting step, we also generate, for every $j \in \mathcal{C}$, a set $F_j \subseteq B(j, r_j)$ with $b(F_j) = p_j$. We refer to this set F_j as a *cluster*. In the SCC setting, it will also be convenient to ensure that $j \in F_j$ as long as $b_j \ne 0$.

As we show in Section 4, any approximation algorithm must either significantly give up a guarantee on the distance, or probability (or both). Our first result is an approximation algorithm which respects the distance guarantee exactly, with constant-factor loss to the probability guarantee:

Theorem 4 *If* p, r *is feasible then one may efficiently construct a* k*-lottery* Ω *satisfying*

$$\Pr_{\mathcal{S} \sim \Omega}[d(j, \mathcal{S}) \le r_j] \ge (1 - 1/e) p_j.$$

Proof Let $b \in \mathcal{P}_{\text{chance}}$ and set $\mathcal{S} = \textsc{DepRound}(b)$. This satisfies $|\mathcal{S}| \le \lceil \sum_{i=1}^n b_i \rceil \le \lceil k \rceil = k$ as desired. Each $j \in \mathcal{C}$ has

$$\Pr[\mathcal{S} \cap F_j = \emptyset] \le \prod_{i \in F_j}(1 - b_i) \le \prod_{i \in F_j} e^{-b_i} = e^{-b(F_i)} = e^{-p_j}.$$

and then simple analysis shows that

$$\Pr[d(j, \mathcal{S}) \le r_j] \ge \Pr[\mathcal{S} \cap F_j \ne \emptyset] \ge 1 - e^{-p_j} \ge (1 - 1/e) p_j.$$ ■

As we will later show in Theorem 20, this approximation constant $1 - 1/e$ is optimal.

We next turn to preserving the probability guarantee exactly with some loss to distance guarantee. As a warm-up exercise, let us consider the special case of "half-homogeneous" problem instances: all the values of p_j are the same, or all the values of r_j are the same. A similar algorithm works both these cases: we first select a set of clusters according to some greedy order, and then open a single item from each cluster. We summarize this as follows:

Algorithm 2 Rounding algorithm for half-homogeneous chance k-coverage

Set $C' = \textsc{GreedyCluster}(F_j, w_j)$
Set $Y = \textsc{DepRound}(p, C')$
Form solution set $\mathcal{S} = \{V_j \mid j \in Y\}$.

Algorithm 2 opens at most k facilities, as the dependent rounding step ensures that $|Y| \le \lceil \sum_{j \in C'} p_j \rceil = \lceil \sum_{j \in C'} b(F_j) \rceil \le \lceil \sum_{i \in \mathcal{F}} b_i \rceil \le k$. The only difference between the two cases is the choice of weighting function w_j for the greedy cluster selection.

Proposition 5 *1. Suppose that* p_j *is the same for every* $j \in \mathcal{C}$*. Then using the weighting function* $w_j = r_j$ *ensures that every* $j \in \mathcal{C}$ *satisfies* $\Pr[d(j, \mathcal{S}) \le 3r_j] \ge p_j$*. Furthermore, in the SCC setting, it satisfies* $\Pr[d(j, \mathcal{S}) \le 2r_j] \ge p_j$.

2. *Suppose r_j is the same for every $j \in \mathcal{C}$. Then using the weighting function $w_j = 1 - p_j$ ensures that every $j \in \mathcal{C}$ satisfies $\Pr[d(j, \mathcal{S}) \le 3r_j] \ge p_j$. Furthermore, in the SCC setting, it satisfies $\Pr[d(j, \mathcal{S}) \le 2r_j] \ge p_j$.*

Proof Let $j \in \mathcal{C}$. By Observation 2 there is $z \in C'$ with $w_z \le w_j$ and $F_j \cap F_z \ne \emptyset$. In either of the two cases, this implies that $p_z \ge p_j$ and $r_z \le r_j$.

Letting $i \in F_j \cap F_z$ gives $d(j, z) \le d(j, i) + d(z, i) \le r_j + r_z \le 2r_j$. This $z \in C'$ survives to Y with probability $p_z \ge p_j$, and in that case we have $d(z, \mathcal{S}) = \theta(z)$. In the SCC setting, this means that $d(z, \mathcal{S}) = 0$; in the general setting, we have $\theta(z) \le r_z \le r_j$. ■

2.1. Approximating the General Case

We next show how to satisfy the probability constraint exactly for the general case of chance k-coverage, with a constant-factor loss in the distance guarantee. Namely, we will find a probability distribution with

$$\Pr_{\mathcal{S} \sim \Omega}[d(j, \mathcal{S}) \le 9r_j] \ge p_j.$$

The algorithm is based on iterated rounding, in which the entries of b go through an unbiased random walk until b becomes integral (and, thus corresponds to a solution set $\mathcal{S}$). Because the walk is unbiased, the probability of serving a client at the end is equal to the fractional probability of serving a client, which will be at least p_j. In order for this process to make progress, the number of active variables must be greater than the number of active constraints. We ensure this by periodically identifying and discarding clients which will be automatically served by serving other clients. This is similar to a method of Krishnaswamy et al. (2018), which also uses iterative rounding for (deterministic) approximations to k-median with outliers and k-means with outliers.

The sets F_j will remain fixed during this procedure. We will maintain a vector $b \in [0, 1]^{\mathcal{F}}$ and maintain two sets C_{tight} and C_{slack} with the following properties:

(C1) $C_{\text{tight}} \cap C_{\text{slack}} = \emptyset$.

(C2) For all $j, j' \in C_{\text{tight}}$, we have $F_j \cap F_{j'} = \emptyset$

(C3) Every $j \in C_{\text{tight}}$ has $b(F_j) = 1$,

(C4) Every $j \in C_{\text{slack}}$ has $b(F_j) \le 1$.

(C5) We have $b(\bigcup_{j \in C_{\text{tight}} \cup C_{\text{slack}}} F_j) \le k$

Given our initial solution b for $\mathcal{P}_{\text{chance}}$, setting $C_{\text{tight}} = \emptyset, C_{\text{slack}} = \mathcal{C}$ will satisfy criteria (C1)–(C5); note that (C4) holds as $b(F_j) = p_j \le 1$ for all $j \in \mathcal{C}$.

Proposition 6 *Given any vector $b \in [0, 1]^{\mathcal{F}}$ satisfying constraints (C1)—(C5) with $C_{slack} \ne \emptyset$, it is possible to generate a random vector $b' \in [0, 1]^{\mathcal{F}}$ such that $\mathbf{E}[b'] = b$, and with probability one b' satisfies constraints (C1) — (C5) as well as having some $j \in C_{slack}$ with $b'(F_j) \in \{0, 1\}$.*

Proof We will show that any basic solution $b \in [0, 1]^{\mathcal{F}}$ to the constraints (C1)—(C5) with $C_{\text{slack}} \ne \emptyset$ must satisfy the condition that $b(F_j) \in \{0, 1\}$ for some $j \in C_{\text{slack}}$. To obtain the stated result, we simply modify b until it becomes basic by performing an unbiased walk in the nullspace of the tight constraints.

So consider a basic solution b. Define $A = \bigcup_{j \in C_{\text{tight}}} F_j$ and $B = \bigcup_{j \in C_{\text{slack}}} F_j$. We assume that $b(F_j) \in (0, 1)$ for all $j \in C_{\text{slack}}$, as otherwise we are done.

First, suppose that $b(A \cap B) > 0$. So there must be some pair $j \in C_{\text{slack}}, j' \in C_{\text{tight}}$ with $i \in F_j \cap F_{j'}$ such that $b_i > 0$. Since $b(F_{j'}) = 1$, there must be some other $i' \in F_{j'}$ with $b_{i'} > 0$. Consider modifying b by incrementing b_i by $\pm\epsilon$ and decrementing $b_{i'}$ by $\pm\epsilon$, for some sufficiently small ϵ. Constraint (C5) is preserved. Since $F_{j'} \cap F_{j''} = \emptyset$ for all $j'' \in C_{\text{tight}}$, constraint (C3) is preserved. Since the (C4) constraints are slack, then for ϵ sufficiently small they are preserved as well. This contradicts that b is basic.

Next, suppose that $b(A \cap B) = 0$ and $b(A \cup B) < k$ strictly. Let $j \in C_{\text{slack}}$ and $i \in F_j$ with $b_i > 0$; if we change b_i by $\pm\epsilon$ for sufficiently small ϵ, this preserves (C4) and (C5); furthermore, since $i \notin A$, it preserves (C3) as well. So again b cannot be basic.

Finally, suppose that $b(A \cap B) = 0$ and $b(A \cup B) = k$. So $b(B) = k - |A|$ and $b(B) > 0$ as $C_{\text{slack}} \ne \emptyset$. Therefore, there must be at least two elements $i, i' \in B$ such that $b_i > 0, b_{i'} > 0$. If we increment b_i by $\pm\epsilon$ while decrementing $b_{i'}$ by $\pm\epsilon$, this again preserves all the constraints for ϵ sufficiently small, contradicting that b is basic. ■

We can now describe our iterative rounding algorithm, Algorithm 3.

Algorithm 3 Iterative rounding algorithm for chance k-coverage

1: Find a fractional solution b to $\mathcal{P}_{\text{chance}}$ and form the corresponding sets F_j.
2: Initialize $C_{\text{tight}} = \emptyset, C_{\text{slack}} = \mathcal{C}$
3: **while** $C_{\text{slack}} \ne \emptyset$ **do**
4: Draw a fractional solution b' with $\mathbf{E}[b'] = b$ according to Proposition 6.
5: Select some $v \in C_{\text{slack}}$ with $b'(F_v) \in \{0, 1\}$.
6: Update $C_{\text{slack}} \leftarrow C_{\text{slack}} - \{v\}$
7: **if** $b'(F_v) = 1$ **then**
8: Update $C_{\text{tight}} \leftarrow C_{\text{tight}} \cup \{v\}$
9: **if** there is any $z \in C_{\text{tight}} \cup C_{\text{slack}}$ such that $r_z \ge r_v/2$ and $F_z \cap F_v \ne \emptyset$ **then**
10: Update $C_{\text{tight}} \leftarrow C_{\text{tight}} - \{z\}, C_{\text{slack}} \leftarrow C_{\text{slack}} - \{z\}$
11: Update $b \leftarrow b'$.
12: For each $j \in C_{\text{tight}}$, open an arbitrary center in F_j.

To analyze this algorithm, define $C^t_{\text{tight}}, C^t_{\text{slack}}, b^t$ to be the values of the relevant variables at iteration t. Since every step removes at least one point from C_{slack}, this process must terminate in $T \le n$ iterations. We will write b^{t+1} to refer to the random value b' chosen at step (4) of iteration t, and v^t denote the choice of $v \in C_{\text{slack}}$ used step in step (5) of iteration t.

Proposition 7 *The vector b^t satisfies constraints (C1) — (C5) for all times $t = 1, \ldots, T$.*

Proof The vector b^0 does so since b satisfies $\mathcal{P}_{\text{chance}}$. Proposition 6 ensures that step (4) does not affect this. Also, removing points from C_{tight} or C_{slack} at step (6) or (1) will not violate these constraints.

Let us check that adding v^t to C_{tight} will not violate the constraints. This step only occurs if $b^{t+1}(v^t) = 1$, and so (C3) is preserved. Since we only move v^t from C_{slack} to C_{tight}, constraints (C1) and (C5) are preserved. Finally, to show that (C2) is preserved, suppose that $F_{v^t} \cap F_{v^s} \neq \emptyset$ for some other v^s which was added to C_{tight} at time $s < t$. If $r_{v^t} \geq r_{v^s}$, then step (10) would have removed v^t from C_{slack}, making it impossible to enter C^t_{tight}. Thus, $r_{v^t} \leq r_{v^s}$; this means that when we add v^t to C^t_{tight}, we also remove v^s from C^t_{tight}. ■

Corollary 8 *Algorithm 3 opens at most k facilities.*

Proof At the final step (12), the number of open facilities is equal to $|C_{\text{tight}}|$. By Proposition 7, the vector b^T satisfies constraints (C1) — (C5). So $b(F_j) = 1$ for $j \in C_{\text{tight}}$ and $F_j \cap F_{j'} = \emptyset$ for $j, j \in C_{\text{tight}}$; thus $|C_{\text{tight}}| = \sum_{j \in C_{\text{tight}}} b(F_j) \leq k$. ■

Proposition 9 *If $j \in C^t_{tight}$ for any time t, then $d(j, \mathcal{S}) \leq 3r_j$.*

Proof Let t be maximal such that $j \in C^t_{\text{tight}}$. We show the desired claim by induction on t. When $t = T$, then this certainly holds as step (12) will open some facility in F_j and thus $d(j, \mathcal{S}) \leq r_j$.

Suppose that j was added into C^s_{tight}, but was later removed from C^{t+1}_{tight} due to adding $z = v^t$. Thus there is some $i \in F_z \cap F_j$. When we added j in time s, we would have removed z from C^s_{tight} if $r_z \geq r_j/2$. Since this did not occur, it must hold that $r_z < r_j/2$.

Since z is present in C^{t+1}_{tight}, the induction hypothesis implies that $d(z, \mathcal{S}) \leq 3r_z$ and so

$$d(j, \mathcal{S}) \leq d(j, i) + d(i, z) + d(z, \mathcal{S}) \leq r_j + r_z + 3r_z \leq r_j + (r_j/2)(1 + 3) = 3r_j. \quad ■$$

Theorem 10 *Every $j \in \mathcal{C}$ has $\Pr[d(j, \mathcal{S}) \leq 9r_j] \geq p_j$.*

Proof We will prove by induction on t the following claim: suppose we condition on the full state of Algorithm 3 up to time t, and that $j \in C^t_{\text{tight}} \cup C^t_{\text{slack}}$. Then

$$\Pr[d(j, \mathcal{S}) \leq 9r_j] \geq b^t(F_j). \tag{2}$$

At $t = T$, this is clear; since $C^T_{\text{slack}} = \emptyset$, we must have $j \in C^T_{\text{tight}}$, and so $d(j, \mathcal{S}) \leq r_j$ with probability one. For the induction step at time t, note that as $\mathbf{E}[b^{t+1}(F_j)] = b(F_j)$, in order to prove (2) it suffices to show that if we also condition on the value of b^{t+1}, it holds that

$$\Pr[d(j, \mathcal{S}) \leq 9r_j \mid b^{t+1}] \geq b^{t+1}(F_j). \tag{3}$$

If j remains in $C^{t+1}_{\text{tight}} \cup C^{t+1}_{\text{slack}}$, then we immediately apply the induction hypothesis at time $t + 1$. So the only non-trivial thing to check is that (3) will hold even if j is removed from $C^{t+1}_{\text{tight}} \cup C^{t+1}_{\text{slack}}$.

If $j = v^t$ and $b^{t+1}(F_j) = 0$, then (3) holds vacuously. Otherwise, suppose that j is removed from C^t_{tight} at stage (10) due to adding $z = v^t$. Thus $r_j \geq r_z/2$ and there is some $i \in F_j \cap F_z$. By Proposition 9, this ensures that $d(z, \mathcal{S}) \leq 3r_z$. Thus with probability one we have

$$d(j, \mathcal{S}) \leq d(j, i) + d(i, z) + d(z, \mathcal{S}) \leq r_j + r_z + 3r_z \leq r_j + (2r_j)(1 + 3) = 9r_j.$$

This proves the induction. The claimed result follows since $b^0(F_j) = p_j$ and $C^0_{\text{slack}} = \mathcal{C}$. ■

3. Chance k-coverage: Approximating the Deterministic Case

An important special case of k-coverage is where $p_j = 1$ for all $j \in \mathcal{C}$. Here, the target distribution Ω is just a single set $\mathcal{S}$ satisfying $\forall j d(j, \mathcal{S}) \leq r_j$. In the homogeneous case, when all the r_j are equal to the same value, this is specifically the k-supplier problem. The usual approximation algorithm for this problem chooses a single approximating set $\mathcal{S}$, in which case the best guarantee available is $d(j, \mathcal{S}) \leq 3r_j$. We improve the distance guarantee by constructing a k-lottery $\tilde{\Omega}$ such that $d(j, \mathcal{S}) \leq 3r_j$ with probability one, and $\mathbf{E}_{\mathcal{S} \sim \tilde{\Omega}}[d(j, \mathcal{S})] \leq cr_j$, where the constant c satisfies the following bounds:

1. In the general case, $c = 1 + 2/e \approx 1.73576$;
2. In the SCC setting, $c = 1.60793$;
3. In the homogeneous SCC setting, $c = 1.592$.[3]

We show matching lower bounds in Section 4; the constant value $1 + 2/e$ is optimal for the general case (even for homogeneous instances), and for the third case the constant c cannot be made lower than $1 + 1/e \approx 1.367$.

We remark that this type of stochastic guarantee allows us to efficiently construct publicly-verifiable lotteries.

Proposition 11 *Let $\epsilon > 0$. In any of the above three cases, there is an expected polynomial time procedure to convert the given distribution Ω into an explicitly-enumerated k-lottery Ω', with support size $O(\frac{\log n}{\epsilon^2})$, such that $\Pr_{\mathcal{S} \sim \Omega'}[d(j, \mathcal{S}) \leq 3r_j] = 1$ and $\mathbf{E}_{\mathcal{S} \sim \Omega'}[d(j, \mathcal{S})] \leq c(1 + \epsilon)r_j$.*

Proof Take $X_1, \ldots, X_t$ as independent draws from Ω for $t = \frac{6 \log n}{c\epsilon^2}$ and set Ω' to be the uniform distribution on $\{X_1, \ldots, X_t\}$. To see that $\mathbf{E}_{\mathcal{S} \sim \Omega'}[d(j, \mathcal{S})] \leq c(1 + \epsilon)r_j$ holds with high probability, apply a Chernoff bound, noting that $d(j, X_1), \ldots, d(j, X_t)$ are independent random variables in the range $[0, 3r_j]$. ■

We use a similar algorithm to Algorithm 2 for this problem: we choose a covering set of clusters C', and open exactly one item from each cluster. The main difference is that

3. This value was calculated using some non-rigorous numerical analysis; we describe this further in what we call "Pseudo-Theorem" 17

instead of opening the nearest item V_j for each $j \in C'$, we instead open a cluster according to the solution b of $\mathcal{P}_{\text{chance}}$.

Algorithm 4 Rounding algorithm with clusters

Set $C' = \textsc{GreedyCluster}(F_j, r_j)$.
Set $F_0 = \mathcal{F} - \bigcup_{j \in C'} F_j$; this is the set of "unclustered" facilities
for $j \in C'$ **do**
  Randomly select a point $W_j \in F_j$ according to the distribution $\Pr[W_j = i] = b_i$
  // This is a valid probability distribution, as $b(F_j) = 1$
Let $\mathcal{S}_0 \leftarrow \textsc{DepRound}(b, F_0)$
Return $\mathcal{S} = \mathcal{S}_0 \cup \{W_j \mid j \in C'\}$

We will need the following technical result in order to analyze Algorithm 4.

Proposition 12 *For any set $U \subseteq \mathcal{F}$, we have*

$$\Pr[\mathcal{S} \cap U = \emptyset] \le \prod_{i \in U \cap F_0} (1 - b_i) \prod_{j \in C'} (1 - b(U \cap F_j)) \le e^{-b(U)}.$$

Proof The set U contains each W_j independently with probability $b(U \cap F_j)$. The set $\mathcal{S}_0$ is independent of them and by (P3) we have $\Pr[U \cap \mathcal{S}_0 = \emptyset] \le \prod_{i \in U \cap F_0} (1 - b_i)$. So

$$\Pr[\mathcal{S} \cap U = \emptyset] \le \prod_{i \in U \cap F_0} (1 - b_i) \prod_{j \in C'} (1 - b(U \cap F_j)) \le \prod_{i \in U \cap F_0} e^{-b_i} \prod_{j \in C'} e^{-b(U \cap F_j)} = e^{-b(U)} \quad \blacksquare$$

At this point, we can show our claimed approximation ratio for the general (non-SCC) setting:

Theorem 13 *For any $j \in \mathcal{C}$, the solution set $\mathcal{S}$ of Algorithm 4 satisfies $d(j, \mathcal{S}) \le 3r_j$ with probability one and $\mathbf{E}[d(j, \mathcal{S})] \le (1 + 2/e) r_j$.*

Proof By Observation 2, there is some $v \in C'$ with $F_j \cap F_v \ne \emptyset$ and $r_v \le r_j$. Letting $i \in F_j \cap F_v$, we have

$$d(j, \mathcal{S}) \le d(j, i) + d(i, v) + d(v, \mathcal{S}) \le r_j + r_v + r_v \le 3r_j.$$

with probability one.

If $\mathcal{S} \cap F_j \ne \emptyset$, then $d(j, \mathcal{S}) \le r_j$. Thus, a necessary condition for $d(j, \mathcal{S}) > r_j$ is that $\mathcal{S} \cap F_j = \emptyset$. Applying Proposition 12 with $U = F_j$ gives

$$\Pr[d(j, \mathcal{S}) > r_j] \le \Pr[\mathcal{S} \cap F_j = \emptyset] \le e^{-b(F_j)} = e^{-1}$$

and so $\mathbf{E}[d(j, \mathcal{S})] \le r_j + 2r_j \Pr[d(j, \mathcal{S}) > r_j] \le (1 + 2/e) r_j$. $\blacksquare$

3.1. The SCC Setting

To motivate the algorithm for the SCC setting (where $\mathcal{C} = \mathcal{F}$), note that if some client $j \in \mathcal{C}$ has some facility i opened in a nearby cluster F_v, then this guarantees that $d(j, \mathcal{S}) \le d(j, v) + d(v, i) \le 3r_j$. This is what we used to analyze the non-SCC setting. But, if instead of opening facility i, we opened v itself, then this would ensure that $d(j, \mathcal{S}) \le 2r_j$. Thus, opening the centers of a cluster can lead to better distance guarantees compared to opening any other facility. We emphasize that this is only possible in the SCC setting, as in general we do not know that $v \in \mathcal{F}$.

We use the following Algorithm 5, which takes a parameter $q \in [0, 1]$ which we will discuss how to set shortly. We recall that we have assumed in this case that $j \in F_j$ for every $j \in \mathcal{C}$. (Also, note our notational convention that $\overline{q} = 1 - q$; this will be used extensively in this section to simplify the formulas.)

Algorithm 5 Rounding algorithm for k-center

Set $C' = \textsc{GreedyCluster}(F_j, r_j)$.
Set $F_0 = \mathcal{F} - \bigcup_{j \in C'} F_j$; this is the set of "unclustered" facilities
for $j \in C'$ **do**
  Randomly select $W_j \in F_j$ according to the distribution $\Pr[W_j = i] = \overline{q} b_i + q[[i = j]]$
Let $\mathcal{S}_0 = \textsc{DepRound}(b, F_0)$
Return $\mathcal{S} = \mathcal{S}_0 \cup \{W_j \mid j \in C'\}$

This is the same as Algorithm 4, except that some of the values of b_i for $i \in F_j$ have been shifted to the cluster center j. In fact, we can think of Algorithm 5 as a two-part process: we first modify the fractional vector b to obtain a new fractional vector b' defined by

$$b'_i = \begin{cases} \overline{q} b_i + q & \text{if } i \in C' \\ \overline{q} b_i & \text{if } i \in F_j - \{j\} \text{ for } j \in C' \\ b_i & \text{if } i \in F_0 \end{cases}.$$

and we then execute Algorithm 4 on the resulting vector b'. In particular, Proposition 12 remains valid with respect to the modified vector b'.

Theorem 14 *Let $j \in \mathcal{C}$. After running Algorithm 5 with $q = 0.464587$ we have $d(j, \mathcal{S}) \le 3r_j$ with probability one and $\mathbf{E}[d(j, \mathcal{S})] \le 1.60793 r_j$.*

Proof Let $D = \{v \in C' \mid F_j \cap F_v \ne \emptyset, r_v \le r_j\}$; note that $D \ne \emptyset$ by Observation 2. For each $v \in D \cup \{0\}$, set $a_v = b(F_j \cap F_v)$ and observe that $a_0 + \sum_{v \in D} a_v = b(F_j) = 1$.

As before, a necessary condition for $d(j, \mathcal{S}) > r_j$ is that $F_j \cap \mathcal{S} = \emptyset$. So by Proposition 12,

$$\begin{aligned} \Pr[d(j, \mathcal{S}) > r_j] \le \Pr[F_j \cap \mathcal{S} = \emptyset] &\le \prod_{i \in F_j \cap F_0} (1 - b'_i) \prod_{v \in C'} (1 - b'(F_v \cap F_j)) \\ &\le \prod_{i \in F_j \cap F_0} (1 - b_i) \prod_{v \in D} (1 - \overline{q} b(F_v \cap F_j)) \le e^{-b(F_j \cap F_0)} \prod_{v \in D} (1 - \overline{q} b(F_v \cap F_j)) \\ &= e^{-a_0} \prod_{v \in D} e^{a_v} (1 - \overline{q} a_v) = (1/e) \prod_{v \in D} e^{a_v} (1 - \overline{q} a_v). \end{aligned}$$

where the last equality comes from the fact that $a_0 + a(D) = 1$.

Similarly, if there is some $i \in \mathcal{S} \cap D$, then $d(j, \mathcal{S}) \le d(v, i) \le 2r_j$. Thus, a necessary condition for $d(j, \mathcal{S}) > 2r_j$ is that $\mathcal{S} \cap (D \cup F_j) = \emptyset$. Applying Proposition 12 with $U = D \cup F_j$ gives:

$$\begin{aligned}\Pr[d(j,\mathcal{S}) > 2r_j] &\le \prod_{v \in (D \cup F_j) \cap F_0} (1 - b_v) \prod_{v \in C'} (1 - b'((D \cup F_j) \cap F_v)) \\ &\le e^{-b(F_j \cap F_0)} \prod_{v \in D} \overline{q}(1 - a_v) = (1/e) \prod_{v \in D} e^{a_v} \overline{q}(1 - a_v)\end{aligned}$$

Putting these together gives:

$$\mathbf{E}[d(j,\mathcal{S})] \le r_j \Big(1 + 1/e \prod_{v \in D} e^{a_v}(1 - \overline{q} a_v) + 1/e \prod_{v \in D} e^{a_v} \overline{q}(1 - a_v)\Big) \tag{4}$$

Let us define $s = a(D)$ and $t = |D|$. By the AM-GM inequality we have:

$$\mathbf{E}[d(j,\mathcal{S})] \le r_j \Big(1 + e^{s-1}\big(1 - \overline{q} s/t\big)^t + e^{s-1} \overline{q}^t (1 - s/t)^t\Big)$$

This is a function of a single real parameter $s \in [0, 1]$ and a single integer parameter $t \ge 1$. Some simple analysis, which we omit here, shows that $\mathbf{E}[d(j, \mathcal{S})] \le 1.60793 r_j$. ■

3.2. The Homogeneous SCC Setting

From the point of view of the target distribution Ω, this setting is equivalent to the classical k-center problem. We may guess the optimal radius, and so we do not need to assume that the common value of r_j is "given" to us by some external process. By rescaling, we assume without loss of generality here that $r_j = 1$ for all j.

We will improve on Theorem 14 through a more complicated rounding process based on greedily-chosen partial clusters. Specifically, we select cluster centers $\pi(1), \ldots, \pi(n)$, wherein $\pi(i)$ is chosen to maximize $b(F_{\pi(i)} - F_{\pi(1)} - \cdots - F_{\pi(i-1)})$. By renumbering $\mathcal{C}$, we may assume without loss of generality that the resulting permutation π is the identity; therefore, we assume throughout this section that $\mathcal{C} = \mathcal{F} = [n]$ and for all pairs $i < j$ we have

$$b(F_i - F_1 - \cdots - F_{i-1}) \ge b(F_j - F_1 - \cdots - F_{i-1}) \tag{5}$$

For each $j \in [n]$, we let $G_j = F_j - F_1 - \cdots - F_{j-1}$ and we define $z_j = b(G_j)$. We say that G_j is a *full cluster* if $z_j = 1$ and a *partial cluster* otherwise. We note that the values of z appear in sorted order $1 = z_1 \ge z_2 \ge z_3 \ge \cdots \ge z_n \ge 0$.

We use the following randomized Algorithm 7 to select the centers. Here, the quantities $Q_{\mathrm{f}}, Q_{\mathrm{p}}$ (short for *full* and *partial*) are drawn from a joint probability distribution which we discuss later.

Algorithm 7 Partial-cluster based algorithm

1: Draw random variables $Q_{\mathrm{f}}, Q_{\mathrm{p}}$.
2: $Z \leftarrow \textsc{DepRound}(z)$
3: **for** $j \in Z$ **do**
4: Randomly select $W_j \in G_j$ according to the distribution $\Pr[W_j = i] = (\overline{q_j} y_i + q_j[[i = j]])/z_j$ where q_j is defined as

$$q_j = \begin{cases} Q_{\mathrm{f}} & \text{if } z_j = 1 \\ Q_{\mathrm{p}} & \text{if } z_j < 1 \end{cases}$$

5: Return $\mathcal{S} = \{W_j \mid j \in Z\}$

Before the technical analysis of Algorithm 7, let us provide some intuition. It may be beneficial to open the center of some cluster near a given client $j \in \mathcal{C}$ as this will ensure $d(j, \mathcal{S}) \le 2$. However, there is no benefit to opening more than one such cluster center. So, we would like a significant negative correlation between opening the centers of distinct clusters near j. Unfortunately, the full clusters all "look alike," and so it seems impossible to enforce any significant negative correlation among them.

Partial clusters break the symmetry. There is at least one full cluster near j, and possibly some partial clusters as well. We will create a probability distribution with significant negative correlation between the event that partial clusters open their centers and the event that full clusters open their centers. This decreases the probability that j will see multiple neighboring clusters open their centers, which in turn gives an improved value of $\mathbf{E}[d(j, \mathcal{S})]$.

The dependent rounding in Algorithm 7 ensures that $|Z| \le \lceil \sum_{j=1}^n z_j \rceil = \sum_{j=1}^n b(F_j - F_1 - \cdots - F_{j-1}) = b(\mathcal{F}) \le k$, and so $|\mathcal{S}| \le k$ as required. We also need the following technical result; the proof is essentially the same as Proposition 12 and is omitted.

Proposition 15 *For any $U \subseteq \mathcal{C}$, we have*

$$\Pr[\mathcal{S} \cap U = \emptyset \mid Q_f, Q_p] \le \prod_{j=1}^n \big(1 - \overline{q}_j b(U \cap G_j) - q_j z_j [[j \in U]]\big).$$

Our main lemma to analyze Algorithm 7 is the following:

Lemma 16 *Let $i \in [n]$. Define $J_f, J_p \subseteq [n]$ as*

$$J_f = \{j \in [n] \mid F_i \cap G_j \ne \emptyset, z_j = 1\}$$
$$J_p = \{j \in [n] \mid F_i \cap G_j \ne \emptyset, z_j < 1\}$$

Let $m = |J_f|$ and $J_p = \{j_1, \ldots, j_t\}$ where $j_1 \le j_2 \le \cdots \le j_t$. For each $\ell = 1, \ldots, t+1$ define

$$u_\ell = b(F_i \cap G_{j_\ell}) + b(F_i \cap G_{j_{\ell+1}}) + \cdots + b(F_i \cap G_{j_t})$$

Then $1 \ge u_1 \ge u_2 \ge \cdots \ge u_t \ge u_{t+1} = 0$ and $m \ge 1$. Furthermore, if we condition on a fixed value of (Q_f, Q_p) then we have

$$\mathbf{E}[d(i,\mathcal{S})] \le 1 + \left(1 - \overline{Q_f}\frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^t (1 - \overline{Q_p}(u_\ell - u_{\ell+1})) + \left(\overline{Q_f}\left(1 - \frac{\overline{u_1}}{m}\right)\right)^m \prod_{\ell=1}^t (\overline{u_\ell} + \overline{Q_p} u_{\ell+1}).$$

Proof For $\ell = 1, \ldots, t$ we let $a_\ell = b(F_i \cap G_{j_\ell}) = u_\ell - u_{\ell+1}$. For $j \in J_\mathrm{f}$, we let $s_j = b(F_i \cap G_j)$.

First, we claim that $z_{j_\ell} \geq u_\ell$ for $\ell = 1, \ldots, t$. For, by (5), we have

$$\begin{aligned} z_{j_\ell} &\geq b(F_i - F_1 - \cdots - F_{j_\ell - 1}) \geq b(F_i) - \sum_{j \in J_\mathrm{f}} b(F_i \cap G_j) - \sum_{v < j_\ell} b(F_i \cap G_v) \\ &= b(F_i) - \sum_{j \in J_\mathrm{f}} b(F_i \cap G_j) - \sum_{v < \ell} b(F_i \cap G_{j_v}) \qquad \text{as } F_i \cap G_v = \emptyset \text{ for } v \notin J_\mathrm{p} \cup J_\mathrm{f} \\ &= 1 - \sum_{j \in J_\mathrm{f}} s_j - \sum_{v=1}^{\ell-1} a_v = u_\ell. \end{aligned}$$

Next, observe that if $m = 0$, we have $u_1 = \sum_{\ell=1}^t b(F_i \cap G_{j_\ell}) = \sum_{j=1}^n b(F_i \cap G_j) = b(F_i) = 1$. Applying the above bound at $\ell = 1$ gives $z_{j_1} \geq u_1 = 1$. But, by definition of J_p we have $z_{j_1} < 1$, which is a contradiction. This shows that $m \geq 1$.

We finish by showing the bound on $\mathbf{E}[d(i, \mathcal{S})]$. All the probabilities here should be interpreted as conditioned on fixed values for parameters $Q_\mathrm{f}, Q_\mathrm{p}$.

A necessary condition for $d(i, \mathcal{S}) > 1$ is that no point in F_i is open. Applying Proposition 15 with $U = F_i$ yields

$$\begin{aligned} \Pr[\mathcal{S} \cap F_i = \emptyset] &\leq \prod_{j \in J_\mathrm{f}} (1 - \overline{Q_\mathrm{f}} b(F_i \cap G_j) - Q_\mathrm{f}[[j \in F_i]]) \prod_{j \in J_\mathrm{p}} (1 - \overline{Q_\mathrm{p}} b(F_i \cap G_j)) - Q_\mathrm{p} z_j [[j \in F_i]]) \\ &\leq \prod_{j \in J_\mathrm{f}} (1 - \overline{Q_\mathrm{f}} b(F_i \cap G_j)) \prod_{j \in J_\mathrm{p}} (1 - \overline{Q_\mathrm{p}} b(F_i \cap G_j)) = \prod_{j \in J_\mathrm{f}} (1 - \overline{Q_\mathrm{f}} s_j) \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} a_\ell). \end{aligned}$$

A necessary condition for $d(i, \mathcal{S}) > 2$ is that we do not open any point in F_i, nor do we open center of any cluster intersecting with F_i. Applying Proposition 15 with $U = F_i \cup J_\mathrm{f} \cup J_\mathrm{p}$, and noting that $z_{j_\ell} \leq u_\ell$, we get:

$$\begin{aligned} \Pr[\mathcal{S} \cap U = \emptyset] &\leq \prod_{j \in J_\mathrm{f}} (1 - \overline{Q_\mathrm{f}} b(U \cap G_j) - Q_\mathrm{f}) \prod_{j \in J_\mathrm{p}} (1 - \overline{Q_\mathrm{p}} b(U \cap G_j)) - Q_\mathrm{p} z_j) \\ &\leq \prod_{j \in J_\mathrm{f}} (1 - \overline{Q_\mathrm{f}} b(F_i \cap G_j) - Q_\mathrm{f}) \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} b(F_i \cap G_{j_\ell}) - Q_\mathrm{p} z_{j_\ell}) \\ &= \prod_{j \in J_\mathrm{f}} \overline{Q_\mathrm{f}} (1 - s_j) \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} a_\ell - Q_\mathrm{p} z_{j_\ell}) \leq \prod_{j \in J_\mathrm{f}} \overline{Q_\mathrm{f}} (1 - s_j) \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} a_\ell - Q_\mathrm{p} u_\ell) \end{aligned}$$

Thus,

$$\mathbf{E}[d(i, \mathcal{S})] \leq 1 + \prod_{j \in J_\mathrm{f}} (1 - \overline{Q_\mathrm{f}} s_j) \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} a_\ell) + \prod_{j \in J_\mathrm{f}} \overline{Q_\mathrm{f}} (1 - s_j) \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} a_\ell - Q_\mathrm{p} u_\ell) \quad (6)$$

The sets G_j partition $[n]$, so $\sum_{j \in \mathrm{f}} s_j = 1 - \sum_{\ell=1}^t a_\ell = 1 - u_1$. So by the AM-GM inequality, we have

$$\mathbf{E}[d(i, \mathcal{S})] \leq 1 + \left(1 - \overline{Q_\mathrm{f}} \frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} a_\ell) + \left(\overline{Q_\mathrm{f}} (1 - \frac{\overline{u_1}}{m})\right)^m \prod_{\ell=1}^t (1 - \overline{Q_\mathrm{p}} a_\ell - Q_\mathrm{p} u_\ell). \quad (7)$$

The claim follows as $a_\ell = u_\ell - u_{\ell+1}$. ■

Lemma 16 gives an upper bound on $\mathbf{E}[d(i, \mathcal{S})]$ for a fixed distribution on $Q_\mathrm{f}, Q_\mathrm{p}$ and for fixed values of the parameters $m, u_1, \ldots, u_t$. By a computer search, along with a number of numerical tricks, we can obtain an upper bound over all values for m and all possible sequences $u_1, \ldots, u_t$ satisfying $1 = u_1 \geq u_2 \geq \cdots \geq u_t$. This gives the following result:

Pseudo-Theorem 17 *Suppose that Q_f, Q_p has the following distribution:*

$$(Q_f, Q_p) = \begin{cases} (0.4525, 0) & \textit{with probability } p = 0.773436 \\ (0.0480, 0.3950) & \textit{with probability } 1 - p \end{cases}.$$

Then for all $j \in \mathcal{C}$ we have $d(j, \mathcal{S}) \leq 3r_j$ with probability one, and $\mathbf{E}[d(j, \mathcal{S})] \leq 1.592 r_j$.

We call this a "pseudo-theorem" because some of the computer calculations used double-precision floating point for convenience, without carefully tracking rounding errors. In principle, this could have been computed using rigorous numerical analysis, giving a true theorem. Since there are a number of technical complications in this calculation, we defer the details to Appendix A.

4. Lower Bounds on Approximating Chance k-coverage

We next show lower bounds for the chance k-coverage problems of Sections 2 and 3. These constructions are adapted from lower bounds for approximability of k-median (Guha and Khuller, 1999), which in turn are based on the hardness of set cover.

Formally, a set cover instance consists of a collection of sets $\mathcal{B} = \{S_1, \ldots, S_m\}$ over a ground set $[n]$. For any set $X \subseteq [m]$ we define $S_X = \bigcup_{i \in X} S_i$. The goal is to select a collection $X \subseteq [m]$ of minimum size such that $S_X = [n]$. The minimum value of $|X|$ thus obtained is denoted by OPT.

We quote a result of Moshkovitz (2015) on the inapproximability of set cover.

Theorem 18 (Moshkovitz (2015)) *Assuming $P \neq NP$, there is no polynomial-time algorithm which guarantees a set-cover solution X with $S_X = [n]$ and $|X| \leq (1-\epsilon) \ln n \times OPT$, where $\epsilon > 0$ is any constant.*

We will need a simple corollary of Theorem 18, which is a (slight reformulation) of the hardness of approximating max-coverage.

Corollary 19 *Assuming $P \neq NP$, there is no polynomial-time algorithm which guarantees a set-cover solution X with $|X| \leq OPT$ and $|S_X| \geq cn$ for any constant $c > 1 - 1/e$.*

Proof Suppose for contradiction that $\mathcal{A}$ is such an algorithm. We will repeatedly apply $\mathcal{A}$ to solve residual instances, obtaining a sequence of solutions $X_1, X_2, \ldots,$. Specifically, for iterations $i \geq 1$, we define $U_i = [n] - \bigcup_{j<i} S_{X_j}$ and $\mathcal{B}_i = \{S \cap U_i \mid S \in \mathcal{B}\}$, and we let X_i denote the output of $\mathcal{A}$ on instance $\mathcal{B}_i$.

Each $\mathcal{B}_i$ has a solution of cost at most OPT. Thus, $|X_i| \leq \mathrm{OPT}$ and $|U_i \cap S_{X_i}| \geq c|U_i|$. Thus $|U_{i+1}| = |U_i - S_{X_i}| \leq (1-c)|U_i|$. So, for $s = \lceil 1 + \frac{\ln n}{\ln(\frac{1}{1-c})} \rceil$ we have $U_s = \emptyset$, and so

the set $X = X_1 \cup \cdots \cup X_s$ solves the original set-cover instance $\mathcal{B}$, and $|X| \le \sum_{i=1}^{s} |X_i| \le (1 + \frac{\ln n}{\ln(\frac{1}{1-c})})\mathrm{OPT}$.

Since $c > 1 - 1/e$, we have $\frac{1}{\ln(\frac{1}{1-c})} \le (1 - \Omega(1))$, which contradicts Theorem 18. ■

Theorem 20 *Assuming $P \neq NP$, there is a family of homogeneous chance k-coverage instances, with a feasible demand of $p_j = r_j = 1$ for all clients j, such that no polynomial-time algorithm can guarantee a distribution Ω with either of the following:*

1. *$\forall j \mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})] \le cr_j$ for constant $c < 1 + 2/e$*
2. *$\forall j \Pr_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S}) < 3r_j] \ge cp_j$ for constant $c > 1 - 1/e$*

In particular, approximation constants in Theorem 4, Proposition 5, and Theorem 13 cannot be improved.

Proof Consider a set cover instance $\mathcal{B} = \{S_1, \ldots, S_m\}$. We begin by guessing the value OPT (there are at most m possibilities, so this can be done in polynomial time). We define a k-center instance with $k = \mathrm{OPT}$ and with disjoint client and facility sets, where $\mathcal{F}$ is identified with $[m]$ and $\mathcal{C}$ is identified with $[n]$. We define d by $d(i,j) = 1$ if $j \in S_i$ and $d(i,j) = 3$ otherwise.

If X is an optimal solution to $\mathcal{B}$ then $d(j,X) \le 1$ for all points $j \in \mathcal{C}$. So there exists a (deterministic) distribution with feasible demand parameters $p_j = 1, r_j = 1$. Also, note that $d(j,\mathcal{S}) \in \{1,3\}$ with probability one for any client j. Thus, if j satisfies the second property $\Pr_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S}) < 3r_j] \ge cp_j$ for constant $c > 1 - 1/e$, then it satisfies $\mathbf{E}[d(j,\mathcal{S})] \le 1 + 2(1 - cp_j) = 3 - 2c = (1 + 2/e - \Omega(1))r_j$. So it will satisfy the first property as well. So it suffices to show that no algorithm $\mathcal{A}$ can satisfy the first property.

Suppose that $\mathcal{A}$ does satisfy the first property. The resulting solution set $\mathcal{S} \subseteq \mathcal{F}$ can be regarded as a solution X to the set cover instance, where $d(j,\mathcal{S}) = 1 + 2[[j \notin S_X]]$. Thus

$$\sum_{j\in[n]} d(j,\mathcal{S}) = |S_X| + 3(n - |S_X|),$$

and so $|S_X| = \frac{3n - \sum_{j\in[n]} d(j,\mathcal{S})}{2}$. As $\mathbf{E}[d(j,\mathcal{S})] \le cr_j = c$ for all j, this implies that $\mathbf{E}[|S_X|] \ge \frac{(3-c)n}{2}$.

After an expected constant number of repetitions of this process we can ensure that $|S_X| \ge c'n$ for some constant $c' > \frac{3-(1+2/e)}{2} = 1 - 1/e$. This contradicts Corollary 19. ■

A slightly more involved construction applies to the homogeneous SCC setting.

Theorem 21 *Assuming $P \neq NP$, there is a family of homogeneous SCC chance k-coverage instances, with a feasible demand of $p_j = r_j = 1$ for all j, such that no polynomial-time algorithm can guarantee a distribution Ω with either of the following:*

1. *$\forall j \mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})] \le cr_j$ for constant $c < 1 + 1/e$*
2. *$\forall j \Pr_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S}) < 2r_j] \ge cp_j$ for constant $c > 1 - 1/e$*

In particular, the approximation constants in Theorem 4 and Proposition 5 cannot be improved for SCC instances, and the approximation factor 1.592 in Pseudo-Theorem 17 cannot be improved below $1 + 1/e$.

Proof Consider a set cover instance $\mathcal{B} = \{S_1, \ldots, S_m\}$, where we have guessed the value $\mathrm{OPT} = k$. We define a k-center instance as follows. For each $i \in [m]$, we create an item v_i and for each $j \in [n]$ we create $t = n^2$ distinct items $w_{j,1}, \ldots, w_{j,t}$. We define the distance by setting $d(v_i, w_{j,t}) = 1$ if $j \in S_i$ and $d(v_i, v_{i'}) = 1$ for all $i, i' \in [m]$, and $d(x,y) = 2$ for all other distances. This problem size is polynomial (in m, n), and so $\mathcal{A}$ runs in time $\mathrm{poly}(m,n)$.

If X is an optimal solution to the set cover instance, the corresponding set $\mathcal{S} = \{v_i \mid i \in X\}$ satisfies $d(j,\mathcal{S}) \le 1$ for all $j \in \mathcal{C}$. So the demand vector $p_j = r_j = 1$ is feasible. Also, note that that $d(j,\mathcal{S}) \in \{0,1,2\}$ with probability one for any j. Thus, if j satisfies the second property $\Pr_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S}) < 2r_j] \ge cp_j$ for constant $c > 1 - 1/e$, then it satisfies $\mathbf{E}[d(j,\mathcal{S})] \le 1 + 1(1 - cp_j) = (1 + 1/e - \Omega(1))r_j$. So it will satisfy the first property as well. So it suffices to show that no algorithm $\mathcal{A}$ can satisfy the first property.

Suppose that algorithm $\mathcal{A}$ satisfies the first property. From the solution set $\mathcal{S}$, we construct a corresponding set-cover solution by $X = \{i \mid v_i \in \mathcal{S}\}$. For $w_{j,\ell} \notin \mathcal{S}$, we can observe that $d(w_{j,\ell}, \mathcal{S}) = 1 + [[j \notin S_X]]$. Therefore, we have

$$\begin{aligned}\sum_{j\in[n]}\sum_{\ell=1}^{t} d(w_{j,\ell},\mathcal{S}) &\ge \sum_{j,\ell: w_{j,\ell}\notin\mathcal{S}} (1 + [[j \notin S_X]]) \ge \sum_{j,\ell}(1 + [[j \notin S_X]]) - 2|\mathcal{S}| \\ &\ge n^2(2n - |S_X|) - 2k,\end{aligned}$$

and so $|S_X| \ge 2n - \frac{\sum_{j,\ell} d(w_{j,\ell},\mathcal{S})}{n^2} - 2k/n^2$.

Taking expectations and using our upper bound on $\mathbf{E}[d(j,\mathcal{S})]$, we have $\mathbf{E}[|S_X|] \ge 2n - cn - 2k/n^2 \ge (2-c)n - 2/n$. Thus, for n sufficiently large, after an expected constant number of repetitions of this process we get $|S_X| \ge (2 - c - o(1))n \ge (1 - 1/e + \Omega(1))n$. This contradicts Corollary 19. ■

5. Approximation Algorithm for $\mathbf{E}[d(j,\mathcal{S})]$

In the chance k-coverage problem, our goal is to achieve certain fixed values of $d(j,\mathcal{S})$ with a certain probability. In this section, we consider another criterion for Ω; we wish to achieve certain values for the expectation $\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})]$. We suppose we are given values t_j for every $j \in C$, such that the target distribution Ω satisfies

$$\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})] \le t_j. \tag{8}$$

In this case, we say that the vector t_j is *feasible*. As before, if all the values of t_j are equal to each other, we say that the instance is *homogeneous*. We show how to leverage any approximation algorithm for k-median with approximation ratio α, to ensure our target distribution $\tilde{\Omega}$ will satisfy

$$\mathbf{E}_{\mathcal{S}\sim\tilde{\Omega}}[d(j,\mathcal{S})] \le (\alpha + \epsilon)t_j.$$

More specifically, we need an approximation algorithm for weighted k-median. In this setting, we have a problem instance $\mathcal{I} = \mathcal{F}, \mathcal{C}, d$ along with non-negative weights w_j for $j \in \mathcal{C}$, and we wish to find $\mathcal{S} \in \binom{\mathcal{F}}{k}$ minimizing $\sum_{j\in\mathcal{C}} w_j d(j,\mathcal{S})$. (Nearly all approximation algorithms for ordinary k-median can be easily adapted to the weighted setting, for example, by replicating clients.) If we fix an approximation algorithm $\mathcal{A}$ for (various classes of) weighted k-median, then for any problem instance $\mathcal{I}$ we define

$$\alpha_{\mathcal{I}} = \sup_{\text{weights } w} \frac{\sum_{j\in\mathcal{C}} w_j d(j, \mathcal{A}(\mathcal{I}, w))}{\min_{\mathcal{S}\in\binom{\mathcal{F}}{k}} \sum_{j\in\mathcal{C}} w_j d(j,\mathcal{S})}.$$

We first show how to use the k-median approximation algorithm to achieve a set $\mathcal{S}$ which "matches" the desired distances t_j:

Proposition 22 *Given a weighted instance $\mathcal{I}$ and a parameter $\epsilon > 0$, there is a polynomial-time algorithm to produce a set $\mathcal{S} \in \binom{\mathcal{F}}{k}$ satisfying:*

1. $\sum_{j\in\mathcal{C}} w_j \frac{d(j,\mathcal{S})}{t_j} \le (\alpha_{\mathcal{I}} + O(\epsilon)) \sum_{j\in\mathcal{C}} w_j$,
2. *Every $j \in \mathcal{C}$ has $d(j,\mathcal{S}) \le n t_j/\epsilon$.*

Proof We assume $\alpha_{\mathcal{I}} = O(1)$, as constant-factor approximation algorithms for k-median exist. By rescaling w, we assume without loss of generality that $\sum_{j\in\mathcal{C}} w_j = 1$. By rescaling ϵ, it suffices to show that $d(j,\mathcal{S}) \le O(n t_j/\epsilon)$.

Let us define the weight vector $z_j = \frac{\epsilon/n + w_j}{t_j}$. Letting Ω be a distribution satisfying (8), we have

$$\mathbf{E}_{\mathcal{S}\sim\Omega}\Big[\sum_{j\in\mathcal{C}} z_j d(j,\mathcal{S})\Big] = \sum_{j\in\mathcal{C}} z_j t_j \le \sum_{j\in\mathcal{C}} \Big(\frac{\epsilon}{n t_j} + \frac{w_j}{t_j}\Big) t_j = \epsilon|\mathcal{C}|/n + \sum_{j\in\mathcal{C}} w_j = 1 + \epsilon.$$

In particular, there exists some $\mathcal{S} \in \binom{\mathcal{F}}{k}$ with $\sum_{j\in\mathcal{C}} z_j d(j,\mathcal{S}) \le 1+\epsilon$. When we apply algorithm $\mathcal{A}$ with weight vector z, we thus get a set $\mathcal{S} \in \binom{\mathcal{F}}{k}$ with $\sum_{j\in\mathcal{C}} z_j d(j,\mathcal{S}) \le \alpha_{\mathcal{I}}(1+\epsilon)$. We claim that this set $\mathcal{S}$ satisfies the two conditions of the theorem. First, we have

$$\sum_{j\in\mathcal{C}} \frac{w_j d(j,\mathcal{S})}{t_j} \le \sum_{j\in\mathcal{C}} z_j d(j,\mathcal{S}) \le \alpha_{\mathcal{I}}(1+\epsilon) \le (\alpha_{\mathcal{I}} + O(\epsilon)) \sum_j w_j.$$

Next, for any given $j \in \mathcal{C}$, we have

$$\frac{d(j,\mathcal{S})}{t_j} \le d(j,\mathcal{S}) z_j (n/\epsilon) \le (n/\epsilon) \sum_{w\in\mathcal{C}} z_w d(w,\mathcal{S}) \le (n/\epsilon)\alpha_{\mathcal{I}}(1+\epsilon) \le O(n/\epsilon). \qquad \blacksquare$$

Theorem 23 *There is an algorithm which takes as input an instance $\mathcal{I}$, a parameter $\epsilon > 0$ and a feasible vector t_j, runs in time poly$(n, 1/\epsilon)$, and returns an explicitly enumerated distribution $\tilde{\Omega}$ with support size n and $\mathbf{E}_{S\sim\tilde{\Omega}}[d(j,\mathcal{S})] \le (\alpha_{\mathcal{I}} + \epsilon) t_j$ for all $j \in \mathcal{C}$.*

Proof We assume without loss of generality that $\epsilon \le 1$; by rescaling ϵ it suffices to show that $\mathbf{E}[d(j,\mathcal{S})] \le (\alpha_{\mathcal{I}} + O(\epsilon)) t_j$.

We begin with the following Algorithm 8, which uses a form of multiplicative weights update with repeated applications of Proposition 22.

Algorithm 8 Approximation algorithm for $\mathbf{E}[d(j,\mathcal{S})]$: first phase

1: **for** $\ell = 1, \ldots, r = \frac{n \ln n}{\epsilon^3}$ **do**

2: Let $X_\ell \in \binom{\mathcal{F}}{k}$ be the resulting of applying the algorithm of Proposition 22 with parameters ϵ, t_j and weight vector w given by

$$w_j = \exp\Big(\epsilon^2 \sum_{s=1}^{\ell-1} \frac{d(j, X_s)}{n t_j}\Big),$$

3: Set $\tilde{\Omega}'$ to be the uniform distribution on $X_1, \ldots, X_r$

Let us define $\phi = \epsilon^2/n$. For each iteration $\ell = 1, \ldots, r+1$ let $u_j^{(\ell)} = \phi d(j, X_\ell)/t_j$, and let $w_j^{(\ell)} = e^{\sum_{s=1}^{\ell-1} u_j^{(s)}}$ denote the weight vector. Proposition 22 ensures that $u_j^{(\ell)} \le \epsilon$, and thus $e^{u_j^{(\ell)}} \le 1 + \frac{e^\epsilon - 1}{\epsilon} u_j^{(\ell)} \le 1 + (1+\epsilon) u_j^{(\ell)}$, as well as ensuring that $\sum_j w_j^{(\ell)} u_j^{(\ell)} \le \phi(\alpha_I + O(\epsilon)) \sum_j w_j^{(\ell)}$.

Now let $\Phi_\ell = \sum_{j\in\mathcal{C}} w_j^{(\ell)}$. Note that $\Phi_1 = n$, and for each $\ell \ge 1$, we have

$$\begin{aligned}\Phi_{\ell+1} &= \sum_{j\in\mathcal{C}} w_j^{(\ell)} e^{u_j^{(\ell)}} \le \sum_{j\in\mathcal{C}} w_j^{(\ell)}\big(1 + (1+\epsilon)u_j^{(\ell)}\big) \le \sum_{j\in\mathcal{C}} w_j^{(\ell)} + (1+\epsilon)\sum_{j\in\mathcal{C}} w_j^{(\ell)} u_j^{(\ell)} \\ &\le \Phi_\ell\big(1 + (1+\epsilon)\phi(\alpha_{\mathcal{I}} + O(\epsilon))\big) \le \Phi_\ell e^{\phi(\alpha_{\mathcal{I}} + O(\epsilon))}\end{aligned}$$

This recurrence relation implies that $\Phi_\ell \le n e^{(\ell-1)\phi(\alpha_{\mathcal{I}} + O(\epsilon))}$. Since $w_j^{(r+1)} \le \Phi^{r+1}$, this implies

$$\sum_{\ell=1}^{r} \phi d(j, X_\ell)/t_j = \ln w_j^{(r+1)} \le \ln \Phi_{r+1} \le \ln n + r\phi(\alpha_{\mathcal{I}} + O(\epsilon)).$$

or equivalently,

$$\sum_{\ell=1}^{r} \frac{d(j, X_\ell)}{r} = t_j\Big(\frac{\ln n}{r\phi} + (\alpha_{\mathcal{I}} + O(\epsilon))\Big)$$

As $r = \frac{n \ln n}{\epsilon^3} = \frac{\ln n}{\epsilon\phi}$, we thus have $\frac{\sum_{\ell=1}^{r} d(j,X_\ell)}{r} \le (\alpha_{\mathcal{I}} + O(\epsilon)) t_j$. Thus, the distribution $\tilde{\Omega}'$ satisfies

$$\forall j \in \mathcal{C} \qquad \mathbf{E}_{\mathcal{S}\sim\tilde{\Omega}'}[d(j,\mathcal{S})] \le (\alpha_I + O(\epsilon)) t_j. \tag{9}$$

Now the distribution $\tilde{\Omega}'$ satisfies the condition on $\mathbf{E}[d(j,\mathcal{S})]$, but its support is too large. We can reduce the support size to $|\mathcal{C}|$ by moving in the nullspace of the $|\mathcal{C}|$ linear constraints (9). $\blacksquare$

Byrka et al. (2017) have shown a $2.675+\epsilon$-approximation algorithm for k-median, which automatically gives a $2.675+\epsilon$-approximation algorithm for k-lottery as well. Some special

cases of k-median have more efficient approximation algorithms. For instance, Cohen-Addad et al. (2016) gives a PTAS for k-median problems derived from a planar graph, and Ahmadian et al. (2017) gives a $2.633+\epsilon$-approximation for Euclidan distances. These immediately give approximation algorithms for the corresponding k-lotteries. We also note that, by Theorem 20, one cannot obtain a general approximation ratio better than $1+2/e$ (or $1+1/e$ in the SCC setting).

6. Determinizing a k-lottery

Suppose that we have a set of feasible weights t_j such some k-lottery distribution Ω satisfies $\mathbf{E}_{\mathcal{S}\sim\Omega}[d(j,\mathcal{S})] \le t_j$; let us examine how to find a *single, deterministic* set $\mathcal{S}$ with $d(j,\mathcal{S}) \approx t_j$. We refer to this as the problem of *determinizing* the lottery Ω. Note that this can be viewed as a converse to the problem considered in Section 3.

We will see that, in order to obtain reasonable approximation ratios, we may need $|\mathcal{S}|$ to be significantly larger than k. We thus define an (α,β)-*determinization* to be a set $\mathcal{S} \in \binom{\mathcal{F}}{k'}$ with $k' \le \alpha k$ and $d(j,\mathcal{S}) \le \beta t_j$ for all $j \in \mathcal{C}$. We emphasize that we cannot necessarily obtain $(1,1)$-determinizations, even with unbounded computational resources. The following simple example illustrates the tradeoff between parameters α and β:

Observation 24 *Let $\alpha,\beta,k \ge 1$. If $\beta < \frac{\alpha k+1}{(\alpha-1)k+1}$, there is a homogeneous SCC instance for which no (α,β)-determinization exists.*

Proof Let $k' = \alpha k$ and consider a problem instance with $\mathcal{F} = \mathcal{C} = \{1,\dots,k'+1\}$, and $d(i,j) = 1$ for every distinct i,j. Clearly, every $\mathcal{S} \in \binom{\mathcal{F}}{k'}$ satisfies $\min_j d(j,\mathcal{S}) = 1$. When Ω is the uniform distribution on $\binom{\mathcal{F}}{k}$, we have $\mathbf{E}[d(j,\mathcal{S})] = 1 - \frac{k}{k'+1}$. Thus $t_j = \frac{k}{k'+1}$ is feasible and therefore $\beta \ge \frac{1}{1-\frac{k}{k'+1}} = \frac{\alpha k+1}{(\alpha-1)k+1}$. ■

In particular, when $\alpha = 1$ we must have $\beta \ge k+1$ and when $k \to \infty$, we must have $\beta \gtrsim \frac{\alpha}{\alpha-1}$.

We examine three main regimes for the parameters (α,β): (1) the case where α,β are scale-free constants; (2) the case where β is close to one, in which case α must be of order $\log n$; (3) the case where $\alpha = 1$, in which case β must be order k.

Our determinization algorithms for the first two cases will based on the following LP denoted $\mathcal{P}_{\text{expectation}}$, defined in terms of fractional vectors $b_i, a_{i,j}$ where i ranges over $\mathcal{F}$ and j ranges over $\mathcal{C}$:

(A1) $\forall j \in \mathcal{C}, \quad \sum_{i\in\mathcal{F}} a_{i,j} d(i,j) \le t_j,$

(A2) $\forall j \in \mathcal{C}, \quad \sum_{i\in\mathcal{F}} a_{i,j} = 1,$

(A3) $\forall i \in \mathcal{F}, y \in \mathcal{C}, \quad 0 \le a_{i,j} \le b_i,$

(A4) $\forall i \in \mathcal{F}, \quad 0 \le b_i \le 1,$

(A5) $\sum_{i\in\mathcal{F}} b_i \le k.$

Theorem 25 *If t_j is feasible, then $\mathcal{P}_{expectation}$ has a fractional solution.*

Proof Let Ω be a probability distribution with $\mathbf{E}[d(j,\mathcal{S})] \le t_j$. For any draw $\mathcal{S} \sim \Omega$, define random variable Z_j to be the facility of $\mathcal{S}$ matched by j. Now consider the fractional vector defined by

$$b_i = \Pr_{\mathcal{S}\sim\Omega}[i \in \mathcal{S}], \qquad a_{i,j} = \Pr_{\mathcal{S}\sim\Omega}[Z_j = i]$$

We claim that this satisfies (A1) — (A5). For (A1), we have

$$\mathbf{E}[d(j,\mathcal{S})] = \mathbf{E}[d(j,Z_j)] = \sum_{i\in\mathcal{F}} d(i,j)\Pr[Z_j = i] = \sum_{i\in F} d(i,j)a_{i,j} \le t_j.$$

For (A2), note that $\sum_i \Pr[Z_j = i] = 1$. For (A3), note that $Z_j = i$ can only occur if $i \in \mathcal{S}$. (A4) is clear, and (A5) holds as $|\mathcal{S}| = k$ with probability one. ■

We next describe upper and lower bounds for these three regimes.

6.1. The Case Where α,β are Scale-free Constants.

For this regime (with all parameters independent of problem size n and k), we may use the following Algorithm 9, which is based on greedy clustering using a solution to $\mathcal{P}_{\text{expectation}}$.

Algorithm 9 (α,β)-determinization algorithm

1: Let a,b be a solution to $\mathcal{P}_{\text{expectation}}$.
2: For every $j \in \mathcal{C}$, select $r_j \ge 0$ to be minimal such that $\sum_{i\in B(j,r_j)} a_{i,j} \ge 1/\alpha$
3: By splitting facilities, form a set $F_j \subseteq B(j,r_j)$ with $b(F_j) = 1/\alpha$.
4: Set $C' = \textsc{GreedyCluster}(F_j, \theta(j)+r_j)$
5: Output solution set $\mathcal{S} = \{V_j \mid j \in C'\}$.

Step (3) is well-defined, as (A3) ensures that $b(B(j,r_j)) \ge \sum_{i\in B(j,r_j)} a_{i,j} \ge 1/\alpha$. Let us analyze the resulting approximation factor β.

Proposition 26 *Every client $j \in \mathcal{C}$ has $r_j \le \frac{\alpha t_j - \theta(j)}{\alpha-1}$.*

Proof Let $s = \frac{\alpha t_j - \theta(j)}{\alpha-1}$. It suffices to show that

$$\sum_{i\in\mathcal{F}, d(i,j)>s} a_{i,j} \le 1 - 1/\alpha.$$

As $d(i,j) \ge \theta(j)$ for all $i \in \mathcal{F}$, we have

$$\sum_{\substack{i\in\mathcal{F}\\ d(i,j)>s}} a_{i,j} \le \sum_{\substack{i\in\mathcal{F}\\ d(i,j)>s}} a_{i,j}\frac{d(i,j)-\theta(j)}{s-\theta(j)} \le \sum_{i\in\mathcal{F}} a_{i,j}\frac{d(i,j)-\theta(j)}{s-\theta(j)} = \frac{\sum_{i\in\mathcal{F}} a_{i,j}d(i,j) - \theta(j)\sum_{i\in\mathcal{F}} a_{i,j}}{s-\theta(j)}$$
$$\le \frac{t_j-\theta(j)}{s-\theta(j)} = 1 - 1/\alpha, \qquad \text{by (A1), (A2).}$$

■

Theorem 27 *Algorithm 9 gives an (α, β)-determinization with the following parameter β:*

1. *In the general setting, $\beta = \max(3, \frac{2\alpha}{\alpha-1})$.*
2. *In the SCC setting, $\beta = \frac{2\alpha}{\alpha-1}$.*

Proof We first claim that the resulting set $\mathcal{S}$ has $|\mathcal{S}| \le \alpha k$. The algorithm opens at most $|C'|$ facilities. The sets F_j are pairwise disjoint for $j \in C'$ and $b(F_j) = 1/\alpha$ for $j \in C'$. Thus $\sum_{j \in C'} b(F_j) = |C'|/\alpha$. On the other hand, $b(\mathcal{F}) = k$, and so $k \ge |C'|/\alpha$.

Next, consider some $j \in \mathcal{C}$; we want to show that $d(j, \mathcal{S}) \le \beta t_j$. By Observation 2, there is $z \in C'$ with $F_j \cap F_z \ne \emptyset$ and $\theta(z) + r_z \le \theta(j) + r_j$. Thus $d(j, \mathcal{S}) \le d(z, \mathcal{S}) + d(z, i) + d(j, i)$ where $i \in F_j \cap F_z$. Step (5) ensures $d(z, \mathcal{S}) = \theta(z)$. We have $d(z, i) \le r_z$ and $d(i, j) \le r_j$ since $i \in F_j \subseteq B(j, r_j)$ and $i \in F_z \subseteq B(z, r_z)$. So

$$d(j, \mathcal{S}) \le \theta(z) + r_z + r_j \le 2r_j + \theta(j).$$

By Proposition 26, we therefore have

$$d(j, \mathcal{S}) \le \frac{2\alpha t_j - 2\theta(j)}{\alpha - 1} + \theta(j) = \frac{2\alpha t_j}{\alpha - 1} + \frac{\alpha - 3}{\alpha - 1}\theta(j) \tag{10}$$

This immediately shows the claim for the SCC setting where $\theta(j) = 0$.

In the general setting, for $\alpha \le 3$, the second coefficient in the RHS of (10) is non-positive and hence the RHS is at most $\frac{2\alpha t_j}{\alpha-1}$ as desired. When $\alpha \ge 3$, then in order for t to be feasible we must have $t_j \ge \theta(j)$; substituting this upper bound on $\theta(j)$ into (10) gives

$$d(j, \mathcal{S}) \le \frac{2\alpha t_j}{\alpha - 1} + \frac{\alpha - 3}{\alpha - 1} t_j = 3t_j$$

■

We note that these approximation ratios are, for α close to 1, within a factor of 2 compared to the lower bound of Observation 24. As $\alpha \to \infty$, the approximation ratio approaches to limiting values 3 (or 2 in the SCC setting).

6.2. The Case of Small β

We now consider what occurs when β becomes smaller than the critical threshold values 3 (or 2 in the SCC setting). We show that in this regime we must take $\alpha = \Omega(\log n)$. Of particular interest is the case when β approaches 1; here, in order to get $\beta = 1 + \epsilon$ for small ϵ we show it is necessary and sufficient to take $\alpha = \Theta(\frac{\log n}{\epsilon})$.

Proposition 28 *For any $\epsilon < 1/2$, there is a randomized polynomial-time algorithm to obtain a $(\frac{3 \log n}{\epsilon}, 1 + \epsilon)$ determinization.*

Proof First, let a, b be a solution to $\mathcal{P}_{\text{expectation}}$. Define $p_i = \min(1, \frac{2 \log n}{\epsilon} b_i)$ for each $i \in \mathcal{F}$ and form $\mathcal{S} = \textsc{DepRound}(p)$. Observe then that $|\mathcal{S}| \le \lceil \sum_i p_i \rceil \le \lceil \frac{2 \log n}{\epsilon} \sum b_i \rceil \le 1 + \frac{2k \log n}{\epsilon} \le \frac{3k \log n}{\epsilon}$.

For $j \in \mathcal{C}$, define $A = B_{j,(1+\epsilon)t_j}$. Let us note that, by properties (A3), (A1) and (A2), we have

$$\sum_{i \in A} b_i \ge \sum_{i \in A} a_{i,j} = 1 - \sum_{i : d(i,j) > (1+\epsilon)t_j} a_{i,j} \ge 1 - \sum_{i : d(i,j) > (1+\epsilon)t_j} a_{i,j} \frac{d(i,j)}{(1+\epsilon)t_j} \ge 1 - \frac{1}{1+\epsilon}$$

So by property (P3) of DepRound, and using the bound $\epsilon < 1/2$, have

$$\Pr[d(j, \mathcal{S}) > (1+\epsilon)t_j] = \Pr[A \cap \mathcal{S} = \emptyset] \le \prod_{i \in A} (1 - p_i) \le \prod_{i \in A} e^{-\frac{2 \log n}{\epsilon} b_i} \le e^{\frac{-2 \log n}{\epsilon}(1 - \frac{1}{1+\epsilon})} \le n^{-4/3}$$

A union bound over $j \in \mathcal{C}$ shows that solution set $\mathcal{S}$ satisfies $d(j, \mathcal{S}) \le (1+\epsilon)t_j$ for all j with high probability. ■

The following shows matching lower bounds:

Proposition 29

1. *There is a universal constant K with the following properties. For any $k \ge 1, \epsilon \in (0, 1/3)$ there is some integer $N_{k,\epsilon}$ such that for $n > N_{k,\epsilon}$, there is a homogeneous SCC instance of size n in which* every *$(\alpha, 1+\epsilon)$-determinization satisfies $\alpha \ge \frac{K \log n}{\epsilon}$.*
2. *For each $\beta \in (1, 2)$ and each $k \ge 1$, there is a constant $K'_{\beta,k}$ such that, for all $n \ge 1$, there is a homogeneous SCC instance of size n in which* every *(α, β)-determinization satisfies $\alpha \ge K'_{\beta,k} \log n$.*
3. *For each $\beta \in (1, 3)$ and each $k \ge 1$, there is a constant $K''_{\beta,k}$ such that, for all $n \ge 1$, there is a homogeneous instance of size n in which* every *(α, β)-determinization satisfies $\alpha \ge K''_{\beta,k} \log n$*

Proof These three results are very similar, so we show the first one in detail and sketch the difference between the other two.

Consider an Erdős-Rényi random graph $G \sim \mathcal{G}(n, p)$, where $p = 3\epsilon/k$; note that $p \in (0, 1)$. As shown by Glebov et al. (2015) asymptotically almost surely the domination number J of G satisfies $J = \Omega(\frac{k \log n}{\epsilon})$.

We construct a related instance with $\mathcal{F} = \mathcal{C} = [n]$, and where $d(i, j) = 1$ if (i, j) is an edge, and $d(i, j) = 2$ otherwise. Note that if X is not a dominating set of G, then some vertex of G has distance at least 2 from it; equivalently, $\max_j d(j, X) \ge 2$ for every set X with $|X| < J$.

Chernoff's bound shows that every vertex of G has degree at least $u = 0.9np$ with high probability. Assuming this event has occured, we calculate $\mathbf{E}[d(j, \mathcal{S})]$ where $\mathcal{S}$ is drawn from the uniform distribution on $\binom{\mathcal{F}}{k}$. Note that $d(j, \mathcal{S}) \le 1$ if j is a neighbor of X and $d(j, \mathcal{S}) = 2$ otherwise, so

$$\mathbf{E}[d(j, \mathcal{S})] \le 1 + \frac{\binom{n-u}{k}}{\binom{n}{k}} \le 1 + e^{-0.9pk} = 1 + e^{-2.7\epsilon}.$$

Both the bound on the domination number and the minimum degree of G hold with positive probability for n sufficiently large (as a function of k, ϵ). In this case, $t_j = 1 + e^{-2.7\epsilon}$

is a feasible homogeneous demand vector. At the same time, every set $\mathcal{S} \in \binom{F}{J-1}$ satisfies $\min_{j \in \mathcal{C}} d(j, \mathcal{S}) \geq 2$. Thus, an (α, β)-determinization cannot have $\alpha < \frac{J}{k} = \Theta(\frac{\log n}{\epsilon})$ and $\beta \leq \frac{2}{1+e^{-2.7\epsilon}}$. Note that $\frac{2}{1+e^{-2.7\epsilon}} \geq 1 + \epsilon$ for $\epsilon < 1/3$. Thus, whenever $\beta \leq 1 + \epsilon$, we have $\alpha \geq \Theta(\frac{\log n}{\epsilon})$.

For the second result, we use the same construction as above with $p = 1 - \frac{1}{2}(\lambda/2)^{1/k}$ where $\lambda = 2 - \beta$. A similar analysis shows that the vector $t_j = 1 + \lambda/2$ is feasible with high probability and $|J| \geq \Omega(k \log n)$ (where the hidden constant may depend upon β, k). Thus, unless $\alpha \geq \Omega(\log n)$, the approximation ratio achieved is $\frac{2}{1+\lambda/2} \geq \beta$.

The third result is similar to the second one, except that we use a random bipartite graph. The left-nodes are associated with $\mathcal{F}$ and the right-nodes with $\mathcal{C}$. For $i \in \mathcal{F}$ and $j \in \mathcal{C}$, we define $d(i, j) = 1$ if (i, j) is an edge and $d(i, j) = 3$ otherwise. ■

6.3. The Case of $\alpha = 1$

We finally consider the case $\alpha = 1$, that is, where the constraint on the number of open facilities is respected *exactly*. By Observation 24, we must have $\beta \geq k + 1$ here. The following greedy algorithm gives a $(1, k + 2)$-determinization, nearly matching this lower bound.

Algorithm 10 $(1, k + 2)$-determinization algorithm

```
Initialize S = ∅
for ℓ = 1, ..., |F| do
    Let C_ℓ denote the set of points j ∈ C with d(j, S) > (k + 2)t_j
    If C_ℓ = ∅, then return S.
    Select the point j_ℓ ∈ C_ℓ with the smallest value of t_{j_ℓ}.
    Update S ← S ∪ {V_{j_ℓ}}
```

Theorem 30 *If the values t_j are feasible, then Algorithm 10 outputs a $(1, k+2)$-determinization in $O(|\mathcal{F}||\mathcal{C}|)$ time.*

Proof For the runtime bound, we first compute V_j for each $j \in \mathcal{C}$; this requires $O(|\mathcal{F}||\mathcal{C}|)$ time upfront. When we update $\mathcal{S}$ at each iteration ℓ, we update and maintain the quantities $d(j, \mathcal{S})$ quantities by computing $d(j, V_{j_\ell})$ for each $j \in \mathcal{C}$. This takes $O(|\mathcal{C}|)$ time per iteration.

To show correctness, note that if this procedure terminates at iteration ℓ, we have $\mathcal{C}_\ell = \emptyset$ and so every point $j \in \mathcal{C}$ has $d(j, \mathcal{S}) \leq (k + 2)t_j$. The resulting set $\mathcal{S}$ at this point has cardinality $\ell - 1$. So we need to show that the algorithm terminates before reaching iteration $\ell = k + 2$.

Suppose not; let the resulting points be $j_1, \ldots, j_{k+1}$ and for each $\ell = 1, \ldots, k + 1$ let $w_\ell = t_{j_\ell}$. Because j_ℓ is selected to minimze t_{j_ℓ} we have $w_1 \leq w_2 \leq \cdots \leq w_{k+1}$.

Now, let Ω be a k-lottery satisfying $\mathbf{E}_{\mathcal{S} \sim \Omega}[d(j, \mathcal{S})] \leq t_j$ for every $j \in \mathcal{C}$, and consider the random process of drawing $\mathcal{S}$ from Ω. Define the random variable $D_\ell = d(j_\ell, \mathcal{S})$ for $\ell = 1, \ldots, k + 1$. For any such $\mathcal{S}$, by the pigeonhole principle there must exist some pair $j_\ell, j_{\ell'}$ with $1 \leq \ell < \ell' \leq k + 1$ which are both matched to a common facility $i \in \mathcal{S}$, that is

$$D_\ell = d(j_\ell, \mathcal{S}) = d(j_\ell, i), D_{\ell'} = d(j_{\ell'}, \mathcal{S}) = d(j_{\ell'}, i).$$

By the triangle inequality,

$$d(j_{\ell'}, V_{j_\ell}) \leq d(j_{\ell'}, i) + d(i, j_\ell) + d(j_\ell, V_{j_\ell}) = D_{\ell'} + D_\ell + \theta(j_\ell)$$

On the other hand, $j_{\ell'} \in C_{\ell'}$ and yet V_{j_ℓ} was in the partial solution set $\mathcal{S}$ seen at iteration ℓ'. Therefore, it must be that

$$d(j_{\ell'}, V_{j_\ell}) > (k + 2)t_{j_{\ell'}} = (k + 2)w_{\ell'}$$

Putting these two inequalities together, we have shown that

$$D_\ell + D_{\ell'} + \theta(j_\ell) > (k + 2)w_{\ell'}.$$

As $\theta(j_\ell) \leq w_\ell \leq w_{\ell'}$, this implies that

$$\frac{D_\ell}{w_\ell} + \frac{D_{\ell'}}{w_{\ell'}} \geq \frac{D_\ell + D_{\ell'}}{w_{\ell'}} > \frac{(k + 2)w_{\ell'} - \theta(j_\ell)}{w_{\ell'}} \geq \frac{(k + 2)w_{\ell'} - w_\ell}{w_{\ell'}} \geq \frac{(k + 2)w_{\ell'} - w_{\ell'}}{w_{\ell'}} = k + 1.$$

We have shown that, with probability one, there is some pair $\ell < \ell'$ satisfying this inequality $D_\ell/w_\ell + D_{\ell'}/w_{\ell'} > k + 1$. Therefore, with probability one it holds that

$$\sum_{\ell=1}^{k+1} D_\ell/w_\ell > k + 1. \tag{11}$$

But now take expectations, observing that $\mathbf{E}[D_\ell] = \mathbf{E}[d(j_\ell, \mathcal{S})] \leq t_{j_\ell} = w_\ell$. So the LHS of (11) has expectation at most $k + 1$. This is a contradiction. ■

We remark that it is possible to obtain an optimal $(1, k+1)$-determinization algorithm for the SCC or homogeneous settings, but we omit this since it is very similar to Algorithm 10.

7. Acknowledgments

Our sincere thanks to Brian Brubach and to the anonymous referees, for many useful suggestions and for helping to tighten the focus of the paper.

Appendix A. Proof of Pseudo-Theorem 17

We would like to use Lemma 16 to bound $\mathbf{E}[d(i, \mathcal{S})]$, over all possible integer values $m \geq 1$ and over all possible sequences $1 \geq u_1 \geq u_2 \geq u_3 \geq \cdots \geq u_t \geq 0$. One technical obstacle here is that this is not a compact space, due to the unbounded dimension t and unbounded parameter m. The next result removes these restrictions.

Proposition 31 *For any fixed integers $L, M \geq 1$, and every $j \in \mathcal{C}$, we have*

$$\mathbf{E}[d(j, \mathcal{S})] \leq 1 + \max_{\substack{m \in \{1,2,\ldots,M\} \\ 1 \geq u_1 \geq u_2 \geq \ldots u_L \geq 0}} \mathbf{E}_Q \hat{R}(m, u_1, u_2, \ldots, u_L),$$

where we define

$$\alpha = \prod_{\ell=1}^{L-1}(1-\overline{Q_p}(u_\ell - u_{\ell+1})) \times e^{-\overline{Q_p}u_L},$$

$$\beta = \prod_{\ell=1}^{L-1}(\overline{u_\ell} + \overline{Q_p}u_{\ell+1}) \times \begin{cases}(1-u_L) & \text{if } u_L \le Q_p \\ e^{-\frac{u_L - Q_p}{1-Q_p}}(1-Q_p) & \text{if } u_L > Q_p\end{cases},$$

$$\hat{R}(m, u_1, \ldots, u_L) = \begin{cases}(1-\overline{Q_f}\frac{\overline{u_1}}{m})^m\alpha + (\overline{Q_f}(1-\frac{\overline{u_1}}{m}))^m\beta & \text{if } m < M \\ e^{-\overline{Q_f}\overline{u_1}}\alpha + \overline{Q_f}^M e^{-\overline{u_1}}\beta & \text{if } m = M\end{cases}.$$

The expectation $\mathbf{E}_Q$ *is taken only over the randomness involved in* Q_f, Q_p.

Proof By Lemma 16,

$$\mathbf{E}[d(i,\mathcal{S}) \mid Q_\mathrm{f}, Q_\mathrm{p}]$$
$$\le 1 + \left(1 - \overline{Q_\mathrm{f}}\frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^{t}(1-\overline{Q_\mathrm{p}}(u_\ell - u_{\ell+1})) + \left(\overline{Q_\mathrm{f}}(1-\frac{\overline{u_1}}{m})\right)^m \prod_{\ell=1}^{t}(\overline{u_\ell} + \overline{Q_\mathrm{p}}u_{\ell+1}).$$

where $u_1, \ldots, u_t, m$ are defined as in Lemma 16; in particular $1 \ge u_1 \ge u_2 \ge \cdots \ge u_t \ge u_{t+1} = 0$ and $m \ge 1$. If we define $u_j = 0$ for all integers $j \ge t$, then

$$\mathbf{E}[d(i,\mathcal{S}) \mid Q_\mathrm{f}, Q_\mathrm{p}]$$
$$\le 1 + \left(1 - \overline{Q_\mathrm{f}}\frac{\overline{u_1}}{m}\right)^m \prod_{\ell=1}^{\infty}(1-\overline{Q_\mathrm{p}}(u_\ell - u_{\ell+1})) + \left(\overline{Q_\mathrm{f}}(1-\frac{\overline{u_1}}{m})\right)^m \prod_{\ell=1}^{\infty}(\overline{u_\ell} + \overline{Q_\mathrm{p}}u_{\ell+1}).$$

The terms corresponding to $\ell > L$ telescope so we estimate these as:

$$\prod_{\ell=L}^{\infty}(1-\overline{Q_\mathrm{p}}(u_\ell - u_{\ell+1})) \le \prod_{\ell=L}^{\infty} e^{-\overline{Q_\mathrm{p}}(u_\ell - u_{\ell+1})} = e^{-\overline{Q_\mathrm{p}}u_L}$$

and

$$\prod_{\ell=L}^{\infty}(\overline{u_\ell} + \overline{Q_\mathrm{p}}u_{\ell+1}) \le (1 - u_L + \overline{Q_\mathrm{p}}u_{L+1}) \prod_{\ell=L+1}^{\infty} e^{-u_\ell + \overline{Q_\mathrm{p}}u_{\ell+1}} \le (1 - u_L + \overline{Q_\mathrm{p}}u_{L+1})e^{-u_{L+1}}.$$

Now consider the expression $(1-u_L+\overline{Q_\mathrm{p}}u_{L+1})e^{-u_{L+1}}$ as a function of u_{L+1} in the range $u_{L+1} \in [0, u_L]$. Elementary calculus shows that it satisfies the bound

$$(1 - u_L + \overline{Q_\mathrm{p}}u_{L+1})e^{-u_{L+1}} \le \begin{cases}(1-u_L) & \text{if } u_L \le Q_\mathrm{p} \\ e^{-\frac{u_L - Q_\mathrm{p}}{1-Q_\mathrm{p}}}(1-Q_\mathrm{p}) & \text{if } u_L > Q_\mathrm{p}\end{cases},$$

Thus

$$\mathbf{E}[d(i,\mathcal{S}) \mid Q_\mathrm{f}, Q_\mathrm{f}] \le 1 + \left(1 - \overline{Q_\mathrm{f}}\frac{\overline{u_1}}{m}\right)^m \alpha + \left(\overline{Q_\mathrm{f}}(1-\frac{\overline{u_1}}{m})\right)^m \beta.$$

If $m < M$ we are done. Otherwise, for $m \ge M$, we upper-bound the Q_f terms as:

$$(1 - \overline{Q_\mathrm{f}}\overline{u_1}/m)^m \le e^{-\overline{Q_\mathrm{f}}\overline{u_1}}, \qquad (\overline{Q_\mathrm{f}}(1-\overline{u_1}/m))^m \le \overline{Q_\mathrm{f}}^M e^{-\overline{u_1}}$$ ■

In light of Proposition 31, we can get an upper bound on $\mathbf{E}[d(i,\mathcal{S})]$ by maximizing $\hat{R}$. We now discuss to bound $\hat{R}$ for a fixed choice of L, M, where we select $Q_\mathrm{f}, Q_\mathrm{p}$ according to the following type of distribution:

$$(Q_\mathrm{f}, Q_\mathrm{p}) = \begin{cases}(\gamma_{0,\mathrm{f}}, 0) & \text{with probability } p \\ (\gamma_{1,\mathrm{f}}, \gamma_{1,\mathrm{p}}) & \text{with probability } 1-p\end{cases}.$$

Note that we are setting $\gamma_{0,\mathrm{p}} = 0$ here in order to keep the search space more manageable. We need to upper-bound the function $\mathbf{E}_Q\hat{R}(m, u_1, \ldots, u_L)$, for fixed values γ and where m, u range over the compact domain $m \in \{1, \ldots, M\}, 1 \ge u_1 \ge \cdots \ge u_L \ge 0$. The most straightforward way to do so would be to divide $(u_1, \ldots, u_L)$ into intervals of size ϵ, and then calculate upper bounds on $\hat{R}$ within each interval and for each m. This process would have a runtime of $M\epsilon^{-L}$, which is too expensive.

But suppose we have fixed $u_j, \ldots, u_L$, and we wish to continue to enumerate over $u_1, \ldots, u_{j-1}$. To compute $\hat{R}(m, u_1, \ldots, u_L)$ as a function of $m, u_1, \ldots, u_L$, observe that we do not need to know all the values $u_{j+1}, \ldots, u_L$, but only the following four summary statistics of them:

1. $a_1 = e^{-\overline{\gamma_{1,\mathrm{p}}}u_L}\prod_{\ell=j}^{L-1}(1-\overline{\gamma_{1,\mathrm{p}}}(u_\ell - u_{\ell+1}))$,
2. $a_2 = \prod_{\ell=j}^{L-1}(\overline{u_\ell} + \overline{\gamma_{1,\mathrm{p}}}u_{\ell+1}) \times \begin{cases}(1-u_L) & \text{if } u_L \le \gamma_{1,\mathrm{p}} \\ e^{-\frac{u_L - \gamma_{1,\mathrm{p}}}{1-\gamma_{1,\mathrm{p}}}}(1-\gamma_{1,\mathrm{p}}) & \text{if } u_L > \gamma_{1,\mathrm{p}}\end{cases}$,
3. $a_3 = e^{-u_L}\prod_{\ell=j}^{L-1}(1-(u_\ell - u_{\ell+1}))$,
4. $a_4 = u_{j+1}$.

We thus use a dynamic program wherein we track, for $j = L, \ldots, 1$, all possible values for the tuple $(a_1, \ldots, a_4)$. Furthermore, since $\hat{R}$ is a monotonic function of a_1, a_2, a_3, a_4, we only need to store the *maximal* tuples $(a_1, \ldots, a_4)$. The resulting search space has size $O(\epsilon^{-3})$.

We wrote C code to perform this computation to upper-bound $\mathbf{E}_Q\hat{R}$ with $M = 10, \epsilon = 2^{-12}, L = 7$. This runs in about an hour on a single CPU core. With some additional tricks, we can also optimize over the parameter $p \in [0, 1]$ while still keeping the stack space bounded by $O(\epsilon^{-3})$.

Note that due to the complexity of the dynamic programming algorithm, the computations were carried out in double-precision floating point arithmetic. The rounding errors were not tracked precisely, and it would be difficult to write completely correct code to do so. We believe that these errors should be orders of magnitude below the third decimal place, and that the computed value 1.592 is a valid upper bound.

References

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. In *Proc. 58th annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 61–72, 2017.

Sharareh Alipour and Amir Jafari. Improvements on the k-center problem for uncertain data. In *Proc. 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principle of Database System (PODS)*, pages 425–433, 2018.

Ian Ayres, Mahzarin Banaji, and Christine Jolls. Race effects on eBay. *The RAND Journal of Economics*, 46(4):891–917, 2015. ISSN 1756-2171. doi: 10.1111/1756-2171.12115. URL `http://dx.doi.org/10.1111/1756-2171.12115`.

Emily Badger. How Airbnb plans to fix its racial-bias problem. *Washington Post*, 2016. URL `https://www.washingtonpost.com/news/wonk/wp/2016/09/08/how-airbnb-plans-to-fix-its-racial-bias-problem/`. September 8, 2016.

Marianne Bertrand and Sendhil Mullainathan. Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. *American Economic Review*, 94(4):991–1013, 2004. doi: 10.1257/0002828042002561. URL `http://www.aeaweb.org/articles.php?doi=10.1257/0002828042002561`.

Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. *ACM Transaction on Algorithms*, 13(2):23:1–23:31, 2017.

Moses Charikar and Shi Li. A dependent LP-rounding approach for the k-median problem. *Automata, Languages, and Programming*, pages 194–205, 2012.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems 30*, pages 5036–5044, 2017.

Vincent Cohen-Addad, Philip N Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in Euclidean and minor-free metrics. In *Proc. 57th annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 353–364, 2016.

Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proceedings on Privacy Enhancing Technologies*, pages 92–112, 2015.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proc. 3rd Innovations in Theoretical Computer Science conference (ITCS)*, pages 214–226, 2012.

Laura P Edgerley, Yasser Y El-Sayed, Maurice L Druzin, Michaela Kiernan, and Kay I Daniels. Use of a community mobile health van to increase early access to prenatal care. *Maternal and Child Health Journal*, 11(3):235–239, 2007.

Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A PTAS for k-means clustering based on weak coresets. In *Proc. 23rd ACM Symposium on Computational Geometry (SOCG)*, pages 11–18, 2007.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268, 2015.

Roman Glebov, Anita Liebenau, and Tibor Szabó. On the concentration of the domination number of the random graph. *SIAM J. Discrete Math.*, 29(3):1186–1206, 2015. doi: 10.1137/12090054X. URL `https://doi.org/10.1137/12090054X`.

Sudipto Guha and Samir Khuller. Greedy strikes back: improved facility location algorithms. *Journal of algorithms*, 31(1):228–248, 1999.

Sepali Guruge, JoAnne Hunter, Keegan Barker, Mary Jane McNally, and Lilian Magalhães. Immigrant womens experiences of receiving care in a mobile health clinic. *Journal of Advanced Nursing*, 66(2):350–359, 2010.

David G. Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. Dependent rounding for knapsack/partition constraints and facility location. *arxiv*, abs/1709.06995, 2017.

David G Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. A lottery model for center-type problems with outliers. *ACM Transactions on Algorithms*, 15(3):Article #36, 2019.

Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, 1986. doi: 10.1145/5925.5933. URL `http://doi.acm.org/10.1145/5925.5933`.

Lingxiao Huang and Jian Li. Stochastic k-center and j-flat-center problems. In *Proc. 28th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 110–129, 2017.

Imanol Arrieta Ibarra, Leonard Goff, Diego Jiménez Hernández, Jaron Lanier, and E. Glen Weyl. Should we treat data as labor? Moving beyond 'free'. *American Economic Association Papers & Proceedings*, 1(1), 2018.

Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proc. 34th annual ACM Symposium on Theory of Computing (STOC)*, pages 731–740, 2002.

Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004. doi: 10.1016/j.comgeo.2004.03.003. URL `https://doi.org/10.1016/j.comgeo.2004.03.003`.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k-median and k-means with outliers via iterative rounding. In *Proc. 50th annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 646–659, 2018.

J. Lanier. *Who Owns the Future?* Simon Schuster, 2014.

Jiří Matoušek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000. doi: 10.1007/s004540010019. URL `https://doi.org/10.1007/s004540010019`.

John F McCarthy, Frederic C Blow, Marcia Valenstein, Ellen P Fischer, Richard R Owen, Kristen L Barry, Teresa J Hudson, and Rosalinda V Ignacio. Veterans affairs health system and mental health treatment retention among patients with serious mental illness: evaluating accessibility and availability barriers. *Health services research*, 42(3p1):1042–1060, 2007.

Cathleen Mooney, Jack Zwanziger, Ciaran S Phibbs, and Susan Schmitt. Is travel distance a barrier to veterans' use of VA hospitals for medical surgical care? *Social science & medicine*, 50(12):1743–1755, 2000.

Dana Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. *Theory of Computing*, 11(7):221–235, 2015.

David B Reuben, Lawrence W Bassett, Susan H Hirsch, Catherine A Jackson, and Roshan Bastani. A randomized clinical trial to assess the benefit of offering on-site mobile mammography in addition to health education for older women. *American Journal of Roentgenology*, 179(6):1509–1514, 2002.

Susan K Schmitt, Ciaran S Phibbs, and John D Piette. The influence of distance on utilization of outpatient mental health aftercare following inpatient substance abuse treatment. *Addictive Behaviors*, 28(6):1183–1192, 2003.

Kevin A. Schulman, Jesse A. Berlin, William Harless, Jon F. Kerner, Shyrl Sistrunk, Bernard J. Gersh, Ross Dub, Christopher K. Taleghani, Jennifer E. Burke, Sankey Williams, John M. Eisenberg, William Ayers, and Jos J. Escarce. The effect of race and sex on physicians' recommendations for cardiac catheterization. *New England Journal of Medicine*, 340(8):618–626, 1999.

Aravind Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proc. 42nd annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 588–597, 2001.

Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. In *Proc. 30th International Conference on Machine Learning (ICML)*, pages 325–333, 2013.

Journal of Machine Learning Research 20 (2019) 1-52 Submitted 11/18; Published 7/19

Convergence guarantees for a class of non-convex and non-smooth optimization problems

Koulik Khamaru[⋆] KOULIK@BERKELEY.EDU
Martin J. Wainwright[⋆,†,‡] WAINWRIG@EECS.BERKELEY.EDU
Department of Statistics[⋆]
Department of Electrical Engineering & Computer Sciences[†]
University of California Berkeley
Berkeley, CA 94720-1776, USA
Voleon Group[‡]
Berkeley, CA

Editor: Alexander Rakhlin

Abstract

We consider the problem of finding critical points of functions that are non-convex and non-smooth. Studying a fairly broad class of such problems, we analyze the behavior of three gradient-based methods (gradient descent, proximal update, and Frank-Wolfe update). For each of these methods, we establish rates of convergence for general problems, and also prove faster rates for continuous sub-analytic functions. We also show that our algorithms can escape strict saddle points for a class of non-smooth functions, thereby generalizing known results for smooth functions. Our analysis leads to a simplification of the popular CCCP algorithm, used for optimizing functions that can be written as a difference of two convex functions. Our simplified algorithm retains all the convergence properties of CCCP, along with a significantly lower cost per iteration. We illustrate our methods and theory via applications to the problems of best subset selection, robust estimation, mixture density estimation, and shape-from-shading reconstruction.

1. Introduction

Non-convex optimization problems arise frequently in statistical machine learning; examples include the use of non-convex penalties for enforcing sparsity (Fan and Li, 2001; Loh and Wainwright, 2013; Wainwright, 2019), non-convexity in likelihoods in mixture modeling (Yan et al., 2017), and non-convexity in neural network training (Li and Yuan, 2017). Of course, minimizing a non-convex problem is NP-hard in general, but problems that arise in machine learning applications are not constructed in an adversarial manner. Moreover, there have been a number of recent papers demonstrating that all first (and/or second) order critical points have desirable properties for certain statistical problems (e.g. Loh and Wainwright 2013; Ge et al. 2017). Given results of this type, it is often sufficient to find critical points that are first-order (and possibly second-order) stationary. Accordingly, recent years have witnessed an explosion of research on different algorithms for non-convex problems, with the goal of trying to characterize the nature of their fixed points, and their convergence properties.

©2019 Koulik Khamaru, Martin J. Wainwright.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-762.html.

There is a lengthy literature on non-convex optimization, dating back more than six decades, and rapidly evolving in the present (e.g., see Tuy 1995; Hartman 1959; Horst et al. 2000; Lanckriet and Sriperumbudur 2009; Yuille and Rangarajan 2003; Lee et al. 2016; Bolte et al. 2014; Panageas and Piliouras 2016; Lipp and Boyd 2016; Cartis et al. 2010; Attouch et al. 2010; Gotoh et al. 2017). Perhaps the most straightforward approach to obtaining a first-order critical point is via gradient descent. Under suitable regularity conditions and step size choices, it can be shown that gradient descent can be used to compute first-order critical points. Moreover, with a random initialization and additional regularity conditions, gradient descent converges almost surely to a second-order stationary point (e.g., Lee et al. 2016; Panageas and Piliouras 2016). These results, like much of the currently available theory for (sub)-gradient methods for non-convex problems, involve smoothness conditions on the underlying objectives. In practice, many machine learning problems have non-smooth components; examples include the hinge loss in support vector machines, the rectified linear unit in neural networks, and various types of matrix regularizers in collaborative filtering and recommender systems. Accordingly, a natural goal is to develop subgradient-based techniques that apply to a broader class of non-convex functions, allowing for non-smoothness.

The main contribution of this paper is to provide precisely such a set of techniques, along with non-asymptotic guarantees on their convergence rates. In particular, we study algorithms that can be used to obtain first-order (and in some cases, also second-order) optimal solutions to a relatively broad class of non-convex functions, allowing for non-smoothness in certain portions of the problem. For each sequence $\{x^k\}_{k\geq 0}$ generated by one of our algorithms, we provide non-asymptotic bounds on the convergence rate of the gradient sequence $\{\|\nabla f(x^k)\|_2\}_{k\geq 0}$. Moreover, for functions that satisfy a form of the Kurdaya-Łojasiewicz inequality, we show that our methods achieve faster rates.

Our work has important points of contact with a recent line of papers on algorithms for non-convex and non-smooth problems, and we discuss a few of them here. Bolte et al. (2014) developed a proximal-type algorithm applicable to objective functions formed as a sum of smooth (possibly non-convex) and a convex (possibly non-differentiable) function. Some recent work by Xu and Yin (2017) extended these ideas and provided analysis for block co-ordinate descent methods for non-convex functions. Hong et al. (2016) analyzed the ADMM method for non-convex problems, whereas in other recent work (An and Nam, 2017; Wen et al., 2018), the authors proposed a proximal-type method for non-convex functions that can be written as a sum of a smooth function, a concave continuous function and a convex lower semi-continuous function; we also analyze this class in one of our results (Theorem 2).

Our results also relate to another interesting sub-area of non-convex optimization, namely functions that can be represented as a difference of two convex functions, popularly known as DC functions. We refer the reader to the papers (Tuy, 1995; Hartman, 1959; Lanckriet and Sriperumbudur, 2009; Yuille and Rangarajan, 2003) for more details on DC functions and their properties. One of the most popular DC optimization algorithms is the Convex Concave Procedure, or CCCP for short; see the papers (Yuille and Rangarajan, 2003; Lipp and Boyd, 2016) for further details. This is a double loop algorithm that minimizes a convex relaxation of the non-convex objective function at each iteration. While the CCCP algorithm has some attractive convergence properties (Lanckriet and Sriperum-

budur, 2009), it can be slow in many situations due to its double loop structure. One outcome of the analysis in this paper is a single-loop proximal-method that retains all the convergence guarantees of CCCP while—as shown in our experimental results—being much faster to run.

1.1. Problem setup

In this paper, we study the problem of minimizing a non-convex and possibly non-smooth function over a closed convex set. More precisely, we consider optimization problems of the form

$$\min_{x \in \mathcal{C}} \Big\{ \underbrace{g(x) - h(x) + \varphi(x)}_{f(x)} \Big\}, \tag{1}$$

where the domain $\mathcal{C}$ is a closed convex set. In all cases, we assume the function f is bounded below over domain $\mathcal{C}$, and that the function h is continuous and convex. Our aim is to derive algorithms for problem (1) for various types of functions g and φ.

Structural assumption on functions g and h

(a) Theorems 1 and 4 are based on the assumption that the function g is continuously differentiable and smooth, and that the function $\varphi \equiv 0$.

(b) In Theorems 2 and 5, we assume that the function g is continuously differentiable and smooth, and that the function φ is convex, proper and lower semi-continuous.[1]

(c) Theorem 3 focuses on the case in which the function g is continuously differentiable, and the function $\varphi \equiv 0$.

The class of non-convex functions covered in part (a) includes, as a special case, the class of differences of convex (DC) functions, for which the first convex function is smooth and the second convex function is continuous. Note that we only put a mild assumption of continuity on the convex function h, meaning that the difference function $g - h$ can be non-smooth and non-differentiable in general. In particular, for any continuously differentiable function h and any smooth function g, the difference function $f = g - h$ is non-smooth. Furthermore, if we take the function $h \equiv 0$, then we recover the class of smooth functions as a special case.

1.2. Overview of our results

- Our first main result (Theorem 1) provides guarantees for a subgradient algorithm as applied to the minimization problem (2), to be defined in the sequel, when constrained to a closed convex set $\mathcal{C}$. We provide convergence bounds in terms of the Euclidean norm of the subgradient and show that our rates are unimprovable in general. We also illustrate some consequences of Theorem 1 by deriving a convergence rate for our algorithm when applied to non-smooth coercive functions; this result has interesting implications for polynomial programming. We also provide a simplification of the CCCP algorithm, along with convergence guarantees. In Corollary 3, we argue that our algorithm can escape strict saddle points for a large class of non-smooth functions, thereby generalizing known results for smooth functions.
- Our second main result (Theorem 2) provides convergence rates for a proximal-type algorithm for problem (1). In Section 4.3, we demonstrate how this proximal-type algorithm can be used to minimize a smooth convex function subject to a sparsity constraint. We demonstrate the performance of this algorithm through the example of best subset selection.
- In Theorem 3, we provide a Frank-Wolfe type algorithm for solving optimization problem (17), and we provide a rate of convergence in terms of the associated Frank-Wolfe gap.
- Finally, in Theorems 4 and 5, we prove that Algorithms 1 and 2, when applied to functions that satisfy a variant of the Kurdaya-Łojasiewicz inequality, have faster convergence rates. In particular, the convergence rate in terms of gradient norm is at least $\mathcal{O}(1/k)$ – whereas the worst case rate for general non-convex functions is $\mathcal{O}(\frac{1}{\sqrt{k}})$. We also provide examples of functions for which the convergence rate is $\mathcal{O}(1/k^r)$ with $r > 1$. In Theorem 6, we characterize the class of functions that can be written as a difference of a smooth function and a differentiable convex function.

Section 4 is devoted to an illustration of our methods and theory via applications to the problems of best subset selection, robust estimation, mixture density estimation and shape-from-shading reconstruction.

Notation: Given a set $\mathcal{C} \subset \mathbb{R}^d$, we use $\mathrm{int}(\mathcal{C})$ to denote its interior. We use $\|x\|_2$, $\|x\|_1$ and $\|x\|_0$ to denote the Euclidean norm, ℓ_1-norm and ℓ_0 norms, respectively, of a vector $x \in \mathbb{R}^d$. We say that a continuously differentiable function g is M_g-smooth if the gradient ∇g is M_g-Lipschitz continuous. In many examples considered in this paper, the objective function f is a linear combination of a differentiable function g and one or more convex functions h and φ. With a slight abuse of notation, for a function $f = g - h + \varphi$, we refer to a vector of the form $\nabla g(x) - u(x) + v(x)$, where $u(x) \in \partial h(x)$ and $v(x) \in \partial \varphi(x)$, as a gradient of the function f at point x — and we denote it by $\nabla f(x)$; here, $\partial h(\cdot)$ and $\partial \varphi(\cdot)$ denote the subgradient sets of the convex functions h and φ respectively. We say a point x is a *critical* point of the function f if $0 \in \nabla f(x)$. For a sequence $\{a^k\}_{k \geq 0}$, we define the running arithmetic mean $\mathrm{Avg}\left(a^k\right)$ as $\mathrm{Avg}\left(a^k\right) := \frac{1}{k} \sum_{\ell=0}^{\ell=k+1} a^\ell$. Similarly, for a non-negative sequence $\{a^k\}_{k \geq 0}$, we use $\mathrm{GAvg}\left(a^k\right) := (\prod_{\ell=0}^{k} a^\ell)^{\frac{1}{k+1}}$ to denote the running geometric mean. Finally, for real-valued sequences $\{a^k\}_{k \geq 0}$ and $\{b^k\}$, we say $a^k = \mathcal{O}\left(b^k\right)$, if there exists a positive constant C, which is independent of k, such that $a^k \leq C b^k$ for all $k \geq 0$. We say $a^k = \Omega(b^k)$ if $a^k = \mathcal{O}\left(b^k\right)$ and $b^k = \mathcal{O}\left(a^k\right)$.

1. Taking the function $\varphi \equiv 0$ yields part (a) as a special case, but it is worthwhile to point out that the assumptions in Theorem 1 are weaker than the assumptions of Theorem 2. Furthermore, we can prove some interesting results about saddle points when the function $\varphi \equiv 0$; see Corollary 3.

2. Main results

Our main results are analyses of three algorithms for this class of non-convex non-smooth problems; in particular, we derive non-asymptotic bounds on their rates of convergence. The first algorithm is a (sub)-gradient-type method, and it is mainly suited for unconstrained optimization; the second algorithm is based on a proximal operator and can be applied to constrained optimization problems. The third algorithm is a Frank-Wolfe-type algorithm, which is also suitable for constrained optimization problems, but it applies to a more general class of non-convex optimization problems.

2.1. Gradient-type method

In this section, we analyze a (sub)-gradient-based method for solving a certain class of non-convex optimization problems. In particular, consider a pair of functions (g, h) such that:

Assumption GR:

(a) The function g is continuously differentiable and M_g-smooth.

(b) The function h is continuous and convex.

(c) There is a closed convex set $\mathcal{C}$ such that the difference function $f := g - h$ is bounded below on the set $\mathcal{C}$.

Under these conditions, we then analyze the behavior of a (sub)-gradient method in application to the following problem

$$f^* = \min_{x \in \mathcal{C}} f(x) \;=\; \min_{x \in \mathcal{C}} \big\{ g(x) - h(x) \big\}. \tag{2}$$

Let $\partial h(x)$ denote the subdifferential of the convex function h at the point x. With a slight abuse of notation, we refer to a vector of the form $\nabla g(x) - u(x)$ with $u(x) \in \partial h(x)$ as a gradient of the function f at the point x.

Algorithm 1 Subgradient-type method

Given an initial point $x^0 \in \operatorname{int}(\mathcal{C})$ and step size $\alpha \in (0, \frac{1}{M_g}]$:
for $k = 0, 1, 2, \ldots$ **do**
  Choose subgradient $u^k \in \partial h(x^k)$.
  Update $x^{k+1} = x^k - \alpha\big(\nabla g(x^k) - u^k\big)$.
end for

In our analysis, we assume that the initial vector $x^0 \in \operatorname{int}(\mathcal{C})$ is chosen such that the associated level set

$$\mathcal{L}(f(x^0)) := \big\{ x \in \mathbb{R}^d \mid f(x) \leq f(x^0) \big\}$$

is contained within $\operatorname{int}(\mathcal{C})$. This condition is standard in the analysis of non-convex optimization methods (e.g., see Nesterov and Polyak 2006). When $\mathcal{C} = \mathbb{R}^d$, it holds trivially. With this set-up, we have the following guarantees on the convergence rate of Algorithm 1.

Theorem 1 *Under Assumption GR, any sequence $\{x^k\}_{k \geq 0}$ produced by Algorithm 1 has the following properties:*

(a) Any limit point is a critical point of the function f, and the sequence of function values $\{f(x^k)\}_{k \geq 0}$ is strictly decreasing and convergent.

(b) For all $k = 0, 1, 2, \ldots$, we have

$$\operatorname{Avg}\Big(\|\nabla f(x^k)\|_2^2 \Big) \leq \frac{2\big(f(x^0) - f^*\big)}{\alpha(k+1)}. \tag{3}$$

See Appendix B.1 for a proof of this theorem.

2.1.1. Comments on convergence rates

Note that the bound (3) guarantees that the gradient norm sequence $\min_{j \leq k} \|\nabla f(x^j)\|_2$ converges to zero at the rate $\mathcal{O}(1/\sqrt{k})$. It is natural to wonder whether this convergence rate can be improved. Interestingly, the answer is no, at least for the general class of functions covered by Theorem 1. Indeed, note that the class of M-smooth functions is contained within the class of functions covered by Theorem 1. It follows from past work by Cartis et al. (2010) that for gradient descent on M-smooth functions, with a step size chosen according to the Goldstein-Armijo rule, the convergence rate of the gradient sequence $\{\|\nabla f(x^k)\|_2\}_{k \geq 0}$ can be lower bounded—for appropriate choices of the function f—as $\Omega(1/\sqrt{k})$. It is not very difficult to see that the same construction also provides a lower bound of $\Omega(1/\sqrt{k})$ for gradient descent with a constant step size. We also note that very recently, Carmon et al. (2017) proved an even stronger result: more precisely, for the class of smooth functions, the rate of convergence of any algorithm given access to only the function gradients and function values cannot be faster than $\Omega(1/\sqrt{k})$. Finally, observe that in the special case $h \equiv 0$, Algorithm 1 reduces to the ordinary gradient descent with fixed step size α. Putting together the pieces, we conclude that for the class of functions which can be written as a difference of smooth and a continuous convex function, Algorithm 1 is *optimal* among all algorithms that have access to the gradients (and/or the sub-gradients) and the function values.

2.2. Consequences for differentiable functions

In the special case when the function h is convex and differentiable, Algorithm 1 reduces to an ordinary gradient descent on the difference function $f = g - h$. However, note that the step size choice required in Algorithm 1 does *not* depend on the smoothness of the function h; consequently, the algorithm can be applied to objective functions f that are not smooth. As a simple but concrete example, suppose that we wish to apply gradient descent to minimize the function $f(x) := g(x) - \|x\|_2^q$, where g is any μ-strongly convex and M_g-smooth function, and $q \in (1, 2)$ is a given parameter. Classical guarantees on gradient descent, which require the smoothness of the function f, would not apply here since the function f itself is not smooth. However, Theorem 1 guarantees that standard gradient descent would converge for any step size $\alpha \in \big(0, \frac{1}{M_g}\big]$.

More generally, given an arbitrary continuously differentiable function f, we can define its *effective smoothness constant* as

$$M_f^* := \inf_h \big\{L \mid (f+h) \text{ is } L\text{-smooth}\big\}, \tag{4}$$

where the infimum ranges over all convex and continuously differentiable functions h. Suppose that this infimum is achieved by some function h^*, then gradient descent on the function f can be viewed as applying Algorithm 1 to the decomposition $f = g^* - h^*$, where the function $g^* := f + h^*$ is guaranteed to be M_f^*-smooth. To be clear, the algorithm itself does *not* need to know the decomposition (g^*, h^*), but the existence of the decomposition ensures the success of a backtracking procedure. Putting together the pieces, we arrive at the following consequence of Theorem 1:

Corollary 1 *Given a closed convex set $\mathcal{C}$, consider a continuously differentiable function f with effective smoothness $M_f^* < \infty$ that is bounded below on $\mathcal{C}$. Then for any sequence $\{x^k\}_{k\geq 0}$ obtained by applying the gradient update with step size $\alpha \in \big(0, \frac{1}{M_f^*}\big)$, we have:*

$$\text{Avg}\Big(\|\nabla f(x^k)\|_2^2\Big) \leq \frac{2\big(f(x^0) - f^*\big)}{\alpha(k+1)}. \tag{5a}$$

Moreover, if we choose step size by backtracking[2] with parameter $\beta \in (0,1)$, then for all $k = 0, 1, 2, \ldots$, we have

$$\text{Avg}\Big(\|\nabla f(x^k)\|_2^2\Big) \leq \frac{2\max\big\{1, M_f^*\big\}\big(f(x^0) - f^*\big)}{\beta^2(k+1)}. \tag{5b}$$

See Appendix B.2 for proof of the above corollary.

Let us reiterate that the advantage of backtracking gradient descent is that it works without knowledge of the scalar M_f^*. The parameter β mentioned in equation (5b) is the user-defined backtracking parameter (see Algorithm 4 for details). In particular, substituting $\beta = \frac{1}{\sqrt{2}}$ in equation (5b) yields

$$\text{Avg}\Big(\|\nabla f(x^k)\|_2^2\Big) \leq \frac{4\max\big\{1, M_f^*\big\}\big(f(x^0) - f^*\big)}{(k+1)},$$

which differs from the rate obtained in equation (5a) only by a factor of two, and a possible multiple of M_f^*.

2.2.1. Consequences for coercive functions

As a consequence of Corollary 1, we can obtain a rate of convergence of the backtracking gradient descent algorithm (Algorithm 4) for a class of non-smooth coercive functions. Consider any twice continuously differentiable coercive function $f : \mathbb{R}^d \mapsto \mathbb{R}$, which is bounded below. Recall that a function f is *coercive* if

$$f(x^\ell) \overset{\ell\to\infty}{\to} \infty \quad \text{for any sequence } \{x^\ell\}_{\ell\geq 0} \text{ such that } \quad \|x^\ell\|_2 \to \infty. \tag{6}$$

2. A detailed description of gradient descent with backtracking is provided in Algorithm 4.

Let $\mathcal{L}(f(x^0)) := \big\{x \in \mathbb{R}^d : f(x) \leq f(x^0)\big\}$ denote the level set of the function f at point x^0. It can be verified that for any coercive function f, the set $\mathcal{L}(f(x^0))$ is bounded above for all $x^0 \in \mathbb{R}^d$. This property ensures that for any descent algorithm and any starting point x^0, the set of iterates $\big\{x^k\big\}_{k\geq 0}$ obtained from the algorithm remains within a bounded set—viz. the level set $\mathcal{L}(f(x^0))$ in this case. Since the function f is twice continuously differentiable, we have that f is smooth over bounded set $\mathcal{L}(f(x^0))$; this fact ensures that f has a finite effective smoothness constant in the set $\mathcal{L}(f(x^0))$, which we denote by M_{f,x^0}^*. Finally, note that Algorithm 4 is a descent algorithm; as a result, a simple application of Corollary 1 yields the following rate of convergence for the backtracking gradient descent algorithm (Algorithm 4):

Corollary 2 *Consider the unconstrained minimization problem of a twice continuously differentiable coercive function f that is bounded below on $\mathbb{R}^d$. Then for any initial point x^0, the sequence $\{x^k\}_{k\geq 0}$ obtained by applying Algorithm 4 satisfies the following property:*

$$\text{Avg}\Big(\|\nabla f(x^k)\|_2^2\Big) \leq \frac{2\max\big\{1, M_{f,x^0}^*\big\}\big(f(x^0) - f^*\big)}{\beta^2(k+1)} \quad \textit{for all } k = 0, 1, 2, \ldots, \tag{7}$$

where $\beta \in (0,1)$ is the backtracking parameter.

Implications for polynomial programming: Corollary 2 has useful implications for problems that involve minimizing polynomials. Such problems of polynomial programming arise in various applications, including phase retrieval and shape-from-shading (Wang et al., 2014), and we illustrate our algorithms for the latter application in Section 4.1. For minimization of a coercive polynomial, Corollary 2 shows that Algorithm 4 achieves a near-optimal rate.

It is worth noting that any even degree polynomial can be represented as a difference of convex (DC) function; hence, such problems are amenable to DC optimization techniques like CCCP, which we discuss at more length in Section 2.3. However, obtaining a good DC decomposition, which is crucial to the success of CCCP, is often a formidable task. In particular, obtaining an optimal decomposition for a polynomial with degree greater than four is NP-hard; indeed, deciding the convexity of an even degree polynomial with degree greater than four is NP-hard (Ahmadi et al., 2013; Wang et al., 2014). Even for a fourth degree polynomial with dimension larger than three, there is no known algorithm for finding an optimal DC decomposition (Ahmadi and Parrilo, 2013). An advantage of Algorithm 4 is that it obviates the need to find a DC decomposition.

2.2.2. Escaping strict saddle points

One of the obstacles with gradient-based continuous optimization method is possible convergence to saddle points. Here we show that with a random initialization this undesirable outcome does not occur for the class of strict saddle points. Recall that for a twice differentiable function f, a point x is called a strict saddle point of the function f if $\lambda_{\min}(\nabla^2 f(x)) < 0$, where $\lambda_{\min}(\nabla^2 f(x))$ denotes the minimum eigenvalue of the Hessian matrix $\nabla^2 f(x)$. The following corollary shows that such saddle points are *not* troublesome:

Corollary 3 *Suppose that, in addition to the conditions on $(g, h, \mathcal{C})$ from Theorem 1, the functions (g, h) are twice continuously differentiable. If Algorithm 1 is applied with step size*

$\alpha \in \left(0, \frac{1}{M_g}\right)$, *then the set of initial points for which it converges to a strict saddle point has measure zero.*

See Appendix B.3 for the proof of this corollary.

We note that similar guarantees of avoidance of strict saddle-points are known when the function $f = g - h$ is twice continuously differentiable and M-smooth (e.g., Lee et al. 2016; Panageas and Piliouras 2016). The novelty of Corollary 3 is that the same guarantee holds without imposing a smoothness condition on the entire function f.

2.3. Connections to the convex-concave procedure

As a consequence of Algorithm 1, we show that one can obtain a convergence rate of the Euclidean norm of the gradient for CCCP (convex-concave procedure), which is a heavily used algorithm in Difference of Convex (DC) optimization problems. Before doing so, let us provide a brief description of DC functions and the CCCP algorithm.

DC functions: Given a convex set $\mathcal{C} \subseteq \mathbb{R}^d$, we say that a function $f : \mathcal{C} \mapsto \mathbb{R}$ is DC if there exist convex functions g and h with domain $\mathcal{C}$ such that $f = g - h$. Note that the DC representation $f = g - h$ mentioned in the definition is not unique. In particular, for any convex function p, we can write $f = (g + p) - (h + p)$. The class of DC functions includes a large number of non-convex problems encountered in practice. Both convex and concave functions are DC in a trivial sense, and the class of DC functions remains closed under addition and subtraction. More interestingly, under mild restrictions on the domain, the class of non-zero DC functions is also closed under multiplication, division, and composition (e.g., Tuy 1995; Hartman 1959). The maximum and minimum of a finite collection of DC functions are also DC functions.

Convex-concave procedure: An interesting class of problems are those that involve minimizing a DC function over a closed convex set $\mathcal{C} \subseteq \mathbb{R}^d$, i.e.

$$f^* := \min_{x \in \mathcal{C}} f(x) \;=\; \min_{x \in \mathcal{C}} \big\{g(x) - h(x)\big\}, \tag{8}$$

where g and h are proper convex functions. The above problem has been studied intensively, and there are various methods for solving it; for instance, see the papers (Tuy, 1995; Lipp and Boyd, 2016; Pham Dinh et al., 2013) and references therein for details. One of the most popular algorithms to solve problem (8) is the Convex-concave Procedure (CCCP), which was introduced by Yuille and Rangarajan (2003). The CCCP algorithm is a special case of a Majorization-Minimization algorithm, one which uses the DC structure of the objective function in problem (8) to construct a convex majorant of the objective function f at each step. We start with a feasible point $x^0 \in \mathrm{int}(\mathcal{C})$. Let x^k denote the iterate at k^{th} iteration; at the $(k+1)^{\text{th}}$ iteration we construct a convex majorant $q(\cdot,\, x^k)$ of the function f via

$$f(x) \leq \underbrace{g(x) - h(x^k) - \langle u^k,\, x - x^k\rangle}_{=:\, q(x, x^k)}, \tag{9}$$

where $u^k \in \partial h(x^k)$, the subgradient set of the convex function h at point x^k. The next iterate x^{k+1} is obtained by solving the convex program

$$x^{k+1} \in \arg\min_{x \in \mathcal{C}} q(x, x^k). \tag{10}$$

The CCCP algorithm has some attractive convergence properties. For instance, it is a descent algorithm; when the function g is strongly convex differentiable and the function h is continuously differentiable, it can be shown (Lanckriet and Sriperumbudur, 2009) that any limit point of the sequence $\big\{x^k\big\}_{k \geq 0}$ obtained from CCCP is stationary. Under the same assumptions, one can also verify that $\lim_{k \to \infty} \|x^k - x^{k+1}\|_2 = 0$.

We now turn to an analysis of CCCP using the techniques that underlie Theorem 1. In the next proposition, we derive a rate of convergence of the gradient sequence and show that all limit points of the sequence $\big\{x^k\big\}_{k \geq 0}$ are stationary. Earlier analyses of CCCP, including the papers (Lanckriet and Sriperumbudur, 2009; Yuille and Rangarajan, 2003), are mainly based on the assumption of strong convexity of the function g, whereas in the next proposition, we only assume that the function g is M_g-smooth. When the function g is strongly convex, our analysis recovers the well-known convergence result in past work (Lanckriet and Sriperumbudur, 2009). In particular, we show that CCCP enjoys the same rate of convergence as that of Algorithm 1.

Proposition 1 *Under Assumption GR and with the function g being convex, the CCCP sequence* (10) *has the following properties:*

(a) Any limit point of the sequence $\big\{x^k\big\}_{k \geq 0}$ is a critical point, and the sequence of function values $\big\{f(x^k)\big\}_{k \geq 0}$ is strictly decreasing and convergent.

(b) Furthermore, for all $k = 1, 2, \ldots$, we have

$$\mathrm{Avg}\left(\|\nabla f(x^k)\|_2^2\right) \leq \frac{2M_g\big(f(x^0) - f^*\big)}{(k+1)}, \tag{11a}$$

and assuming moreover that g is μ-strongly convex,

$$\mathrm{Avg}\left(\|x^k - x^{k+1}\|_2^2\right) \leq \frac{2\big(f(x^0) - f^*\big)}{\mu(k+1)}. \tag{11b}$$

The proof of this proposition builds on the argument used for Theorem 1; see Appendix B.4 for details.

2.3.1. Simplifying CCCP

Algorithm 1 provides us an alternative procedure for minimizing a difference of convex functions when the first convex function is smooth. The benefit of Algorithm 1 over standard CCCP is that Algorithm 1 is a single loop algorithm and is expected to be faster than standard double loop CCCP algorithm in many situations. Furthermore, Algorithm 1 shares convergence guarantees similar to a standard CCCP algorithm.

2.4. Proximal-type method

We now turn to a more general class of optimization problems of the form

$$f^* := \min_{x\in\mathbb{R}^d} f(x) \;=\; \min_{x\in\mathbb{R}^d} \Big\{ \big(g(x) - h(x)\big) + \varphi(x) \Big\}. \tag{12}$$

We assume that the functions g, h and φ satisfy the following conditions:

Assumption PR

(a) The function $f = g - h + \varphi$ is bounded below on $\mathbb{R}^d$.

(b) The function g is continuously differentiable and M_g-smooth; the function h is continuous and convex; and the function φ is proper, convex and lower semi-continuous.

Typical examples of the function φ include $\varphi(x) = \|x\|_1$, or the indicator of a closed convex convex set $\mathcal{X}$. Since for a general lower semi-continuous function φ, the sum-function $g+\varphi$ is neither differentiable nor smooth, a gradient-based method cannot be applied. One way to minimize such functions is via a proximal-type algorithm, of which the following is an instance.

Algorithm 2 Proximal-type algorithm

1: Given an initial vector $x^0 \in \mathrm{dom}(f)$ and step size $\alpha \in \big(0, \frac{1}{M_g}\big]$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Update $x^{k+1} = \mathrm{prox}^{\varphi}_{1/\alpha}\Big(x^k - \alpha\big(\nabla g(x^k) - u^k\big)\Big)$ for some $u^k \in \partial h(x^k)$.
4: **end for**

The proximal update in line 3 of Algorithm 2 is very easy to compute and often has a closed form solution (see (Parikh et al., 2014)). Let us now derive the rate of convergence result of Algorithm 2.

Theorem 2 *Under Assumption PR, any sequence $\{x^k\}_{k\geq 0}$ obtained from Algorithm 2 has the following properties:*

(a) Any limit point of the sequence $\{x^k\}_{k\geq 0}$ is a critical point, and the sequence of function values $\{f(x^k)\}_{k\geq 0}$ is strictly decreasing and convergent.

(b) For all $k = 1, 2, \ldots$, we have

$$\mathrm{Avg}\Big(\|x^k - x^{k-1}\|_2^2\Big) \leq \frac{2\alpha\big(f(x^0) - f^*\big)}{(k+1)}. \tag{13a}$$

If moreover the function h is M_h-smooth, then

$$\mathrm{Avg}\Big(\|\nabla f(x^k)\|_2^2\Big) \leq \frac{2\alpha C_{M,\alpha}\big(f(x^0) - f^*\big)}{(k+1)}, \tag{13b}$$

where $C_{M,\alpha} = \big(M_g + M_h + \frac{1}{\alpha}\big)^2$.

See Appendix C for the proof of the theorem.

Comments: The proof of Theorem 2 reveals that the smoothness condition on the function h in Theorem 2 can be replaced by the local smoothness of h, when the sequence $\{x^k\}_{k\geq 0}$ is bounded. Note that the local smoothness condition is weaker than the global smoothness condition. For instance, any twice continuously differentiable function is locally smooth. The boundedness assumption on the iterates $\{x^k\}_{k\geq 0}$ holds in many situations. For instance, if the function f is coercive (6), then it follows that the iterates $\{x^k\}_{k\geq 0}$ remain bounded. Another instance is when the function φ is the indicator function of a compact convex set. Finally, we point out that when the function h is non-smooth but the proximal-function φ is smooth, the existing proof can be easily modified to obtain a rate of convergence of the gradient-norm $\|\nabla f(x^k)\|_2$.

Projected Gradient Descent: A special case of the Algorithm 2 is when φ is equal to the indicator function $\mathbb{1}_{\mathcal{X}}$ of a closed convex set $\mathcal{X}$. Consider the following constrained optimization problem

$$f^* := \min_{x\in\mathcal{X}} \; \big\{ \underbrace{g(x) - h(x)}_{f(x)} \big\}, \tag{14}$$

where $\mathcal{X}$ is a closed convex set, the function g is M_g-smooth, and the function h is convex continuous. Using Algorithm 2, the update equation in this case is given by

$$x^{k+1} = \Pi_{\mathcal{X}}\big(x^k - \alpha(\nabla g(x^k) - u^k)\big). \tag{15}$$

In projected-gradient-type methods, we should not expect a rate in terms of the gradient. In such cases, the projected gradient step may not be aligned with the gradient direction, or the step size may be arbitrarily small due to projection. Rather, an appropriate analogue of the gradient in this case is as follows:

$$\nabla f_{\mathcal{X}}(x^k) = \frac{1}{\alpha}\big(x^k - \Pi_{\mathcal{X}}(x^k - \alpha(\nabla g(x^k) - u^k))\big). \tag{16}$$

The analysis of the projected gradient method using $\nabla f_{\mathcal{X}}(x^k)$ is standard in the optimization literature (Bubeck et al., 2015). It is worth pointing out that the quantity $\nabla f_{\mathcal{X}}(x^k)$ is the analogue of the gradient in the constrained optimization setup, and coincides with the gradient in the unconstrained setup. Concretely, we have $\nabla f_{\mathcal{X}}(x^k) = \nabla f(x^k)$ where $f := g - h$, and $\mathcal{X} = \mathbb{R}^d$. Combining equations (15) and (16) and applying the bound (13b) from Theorem 2, we find that

$$\mathrm{Avg}\Big(\|\nabla f_{\mathcal{X}}(x^k)\|_2^2\Big) \leq \frac{2\big(f(x^0) - f^*\big)}{\alpha(k+1)}.$$

2.5. Frank-Wolfe type method

In our analysis of the previous two algorithms, we assumed that the objective function f has a smooth component g, and we leveraged the smoothness property of g to establish convergence rates. In many situations, the objective function may not have a smooth component; consequently, neither the gradient-type algorithm nor the prox-type algorithm provides any theoretical guarantee. In this section, we analyze a Frank-Wolfe-type algorithm

for solving such optimization problems. In particular, consider an optimization problem of the form

$$f^* := \min_{x \in \mathcal{C}} f(x) \;= \min_{x \in \mathcal{C}} \big\{ g(x) - h(x) \big\}, \tag{17}$$

where $\mathcal{C}$ is a closed convex set, and the functions (g, h) satisfy the following conditions:

Assumption FW:

(a) The difference function $f = g - h$ is bounded below over range $\mathcal{C}$.

(b) The function g is continuously differentiable, whereas the function h is convex and continuous.

The analysis of the Frank-Wolfe algorithm for a convex problem is based on the *curvature constant* $\mathcal{C}_f$ of the convex objective function with respect to the closed convex set $\mathcal{C}$. This curvature constant can be defined for any differentiable function, which need not be convex (Lacoste-Julien, 2016).

Here we define a slight generalization of this notion, applicable to a non-differentiable function $f = g - h$ that can be written as a difference of a differentiable function g and a continuous convex function h (which may be non-differentiable). Define the set

$$S_\gamma := \big\{ x, y \in \mathcal{C} \mid \text{there exist } \gamma \in (0, 1] \text{ and } u \in \mathcal{C} \text{ with } y = x + \gamma(u - x) \big\},$$

and the curvature constant

$$\mathcal{C}_f = \sup_{\substack{x, y \in S_\gamma \\ u \in \partial h(x)}} \frac{2}{\gamma^2} \big[f(y) - f(x) - \langle y - x, \nabla g(x) - u \rangle \big]. \tag{18}$$

Note that in the special case $h \equiv 0$, we recover the curvature constant of the differentiable function g used by Lacoste-Julien (Lacoste-Julien, 2016). We refer to the scalar $\mathcal{C}_f$ as the generalized curvature constant of the function f with respect to the closed convex set $\mathcal{C}$.

Algorithm 3 Frank-Wolfe type method

1: Given initial vector $x^0 \in \int(\mathcal{C})$:
2: **for** $k = 1, \ldots, K$ **do**
3: Choose any $u^k \in \partial h(x^k)$.
4: Compute $s^k := \arg\min_{s \in \mathcal{C}} \langle s, \nabla g(x^k) - u^k \rangle$.
5: Define $d^k := s^k - x^k$ and $g^k := -\langle d^k, \nabla g(x^k) - u^k \rangle$. *(Frank-Wolfe gap)*
6: Set $\gamma^k = \min\big\{ \frac{g^k}{C_0}, 1 \big\}$ for some $C_0 \geq \mathcal{C}_f$.
7: Update $x^{k+1} = x^k + \gamma^k d^k$.
8: **end for**

Next, we provide an analysis of Algorithm 3 in terms of the Frank-Wolfe (FW) gap g^k defined Step 5. We show that the minimum FW gap $\{g^k\}_{k \geq 0}$ defined in Algorithm 3 converges to zero at the rate $\frac{1}{\sqrt{k+1}}$.

Theorem 3 *Under Assumption FW, the Frank-Wolfe gap sequence $\{g^k\}_{k \geq 0}$ from Algorithm 3 satisfies the following property:*

$$\min_{0 \leq j \leq k} g^j \leq \frac{\max\big\{ 2\big(f(x^0) - f^*\big), C_0 \big\}}{\sqrt{k+1}} \qquad \textit{for all } k = 0, 1, 2, \ldots.$$

See Appendix D.1 for the proof of this theorem.

Comments: The FW gap appearing in Theorem 3 is standard in the analysis of Frank-Wolfe algorithm; note that it is invariant to an affine transformation of the set $\mathcal{C}$. Similar convergence guarantees for the minimum FW-gap are available for differentiable functions; for instance, see the paper (Lacoste-Julien, 2016). The novelty of the above theorem is that it provides convergence guarantees of minimum FW-gap for a class of non-differentiable functions.

Upper bound on generalized curvature constant: It is worth mentioning that Algorithm 3 only requires an upper bound of the generalized curvature constant $\mathcal{C}_{g-h}$. Consequently, it is interesting to obtain an upper bound for the scalar $\mathcal{C}_{g-h}$. For a M_g-smooth function g, one well-known upper bound of the curvature constant $\mathcal{C}_g$ is $M_g \times \big(\operatorname{diam}_{\|\cdot\|_2}(\mathcal{C}) \big)^2$; see also Jaggi (2013). A similar upper bound also holds for the generalized curvature constant defined in equation (59). In particular, we prove that for a difference function $f = g - h$, with the function h being convex continuous, the scalar $\mathcal{C}_{g-h}$ is always upper bounded by $\mathcal{C}_g$, the curvature constant of the function g (see Lemma 6).

3. Faster rate under KL-inequality

In the preceding sections, we have derived rates of convergence for the gradient norms for various classes of problems. It is natural to wonder if faster convergence rates are possible when the objective function is equipped with some additional structure. Based on Theorems 1 and 2, we see that both Algorithms 1 and 2 ensure that $\|x^k - x^{k+1}\|_2 \to 0$, meaning that the successive differences between the iterates converge to zero. Although we proved that any limit point of the sequence $\{x^k\}_{k \geq 0}$ has desirable properties, the condition $\|x^k - x^{k+1}\|_2 \to 0$ is not sufficient—at least in general—to prove convergence[3] of the sequence $\{x^k\}_{k \geq 0}$. In this section, we provide a sufficient condition under which Algorithm 1 and Algorithm 2 yield convergent sequences of iterates $\{x^k\}_{k \geq 0}$, and we establish that the gradient sequences $\{\|\nabla f(x)\|_2\}_{k \geq 0}$ converge at faster rates.

3.1. Kurdaya-Łojasiewicz inequality

Let us now establish a faster local rate of convergence of Algorithms 1 and 2 for functions that satisfy a form of the Kurdaya-Łojasiewicz (KL) inequality. More precisely, suppose that there exists a constant $\theta \in [0, 1)$ such that the ratio $\frac{(f(x) - f(\bar{x}))^\theta}{\|\nabla f(x)\|_2}$ is bounded above in a neighborhood of every point $\bar{x} \in \operatorname{dom}(f)$. This type of inequality is known as a Kurdaya-Łojasiewicz inequality, and the exponent θ is known as the Kurdaya-Łojasiewicz

3. The convergence of the sequence $\{x^k\}_{k \geq 0}$ for Algorithm 2 was studied in the papers (An and Nam, 2017; Wen et al., 2018). We provide the proof under a weaker set of assumptions.

exponent (*KL-exponent*) of the function f at the point $\bar{x}$. These type of inequalities were first proved by Łojasiewicz (1963) for real analytic functions; in later work, Kurdyka (1998) and Bolte et al. (2007) proved similar inequalities for non-smooth functions, and the authors also provided examples of many functions that satisfy a form of the KL inequality. See Appendix A.2 for further details on functions of the KL type.

Assumption KL: For any point[4] $\bar{x} \in \mathrm{dom}(f)$, there exists a scalar $\theta \in [0, 1)$ such that the ratio $\frac{|f(x)-f(\bar{x})|^\theta}{\|\nabla f(x)\|_2}$ is bounded above in a neighborhood of $\bar{x}$.

3.2. Convergence guarantees

Theorem 4 *Under Assumptions GR and KL, any bounded sequence $\{x^k\}_{k\geq 0}$ obtained from Algorithm 1 satisfies the following properties:*

(a) The sequence $\{x^k\}_{k\geq 0}$ converges to a critical point $\bar{x}$, and for all $k = 1, 2, \ldots$

$$\mathrm{Avg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_1}{k},$$

(b) Suppose that at the point $\bar{x}$, the function f has a KL exponent $\bar{\theta} \in \left[\frac{1}{2}, \frac{r}{2r-1}\right)$ for some $r > 1$. Then we have

$$\mathrm{GAvg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_2}{k^r} \qquad \textit{for all} \quad k = 1, 2, \ldots,$$

where the constants (c_1, c_2) are independent of k, but they may depend on the KL parameters at the point $\bar{x}$.

See Appendix E.1 for proof of this theorem.

Comments: It is worth pointing out that Theorem 4 does *not* require the function h to satisfy any smoothness assumption. Such conditions are needed for applying Algorithm 2, so that Theorem 4 is based on milder conditions than Theorem 5.

Our next result is to exhibit a faster convergence rate for Algorithm 2 under the KL assumption:

Theorem 5 *Suppose that, in addition to Assumptions PR & KL, the function h in Algorithm 2 is locally smooth. Then any bounded sequence $\{x^k\}_{k\geq 0}$ obtained from Algorithm 2 satisfy the following properties:*

(a) The sequence $\{x^k\}_{k\geq 0}$ converges to a critical point $\bar{x}$, and for all $k = 1, 2 \ldots$

$$\mathrm{Avg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_1}{k}.$$

4. It can be shown that such an inequality would hold at non-critical point of a continuous function f; see Remark 3.2 of (Bolte et al., 2007). Note that the parameter θ and the neighborhood mentioned in Assumption KL above may depend on the point $\bar{x}$.

(b) Given some $r > 1$, suppose that at the point $\bar{x}$ the function f has a KL exponent $\bar{\theta} \in \left[\frac{1}{2}, \frac{r}{2r-1}\right)$. Then

$$\mathrm{GAvg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_2}{k^r} \qquad \textit{for all} \quad k = 1, 2, \ldots,$$

where the constants (c_1, c_2) are independent of k, but they may depend on the KL parameters at the point $\bar{x}$.

See Appendix E.2 for the proof of this theorem.

Comments: Note that $\min_{1\leq i\leq k} \|\nabla f(x^k)\|_2$ is upper bounded by the quantities $\mathrm{Avg}\left(\|\nabla f(x^k)\|_2\right)$ and $\mathrm{GAvg}\left(\|\nabla f(x^k)\|_2\right)$. It thus follows that the sequence $\{\|\nabla f(x^k)\|_2\}_{k\geq 0}$ converges to zero at a rate of at least $1/k$, thereby improving the rate of convergence of $\|\nabla f(x)\|_2$ obtained in Theorems 1 and 2. When $\theta < \frac{1}{2}$, a simple modification of the proof (using $\gamma = 2$) shows that, Algorithms 1 and 2 converge in a finite number of steps. Finally, we point out that when the function h is non-smooth but the proximal-function φ is smooth, the existing proof can be easily modified to obtain a rate of convergence of the gradient-norm $\|\nabla f(x^k)\|_2$.

4. Some illustrative applications

In this section, we study four interesting classes of non-convex problems that fall within the framework of this paper. We also discuss various consequences of Theorems 1—5 as well as Corollaries 1—3 when applied to these problems.

4.1. Shape from shading

The problem of shape from shading is to reconstruct the three-dimensional (3D) shape of an object based on observing a two-dimensional (2D) image of intensities, along with some information about the light source direction. It is assumed that the observed 2D image intensity is determined by the angle between the light source direction and the surface normals of the object (Ecker and Jepson, 2010).

In more detail, suppose that both the object and its 2D image are supported on a rectangular grid of size $r \times c$. We introduce the shorthand notation $[r] = \{1, 2, \ldots, r\}$ and $[c] = \{1, 2, \ldots, c\}$ for the rows and columns of this grid. For each pair $(i, j) \in [r] \times [c]$, we let $I_{ij} \in \mathbb{R}$ denote the observed intensity at location (i, j) in the image, and we let $N_{ij} \in \mathbb{R}^3$ denote the surface normal at the vertex $v_{ij} := (x_{ij}, y_{ij}, z_{ij})$ of the object. Based on observing the 2-dimensional image, both the intensity I_{ij} and co-ordinate pair (x_{ij}, y_{ij}) are known for each pair $(i, j) \in [r] \times [c]$. The goal of shape from shading is to estimate the unknown coordinate z_{ij}, which corresponds to the height of the object at location (i, j). Knowledge of these z-coordinates allows us to generate a 3D representation of the object, as illustrated in Figure 1.

Lambertian lighting model: In order to reconstruct the z-coordinates, we require a model that relates the observed intensity I_{ij} to the surface normal. In a Lambertian model, for a given light source direction $L := (\ell_1, \ell_2, \ell_2)^\top \in \mathbb{R}^3$, it is assumed that the surface

normal N_{ij} and intensity I_{ij} are related via the relation

$$I_{ij} = \frac{\langle L,\, N_{ij}\rangle}{\|N_{ij}\|_2}. \tag{19}$$

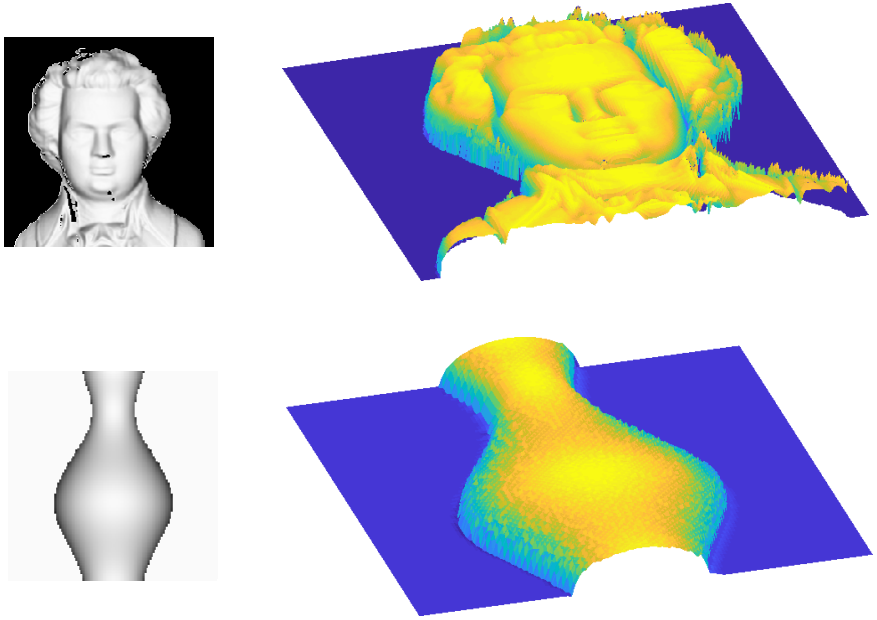


Figure 1. Figure shows 3D shape reconstruction of *Mozart* (first row) and *Vase* (second row) from corresponding 2D images. The gray-scale images in the left column are the 2D input images; the two colored images in the right column are the reconstructed 3D shapes. The 3D shapes are constructed by solving the problem (21) using Algorithm 4.

In one standard model (Wang et al., 2014), the surface normal $N_{ij} := (p_{ij}, q_{ij}, 1)^\top$ is assumed to be determined by the triplet of vertices $(v_{ij}, v_{i+1,j}, v_{i,j+1})$ via the equations

$$p_{ij} = \frac{(y_{i,j+1} - y_{i,j})(z_{i+1,j} - z_{ij}) - (y_{i+1,j} - y_{i,j})(z_{i,j+1} - z_{ij})}{(x_{i,j+1} - x_{ij})(y_{i+1,j} - y_{ij}) - (x_{i+1,j} - x_{ij})(y_{i,j+1} - y_{ij})},$$
$$q_{ij} = \frac{(x_{i,j+1} - x_{i,j})(z_{i+1,j} - z_{ij}) - (x_{i+1,j} - x_{i,j})(z_{i,j+1} - z_{ij})}{(x_{i,j+1} - x_{ij})(y_{i+1,j} - y_{ij}) - (x_{i+1,j} - x_{ij})(y_{i,j+1} - y_{ij})}.$$

Squaring both sides of equation (19) and substituting the expression for surface normal N_{ij} yields the polynomial equation

$$\big(p_{ij}^2 + q_{ij}^2 + 1\big)I_{ij} - (\ell_1 p_{ij} + \ell_2 q_{ij} + \ell_3)^2 = 0, \tag{20}$$

which should be satisfied under the assumed model.

In practice, this equality will not be exactly satisfied, but we can estimate the z-coordinates by solving the following non-convex optimization problem in the $r \times c$ matrix z with entries $\{z_{ij} \mid (i,j) \in [r] \times [c]\}$:

$$\min_{z \in \mathbb{R}^{r \times c}} \; \Big\{ \underbrace{\sum_{i=1}^{r} \sum_{j=1}^{c} \big((1 + p_{ij}^2 + q_{ij}^2) I_{ij}^2 - (\ell_1 p_{ij} + \ell_2 q_{ij} + \ell_3)^2\big)^2}_{P(z)} \Big\}. \tag{21}$$

Some reconstruction experiments: In order to illustrate the behavior of our method for this problem, we considered two synthetic images for simulated experiments. The first one is a 256×256 image of *Mozart* (Zhang et al., 1999), and the second one is a 128×128 image of *Vase*. The 3D shapes were constructed from the 2D images by solving optimization problem (21) using the backtracking gradient descent algorithm 4. The reconstructed surfaces for *Vase* and *Mozart* are provided in Figure 1. We ran 500 iterations of Algorithm 4 for both the images. The runtime for *Mozart*-example was 87 seconds, whereas the runtime for *Vase*-example was 39 seconds. The implementation of Algorithm 4 for Problem (21) is parallelizable; hence, the runtime can be much lower than our runtime with a parallel implementation. It is worth mentioning that the polynomial P is a fourth-degree polynomial with dimension $r \times c$; polynomial P is coercive and bounded below by zero. Consequently, we can apply Corollary 2 to the problem (21) which guarantees that average of the squared gradient norm $\text{Avg}\left(\|\nabla P\|_2^2\right)$ converges to zero at a rate $\frac{1}{k}$.

One might also consider applying the CCCP method to this problem. In a recent paper, Wang et al. (2014) provided a DC decomposition of the polynomial P using a sum of square (SOS) optimization technique. However, it is crucial to note that the DC decomposition of polynomial P obtained from the SOS-optimization method need not be optimal. In order to see this, note that the dimension of the polynomial P is much larger than three. In particular, the variable z_{ij} is used in the computation of surface normals $N_{ij}, N_{i,j-1}$ and $N_{i-1,j}$, hence is related to variables $(z_{i,j+1}, z_{i+1,j}, z_{i-1,j}, z_{i,j-1})$—which are again related to the other variables. Ahmadi and Parrilo (2013) showed that SOS techniques for deriving a DC decomposition are sub-optimal for a fourth-degree polynomial when the dimension of the polynomial is greater than three. Consequently, deriving an optimal DC decomposition for the polynomial P will be computationally intensive.

4.2. Robust regression using Tukey's bi-weight

Next, we turn to the problem of robust regression with Tukey's bi-weight penalty function. Suppose that we observe pairs $(y_i, z_i) \in \mathbb{R} \times \mathbb{R}^d$ linked via the noisy linear model

$$y_i = \langle z_i,\, \mu^* \rangle + \varepsilon_i \qquad \text{for } i = 1, \ldots, n.$$

Here the vector $\mu^* \in \mathbb{R}^d$ is the unknown parameter of interest, whereas the variables $\{\varepsilon_i\}_{i=1}^n$ correspond to additive noise. In robust regression, we obtain an estimate of the parameter

vector μ^* by computing

$$\min_{\mu \in \mathbb{R}^d} \underbrace{\Big\{ \frac{1}{n} \sum_{i=1}^{n} \Psi\big(y_i - \langle z_i, \mu \rangle\big) \Big\}}_{=: f(\mu)} \tag{22}$$

where Ψ is a known loss function with some robustness properties. One popular example of the loss function Ψ is Tukey's bi-weight function, which is given by

$$\Psi(t) = \begin{cases} 1 - (1 - (t/\lambda)^2)^3 & \text{if } |t| \leq \lambda \\ 1 & \text{otherwise} \end{cases}, \tag{23}$$

where $\lambda > 0$ is a tuning parameter. Note that Ψ is a smooth function, whence the function f in the objective (22) is also smooth, implying that Algorithm 1 is suitable for the problem.

With this set-up, applying Theorem 1, Theorem 4 and Corollary 3, we obtain the following guarantee:

Corollary 4 *Given a random initialization, any bounded sequence $\{\mu^k\}_{k \geq 0}$ obtained by applying Algorithm 1 to the objective (22) has the following properties:*

(a) Almost surely with respect to the random initialization, the sequence $\{\mu^k\}_{k \geq 0}$ converges to a point $\bar{\mu}$ such that $\nabla f(\bar{\mu}) = 0$ and $\nabla^2 f(\bar{\mu}) \succeq 0$.

(b) There is a universal constant c_1 such that

$$\text{Avg}\Big(\|\nabla f(\mu^k)\|_2 \Big) \leq \frac{c_1}{k} \qquad \text{for all } k = 1, 2, \ldots.$$

We provide the proof in Appendix F.1.

4.3. Smooth function minimization with sparsity constraints

Moving beyond the robust regression problem, we now discuss another interesting problem of minimizing a smooth function subject to sparsity penalty. Consider the following optimization problem

$$\min_{\substack{x \in \mathbb{R}^d \\ \|x\|_0 \leq s}} \; g(x), \tag{24}$$

where g is a smooth function, the ℓ_0-"norm" $\|x\|_0$ counts the number of non-zero entries in the vector x, and $s \in \{1, \ldots, d\}$ is a sparsity parameter. The constraint set $\big\{ x \in \mathbb{R}^d \mid \|x\|_0 \leq s \big\}$ is non-convex, and consequently, the optimization problem (24) is non-convex. However, the constraint set can be expressed as the level set of a certain DC function (Gotoh et al., 2017). In particular, let $|x|_{(d)} \geq |x|_{(d-1)} \geq \cdots \geq |x|_{(1)}$ denote the values of $x \in \mathbb{R}^d$ re-ordered in terms of their absolute magnitudes. In terms of this

notation, we have $\|x\|_1 \geq \sum_{i=d-s+1}^{d} |x|_{(i)}$ for all $x \in \mathbb{R}^d$, with equality holding if and only if x is s–sparse. This fact ensures that

$$\Big\{ x \in \mathbb{R}^d : \|x\|_0 \leq s \Big\} = \Big\{ x \in \mathbb{R}^d : \|x\|_1 - \sum_{i=d-s+1}^{d} |x|_{(i)} \leq 0 \Big\}. \tag{25}$$

Since both of the functions $x \mapsto \|x\|_1$ and $x \mapsto \sum_{i=d-s+1}^{d} |x|_{(i)}$ are convex (Boyd and Vandenberghe, 2004), this level set formulation is a DC constraint. Now using the representation (25), we can rewrite problem (24) as $\min_{x \in \mathbb{R}^d} g(x)$ such that $\|x\|_1 - \sum_{i=d-s+1}^{d} |x|_{(i)} \leq 0$. For our experiments, it is more convenient to solve the penalized analogue of the last problem, given by

$$\min_{x \in \mathbb{R}^d} \; \Big\{ g(x) + \lambda \Big(\|x\|_1 - \sum_{i=d-s+1}^{d} |x|_{(i)} \Big) \Big\}, \tag{26}$$

where $\lambda > 0$ is a tuning parameter. The optimization problem (26) can be solved using Algorithm 2 with $g(x) = g(x)$, $\varphi(x) = \lambda \|x\|_1$ and $h(x) = \lambda \sum_{i=d-s+1}^{d} |x|_{(i)}$. For the non-smooth component $\varphi(x) = \lambda \|x\|_1$, there is a closed form expression of the proximal update in Algorithm 2, so that the method is especially efficient in this case.

4.3.1. Best subset selection

A special case of problem (26) arises from best subset selection in linear regression. Suppose that we observe a vector $y \in \mathbb{R}^n$ and a matrix $B \in \mathbb{R}^{n \times d}$ that are linked via the standard linear model $y = Bx^* + \varepsilon$. Here the vector $\varepsilon \in \mathbb{R}^n$ corresponds to additive noise, whereas $x^* \in \mathbb{R}^d$ is the unknown regression vector. We wish to estimate the unknown parameter vector x^* subject to a sparsity constraint, and we do so by solving the following optimization problem:

$$\min_{\substack{x \in \mathbb{R}^d \\ \|x\|_0 \leq s}} \|y - Bx\|_2^2. \tag{27}$$

Here the non-negative integer s is a tuning parameter that controls maximum number of allowable non-zero entries in the vector x. Following the development leading to the formulation (26), let us consider instead the problem of minimizing the function

$$f(x) := \|y - Bx\|_2^2 + \lambda \Big(\|x\|_1 - \sum_{i=d-s+1}^{d} |x|_{(i)} \Big) \Big\}. \tag{28}$$

Note that the function f can be decomposed as a difference of two convex functions as follows:

$$f(x) = \underbrace{\|y - Bx\|_2^2 + \lambda \|x\|_1}_{\text{convex}} - \underbrace{\lambda \sum_{i=d-s+1}^{d} |x|_{(i)}}_{\text{convex}}. \tag{29}$$

Consequently, problem (28) is a DC optimization problem; hence, it is amenable to standard DC optimization techniques like CCCP. We can also apply Algorithm 2 on problem (28) with $g(x) = \|y - Bx\|_2^2$, $\varphi(x) = \lambda \|x\|_1$ and $h(x) = \lambda \sum_{i=d-s+1}^{d} |x|_{(i)}$.

4.3.2. Comparison of Algorithm 2 and CCCP

Let us compare the performance of our Algorithm 2 (prox-type method) with the popular convex-concave procedure (CCCP) for minimizing differences of convex functions. We apply both algorithms to the best subset selection problem (28).

Let us reiterate that problem (28) can be written as a difference of two convex functions, and one can apply CCCP update (10) to the decomposition (29). The inner convex optimization problem in update (10) is solved by proximal methods for minimizing the sum of a smooth convex function and a ℓ_1 regularizer. We also apply Algorithm 2 on problem (28) with $g(x) = \|y - Bx\|_2^2$, $h(x) = \lambda \sum_{i=d-s+1}^{d} |x|_{(i)}$ and $\varphi(x) = \lambda \|x\|_1$.

Synthetic data generation: We generated the rows of the $n \times d$ matrix B from a d-dimensional Gaussian distribution with zero mean and an equicovariance matrix Σ, where $\Sigma_{ii} = 1$ for all i, and $\Sigma_{ij} = 0.7$ for all $i \neq j$. The regression vector $x^* \in \mathbb{R}^d$ (true value) was chosen to be a binary vector with sparsity s ($s \ll d$). The location of the nonzero entries of the vector x^* was chosen uniformly without replacement form the set $\{1, \ldots, d\}$.

Performance measures: We use the following two criteria to compare the performance of the prox-type method and CCCP.

(a) *Total runtime:* Firstly, we compare the algorithms in terms of their total runtime. The runtime was measured in units of seconds.

(b) *Estimation error:* Secondly, we use average estimation error of the algorithms as a measure of performance. Let us recall that if $\bar{x} \in \mathbb{R}^d$ is the estimated value of the unknown regression vector x^*, then the average estimation error is defined as $\frac{\|\bar{x} - x^*\|_2}{\sqrt{p}\|\bar{x}\|_2}$. Note that the average estimation error used here is invariant under scaling.

Comparison results: Figure 2 shows the performances of the prox-type method and CCCP for synthetic data simulated as above, with problem parameters $(n, p) = (190, 300)$ and $(n, p) = (380, 600)$ and different choices of sparsity s.

For both the algorithms, the tolerance level η was set to $\eta = 10^{-8}$, whereas the maximum number of iterations was 1000. Figure 2 suggests that total runtime of the prox-type method is significantly smaller than the runtime of CCCP. Furthermore, the estimation error for the prox-type method is lower compared to CCCP, which possibly suggests that prox-type method is finding better local minima compared to CCCP for the non-convex optimization problem (28). In all our simulations we used same initializations for both the algorithms. The simulation results shown in Figure 2 are average over 100 replications, and we also provide the pointwise error bar in the plots.

4.3.3. Some theoretical guarantees

Interestingly, it turns out that when applied to problem (28), the convergence behavior of Algorithm 2 to a given stationary point $\bar{x}$ depends on the behavior of a certain convex program defined in terms of $\bar{x}$. More precisely, for any point $\bar{x} \in \mathbb{R}^d$ with $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)}$, consider the following convex relaxation of problem (28):

$$\mathcal{P}(\bar{x}) := \min_{x \in \mathbb{R}^d} \left\{ \|y - Bx\|_2^2 + \lambda \|x\|_1 - \lambda \langle \nabla h(\bar{x}), x - \bar{x} \rangle \right\}. \tag{30}$$

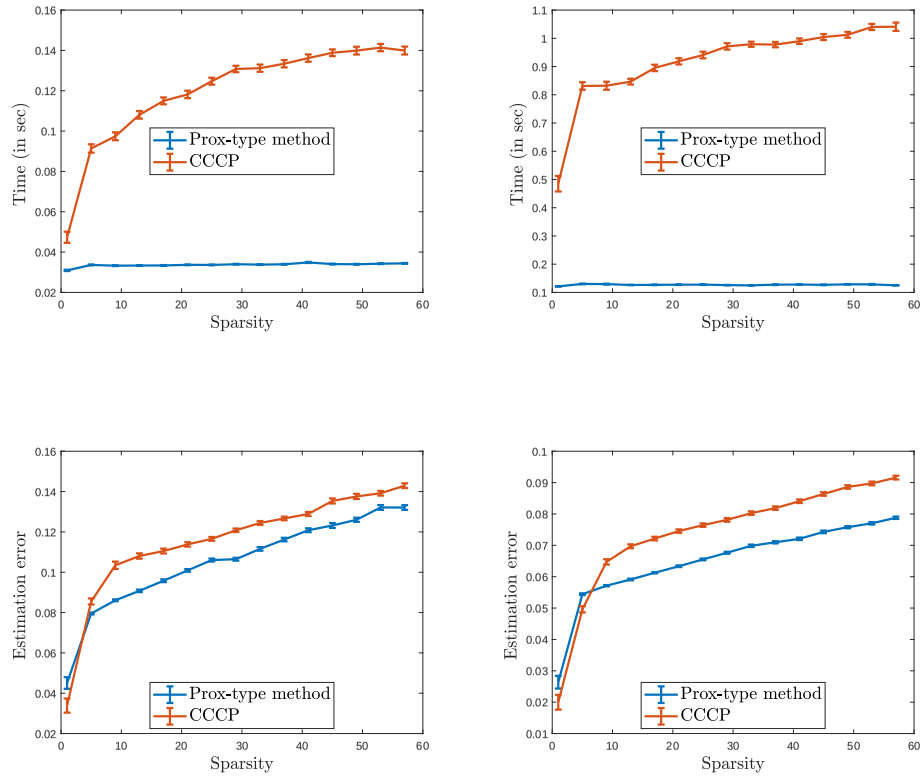


Figure 2. Performance of CCCP compared to that of Algorithm 2 on the best subset selection problem for synthetic data for different values of (n, p). The left columns correspond to $(n, p) = (190, 300)$, whereas the right columns correspond to $(n, p) = (380, 600)$. Plots in the first row compare the performance in terms of total runtime, those in the second row compare algorithms in terms of estimation error. We see that Algorithm 2 outperforms CCCP in terms of runtime. The performance of Algorithm 2 and CCCP in terms of estimation error are similar for low values of sparsity, whereas Algorithm 2 outperforms CCCP when sparsity is moderate to large. We initialized both the algorithms from the same starting point. Results shown above are averaged over 100 replications, and we also provide point-wise error bars in the plots.

Note that $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)}$ implies the differentiability of the function $h := \lambda \sum_{i=d-s+1}^{d} |x|_{(i)}$ which ensures that the above problem is well-defined.

Corollary 5 *Let $\{x^k\}_{k\geq 0}$ be any bounded sequence obtained by applying Algorithm 2 on problem (28). Suppose there exists a limit point $\bar{x}$ of the sequence $\{x^k\}_{k\geq 0}$ satisfying $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)}$, and the convex problem (30) has unique solution. Then the sequence $\{x^k\}_{k\geq 0}$ converges to the point $\bar{x}$, and for all $k = 1, 2, \ldots$, we have*

$$\text{Avg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_1}{k}, \quad \textit{and} \quad \|x^k - \bar{x}\|_2 \leq c q^k,$$

where $q \in (0,1)$, and (c, c_1) are positive constants independent of k.

Comments on problem (30): It can be shown that when the matrix B is of full rank, the objective function in problem (30) is strictly convex, and as a result, the problem (30) has unique solution. In the proof of Corollary 5, we show that the point $\bar{x}$ is always a minimizer of the convex problem (30), so that the uniqueness assumption implies that $\bar{x}$ is in fact the unique solution.

4.4. Mixture density estimation

As a final example, we consider the problem of estimating a two-component mixture density, where each of the constituent densities belong to an exponential family. The density of an exponential family (with respect to a fixed base measure, typically counting or Lebesgue) takes the form

$$p(y;\eta) = g(y)\exp\left\{\langle \eta, T(y)\rangle - A(\eta)\right\}. \tag{31}$$

Here the function $T : \mathcal{Y} \to \mathbb{R}^d$ is a vector of sufficient statistics, whereas the log-partition function

$$A(\eta) := \log\Big(\int_{\mathcal{Y}} g(y)\exp\{\langle \eta, T(y)\rangle\}dy\Big)$$

serves to normalize the density. The parameter vector $\eta \in \mathbb{R}^d$ determines the choice of density within the family. See Table 1 for some examples of 1-dimensional exponential families of this type. It includes various familiar examples, such as the Gaussian, Poisson and Beta families.

In the problem of mixture density estimation, one is interested in densities of the form

$$\zeta(y; \underbrace{\pi, \eta_0, \eta_1}_{\theta}) = \pi\, p(y;\eta_0) + (1-\pi)p(y;\eta_1), \tag{32}$$

where $\pi \in (0,1)$ is an unknown mixing proportion, and (η_0, η_1) are the unknown parameters of the two underlying densities.

Given n i.i.d. samples $\{y_i\}_{i=1}^n$ drawn from a mixture density of the form (32), a standard goal is to estimate the unknown parameter vector $\theta := (\pi, \eta_0, \eta_1)$. One way to do so is by

Distribution Name	η	$A(\eta)$	Twice continuously differentiable and sub-analytic
Poisson (λ)	$\ln(\lambda)$	$\exp\eta$	✓
Geometric (p)	$\ln(p)$	$-\ln(1-\exp\eta)$	✓
Gaussian(μ, σ^2)	$\left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right)^\top$	$-\frac{\eta_1^2}{4\eta_2} - \frac{1}{2}\ln(-2\eta_2)$	✓
Exponential (λ)	$-\lambda$	$-\ln(-\eta)$	✓
Gamma (α, β)	$(\alpha-1, \beta)^\top$	$\ln\Gamma(\eta_1+1) - (\eta_1+1)\ln(\eta_2)$	✓
Weibull (λ, k^5)	$-\frac{1}{\lambda^k}$	$\ln(-\eta) - \ln(k)$	✓
Beta (α, β)	$(\alpha, \beta)^\top$	$\ln\Gamma(\eta_1)+\ln\Gamma(\eta_2)-\ln\Gamma(\eta_1+\eta_2)$	✓

Table 1. Table showing the natural parameter η and the log-partition function A for different densities of exponential family, which are twice continuously differentiable and sub-analytic. In Appendix F.3 we prove the log-partition functions A mentioned in the above table are sub-analytic.

computing the maximum likelihood estimate (MLE), obtained via minimizing the negative log-likelihood of parameter θ given by the data. Frequently, a regularized form of the MLE is used, say of the form

$$\min_{\theta}\Big\{\underbrace{-\sum_{i=1}^{n}\log\big(\zeta(y_i;\theta)\big)}_{g(\theta)}\Big\} \qquad \text{such that } \eta_0, \eta_1 \in \mathbb{R}^d,\ \pi \in [0,1], \text{ and } \|\eta_0\|_2 \leq R_0,\ \|\eta_1\|_2 \leq R_1. \tag{33}$$

Here $R_0 > 0$ and $R_1 > 0$ are tuning parameters providing upper bound on the ℓ_2-norms of the parameters η_0 and η_1 respectively, often chosen by a data-dependent procedure (such as cross-validation).

By inspection, the objective function g in problem (33) is non-convex. By standard theory on exponential families, the function A is always infinitely differentiable on its domain, so that the objective function g is infinitely differentiable on the convex set

$$\mathcal{X} = \Big\{\theta = (\eta_0, \eta_1, \pi) \mid \eta_j \in \text{dom}(A), \pi \in [0,1], \|\eta_j\|_2 \leq R_j \quad \text{for } j = 0, 1\Big\}.$$

Consequently, we may apply Algorithm 2 with $g(\cdot) = -\sum_{i=1}^{n}\log\big(\zeta(\cdot; y_i)\big)$, $h \equiv 0$ and $\varphi(\cdot) = \mathbb{1}_{\mathcal{X}}(\cdot)$ and $f = g - h + \varphi$. Interestingly, the log-partition function A is sub-analytic for many exponential family densities (see Table 1), which ensures that the function g is also sub-analytic. In Appendix A.3, we show that continuous sub-analytic functions satisfy Assumption KL so that we can apply Theorem 5 to obtain the following:

Corollary 6 *Any sequence $\{\theta^k\}_{k\geq 0} = \{\eta_0^k, \eta_1^k, \pi^k\}_{k\geq 0}$ obtained by applying Algorithm 2 to problem (33) satisfies the following properties:*

(a) It converges to a first order stationary point.

(b) *For all $k = 1, 2, \ldots$, we have* $\mathrm{Avg}\left(\|\nabla f(\theta^k)\|_2\right) \leq \frac{c_1}{k}$, *where c_1 is a universal constant independent of k.*

See Appendix F.3 for the proof of this corollary.

5. Discussion

In this paper, we analyzed the behavior of three gradient-based algorithms—namely gradient descent, a proximal method, and an algorithm of the Frank-Wolfe type—for finding critical points of a class of non-convex non-smooth optimization problems. For each of the three algorithms, we provided non-asymptotic bounds on the rate of convergence to a first-order stationary point. We showed that our algorithm can escape strict saddle point for a class of non-smooth functions, thereby generalizing existing results for smooth functions. As a consequence of our theory, we obtained a simplification of the popular CCCP algorithm, and the simplified algorithm retains all the convergence properties of CCCP. Finally, we showed that for a large subclass of functions, which include continuous sub-analytic functions as a special case, we can have a significant improvement in the rate of convergence.

Our work leaves open a number of questions for future research. For instance, it would be interesting to characterize the class of DC-based functions mentioned in problem (2) when the convex function h is non-differentiable. Indeed, we then obtain a larger non-class of non-differentiable functions, and we suspect that Theorem 6 can be suitably generalized. Finally, we suspect that the proof techniques used here can be leveraged in order to establish sharper results for other forms of non-convex optimization problems.

Acknowledgments

This work was partially supported by the Office of Naval Research Grant DOD ONR-N00014 and National Science Foundation Grant NSF-DMS-1612948.

Appendix A. Technical background

In this appendix, we collect some technical background on subdifferentials and sub-analytic functions.

A.1. Fréchet and limiting subdifferential

We first recall the definitions and some useful properties of sub-differentials, which will be useful in subsequent sections.

Definition 1 *Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be a lower semicontinuous function. For any $x \in \mathrm{dom}(f)$, the Fréchet subgradient of the function f at point x is defined as*

$$\widehat{\partial} f(x) = \left\{ u \;\middle|\; \liminf_{y \neq x, y \to x} \frac{f(y) - f(x) - \langle u, y - x \rangle}{\|y - x\|_2} \geq 0 \right\}.$$

Definition 2 *Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be a lower semi-continuous function. For any $x \in \mathrm{dom}(f)$, the limiting subdifferential of the function f at point x is defined as*

$$\partial_L f(x) = \left\{ u \;\middle|\; \exists\; x^k \to x, u^k \to u \text{ with } f(x^k) \to f(x) \text{ and } u^k \in \widehat{\partial} f(x^k) \text{ as } k \to \infty \right\}.$$

Properties: The following properties of Fréchet and limiting sub-differential are provided in Chapter 8 of Rockafellar and Wets (2009).

(a) For any proper convex function h, we have $\partial_L h(x) = \widehat{\partial} h(x)$ for all $x \in \mathrm{dom}(h)$, and both quantities agree with the usual subgradient of the convex function h.

(b) If a function g is smooth in a neighborhood of a point x, then $\partial_L f(x) = \nabla f(x)$.

(c) Consider a function f of the form $f = g + \varphi$, where the function g is smooth in a neighborhood of a point x, and the function φ is proper convex and finite at the point x. Then the limiting sub-differential of the function f at the point x is given by $\partial_L f(x) = \nabla g(x) + \partial \varphi(x)$.

(d) *(Graph continuity:)* Consider a sequence $\left\{(x^k, u^k)\right\}_{k \geq 1}$ in $\mathrm{graph}(\partial_L f)$ such that the sequence $\{(x^k, u^k, f(x^k)\}_{k \geq 0}$ converges to a point $(x, u, f(x))$. Then $(x, u) \in \mathrm{graph}(\partial_L f)$. Recall that $\mathrm{graph}(\partial_L f) := \left\{(x, u) \in \mathbb{R}^d \times \mathbb{R} \mid u \in \partial_L f(x)\right\}$.

.

A.2. Sub-analytic functions satisfy KL-assumption

In this appendix, we show that continuous sub-analytic functions satisfy the KL-inequality. We also provide examples of functions which are sub-analytic.

Comments on limiting sub-differential: In order to facilitate our discussion, we mention some simple facts on limiting subdifferential of a function f, where f is of the form $f = g - h$ (Theorems 1 and 4) or $f = g + \varphi - h$ (Theorems 2 and 5). The following properties are direct consequences of properties of the limiting subdifferential mentioned in Appendix A.1.

- Suppose that the difference function $f = g - h$ satisfies parts (a) and (b) of Assumption GR. Then we have

$$\partial_L(-f)(x) = \partial h(x) - \nabla g(x), \quad \text{and moreover}$$
$$\|\nabla f(x)\|_2 := \|\nabla g(x) - \partial h(x)\|_2 = \|\partial_L(-f)(x)\|_2$$

- Suppose that the function $f = g + \varphi - h$, where the function h is locally smooth, and the function f satisfies Assumption PR part (b). Then $\partial_L f(x) = \nabla g(x) - \nabla h(x) + \partial \varphi(x)$. Consequently, we have that $\|\nabla f(x)\|_2 = \|\partial_L f(x)\|_2$.

We prove that continuous sub-analytic functions satisfy Assumption KL by exploiting results due to Bolte et al. (2007). Let us introduce some notation used in this paper. We use $m_f(x)$ to denote the ℓ_2 distance of the set $\partial_L f(x)$ from zero; concretely, $m_f(x) := \mathrm{dist}_{\|\cdot\|_2}\left(0, \partial_L f(x)\right)$. In Theorem 3.1 (for critical points of the function f) and Remark 3.2 (for non-critical points of the function f), Bolte et al. (2007) proved the following fact about sub-analytic functions.

Lemma 1 (Bolte et al. (2007)): *Let $f : \mathbb{R}^d \mapsto \mathbb{R} \cup \{+\infty\}$ be a sub-analytic function with closed domain, and assume that $f|_{\mathrm{dom}(f)}$ is continuous. Then for any $a \in \mathrm{dom}(f)$, there exists an exponent $\theta \in [0, 1)$ such that, the function $\frac{|f-f(a)|^\theta}{m_f}$ is bounded above in a neighborhood of a.*

Using Lemma 1, we now argue that sub-analytic functions, under the conditions of Theorem 4 or Theorem 5, satisfy Assumption KL.

Lemma 2 *Any sub-analytic function f satisfying Assumption GR also satisfies Assumption KL.*

Proof First, note that the function f is continuous by Assumption GR; suppose f is sub-analytic, then from properties of sub-analytic functions, we have that the function $-f$ is also sub-analytic. Furthermore, the function $-f$ is continuous in the closed domain $\mathcal{C}$ — which by Lemma 1 guarantees that, for any $a \in \mathcal{C}$, there exists $\theta \in [0, 1)$ such that the ratio $\frac{|-f-(-f(a))|^\theta}{m_{(-f)}}$ is bounded above in a neighborhood of the point a. Since $|-f-(-f(a))| = |f - f(a)|$, proving satisfiability of Assumption KL reduces to showing that $m_{(-f)}(x)$ is upper bounded by $\|\nabla f(x)\|_2$. To this end, note that from the discussion about limiting subdifferential in the paragraph above Lemma 1, we have

$$\|\nabla f(x)\|_2 = \|\partial_L(-f)(x)\|_2 \overset{(i)}{\geq} m_{(-f)}(x), \tag{34}$$

where step (i) follows from the definition of $m_{(-f)}(x)$. Putting together the pieces, we conclude that any sub-analytic function f which satisfies Assumption GR, also satisfies Assumption KL. ■

Lemma 3 *Suppose that, in addition to the conditions on the functions (g, h, φ) from Theorem 2, the function $f := g - h + \varphi$ is continuous and sub-analytic in its domain $\mathrm{dom}(f)$, and the domain $\mathrm{dom}(f)$ is closed. Then the function f satisfies Assumption KL.*

Proof Since the function $f|_{\mathrm{dom}(f)}$ is continuous and sub-analytic by assumption, from Lemma 1, we have that for any $a \in \mathrm{dom}(f)$ there exists a $\theta \in [0, 1)$ such that, the ratio $\frac{|f-f(a)|^\theta}{m_f}$ is bounded above in a neighborhood of the point a. In order to justify satisfiability of Assumption KL, it suffices to prove that $m_f(x)$ is upper bounded by $\|\nabla f(x)\|_2$. To this end, note that the function h is locally smooth by assumptions of Theorem 2 part (b). Hence, from the discussion about limiting subdifferential in the paragraph above Lemma 1, we have

$$\|\nabla f(x)\|_2 = \|\partial_L f(x)\|_2 \overset{(i)}{\geq} m_f(x), \tag{35}$$

where step (i) follows from the definition of $m_f(x)$. Putting together the pieces, guarantees that the function f satisfies Assumption KL. ■

A.3. Instances of sub-analytic functions

In Appendix A.2, we proved that continuous sub-analytic functions satisfy Assumption KL, and in those cases,—by Theorems 4 and 5—we have a faster rate of convergence of Algorithms 1 and 2. In this appendix, we provide examples of functions which are sub-analytic. We start by providing definitions of sub-analytic functions following the definition of (Bolte et al., 2007).

A subset $S \subset \mathbb{R}^d$ is called *semi-analytic* if each point of $\mathbb{R}^d$ admits a neighborhood V such that the set $S \cap V$ has the form

$$S \cap V = \cup_{i=1}^p \cap_{j=1}^q \big\{x \in V \mid h_{ij} = 0, g_{ij} > 0\big\},$$

where the functions $h_{ij}, g_{ij} : V \mapsto \mathbb{R}$ are real-analytic.

A set S is called *sub-analytic*, if each point of $\mathbb{R}^d$ admits a neighborhood V such that

$$S \cap V = \big\{x \in \mathbb{R}^d : \big(x, y\big) \in B\big\},$$

where B is a bounded semi-analytic subset of $\mathbb{R}^d \times \mathbb{R}^m$ for some $m \geq 1$. A function f is called sub-analytic if the graph of f, defined by $\mathrm{graph}(f) := \big\{(x, y) \in \mathbb{R}^d \times \mathbb{R} : f(x) = y\big\}$, is sub-analytic.

The class of sub-analytic functions is quite large. In order to motivate the reader, we provide few examples here. The following results can be found in Bolte et al. (2014) and Chapter 6 in the book by Facchinei and Pang (2007).

(a) Any real-valued polynomial or analytic function is sub-analytic.

(b) Any real-valued semi-algebraic or semi-analytic function is sub-analytic.

(c) Indicator function of a semi-algebraic set is sub-analytic.

(d) Sub-analytic functions are closed under finite linear combinations, and the product of two sub-analytic functions is sub-analytic.

(e) Point-wise maximum and minimum of a finite collection of sub-analytic functions are sub-analytic.

(f) *Composition rule:* If g_1 and g_2 are two sub-analytic functions with the function g_1 being continuous, then the composition function $g_2 \circ g_1$ is sub-analytic. In fact, the class of continuous sub-analytic functions are *closed under algebraic operations.*

Appendix B. Proofs related to Algorithm 1

In this appendix, we collect the proofs of various results related to the gradient-based Algorithm 1, including Theorem 1, Corollaries 1 and 3, and Proposition 1.

B.1. Proof of Theorem 1

Our proof of this theorem, as well as subsequent ones, depends on the following descent lemma:

Lemma 4 *Under the conditions of Theorem 1, we have*

$$x^k \in \operatorname{int}(\mathcal{C}) \quad \textit{and} \quad f(x^{k+1}) \leq f(x^k) - \frac{\alpha}{2}\|\nabla f(x^k)\|_2^2 \quad \textit{for all } k = 0, 1, 2, \ldots. \tag{36}$$

See Appendix B.1.1 for the proof of this lemma.

We now prove Theorem 1 using Lemma 4.

Convergence of function values: We first prove that the function value sequence $\{f(x^k)\}_{k\geq 0}$ is convergent. Since $f^* := \min_{x\in\mathcal{C}} f(x)$ is finite by assumption, and $x^k \in \operatorname{int}(\mathcal{C})$ for all $k \geq 0$ by Lemma 4, the sequence $\{f(x^k)\}_{k\geq 0}$ is bounded below. For any non-stationary x^k, inequality (36) also ensures that $f(x^k) > f(x^{k+1})$; hence, there must exist some scalar $\bar{f}$ such that $\lim_{k\to\infty} f(x^k) = \bar{f}$.

Stationarity of limit points: Next, we establish that any limit point of the sequence $\{x^k\}_{k\geq 0}$ must be stationary. Consider a subsequence $\{x^{k_j}\}_{j\geq 0}$ of $\{x^k\}_{k\geq 0}$ such that $x^{k_j} \to \bar{x}$, and let $\{u^{k_j}\}_{j\geq 0}$ be the associated sequence of subgradients. It suffices to exhibit a subgradient $\bar{u} \in \partial h(\bar{x})$ such that $\nabla g(\bar{x}) - \bar{u} = 0$. Since the sequence $\{x^{k_j}\}_{j\geq 0}$ converges to $\bar{x}$, we must have

$$\|\nabla f(x^{k_j})\|_2 = \|\nabla g(x^{k_j}) - u^{k_j}\|_2 \to 0.$$

The function g is continuously differentiable by assumption, and we have $\nabla g(x^{k_j}) \to \nabla g(\bar{x})$. Combining these we find that $u^{k_j} \to \nabla g(\bar{x})$. Furthermore, by continuity of the function g, we have $g(x^{k_j}) \to g(\bar{x})$. Putting together the pieces we have established above that $\big(x^{k_j}, u^{k_j}, g(x^{k_j})\big) \to \big(\bar{x}, \bar{u}, g(\bar{x})\big)$, where $\bar{u} := \nabla g(\bar{x})$. Consequently, the graph continuity of limiting-sub-differentials (see Appendix A.1) guarantees that $\bar{u} = \nabla g(\bar{x}) \in \partial h(\bar{x})$. Overall, we conclude that $\nabla f(\bar{x}) := \nabla g(x) - \bar{u} = 0$, so that $\bar{x}$ is a stationary point as claimed.

Establishing the bound (3)**:** Finally, we prove the claimed bound (3) on the averaged squared gradient. Recalling that $f^* := \min_{x\in\mathcal{C}} f(x)$ is finite, we have

$$\begin{aligned} f(x^0) - f^* \geq f(x^0) - f(x^{k+1}) &= \sum_{j=0}^{k} f(x^j) - f(x^{j+1}) \\ &\overset{(i)}{\geq} \frac{\alpha}{2}\sum_{j=0}^{k} \|\nabla f(x^k)\|_2^2 \\ &= \frac{\alpha(k+1)}{2}\operatorname{Avg}\left(\|\nabla f(x^k)\|_2^2\right), \end{aligned}$$

where step (i) follows from equation (36). Rearranging yields the claimed bound (3) on the averaged squared gradient.

B.1.1. Proof of Lemma 4

Recall that by assumption, the function g is continuously differentiable and M_g-smooth, and the function h is convex. As a consequence, for any vector $x^k \in \mathcal{C}$ and subgradient $u^k \in \partial h(x^k)$, we have

$$g(x) \leq g(x^k) + \langle \nabla g(x^k), x - x^k\rangle + \frac{M_g}{2}\|x - x^k\|_2^2 \tag{37a}$$

$$h(x) \geq h(x^k) + \langle u^k, x - x^k\rangle. \tag{37b}$$

Combining inequalities (37a) and (37b) yield

$$f(x) = g(x) - h(x) \leq f(x^k) + \langle \nabla g(x^k) - u^k, x - x^k\rangle + \frac{M_g}{2}\|x - x^k\|_2^2. \tag{38}$$

Substituting $x = x^{k+1} := x^k - \alpha\big(\nabla g(x^k) - u^k\big)$ in equation (38) and simplifying yields

$$\begin{aligned} f(x^k) - f(x^{k+1}) \geq \big(\frac{1}{\alpha} - \frac{M_g}{2}\big)\|x^{k+1} - x^k\|_2^2 &= \alpha\big(1 - \frac{\alpha M_g}{2}\big)\|\nabla g(x^k) - u^k\|_2^2 \\ &\overset{(i)}{\geq} \frac{\alpha}{2}\|\nabla f(x^k)\|_2^2, \end{aligned}$$

where inequality (i) follows from the upper bound $\alpha \leq \frac{1}{M_g}$. This proves the second part of the stated lemma. As for the claim that the sequence remains in the interior of the set $\mathcal{C}$, note that $f(x^{k+1}) \leq f(x^k) \leq f(x^0)$, which ensures that $x^{k+1} \in \mathcal{L}(f(x^0)) \subset \operatorname{int}(\mathcal{C})$, as claimed.

B.2. Proof of Corollary 1

The first part of the proof builds on a simple application of Theorem 1 and the definition of effective smoothness constant M_f^*. The second part of the proof utilizes a relation between the backtracking step size and the effective smoothness constant. For sake of completeness, we first describe the gradient descent backtracking algorithm.

Algorithm 4 Gradient descent with backtracking

1: Given an initial point $x^0 \in \operatorname{int}(\mathcal{C})$ and parameter $\beta \in (0, 1)$:
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Choose the smallest nonnegative integer i_k such that the step size $t^k := \beta^{i_k}$ satisfies:

$$f\big(x^k - t^k\nabla f(x^k)\big) \leq f(x^k) - \frac{t^k}{2}\|\nabla f(x^k)\|^2. \tag{39}$$

4: Update $x^{k+1} = x^k - t^k\nabla f(x^k)$.
5: **end for**

Establishing the bound in (5a)**:** For any step size α in the interval $\big(0, \frac{1}{M_{f^*}}\big)$, the definition of the effective smoothness constant M_{f^*} ensures the following property. There

exists a M_g-smooth function g and a convex-differentiable function h with $f = g - h$, and the scalar M_g satisfies $\alpha < \frac{1}{M_g} \leq \frac{1}{M_{f^*}}$. Since the function f is differentiable, applying Algorithm 1 on the function f with the decomposition $f = g - h$ is equivalent to applying gradient descent on f. Furthermore, the step size α satisfies the upper bound $\alpha \leq \frac{1}{M_g}$, and applying the bound (3) from Theorem 1 yields:

$$\text{Avg}\left(\|\nabla f(x^k)\|_2^2\right) \leq \frac{2\big(f(x^0) - f^*\big)}{\alpha(k+1)}. \tag{40}$$

Establishing the backtracking bound (5b)**:** For any fraction $\beta \in (0,1)$, the definition of the effective smoothness constant M_{f^*} guarantees the following. There exists a M_g-smooth function g and a convex and differentiable function h with $f = g - h$, and the scalar M_g satisfies $\beta M_g \leq M_{f^*} \leq M_g$. Comparing the descent step (36) from Lemma 4 and step (39) in Algorithm 4, we conclude that the step size t^k satisfies the lower bound $t^k \geq \min\left\{1, \frac{\beta}{M_g}\right\} \geq \min\left\{1, \frac{\beta^2}{M_f^*}\right\}$. Applying the descent step (39) in Algorithm 4 repeatedly and then utilizing the last lower bound on step size t^k, we find that for all $k = 0, 1, 2 \ldots$

$$f(x^0) - f(x^{k+1}) \geq \sum_{i=0}^{k} \frac{t^k}{2}\|\nabla f(x^k)\|^2 \geq \min\left\{\frac{1}{2}, \frac{\beta^2}{2M_f^*}\right\} \sum_{i=0}^{k} \|\nabla f(x^k)\|^2.$$

Rearranging the last inequality yields:

$$\begin{aligned}\text{Avg}\left(\|\nabla f(x^k)\|^2\right) &\leq \frac{2\max\left\{1, \frac{M_f^*}{\beta^2}\right\}\big(f(x^0) - f(x^{k+1})\big)}{(k+1)} \\ &\overset{(i)}{\leq} \frac{2\max\left\{1, M_f^*\right\}\big(f(x^0) - f^*\big)}{\beta^2(k+1)},\end{aligned} \tag{41}$$

where step (i) follows since $\beta \in (0,1)$, along with the lower bound $f(x^{k+1}) \geq f^*$.

B.3. Proof of Corollary 3

Based on Theorem 4 of (Lee et al., 2016), it suffices to show that the gradient map $G(x) := x - \alpha\nabla f(x)$ is a diffeomorphism for any step size $\alpha \in \big(0, \frac{1}{M_g}\big)$. Recall that a map $G : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a diffeomorphism if the map G is a bijection, and both the maps G and G^{-1} are continuously differentiable.

Injectivity: We first prove that G is an injective map. Consider a pair of vectors x, y such that $G(x) = G(y)$; our aim is to prove that $x = y$. The condition $G(x) = G(y)$ is equivalent to $x - y = \alpha\big(\nabla f(x) - \nabla f(y)\big)$, and we have that

$$\begin{aligned}\|x - y\|_2^2 &= \alpha\langle x - y, \nabla f(x) - \nabla f(y)\rangle \\ &= \alpha\langle x - y, \nabla g(x) - \nabla g(y)\rangle - \alpha\langle x - y, \nabla h(x) - \nabla h(y)\rangle \\ &\overset{(i)}{\leq} \alpha M_g\|x - y\|_2^2 - \alpha\langle x - y, \nabla h(x) - \nabla h(y)\rangle \\ &\overset{(ii)}{\leq} \alpha M_g\|x - y\|_2^2.\end{aligned}$$

Here inequality (i) follows because the gradient ∇g is M_g-Lipschitz by assumption; inequality (ii) follows from the convexity of the function h, which implies the monotonicity of the gradient ∇h. Finally, since the step size $\alpha < \frac{1}{M_g}$ by assumption, the inequality $\|x - y\|_2^2 \leq \alpha M_g\|x - y\|_2^2$ can hold only when $x = y$.

Surjectivity: For any fixed vector $y \in \mathbb{R}^d$, consider the following problem

$$\arg\min_{x \in \mathbb{R}^d} \left\{\frac{1}{2}\|x - y\|_2^2 - \alpha g(x) + \alpha h(x)\right\}. \tag{42}$$

Observe that for any step size $\alpha \in \big(0, \frac{1}{M_g}\big)$ and any fixed vector $y \in \mathbb{R}^d$, the map $x \mapsto \frac{1}{2}\|x - y\|_2^2 - \alpha g(x)$ is strongly convex, whence the map $x \mapsto \frac{1}{2}\|x - y\|_2^2 - \alpha g(x) + \alpha h(x)$ is also strongly convex. Consequently, the convex problem (42) has a unique minimizer, and we denote it by x_y. In order to prove surjectivity of the map G, it suffices to show the point x_y is mapped to the point y. Recalling the KKT conditions of the problem (42), we have that

$$y = x_y - \alpha\nabla f(x_y) = G(x_y),$$

which completes the proof of surjectivity of the map G.

Combining the injectivity and the surjectivity of the map G, we conclude that the inverse map G^{-1} exists. Next, let $DG(\cdot)$ denote the Jacobian of the map G, then $DG(x) = \mathrm{I} - \alpha\nabla^2 g(x) + \alpha\nabla^2 h(x)$. Since the function g is M_g-smooth, and the map G is continuously differentiable, standard application of the inverse-function theorem guarantees that for all step size $\alpha < \frac{1}{M_g}$, the inverse map G^{-1} is continuously differentiable. Putting together the pieces, we conclude that map G^{-1} exists, and both the maps (G, G^{-1}) are continuously differentiable. Overall, we have established that the map G is a diffeomorphism, as claimed.

B.4. Proof of Proposition 1

The CCCP update at step $(k+1)$ is given by $x^{k+1} = \arg\min_{x \in \mathcal{C}} q(x, x^k)$, where

$$q(x, x^k) := g(x) - h(x^k) - \langle \nabla h(x^k), x - x^k\rangle. \tag{43}$$

Observe that step $(k+1)$ of Algorithm 1 is equivalent to a gradient descent update with step size α on the map $x \mapsto q(x, x^k)$. Accordingly, if we define $y^{k+1} = x^k - \alpha\nabla q(x, x^k)$, then we have $q(y^{k+1}, x^k) \geq q(x^{k+1}, x^k)$; moreover

$$\begin{aligned}f(x^k) - f(x^{k+1}) &\overset{(i)}{\geq} q(x^k, x^k) - q(x^{k+1}, x^k) \\ &\overset{(ii)}{\geq} q(x^k, x^k) - q(y^{k+1}, x^k) \\ &\overset{(iii)}{\geq} \frac{1}{2M_g}\|\nabla f(x^k)\|_2^2.\end{aligned} \tag{44}$$

Here inequality (i) follows from the equality $q(x^k, x^k) = f(x^k)$ combined with the lower bound $q(x, x^k) \geq f(x)$. Inequality (ii) follows since $q(y^{k+1}, x^k) \geq q(x^{k+1}, x^k)$, and inequality

(iii) follows from Lemma 4 with step size $\alpha = \frac{1}{M_g}$. Note that equation (44) guarantees that the function value sequence $\{f(x^k)\}_{k\geq 0}$ is decreasing. Since the function f is bounded below, we have that the sequence $\{f(x^k)\}_{k\geq 0}$ converges. In order to prove that all limit points of the sequence $\{x^k\}_{k\geq 0}$ are critical points, we follow the corresponding argument in proof of Theorem 1. This completes the proof of part (a) in Proposition 1.

Turning to part (b), unwrapping the recursive lower bound (44) and re-arranging yields inequality (11a). Finally, we turn to the proof of inequality (11b) under the additional strong convexity condition. Under this condition, the map $x \mapsto q(x, x^k)$ in equation (43) is μ-strongly convex, so that

$$f(x^k) - f(x^{k+1}) \geq q(x^k, x^k) - q(x^{k+1}, x^k) \overset{(i)}{\geq} \frac{\mu}{2}\|x^k - x^{k+1}\|_2^2, \tag{45}$$

where inequality (i) follows from the strong convexity of the map $x \mapsto q(x, x^k)$ and the fact that $\nabla q(x^{k+1}, x^k) = 0$. Using this equation repeatedly, we find that

$$\begin{aligned} f(x^0) - f^* \geq f(x^0) - f(x^{k+1}) &= \sum_{j=0}^{k} \left\{ f(x^j) - f(x^{j+1}) \right\} \\ &\geq \frac{\mu}{2} \sum_{j=0}^{k} \|x^j - x^{j+1}\|_2^2 \\ &= \frac{\mu(k+1)}{2} \mathrm{Avg}\left(\|x^k - x^{k+1}\|_2^2 \right). \end{aligned}$$

Rearranging the last inequality yields the bound (11b). Finally, let us reiterate that bounds similar to (11b) are known in the literature; see the paper (Lanckriet and Sriperumbudur, 2009) for example. We provide the proof of bound (11b) for completeness.

Appendix C. Proof of Theorem 2

This proof shares some important steps with Theorem 1, but it requires a more refined argument due to the presence of a non-smooth and non-continuous function φ. We start by stating an auxiliary lemma that underlies the proof of Theorem 2. In the proof, the subgradients of the convex functions h and φ at a point x^k are denoted by u^k and v^k, respectively.

Lemma 5 *Under the conditions of Theorem 2, we have*

$$x^{k+1} = x^k - \alpha(\nabla g(x^k) + v^{k+1} - u^k), \qquad \textit{and} \tag{46a}$$

$$f(x^k) - f(x^{k+1}) \geq \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2, \tag{46b}$$

valid for all $k = 0, 1, 2, \ldots$. Furthermore, for any convergent subsequence $\{x^{k_j}\}_{j\geq 0}$ of the sequence $\{x^k\}_{k\geq 0}$ with $x^{k_j} \to \bar{x}$, we have

$$\lim_{j\to\infty} \varphi(x^{k_j+1}) = \varphi(\bar{x}).$$

See Appendix C.1 for the proof of this lemma.

We now prove Theorem 2 using Lemma 5.

Convergence of function value: We first prove that the sequence of function values $\{f(x^k)\}_{k\geq 0}$ is convergent. Since $f^* := \min_{x\in\mathbb{R}^d} f(x)$ is finite by assumption, the sequence $\{f(x^k)\}_{k\geq 0}$ is bounded below. If $x^k = x^{k+1}$ for some k, the convergence of the sequence $\{f(x^k)\}_{k\geq 0}$ is trivial. Hence, we may assume without loss of generality that $x^k \neq x^{k+1}$ for all $k = 0, 1, 2, \ldots$ In that case, inequality (46b) ensures that $f(x^k) > f(x^{k+1})$, and consequently, there must exist some scalar $\bar{f}$ such that $\lim_{k\to\infty} f(x^k) = \bar{f}$.

Stationarity of limit points: Next, we establish that any limit point of the sequence $\{x^k\}_{k\geq 0}$ must be stationary. Consider a subsequence $\{x^{k_j}\}_{j\geq 0}$ such that $x^{k_j} \to \bar{x}$. Let $\{v^{k_j}\}_{j\geq 0}$ and $\{u^{k_j}\}_{j\geq 0}$ be the associated sequence of subgradients. It suffices to exhibit subgradients $\bar{v} \in \partial\varphi(\bar{x})$ and $\bar{u} \in \partial h(\bar{x})$ such that, $\nabla g(\bar{x}) + \bar{v} - \bar{u} = 0$.

Step 1: *Existence of subgradient* $\bar{u}$: Since the sequence $\{x^{k_j}\}_{j\geq 0}$ is convergent, we may assume that the sequence $\{x^{k_j}\}_{j\geq 0}$ is bounded, and it lies in a compact set S. The function h is convex continuous, and we have that $h(x^{k_j}) \to h(\bar{x})$, and the subgradient sequence $\{u^{k_j}\}_{j\geq 0}$ is bounded; see example 9.14 in the book (Rockafellar and Wets, 2009). Passing to a subsequence if necessary, we may assume that the sequence $\{u^{k_j}\}_{j\geq 0}$ converges to $\bar{u}$. Putting together these pieces, we conclude that $(x^{k_j}, u^{k_j}, h(x^{k_j})) \to (\bar{x}, \bar{u}, h(\bar{x}))$ as $j \to \infty$; consequently, the graph continuity of limiting sub-differentials guarantees that $\bar{u} \in \partial h(\bar{x})$ (see Appendix A.1 for graph continuity).

Step 2: *Existence of subgradient* $\bar{v}$: In order to complete the proof, it suffices to show that the vector $\bar{v} := -\nabla g(\bar{x}) + \bar{u}$ belongs to the subgradient set $\partial\varphi(\bar{x})$. Since the norm of successive difference $\|x^{k_j} - x^{k_j+1}\|_2$ converges to zero, Lemma 5 yields $\|\nabla g(x^{k_j}) + v^{k_j+1} - u^{k_j}\|_2 \to 0$, and $x^{k_j+1} \to \bar{x}$. Furthermore, continuity of the gradient ∇g yields $\nabla g(x^{k_j}) \to \nabla g(\bar{x})$, and step 1 above guarantees $u^{k_j} \to \bar{u}$. Combining these two facts with $\|\nabla g(x^{k_j}) + v^{k_j+1} - u^{k_j}\|_2 \to 0$, we obtain $v^{k_j+1} \to \bar{v} := -\nabla g(\bar{x}) + \bar{u}$, and by Lemma 5, we have $\varphi(x^{k_j+1}) \to \varphi(\bar{x})$. Putting together the pieces, we conclude that $(x^{k_j+1}, v^{k_j+1}, \varphi(x^{k_j+1})) \to (\bar{x}, \bar{v}, \varphi(\bar{x}))$. Consequently, the graph continuity of limiting subdifferentials guarantees that $\bar{v} \in \partial\varphi(\bar{x})$ (see Appendix A.1 for graph continuity).

Finally, the subgradients $\bar{u} \in \partial h(\bar{x})$ and $\bar{v} \in \partial\varphi(\bar{x})$ obtained from steps 1 and and 2 respectively satisfy the relation $\nabla g(\bar{x}) + \bar{v} - \bar{u} = 0$, which establishes the claimed stationarity of $\bar{x}$.

Establishing the bound (13a)**:** Next, we establish the claimed bound (13a) on the averaged squared successive difference. Recalling that $f^* := \min_{x\in\mathbb{R}^d} f(x)$ is finite, we have

$$\begin{aligned} f(x^0) - f^* \geq f(x^0) - f(x^{k+1}) &= \sum_{j=0}^{k} f(x^j) - f(x^{j+1}) \\ &\overset{(i)}{\geq} \frac{1}{2\alpha}\sum_{j=0}^{k} \|x^j - x^{j+1}\|_2^2 \\ &= \frac{(k+1)}{2\alpha}\mathrm{Avg}\left(\|x^k - x^{k+1}\|_2^2\right), \end{aligned} \tag{47}$$

where step (i) follows from equation (46b). Rearranging the last inequality yields the claimed bound (13a) on the averaged squared successive difference.

Establishing the bound (13b)**:** In order to establish the bound (13b) on the averaged squared gradient, we start by establishing the following upper bound on the gradient-norm $\|\nabla f(x^{k+1})\|_2$:

$$\|\nabla f(x^{k+1})\|_2 \leq \big(M_g + M_h + \frac{1}{\alpha}\big)\|x^k - x^{k+1}\|_2. \tag{48}$$

Recall that the function h is M_h smooth by assumption, and we have

$$\begin{aligned} \|\nabla g(x^{k+1}) - \nabla h(x^{k+1}) + v^{k+1}\|_2 &\overset{(i)}{=} \|\nabla g(x^{k+1}) - \nabla h(x^{k+1}) + \big(\nabla h(x^k) - \nabla g(x^k) + \frac{1}{\alpha}(x^k - x^{k+1})\big)\|_2 \\ &\overset{(ii)}{\leq} \|\nabla g(x^k) - \nabla g(x^{k+1})\|_2 + \|\nabla h(x^k) - \nabla h(x^{k+1})\|_2 + \frac{1}{\alpha}\|x^k - x^{k+1}\|_2 \\ &\overset{(iii)}{\leq} \big(M_g + M_h + \frac{1}{\alpha}\big)\|x^k - x^{k+1}\|_2. \end{aligned}$$

Here step (i) follows from the update equation of x^{k+1} in Lemma 5 and from differentiability of the function g; step (ii) follows from triangle inequality, and step (iii) follows from the smoothness of the functions g and h. Putting together the bounds (48) and (47), we obtain the desired bound (13b).

C.1. Proof of Lemma 5

Here we prove the claims of Lemma 5.

Establishing update equation (46a)**:** Recalling the convex majorant defined in equation (38), we define a convex majorant $q(\cdot, x^k)$ of the function f as follows:

$$q(x, x^k) = g(x^k) - h(x^k) + \langle \nabla g(x^k) - u^k,\, x - x^k\rangle + \frac{1}{2\alpha}\|x - x^k\|_2^2 + \varphi(x), \tag{49}$$

where subgradient $u^k \in \partial h(x^k)$, and the step size α satisfies $0 < \alpha \leq \frac{1}{M_g}$. Observe that minimizer of the convex function $x \mapsto q(x, x^k)$ over $x \in \mathbb{R}^d$ is same as $\mathrm{prox}^{\varphi}_{1/\alpha}\big(x^k - \alpha(\nabla g(x^k) - u^k)\big)$, which implies that x^{k+1} is a minimizer of the convex function $x \mapsto q(x, x^k)$ over $x \in \mathbb{R}^d$. Consequently, the optimality condition of x^{k+1} guarantees that there exists subgradient $v^{k+1} \in \partial g(x^{k+1})$ satisfying the following equation:

$$\nabla g(x^k) - u^k + v^{k+1} + \frac{1}{\alpha}(x^{k+1} - x^k) = 0. \tag{50}$$

Rewriting the above equation yields the update equation (46a).

Establishing the descent step (46b)**:** Note that

$$\begin{aligned} f(x^k) - q(x^{k+1}, x^k) &\overset{(i)}{\geq} g(x^k) - h(x^k) + \varphi(x^{k+1}) + \langle v^{k+1},\, x^k - x^{k+1}\rangle - q(x^{k+1}, x^k) \\ &\overset{(ii)}{\geq} \langle \nabla g(x^k) - u^k + v^{k+1},\, x^k - x^{k+1}\rangle - \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2 \\ &\overset{(iii)}{\geq} \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2. \end{aligned} \tag{51}$$

Here step (i) follows from the convexity of the function φ; step (ii) follows by substituting $q(x^{k+1}, x^k)$ from equation (49). In step (iii), we use the relation $\nabla g(x^k) - u^k + v^{k+1} = \frac{1}{\alpha}(x^k - x^{k+1})$, which follows from equation (50). Finally, recall that the function $x \mapsto q(x, x^k)$ is a majorant for the function f, and we deduce that

$$\begin{aligned} f(x^k) - f(x^{k+1}) &\geq f(x^k) - q(x^{k+1}, x^k) \\ &\geq \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2. \end{aligned} \tag{52}$$

Limit of the sequence $\{\varphi(x^{k_j+1})\}_{j\geq 0}$**:** Consider any convergent subsequence $\{x^{k_j}\}_{j\geq 0}$ of the sequence $\{x^k\}_{k\geq 0}$ with $x^{k_j} \to \bar{x}$. Recall that $f^* = \inf_{x\in\mathbb{R}^d} f(x)$ is finite by assumption; combining this with step (46b) in Lemma 5, we have that $\|x^k - x^{k+1}\|_2 \to 0$, and that $x^{k_j+1} \to \bar{x}$. The function φ is lower semi-continuous, and we have

$$\liminf_{j\to\infty}\ \varphi(x^{k_j+1}) \geq \varphi(\bar{x}). \tag{53}$$

Since we already proved x^{k_j+1} is a minimizer of the convex function $x \mapsto q(x, x^{k_j})$, we have $q(x^{k_j+1}, x^{k_j}) \leq q(\bar{x}, x^{k_j})$. Unwrapping the last inequality and taking lim sup yields

$$\begin{aligned} \limsup_{j\to\infty}\ \varphi(x^{k_j+1}) &\overset{(i)}{\leq} \varphi(\bar{x}) + \limsup_{j\to\infty}\ \Big(\langle \bar{x} - x^{k_j},\, \nabla g(x^{k_j}) - u^{k_j}\rangle + \frac{1}{2\alpha}\|x^{k_j} - \bar{x}\|_2^2\Big) \\ &\overset{(ii)}{=} \varphi(\bar{x}). \end{aligned} \tag{54}$$

Here step (i) holds since $\|x^{k_j} - x^{k_j+1}\|_2 \to 0$, and the sequence $\{\nabla g(x^{k_j})\} - u^{k_j}\}_{j\geq 0}$ is bounded—which we prove shortly; step (ii) above follows from $x^{k_j} \to \bar{x}$ and boundedness of the sequence $\{\nabla g(x^{k_j}) - u^{k_j}\}_{j\geq 0}$. Combining equations (53) and (54) we obtain the claimed result.

Boundedness of the sequence $\{\nabla g(x^{k_j}) - u^{k_j}\}_{j\geq 0}$: In order to prove the boundedness of the sequence $\{\nabla g(x^{k_j}) - u^{k_j}\}_{j\geq 0}$, it suffices to show that the gradient sequence $\{\nabla g(x^{k_j})\}_{j\geq 0}$ and the sub-gradient sequence $\{u^{k_j}\}_{j\geq 0}$ are bounded. Recall that $x^{k_j} \to \bar{x}$, and we have that the sequence $\{x^{k_j}\}_{j\geq 0}$ is bounded. Consequently, from the smoothness of the function g, we find that the sequence $\{\nabla g(x^{k_j})\}_{j\geq 0}$ is bounded. Finally, note that the function h is convex continuous, and we already argued that the sequence $\{x^{k_j}\}_{j\geq 0}$ is bounded. Combining this with example 9.14 in the book (Rockafellar and Wets, 2009), we conclude that the subgradient sequence $\{u^{k_j}\}_{j\geq 0}$ bounded.

Appendix D. Proofs related to Algorithm 3

In this appendix, we provide the proof of Theorem 3, which applies to the Frank-Wolfe based method (Algorithm 3). We also provide an upper bound on the generalized curvature constant $\mathcal{C}_f$, which is stated in Lemma 6.

D.1. Proof of Theorem 3

Let $x^\gamma := x^k + \gamma d^k$, where the difference d^k is defined as $d^k := s^k - x^k$, and the vector s^k is the Frank-Wolfe direction defined in Algorithm 3. Unpacking the definition (18) of the generalized curvature constant $\mathcal{C}_f$, we find that for any scalar $\gamma \in (0,1)$ and subgradient $u^k \in \partial h(x^k)$, we have the following:

$$\begin{aligned} f(x^\gamma) &\leq f(x^k) + \gamma\langle \nabla g(x^k) - u^k,\, d^k\rangle + \frac{\gamma^2}{2}\mathcal{C}_f \\ &\overset{(i)}{\leq} f(x^k) - \gamma g^k + \frac{\gamma^2}{2} C_0. \end{aligned} \tag{55}$$

Here inequality (i) is obtained by substituting $g^k = \langle d^k,\, u^k - \nabla g(x^k)\rangle$ and using $C_0 \geq \mathcal{C}_f$. Substituting $\gamma = \gamma^k := \min\{\frac{g^k}{C_0}, 1\}$ in equation (55) yields

$$f(x^{k+1}) \leq f(x^k) - \min\{\frac{(g^k)^2}{2C_0}, g^k - \frac{C_0}{2}\mathbb{1}_{\{g^k > C_0\}}\}, \tag{56}$$

where $x^{k+1} = x^k + \gamma^k d^k$. Let $\bar{g}^k := \min_{0\leq j\leq k} g^j$ denote the minimum FW gap up to iteration k, then repeated application of equation (56) yields

$$\begin{aligned} f(x^0) - f(x^{k+1}) &\geq \sum_{j=0}^{k} \min\{\frac{(g^j)^2}{2C_0}, g^j - \frac{C_0}{2}\mathbb{1}_{\{g^j > C_0\}}\} \\ &\geq (k+1)\min\{\frac{(\bar{g}^k)^2}{2C_0}, \bar{g}^k - \frac{C_0}{2}\mathbb{1}_{\{\bar{g}^k > C_0\}}\}. \end{aligned} \tag{57}$$

Rewriting the last equation yields the following upper bound

$$\min\{\frac{(\bar{g}^k)^2}{2C_0}, \bar{g}^k - \frac{C_0}{2}\mathbb{1}_{\{\bar{g}^k > C_0\}}\} \overset{(i)}{\leq} \frac{f(x^0) - f^*}{k+1},$$

where step (i) follows from the lower bound $f(x^{k+1}) \geq f^* := \min_{x\in\mathcal{C}} f(x)$. Considering the cases where $\bar{g}^k \leq C_0$ and $\bar{g}^k > C_0$ separately, it can be shown following (Lacoste-Julien, 2016) that

$$\bar{g}^k \leq \begin{cases} \frac{2(f(x^0)-f^*)}{\sqrt{k+1}} & \text{for } k+1 \leq \frac{2(f(x^0)-f^*)}{C_0} \\ \sqrt{\frac{2C(f(x^0)-f^*)}{k+1}} & \text{otherwise}. \end{cases}$$

Finally, note that $\sqrt{2C_0(f(x^0) - f^*)} \leq \max\{2(f(x^0) - f^*), C_0\}$ and we conclude that

$$\bar{g}^k \leq \frac{\max\{2\big(f(x^0) - f^*\big), C_0\}}{\sqrt{k+1}}.$$

D.2. Upper bound on generalized curvature constant

In this section, we provide an upper bound on the generalized curvature constant $\mathcal{C}_f$, where the function f is a difference of a differentiable function g and a continuous function h. For better readability, we use $\mathcal{C}_{g-h}$ instead of $\mathcal{C}_f$ in the following lemma.

Lemma 6 *Suppose that the function g is continuously differentiable and function h is convex, then we have $\mathcal{C}_{g-h} \leq \mathcal{C}_g$. Furthermore, if the function g is M_g-smooth, and the function h is a μ strongly convex function with $0 \leq \mu < M$, then*

$$\mathcal{C}_{g-h} \leq \big(M - \mu\big) \times \big(\operatorname{diam}_{\|\cdot\|_2}(\mathcal{C})\big)^2, \tag{58}$$

where $\operatorname{diam}_{\|\cdot\|_2}$ denote the diameter of the set $\mathcal{C}$, measured in ℓ_2 norm.

Comments: The first upper bound on $\mathcal{C}_{g-h}$ in Lemma 6 posits that the curvature constant of the difference function $g - h$ is upper bounded by curvature constant of the function g, whenever the second function h is convex. Let us try to understand an implication of this result through an example. One of the well-known upper bound of curvature constant for M_g-smooth function g is $M_g \times \big(\operatorname{diam}_{\|\cdot\|_2}(\mathcal{C})\big)^2$; see the paper by (Jaggi, 2013). Now consider continuously differentiable functions g and h such that the function g is M_g-smooth and the function h is non-smooth and convex. It can be verified that the difference function $g - h$ is *not* smooth in this case; consequently, the earlier bound on curvature constant $\mathcal{C}_{g-h}$ is ∞, whereas Lemma 6 ensures that

$$\mathcal{C}_{g-h} \leq \mathcal{C}_g \leq M_g \times \big(\operatorname{diam}_{\|\cdot\|_2}(\mathcal{C})\big)^2.$$

Proof of the upper bound $\mathcal{C}_{g-h} \leq \mathcal{C}_g$: Unwrapping the definition of $\mathcal{C}_{g-h}$, we have

$$\begin{aligned} \mathcal{C}_{g-h} &= \sup_{\substack{x,y\in c_\gamma \\ u\in\partial h(x)}} \frac{2}{\gamma^2}\big[f(y) - f(x) - \langle y - x,\, \nabla g(x) - u\rangle\big] \\ &= \sup_{\substack{x,y\in c_\gamma \\ u\in\partial h(x)}} \frac{2}{\gamma^2}\big[g(y) - g(x) - \langle y - x,\, \nabla g(x)\rangle - \Delta_h(y,x,u)\big] \\ &\overset{(i)}{\leq} \underbrace{\sup_{x,y\in c_\gamma} \frac{2}{\gamma^2}\big[f(y) - f(x) - \langle y - x,\, \nabla g(x)\rangle\big]}_{\mathcal{C}_g}, \end{aligned} \tag{59}$$

where $\Delta_h(y,x,u) := h(y) - h(x) - \langle y-x, u\rangle$. Here inequality (i) follows by noting that, for any pair of points $x, y \in \mathcal{C}$, and for any convex function h with $u \in \partial h(x)$, we have $\Delta_h(y,x,u) \geq 0$.

Proof of upper bound (58)**:** Suppose in addition, the function g is M_g-smooth, and the function h is μ-strongly convex with $\mu \geq 0$. Then we have $\Delta_h(y,x,u) \geq \frac{\mu}{2}\|x-y\|_2^2$, and equation (59) yields

$$\begin{aligned}\mathcal{C}_{g-h} &\leq \sup_{x,y\in c_\gamma} \frac{2}{\gamma^2}\big[g(y) - g(x) - \langle y-x, \nabla g(x)\rangle - \frac{\mu}{2}\|x-y\|_2^2\big] \\ &\overset{(i)}{\leq} \sup_{x,y\in c_\gamma} \frac{2}{\gamma^2}\left[\frac{M_g-\mu}{2}\|x-y\|_2^2\right],\end{aligned}$$

where step (i) follows since the function g is M_g-smooth. Substituting $y - x = \gamma s$ with $s \in \mathcal{C}$, we obtain the claimed upper bound

$$\mathcal{C}_{g-h} \leq (M_g - \mu) \times \big(\operatorname{diam}_{\|\cdot\|_2}(\mathcal{C})\big)^2.$$

Appendix E. Proofs of faster rates under Assumption KL

In this appendix, we prove our results on improved convergence rates for functions which satisfy Assumption KL—as stated in Theorems 4 and 5. We begin by stating an auxiliary lemma that underlies the proofs of Theorems 4 and 5.

Lemma 7 *Under assumptions of either Theorem 4 or Theorem 5, there exists constants $\theta \in [0,1)$, $C > 0$ and positive integer k_1 such that for all $k \geq k_1$, we have*

$$|f(x^k) - \bar{f}|^\theta \leq C\|\nabla f(x^k)\|_2,$$

where $f(x^k) \downarrow \bar{f}$. Furthermore, if $x^k \to \bar{x}$, then the parameters (θ, C), obtained from KL-inequality of the function f at the point $\bar{x}$, satisfy the above inequality.

See Appendix E.3 for the proof of this lemma.

E.1. Proof of Theorem 4

Now we prove Theorem 4 using Lemma 7.

Convergence of the sequence $\{x^k\}_{k\geq 0}$: We demonstrate the convergence of the sequence $\{x^k\}_{k\geq 0}$ by proving that the sequence has finite length property; more precisely, we show that $\sum_{k=0}^{\infty}\|x^k - x^{k+1}\|_2 < \infty$. First, note that for any scalar $0 \leq \theta < 1$, the function $t \mapsto t^{1-\gamma\theta}$ is concave for $0 < \gamma < \frac{1}{\theta}$; consequently, for iteration $k \geq k_1$ we have

$$\begin{aligned}\big(f(x^k) - \bar{f}\big)^{1-\gamma\theta} - \big(f(x^{k+1}) - \bar{f}\big)^{1-\gamma\theta} &\geq (1-\gamma\theta)\big(f(x^k) - \bar{f}\big)^{-\gamma\theta}\big(f(x^k) - f(x^{k+1})\big) \\ &\overset{(i)}{\geq} (1-\gamma\theta)\big(|f(x^k) - \bar{f}|\big)^{-\gamma\theta} \times \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2 \\ &\overset{(ii)}{\geq} \frac{(1-\gamma\theta)}{C\|\nabla f(x^k)\|_2^\gamma} \times \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2 \\ &\overset{(iii)}{=} \frac{(1-\gamma\theta)}{2C\alpha^{1-\gamma}}\|x^k - x^{k+1}\|_2^{2-\gamma}. \qquad (60)\end{aligned}$$

Here inequality (i) follows from the descent property in equation (36) and from the fact that $f(x^k) \downarrow \bar{f}$. Inequality (ii) follows from Lemma 7, and equality (iii) follows from the relation $x^k - x^{k+1} = \alpha\big(\nabla g(x^k) - u^k\big) = \alpha\nabla f(x^k)$. Substituting $\gamma = 1$ and summing both side of inequality (60) from index $k = k_1$ to $k = \infty$, we obtain

$$\begin{aligned}\big(f(x^{k_1}) - \bar{f}\big)^{1-\theta} &= \sum_{k=k_1}^{\infty}\big(f(x^k) - \bar{f}\big)^{1-\theta} - \big(f(x^{k+1}) - \bar{f}\big)^{1-\theta} \\ &\geq \sum_{k=k_1}^{\infty}\frac{(1-\theta)}{2C}\|x^k - x^{k+1}\|_2,\end{aligned}$$

which proves the finite length property of the sequence $\{x^k\}_{k\geq 0}$. Consequently, we are guaranteed to have a vector $\bar{x}$ such that $x^k \to \bar{x}$ as $k \to \infty$.

Rate of convergence of Avg $\big(\|\nabla f(x^k)\|_2\big)$: Rewriting equation (60), we have the following:

$$\begin{aligned}C_\gamma &:= \sum_{\ell=0}^{k_1}\frac{(1-\gamma\theta)}{2C\alpha^{1-\gamma}}\|x^\ell - x^{\ell+1}\|_2^{2-\gamma} + \big(f(x^{k_1}) - \bar{f}\big)^{(1-\gamma\theta)} \\ &\overset{(i)}{\geq} \sum_{\ell=0}^{k-1}\frac{(1-\gamma\theta)}{2C\alpha^{1-\gamma}}\|x^\ell - x^{\ell+1}\|_2^{2-\gamma} \\ &= \frac{k(1-\gamma\theta)}{2C\alpha^{1-\gamma}}\operatorname{Avg}\left(\|x^k - x^{k+1}\|_2^{2-\gamma}\right), \qquad (61)\end{aligned}$$

where step (i) above follows from equation (60), and $\operatorname{Avg}\left(\|x^k - x^{k+1}\|_2^{2-\gamma}\right) := \frac{1}{k}\sum_{\ell=0}^{k-1}\|x^\ell - x^{\ell+1}\|_2^{2-\gamma}$ denote the running arithmetic average. Since $0 \leq \theta < 1$, we can take $\gamma = 1$ in equation (61), and we obtain the following rate:

$$\operatorname{Avg}\left(\|\nabla f(x^k)\|_2\right) = \frac{1}{\alpha}\operatorname{Avg}\left(\|x^k - x^{k+1}\|_2\right) \leq \frac{c_1}{k},$$

where $c_1 = \frac{2CC_\gamma}{\alpha(1-\theta)}$. Finally, note that the last equality holds trivially for iteration $k \leq k_1$ with the given choice of the constant c_1.

Rate of convergence of GAvg $\big(\|\nabla f(x^k)\|_2\big)$: Since we proved that the sequence $\{x^k\}_{k\geq 0}$ is convergent to the point $\bar{x}$, we have that the parameter θ in Lemma 7 can be taken to be the KL-exponent of the function f at point $\bar{x}$. Suppose $\frac{1}{2} \leq \theta < \frac{r}{2r-1}$, then substituting $\gamma = \frac{2r-1}{r}$ in equation (61) yields,

$$\begin{aligned}\operatorname{GAvg}\left(\|\nabla f(x^k)\|_2\right) &= \frac{1}{\alpha}\operatorname{GAvg}\left(\|x^k - x^{k+1}\|_2\right) \\ &\overset{(i)}{\leq} \frac{1}{\alpha}\{\operatorname{Avg}\left(\|x^k - x^{k+1}\|_2^{\frac{1}{r}}\right)\}^r \\ &\overset{(ii)}{\leq} \frac{c_2}{k^r},\end{aligned}$$

where $c_2 = \frac{1}{\alpha}\big(\frac{2CC_\gamma \alpha^{1-\gamma\theta}}{1-\gamma\theta}\big)^r$ with $\gamma = \frac{2r-1}{r}$, and $\text{GAvg}\Big(\|x^k - x^{k+1}\|_2^{2-\gamma}\Big) := \prod_{\ell=0}^{k-1}\big(\|x^\ell - x^{\ell+1}\|_2\big)^{\frac{1}{k}}$, the geometric average of the sequence $\big\{\|x^\ell - x^{\ell+1}\|_2\big\}_{l=0}^{k-1}$. Here step (i) above follows from arithmetic-geometric mean (AM/GM) inequality; step (ii) follows from the bound in equation (61) and from the fact that $\gamma = \frac{2r-1}{r}$. Finally, note that the last equality holds trivially for iteration $k \le k_1$ with the given choice of constant c_2.

E.2. Proof of Theorem 5

The proof of Theorem 5 builds on the techniques used in the proof of Theorem 4 but requires additional technical care due to the presence of possibly non-continuous function φ.

Convergence of the sequence $\{x^k\}_{k\ge 0}$: The proof of Theorem 5 has two steps. First, we prove a descent condition similar to equation (60). We then leverage this descent condition and weighted AM-GM inequality to obtain the desired result.

Step 1: Following the proof of Theorem 4, we prove the convergence of the sequence $\{x^k\}_{k\ge 0}$ by showing that the sequence $\{x^k\}_{k\ge 0}$ has finite length property. First, note that for scalars $0 \le \theta < 1$ and $0 < \gamma < \frac{1}{\theta}$, the function $t \mapsto t^{1-\gamma\theta}$ is concave. Consequently, for iteration $k \ge k_1$, from Lemma 7 we have

$$\begin{aligned}\big(f(x^k) - \bar{f}\big)^{1-\gamma\theta} - \big(f(x^{k+1}) - \bar{f}\big)^{1-\gamma\theta} &\ge (1-\gamma\theta)\big(f(x^k) - \bar{f}\big)^{-\gamma\theta}\big(f(x^k) - f(x^{k+1})\big)\\ &\overset{(i)}{\ge} (1-\gamma\theta)\big(|f(x^k) - \bar{f}|\big)^{-\gamma\theta} \times \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2\\ &\overset{(ii)}{\ge} \frac{(1-\gamma\theta)}{C\|\nabla f(x^k)\|_2^\gamma} \times \frac{1}{2\alpha}\|x^k - x^{k+1}\|_2^2. \end{aligned} \tag{62}$$

Here step (i) follows from the descent property in equation (52) and from the fact that $f(x^k) \downarrow \bar{f}$; step (ii) follows from Lemma 7. The function h is locally smooth by assumption; as a result, we have that the difference function $g - h$ is locally smooth. We also assumed that the sequence $\{x^k\}_{k\ge 0}$ is bounded (lies in a compact set S); consequently, we may assume that the difference function $g - h$ is smooth in the compact set S with a smoothness parameter M_{g-h}(say). Borrowing the argument of Theorem 2 part(b), it follows that:

$$\|\nabla g(x^k) - \nabla h(x^k) + v^k\|_2 \le \big(M_{g-h} + \frac{1}{\alpha}\big)\|x^k - x^{k-1}\|_2. \tag{63}$$

Combining the last inequality with inequality (62) yields the following descent property

$$\big(f(x^k) - \bar{f}\big)^{1-\gamma\theta} - \big(f(x^{k+1}) - \bar{f}\big)^{1-\gamma\theta} \ge \frac{(1-\gamma\theta)}{2\alpha C\big(M_{g-h} + \frac{1}{\alpha}\big)^\gamma} \times \frac{\|x^k - x^{k+1}\|_2^2}{\|x^k - x^{k-1}\|_2^\gamma}. \tag{64}$$

Step 2: We now leverage the descent condition obtained from step 1 to prove finite length property of the sequence $\{x^k\}_{k\ge 0}$. In order to facilitate further discussion, we use Δ_γ^k to denote the following:

$$\Delta_\gamma^k := C_3\Big(\big(f(x^k) - \bar{f}\big)^{1-\gamma\theta} - \big(f(x^{k+1}) - \bar{f}\big)^{1-\gamma\theta}\Big),$$

where the constant $C_3 := \frac{2\alpha C\big(M_{g-h} + \frac{1}{\alpha}\big)^\gamma}{(1-\gamma\theta)}$. With this notation, we can rewrite the equation (64) as

$$\Delta_\gamma^k \|x^{k-1} - x^k\|_2^\gamma \ge \|x^k - x^{k+1}\|_2^2. \tag{65}$$

Combining equation (65) with the weighted AM-GM inequality, we obtain

$$\begin{aligned}\big(1 + \frac{\gamma}{2-\gamma}\big) \times \sum_{j=k_1+1}^{k} \|x^j - x^{j+1}\|_2^{2-\gamma} &\overset{(i)}{\le} \big(1 + \frac{\gamma}{2-\gamma}\big) \times \sum_{k=k_1+1}^{k} \Big(\sqrt{\Delta_\gamma^j\, \|x^{j-1} - x^j\|_2^\gamma}\Big)^{\frac{2-\gamma}{2}}\\ &\overset{(ii)}{\le} \sum_{j=k_1+1}^{k} \big(\Delta_\gamma^j + \frac{\gamma}{2-\gamma}\|x^{j-1} - x^j\|_2^{2-\gamma}\big)\\ &\overset{(iii)}{\le} C_3\big(f(x^{k_1}) - \bar{f}\big)^{1-\gamma\theta} + \sum_{j=k_1+1}^{k} \frac{\gamma}{2-\gamma}\|x^{j-1} - x^j\|_2^{2-\gamma}. \end{aligned} \tag{66}$$

Here step (i) follows from equation (65), and step (ii) is implied by applying weighted AM-GM inequality as follows:

$$\frac{\Delta_\gamma^j + \frac{\gamma}{2-\gamma}\|x^{j-1} - x^j\|_2^{2-\gamma}}{1 + \frac{\gamma}{2-\gamma}} \ge \Big(\Delta_\gamma^j \|x^{j-1} - x^j\|_2^\gamma\Big)^{\frac{1}{1+\frac{\gamma}{2-\gamma}}}.$$

Step (iii) in equation (66) follows from the following observation

$$\begin{aligned}\sum_{j=k_1}^{k} \Delta_\gamma^j &= C_3 \sum_{j=k_1}^{k} \big(f(x^j) - \bar{f}\big)^{1-\gamma\theta} - \big(f(x^{j+1}) - \bar{f}\big)^{1-\gamma\theta}\\ &\le C_3\big(f(x^{k_1}) - \bar{f}\big)^{1-\gamma\theta}.\end{aligned}$$

Rewriting inequality (66), we have for all $k \ge k_1 + 2$

$$\begin{aligned}\sum_{j=k_1+1}^{k-1} \|x^j - x^{j+1}\|_2^{2-\gamma} &\le C_3\big(f(x^{k_1}) - \bar{f}\big)^{1-\gamma\theta} + \frac{\gamma}{2-\gamma}\|x^{k_1} - x^{k_1+1}\|_2^{2-\gamma} - \big(1 + \frac{\gamma}{2-\gamma}\big)\|x^k - x^{k+1}\|_2^{2-\gamma}\\ &\le C_3\big(f(x^{k_1}) - \bar{f}\big)^{1-\gamma\theta} + \frac{\gamma}{2-\gamma}\|x^{k_1} - x^{k_1+1}\|_2^{2-\gamma} < \infty. \end{aligned} \tag{67}$$

Finally, by substituting $\gamma = 1$ and letting $k \to \infty$ in the last equation, we deduce the finite length property of the sequence $\{x^k\}_{k\ge 0}$.

Rate of convergence of Avg $\big(\|\nabla f(x^k)\|_2\big)$ and GAvg $\big(\|\nabla f(x^k)\|_2\big)$: The proof of this part follows from the corresponding proof in Theorem 4 and using the inequality (67) and upper bound (63).

E.3. Proof of Lemma 7

Since the sequence $\{x^k\}_{k\geq 0}$ is bounded by assumption, without loss of generality, we may assume that the set of limit points of the sequence $\{x^k\}_{k\geq 0}$ — which we denote by $\bar{\mathcal{X}}$ — is a compact set. From Theorem 1 (respectively Theorem 2), we have that all the limit points of the sequence $\{x^k\}_{k\geq 0}$ are critical points of the function f; furthermore, since $f(x^k) \downarrow \bar{f}$, we also have that the function f is constant on the set of limit points $\bar{\mathcal{X}}$, and the function value on $\bar{\mathcal{X}}$ equals $\bar{f}$. Combining this with Assumption KL, we have for all $z \in \bar{\mathcal{X}}$, there exists constants $\theta(z) \in [0,1)$, $r_z > 0$ and $C(z) > 0$ such that, $\mid f(x) - \bar{f} \mid^{\theta(z)} \leq C(z) \times \|\nabla f(x)\|_2$ for all $x \in B(z, r_z)$. Now, consider the open cover $\{B(z, r_z) : z \in \bar{\mathcal{X}}\}$ of the set $\bar{\mathcal{X}}$. From compactness of the set $\bar{\mathcal{X}}$, we are guaranteed to have a finite subcover; more precisely, there exists $\{z_1, \ldots z_p\} \subseteq \bar{\mathcal{X}}$ such that $\bar{\mathcal{X}} \subseteq \bigcup_{i=1}^{p} B(z_i, r_{z_i})$. Define constants $\theta := \max\{\theta(z_i) : 1 \leq i \leq p\}$, $C := \max\{C(z_i) : 1 \leq i \leq p\}$, and $r := \min\left\{\frac{r_{z_i}}{2} : 1 \leq i \leq p\right\}$. Utilizing the result $\|x^k - x^{k+1}\|_2 \to 0$ from Theorem 1 (respectively Theorem 2), one can show that, there exists positive integer k_1 such that for all $k \geq k_1$ we have $\|x^k - x^{k+1}\|_2 < \frac{r}{2}$, and $x^k \in \bigcup_{i=1}^{p} B(z_i, r_{z_i})$. Putting together these pieces, we conclude that for all $k \geq k_1$

$$x^k \in \bigcup_{i=1}^{p} B(z_i, r_{z_i}), \quad \text{and} \quad \mid f(x^k) - \bar{f} \mid^{\theta} \leq C\|\nabla f\|_2,$$

which proves the first part of claimed lemma. Now suppose the sequence $\{x^k\}_{k\geq 0}$ converges to a point $\bar{x}$, then we have that the set of limit points $\bar{\mathcal{X}} = \{\bar{x}\}$, is a singleton set. The rest of the proof is immediate by repeating the argument so far, with the additional information that $\bar{\mathcal{X}} = \{\bar{x}\}$.

Appendix F. Proofs of Corollaries

In this appendix, we collect the proofs of Corollaries 4, 5 and 6 from Section 4.

F.1. Proof of Corollary 4

First, note that in order to apply Theorem 1 and Theorem 4 to Corollary 4, it is enough to show that the function $\mu \mapsto f(\mu)$ is M_f-smooth (in this example, function $h \equiv 0$, and hence $f \equiv g$), and the function f satisfies Assumption KL. We verify that Assumption KL is satisfied by proving that the objective function f in problem (22) is continuous sub-analytic (see Appendix A.2). For proving sub-analyticity, we heavily use the properties mentioned in Appendix A.3. In the following proof, we assume without loss of generality that $\lambda = 1$.

The function f is continuous sub-analytic: First, we show that the function Ψ is sub-analytic. We begin by observing that Ψ is piecewise polynomial. Polynomials are analytic functions and intervals are semi-analytic sets. Since piecewise analytic functions with semi-analytic pieces are semi-analytic (hence sub-analytic), we conclude that the function Ψ is sub-analytic. Now, the function $\mu \mapsto y_i - \langle z_i, \mu\rangle$ is linear, and hence continuous sub-analytic. Furthermore, since continuous sub-analytic functions are closed under composition, we have that the function $\mu \mapsto \Psi\big(y_i - \langle z_i, \mu\rangle\big)$ is sub-analytic. Finally, note that sub-analytic functions are closed under linear combination, and we conclude that the function f is sub-analytic. The continuity of the function f is immediate by inspection.

The function f is smooth: Since the vectors $\{(z_i, y_i)\}_{i=1}^{n}$ are fixed, it suffices to prove that the function Ψ is smooth. A straightforward calculation shows that Ψ is continuously differentiable and smooth; in particular, it has a smoothness parameter 36 when $\lambda = 1$.

Putting together the pieces, we conclude that Theorem 1 and Theorem 4 are applicable for problem (22). Convergence of the sequence $\{\mu^k\}_{k\geq 0}$ to a point $\bar{\mu}$ and the convergence rate of gradient norms follows from Theorem 4, and the stationary condition $\nabla f(\bar{\mu}) = 0$ follows from Theorem 1.

Escaping strict saddle points: Note that the functions (g, h) are twice continuously differentiable, and the function g is smooth. Consequently, from Corollary 3, it follows that with random initializations, Algorithm 1 avoids strict saddle points almost surely.

F.2. Proof of Corollary 5

We begin by providing a high-level outline of the proof. First, note that from Theorem 2, we have the successive difference $\|x^k - x^{k+1}\|_2 \to 0$, and as a result, the set of limit point of the sequence $\{x^k\}_{k\geq 0}$—call it $\bar{\mathcal{X}}$—is a connected set (Ostrowski, 2016). We prove that the connected-set $\bar{\mathcal{X}}$ is singleton by showing that the set $\bar{\mathcal{X}}$ has an isolated point — this also proves that sequence $\{x^k\}_{k\geq 0}$ is convergent. Next, we show that the objective-function f, in the problem (28), satisfies Assumption KL with exponent $\theta = \frac{1}{2}$. Finally, we show that condition $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)} \geq 0$ implies that function $x \mapsto h(x) := \sum_{i=d-s+1}^{d} |x|_{(i)}$ is smooth in a neighborhood of point $\bar{x}$, and we use the proof techniques of Theorem 5 to establish the convergence rate of the gradient sequence. In order to obtain the rate of convergence of the sequence $\{x^k\}_{k\geq 0}$, we use ideas similar to those in the paper (Lee et al., 2016).

Convergence of the sequence $\{x^k\}_{k\geq 0}$: For notational convenience, let us use $g(x) := \|y - Bx\|_2^2$, $\varphi(x) := \lambda\|x\|_1$, and $h(x) := \lambda \sum_{i=d-s+1}^{d} |x|_{(i)}$. Since the point $\bar{x}$ satisfies the condition $|\bar{x}|_{(r)} > |\bar{x}|_{(r+1)} \geq 0$ by assumption, there must exist a neighborhood $B(\bar{x}, r)$ such that the function h is differentiable in the neighborhood $B(\bar{x}, r)$, and all points $x \in B(\bar{x}, r)$ satisfy $\text{sign}(x_{(i)}) = \text{sign}(\bar{x}_{(i)})$ for $1 \leq i \leq r$. We show that, in a neighborhood of the point $\bar{x}$, it is the only critical point, thereby proving that the point $\bar{x}$ is an isolated critical point. To this end consider the convex sub-problem mentioned in Corollary 5

$$\mathcal{P}(\bar{x}) := \min_{x \in \mathbb{R}^d} g(x) + \lambda\varphi(x) - \lambda\langle \nabla h(\bar{x}), x\rangle. \tag{68}$$

For any point x^* such that $x^* \in B(\bar{x}, r) \cap \bar{\mathcal{X}}$, from Theorem 2, we know that

$$\nabla g(\bar{x}) + \lambda\bar{u} - \lambda\nabla h(\bar{x}) = 0 \quad \text{and} \quad \nabla g(x^*) + \lambda u^* - \lambda\nabla h(x^*) = 0, \tag{69}$$

where subgradients $u^* \in \partial\varphi(x^*)$ and $\bar{u} \in \partial\varphi(\bar{x})$. Next, note that from the choice of neighborhood $B(\bar{x}, r)$, it follows that for all $x \in B(\bar{x}, r)$ we have $\nabla h(x) = \nabla h(\bar{x})$, and in particular, we deduce $\nabla h(x^*) = \nabla h(\bar{x})$. Combining this relation with equation (69) yields:

$$\nabla g(\bar{x}) + \lambda\bar{u} - \lambda\nabla h(\bar{x}) = 0 \quad \text{and} \quad \nabla g(x^*) + \lambda u^* - \lambda\nabla h(\bar{x}) = 0,$$

which implies both the points x^* and $\bar{x}$ are zero sub-gradient points of convex problem (68); this contradicts the assumption that problem (68) has an unique solution. Hence, we

conclude that $x^* = \bar{x}$, and the point $\bar{x}$ is an isolated critical point of the sequence $\{x^k\}_{k\geq 0}$, and $\bar{\mathcal{X}}$. Putting together the pieces, we conclude that $x^k \to \bar{x}$.

Smoothness of function h in a neighborhood of $\bar{x}$: We already argued above that for all $x \in B(\bar{x}, r)$, the function h is differentiable and $\nabla h(x) = \nabla h(\bar{x})$. Consequently, we have that in the neighborhood $B(\bar{x}, r)$, the function h is smooth with a smoothness parameter $M_h = 0$.

The function f satisfies Assumption KL with exponent $\theta = \frac{1}{2}$: Recently, in the paper (Li and Pong, 2016) (Corollaries 5.1 and 5.2), the authors showed that if the functions $f_1, f_2, \ldots, f_T$ satisfy the KL-inequality with an exponent $\theta = \frac{1}{2}$, then the function $f := \min\{f_1, f_2, \ldots, f_T\}$ also satisfies KL-inequality with the exponent $\theta = \frac{1}{2}$. Interestingly, the function f can be represented as is minimum of finitely many functions as follows:

$$f(x) = \min_{a \in \mathcal{A}} \{\|y - Bx\|_2^2 + \lambda\|x\|_1 - \lambda a^\top x\}, \tag{70}$$

where $\mathcal{A} := \{a \in \{-1, 0, 1\}^d : \sum_{i=1}^d |a_i| = r\}$. Note that the set $\mathcal{A}$ has cardinality at most 3^d. It is known that functions of the form $x \mapsto \frac{1}{2}x^\top Ax + P(x) + b^\top x$ satisfy the KL-inequality with exponent $\theta = \frac{1}{2}$, where P is a proper closed polyhedral function, and A is a positive semi-definite matrix; see Corollaries 5.1 and 5.2 in the paper (Li and Pong, 2016). Putting together these two observations, we conclude that the function f satisfies KL-assumption with KL-exponent $\theta = \frac{1}{2}$.

Combining the pieces: Since we proved $x^k \to \bar{x}$, we have that for a suitable choice of k_1, the tail sequence $\{x^k\}_{k\geq k_1}$ lies in the neighborhood $B(\bar{x}, r)$. Now, the function f satisfies Assumption KL with exponent $\theta = \frac{1}{2}$, and the function h is smooth in the neighborhood $B(\bar{x}, r)$; hence, following the argument in proof of Theorem 5 part(b), we conclude that:

$$\text{Avg}\left(\|\nabla f(x^k)\|_2\right) \leq \frac{c_1}{k}.$$

Rate of convergence of sequence $\{x^k\}_{k\geq 0}$: The KL-exponent for the function f is $\theta = \frac{1}{2}$, and we may use $\gamma = 1$ in equation (67) which yields

$$\sum_{\ell=k_1+1}^{\infty} \|x^\ell - x^{\ell+1}\|_2 \leq \|x^{k_1} - x^{k_1+1}\|_2 + C_3\big(f(x^{k_1}) - \bar{f}\big)^{\frac{1}{2}}, \tag{71}$$

for some constant C_3. From Lemma 7 and equation (48), we have

$$\big(f(x^{k_1}) - \bar{f}\big)^{\frac{1}{2}} \leq C\|\nabla f(x^{k_1})\|_2 \leq C(M + M_h + \frac{1}{\alpha})\|x^{k_1} - x^{k_1-1}\|_2. \tag{72}$$

Combining equations (71) and (72) we have

$$\begin{aligned}\sum_{\ell=k_1}^{\infty} \|x^\ell - x^{\ell+1}\|_2 &\leq 2\|x^{k_1} - x^{k_1+1}\|_2 + C_3\big(f(x^{k_1}) - \bar{f}\big)^{\frac{1}{2}} \\ &\overset{(i)}{\leq} 2\|x^{k_1} - x^{k_1+1}\|_2 + CC_3(M + M_h + \frac{1}{\alpha})\|x^{k_1} - x^{k_1-1}\|_2 \\ &\overset{(ii)}{\leq} \bar{C}\|x^{k_1} - x^{k_1-1}\|_2,\end{aligned} \tag{73}$$

where $\bar{C}$ is a constant depending on M, M_h, α, C_3 and C, and step (i) above follows from equation (72). We justify step (ii) shortly, but let us first derive the linear rate of convergence of the sequence $\{x^k\}_{k\geq 0}$ using the derivation in equation (73). Denote $e_k = \sum_{\ell=k}^{\infty} \|x^\ell - x^{\ell+1}\|_2$. Then equation (73) provides the following recursion

$$e_{k_1} \leq \bar{C}(e_{k_1-1} - e_{k_1}).$$

Simple inspection of proof of Theorem 5 and derivations so far ensure that we can derive the equations (71) and (72) for all $k \geq k_1$; this provides us a recursion relation as above with k_1 replaced by k. Furthermore, by choosing a larger value of the constant $\bar{C}$ if necessary, we may conclude that for all $k \geq 1$ we have

$$e_k \leq \bar{C}(e_{k-1} - e_k).$$

Rearranging the above inequality yields $e_k \leq \frac{\bar{C}}{\bar{C}+1} e_{k-1}$, which guarantees that the sequence $\{e_k\}_{k\geq 0}$ converges to zero at a linear rate. Finally, observe that $\|x^k - x^*\|_2 \leq \sum_{\ell=k}^{\infty} \|x^\ell - x^{\ell+1}\|_2 = e_k$, and the linear rate of convergence of the sequence $\{\|x^k - x^*\|_2\}_{k\geq 0}$ to zero follows.

Justification for step (ii) in equation (73): Note that it suffices to show that the object $\|x^{k_1} - x^{k_1+1}\|_2$ is upper bounded by a constant multiple of $\|x^{k_1} - x^{k_1-1}\|_2$, where the constant depends only on M, M_h, α and C. Recalling the decent property proved in equation (52) we have:

$$\big(f(x^{k_1}) - \bar{f}\big)^{\frac{1}{2}} \geq \big(f(x^{k_1}) - f(x^{k_1+1})\big)^{\frac{1}{2}} \geq \frac{1}{\sqrt{2\alpha}}\|x^{k_1} - x^{k_1+1}\|_2. \tag{74}$$

Combining equations (74) and (72) we obtain the following upper and lower bound of $\big(f(x^{k_1}) - \bar{f}\big)^{\frac{1}{2}}$:

$$\frac{1}{\sqrt{2\alpha}}\|x^{k_1} - x^{k_1+1}\|_2 \leq \big(f(x^{k_1}) - \bar{f}\big)^{\frac{1}{2}} \leq C(M + M_h + 1/\alpha)\|x^{k_1} - x^{k_1-1}\|_2.$$

Rearranging the last equality proves the desired upper bound. Finally, we reiterate that the above justification also hold for any iterate k with $k \geq k_1$.

F.3. Proof of Corollary 6

The proof of this corollary is based on application of Theorems 2 and 5. We verify the assumptions of Theorems 2 and 5 with $g(\theta) = -\sum_{i=1}^{n} \log\big(\zeta(y_i; \theta)\big)$, $h \equiv 0$, $\varphi = \mathbb{1}_{\mathcal{X}}$ and function $f := g - \varphi + h$. Note that the domain $\text{dom}(f) = \mathcal{X}$ is compact, which guarantees that the iterate sequence $\{\theta^k\}_{k\geq 0}$ obtained from Algorithm 2 is bounded. The function $h \equiv 0$ is smooth. The log-partition function A is twice continuously differentiable by assumption, which guarantees that the function g is also twice continuously differentiable, whence smooth in the compact domain $\mathcal{X}$. Finally, we verify that the function f satisfies Assumption KL by proving that f is continuous sub-analytic in its domain $\mathcal{X}$, and the domain $\mathcal{X}$ is closed; see Lemma 3. Clearly, $\text{dom}(f) = \mathcal{X}$ is closed, and the function f is continuous in $\text{dom}(f)$. Finally, we show that the functions (g, φ) are sub-analytic, and invoking the property (d) of sub-analytic functions form Appendix A.3, we conclude that the function $f := g + \varphi$ is sub-analytic.

The function φ is sub-analytic: Here, we use a simple result by (Attouch et al., 2010), which states that the indicator function of a semi-algebraic set is a semi-algebraic function (hence a sub-analytic function). In order to show that the set $\mathcal{X}$ is semi-algebraic, we note the following representation of the set $\mathcal{X}$

$$\mathcal{X} = \{\sum_{i=1}^{d} \theta_i^2 > R_1^2\}^c \bigcap \{\sum_{i=d+1}^{2d} \theta_i^2 > R_2^2\}^c \bigcap \{\theta_{2d+1} > 1\}^c \bigcap \{-\theta_{2d+1} > 0\}^c. \tag{75}$$

Each of the four sets in representation (75) are semi-algebraic by definition, and semi-algebraic sets are closed under finite intersection and complements; see the book by (Coste, 2002). Putting together these two observations, we conclude that the set $\mathcal{X}$ is semi-algebraic, and that $\mathbb{1}_{\mathcal{X}}$ is a sub-analytic function.

The function g is sub-analytic: The log-partition function A is sub-analytic by assumption. For a fixed vector y, the map $\eta \mapsto \eta^\top T(y)$ is linear, and hence sub-analytic. Since sub-analytic functions are closed under a finite linear combination, we conclude that the map $\eta \mapsto \eta^\top T(y) - A(\eta)$ is sub-analytic. Continuous sub-analytic functions are closed under multiplication and composition; since the $\exp(\cdot)$ function is continuous sub-analytic, we have for every fixed vector y the following map

$$(\eta_0, \eta_1, p) \mapsto \zeta(y; \eta_0, \eta_1, p) := p \exp(\eta_0^\top T(y) - A(\eta_0)) + (1-p)\exp(\eta_1^\top T(y) - A(\eta_1))$$

is sub-analytic. Furthermore, the $\log(\cdot)$ function analytic on the interval $(0, \infty)$, and using the composition rule for continuous sub-analytic functions, we obtain that the map $\theta \mapsto \log(\zeta(y_i; \theta))$ is sub-analytic, where $\theta := (\eta_0, \eta_1, p)$. Finally, the target function g is a linear combination of sub-analytic functions $\log(\zeta(y_i; \theta))$, and we conclude that the map $\theta \mapsto g(\theta)$ is sub-analytic.

Combining the pieces: Putting together the pieces, we conclude that the function f is sub-analytic, with the function f being continuous in $\mathrm{dom}(f)$, whereas $\mathrm{dom}(f)$ is closed; furthermore, the functions g and h are smooth. This allows us to apply Theorem 2 and Theorem 5 and the corollary follows.

Sub-analyticity of the log-partition functions A in Table 1: The sub-analyticity of the log-partition function A mentioned in Table 1 follows from the following two observations. First, note that the functions $\exp, \ln$ and Γ are continuous and analytic (hence sub-analytic). Given two continuous sub-analytic functions g_1 and g_2, the composition function $g_2 \circ g_1$ is also continuous sub-analytic. Secondly, any linear combination of sub-analytic functions is also sub-analytic function. See Appendix A.3 for properties of sub-analytic functions.

Appendix G. Characterizing "smooth - convex" function class

In Theorem 1 and Theorem 2 we discussed a class of non-smooth non-convex functions, where a gradient or a prox-type algorithm provides satisfactory convergence to a critical point. One possible deficiency of the theory discussed so far is that, in Algorithm 1 (respectively Algorithm 2), we need to specify a decomposition of the objective function f as

a difference of a smooth and a convex function (respectively, smooth + convex - convex). Consequently, it is natural to wonder if we can characterize the class of functions which has a decomposition needed in Algorithms 1 and 2. Furthermore, if a function has this a decomposition, how can we obtain such a decomposition easily. It is worth pointing out that for the case of Algorithm 2, the convex function φ is known in many cases. For instance, in the case of constrained optimization, the function φ is the indicator of the constraint set; in many statistical estimation problems, φ is a penalty function on the parameters; a well-known example of such penalty function is the ℓ_1 penalty, which is used to obtain sparse solutions. Hence, for all practical purposes, the task of characterizing the function class mentioned in Theorems 1 and 2 reduces to characterizing functions which can be decomposed as a difference of a smooth function(g) and a convex function (h). In the next theorem, we characterize the class of of continuously differentiable functions that can be written as a difference of a smooth function and a convex function.

Theorem 6 *Given any continuously differentiable function $f : \mathbb{R}^d \mapsto \mathbb{R}$, the following two properties are equivalent:*

(a) There exists a M-smooth function g, and a convex continuously differentiable function h such that:

$$f(x) = g(x) - h(x) \quad \text{for all } x \in \mathbb{R}^d.$$

(b) The gradient of the function f satisfies the following inequality:

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq M\|x-y\|^2 \quad \text{for all } x, y \in \mathbb{R}^d.$$

Proof We establish the equivalence by proving the circle of implications $(a) \Longrightarrow (b) \Longrightarrow (a)$.

Implication (a) $\Longrightarrow$ (b): For any M-smooth function g, we have the following:

$$\begin{aligned} \langle \nabla g(x) - g(y), x - y \rangle &\leq \|\nabla g(x) - g(y)\|_2 \times \|x-y\|_2 \\ &\overset{(i)}{\leq} M\|x-y\|_2^2, \quad \text{for all } x, y \in \mathbb{R}^d, \end{aligned} \tag{76}$$

where step (i) follows since the gradient ∇g is M Lipschitz. Next note that the gradient of a differentiable convex function is a monotone operator, and we have that for all $x, y \in \mathbb{R}^d$:

$$\langle \nabla h(x) - \nabla h(y), x - y \rangle \geq 0. \tag{77}$$

Subtracting equation (77) from equation (76), we obtain the desired upper bound in part (b).

Implication (b) $\Longrightarrow$ (a): We prove this implication by finding a M–smooth function g and a convex differentiable function h such that $f = g - h$. To this end, we fix any $x_0 \in \mathbb{R}^d$ and consider the following two functions:

$$g(x) := f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle + \frac{M}{2}\|x - x_0\|_2^2 \tag{78a}$$

$$h(x) := g(x) - f(x). \tag{78b}$$

The function g in definition (78a) is M-smooth by inspection. Since both the functions f and g are continuously differentiable, the function h is continuously differentiable by construction. In order to complete the proof, it suffices to show that the function h is convex. To this end, the first order Taylor series expansion of the function h yields

$$\begin{aligned} h(x) &= h(y) + \big\langle \nabla h(y + t(x-y)), x - y \big\rangle \quad \text{for some } t \in [0,1] \\ &= h(y) + \big\langle \nabla h(y), x - y \big\rangle + \big\langle \nabla h(y + t(x-y)) - \nabla h(y), x - y \big\rangle. \end{aligned} \tag{79}$$

Expanding the term $\big\langle \nabla h(y + t(x-y)) - \nabla h(y), x - y \big\rangle$ above yields,

$$\begin{aligned} \big\langle \nabla h(y + t(x-y)) - \nabla h(y), x - y \big\rangle &\overset{(i)}{=} M\|x-y\|_2^2 - \frac{\big\langle \nabla f(y + t(x-y)) - \nabla f(y), t(x-y) \big\rangle}{t} \\ &\overset{(ii)}{\geq} M\|x-y\|_2^2 - Mt\|x-y\|_2^2 \\ &\overset{(iii)}{\geq} 0. \end{aligned}$$

Here step (i) follows by substituting the expression of the function h; step (ii) follows from the gradient inequality of part (b), and step (iii) follows from the inequality $0 \leq t \leq 1$. Since the vectors $x, y \in \mathbb{R}^d$ were arbitrary, the inequality $\big\langle \nabla h(y + t(x-y)) - \nabla h(y), x - y \big\rangle \geq 0$ combined with equation (79) proves the convexity of the function h, thereby proving the claimed result in part (a).

■

Comments: It would be interesting to characterize the class of DC-based functions mentioned in problem (2) when the convex function h is non-differentiable. Indeed, we obtain a larger and more interesting non-differentiable class of functions. It would interesting to see whether Theorem 6 can be suitably generalized in this setting.

References

Stanislaw Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. *Les équations aux dérivées partielles*, 117:87–89, 1963.

Amir A Ahmadi and Pablo A Parrilo. A complete characterization of the gap between convexity and sos-convexity. *SIAM Journal on Optimization*, 23(2):811–833, 2013.

Amir A Ahmadi, Alex Olshevsky, Pablo A Parrilo, and John N Tsitsiklis. Np-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 137(1-2):453–476, 2013.

Nguyen Thai An and Nguyen Mau Nam. Convergence analysis of a proximal point algorithm for minimizing differences of functions. *Optimization*, 66(1):129–147, 2017.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka- Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2): 459–494, 2014.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

Coralia Cartis, Nicholas IM Gould, and Ph L Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *Siam journal on optimization*, 20(6):2833–2852, 2010.

Michel Coste. An introduction to semialgebraic geometry. *RAAG network school*, 145:30, 2002.

Ady Ecker and Allan D Jepson. Polynomial shape from shading. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 145–152. IEEE, 2010.

Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

Jianqing Fan and Runz Li. Variable selection via non-concave penalized likelihood and its oracle properties. *Jour. Amer. Stat. Ass.*, 96(456):1348–1360, December 2001.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. *arXiv preprint arXiv:1704.00708*, 2017.

Jun-Ya Gotoh, Akiko Takeda, and Katsuya Tono. DC formulations and algorithms for sparse optimization problems. *Mathematical Programming*, pages 1–36, 2017.

Philip Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):707–713, 1959.

Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

Reiner Horst, Panos M. Pardalos, and Nguyen Van Thoai. *Introduction to Global Optimization*, volume 9 of *Nonconvex Optimization and its Applications*. Elsevier, 2000.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pages 769–784. Chartres: L'Institut, 1950-, 1998.

Simon Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

Gert R Lanckriet and Bharath K Sriperumbudur. On the convergence of the concave-convex procedure. In *Advances in neural information processing systems*, pages 1759–1767, 2009.

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

Guoyin Li and Ting Kei Pong. Calculus of the exponent of Kurdya- Łojasiewicz inequality and its applications to linear convergence of first-order methods. *arXiv preprint arXiv:1602.02915*, 2016.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

Thomas Lipp and Stephen Boyd. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2):263–287, 2016.

Po-Ling Loh and Martin J Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pages 476–484, 2013.

Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Alexander M Ostrowski. *Solution of equations and systems of equations: Pure and applied mathematics: A Series of monographs and textbooks*, volume 9. Elsevier, 2016.

Ioannis Panageas and Georgios Piliouras. Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions. *arXiv preprint arXiv:1605.00405*, 2016.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

T Pham Dinh, HV Ngai, and HA Le Thi. Convergence analysis of the DC algorithm for DC programming with subanalytic data. *preprint*, 2013.

R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Hoang Tuy. DC optimization: theory, methods and algorithms. In *Handbook of global optimization*, pages 149–216. Springer, 1995.

M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, Cambridge, UK, 2019.

Shenlong Wang, Alex Schwing, and Raquel Urtasun. Efficient inference of continuous Markov random fields with polynomial potentials. In *Advances in neural information processing systems*, pages 936–944, 2014.

Bo Wen, Xiaojun Chen, and Ting Kei Pong. A proximal difference-of-convex algorithm with extrapolation. *Computational Optimization and Applications*, 69(2):297–324, 2018.

Yangyang Xu and Wotao Yin. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing*, 72(2):700–734, 2017.

Bowei Yan, Mingzhang Yin, and Purnamrita Sarkar. Convergence of gradient EM on multi-component mixture of Gaussians. In *Advances in Neural Information Processing Systems*, pages 6959–6969, 2017.

Alan L Yuille and Anand Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999.

Journal of Machine Learning Research 20 (2019) 1-35 Submitted 1/19; Revised 6/19; Published 10/19

Quantifying Uncertainty in Online Regression Forests

Theodore Vasiloudis TVAS@SICS.SE
RISE SICS
Stockholm, Sweden

Gianmarco De Francisci Morales GDFM@ACM.ORG
ISI Foundation
Torino, Italy

Henrik Boström BOSTROMH@KTH.SE
KTH Royal Institute of Technology
Stockholm, Sweden

Editor: Alexander Rakhlin

Abstract

Accurately quantifying uncertainty in predictions is essential for the deployment of machine learning algorithms in critical applications where mistakes are costly. Most approaches to quantifying prediction uncertainty have focused on settings where the data is static, or bounded. In this paper, we investigate methods that quantify the prediction uncertainty in a streaming setting, where the data is potentially unbounded.

We propose two meta-algorithms that produce prediction intervals for online regression forests of arbitrary tree models; one based on conformal prediction, and the other based on quantile regression. We show that the approaches are able to maintain specified error rates, with constant computational cost per example and bounded memory usage. We provide empirical evidence that the methods outperform the state-of-the-art in terms of maintaining error guarantees, while being an order of magnitude faster. We also investigate how the algorithms are able to recover from concept drift.

Keywords: Online learning, Uncertainty, Decision Trees, Regression

1. Introduction

Machine learning algorithms are increasingly being deployed in critical settings, such as the financial and medical domains. In such domains, the ability to quantify the uncertainty in the algorithms' predictions is crucial to guide decision making. In addition, in time-critical applications, such as autonomous driving, algorithms need to produce predictions and uncertainty estimates under tight computational and time budgets. Online learning methods are often used in such settings, alas, without providing uncertainty estimates, which makes them unsuitable for scenarios where errors are costly.

There exists a variety of applications where such critical decisions need to be made in real time, while using limited computational resources. In the financial sector, time-critical decisions need to be made based on noisy signals, and the models must be kept continuously up to date with the latest market data (Gama, 2017). In autonomous vehicles (AV), an embedded system needs to continuously translate noisy input from multiple sensors into driving commands, with the safety of passengers and pedestrians at stake. Quantification of the uncertainty in the predictions of the models is paramount to avoid potentially catastrophic mishappenings (McAllister et al., 2017).

For example, take the problem of estimating the distance from the vehicle ahead for an AV. A simple point estimate could be dangerous in the presence of uncertainty: assume it takes the AV 100 meters to stop, and the distance point prediction is 110m but the true distance is 90m. In an emergency situation, the AV would not be able to stop before crashing into the vehicle ahead. If instead the model produces a 99.9%-confidence interval of the distance, we could use its lower end to ensure with high probability that there is always enough braking distance from the vehicle ahead.

In this work, we focus on ensembles of decision trees as predictors, which have been shown to be highly competitive in a variety of settings (Fernández-Delgado et al., 2014), including their online variants (Oza, 2005). Most previous work on decision trees has focused on one of the aspects of the problem: either providing online algorithms, or providing uncertainty estimates for batch algorithms. A number of approaches for quantifying the uncertainty of predictions in random forests (RF) have been proposed in the past, but they all assume a bounded, static dataset (Mentch and Hooker, 2016; Wager et al., 2014; Chipman et al., 2010; Meinshausen, 2006; Johansson et al., 2014b; Boström et al., 2017). On the other hand, most approximate tree algorithms that fit the computational and time limitations of the domain (Domingos and Hulten, 2000; Ben-Haim and Tom-Tov, 2010; Ikonomovska et al., 2011) do not provide uncertainty estimates. One notable exception are Mondrian Forests (Lakshminarayanan et al., 2016), whose runtime cost, however, is still prohibitive for limited-memory, real-time systems, as both its computational and memory costs increase with each incoming example.

In this study, we investigate approaches to quantify model uncertainty for regression tasks, when using ensembles of online decision tree learners with bounded computational and memory cost. We propose two online algorithms, one based on conformal prediction (Vovk et al., 2005), and another based on quantile regression (Meinshausen, 2006), and perform a large-scale empirical evaluation.

The proposed algorithms are meta-algorithms, in the sense that they allow to use different underlying learning algorithms. However, optimizing the selection of the underlying learner and tuning the approximate data structures used are out of the scope of the current study. We instead focus on the performance of the meta-algorithms themselves, while keeping all other parts constant.

In summary, our contributions are the following:

- We describe how to adapt inductive conformal prediction and quantile regression forests to the online setting by bounding their computational and memory use (Section 3).
- We report on an extensive experimental evaluation with thirty datasets of various sizes, including datasets that exhibit concept drift, and compare the proposed approaches against both a simple baseline and a state-of-the-art algorithm (Section 4).
- We empirically show that the proposed algorithms are able to maintain prediction error bounds at a rate better than the state-of-the-art, while being bounded in memory and computation, regardless of the number of data points in the input.[1]

1. All algorithm implementations, data, and experiment automation scripts used in this study are available as open source to ensure reproducibility at `https://github.com/thvasilo/uncertain-trees-reproducible`

©2019 Theodore Vasiloudis, Gianmarco De Francisci Morales, Henrik Boström.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/19-006.html`.

2. Background

In this section, we formally define the problem of online interval prediction for regression, and provide descriptions of the algorithms that we later adapt to the online scenario.

2.1. Problem definition

We assume a sequentially arriving data stream $(\mathbf{X}_i, y_i)$, sampled from some fixed underlying distribution, where $\mathbf{X}_i \in \mathbb{R}^p$ are the feature vectors and $y_i \in \mathbb{R}$ the labels. Additionally $i \in \mathbb{N}^+$ indicates the index of a potentially unbounded dataset. Our goal is to learn a function $\Gamma(\mathbf{X}_i, \alpha) : \mathbb{R}^p \times (0,1) \to [l, u]$, where $l, u \in \mathbb{R}$ and $l \leq u$, such that the probability of drawing an example $(\mathbf{X}_i, y_i)$ from the fixed underlying distribution where $y_i \notin \Gamma(\mathbf{X}_i, \alpha)$ is *less than or equal* to α, which is referred to as the *significance* level. We refer to $1 - \alpha$ as the *confidence* level.

In the conformal prediction literature, the property described above is termed *conservative* validity, as opposed to *exact* validity, for which the above probability is exactly α. This distinction makes our problem different from quantile regression (Koenker, 1996) which aims for exact validity as well.

Henceforth, for the description of the algorithms, we assume an ensemble that consists of a random forest ℓ of T decision trees, each denoted by $\ell_t, t \in [1, T]$. We consider an interval predictor to be *valid* if it makes errors, when $y_i \notin \Gamma(\mathbf{X}_i, \alpha)$, at a rate at most α.

2.2. Inductive Conformal Prediction

Conformal predictors (CP) output *prediction regions*, that is, sets of labels for classification and intervals for regression, instead of traditional point predictions. The main property of a conformal predictor is that it is *valid*: the expected error rate is upper-bounded by a predetermined significance level. By leveraging past experiences, the conformal predictor can guarantee that, as long as the examples are drawn from the same distribution, the probability of excluding the true label from the prediction region is less than or equal to the specified significance. In addition, it is a meta-algorithm, which can use any base point predictor for classification or regression. For a comprehensive review of conformal prediction see (Vovk et al., 2005).

The original framework for conformal prediction was designed for an online scenario, in which observations are received one-by-one. Each observation triggers a prediction, after which the true label is revealed, which in turn allows the predictor to be updated. However, this framework, which is called *transductive conformal prediction*, requires retraining the underlying model with the entire dataset for each new observation to obtain the so-called *non-conformity scores*. These scores measure how "out of the ordinary" examples are compared to the already observed data, and are used to produce the prediction intervals. The retraining requirement makes the method very costly computationally, and prevents it from being used straightforwardly for large datasets.

To overcome the computational cost of the transductive framework, another instantiation of the conformal prediction framework was proposed by Papadopoulos et al. (2002), named *inductive conformal prediction* (ICP). This approach is aimed at the batch setting, and does not require continuous re-training of the underlying model. Instead, it sets aside a

subset of the training examples, referred to as the *calibration set*, $\mathbf{C}$. The rest of the training data are used to train a batch predictor. The calibration set is used to produce sorted *non-conformity* scores, $\mathbf{S}$, which are again used to produce the prediction intervals. In regression, we commonly measure the non-conformity of an example by the absolute error of the model's prediction for that example. To determine the interval, we scan the (ascending) sorted list of non-conformity scores, $\mathbf{S}$, until we pass $\lfloor (1-\alpha) \cdot |\mathbf{S}| \rfloor$ values. The non-conformity score at this point gives us the prediction interval. Specifically, let $\phi = \mathbf{S}[\lfloor (1-\alpha) \cdot |\mathbf{S}| \rfloor]$ be the value selected from $\mathbf{S}$. Then the produced interval will be $\Gamma(\mathbf{X}_i, \alpha) = [\hat{y} - \phi, \hat{y} + \phi]$, where $\hat{y}$ is the point prediction of the model for $\mathbf{X}_i$.

Vovk (2002) proposed a way to adapt ICP for sequentially arriving datasets. The proposed method re-trains a new model from scratch after a fixed number of data points have been observed. This choice creates a trade-off between the computational efficiency of the algorithm and the predictive effectiveness in terms of the interval size. However, due to the need to store the complete training set, and completely retrain a new model at pre-set intervals, this method is still not a viable option for streaming datasets.

2.3. Quantile Regression Forests

Quantile Regression Forests (QRF) were introduced by Meinshausen (2006) with the purpose of extending random forests from computing the conditional mean to computing the full conditional cumulative distribution function (CDF). Following the notation by Meinshausen, we define the conditional CDF as

$$F(y \mid X = x) = P(Y \leq y \mid X = x). \tag{1}$$

Given Equation 1 we can define the β-quantile $Q_\beta(x)$ as the value for which the probability of Y being smaller than $Q_\beta(x)$ for a given x, is exactly β:

$$Q_\beta(x) = \inf\{y : F(y \mid X = x) \geq \beta\}.$$

Having access to the conditional CDF, we are able to produce prediction intervals by calculating the quantiles at the endpoints of the desired interval. For example, a prediction interval for significance level $\alpha = 0.1$ can be obtained by

$$\Gamma(x, \alpha) = [Q_{0.05}(x), Q_{0.95}(x)].$$

QRF, unlike other quantile regression methods, does not change the loss function of the underlying random forest algorithm. It instead calculates the conditional distribution as the *weighted distribution* of the observed labels across the forest. Therefore, only minimal changes to the underlying learning algorithm are needed in order to provide prediction intervals. Meinshausen shows that, under some mild assumptions, QRF is a consistent estimator of the conditional distribution.

Let us describe batch QRF briefly. It grows trees in the random forest as a regular RF algorithm. However, at every leaf, it stores all the label values, rather than only their average. Each new observation x is routed down all trees of the forest until it reaches a leaf. Denote $\texttt{leaf}(x, \ell_t)$ the leaf x reaches for tree $\ell_t \in \ell$. The weight of the example at each tree is given by $w_i(x, \ell_t) = \mathbb{1}_{\mathbf{X}_i \in \texttt{leaf}(x, \ell_t)} / |j : \mathbf{X}_j \in \texttt{leaf}(x, \ell_t)|$, i.e., the reciprocal of the number

of observations already in that leaf for tree ℓ_t. To get the observation's weight over the whole forest, we average over the individual trees: $w_i(x) = T^{-1}\sum_{t=1}^{T} w_i(x, \ell_t)$. The estimate for the conditional distribution is then given by the weighted sum over all the observations in the corresponding leaves for which the label is less than or equal to the requested y:

$$\hat{F}(y \mid X = x) = \sum_{i}^{N} w_i(x)\mathbb{1}_{\{Y_i \leq y\}} \tag{2}$$

where N is the number of data points in the forest.

QRF is a batch method, which requires access to the complete dataset beforehand, as we need to ensure the weights of all examples sum to one. In addition, in order to provide access to the full conditional distribution, the labels from all N examples must be stored in the leaves of the trees, in a mapping from the example weight to the corresponding label. This requirement imposes a prohibitive memory cost when datasets are massive or unbounded, which makes the algorithm unsuitable for the online setting.

3. Methods

This section describes the algorithms developed for this work, one based on inductive conformal prediction, and one on quantile regression forests.

3.1. Conformal Prediction with Online Regression Forests

As mentioned in Section 2, inductive conformal prediction was proposed by Papadopoulos et al. (2002) as an offline method, while the modification proposed by Vovk (2002) requires constant retraining of a model at preset intervals, and maintaining the complete training set. Both methods are thus unsuitable for settings where storing the complete dataset is impossible, e.g., when streaming massive amounts of data or when the dataset is unbounded.

Our proposed algorithm overcomes these issues, and removes the need to store all data points. Instead of retraining a new model with an increasing subset of the observed data, we use a single-pass, online model which continuously incorporates information as new examples arrive. In addition, by using an online random forest, we are able to maintain an up-to-date calibration set without setting aside examples exclusively for it. We keep the size of the calibration set bounded, thus making the memory requirements of the algorithm constant. Algorithm 1 shows the training function, and Algorithm 2 the corresponding prediction one.

3.1.1. Algorithm Description

We start by describing the training algorithm shown in Algorithm 1. It uses a combination of the online bagging algorithm by Oza (2005) and the out-of-bag conformal regression algorithm by Johansson et al. (2014a). Oza made the observation that for a large number of samples, the binomial distribution used to select samples in bagging tends to a Poisson(1) distribution. Therefore, this distribution can be used to approximate the bagging process online when we do not have access to the number of samples beforehand. Given an example to train on, for each member of the ensemble we make a draw k from a Poisson distribution with rate parameter $\lambda = 1$. If the drawn sample k is larger than zero, we use the example to train the member of the ensemble (with weight k). Oza showed that this process converges

Algorithm 1: OnlineCP Training (ℓ, $(\mathbf{X}_i, y_i)$), λ

input : ℓ: the current decision tree ensemble; $\mathbf{C}$: the set of calibration examples; $(\mathbf{X}_i, y_i)$: a labeled training example, c: max calibration set size.
output : $\mathbf{C}$: Updated set of calibration examples; ℓ_{oob}: a mapping from calibration examples to learners.

```
Set isOutOfBag to False
foreach ℓ_t ∈ ℓ do                                   // in order t ∈ [1, T]
    k = Poisson(1)
    if k > 0 then                                    // example in-bag for ℓ_t
        Train ℓ_t with (X, y), weighted by k
    else                                             // example out-of-bag for ℓ_t
        Set isOutOfBag to True
        Add ℓ_t to ℓ_oob for X_i
if isOutOfBag then                                   // ∃ ℓ_t : (X_i, y_i) was out-of-bag
    if |C| < c then
        Add (X_i, y_i) to bounded set of calibration examples, C
    else
        Remove oldest example from set C
        Add (X_i, y_i) to set C
```

Algorithm 2: OnlineCP Interval Prediction (ℓ, $\mathbf{X}_i$, $\mathbf{C}$, α)

input : ℓ: the current decision tree ensemble; $\mathbf{X}_i$: an unlabeled example; $\mathbf{C}$: the calibration set; α: the desired significance level.
output : $\Gamma(\mathbf{X}_i, \alpha)$: the prediction interval for $\mathbf{X}_i$ at the requested significance α

```
foreach ℓ_t ∈ ℓ do                                   // in order t ∈ [1, T]
    Compute the prediction p_t = ℓ_t(X)
ŷ = (1/T) Σ_t p_t // Compute ensemble prediction
Compute the sorted non-conformity scores S using Algorithm 3
φ = S[⌊(1 − α) · |C|⌋]
Γ(X_i, α) = [ŷ − φ, ŷ + φ]
return Γ(X_i, α) // Return the prediction interval
```

Algorithm 3: Update Non-conformity Scores (ℓ, $\mathbf{C}$)

input : ℓ: the current decision tree ensemble; $\mathbf{C}$: the calibration set, $(\mathbf{X}_i, y_i)$, $i \in [1, c]$; ℓ_{oob}: mapping from calibration examples to learners.
output : $\mathbf{S}$: the sorted set of non-conformity scores

```
foreach (X_i, y_i) ∈ C do
    // Only use learners for which (X_i, y_i) was oob
    foreach ℓ_t ∈ ℓ_oob(X_i) do
        ŷ_i^t = ℓ_t(X_i) // Compute oob prediction
    ŷ_i = (1/|ℓ_oob(X_i)|) Σ_t ŷ_i^t // Aggregate oob predictions
// Compute non-conformity scores.
S = |y − ŷ| // y true labels, ŷ oob predictions
Sort and return the non-conformity scores S
```

to the equivalent batch bagging model. We note that λ can be set to larger values as done by Bifet et al. (2010b), however we follow the original work by Oza and set $\lambda = 1$.

If k is zero for some tree in the forest, we add the example to a bounded set, $\mathbf{C}$, of calibration examples. This set is the equivalent of the *calibration set* in batch inductive conformal prediction. The set is kept to a fixed size c by using a FIFO policy. When it is time to make a prediction, these examples are used to compute the *non-conformity scores* by using Algorithm 3. To ensure we compute non-conformity scores only on out-of-bag examples, we keep track of the learners for which the current example is out-of-bag, by using a mapping from example identifier to learner index.

The first part of the prediction step in Algorithm 2 works similarly to any bagging algorithm. It computes and averages the predictions from each member of the ensemble (Algorithm 2, lines 1-3). However, in order to compute the interval size, it also needs to have up-to-date non-conformity scores, which Algorithm 3 is responsible for. This algorithm takes the calibration set $\mathbf{C}$, current ensemble ℓ, and the ℓ_{oob} mapping as input, and produces a sorted (ascending) list $\mathbf{S}$ of non-conformity scores. As in batch ICP for random forests (Johansson et al., 2014a), our algorithm makes a prediction for each example in the calibration set by using only the trees for which it was an out-of-bag sample. Then, it uses a *non-conformity function* to compute the non-conformity scores. For this work, we choose the absolute error as our non-conformity function, which is a standard choice for conformal regression models.

Once we have the non-conformity scores $\mathbf{S}$, we can calculate the interval for the prediction. Given the desired significance level α and a sorted list $\mathbf{S}$ with c elements, the algorithm picks the element in the list whose index is $\lfloor (1-\alpha) \cdot c \rfloor$. The value $\phi = \mathbf{S}[\lfloor (1-\alpha) \cdot c \rfloor]$ represents half the interval width for the current prediction at the given significance level α. Finally, we use this interval width in line 6 of Algorithm 2 to produce a prediction interval for the example.

This version of ICP produces the same interval size for all examples, given a fixed calibration set $\mathbf{C}$. While techniques that adapt the interval size based on an approximation of the difficulty to predict each example exist (Papadopoulos et al., 2011; Papadopoulos and Haralambous, 2011), we elect to use the simpler version of ICP. Since the calibration set is continuously updated, the interval sizes will also be updated based on the performance of the underlying learner.

Our proposed method is designed as a meta-algorithm, therefore any online tree-based regression algorithm can be used as the underlying learner. For this paper, we use the Fast and Incremental Model Tree (FIMT) algorithm (Ikonomovska et al., 2011). FIMT is an extension to the Very Fast Decision Tree (Domingos and Hulten, 2000) and it inherits its memory management features, such as deactivating the least promising leaves or dropping unpromising attributes in order to keep the memory requirements of the algorithm bounded. It also provides a concept drift adaptation mechanism that allows our methods to adapt to changes in the underlying data distribution. We call this meta-algorithm CPExact.

3.1.2. Optimizations

In the previous section, we mention that the set of calibration examples is bounded, that is, we ensure that we never have to keep more than $|\mathbf{C}| = c$ examples in memory. In our case,

the size of the calibration set affects the computational cost significantly, as in the worst case we need to make an ensemble prediction for each calibration example before making an interval prediction.

The naïve way to implement Algorithm 3 would require each member of the ensemble to make a prediction for each (out-of-bag) example in the calibration set. However, this step is unnecessary; by storing the predictions of each learner for which a calibration example was out-of-bag, we only need to update that prediction when the learner itself gets updated, otherwise we can use its previous prediction. We call this technique prediction memoization.

While this small optimization can reduce the number of required predictions somewhat, the factor of c still remains, which implies that for all ensemble members that have been updated since the last prediction, up to c new predictions are needed to compute the updated calibration scores. In our implementation, we mitigate this issue by using parallelism across the members of the ensemble and updating the calibration scores asynchronously as soon as the training step has finished, thus allowing for overlapping computation until the next call to the prediction function.

3.1.3. Approximate algorithm

In the common prequential evaluation scenario of online learning (Gama et al., 2009), the model makes a prediction for an example before being shown its label, which is then used (together with the example) to update the model. Given that the model is updated for each new example, in the worst case, we need to calculate the non-conformity scores for the complete calibration set for each example, thus incurring a high computational cost. While prediction memoization and asynchronous execution can reduce the runtime of CPExact to some extent, the core computational constraint of updating the predictions for the complete calibration set remains.

To address this computational issue, we propose to use an approximate solution to the computation of the non-conformity scores. In particular, we only use the prediction from the model obtained when the new example enters the calibration set.

In other words, regardless of whether a tree has been updated, we never re-evaluate its predictions for all of its calibration examples. Rather, we only make predictions for the newly introduced examples, i.e., those examples that have been added to the calibration set since the last time we made a prediction. Note that, as more and more examples accumulate, the predictions of the underlying models for a given example will change progressively less. By not making predictions on older examples, we achieve significant computational savings and remain true to the online spirit of the algorithm. Moreover, by constantly introducing new examples in the calibration set, we ensure that it remains an accurate representation of the non-conformity of the examples given the current model. We call this meta-algorithm CPApproximate.

CPApproximate has an inherent tradeoff: when many samples are used in its calibration set, it might end up using several stale predictions to calculate the intervals. As we show in Section 4.4, this tradeoff has an effect on the ability of the algorithm to adapt to concept drift. By maintaining predictions that were made using a distribution different from the current concept, the algorithm is slow to adapt its intervals to the newly introduced concept. This delay can lead to intervals that do not maintain the requested error rate.

3.2. Online Quantile Regression Forests

In Section 2.3, we mention that batch QRF requires the complete dataset to be available in order to assign a weight to each example, and has large memory requirements due to the need to store a mapping from the weight of each example to its label, for every data point.

The main idea for our adaptation of the algorithm to the online setting is to use an approximate representation of the label values to bound the memory requirements of the algorithm and doing away with the need to maintain a weight for each example. In particular, we employ the *KLL* streaming quantile estimates proposed by Karnin et al. (2016) to store the approximate cumulative distribution of label values in the leaves. By using the online bagging algorithm of Oza (2005) in combination with a feature subset selection scheme (Gomes et al., 2017), we propose OnlineQRF (oQRF): online regression forests that provide quantile estimates.

We now briefly explain the core concepts of the KLL sketch, and refer the reader to Karnin et al. (2016) for a detailed description. Estimating streaming quantiles is an established problem in data management. The task is to estimate quantile values from an incoming stream of potentially infinite real values, by utilizing minimal memory and computational cost. See the work by Greenwald and Khanna (2016); Wang et al. (2013) for surveys on the subject. The problem can be formulated as finding the approximate rank of a given value x in a sorted set of n items. An ϵ approximate quantile sketch returns the approximate rank up to an additive error of ϵn. As customary, the KLL algorithm gives a probabilistic guarantee that the resulting sketch is an ϵ-approximation with probability $1-\delta$ (over its executions). Another important property is the *mergeability* of a sketch; this allows us to sketch different parts of a stream in parallel, and merge the partial sketches to a final sketch that has the same accuracy of a single sketch over the complete stream. This scenario is common in distributed settings (De Francisci Morales and Bifet, 2015), and we make use of this property to merge the quantile sketches of each tree in the ensemble into a single quantile prediction.

KLL belongs in the family of methods for quantile estimation that make use of *compactors*, a data structure that can store c items, with the same weight w. Compactors can compact their c elements into $c/2$ by first sorting them, and then discarding either the odd or even items, while doubling the weight of the remaining items. Multiple compactors can be chained together to create a rank estimator; each compactor takes a stream as input and outputs a halved stream to be used as input for the next compactor, until a sequence of length c is generated, which can be simply stored in memory. For a new value x, we can estimate its approximate rank in the original stream from its rank the final sequence of c items. The KLL sketch uses a cascade of such compactors and samplers to process the stream and selectively store a small number of elements, which grows logarithmically with the number of elements in the stream. The only parameter of the algorithm is K, which determines a tradeoff between the memory cost and accuracy of the algorithm. Higher K means more elements will be stored, but also higher accuracy. Note that the error ϵ is not set explicitly by the user, but depends on K.

Specifically, the KLL sketch is an improvement upon the quantile sketch of Agarwal et al. (2012), which chooses whether to keep the even or odd positions during compaction by flipping a fair coin, and subsamples the stream before feeding it to the first compactor. KLL

improves the memory footprint of the algorithm by Agarwal et al. by using compactors of exponentially decreasing size and replacing some of the compactors with a sampler, thus resulting in a mergeable sketch with a space complexity of $O((1/\epsilon)\log^2\log(1/\delta\epsilon))$, which is currently the state of the art. Their limited memory footprint and mergeability make them perfect candidates to be used as quantile estimators for OnlineQRF.

Similarly to online conformal prediction, OnlineQRF is meta-algorithm for which any tree-learning algorithm can be used. For consistency, we again choose to use a variation of the FIMT algorithm, but this time we maintain an efficient quantile sketch at each leaf. We use these sketches at prediction time to provide the prediction intervals of the forest. The parameter K of the KLL sketches constitutes the only parameter of the algorithm.

While QRF is a consistent estimator of the conditional distribution, i.e., the empirical distribution used at the leaves of QRF converges to the real conditional distribution when the number of observations tends to infinity, OnlineQRF lacks this property. Indeed, the use of sketching introduces a fixed error in the approximation of the distribution, which is guaranteed to be smaller than ϵ, but is nevertheless constant. Therefore, adding observations to the sketch does not improve the approximation of the real conditional distribution beyond a given point. Looking at the issue from a different angle; we cannot hope to approximate an arbitrary distribution with infinitesimal error by using a fixed-size sketch. The loss of consistency of the estimation is thus a necessary sacrifice to make the algorithm viable in an online setting.

Algorithm 4: OnlineQRF Training $(\ell, (\mathbf{X}_i, y_i)), \lambda$

input : ℓ: the current decision tree ensemble; $(\mathbf{X}, y)$: a labeled training example.
foreach $\ell_t \in \ell$ **do** `// in order` $t \in [1, T]$
  k = Poisson(1)
  if $k > 0$ **then**
    Train ℓ_t with $(\mathbf{X}, y)$, weighted by k
    Add y_i to H_t, the sketch at the leaf where the example is sorted into.

Algorithm 4 describes the training step of OnlineQRF, which is similar to the original training for bagged learners by Oza (2005). However, we modify the underlying learners to maintain an online quantile sketch of the labels. After updating the learner with the example $(\mathbf{X}_i, y_i)$, we add the label y_i to the corresponding sketch at the leaf, which is a constant time operation.

We describe the prediction process of OnlineQRF in Algorithm 5. As opposed to the original QRF algorithm, which requires summing over all the data points (Eq. 2), our algorithm only requires $T-1$ merge operations, and one operation to calculate the quantiles, both of which take constant time.

Because of the online nature of the sketches, we are able to merge them incrementally, and so maintain the constant memory requirements of the algorithm.

Additionally, when executing the algorithm in parallel, we are able to use a parallel reduce operation, therefore further reducing the runtime of the predict operation, in which the sketch merge is the most expensive operation. Given that the merge operation is the most time consuming operation for the algorithm, we also experimented with approximate alternatives, such as computing the quantiles for each sketch separately and averaging the

Algorithm 5: OnlineQRF Interval Prediction (ℓ, $\mathbf{X}_i$, α)

input : ℓ: the current decision tree ensemble; $\mathbf{X}_i$: an unlabeled example; α: the desired significance level.
output : $\Gamma(\mathbf{X}_i, \alpha)$: the prediction interval for $\mathbf{X}_i$ at the requested significance α

```
Ĥ = ∅ // Initialize empty sketch
foreach ℓ_t ∈ ℓ do                                   // in order t ∈ [1, T]
    Drop X_i down the tree ℓ_t until we arrive at a leaf
    Retrieve the sketch of the leaf, H_t
    Merge H_t into Ĥ
Γ(X_i, α) = [Q_{α/2}(Ĥ), Q_{1−α/2}(Ĥ)]
return Γ(X_i, α) // Return the calculated interval
```

results, but the intervals produced were not valid. This result indicates that information from all the trees needs to be aggregated in order for the algorithm to work as expected, which is also the case for batch QRFs.

One positive characteristic that OnlineQRF shares with the online conformal prediction methods developed for this work is that the significance level α does not need to be set from the start. Rather, it is a runtime parameter that the user can set when making a prediction. Therefore, the user may also request multiple significance levels without the need to train separate models, which is necessary in QR algorithms that modify the loss function, such as linear QR (Koenker, 1996).

4. Empirical evaluation

In this section, we present results from an extensive empirical investigation, concerning 20 small-scale datasets, drawn from a diverse set of domains, and another 10 datasets for studying the effect of concept drift, with millions of data points.

4.1. Experiment Design

The evaluation follows the prequential evaluation design that is commonly employed in online learning settings (Gama et al., 2009). For each example in the dataset, we first make a prediction, then update our metrics based on the predicted and true labels, and finally, reveal the true label to the algorithm. When reporting results over time for the concept drift datasets, we use tumbling windows of size 10 000, that is, we measure the mean performance of each algorithm for every 10 000 samples. Because random forests are non-deterministic, we repeat each experiment 10 times. We report the mean of the metrics across the repetitions and we show the standard deviation as a shaded area in Figures 5 and 6.

4.1.1. Metrics

We report the Mean Error Rate (MER), Relative Interval Size (RIS), and use Quantile Loss and Utility as combined metrics of the MER and RIS. The Mean Error Rate measures the percentage of errors that a method incurs, essentially the average of the 0-1 loss over the complete dataset:

$$\text{MER} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{y_i \notin [l_i, u_i]\}},$$

where N is the size of the dataset and $[l_i, u_i]$ the predicted interval for $\mathbf{X}_i$. This metric quantifies the *validity* of the method; for a significance level α, a method should have an MER of at most α in order to be considered valid. For example, in our validity experiments, we set a significance level of 0.1, so we should observe an MER of at most 0.1.

However, the MER metric offers only a partial view of the performance of a method; it is easy to produce a naïve algorithm with a valid MER by always predicting very large intervals, which however would not be informative. To measure how informative the produced intervals are, a property sometimes referred to as the *efficiency* of the intervals (Vovk et al., 2005), we employ the Relative Interval Size (RIS) metric. RIS measures the average size of the intervals over the data, normalized by the range of values in the data to enable comparisons across datasets:

$$\text{RIS} = \frac{1}{N}\sum_{i=1}^{N} \frac{u_i - l_i}{\rho},$$

where ρ is the range of the dependent variable (label) of the dataset, $\rho = max(\mathbf{y}) - min(\mathbf{y})$.

Lower values are better for the RIS, with a value of one indicating a non-informative interval that covers the whole range of observed values for the target. Values larger than one are possible for methods that can produce interval predictions outside the range of observed values, such as the conformal prediction methods. RIS needs to be viewed together with the MER to get a complete picture of the performance of a method.

To ease presentation, we also employ two combined metrics, Quantile Loss and Utility. Quantile Loss is based on the established quantile error objective function used in quantile regression (Koenker, 1996). Since in our case we are dealing with intervals, we can take the quantile loss for each quantile, $Q_{\alpha/2}$ for the lower and $Q_{1-\alpha/2}$ for the upper, and sum their losses. So for the case of a correct interval prediction, when $y \in [l, u]$ we have:

$$\begin{aligned}\text{Quantile Loss}_{y\in[l,u]} &= \alpha(y-l) + (1-(1-\alpha))(u-y)\\ &= \alpha(u-l).\end{aligned}$$

While in the case where $y \notin [l, u]$ and, for example, $y < l$:

$$\begin{aligned}\text{Quantile Loss}_{y<l} &= (1-\alpha)(l-y) + (1-(1-\alpha))(u-y)\\ &= \alpha(u-l) + (l-y).\end{aligned}$$

The loss is symmetric for the case where $y > u$. The purpose of this metric is to penalize methods that deviate from the requested error rate, according to the distance from the error rate, scaled by the size of the interval. Combining the above we can define Quantile Loss as follows:

$$\text{Quantile Loss} = \text{RIS} \cdot \alpha + \beta.$$

We use RIS here to normalize the interval widths and allow comparison across datasets. The factor β is the distance of the true y value from the closest interval boundary:

$$\beta = \frac{\mathbb{1}_{(y<l)}|y-l| + \mathbb{1}_{(y>u)}|y-u|}{\rho}.$$

In other words, β tells us how far outside the predicted interval the true value lies. Like the RIS, β is normalized by the range of the dependent. While the quantile loss is an established metric, it does not match exactly our problem definition, as it penalizes conservative methods whose MER is lower than the requested significance. For that purpose we also present a metric that does not penalize conservative methods, called Utility.

Utility is a single metric based on the time-utility functions commonly used in real-time systems (Ravindran et al., 2005). These functions combine two measurements: the *utility* of a result as a factor of the *time* it takes to obtain it. We set a time deadline after which the method is penalized, while before the deadline the method incurs no penalty. We use $(1 - \text{RIS})$ as the utility, and set the requested significance as the "time" deadline, with exponential decay. Formally, we define utility as:

$$\text{Utility} = \begin{cases} 1 - \text{RIS}, & \text{MER} \leq \alpha \\ (1 - \text{RIS}) \cdot \exp\left(-\gamma(\text{MER} - \alpha)\right), & \text{otherwise,} \end{cases}$$

where α denotes the requested significance level, and γ is set such that the half-life of the utility is at $1.5 \cdot \alpha$. In other words, once a method's MER passes the requested significance, its utility drops off exponentially, and reaches half its original value when the MER is 50% larger than the requested significance. This metric takes values from 0 (worst) to 1 (best) and indicates how useful the produced intervals are. If the algorithm is able to maintain the requested significance level, $1 - RIS$ indicates how tight the intervals are. Otherwise the intervals are invalid and we discount the utility accordingly. Finally, if the RIS is larger than one we set the utility to zero.

4.1.2. Baselines

We compare our algorithms with the state-of-the-art Mondrian Forest (MF) algorithm (Lakshminarayanan et al., 2016). MF makes use of Mondrian processes (Roy and Teh, 2009) to create tree-like online prediction models that use a hierarchy of Gaussians to model the dependent variable and can produce the full predictive posterior distribution. Mondrian Forest maintains a Gaussian at each *node* in the forest, consequently it has an unbounded computational cost that increases with each incoming example. In comparison, our methods perform learning only at the leaves, and are thus more efficient computationally. We provide more details for the method in Section 5. As done in the original work, we use the predicted mean and variance to extract Gaussian quantiles for the requested significance level.

We also employ a simple baseline as an illustration that the problem is non-trivial, similar to the one used by Lakshminarayanan et al. (2016). We compute the predicted mean and variance among the trees in an online forest that uses FIMT as the underlying learner, and

Dataset	Examples	Features	Domain
2dplanes	40768	10	Artificial
abalone	4177	8	Biology
ailerons	13750	40	Control
bank32	8192	32	Simulation
calHouse	20640	8	Housing
cpuAct	8192	21	Computing
elevators	16599	18	Control
energy	19735	27	Energy Use
friedman	40768	10	Artificial
house8	22784	8	Housing
house16	22784	16	Housing
kin8nm	8192	8	Robotics
mv	40768	10	Artificial
newsPop	39644	60	Web
puma8	8192	8	Robotics
puma32	8192	32	Robotics
qs240	6003	43	Bioinformatics
qs253	4332	26	Bioinformatics
sulfur	10081	6	Monitoring
yprop	8885	251	Drug Design

Table 1: The small-scale datasets used in this study.

take Gaussian quantiles to create the prediction intervals. We report the results of this method in Section 4.2.

4.1.3. Data

We use 20 small-scale datasets from a variety of domains to test the validity and efficiency of the algorithms, and 10 data sets to test their ability to deal with concept drift. The small-scale datasets are gathered from the OpenML repository (Vanschoren et al., 2013). Table 1 provides summary information about these datasets. We provide more detailed information about the small-scale data in Appendix A.1.

For the concept drift experiments, we follow Ikonomovska et al. (2011) and generate datasets using the three "Friedman" functions first introduced in Friedman's MARS paper (Friedman, 1991). The first Friedman function includes non-linear dependencies between five relevant features and the dependent variable, and in addition includes five irrelevant features that do not affect the dependent variable. All features are independent and uniformly distributed over $[0, 1]$. The second and third Friedman functions include four uniformly distributed features and a noise factor. The dependent variable simulates the impedance and phase shift in an alternating current circuit. We refer the reader to Friedman (1991) for more details on these functions.

As done by Ikonomovska et al. (2011), we introduce three different kinds of concept drift for each original function, and produce 1M data points for each dataset, for a total of nine artificial concept drift datasets. We provide more information about the exact function definitions in Appendix A.2, and give a brief overview of each type of concept drift below.

- *Local expanding*: The first type of drift appears only in two distinct regions of the input space. There are three change points, one every 1/4 of the data, which modify the generating function if the point lies in one of two designated regions. These regions expand with each consecutive change point.
- *Global reoccurring abrupt drift:* This type of drift is global and abrupt. After 1/2 of the examples have been observed, we abruptly change the generating function. After 3/4 of the examples have been observed, we change back to the original function.
- *Global slow:* The third type of drift is also global but gradual. Starting after 1/2 of the examples have been observed, we gradually introduce data points from a different concept, with a probability that linearly increases to one after 100k additional points have been observed. At the 3/4 point, we introduce a new concept in a similar manner, which completely replaces the previous concept after 100k points.

We also use a real-world dataset of flight delays. [2] This dataset contains flight arrival and departure details for all commercial flights in the US during 2008, with the task being to predict the delay of flights based on attributes such as the age of the airplane or the day of the year. This dataset was already used in the evaluation of Mondrian Forests (Lakshminarayanan et al., 2016), and we have re-created the same splits of the data in samples with 700K, 2M, and 5M data points. [3] We perform a standard pre-processing step to transform some categorical attributes, such as day of week or month, to a one-hot (binary) representation, resulting in a dataset with fifty-five features.

4.1.4. Parameter Settings

For all the experiments, we use an ensemble size of 10. We use $|C| = 1000$ calibration examples for the conformal prediction algorithms, while for OnlineQRF we use quantile sketches at the leaves with the default accuracy parameter $K = 200$, as recommended by Karnin et al., which yields a normalized rank error of 1.65%.

We experimented with different numbers of K and calibration examples, and these parameters did not significantly affect the results. We have implemented these algorithms in the MOA online learning framework (Bifet et al., 2010a). For the Mondrian Forest, we use the optimized implementation available in the `scikit-garden`[4] Python library. We have verified with the author of the original work that this implementation produces correct results in online learning mode for the parameter settings we use. Specifically, we set `min_samples_split` = 2, which determines the minimum number of samples necessary for a leaf to be considered for a split. We note that for values larger than this, MF requires that the entire dataset be stored in memory at the leaves, making it infeasible for the large datasets and limited resources common in online learning.

2. `http://stat-computing.org/dataexpo/2009`
3. We thank M. Deisenroth for providing the pre-processing scripts.
4. `https://github.com/scikit-garden/scikit-garden`

Dataset	MER	RIS	Dataset	MER	RIS
2dplanes	0.394	0.10	house16	0.367	0.11
abalone	0.727	0.05	kin8nm	0.659	0.14
ailerons	0.271	2.89	mv	0.147	0.16
bank32	0.413	0.20	newsPop	0.053	0.06
calHouse	0.524	0.15	puma8	0.581	0.22
cpuAct	0.525	0.15	puma32	0.502	0.24
elevators	0.392	560.95	qs240	0.448	0.28
energy	0.137	111.9	qs253	0.510	0.48
friedman	0.379	0.12	sulfur	0.462	0.04
house8	0.405	0.09	yprop	0.073	8.32

Table 2: Mean Error Rate and Relative Interval Size for the Gaussian quantiles baseline. Desired MER is 0.1.

We set the requested significance level α to 0.1 for the validity experiments, but explore the effect of different significance levels in Section 4.5.

4.2. Baseline

We start by reporting the validity and interval sizes of the Gaussian quantiles baseline method, as a way to illustrate the difficulty of the problem. Table 2 shows the MER and RIS for the Gaussian quantiles baseline, with a desired MER of 0.1. As can be seen, the observed mean error rate is several times larger than this; the average MER over all datasets is 0.4. This indicates that merely assuming a Gaussian distribution over the trees' predictions is not an appropriate way to quantify the uncertainty.

4.3. Small-scale data

Figure 1 breaks down the MER and RIS for each method, while Figure 2 reports the utility for the small-scate data. Each point in Figure 1 represents the average MER and RIS of a method for a single dataset, over the ten repeats of each experiment.

The ability to maintain the desired error rate, and the required interval size to do so, differ significantly between methods. OnlineQRF is the overall best performing method for these datasets, managing to strike a balance between maintaining the requested error rate and having small prediction intervals. On the other hand, CPExact is the only method that is consistently valid for all the datasets, i.e., stays below the requested 0.1 error level on average. However, it will sometimes produce intervals with an RIS close or larger than one, leading to a utility near zero for those datasets. As a result, its mean utility is lower than OnlineQRF, with CPApproximate following closely. Mondrian Forests, while generally producing tight intervals, manage to maintain the requested error rate on average for only nine out of the twenty datasets, leading to the lowest utility among all methods.

In Figure 3 we can see that, modulo one outlier for the conformal prediction methods, all methods perform roughly equally in terms of quantile loss. A slight advantage for the

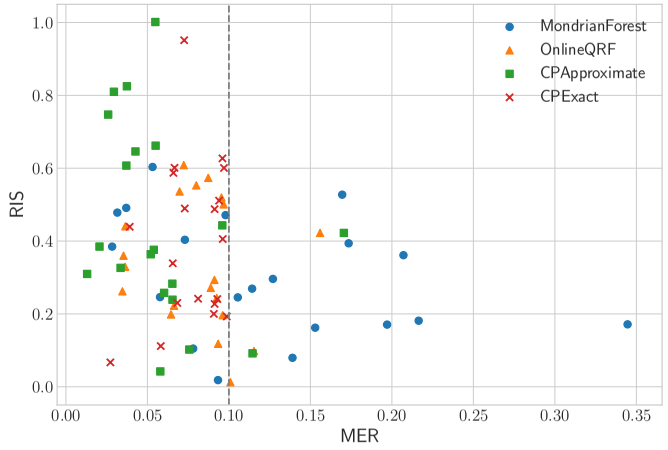


Figure 1: MER and RIS for small-scale data. Each point corresponds to one dataset. The dashed vertical line indicates the desired error rate (significance) at 0.1.

methods proposed in this paper can be seen in Figure 2 in terms of utility, which unlike quantile loss does not penalize methods with conservative predictions.

These results can, to some degree, be explained by the nature of each algorithm. It is likely that aggregating the mixture of Gaussians produced by the Mondrian Forest to a single Gaussian distribution to be used for quantile estimation is the main source of errors for the method. In MF, each node in the path from the root to the leaf for an example produces a mean and a variance. These are averaged to produce one mean and variance per tree. Then the means and variances of all trees in the ensemble are averaged again to produce a final Gaussian distribution that is used to produce the intervals. These two levels of aggregation are a potential source of error for MF.

Unlike MF and the CP methods, the intervals of OnlineQRF never lie outside the range of observed labels, thus making the method less likely to produce outliers which the conformal prediction methods suffer from, as exemplified in Figure 1. Finally, we note the ability of CPApproximate and particularly CPExact to maintain the expected error rate. However, this result often comes at the cost of larger interval sizes and, as shown later in Section 4.6, significantly increased runtime for CPExact.

4.4. Concept Drift

In this section, we examine the validity of the algorithms on datasets with concept drift.

We illustrate the performance of the algorithms on the artificial Friedman #1 data in Figure 5. The results are consistent with the other Friedman datasets, which we summarize in Figures 7 and 8 for utility and quantile loss respectively, and Figure 4 for a combined view into MER and RIS.

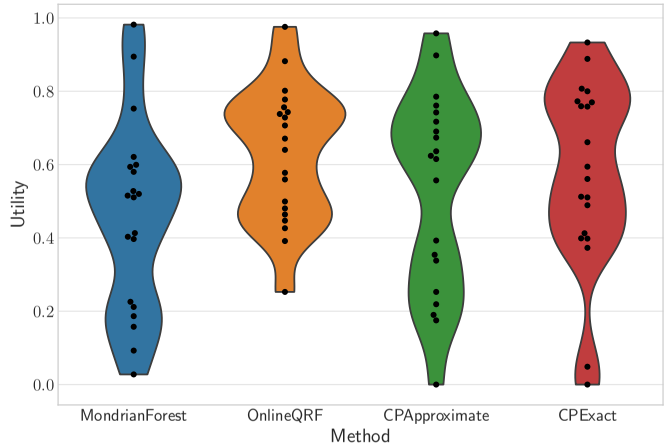


Figure 2: Utility for the small-scale data. Higher is better.

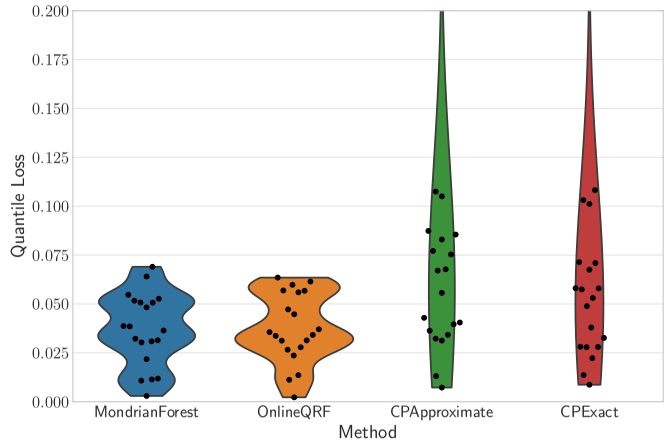


Figure 3: Quantile loss for the small-scale data. Lower is better. The conformal prediction methods have an outlier for the `yprop` dataset which is not shown here to ease presentation.

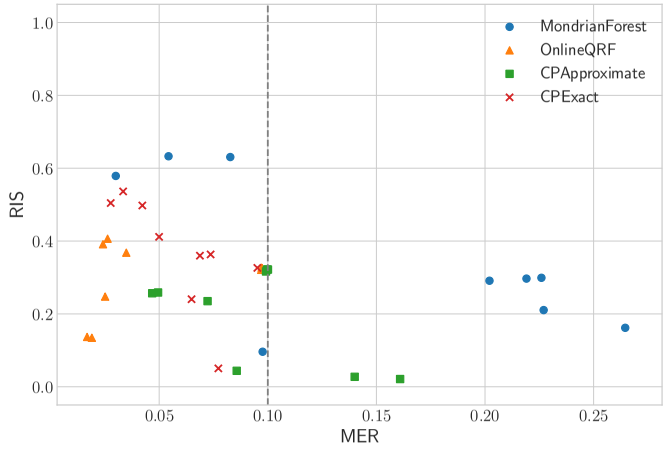


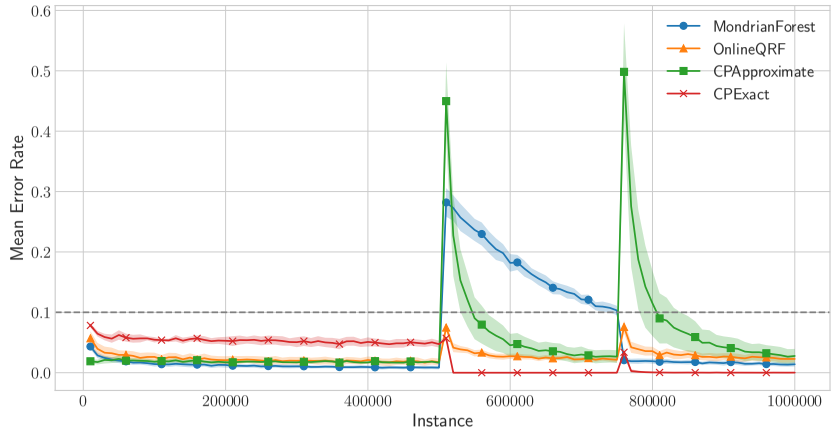
Figure 4: MER and RIS for Friedman data. Each point corresponds to one dataset. The dashed vertical line indicates the desired error rate (significance) at 0.1.

Starting from Figure 5, the first concept drift occurs at 500k examples, and the second at 750k. We can observe different behaviors of the algorithms. Starting with CPExact, after the first concept drift, its RIS rises rapidly to account for the additional uncertainty in the model. As a result, it manages to maintain the requested error rate throughout the experiment, but its intervals remain large after the concept drift. In contrast, CPApproximate maintains a stable RIS throughout the experiment, with a MER that after an initial spike drops off exponentially to return to the requested error rate. This behavior follows from the fact that the method does not update the old non-conformity scores.

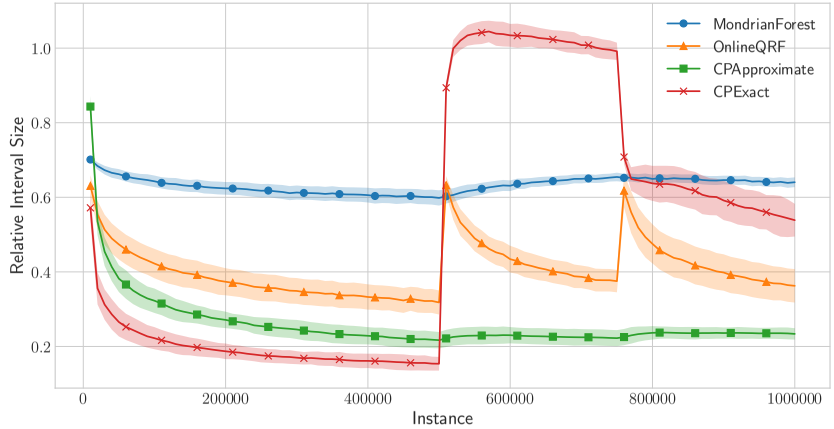
OnlineQRF lies in the middle between the two; it also exhibits a spike in RIS at the concept drift points, but it is quick to recover and, importantly, it remains valid throughout the experiment. Finally, Mondrian Forest has consistently wide intervals, and its error is slow to recover after the first concept drift.

As with the small-scale datasets, OnlineQRF is a strong performer, providing consistent results. However, in this set of experiments, CPApproximate performs better than CPExact due to its tight intervals, and actually has the best average utility among all methods, with OnlineQRF nearly matching it. Mondrian Forest is only able to maintain the requested error rate for half of the datasets (Figure 4), and even in those it mostly has the lowest utility.

The performance of the algorithms in the real-world flight delay data is shown in Figure 6 for the two million examples dataset. The concept drift occurs after having observed around 1.1 million examples, which causes the MER of all the algorithms to spike considerably. Mondrian Forest and OnlineQRF are able to return to the desired error rate before observing 1.2 million examples. On the other hand, CPApproximate requires close to 900 000 additional

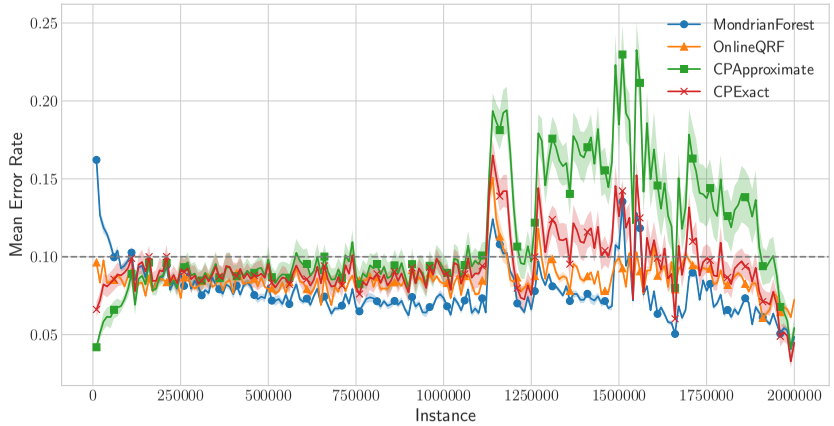


(a) Friedman #1 Mean Error Rate. The dashed horizontal line indicates the desired error rate at 0.1.

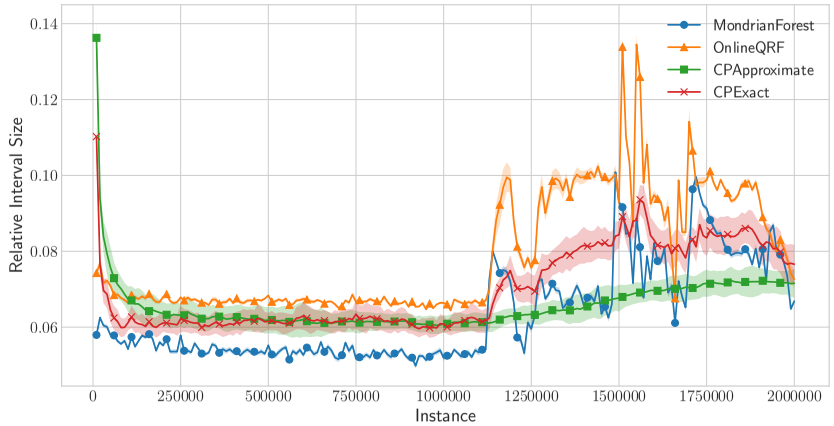


(b) Friedman #1 Relative Interval Size.

Figure 5: MER and RIS evolution over time for Friedman #1 with global abrupt drift.



(a) Flight delay Mean Error Rate.



(b) Flight delay Relative Interval Size.

Figure 6: MER and RIS evolution over time for the flight delay dataset.

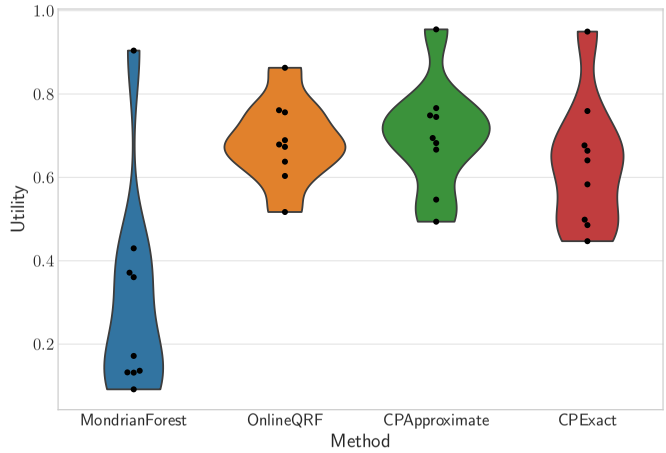


Figure 7: Utility for the concept drift data. Higher is better.

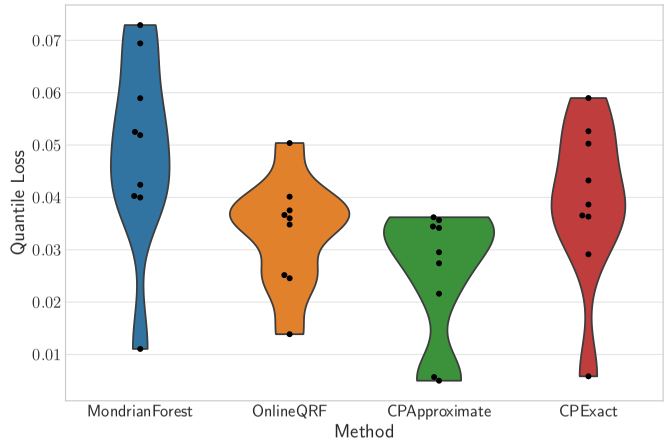


Figure 8: Quantile loss for the concept drift data. Lower is better.

Method	Metric	α				
		0.3	0.2	0.1	0.05	0.01
MF	MER	0.28	0.20	0.13	0.094	0.062
	RIS	0.19	0.23	0.29	0.349	0.460
oQRF	MER	0.23	0.14	0.07	0.036	0.015
	RIS	0.20	0.23	0.31	0.345	0.510
CP_{approx}	MER	0.22	0.13	0.06	0.032	0.006
	RIS	0.21	0.31	0.48	0.785	1.130
CP_{exact}	MER	0.25	0.16	0.08	0.039	0.009
	RIS	0.28	0.37	0.57	0.622	0.944

Table 3: MER and RIS for different significance levels. The target MER is α where α the significance level. See Appendix B for detailed results.

examples before returning to under the requested error rate. CPExact is able to adapt faster than CPApproximate, and exhibits a lower MER during the concept drift.

Overall, CPApproximate is slow to adapt its intervals, which can be explained by the slow update of its nonconformity scores, but as a result maintains tighter intervals. Conversely, CPExact is able to adapt faster, since it updates the predictions for all elements in the calibration set to derive the non-conformity scores. This means that as the learners shift to the new distribution, so will all the non-conformity scores, as opposed to one-at-a-time with CPApproximate. However, this adaptivity comes at the cost of increased interval sizes, and much slower computation (Section 4.6). Mondrian Forest is also slow to adapt to changes in the underlying distribution, thus leading to large areas where its intervals are wrong, the result of which can be seen in Figure 7, where its utility is the lowest among all methods.

OnlineQRF provides a good compromise, both for the artificial and the real-world datasets. Its intervals are quickly adapted to distribution changes, which makes it recover the requested error rates, and also produce tight intervals in a timely manner, therefore making its predictions both valid and efficient.

The situation is similar in terms of quantile loss, shown in Figure 8, with CPApproximate having the lowest median loss, and Mondrian Forest performing the worst among all methods.

4.5. Significance level

Depending on how critical an application is, a user might require different error guarantees. A natural question that arises is how different settings of the significance level affect the methods. We investigate this question by running all the algorithms on the small-scale data with a significance level varying from 0.3 to 0.01, and report the aggregated results, averaged over all datasets, for each method and significance level in Table 3. We provide a more in-depth look with MER/RIS figures similar to Figure 1 in Appendix B. Ideally, we would like the MER to never cross the value α, where α is the significance level, while keeping the RIS low.

Dataset	MF	oQRF	CP_{approx}	CP_{exact}
Small-scale data	41.6	5.4	5.7	102.3
Flight delays	6340	836	899	27863
Friedman	2380	114	213	2011

Table 4: Average running times for all algorithms (seconds).

Table 3 shows that, for larger significance values, Mondrian Forest maintains a valid MER. However, Mondrian Forest fails to meet the MER requirements for significance values below 0.1, which represent critical use cases with little room for errors. OnlineQRF shows a very robust tradeoff between MER and RIS for a range of values of α. It fails to meet the MER requirements only for $\alpha = 0.01$, while maintaining an RIS competitive with MF for all significance levels. Finally, the conformal prediction methods are able to maintain the error guarantee across the range of significance values. The price to pay is increased interval sizes, which with lower significance levels approach, or even exceed, the range of observed label values.

4.6. Computational cost

This section reports the computational cost of each method, which is one of the main considerations when selecting an online learning method. For the running time comparison, note that Mondrian Forest is implemented in Cython using Numpy primitives. Therefore, a perfectly fair comparison with the Java implementations of the algorithms developed for this study is not possible without a full porting, which is outside of the scope of the current work.

However, we mention that for each tree in a Mondrian Forest, the cost to process the nth data point is $\mathcal{O}(\log n)$, and hence growing for each additional data point, so the total cost to process N data points is $\mathcal{O}(\sum_{n=1}^{N} \log n) = \mathcal{O}(\log N!)$. In contrast, the computational cost of the methods developed in this paper, and the underlying FIMT learner, are constant, regardless of the number of data points. They depend only on constant values, such as the number of features for FIMT, sketch accuracy for OnlineQRF, and the size of the calibration set for the conformal prediction methods, and are generally bounded by the number of features p, so the cost to train the nth data point is always $\mathcal{O}(p)$.

Table 4 lists the results; to ease presentation, we average the runtimes of all the different datasets in each group. OnlineQRF is the fastest algorithm for all the data groups, with CPApproximate a close second. We note that both approaches are an order of magnitude faster than Mondrian Forest, confirming the theoretical computational costs of the algorithms. Finally, we can clearly see the drawbacks of CPExact: for every data point, it requires a pass through the entire calibration set to get updated predictions. This incurs a significant computational cost, making it more than an order of magnitude slower than the approximate version of the algorithm.

5. Related Work

Decision trees have attracted a large body of research, both in order to extend them to the online learning setting, and to quantify the uncertainty of their predictions. In this section, we describe work in these two areas that is directly related to this study and refer the interested reader to Kotsiantis (2013) for a general survey of decision tree learning.

Online decision trees and forests

One of the first adaptations of decision tress to the online setting is the Very Fast Decision Tree (VFDT), or Hoeffding Tree (Domingos and Hulten, 2000). Unlike batch decision trees, which recursively partition all examples when growing the tree, VFDT is a single-pass algorithm, where new examples can only affect the leafs of the tree. The statistics of each feature are accumulated at leafs denoted as "learning leafs" which are tested periodically to determine whether they should be split. The Hoeffding bound provides a δ-guarantee on the optimality of the selected split.

While single trees can deal with simpler problems and provide better interpretability, ensemble techniques such as random forests (RF) have been shown to achieve better accuracy for a variety of tasks (Fernández-Delgado et al., 2014). As a result, there have been multiple studies to adapt them to the online learning setting. The bagging part of the RF algorithm was originally adapted to online learning by Oza (2005). Saffari et al. (2009) use a combination of the online bagging algorithm by Oza and propose a tree randomization scheme to adapt random forests to the online setting. Adaptation to concept drift was proposed to be handled by discarding trees that exhibit a high out-of-bag error. Gomes et al. (2017) also leverage the online bagging algorithm by Oza (2005). The authors restrict the number of features that leaves can consider for splitting to create an algorithm that is similar the original random forests by Breiman, but adapted to the online setting through the use of Hoeffding trees as the base learner. The algorithm includes a warning mechanism to deal with concept drift that learns "background" trees, which eventually replace original trees if drift is confirmed for a tree.

Confidence and prediction intervals in tree models

Decision and regression trees can be modified to output not only a single label, i.e., class or regression value, but also some estimate of the *confidence* that the model has in its prediction. Many alternative approaches have been proposed in the batch setting: Mentch and Hooker (2016) use formal statistical inference to produce prediction intervals, Wager et al. (2014) make use of the jackknife to estimate standard errors for random forests, and Chipman et al. (2010) propose Bayesian additive regression trees that estimate a distribution over decision trees which can be used to produce prediction intervals. Johansson et al. (2014b) propose a model for conformal prediction that uses a batch trained decision tree that updates its predictions in an online manner. The structure of the tree however remains static, i.e. no online training is performed.

Lakshminarayanan et al. (2016) use Mondrian processes (Roy and Teh, 2009) to create tree-like online prediction models which have the property of being "extensible", i.e., the distribution of trees in the online version is the same as the batch one. This property can be leveraged to create online random forests that can be updated efficiently and produce

models comparable to their batch version. Mondrian forests are able to predict the full conditional distribution for the predicted variable, and therefore can produce prediction intervals. However, the prediction model is parametric: it models the variable as a hierarchy of Gaussians, one for each node in the tree. The benefit of having access to the complete predictive posterior comes at cost as well: the memory and computational cost for each tree grows for each data point processed, making them unsuitable for unbounded streaming data.

6. Conclusions

In this paper, we proposed two algorithms that quantify the prediction uncertainty of regression forests by producing prediction intervals for unbounded data streams. We have presented an extensive empirical investigation, which shows that our approaches outperform naïve and state-of-the-art methods for a variety of datasets in terms of error rate and interval size, while being an order of magnitude faster to execute.

Comparing the proposed algorithms in terms of practicality, OnlineQRF appears to offer the best trade-off between maintaining the requested error rate and mean interval size, while being computationally efficient. Although the conformal prediction methods are able to better maintain the requested error rate at a confidence level above 95%, they do so at a cost of significantly increased interval size and, in the case of CPExact, an order of magnitude longer running time.

As evidenced by our experiments on the datasets exhibiting concept drift, when the theoretical assumptions of the algorithms are violated, OnlineQRF is able to adapt relatively quickly to the concept drift. CPApproximate is slower to adapt but maintains tight intervals, and while CPExact can maintain the requested error rate, it does so at the cost of the interval's utility. Moreover, its computational cost makes it unsuitable for large scale data. That said, CPExact is the only method that consistently maintains validity in both stationary and concept drift settings, so it is our recommendation in cases where maintaining the error rate is of utmost importance.

This work opens several important research avenues. First, the empirical results presented here can be reinforced by providing validity guarantees for the conformal prediction methods. They can also be extended to incorporate normalization to possibly produce more informative intervals, see e.g., (Boström et al., 2017). Second, we aim to properly address the issues of concept drift at the meta-algorithm level, by introducing mechanisms that inject more uncertainty in the predictions when a concept drift is detected. Finally, we would like to provide more scalable implementations of the methods, for example by developing efficient distributed algorithms for OnlineQRF and the conformal prediction methods.

Appendix A. Data Descriptions

A.1. Small-scale data

In this appendix we provide a brief description of each small-scale dataset used in this study. When possible we have used the description of the data provided by the original authors.

2dplanes: This is an artificial dataset described in Breiman et al. (1984), with variance 1 instead of 2. The 10 attributes are generated independently using:

$$P(X_1 = -1) = P(X_1 = 1) = 1/2$$
$$P(X_m = -1) = P(X_m = 0) = P(X_m = 1) = 1/3, m = 2, ..., 10.$$

We obtain the value of the target variable Y using the rule:

$$\text{if } X_1 = 1 \text{ set } Y = 3 + 3X_2 + 2X_3 + X_4 + \sigma(0,1)$$
$$\text{if } X_1 = -1 \text{ set } Y = -3 + 3X_5 + 2X_6 + X_7 + \sigma(0,1)$$

abalone: The task is to predict the age of abalone from physical measurements. Features include sex, dimensions, and weight.

ailerons: This dataset addresses a control problem, namely flying a F16 aircraft. The attributes describe the status of the aeroplane, while the goal is to predict the control action on the ailerons of the aircraft.

bank32nh: A synthetically generated dataset from a simulation of how bank-customers choose their banks. Tasks are based on predicting the fraction of bank customers who leave the bank because of full queues.

calHousing: The task here is to predict the median house value based on features like median income and median age, collected from the 1990 California census.

cpu_act: A collection of computer systems activity measures. The task is to predict the portion of time that CPUs run in user mode, based on attributes such as number of reads/writes to the system, system calls, and page requests.

energy: Appliances energy prediction dataset. The task is to predict the energy consumption of appliances in a home. From the UCI data repository: `https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction`

elevators: This dataset also concerns the task of controlling a F16 aircraft, although the target variable and attributes are different from the ailerons domain. In this case the goal variable is related to an action taken on the elevators of the aircraft.

friedman: This is an artificial dataset used in Friedman (1991). The examples are generated using the following method: Generate the values of 10 attributes, $X_1, ..., X_{10}$ independently, each of which is uniformly distributed over $[0, 1]$. Obtain the value of the target variable Y using the equation:

$$Y = 10 * sin(\pi * X_1 * X_2) + 20 * (X_3 - 0.5)^2 + 10 * X_4 + 5 * X_5 + \sigma(0,1),$$

where $\sigma(0,1)$ is a random number generated from a normal distribution with mean 0 and variance 1.

house: Both the `house_8L` and `house_16H` datasets are concerned with predicting the median price of a house in a region based on demographic composition and a state of housing market. The number signifies the approximate difficulty of the task, L for low difficulty, H for high.

kin8nm: This dataset is concerned with the forward kinematics of an 8 link robot arm. Among the existing variants of this dataset we have used the variant 8nm, which is known to be highly non-linear and medium noisy.

mv: This is an artificial dataset with dependencies between the attribute values. It contains three nominal and seven continuous features, which contain dependencies of the form `IF` $(X_5 < 0.5)$ `THEN` $X_8 =$ `normal` `ELSE` $X_8 =$ `large`, where $X_5 \sim U(-1, 1)$. The full interactions can be found on the dataset's website [5].

newsPopularity: This dataset summarizes a heterogeneous set of features about articles published on Mashable[6] in a period of two years. The goal is to predict the number of shares in social networks (popularity) based on attributes such as the number of words, images, and videos in the article, publication time and other word-based derived features.

puma: This is a family of datasets synthetically generated from a realistic simulation of the dynamics of a Unimation Puma 560 robot arm. There are eight datasets in this family. The task is to predict the angular acceleration of one of the robot arm's links. The inputs include angular positions, velocities and torques of the robot arm. The number in the name of the dataset indicates the number of features, 8 or 32.

qsar-chembl: This dataset contains QSAR data (from ChEMBL[7] version 17) showing activity values (unit is pseudo-pCI50) of several compounds on drug target ChEMBL_ID.

sulfur: These are measurements for the amount of sulfur recovered by recovery units that remove environmental pollutants from acid gas streams in industrial settings before they are released into the atmosphere. The features are gas and air flows.

yprop_4_1: A drug design dataset, used in Feng et al. (2003). The task is to predict the toxic effects of a substance based on a number of chemical descriptors.

A.2. Friedman Functions

In this section we provide a description for all the Friedman function concept drift datasets used in this study.

We follow the concept drift types and equations used by Ikonomovska et al. (2011) for the first Friedman function, and apply similar transformations for the second and third functions.

Friedman #1

The original Friedman #1 function is the one used for our small-scale data: it generates the values of 10 attributes, $X_1, ..., X_{10}$ independently, each of which is uniformly distributed over $[0, 1]$, out of which only the five first affect the dependent. We obtain the value of the target variable Y using the equation:

$$Y = 10 * sin(\pi * X_1 * X_2) + 20 * (X_3 - 0.5)^2 + 10 * X_4 + 5 * X_5 + \sigma(0,1),$$

where $\sigma(0,1)$ is again drawn from a standard normal distribution.

The concept drift functions are the same as in (Ikonomovska et al., 2011):

Local expanding abrupt drift: At the first change point (250k points) two regions of the input space are introduced, each of which has a different generating function than the original. The first region is $R1 \equiv X_2 < 0.3$ & $X_4 > 0.7$ & $X_5 < 0.3$. The new generating function for

5. http://www.dcc.fc.up.pt/ ltorgo/Regression/DataSets.html (MV Artificial Domain)
6. https://www.mashable.com
7. https://www.ebi.ac.uk/chembl/index.php/

this region is:

$$y_{1,R1} = 10 * X_1 * X_2 + 20 * (X_3 - 0.5) + 10 * X_4 + 5 * X_5 + \sigma(0,1)$$

The second region is $R2 \equiv X_2 > 0.7$ & $X_3 > 0.7$ & $X_4 < 0.3$ & $X_5 > 0.7$. The new generating function for this region is:

$$y_{1,R2} = 10 * cos(X_1 * X_2) + 20 * (X_3 - 0.5) + e^{X_4} + 5 * {X_5}^2 + \sigma(0,1).$$

At the second point of change at 500k points the regions are expanded by removing the last inequality ($x_5 < 0.3$ and $x_5 > 0.7$, respectively). At the third change point (750k instances) we again remove the last inequality ($x_4 > 0.7$ and $x_4 < 0.3$, respectively).

Global reoccurring concept drift: At the first change (500k) point we permute the positions of independent variables so the generating function becomes:

$$y_2 = 10 * sin(\pi * X_4 * X_5) + 20 * (X_2 - 0.5)^2 + 10 * X_1 + 5 * X_3 + \sigma(0,1).$$

At the second change point (750k) the original function reoccurs.

Global slow gradual drift: In this drift we gradually introduce instances generated from a different concept after the change point, with a linearly increasing probability, until 100k instances later the new concept takes over. At the first change point (500k) we introduce points from the function:

$$y_3 = 10 * sin(\pi * X_4 * X_5) + 20 * (X_2 - 0.5)^2 + 10 * X_1 + 5 * X_3 + \sigma(0,1).$$

At the second point of change (750k) we gradually introduce instances from the second concept, which is a permutation of the previous concept:

$$y_4 = 10 * sin(\pi * X_2 * X_5) + 20 * (X_4 - 0.5)^2 + 10 * X_3 + 5 * X_1 + \sigma(0,1).$$

Friedman #2 and #3

The second and third Friedman functions model an alternating current series circuit involving a resistor R, inductor L and capacitor C. A generator places voltage on the circuit with an angular frequency $\omega = 2\pi f$, where f the cyclic frequency.

Then the impedance, Z, and phase shift, ϕ, both depend on the components of the circuit:

$$Z(R,\omega,L,C) = [R^2 + (\omega L - 1/\omega C)^2]^{1/2}$$
$$\phi(R,\omega,L,C) = \tan^{-1}\left[\frac{\omega L - 1/\omega C}{R}\right].$$

We add an error factor to the values, drawn from a standard normal.

The independent variables are uniformly distributed and lie in the range:

$$0 \le R \le 100 \text{ ohms},$$
$$20 \le f \le 280 \text{ hertz},$$
$$0 \le L \le 1 \text{ henries},$$
$$1 \le C \le 11 \text{ microfarads}.$$

Similarly to the above we introduce artificial concept drift to each function:

Local expanding abrupt drift: For the drift regions we use the same criteria as for the first Friedman function, only on normalized values of the independent variables, so that they again belong in the $[0,1]$ range. We again permute the positions of the independent variables when the concept changes. For the impedance Z we use the functions:

$$Z_{1,R1}(R,\omega,L,C) = [C^2 + (\omega R - 1/\omega L)^2]^{1/2}$$
$$Z_{1,R2}(R,\omega,L,C) = [R^2 + (RC - 1/RL)^2]^{1/2},$$

for regions $R1$ and $R2$.

For the phase shift we use the functions:

$$\phi_{1,R1}(R,\omega,L,C) = \tan^{-1}\left[\frac{RC - 1/\omega L}{\omega}\right]$$
$$\phi_{1,R2}(R,\omega,L,C) = \tan^{-1}\left[\frac{\omega C - 1/\omega R}{R}\right],$$

for regions $R1$ and $R2$.

Global abrupt reoccurring: The new concepts that are introduced between 500k and 750k instances for Z and ϕ are:

$$Z_2(R,\omega,L,C) = [C^2 + (\omega E - 1/\omega L)^2]^{1/2} \tag{3}$$
$$\phi_2(R,\omega,L,C) = \tan^{-1}\left[\frac{RC - 1/\omega L}{R}\right]. \tag{4}$$

As before, we use the original functions for instances 0-250k and 750k-1M.

Global slow gradual drift: The concepts introduced gradually at the first change point are the same used above for the global abrupt reoccurring, i.e. Equation 3 for Z and Equation 4 for ϕ.

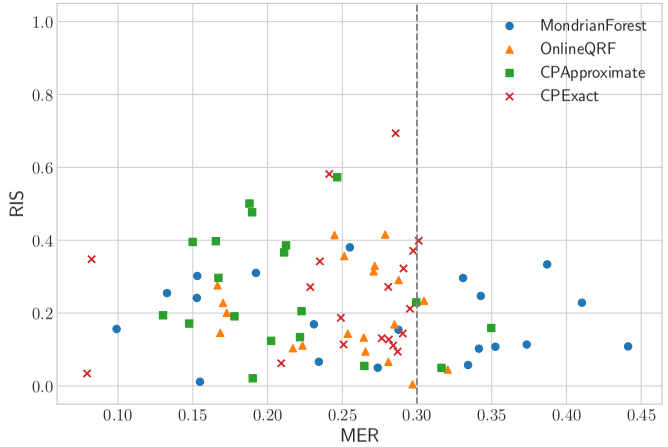
At the second change point we introduce:

$$Z_4(R,\omega,L,C) = [L^2 + (\omega C - 1/\omega R)^2]^{1/2}$$
$$\phi_4(R,\omega,L,C) = \tan^{-1}\left[\frac{\omega C - 1/\omega R}{R}\right].$$

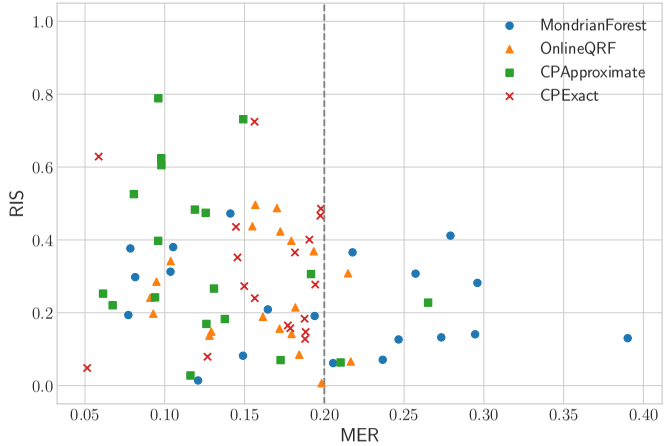
Appendix B. Significance Level Figures

In this appendix we provide combined MER-RIS figures for the confidence experiments of Section 4.5, which are shown aggregated in Table 3. The results for $\alpha = 0.1$ (90% confidence) can be seen in Figure 1 so we omit it here.

These figures clearly demonstrate the inability of Mondrian Forest to maintain the requested error, especially for a confidence of 95% and 99%, as shown in Figure 10. We also note the ability of CPExact to maintain the requested error rate across all experiments.

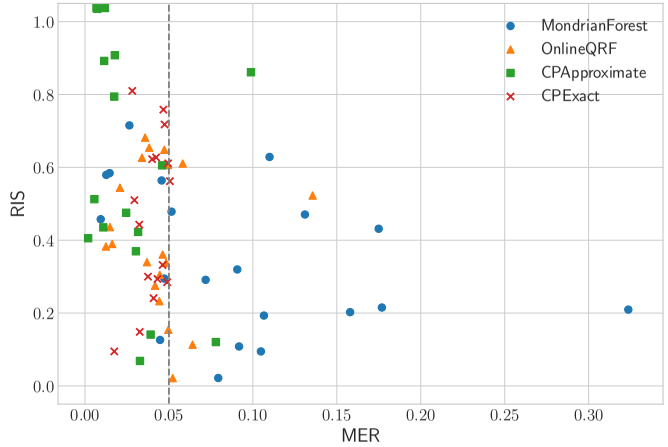


(a) Significance 0.3 (70% confidence).

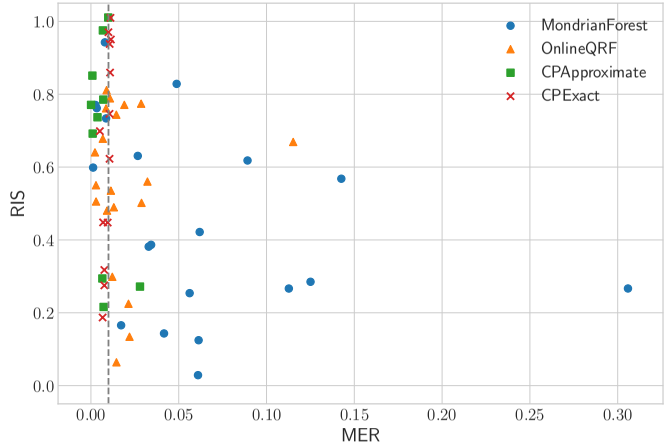


(b) Significance 0.2 (80% confidence).

Figure 9: MER - RIS plots for significance 0.3 (top) and 0.2 (bottom).



(a) Significance 0.05 (95% confidence).



(b) Significance 0.01 (99% confidence).

Figure 10: MER - RIS plots for significance 0.05 (top) and 0.01 (bottom).

References

Pankaj K. Agarwal, Graham Cormode, Zengfeng Huang, Jeff Phillips, Zhewei Wei, and Ke Yi. Mergeable summaries. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '12, pages 23–34, 2012.

Yael Ben-Haim and Elad Tom-Tov. A streaming parallel decision tree algorithm. *Journal of Machine Learning Research*, 11:849–872, 2010.

Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive online analysis. *Journal of Machine Learning Research*, 11:1601–1604, 2010a.

Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. Leveraging bagging for evolving data streams. In *Machine Learning and Knowledge Discovery in Databases*, pages 135–150, 2010b.

Henrik Boström, Henrik Linusson, Tuve Löfström, and Ulf Johansson. Accelerating difficulty estimation for conformal regression forests. *Annals of Mathematics and Artificial Intelligence*, 81(1-2):125–144, 2017.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and Regression Trees*. CRC press, 1984.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian Additive Regression Trees. *The Annals of Applied Statistics*, 4:266–298, 2010.

Gianmarco De Francisci Morales and Albert Bifet. SAMOA: Scalable Advanced Massive Online Analysis. *Journal of Machine Learning Research*, 16:149–153, 2015.

Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 71–80, 2000.

Jun Feng, Laura Lurati, Haojun Ouyang, Tracy Robinson, Yuanyuan Wang, Shenglan Yuan, and S. Stanley Young. Predictive Toxicology: Benchmarking Molecular Descriptors and Statistical Methods. *Journal of Chemical Information and Computer Sciences*, 43: 1463–1470, 2003.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.

Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19: 1–67, 1991.

João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 329–338, 2009.

Joao Gama. Evolving Data, Evolving Models in Economy and Finance. In *MIDAS '17: Second Workshop on Mining Data for Financial Applications*, 2017. Keynote.

Heitor M. Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfharinger, Geoff Holmes, and Talel Abdessalem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106:1469–1495, 2017.

Michael B. Greenwald and Sanjeev Khanna. *Quantiles and Equi-depth Histograms over Streams*, pages 45–86. Springer, 2016.

Elena Ikonomovska, João Gama, and Sašo Džeroski. Learning model trees from evolving data streams. *Data Mining and Knowledge Discovery*, 23:128–168, 2011.

Ulf Johansson, Henrik Boström, Tuve Löfström, and Henrik Linusson. Regression conformal prediction with random forests. *Machine Learning*, 97:155–176, 2014a.

Ulf Johansson, Cecilia Sönströd, Henrik Linusson, and Henrik Boström. Regression trees for streaming data with local performance guarantees. In *2014 IEEE International Conference on Big Data*, pages 461–470, 2014b.

Zohar Karnin, Kevin Lang, and Edo Liberty. Optimal quantile approximation in streams. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 71–78, Oct 2016.

Roger Koenker. *Quantile Regression*. Cambridge University Press, 1996.

Sotiris Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4): 261–283, 2013.

Balaji Lakshminarayanan, Daniel M. Roy, and Yee Whye Teh. Mondrian Forests for Large-Scale Regression when Uncertainty Matters. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 1478–1487, 2016.

Rowan McAllister, Yarin Gal, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4745–4753, 2017.

Nicolai Meinshausen. Quantile Regression Forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

Lucas Mentch and Giles Hooker. Quantifying Uncertainty in Random Forests via Confidence Intervals and Hypothesis Tests. *Journal of Machine Learning Research*, 17:1–41, 2016.

Nikunj C. Oza. Online bagging and boosting. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2340–2345, 2005.

Harris Papadopoulos and Haris Haralambous. Reliable prediction intervals with regression neural networks. *Neural Networks*, 24(8):842 – 851, 2011.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Proceedings of the 13th European Conference in Machine Learning*, pages 345–356, 2002.

Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, 40(1):815–840, 2011.

Binoy Ravindran, E. Douglas Jensen, and Peng Li. On recent advances in time/utility function real-time scheduling and resource management. In *8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*, pages 55–60, 2005.

Daniel M Roy and Yee W. Teh. The Mondrian Process. In *Advances in Neural Information Processing Systems 21*, pages 1377–1384, 2009.

Amir Saffari, Christian Leistner, Jakob Santner, Martin Godec, and Horst Bischof. Online random forests. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1393–1400, 2009.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*, 15:49–60, 2013.

Vladimir Vovk. On-line confidence machines are well-calibrated. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 187–196, 2002.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife. *Journal of Machine Learning Research*, 15: 1625–1651, 2014.

Lu Wang, Ge Luo, Ke Yi, and Graham Cormode. Quantiles over data streams: An experimental study. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 737–748, 2013.

Journal of Machine Learning Research 20 (2019) 1-8 Submitted 3/19; Revised 9/19; Published 10/19

SimpleDet: A Simple and Versatile Distributed Framework for Object Detection and Instance Recognition

Yuntao Chen CHENYUNTAO2016@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China

Chenxia Han CHENXIAHAN18@GMAIL.COM

Yanghao Li LYTTONHAO@GMAIL.COM

Zehao Huang ZEHAOHUANG18@GMAIL.COM
TuSimple, Beijing, China

Yi Jiang JIANGYI0425@GMAIL.COM

Naiyan Wang WINSTY@GMAIL.COM
TuSimple, Beijing, China

Zhaoxiang Zhang ZHAOXIANG.ZHANG@IA.AC.CN
Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China

Editor: Balazs Kegl

Abstract

Object detection and instance recognition play a central role in many AI applications like autonomous driving, video surveillance and medical image analysis. However, training object detection models on large scale datasets remains computationally expensive and time consuming. This paper presents an efficient and open source object detection framework called *SimpleDet* which enables the training of state-of-the-art detection models on consumer grade hardware at large scale. SimpleDet covers a wide range of models including both high-performance and high-speed ones. SimpleDet is well-optimized for both low precision training and distributed training and achieves 70% higher throughput for the Mask R-CNN detector compared with existing frameworks. Codes, examples and documents of SimpleDet can be found at `https://github.com/tusimple/simpledet`.

Keywords: Object Detection, Instance Recognition, Distributed Training, Mixed Precision Training

1. Introduction

During recent years, challenging datasets and sophisticated detection models have emerged with an ever-growing demand in high performance detection framework. The public available large-scale datasets grow exponentially in size. Open Images (Kuznetsova et al., 2018) contains 1700× images compared with PASCAL VOC (Everingham et al., 2010). In order to take full advantage of large-scale datasets, the state-of-the-art CNNs also scale up in companion with the datasets. For example, EfficientNet (Tan and Le, 2019) requires 50× FLOPs compared with AlexNet (Krizhevsky et al., 2012).

©2019 Chen Yuntao, Han Chenxia, Li Yanghao, Huang Zehao, Jiang Yi, Wang Naiyan, Zhang Zhaoxiang.
License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/19-205.html`.

These two factors bring the training time of a detector from tens of GPU hours up to tens of thousands of GPU hours, which calls for a distributed detection framework that scales out. Built on top of MXNet, SimpleDet is a high performance open source detection framework which provides an optimized training pipeline. Compared with existing frameworks, SimpleDet could achieve up to 70% higher mixed precision distributed training throughput and nearly linear scalability for the training of Mask R-CNN. Besides its high efficiency, SimpleDet also takes user experience as priority. SimpleDet features a modular design of detector and a configuration system in pure python, which could ease the use for users and also provides great extendibility. To further ease the adoption of SimpleDet, we provide pre-built Python wheels, Singularity images, and Docker containers for installation. The full codes, examples and documents of SimpleDet can be found at `https://github.com/tusimple/simpledet`.

2. Benchmark Platforms

We evaluate all frameworks on two platforms P1 and P2. P1 reflects the traditional GPGPU accelerators and P2 reflects the specialized accelerator (Tensor Core) built for Deep Learning. Detailed configuration could be found in `doc/BENCHMARK.md`

P1:

- 2X E5-2682 v4 + 8X 1080Ti + 25GbE
- CUDA 9.0 + cuDNN 7.5.0

P2:

- 2X Platinum 8163 + 8X 2080Ti + 25GbE
- CUDA 10.0 + cuDNN 7.5.0

3. Features of SimpleDet

In this section, we will introduce the model coverage, dataset support, scaling efficiency, memory saving techniques and other features for SimpleDet.

3.1. Models and Methods

SimpleDet covers both high performance (Dai et al., 2017; Zhu et al., 2019; Cai and Vasconcelos, 2018; Li et al., 2019; Ghiasi et al., 2019) and high speed detectors (Tan and Le, 2019; Redmon and Farhadi, 2018; Hinton et al., 2014), in addition to standard benchmark detectors (Ren et al., 2015; He et al., 2017; Lin et al., 2017). Furthermore, we provide an integrated knowledge distillation utility which could combine merits the high performance and high speed detectors for real-work applications.[1]

Standard Methods:

- Faster R-CNN
- Mask R-CNN
- RetinaNet

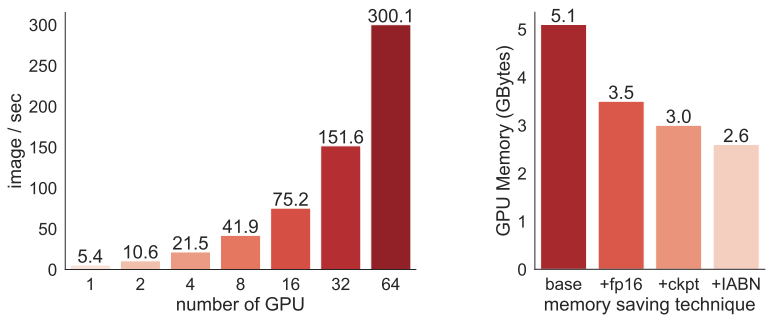
Performance-Oriented Methods:

- DCN v1/v2
- Cascade R-CNN
- TridentNet
- NASFPN

Speed-Oriented Methods:

- EfficientNet
- YOLOv3 (WIP)
- FCOS
- Knowledge Distill

1. Up to git commit 03b808, Sep 2019, we will keep this project up to date.



(a) (b)

Figure 1: (a) Scaling efficiency of SimpleDet. Throughput results are measured with a R50-FPN Mask R-CNN on the P1 platform. (b) GPU memory usages ablation with different memory saving techniques. Training memory usages with mixed precision training(+fp16), memory checkpoint(+ckpt), and in-place activation BN(+IABN) are reported. Results are obtained with a R50-C4 Faster R-CNN.

Besides a good coverage of models, SimpleDet also provides extensive pre-processing and post-processing routines in detection with their best practices, including various data augmentation techniques, multi-scale training and testing, soft NMS (Bodla et al., 2017) and weighted NMS, etc. All these features are provided based on the unified and versatile interfaces in SimpleDet, which allows the users to easily customize and extend these features in training.

3.2. Supported Datasets

SimpleDet provides generic parsing utilities for both COCO-style and VOC-style annotations. These tools allow users to utilize a wide range of existing annotations (Wang et al., 2019). A light-weight JSON format of annotation is also supported, which enables users to train on their own data. More details can be found in `doc/DATASET.md`. Currently supported datasets includes

1. Cityscapes
2. COCO
3. DeepLesion
4. DOTA
5. Kitchen
6. KITTI
7. VOC
8. WiderFace

3.3. Distributed Training

In order to take full advantage of ever-growing data, we need to utilize data parallelism to scale out to multiple machines. The efficiency of parameter communication lies in the core of scalable distributed training. Thanks to the underlying MXNet, SimpleDet supports

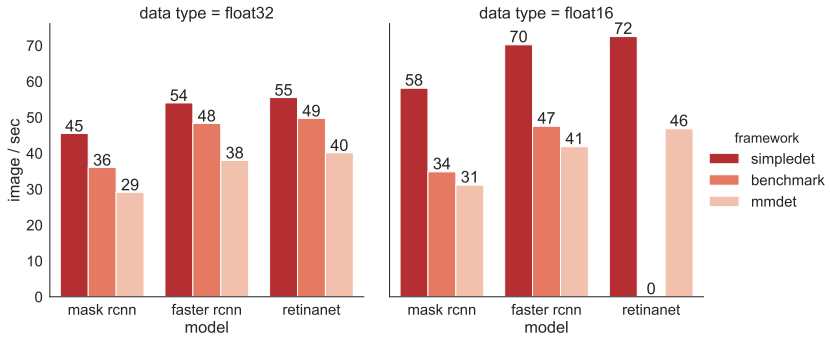


Figure 2: Compare with other detection frameworks on various detectors with different input data types. Here benchmark denotes maskrcnn-benchmark. Throughput results are obtained on the P2 platform with the R50-FPN backbone. RetinaNet with FP16 are not supported on maskrcnn-benchmark@24c8c9, so the result is 0.

both parameter server and all-reduce algorithms for model parameter update. Sophisticated communication algorithms like gradient aggregation and compression are also available out-of-the-box. Along with the mixed precision training technique which will be introduced in the next section, SimpleDet could give nearly linear scaling efficiency for a 8-node cluster as shown in Figure 1a. SimpleDet achieves 20% higher performance on the training of Mask R-CNN than MMDet with much lower end harwards (1080Ti vs V100, and 25Gb Ethernet vs 100Gb Infiniband).

3.4. Mixed Precision Training

Modern specialized hardwares like NVIDIA Tensor Core provide 10 times throughput for computation in half precision (FP16) over traditional GPGPUs. Besides speed up the training, low precision training also reduces the memory footprint and in turn saving the communication bandwidth. The main obstacle for the adoption in mixed precision training is the convergence issue and accuracy drop due to the limited range of representation. SimpleDet implements the scale loss proposed by Micikevicius et al. (2018) to mitigate these problems. In practice, we find that the mixed precision training yields identical training curves and detection mAP as the full precision one.

3.5. INT8 Training

Low-bit neural networks have been gaining more popularity in practice. The reduced storage usage, memory footprint and latency makes low-bit networks especially favorable. The traditional post-training quantization techniques work well in general scenarios but provide little flexibility for the corner cases. *SimpleDet* provides an integrated pseduo-int8 training pipeline, which enables the quantization of a wider range of models and provides more

flexibility for the users. In practice, we find that our minmax-based quantization method gives similar detection mAP as the full precision one.

3.6. Beyond Fixed Batch Normalization

Due to the limit of GPU memory, modern detectors are trained in a $1 \sim 2$ images per GPU setting. But BN is usually implemented in a per-GPU manner, which forces researchers to freeze BN statistics and parameters during training as a workaround. As indicated by Peng et al. (2018), fixed BN detectors trained with linear learning rate scaling scheme fail to converge when the total batch size increases beyond a threshold. This harms the scalability of a detection framework. In order to mitigate this limitation, SimpleDet integrates Cross-GPU Batch Normalization(CGBN) and Group Normalization. These normalization methods are available to users as a one-line configuration. In practice, we find that scaling a detector to a mini-batch size of 256 with CGBN or GN leads to stable convergence.

3.7. Memory Saving Techniques

A major limiting factor for the design of new detectors is the amount of memory available for a single GPU. Since the main training paradigm of CNN detector is data parallelism, designs are bounded by the amount of memory that a single GPU could provide. To mitigate this problem, SimpleDet combines mixed precision training, in-place activation batch normalization (Rota Bulò et al., 2018), computation graph merge, and layer-wise memory checkpointing (Chen et al., 2016) together to minimize the demand of GPU memory. Combining all these techniques, SimpleDet could save up to 50% memory as in Figure 1b with a marginal increase in computation cost compared with the vanilla setting.

4. Framework Design

Figure 3 gives a brief overview of the API design of SimpleDet. SimpleDet explicitly separates the implementation of the training and the test phases to reduce the code complexity. For example, `RpnHead` gives all proposals for test while only sampled proposals for training. Components are well-decoupled and only needs to process a small sets of input and output tensors. More details could be found in `doc/FRAMEWORK_OVERVIEW.md`.

5. Comparison with Other Detection Frameworks

We compare four other detection frameworks with SimpleDet in terms of training speed, supported models and advanced features in Table 1.

1. `detectron`[2] is the first general framework for object detection. But its training speed is a major problem as it uses Python operators in the core part of the framework extensively.

2. `mmdetection`(Chen et al., 2019) is a well-designed framework written in PyTorch which supports a wide range of detection models. Again, the training speed is a major problem of `mmdetection` as demonstrated in Figure 2.

2. https://github.com/facebookresearch/Detectron

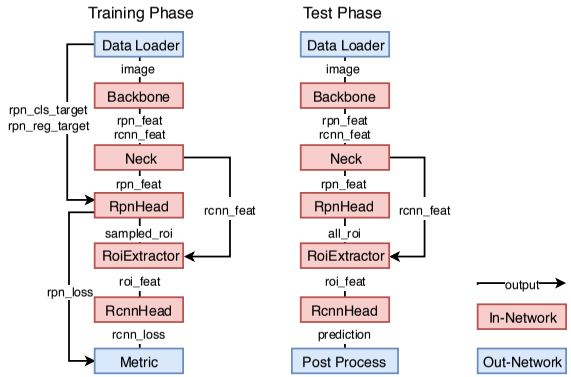


Figure 3: An overview of a generalized R-CNN detector in SimpleDet. The colored rectangles denotes components for a detector and the arrowed lines denote input and output tensors.

3. `tensorpack`[3] supports some advanced training features like Cross-GPU Batch Normalization and distributed training, but it lacks supports of some new models.

4. `maskrcnn-benchmark`[4] is a well optimized framework with amazing training speed. But it supports the least models of all frameworks. The apex-based FP16 support is also quite limited.

	`detectron`	`tensorpack`	`mmdetection`	`maskrcnn-benchmark`	`simpledet`
Commit ID	6efa99	cda5fd	d71184	24c8c9	2d8144
R50-FPN FRCNN Speed	29 images/s	29 images/s	38 images/s	48 images/s	54 images/s
FasterRCNN	✓	✓	✓	✓	✓
MaskRCNN	✓	✓	✓	✓	✓
RetinaNet	✓	✗	✓	✓	✓
Other Models	few	few	lots	few	some
Cross-GPU BN	✗	✓	✓	✗	✓
Mixed Precision Training	✗	✗	✓	✓	✓
Distributed Training	✗	✓	✓	✗	✓
Memory Checkpointing	✗	✗	✓	✗	✓

Table 1: Comparison of the training speed and supported features for `detectron`, `tensorpack`, `mmdetection`, `maskrcnn-benchmark` and `simpledet`.

3. https://github.com/tensorpack/tensorpack/tree/master/examples/FasterRCNN
4. https://github.com/facebookresearch/maskrcnn-benchmark

6. Conclusion

In this work, we present the SimpleDet framework for object detection and instance segmentation. SimpleDet features optimized mixed precision training and nearly linear scaling distributed training over 25Gb Ethernet, which achieves 70% higher throughput for Mask R-CNN compared with existing frameworks. It also integrates various memory-saving techniques to enable the training of large detectors. Besides, SimpleDet covers a wide range of detection models and datasets. We hope that this framework could help users design and benchmark new detection systems more efficiently.

References

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *International Conference on Computer Vision*, 2017.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv:1604.06174*, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *International Conference on Computer Vision*, 2017.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *International Conference on Computer Vision*, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems Workshop*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *International Conference on Computer Vision*, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, 2017.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. MegDet: A large mini-batch object detector. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of DNNs. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets v2: More deformable, better results. In *Conference on Computer Vision and Pattern Recognition*, 2019.

Journal of Machine Learning Research 20 (2019) 1-28 Submitted 4/19; Published 9/19

Simultaneous Phase Retrieval and Blind Deconvolution via Convex Programming

Ali Ahmed ALI.AHMED@ITU.EDU.PK
Department of Electrical Engineering
Information Technology University
Lahore, Pakistan

Alireza Aghasi AAGHASI@GSU.EDU
Department of Business Analytics
Georgia State University
Atlanta, GA, USA

Paul Hand P.HAND@NORTHEASTERN.EDU
Department of Mathematics and Khoury College of Computer Sciences
Northeastern University
Boston, MA, USA

Editor: Animashree Anandkumar

Abstract

We consider the task of recovering two real or complex m-vectors from phaseless Fourier measurements of their circular convolution. Our method is a novel convex relaxation that is based on a lifted matrix recovery formulation that allows a non-trivial convex relaxation of the bilinear measurements from convolution. We prove that if the two signals belong to known random subspaces of dimensions k and n, then they can be recovered up to the inherent scaling ambiguity with $m \gg (k+n)\log^2 m$ phaseless measurements. Our method provides the first theoretical recovery guarantee for this problem by a computationally efficient algorithm and does not require a solution estimate to be computed for initialization. Our proof is based on Rademacher complexity estimates. Additionally, we provide an alternating direction method of multipliers (ADMM) implementation and provide numerical experiments that verify the theory.

Keywords: Hyperbolic Constraints, Blind Deconvolution, Phase Retrieval, Convex Analysis, Rademacher Complexity

1. Introduction

This paper considers recovery of two unknown signals (real- or complex-valued) from the magnitude only measurements of their convolution. Let $\boldsymbol{w}$, and $\boldsymbol{x}$ be vectors residing in $\mathcal{H}^m$, where $\mathcal{H}$ denotes either $\mathbb{R}$, or $\mathbb{C}$. Moreover, denote by $\boldsymbol{F}$ the DFT matrix with entries $F[\omega,t] = \frac{1}{\sqrt{m}}\mathrm{e}^{-j2\pi\omega t/m},\ 1 \le \omega, t \le m$. We observe the phaseless Fourier coefficients of the circular convolution $\boldsymbol{w} \circledast \boldsymbol{x}$ of $\boldsymbol{w}$, and $\boldsymbol{x}$

$$\tilde{\boldsymbol{y}} = |\boldsymbol{F}(\boldsymbol{w} \circledast \boldsymbol{x})|, \tag{1}$$

where $|\boldsymbol{z}|$ returns the element wise absolute value of the vector $\boldsymbol{z}$. We use $\tilde{\boldsymbol{y}}$ to denote noiseless measurements, and reserve the notation $\boldsymbol{y}$ for more general noisy measurements.

©2019 Ali Ahmed and Alireza Aghasi and Paul Hand.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-332.html.

We are interested in recovering $\boldsymbol{w}$ and $\boldsymbol{x}$ from the phaseless measurements $\tilde{\boldsymbol{y}}$ or $\boldsymbol{y}$ of their circular convolution. In other words, the problem concerns blind deconvolution of two signals from phaseless measurements. The problem can also be viewed as identifying the structural properties on $\boldsymbol{w}$ such that its convolution with the signal/image of interest $\boldsymbol{x}$ makes the phase retrieval of a signal $\boldsymbol{x}$ well-posed. Since $\boldsymbol{w}$ and $\boldsymbol{x}$ are both unknown, and in addition, the measurements are phaseless, the inverse problem becomes severely ill-posed as many pairs of $\boldsymbol{w}$ and $\boldsymbol{x}$ correspond to the same $\boldsymbol{y}$. We show that this non-linear problem can be efficiently solved, under Gaussian measurements, using a semidefinite program and also theoretically prove this assertion. We also propose a heuristic approach to solve the proposed semidefinite program computationally efficiently. Numerical experiments show that, using this algorithm, one can successfully recover a blurred image from the magnitude only measurements of its Fourier spectrum.

Phase retrieval has been of continued interest in the fields of signal processing, imaging, physics, computational science, etc. Perhaps, the single most important context in which phase retrieval arises is the X-ray crystallography (Harrison, 1993; Millane, 1990), where the far-field pattern of X-rays scattered from a crystal form a Fourier transform of its image, and it is only possible to measure the intensities of the electromagnetic radiation. However, with the advancement of imaging technologies, the phase retrieval problem continues to arise in several other imaging modalities such as diffraction imaging (Bunk et al., 2007), microscopy (Miao et al., 2008), and astronomical imaging (Fienup and Dainty, 1987). In the imaging context, the result in this paper would mean that if rays are convolved with a *generic* pattern (either man made or naturally arising due to propagation of light through some unknown media) $\boldsymbol{w}$ prior to being scattered/reflected from the object, the image of the object can be recovered from the Fourier intensity measurements later on. As is well known from Fourier optics (Goodman, 2008), the convolution of a visible light with a generic pattern can be implemented using a lens-grating-lens setup.

Despite recent advances in theoretical understanding of phase retrieval (Candes et al., 2013, 2015b), the application to actual problems such as crystallography remains challenging owing partly to the simplistic mathematical models that may not fully capture the actual physical problem at hand. Our comparatively more complex model in (1) more elaborately encompasses structure in actual physical problems, for example, crystallography, where due to the natural periodic arrangement of a crystal structural unit, the observed electron density function of the crystal exactly takes the form (1); for details, see, Section 2 of (Elser et al., 2017).

Blind deconvolution is a fundamental problem in signal processing, communications, and in general system theory. Visible light communication has been proposed as a standard in 5G communications for local area networks (Azhar et al., 2013; Retamal et al., 2015; Azhar et al., 2010). Propagation of information carrying light through an unknown communication medium is modelled as a convolution. The channel is unknown and at the receiver it is generally difficult to measure the phase information in the propagated light. The result in this paper says that the transmitted signal can be blindly deconvolved from the unknown channel using the Fourier intensity measurements of the light only. The reader is referred to the first section of the supplementary note for a detailed description of the visible light communication and its connection to our formulation.

Main Contributions. In this paper, we study the combination of two important and notoriously challenging signal recovery problems: phase retrieval and blind deconvolution. We introduce a novel convex formulation that is possible because the algebraic structure from lifting resolves the bilinear ambiguity just enough to permit a non-trivial convex relaxation of the measurements. The strengths of our approach are that it allows a novel convex program that is the first to provably permit recovery guarantees with optimal sample complexity for the joint task of phase retrieval and blind deconvolution when the signals belong to known random subspaces. Additionally, unlike many recent convex relaxations and non-convex approaches, our approach does not require an initialization or estimate of the true solution in order to be stated or solved. While our convex formulation is presented in a lifted domain (with increased dimensionality), in implementing the convex problem, we have been able to use some recent results in Burer-Monteiro-type approaches and perform the optimization in a factored space (solving a series of non-convex programs which are guaranteed to land on the global minima).

Finally, an earlier version of this paper with only the exact recovery result form noiseless measurements appeared in (Ahmed et al., 2018) by the same authors. This paper extends the previous result to more general noisy measurements with a significantly modified proof. Moreover, the implementation in (Ahmed et al., 2018) was performed in a lifted domain and the proposed scheme required iterative projections onto the positive semidefinite cone, which was computationally prohibitive for large scale problems. By considering a different way of modelling the optimization problem, in Section 2 we present a more efficient algorithm, which is solved in a factored space using a Burer-Monteiro-type approach. This makes our implementation applicable to a much larger class of problems.

1.1. Observations in Matrix Form

The phase retrieval, and blind deconvolution problem has been extensively studied in signal processing community in recent years (Candes et al., 2015a; Ahmed et al., 2014) by lifting the unknown vectors to a higher dimensional matrix space formed by their outer products. The resulting rank-1 matrix is recovered using nuclear norm as a convex relaxation of the non-convex rank constraint. Recently, other forms of convex relaxations have been proposed (Bahmani and Romberg, 2017b; Goldstein and Studer, 2018; Aghasi et al., 2017a,b, 2018) that solve both the problems in the native (unlifted) space leading to computationally efficiently solvable convex programs. This paper handles the non-linear convolutional phase retrieval problem by lifting it into a bilinear problem. The resulting problem, though still non-convex, gives way to an effective convex relaxation that provably recovers $\boldsymbol{w}$ and $\boldsymbol{x}$ exactly.

We consider the problem of recovering $(\boldsymbol{w}^\natural, \boldsymbol{x}^\natural) \in \mathcal{H}^\ell \times \mathcal{H}^\ell$ from measurements of the form (1). It is clear that uniquely recovering $\boldsymbol{w}^\natural$ and $\boldsymbol{x}^\natural$, even up to the global bilinear ambiguity, is not possible without extra knowledge or information about the problem. We will address the problem under the broad and generally applicable structural assumptions that both $\boldsymbol{w}^\natural$ and $\boldsymbol{x}^\natural$ are members of known subspaces of $\mathcal{H}^m$. This means that $\boldsymbol{w}^\natural$ and $\boldsymbol{x}^\natural$ can be parametrized in terms of unknown lower dimensional vectors $\boldsymbol{h}^\natural \in \mathcal{H}^k$ and $\boldsymbol{m}^\natural \in \mathcal{H}^n$, respectively, as follows

$$\boldsymbol{w}^\natural = \boldsymbol{B}\boldsymbol{h}^\natural, \ \boldsymbol{x}^\natural = \boldsymbol{C}\boldsymbol{m}^\natural, \tag{2}$$

where $\boldsymbol{B} \in \mathcal{H}^{m\times k}$, and $\boldsymbol{C} \in \mathcal{H}^{m\times n}$ are known matrices whose columns span the subspaces in which $\boldsymbol{w}^\natural$ and $\boldsymbol{x}^\natural$ belong, respectively. Since the circular convolution operator diagonalizes in the Fourier domain, noiseless measurements become

$$\tilde{\boldsymbol{y}} = \tfrac{1}{\sqrt{m}}|\hat{\boldsymbol{B}}\boldsymbol{h}^\natural \odot \hat{\boldsymbol{C}}\boldsymbol{m}^\natural|, \tag{3}$$

where $\hat{\boldsymbol{B}} = \sqrt{m}\boldsymbol{F}\boldsymbol{B}$, $\hat{\boldsymbol{C}} = \sqrt{m}\boldsymbol{F}\boldsymbol{C}$, and $\odot$ represents the the Hadamard product. Denoting by $\boldsymbol{b}_\ell$ and $\boldsymbol{c}_\ell$ the rows of $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{C}}$, respectively, the entries of the noiseless measurements $\tilde{\boldsymbol{y}}$ can be expressed as

$$\tilde{y}_\ell^2 = \tfrac{1}{m}|\langle \boldsymbol{b}_\ell, \boldsymbol{h}^\natural\rangle\langle \boldsymbol{c}_\ell, \boldsymbol{m}^\natural\rangle|^2, \ \ell = 1\dots m.$$

This problem is non-linear in both unknowns; however, it reduces to a bilinear problem in the lifted variables $\boldsymbol{h}^\natural\boldsymbol{h}^{\natural *}$ and $\boldsymbol{m}^\natural\boldsymbol{m}^{\natural *}$, taking the form

$$\tilde{y}_\ell^2 = \tfrac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{h}^\natural\boldsymbol{h}^{\natural *}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{m}^\natural\boldsymbol{m}^{\natural *}\rangle = \tfrac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}^\natural\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}^\natural\rangle, \tag{4}$$

where $\boldsymbol{H}^\natural = \boldsymbol{h}^\natural\boldsymbol{h}^{\natural *}$ and $\boldsymbol{M}^\natural = \boldsymbol{m}^\natural\boldsymbol{m}^{\natural *}$. Treating the lifted variables $\boldsymbol{H}^\natural$ and $\boldsymbol{M}^\natural$ as unknowns makes the measurements bilinear in the unknowns; a structure that will help us formulate an effective convex relaxation.

In the case of noisy measurements, we will write without loss of generality that

$$\boldsymbol{y} = \tfrac{1}{\sqrt{m}}|\hat{\boldsymbol{B}}\boldsymbol{h}^\natural \odot \hat{\boldsymbol{C}}\boldsymbol{m}^\natural| \odot (1 + \boldsymbol{\xi}), \tag{5}$$

$$\xi_\ell \geq -1 \text{ for every } \ell = 1\dots m. \tag{6}$$

The noiseless case is given by $\boldsymbol{\xi} = 0$.

1.2. Novel Convex Relaxation

The task of recovering $\boldsymbol{H}^\natural$ and $\boldsymbol{M}^\natural$ from the noiseless measurements $\tilde{\boldsymbol{y}}$ in (5) can be naturally posed as an optimization program

$$\begin{aligned} \text{find} \quad & \boldsymbol{H}, \boldsymbol{M} \\ \text{subject to} \quad & \tfrac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle = \tilde{y}_\ell^2, \ \ell = 1\dots m. \\ & \operatorname{rank}(\boldsymbol{H}) = 1, \ \operatorname{rank}(\boldsymbol{M}) = 1. \end{aligned} \tag{7}$$

Both the measurement and the rank constraints are non-convex. Further, the immediate convex relaxation of each measurement constraint is trivial, as the convex hull of the set of $(\boldsymbol{H}, \boldsymbol{M})$ satisfying $\tilde{y}_\ell^2 = \frac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle$ is the set of all possible $(\boldsymbol{H}, \boldsymbol{M})$.

To derive our convex relaxation, recall that the true $\boldsymbol{H}^\natural = \boldsymbol{h}^\natural\boldsymbol{h}^{\natural *}$, and $\boldsymbol{M}^\natural = \boldsymbol{m}^\natural\boldsymbol{m}^{\natural *}$ are also positive semidefinite (PSD). This means that incorporating the PSD constraint in the optimization program translates into the fact that the variables $u_\ell = \langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle$ and $v_\ell = \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle$ are necessarily non-negative. That is,

$$\boldsymbol{H} \succcurlyeq \boldsymbol{0}, \text{ and } \boldsymbol{M} \succcurlyeq \boldsymbol{0} \implies u_\ell \geq 0, \text{ and } v_\ell \geq 0,$$

where the implication follows by the definition of PSD matrices. This observation restricts the hyperbolic constraint set in Figure 1 to the first quadrant only. For a fixed ℓ, we propose

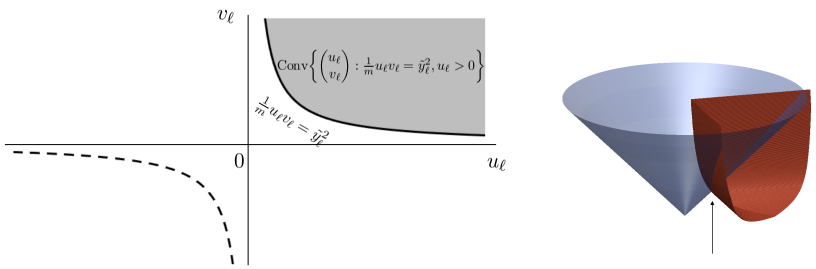


Figure 1: Left: The restriction of the hyperbolic constraint to the first quadrant; Right: Abstract illustration of the geometry of the convex relaxation. The PSD cone (blue) and the surface of the hyperbolic set (red) formed by two intersecting hyperbolas ($m = 2$). Evidently, there are multiple points on the surface and also in the convex hull of the hyperbolic set that lie on the PSD cone. The minimizer of the optimization program (8) picks the one with minimum trace that happens to lie at the intersection of hyperbolic ridge and the PSD cone (pointed out by an arrow).

replacing the non-convex hyperbolic set $\{(u_\ell, v_\ell) \in \mathbb{R}^2 \mid \frac{1}{m}u_\ell v_\ell = \tilde{y}_\ell^2, u_\ell \geq 0,\ v_\ell \geq 0\}$ with its convex hull $\{(u_\ell, v_\ell) \in \mathbb{R}^2 \mid \frac{1}{m}u_\ell v_\ell \geq \tilde{y}_\ell^2, u_\ell \geq 0,\ v_\ell \geq 0\}$. In short, our convex relaxation is possible because the PSD constraint from lifting happens to select a specific branch of the hyperbola given by any particular bilinear measurement, and this single branch has a non-trivial convex hull.

The rest of the convex relaxation is standard, as the rank constraint in (7) is then relaxed with a nuclear-norm minimization, which reduces to trace minimization in the PSD case:

$$\begin{aligned} &\text{minimize } \ \mathrm{Tr}(\boldsymbol{H}) + \mathrm{Tr}(\boldsymbol{M}) \\ &\text{subject to } \ \tfrac{1}{m}\langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle \langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle \geq \tilde{y}_\ell^2,\ \ell = 1 \dots m \\ &\qquad\qquad \boldsymbol{H} \succcurlyeq \boldsymbol{0},\ \boldsymbol{M} \succcurlyeq \boldsymbol{0}. \end{aligned}$$

In the noiseless or noisy cases, we will study the following program, which only differs in that the noiseless observations are substituted by the possibly noisy ones given from (5):

$$\begin{aligned} &\text{minimize } \ \mathrm{Tr}(\boldsymbol{H}) + \mathrm{Tr}(\boldsymbol{M}) \\ &\text{subject to } \ \tfrac{1}{m}\langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle \langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle \geq y_\ell^2,\ \ell = 1 \dots m \\ &\qquad\qquad \boldsymbol{H} \succcurlyeq \boldsymbol{0},\ \boldsymbol{M} \succcurlyeq \boldsymbol{0}. \end{aligned} \tag{8}$$

The convexity of the optimization program above is established in the lemma below. A formal proof of he lemma can be found in Appendix A.

Lemma 1 *The optimization problem* (8) *is a convex program.*

1.3. Main Results

We consider the case of i.i.d. Gaussian measurements,

$$\boldsymbol{b}_\ell \sim \mathrm{Normal}(0, \tfrac{1}{m}\boldsymbol{I}_k), \quad \boldsymbol{c}_\ell \sim \mathrm{Normal}(0, \tfrac{1}{m}\boldsymbol{I}_n), \quad \ell = 1, \dots m. \tag{9}$$

We show that with this choice, (8) recovers a global scaling $(\alpha \boldsymbol{H}^\natural, \alpha^{-1}\boldsymbol{M}^\natural)$ of $(\boldsymbol{H}^\natural, \boldsymbol{M}^\natural)$ The exact value of the unknown scalar multiple α can be characterized for the solution of (8). Observe that the solution $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ of the convex optimization program in (8) obeys $\mathrm{Tr}(\widehat{\boldsymbol{H}}) = \mathrm{Tr}(\widehat{\boldsymbol{M}})$. We aim to show that the solution of the optimization program recovers the following scaling $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ of the true solution $(\boldsymbol{H}^\natural, \boldsymbol{M}^\natural)$:

$$\tilde{\boldsymbol{H}} = \sqrt{\frac{\mathrm{Tr}(\boldsymbol{M}^\natural)}{\mathrm{Tr}(\boldsymbol{H}^\natural)}}\boldsymbol{H}^\natural,\ \tilde{\boldsymbol{M}} = \sqrt{\frac{\mathrm{Tr}(\boldsymbol{H}^\natural)}{\mathrm{Tr}(\boldsymbol{M}^\natural)}}\boldsymbol{M}^\natural. \tag{10}$$

It is worth noting that $\mathrm{Tr}(\tilde{\boldsymbol{H}}) = \mathrm{Tr}(\tilde{\boldsymbol{M}})$, $\tilde{\boldsymbol{H}} = \tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^*$, and $\tilde{\boldsymbol{M}} = \tilde{\boldsymbol{m}}\tilde{\boldsymbol{m}}^*$.

We show that if $\boldsymbol{B}$ and $\boldsymbol{C}$ are random, and m is sufficiently large with respect to $k + n$, then the convex program (8) stably recovers the true solution $(\boldsymbol{H}^\natural, \boldsymbol{M}^\natural)$ up to the global bilinear scaling, with high probability.

Theorem 2 (Stable Recovery) *Given the magnitude only Fourier measurements* (5) *of the convolution of two unknown vectors* $\boldsymbol{w}^\natural$, *and* $\boldsymbol{x}^\natural$ *in* $\mathcal{H}^m$ *contaminated with additive noise* $\boldsymbol{\xi}$ *in* $\mathbb{R}^m$. *Suppose that* $\boldsymbol{w}^\natural$, *and* $\boldsymbol{x}^\natural$ *are generated as in* (2), *where* $\boldsymbol{B}$, *and* $\boldsymbol{C}$ *are known standard Gaussian matrices as in* (9). *Assume without loss of generality that noise components* $\xi_\ell \geq -1$ *for every* $\ell = 1, 2, 3, \dots, m$. *Then for any* $t > 0$, *when* $m \geq c(\sqrt{(k+n)}\log m + t)^2$, *with probability at least* $1 - \exp(-\frac{1}{2}mt^2)$, *the solution* $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ *of the convex optimization program in* (8) *obeys*

$$\|\widehat{\boldsymbol{H}} - \alpha\boldsymbol{H}^\natural\|_F^2 + \|\widehat{\boldsymbol{M}} - \alpha^{-1}\boldsymbol{M}^\natural\|_F^2 \leq 44^2\|\boldsymbol{\xi}\|_\infty(\|\tilde{\boldsymbol{H}}\|_F^2 + \|\tilde{\boldsymbol{M}}\|_F^2),$$

where $\alpha = \sqrt{\frac{\mathrm{Tr}(\boldsymbol{M}^\natural)}{\mathrm{Tr}(\boldsymbol{H}^\natural)}}$, *and* c *is an absolute constant.*

As a straightforward special case, for noiseless measurements, solving the proposed convex program would identify the true signals exactly, up to the global bilinear ambiguity, with high probability.

Corollary 3 (Exact Recovery) *Consider the magnitude-only Fourier measurements in* (3) *and a similar setting as Theorem 2. Fixing* $t > 0$, *the convex optimization in* (8) *uniquely recovers* $(\alpha\boldsymbol{H}^\natural, \alpha^{-1}\boldsymbol{M}^\natural)$ *for* $\alpha = \sqrt{\frac{\mathrm{Tr}\,\boldsymbol{M}^\natural}{\mathrm{Tr}\,\boldsymbol{H}^\natural}}$ *with probability at least* $1 - \exp(-\frac{1}{2}mt^2)$ *whenever* $m \geq c(\sqrt{(k+n)}\log m + t)^2$, *where* c *is an absolute constant.*

Both Theorem 2 and Corollary 3 establish high probability recovery for phaseless blinear inversion within random subspaces, provided that m on the order of $(k + n)$. Except for log factors, this sample complexity is optimal. Proof for the theorem is in the appendix and is based on Rademacher complexity estimates of descent directions objective.

2. Implementing the Convex Program

A conference paper by the authors (Ahmed et al., 2018) presented an ADMM scheme to address the central convex program (8). One of the main computational challenges with

that proposed scheme is that it uses a projection onto the positive semi-definite cone at every ADMM iteration. Such an operation makes the algorithm prohibitively expensive for large problem sizes. In this section, we consider an alternative ADMM scheme which uses a Burer-Monteiro low-rank factorization (Burer and Monteiro, 2003, 2005; Bhojanapalli et al., 2018) to bypasses the PSD projection and speed up the algorithm convergence[1].

To proceed, consider our central convex program

$$\begin{aligned} \underset{\boldsymbol{X}_1,\boldsymbol{X}_2}{\text{minimize}} \;\; & \mathrm{Tr}(\boldsymbol{X}_1)+\mathrm{Tr}(\boldsymbol{X}_2) \\ \text{subject to} \;\; & \langle \boldsymbol{a}_{1,\ell}\boldsymbol{a}_{1,\ell}^*, \boldsymbol{X}_1\rangle \langle \boldsymbol{a}_{2,\ell}\boldsymbol{a}_{2,\ell}^*, \boldsymbol{X}_2\rangle \geq \delta_\ell \geq 0, \quad \ell = 1\dots m \\ & \boldsymbol{X}_1 \succcurlyeq \boldsymbol{0},\; \boldsymbol{X}_2 \succcurlyeq \boldsymbol{0}. \end{aligned} \tag{11}$$

Note that complex-valued positive semidefinite matrices are necessarily Hermitian. For a simpler notation, we define the convex set

$$\mathcal{C} = \{(\boldsymbol{u},\boldsymbol{v}) \in \mathbb{R}^m \times \mathbb{R}^m : u_\ell v_\ell \geq \delta_\ell > 0, u_\ell \geq 0\}. \tag{12}$$

An alternative way of formulating program (11) is

$$\begin{aligned} \underset{\{\boldsymbol{X}_j,\boldsymbol{u}_j\}_{j=1,2}}{\text{minimize}} \quad & \mathbb{I}_{\mathcal{C}}(\boldsymbol{u}_1,\boldsymbol{u}_2) + \sum_{j=1}^{2} \mathrm{Tr}(\boldsymbol{X}_j) + \mathbb{I}_+(\boldsymbol{X}_j) \\ \text{subject to} \quad & u_{j,\ell} = \langle \boldsymbol{a}_{j,\ell}\boldsymbol{a}_{j,\ell}^*, \boldsymbol{X}_j\rangle,\; \ell = 1\dots m,\; j=1,2, \end{aligned} \tag{13}$$

where

$$\mathbb{I}_{\mathcal{C}}(\boldsymbol{u},\boldsymbol{v}) = \begin{cases} 0 & (\boldsymbol{u},\boldsymbol{v}) \in \mathcal{C} \\ +\infty & (\boldsymbol{u},\boldsymbol{v}) \notin \mathcal{C} \end{cases}, \quad \mathbb{I}_+(\boldsymbol{X}) = \begin{cases} 0 & \boldsymbol{X} \succeq \boldsymbol{0} \\ +\infty & \boldsymbol{X} \not\succeq \boldsymbol{0} \end{cases}.$$

Defining the dual vectors $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{R}^m$, the augmented Lagrangian for (13) takes the form

$$\begin{aligned} \mathcal{L}\left(\{\boldsymbol{X}_j,\boldsymbol{u}_j,\boldsymbol{\alpha}_j\}_{j=1,2}\right) = \mathbb{I}_{\mathcal{C}}(\boldsymbol{u}_1,\boldsymbol{u}_2) + \sum_{j=1}^{2} \mathrm{Tr}(\boldsymbol{X}_j) + \mathbb{I}_+(\boldsymbol{X}_j) \\ + \frac{\rho}{2}\sum_{j=1}^{2}\sum_{\ell=1}^{m} \left(u_{j,\ell} - \langle \boldsymbol{a}_{j,\ell}\boldsymbol{a}_{j,\ell}^*, \boldsymbol{X}_j\rangle + \alpha_{j,\ell}\right)^2. \end{aligned} \tag{14}$$

To set up an ADMM scheme, each variable update at the k-th iteration is performed by minimizing $\mathcal{L}$ with respect to that variable while fixing the others. More specifically, using the superscript (k) to denote the iteration, for $j=1,2$ we have the primal updates

$$\boldsymbol{X}_j^{(k+1)} = \underset{\boldsymbol{X}_j \succeq \boldsymbol{0}}{\arg\min} \quad \mathrm{Tr}(\boldsymbol{X}_j) + \frac{\rho}{2}\sum_{\ell=1}^{m}\left(\langle \boldsymbol{a}_{j,\ell}\boldsymbol{a}_{j,\ell}^*, \boldsymbol{X}_j\rangle - u_{j,\ell}^{(k)} - \alpha_{j,\ell}^{(k)}\right)^2, \tag{15}$$

$$\left(\boldsymbol{u}_1^{(k+1)}, \boldsymbol{u}_2^{(k+1)}\right) = \underset{(\boldsymbol{u}_1,\boldsymbol{u}_2)\in\mathcal{C}}{\arg\min} \quad \frac{1}{2}\sum_{j=1}^{2}\sum_{\ell=1}^{m}\left(u_{j,\ell} - \left\langle \boldsymbol{a}_{j,\ell}\boldsymbol{a}_{j,\ell}^*, \boldsymbol{X}_j^{(k+1)}\right\rangle + \alpha_{j,\ell}^{(k)}\right)^2, \tag{16}$$

1. An implementation of our solver is publicly available at: `https://github.com/branchhull/BDPR`

along with the dual updates

$$\alpha_{j,\ell}^{(k+1)} = \alpha_{j,\ell}^{(k)} + u_{j,\ell}^{(k+1)} - \left\langle \boldsymbol{a}_{j,\ell}\boldsymbol{a}_{j,\ell}^*, \boldsymbol{X}_j^{(k+1)}\right\rangle.$$

In the sequel we outline a computational procedure for each step of the proposed ADMM scheme.

2.1. Performing the $\boldsymbol{X}$-update

Central to the ADMM step (15), in this section we focus on addressing the convex program

$$\underset{\boldsymbol{X}\succeq\boldsymbol{0}}{\text{minimize}} \quad \mathrm{Tr}(\boldsymbol{X}) + \frac{\rho}{2}\sum_{\ell=1}^{m}(\boldsymbol{a}_\ell^*\boldsymbol{X}\boldsymbol{a}_\ell - \theta_\ell)^2. \tag{17}$$

One of the most successful heuristics to address (17), which was brought into attention by Burer and Monteiro (2003), is to consider the PSD factorization $\boldsymbol{X} = \boldsymbol{V}\boldsymbol{V}^*$ and to address the non-convex program

$$\underset{\boldsymbol{V}}{\text{minimize}} \quad \|\boldsymbol{V}\|_F^2 + \frac{\rho}{2}\sum_{\ell=1}^{m}\left(\|\boldsymbol{V}^*\boldsymbol{a}_\ell\|_2^2 - \theta_\ell\right)^2. \tag{18}$$

For a large class of objectives, there have been theoretical arguments that local minimizers to (18) can form the global minimizer to (17). Specifically, for the objective form (17), Bhojanapalli et al. (2018) have recently shown that for almost all objectives of this form, if $\tilde{\boldsymbol{V}} \in \mathbb{R}^{n\times r}$ is a second-order stationary solution to (18) and $r(r+1) > 2m$, then $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{V}}\tilde{\boldsymbol{V}}^*$ is a global minimizer to (17) (see Corollary 2 in the aforementioned reference).

Finding solutions to (18) can be performed via standard optimization toolboxes. In particular, we use quasi-Newton methods with cubic line search as implemented in Schmidt (2005), which only need the gradient of the objective in (18), calculated as

$$2\boldsymbol{V} + \rho\sum_{\ell=1}^{m}\left(\|\boldsymbol{V}^*\boldsymbol{a}_\ell\|_2^2 - \theta_\ell\right)\boldsymbol{a}_\ell\boldsymbol{a}_\ell^*\boldsymbol{V}.$$

It is noteworthy that the gradient calculation only requires a series of matrix-vector multiplications.

With the proposed computational scheme, to update $\boldsymbol{X}$ at each ADMM iteration, another iterative scheme needs to be carried out to solve (18). Despite the nested nature of this framework, a very good initialization for $\boldsymbol{V}$ at the start of each ADMM update is the optimal $\boldsymbol{V}$ from the previous ADMM step. Aside from the factorization technique, such choice of initialization further contributes to fast solutions of (17).

2.2. Performing the $\boldsymbol{u}$-update

The $\boldsymbol{u}$-update in (16) is a standard projection problem onto the set $\mathcal{C}$. It is straightforward to see that program

$$\min_{(\boldsymbol{u}_1,\boldsymbol{u}_2)\in\mathcal{C}} \; \frac{1}{2}\sum_{j=1}^{2}\sum_{\ell=1}^{m}(u_{j,\ell} - \theta_{j,\ell})^2 \tag{19}$$

decouples into m distinct programs of the form

$$\min_{u_1,u_2} \frac{1}{2}\sum_{j=1}^{2}(u_j-\theta_j)^2 \quad \text{subject to:} \quad u_1u_2 \geq \delta > 0,\ u_1 \geq 0. \tag{20}$$

In the sequel we focus on addressing (20), as solving (20) for each component ℓ would deliver the solution to (19). We proceed by forming the Lagrangian for the constrained problem (20)

$$l(u_1,u_2,\mu_1,\mu_2) = \frac{1}{2}\left\|\begin{pmatrix}u_1\\u_2\end{pmatrix}-\begin{pmatrix}\theta_1\\\theta_2\end{pmatrix}\right\|_2^2 + \mu_1(\delta-u_1u_2) - \mu_2u_1.$$

Along with the primal constraints, the Karush-Kuhn-Tucker optimality conditions are

$$\frac{\partial l}{\partial u_1} = u_1-\theta_1-\mu_1u_2-\mu_2 = 0, \tag{21}$$

$$\frac{\partial l}{\partial u_2} = u_2-\theta_2-\mu_1u_1 = 0, \tag{22}$$

$$\mu_1 \geq 0, \quad \mu_1(\delta-u_1u_2) = 0,$$

$$\mu_2 \geq 0, \quad \mu_2u_1 = 0.$$

We now proceed with the possible cases.

Case 1. $\mu_1=\mu_2=0$:
In this case we have $(u_1,u_2)=(\theta_1,\theta_2)$ and this result would only be acceptable when $u_1u_2\geq\delta$ and $u_1\geq 0$.

Case 2. $\mu_1=0$, $u_1=0$:
In this case the first feasibility constraint of (20) requires that $\delta\leq 0$, which is not a possibility

Case 3. $\delta-u_1u_2=0$, $u_1=0$:
Similar to the previous case, this cannot happen when $\delta>0$.

Case 4. $\mu_2=0$, $\delta-u_1u_2=0$:
In this case we have $\delta=u_1u_2$, combining which with (22) yields $\delta=(\theta_2+\mu_1u_1)u_1$, or

$$\mu_1 = \frac{\delta-\theta_2u_1}{u_1^2}. \tag{23}$$

Similarly, (21) yields

$$u_1 = \theta_1+\mu_1(\theta_2+\mu_1u_1). \tag{24}$$

Since the condition $\delta=u_1u_2$ requires that $u_1>0$, μ_1 can be eliminated between (23) and (24) to generate the following forth order polynomial equation in terms of u_1:

$$u_1^4-\theta_1u_1^3+\delta\theta_2u_1-\delta^2=0.$$

After solving this fourth order polynomial equation, we pick the real root u_1 which obeys

$$u_1\geq 0, \qquad \delta-\theta_2u_1\geq 0. \tag{25}$$

Note that the second inequality in (25) warrants non-negative values for μ_1 thanks to (23). After picking the right root, we can explicitly obtain μ_1 using (24) and calculate the u_2 using (22). The resulting (u_1,u_2) pair presents the solution to (20), and finding such pair for every ℓ provides the solution to (19). Thanks to the decoupling of the projection step in ℓ, the $\boldsymbol{u}$-update can enjoy a parallel computing framework.

3. Experiments and Application

We now present numerical experiments that verify the recovery guarantee for bilinear inversion from phaseless Fourier measurements by program (8). We consider the noiseless case with i.i.d. Gaussian matrices $\boldsymbol{B}$ and $\boldsymbol{C}$. In Figure 2 we present the phase portrait associated with the proposed convex framework. To obtain the diagram on the left panel, for each fixed value of m, we run the algorithm for 100 different combinations of n and k, each time using independently generated Gaussian matrices $\boldsymbol{B}$ and $\boldsymbol{C}$. If the algorithm converges to a sufficiently close neighbourhood of the ground-truth solution (a relative error of less than 1% with respect to the ℓ_2 norm), we label the experiment as successful. Figure 2 shows the collected success frequencies, where solid black corresponds to 100% success and solid white corresponds to 0% success. For an empirically selected constant c, the success region almost perfectly stands on the left side of the line $n+k=cm\log^{-2}m$. The results indicate that the constants in the Theorem are not unreasonably large in practice.

While the analysis in this paper is specifically focused on the Gaussian subspace embeddings for $\boldsymbol{w}$ and $\boldsymbol{x}$, we additionally consider the case where $\boldsymbol{B}$ is deterministic and $\boldsymbol{C}$ is Gaussian. Specifically $\boldsymbol{B}$ will be an equispaced sampling of the columns of the identity matrix. On the right panel of Figure 2, we have plotted the phase diagram for this case of deterministic $\boldsymbol{B}$ and random $\boldsymbol{C}$. These results hint that the convex framework is applicable to more realistic deterministic subspace models.

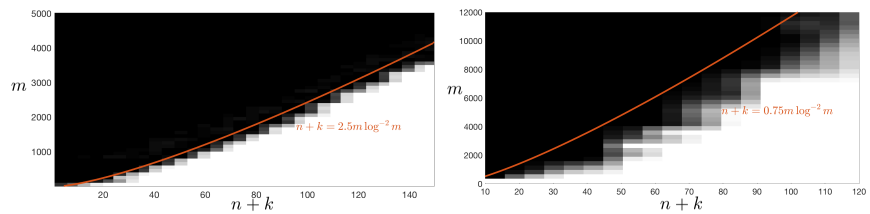


Figure 2: Phase portraits highlighting the frequency of successful recoveries of the proposed convex program for random and deterministic channel subspaces (see the text for the experiment details).

We do not want to give the reader the impression that the present paper solves the problem of blind deconvolutional phase retrieval in practice. The numerical experiments we perform do indeed show excellent agreement with the theorem in the case of random subspaces. Such subspaces are unlikely to appear in practice, and typically appropriate subspaces would both be deterministic, including partial Discrete Cosine Transforms or partial Discrete Wavelet Transforms. Numerical experiments, not shown, indicate that our convex relaxation is less effective for the cases of these doubly deterministic subspaces. We suspect this is due to the fact that the subspaces for both measurements should be mutually incoherent, in addition to both being incoherent with respect to the Fourier basis given by the measurements. As with the initial recovery theory for the problems of compressed sensing and phase retrieval, we have studied the random case in order to show information theoretically optimal sample complexity is possible by efficient algorithms. Based on this work, it is clear that blind deconvolutional phase retrieval is still a very challenging problem in the presence

of deterministic matrices, and one for which development of convex or non-convex methods may provide substantial progress in applications. In fact, we are still far from phase retrieval, and blind deconvolution problems individually in their natural and practical settings. The formulation (1) is general, and the subspace assumptions on $\boldsymbol{w}$ and $\boldsymbol{x}$ only cover some applied scenarios. There are some recent articles (Kuo et al., 2019; Zhang et al., 2018) that, though study the blind deconvolution or phase retrieval individually, make some advances in going beyond the subspace model to arguably more realistic signal models in real applications.

3.1. Related Real-World Applications

As discussed earlier, the proposed framework addresses a general version of the phase retrieval, where as a result of the light propagation through a medium, the rays are convolved with an unknown kernel. Aside from this general setup, in this section we will point out two specific physical problems, solving which requires simultaneously addressing variants of the phase retrieval and blind deconvolution problems.

3.1.1. Stylized Application in Visible Light Communications

As discussed in the body of the paper, an important application domain where blind deconvolution from phaseless Fourier measurements arises is the visible light communication (VLC). A stylized VLC setup is shown in Figure 3. A message $\boldsymbol{m} \in \mathbb{R}^n$ is to be transmitted using visible light. The message is first coded by multiplying it with a tall coding matrix $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ and the resultant information $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{m}$ is modulated on a light wave. The light wave propagates through an unknown media. This propagation can be modelled as a convolution $\boldsymbol{x} \circledast \boldsymbol{w}$ of the information signal $\boldsymbol{x}$ with unknown channel $\boldsymbol{w} \in \mathbb{R}^m$. The vector $\boldsymbol{w}$ contains channel taps, and frequently in realistic applications has only few significant taps. In this case, one can model

$$\boldsymbol{w} \approx \boldsymbol{B}\boldsymbol{h},$$

where $\boldsymbol{h} \in \mathbb{R}^k$ is a short ($k \ll m$) vector, and $\boldsymbol{B} \in \mathbb{R}^{m\times k}$ in this case is a subset of the columns of an identity matrix. Generally, the multipath channels are well modelled with non-zero taps in top locations of $\boldsymbol{w}$. In that case, $\boldsymbol{B}$ is exactly known to be the top few columns of the identity matrix.

In visible light communication, there is always a difficulty associated with measuring phase information in the received light. Figure 3 shows a setup, where we measure the phaseless Fourier transform (light through the lens) of this signal. The measurements are therefore

$$\tilde{\boldsymbol{y}} = |\boldsymbol{F}(\boldsymbol{B}\boldsymbol{h} \circledast \boldsymbol{C}\boldsymbol{m})|,$$

and one wants to recover $\boldsymbol{m}$, and $\boldsymbol{h}$ given the knowledge of $\boldsymbol{B}$ and the coding matrix $\boldsymbol{C}$. Since we chose $\boldsymbol{C}$ to be random Gaussian, and $\boldsymbol{B}$ is the columns of identity. As mentioned at the end of the numerics section that with this subspace model, we obtain similar recovery results as one would have for both $\boldsymbol{B}$, and $\boldsymbol{C}$ being random Gaussian. The proposed convex program solves this difficult inverse problem and recovers the true solution with these subspace models.

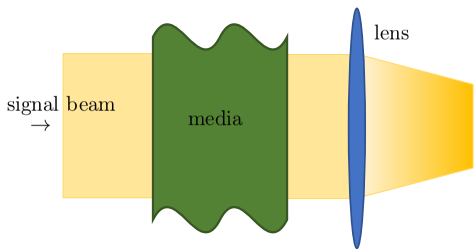


Figure 3: Visible light communication optical setup; the media block normally consists of phosphor, filter and a linear polarizer. The lens takes the Fourier transform of the light and one can only measure the intensity only measurements of this transformed light source signal.

3.1.2. Crystallography

In crystallography, the lattice structural information is carried in the electron density function of the crystal, which may be represented as

$$\rho(\boldsymbol{x}) = \sum_{\boldsymbol{y}\in S} \rho_c(\boldsymbol{x}-\boldsymbol{y}). \tag{26}$$

Here, $\rho_c(\boldsymbol{x})$ is a compactly supported central motif, and S is a finite, but large compact set of translation vectors. In a sense, the electron density function is the result of convolving the central motif with the indicator of the set S.

Denoting the Fourier transforms of $\rho(\boldsymbol{x})$ and $\rho_c(\boldsymbol{x})$ by $\hat{\rho}(\boldsymbol{\omega})$ and $\hat{\rho}_c(\boldsymbol{\omega})$, similar to the other phase retrieval problems, X-ray experiments measure the magnitude of the Fourier transform of $\hat{\rho}(\boldsymbol{\omega})$, which can be written as

$$\hat{\rho}(\boldsymbol{\omega}) = \sum_{\boldsymbol{y}\in S} \exp\left(-i2\pi\langle \boldsymbol{x}, \boldsymbol{\omega}\rangle\right) \hat{\rho}_c(\boldsymbol{\omega}).$$

Identifying the motif $\hat{\rho}_c(\boldsymbol{\omega})$ and the set S, using measurements of the form $|\hat{\rho}_c(\boldsymbol{\omega})|^2$ would be a problem which involves simultaneously addressing a phase retrieval and blind deconvolution problem. The reader is referred to (Elser et al., 2017) and the references therein for more details of the underlying physics and measurement system.

4. Proof of Theorem 2

As shown in Appendix A, the hyperbolic feasible set $\{(\boldsymbol{H},\boldsymbol{M}) : \frac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle \geq y_\ell^2\}$ is convex in $(\boldsymbol{H},\boldsymbol{M})$, however, the corresponding constraint function[2] $f_\ell^\circ(\boldsymbol{H},\boldsymbol{M}) = y_\ell^2 - \frac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle$ is a non-convex function of $(\boldsymbol{H},\boldsymbol{M})$. In the analysis later, it is easier to work with convex constraint functions instead; therefore, we replace the function $f_\ell^\circ(\boldsymbol{H},\boldsymbol{M})$

2. We will abuse the notation by specifying the same using (u_ℓ, v_ℓ) as parameters, i.e., $f^\circ(u_\ell, v_\ell) = y_\ell^2 - \frac{1}{m}u_\ell v_\ell$, where recall that $u_\ell = \langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle$, and $v_\ell = \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle$

above with a convex counterpart whose 0-level-set is the same under the additional constraints that $\boldsymbol{H} \succcurlyeq 0$ and $\boldsymbol{M} \succcurlyeq 0$:

$$f_\ell(\boldsymbol{H},\boldsymbol{M}) := \gamma_\ell(\boldsymbol{H},\boldsymbol{M})\left(\sqrt{4y_\ell^2 + \frac{1}{m}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle - \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle)^2} - \frac{1}{\sqrt{m}}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle + \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle)\right), \quad (27)$$

where

$$\gamma_\ell(\boldsymbol{H},\boldsymbol{M}) := \begin{cases} \min\left\{1, \frac{\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\tilde{\boldsymbol{H}}\rangle + \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\tilde{\boldsymbol{M}}\rangle}{2}\right\}, & \frac{\sqrt{4y_\ell^2 + \frac{1}{m}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle - \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle)^2}}{\frac{1}{\sqrt{m}}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle + \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle} \le 1 \\ \frac{\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\tilde{\boldsymbol{H}}\rangle + \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\tilde{\boldsymbol{M}}\rangle}{2}, & \text{otherwise.} \end{cases} \quad (28)$$

is a scalar chosen to normalize the gradients computed below. Recall that $y_\ell^2 = \tilde{y}_\ell^2(1+\xi_\ell)$. It is now easy to check that feasible sets drawn by the convex and non-convex functions are equal under the additional constraint of $u_\ell = \langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle \ge 0$, and $v_\ell = \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle \ge 0$ for every $\boldsymbol{H}$, and $\boldsymbol{M}$, i.e., $\boldsymbol{H} \succcurlyeq \boldsymbol{0}$, and $\boldsymbol{M} \succcurlyeq \boldsymbol{0}$, respectively. Mathematically,

$$\left\{(u_\ell,v_\ell) \mid f_\ell^\circ(u_\ell,v_\ell) := y_\ell^2 - \tfrac{1}{m}u_\ell v_\ell \le 0, u_\ell \ge 0,\ v_\ell \ge 0\right\} = \left\{(u_\ell,v_\ell) \mid f_\ell(u_\ell,v_\ell) := \gamma_\ell\left(\sqrt{4y_\ell^2 + \tfrac{1}{m}(u_\ell - v_\ell)^2} - \tfrac{1}{\sqrt{m}}(u_\ell + v_\ell)\right) \le 0\right\},$$

for any $\gamma_\ell > 0$. Note that $f_\ell(u_\ell,v_\ell) \le 0$ automatically constrains $u_\ell \ge 0$ and $v_\ell \ge 0$. It is easy to check that $f_\ell(u_\ell,v_\ell)$ is a convex function. Since $\boldsymbol{H} \succcurlyeq 0$ and $\boldsymbol{M} \succcurlyeq \boldsymbol{0}$ imply that $u_\ell \ge 0$, and $v_\ell \ge 0$, respectively, and since $\gamma_\ell(\boldsymbol{H},\boldsymbol{M}) \ge 0$, we can write the above conclusion in the matrix space as

$$\left\{(\boldsymbol{H},\boldsymbol{M}) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{n\times n} \mid y_\ell^2 \le \tfrac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle, \boldsymbol{H} \succcurlyeq 0,\ \boldsymbol{M} \succcurlyeq 0\right\} = \left\{(\boldsymbol{H},\boldsymbol{M}) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{n\times n} \mid f_\ell(\boldsymbol{H},\boldsymbol{M}) \le 0, \boldsymbol{H} \succcurlyeq 0,\ \boldsymbol{M} \succcurlyeq 0\right\}.$$

In the sequel, we will refer to

$$\tilde{f}_\ell(\boldsymbol{H},\boldsymbol{M}) := \gamma_\ell(\boldsymbol{H},\boldsymbol{M})\left(\sqrt{4\tilde{y}_\ell^2 + \tfrac{1}{m}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle - \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle)^2} - \tfrac{1}{\sqrt{m}}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\boldsymbol{H}\rangle + \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\boldsymbol{M}\rangle)\right),$$

which is same as $f_\ell(\boldsymbol{H},\boldsymbol{M})$ except the measurements y_ℓ^2 is now replaced by corresponding noiseless measurements $\tilde{y}_\ell^2$. Define a convex indicator function for the positive semidefinite cone:

$$\iota(\boldsymbol{H},\boldsymbol{M}) := \begin{cases} 0, & \boldsymbol{H} \succcurlyeq 0 \text{ and } \boldsymbol{M} \succcurlyeq 0 \\ +\infty, & \text{otherwise.} \end{cases}$$

Introduce the convex regularizer

$$\mathcal{J}(\boldsymbol{H},\boldsymbol{M}) = \mathrm{Tr}(\boldsymbol{H}) + \mathrm{Tr}(\boldsymbol{M}) + \iota(\boldsymbol{H},\boldsymbol{M}).$$

For analysis purposes, we will work with the following optimization program

$$\begin{aligned} &\text{minimize } \mathcal{J}(\boldsymbol{H},\boldsymbol{M}) \\ &\text{subject to } f_\ell(\boldsymbol{H},\boldsymbol{M}) \le 0,\ \ell = 1\dots m, \end{aligned} \quad (29)$$

where $f_\ell(\boldsymbol{H},\boldsymbol{M})$ is given by (27).The optimization program (29) is equivalent to (8) as the objective and constraint set remain unchanged. In the analysis later, we will also need the subdifferential $\nabla\tilde{f}_\ell$, evaluated at $(\tilde{\boldsymbol{H}},\tilde{\boldsymbol{M}})$, which are given by (10). One can verify that

$$-\frac{1}{\sqrt{m}}(\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\tilde{\boldsymbol{M}}\rangle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\tilde{\boldsymbol{H}}\rangle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*) \in \nabla\tilde{f}_\ell(\tilde{\boldsymbol{H}},\tilde{\boldsymbol{M}}). \quad (30)$$

To see this, refer to a brief derivation below

$$\begin{aligned} \frac{\partial\tilde{f}_\ell}{\partial\boldsymbol{H}}(\tilde{\boldsymbol{H}},\tilde{\boldsymbol{M}}) &= \gamma_\ell(\tilde{\boldsymbol{H}},\tilde{\boldsymbol{M}})\left(\frac{\frac{1}{m}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\tilde{\boldsymbol{H}}\rangle - \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\tilde{\boldsymbol{M}}\rangle)\boldsymbol{b}_\ell\boldsymbol{b}_\ell^*}{\sqrt{4\tilde{y}_\ell^2 + \frac{1}{m}(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\tilde{\boldsymbol{H}}\rangle - \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\tilde{\boldsymbol{M}}\rangle)^2}} - \frac{1}{\sqrt{m}}\boldsymbol{b}_\ell\boldsymbol{b}_\ell^*\right) \\ &= -\frac{1}{\sqrt{m}}\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\tilde{\boldsymbol{M}}\rangle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \end{aligned}$$

where the last equality follows by using $\tilde{y}_\ell^2 = \frac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*,\tilde{\boldsymbol{H}}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*,\tilde{\boldsymbol{M}}\rangle$.

We now build some preliminaries required to characterize the set of descent directions for the objective function of the optimization program (29). Let $T_{\tilde{h}}$, and $T_{\tilde{m}}$ be the set of symmetric matrices of the form

$$T_{\tilde{h}} := \{\boldsymbol{X} = \tilde{\boldsymbol{h}}\boldsymbol{z}^* + \boldsymbol{z}\tilde{\boldsymbol{h}}^*\},\ T_{\tilde{m}} := \{\boldsymbol{X} = \tilde{\boldsymbol{m}}\boldsymbol{z}^* + \boldsymbol{z}\tilde{\boldsymbol{m}}^*\},$$

and denote the orthogonal complements by $T_{\tilde{h}}^\perp$, and $T_{\tilde{m}}^\perp$, respectively. Note that $\boldsymbol{X} \in T_{\tilde{h}}^\perp$ iff both the row and column spaces of $\boldsymbol{X}$ are perpendicular to $\tilde{\boldsymbol{h}}$. $\mathcal{P}_{T_{\tilde{h}}}$ denotes the orthogonal projection onto the set $T_{\tilde{h}}$, and a matrix $\boldsymbol{X}$ of appropriate dimensions can be projected into $T_{\tilde{h}}$ as

$$\mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{X}) := \frac{\tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2}\boldsymbol{X} + \boldsymbol{X}\frac{\tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2} - \frac{\tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2}\boldsymbol{X}\frac{\tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2}.$$

Similarly, define the projection operator $\mathcal{P}_{T_{\tilde{m}}}$. The projection onto orthogonal complements are then simply $\mathcal{P}_{T_{\tilde{h}}^\perp} := \mathcal{I} - \mathcal{P}_{T_{\tilde{h}}}$, and $\mathcal{P}_{T_{\tilde{m}}^\perp} := \mathcal{I} - \mathcal{P}_{T_{\tilde{m}}}$, where $\mathcal{I}$ is the identity operator. We use $\boldsymbol{X}_{T_{\tilde{h}}}$ as a shorthand for $\mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{X})$. Let $(\boldsymbol{W}_1,\boldsymbol{W}_2) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{n\times n}$, and $(\boldsymbol{W}_{1,T_{\tilde{h}}^\perp},\boldsymbol{W}_{2,T_{\tilde{m}}^\perp}) := (\mathcal{P}_{T_{\tilde{h}}^\perp}(\boldsymbol{W}_1),\mathcal{P}_{T_{\tilde{m}}^\perp}(\boldsymbol{W}_2))$. The subgradient $\partial\mathcal{J}(\boldsymbol{H},\boldsymbol{M})$ of the objective $\mathcal{J}(\tilde{\boldsymbol{H}},\boldsymbol{M})$ at the proposed solution $(\tilde{\boldsymbol{H}},\tilde{\boldsymbol{M}})$ is

$$\partial\mathcal{J}(\tilde{\boldsymbol{H}},\tilde{\boldsymbol{M}}) = \left\{\boldsymbol{G} \in \mathcal{H}^{k\times k} \times \mathcal{H}^{n\times n} \mid \boldsymbol{G} = \left(\frac{\tilde{\boldsymbol{H}}}{\|\tilde{\boldsymbol{H}}\|_F}, \frac{\tilde{\boldsymbol{M}}}{\|\tilde{\boldsymbol{M}}\|_F}\right) + (\boldsymbol{W}_{1,T_{\tilde{h}}^\perp},\boldsymbol{W}_{2,T_{\tilde{m}}^\perp}),\ \lambda_{\max}(\boldsymbol{W}_{1,T_{\tilde{h}}^\perp},\boldsymbol{W}_{2,T_{\tilde{m}}^\perp}) \le 1\right\}, \quad (31)$$

for details; see, Section 8.6 in Tropp (2015), and references therein.

Given the measurements (5), one can only identify the true solution $(\boldsymbol{H}^\natural,\boldsymbol{M}^\natural)$ up to the bilinear scaling ambiguity. To formalize this, begin by defining a set

$$\mathcal{N} := \{(\boldsymbol{H},\boldsymbol{M}) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{n\times n} \mid (\boldsymbol{H},\boldsymbol{M}) = \beta(-\tilde{\boldsymbol{H}},\tilde{\boldsymbol{M}}), \beta \in \mathbb{R}\}, \quad (32)$$

and denote by $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ a set $\mathcal{N}$ shifted by a point $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$. Mathematically,

$$(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N} = \{(\boldsymbol{H}, \boldsymbol{M}) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{n\times n} \mid (\boldsymbol{H}, \boldsymbol{M}) = ((1-\beta)\tilde{\boldsymbol{H}}, (1+\beta)\tilde{\boldsymbol{M}}), \beta \in \mathbb{R}\}. \tag{33}$$

We will refer to this set as the linearized global scaling of $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$.

The main argument of stable recovery in the noisy case is summarized as follows: Let

$$\mathcal{N}_\perp := \{(\boldsymbol{H}, \boldsymbol{M}) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{n\times n} \mid \langle \boldsymbol{H}, -\tilde{\boldsymbol{H}}\rangle + \langle \boldsymbol{M}, \tilde{\boldsymbol{M}}\rangle = 0\}$$

be the orthogonal complement of the subspace $\mathcal{N}$. The **first step** consists of showing that any feasible perturbation $(\delta\boldsymbol{H}, \delta\boldsymbol{M}) \in \mathcal{N}_\perp$ about the linearized scaling ambiguity $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ cannot be too large. This only shows that a large perturbation in the $\mathcal{N}_\perp$ direction is not allowed, however, the movement away from the ground truth $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ along the line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ can still be arbitrarily large. In the **second step**, we note that the straight line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ touches (in the noiseless case) the hyperbolic feasible set at $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$, and diverges away from it as we deviate away (large β) from the point $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ along the line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$. However, moving too far away along the line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ would make it impossible to jump back into the hyperbolic feasible region while not exceeding the allowed leap length in the $\mathcal{N}_\perp$ direction prescribed in the first step. This allows us to control deviations from $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ along $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ as well. Combining the limited allowed deviations both in $\mathcal{N}$, and $\mathcal{N}_\perp$ from the ground truth $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ enables us to show that program recovers a solution $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ in the neighbourhood of $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$. In the noiseless case, the same argument leads to an exact recovery result.

We now formally proceed with the proof argument. The set $\mathcal{Q}$ of descent directions only in $\mathcal{N}_\perp$ of the objective function in (29) is characterized as follows

$$\begin{aligned}
&\{(\delta\boldsymbol{H}, \delta\boldsymbol{M}) \in \mathcal{N}_\perp \mid \langle (\boldsymbol{G}, (\delta\boldsymbol{H}, \delta\boldsymbol{M})\rangle \le 0, \forall \boldsymbol{G} \in \partial\mathcal{J}(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\} \subseteq \\
&\left\{(\delta\boldsymbol{H}, \delta\boldsymbol{M}) \in \mathcal{N}_\perp \;\middle|\; \left\langle \left(\frac{\tilde{\boldsymbol{H}}}{\|\tilde{\boldsymbol{H}}\|_F}, \frac{\tilde{\boldsymbol{M}}}{\|\tilde{\boldsymbol{M}}\|_F}\right), (\delta\boldsymbol{H}_{T_{\tilde{h}}}, \delta\boldsymbol{M}_{T_{\tilde{m}}})\right\rangle + \mathrm{Tr}(\delta\boldsymbol{H}_{T_{\tilde{h}}^\perp}, \delta\boldsymbol{M}_{T_{\tilde{m}}^\perp}) \le 0\right\} \subseteq \\
&\{(\delta\boldsymbol{H}, \delta\boldsymbol{M}) \in \mathcal{N}_\perp \mid \mathrm{Tr}(\delta\boldsymbol{H}_{T_{\tilde{h}}^\perp}, \delta\boldsymbol{M}_{T_{\tilde{m}}^\perp}) \le \sqrt{2}\|(\delta\boldsymbol{H}_{T_{\tilde{h}}}, \delta\boldsymbol{M}_{T_{\tilde{m}}})\|_F\} =: \mathcal{Q},
\end{aligned} \tag{34}$$

where the first set containment follows by using $\langle (\boldsymbol{W}_{1,T_{\tilde{h}}^\perp}, \boldsymbol{W}_{2,T_{\tilde{m}}^\perp}), (\delta\boldsymbol{H}_{T_{\tilde{h}}^\perp}, \delta\boldsymbol{M}_{T_{\tilde{m}}^\perp})\rangle \le \mathrm{Tr}(\delta\boldsymbol{H}_{T_{\tilde{h}}^\perp}, \delta\boldsymbol{M}_{T_{\tilde{m}}^\perp})$, which follows from $\lambda_{\max}(\boldsymbol{W}_{1,T_{\tilde{h}}^\perp}, \boldsymbol{W}_{2,T_{\tilde{m}}^\perp}) \le 1$, and $(\delta\boldsymbol{H}_{T_{\tilde{h}}^\perp}, \delta\boldsymbol{M}_{T_{\tilde{m}}^\perp}) \succcurlyeq \boldsymbol{0}$ as any feasible perturbation must obey $(\tilde{\boldsymbol{H}} + \delta\boldsymbol{H}, \tilde{\boldsymbol{M}} + \delta\boldsymbol{M}) \succcurlyeq \boldsymbol{0}$. Last containment simply uses Cauchy-Schwartz inequality, and the fact that

$$\left\|\left(\frac{\tilde{\boldsymbol{H}}}{\|\tilde{\boldsymbol{H}}\|_F}, \frac{\tilde{\boldsymbol{M}}}{\|\tilde{\boldsymbol{M}}\|_F}\right)\right\|_F = \sqrt{2}.$$

We quantify the "width" of the set of descent directions $\mathcal{Q}$ through a Rademacher complexity, and a probability that the gradients $\nabla\tilde{f}_\ell$ in (30) of the constraint functions of (29) lie in a certain half space. This enables us to build an argument using the small ball method (Koltchinskii and Mendelson, 2015; Mendelson, 2014) that it is unlikely to have points that meet the constraints in (29) and still be in $\mathcal{Q}$. Before moving forward, we introduce the above mentioned Rademacher complexity and probability term.

For a set $\mathcal{Q} \subset (\mathcal{H}^{k\times k}, \mathcal{H}^{n\times n})$, the Rademacher complexity of the gradients $\nabla\tilde{f}_\ell$ in (30) is defined as

$$\mathfrak{C}(\mathcal{Q}) := \mathbb{E} \sup_{(\boldsymbol{H},\boldsymbol{M})\in\mathcal{Q}} \sum_{\ell=1}^{m} \varepsilon_\ell \left\langle \nabla\tilde{f}_\ell, \frac{(\boldsymbol{H},\boldsymbol{M})}{\|(\boldsymbol{H},\boldsymbol{M})\|_F}\right\rangle, \tag{35}$$

where $\varepsilon_\ell,\ \ell = 1\dots m$ are iid Rademacher random variables independent of everything else in the expression. For a convex set $\mathcal{Q}$, $\mathfrak{C}(\mathcal{Q})$ is a measure of the width of $\mathcal{Q}$ around origin in terms of the gradients $\nabla\tilde{f}_\ell,\ \ell = 1\dots m$. For example, random choice of gradient might yield little overlap with a structured set $\mathcal{Q}$ leading to a smaller complexity $\mathcal{Q}$.

Our result also depends on a probability $p_\tau(\mathcal{Q})$ and a positive parameter τ defined as

$$p_\tau(\mathcal{Q}) := \inf_{(\boldsymbol{H},\boldsymbol{M})\in\mathcal{Q}} \mathbb{P}\Big(\langle \nabla\tilde{f}, (\boldsymbol{H},\boldsymbol{M})\rangle \ge \frac{\tau}{\sqrt{m}}\|(\boldsymbol{H},\boldsymbol{M})\|_F\Big). \tag{36}$$

The probability $p_\tau(\mathcal{Q})$ quantifies visibility of the set $\mathcal{Q}$ through the gradient vectors[3] $\nabla\tilde{f}$. A small value of τ and $p_\tau(\mathcal{Q})$ means that the set $\mathcal{Q}$ mainly remains invisible through the lenses of $\nabla\tilde{f}_\ell, \ell = 1, 2, 3, \dots, m$. This can be appreciated just by noting that $p_\tau(\mathcal{Q})$ depends on the correlation of the elements of $\mathcal{Q}$ with the gradient vectors $\nabla\tilde{f}_\ell$.

The following lemma shows that the minimizer of (29) almost always resides in the desired set $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ for a sufficiently large m quantified interms of $\mathfrak{C}(\mathcal{Q})$, $p_\tau(\mathcal{Q})$, and τ.

Lemma 4 *Given the noisy measurements* (5)*, where the additive noise* $\boldsymbol{\xi}$ *obeys* (6)*. For signal recovery, consider the optimization program in* (29)*, and let* $\mathcal{Q}$*, characterized in* (34)*, be the set of descent directions for which* $\mathfrak{C}(\mathcal{Q})$*, and* $p_\tau(\mathcal{Q})$ *can be determined using* (35) *and* (36)*. Choose*

$$m \ge \left(\frac{2\mathfrak{C}(\mathcal{Q}) + t\tau}{\tau p_{\tau(\mathcal{Q})}}\right)^2$$

for any $t > 0$*. Then the minimizer* $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ *of* (29) *satisfies*

$$\|(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}}) - (\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F^2 \le 44^2\|\boldsymbol{\xi}\|_\infty \left(\|\tilde{\boldsymbol{H}}\|_F^2 + \|\tilde{\boldsymbol{M}}\|_F^2\right)$$

with probability at least $1 - \mathrm{e}^{-2mt^2}$.

Proof of this lemma is based on small ball method developed in Koltchinskii and Mendelson (2015); Mendelson (2014) and further studied in Lecué et al. (2018); Lecué and Mendelson (2017). The proof is repeated using the argument in Bahmani and Romberg (2017a), and is provided in the supplementary material for completeness.

Lemma 4 proves that under the choice of m outlined in Lemma 4, the optimization program (8) recovers $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ exactly in the noiseless case $\boldsymbol{\xi} = \boldsymbol{0}$, and stably in the noisy case. The last missing piece in the proof of Theorem 2 is to quantify the Rademacher complexity $\mathfrak{C}(\mathcal{Q})$, and $p_\tau(\mathcal{Q})$ for the $\mathcal{Q}$ appearing in the measurement bound.

3. We drop the subscript ℓ here as $\nabla\tilde{f}_\ell,\ \ell = 1\dots m$ are identically distributed

4.1. Rademacher Complexity

We begin with evaluation of the complexity $\mathfrak{C}(\mathcal{Q})$

$$\mathfrak{C}(\mathcal{Q}) := \mathbb{E} \sup_{(\delta \boldsymbol{H}, \delta \boldsymbol{M}) \in \mathcal{Q}} \sum_{\ell=1}^{m} \varepsilon_\ell \Big\langle \nabla \tilde{f}_\ell, \tfrac{(\delta \boldsymbol{H}, \delta \boldsymbol{M})}{\|(\delta \boldsymbol{H}, \delta \boldsymbol{M})\|_F} \Big\rangle$$

Splitting $(\delta \boldsymbol{H}, \delta \boldsymbol{M})$ between $(T_{\tilde{h}}, T_{\tilde{m}})$, and $(T_{\tilde{h}}^{\perp}, T_{\tilde{m}}^{\perp})$, and using Holder's inequalities, we obtain

$$\begin{aligned}
&\mathfrak{C}(\mathcal{Q}) \leq \\
&\mathbb{E}\Big\| \tfrac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell (\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*), \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \mathcal{P}_{T_{\tilde{m}}}(\boldsymbol{c}_\ell \boldsymbol{c}_\ell^*)) \Big\|_F \cdot \sup_{(\delta \boldsymbol{H}, \delta \boldsymbol{M}) \in \mathcal{Q}} \Big\| \tfrac{(\delta \boldsymbol{H}_{T_{\tilde{h}}}, \delta \boldsymbol{M}_{T_{\tilde{m}}})}{\|(\delta \boldsymbol{H}, \delta \boldsymbol{M})\|_F} \Big\|_F \\
&+ \mathbb{E}\Big\| \frac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell (\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*) \Big\| \cdot \sup_{(\delta \boldsymbol{H}, \delta \boldsymbol{M}) \in \mathcal{Q}} \Big\| \tfrac{\left(\delta \boldsymbol{H}_{T_{\tilde{h}}^{\perp}}, \delta \boldsymbol{M}_{T_{\tilde{m}}^{\perp}}\right)}{\|(\delta \boldsymbol{H}, \delta \boldsymbol{M})\|_F} \Big\|_* .
\end{aligned}$$

Recall that $(\delta \boldsymbol{H}_{T_{\tilde{h}}^{\perp}}, \delta \boldsymbol{M}_{T_{\tilde{m}}^{\perp}}) \succcurlyeq \boldsymbol{0}$, and, therefore, $\mathrm{Tr}(\delta \boldsymbol{H}_{T_{\tilde{h}}^{\perp}}, \delta \boldsymbol{M}_{T_{\tilde{m}}^{\perp}}) = \|(\delta \boldsymbol{H}_{T_{\tilde{h}}^{\perp}}, \delta \boldsymbol{M}_{T_{\tilde{m}}^{\perp}})\|_*$. On the set $\mathcal{Q}$, defined in (34), we have

$$\frac{\mathrm{Tr}(\delta \boldsymbol{H}_{T_{\tilde{h}}^{\perp}}, \delta \boldsymbol{M}_{T_{\tilde{m}}^{\perp}})}{\|(\delta \boldsymbol{H}, \delta \boldsymbol{M})\|_F} \leq \sqrt{2} \Big\| \frac{\left(\delta \boldsymbol{H}_{T_{\tilde{h}}}, \delta \boldsymbol{M}_{T_{\tilde{m}}}\right)}{\|(\delta \boldsymbol{H}, \delta \boldsymbol{M})\|_F} \Big\|_F \leq \sqrt{2}.$$

Using Jensen's inequality, the first expectation simply becomes

$$\begin{aligned}
&\mathbb{E}\Big\| \frac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell \big(\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*), \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \mathcal{P}_{T_{\tilde{m}}}(\boldsymbol{c}_\ell \boldsymbol{c}_\ell^*)\big) \Big\|_F \\
&\leq \sqrt{\frac{1}{m} \mathbb{E}\Big\| \sum_{\ell=1}^{m} \varepsilon_\ell \big(\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*), \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \mathcal{P}_{T_{\tilde{m}}}(\boldsymbol{c}_\ell \boldsymbol{c}_\ell^*)\big) \Big\|_F^2} \\
&= \sqrt{\frac{1}{m} \sum_{\ell=1}^{m} \mathbb{E}\Big(\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \|\mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*)\|_F^2 + \|\langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \mathcal{P}_{T_{\tilde{m}}}(\boldsymbol{c}_\ell \boldsymbol{c}_\ell^*)\|_F^2 \Big)},
\end{aligned}$$

where the last equality follows by going through with the expectation over ε_ℓ's. Recall from the definition of the projection operator that $\mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*) := \frac{\tilde{\boldsymbol{h}} \tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2} \boldsymbol{b}_\ell \boldsymbol{b}_\ell^* + \boldsymbol{b}_\ell \boldsymbol{b}_\ell^* \frac{\tilde{\boldsymbol{h}} \tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2} - \frac{\tilde{\boldsymbol{h}} \tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2} \boldsymbol{b}_\ell \boldsymbol{b}_\ell^* \frac{\tilde{\boldsymbol{h}} \tilde{\boldsymbol{h}}^*}{\|\tilde{\boldsymbol{h}}\|_2^2}$. It can be easily verified that $\|\mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*)\|_F^2 = 2 \frac{|\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^2}{\|\tilde{\boldsymbol{h}}\|_2^2} \|\boldsymbol{b}_\ell\|_2^2 - \frac{|\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^4}{\|\tilde{\boldsymbol{h}}\|_2^4}$, and, therefore,

$$\mathbb{E}\|(\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*)\|_F^2 \leq \mathbb{E}|\boldsymbol{c}_\ell^* \tilde{\boldsymbol{m}}|_2^4 \cdot \mathbb{E}\left(2 \frac{|\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^2}{\|\tilde{\boldsymbol{h}}\|_2^2} \|\boldsymbol{b}_\ell\|_2^2 - \frac{|\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^4}{\|\tilde{\boldsymbol{h}}\|_2^4}\right) \leq 3 \|\tilde{\boldsymbol{m}}\|_2^4 \,(6k-3)\,,$$

where we used a simple calculation involving fourth moments of Gaussian $\mathbb{E}|\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^2 \|\boldsymbol{b}_\ell\|_2^2 = 3k \|\tilde{\boldsymbol{h}}\|_2^2$. In an exactly similar manner, we can also show that $\|(\langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \mathcal{P}_{T_{\tilde{m}}}(\boldsymbol{c}_\ell \boldsymbol{c}_\ell^*)\|_F^2 \leq 3 \|\tilde{\boldsymbol{h}}\|_2^4 (6n-3)$. Putting these together gives us

$$\mathbb{E}\Big\| \frac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell \big(\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \mathcal{P}_{T_{\tilde{h}}}(\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*), \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \mathcal{P}_{T_{\tilde{m}}}(\boldsymbol{c}_\ell \boldsymbol{c}_\ell^*)\big) \Big\|_F \leq 5 \max(\|\tilde{\boldsymbol{h}}\|_2^2, \|\tilde{\boldsymbol{m}}\|_2^2) \sqrt{k+n}.$$

Moreover,

$$\begin{aligned}
&\mathbb{E}\Big\| \tfrac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell \big(\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*\big) \Big\| \leq \\
&\qquad \mathbb{E} \max_\ell (\langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle, \langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle) \cdot \mathbb{E}\Big\| \tfrac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell (\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*) \Big\|.
\end{aligned}$$

Standard net arguments; see, for example, Sec. 5.4.1 of Eldar and Kutyniok (2012) show that

$$\mathbb{P}\left(\Big\| \tfrac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell (\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*) \Big\| \geq c\sqrt{k+n} \right) \leq \mathrm{e}^{-cm}, \text{ provided that } m \geq c(k+n).$$

This directly implies that $\mathbb{E}\Big\| \frac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell (\boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*) \Big\| \leq c\sqrt{k+n}$. The random variables u_ℓ and v_ℓ being sub-exponential have Orlicz-1 norms bounded by $c \max(\|\tilde{\boldsymbol{h}}\|_2^2, \|\tilde{\boldsymbol{m}}\|_2^2)$. Using standard results, such as Lemma 3 in van de Geer and Lederer (2013), we then have $\mathbb{E} \max_\ell (u_\ell, v_\ell) \leq c \log m$. Putting these together yields

$$\mathbb{E}\Big\| \tfrac{1}{\sqrt{m}} \sum_{\ell=1}^{m} \varepsilon_\ell \big(\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}} \rangle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}} \rangle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*\big) \Big\| \leq c \max(\|\tilde{\boldsymbol{h}}\|_2^2, \|\tilde{\boldsymbol{m}}\|_2^2) \sqrt{(k+n) \log^2 m}. \quad (37)$$

We have all the ingredients for the final bound on $\mathfrak{C}(\mathcal{Q})$ stated below

$$\mathfrak{C}(\mathcal{Q}) \leq c \max(\|\tilde{\boldsymbol{h}}\|_2^2, \|\tilde{\boldsymbol{m}}\|_2^2) \sqrt{(k+n) \log^2 m}. \quad (38)$$

4.2. Probability $p_\tau(\mathcal{Q})$

In this section, we determine the probability $p_\tau(\mathcal{Q})$, and the positive parameter τ in (36) for the set $\mathcal{Q}$ in (34). For a point $(\delta \boldsymbol{H}, \delta \boldsymbol{M}) \in \mathcal{Q}$, and randomly chosen $\nabla \tilde{f}_\ell$, we have via the Paley-Zygmund inequality that

$$\mathbb{P}\Big(\Big| \big\langle \nabla \tilde{f}_\ell, (\delta \boldsymbol{H}, \delta \boldsymbol{M}) \big\rangle \Big|^2 \geq \frac{1}{2} \mathbb{E} \Big| \big\langle \nabla \tilde{f}_\ell, (\delta \boldsymbol{H}, \delta \boldsymbol{M}) \big\rangle \Big|^2 \Big) \geq \frac{1}{4} \frac{\left(\mathbb{E} \big| \big\langle \nabla \tilde{f}_\ell, (\delta \boldsymbol{H}, \delta \boldsymbol{M}) \big\rangle \big|^2\right)^2}{\mathbb{E} \big| \big\langle \nabla \tilde{f}_\ell, (\delta \boldsymbol{H}, \delta \boldsymbol{M}) \big\rangle \big|^4}.$$

The particular choice of random gradient vectors we are using is

$$\nabla \tilde{f}_\ell = (1/\sqrt{m})(|\boldsymbol{c}_\ell^* \tilde{\boldsymbol{m}}|^2 \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, |\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^2 \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*),$$

giving us $\big\langle \nabla \tilde{f}_\ell, (\delta \boldsymbol{H}, \delta \boldsymbol{M}) \big\rangle = (1/\sqrt{m}) |\boldsymbol{c}_\ell^* \tilde{\boldsymbol{m}}|^2 \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \delta \boldsymbol{H} \rangle + |\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^2 \langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \delta \boldsymbol{M} \rangle$. Since $\boldsymbol{b}_\ell$, and $\boldsymbol{c}_\ell$ are standard Gaussian vectors, using the equivalence of L_p-norms for Gaussian, we deduce that

$$\begin{aligned}
&\left(\mathbb{E} \Big| |\boldsymbol{c}_\ell^* \tilde{\boldsymbol{m}}|^2 \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \delta \boldsymbol{H} \rangle + |\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^2 \langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \delta \boldsymbol{M} \rangle \Big|^4 \right)^{1/4} \leq \\
&\qquad c \left(\mathbb{E} \Big| |\boldsymbol{c}_\ell^* \tilde{\boldsymbol{m}}|^2 \langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \delta \boldsymbol{H} \rangle + |\boldsymbol{b}_\ell^* \tilde{\boldsymbol{h}}|^2 \langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \delta \boldsymbol{M} \rangle \Big|^2 \right)^{1/2}.
\end{aligned}$$

Plugging last two inequalities in (36) reveals that

$$p_\tau(\mathcal{Q}) \geq c > 0 \tag{39}$$

for an absolute constant c. To compute τ, we expand $\mathbb{E}\left|\langle \boldsymbol{g}_\ell, (\delta\boldsymbol{H}, \delta\boldsymbol{M})\rangle\right|^2$ giving us

$$\begin{aligned}\mathbb{E}\left||\boldsymbol{c}_\ell^*\tilde{\boldsymbol{m}}|^2\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \delta\boldsymbol{H}\rangle + |\boldsymbol{b}_\ell^*\tilde{\boldsymbol{h}}|^2\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \delta\boldsymbol{M}\rangle\right|^2 &= 3\|\tilde{\boldsymbol{m}}\|_2^4(\langle \mathrm{diag}(\delta\boldsymbol{H}), \delta\boldsymbol{H}\rangle + 2\|\delta\boldsymbol{H}\|_F^2)\\ &+ 3\|\tilde{\boldsymbol{h}}\|_2^4(\langle \mathrm{diag}(\delta\boldsymbol{M}), \delta\boldsymbol{M}\rangle + 2\|\delta\boldsymbol{M}\|_F^2) + 2|\tilde{\boldsymbol{h}}^*\mathrm{diag}(\delta\boldsymbol{H})\tilde{\boldsymbol{h}} + 2\tilde{\boldsymbol{h}}^*\delta\boldsymbol{H}\tilde{\boldsymbol{h}}|^2,\end{aligned} \tag{40}$$

where we have made use of multiple simple facts including that $\mathbb{E}|\boldsymbol{b}_\ell^*\tilde{\boldsymbol{h}}|^4 = 3\|\tilde{\boldsymbol{h}}\|_2^4$, and similarly for $|\boldsymbol{c}_\ell^*\tilde{\boldsymbol{m}}|^2$, and two identities: $\mathbb{E}|\boldsymbol{b}_\ell^*\tilde{\boldsymbol{h}}|^2\boldsymbol{b}_\ell^*\delta\boldsymbol{H}\boldsymbol{b}_\ell = \tilde{\boldsymbol{h}}^*\mathrm{diag}(\delta\boldsymbol{H})\tilde{\boldsymbol{h}} + 2\tilde{\boldsymbol{h}}^*\delta\boldsymbol{H}\tilde{\boldsymbol{h}}$, and $\mathbb{E}(\boldsymbol{b}_\ell^*\delta\boldsymbol{H}\boldsymbol{b}_\ell)\boldsymbol{b}_\ell\boldsymbol{b}_\ell^* = \mathrm{diag}(\delta\boldsymbol{H}) + 2(\delta\boldsymbol{H}) \implies \mathbb{E}|\boldsymbol{b}_\ell^*\delta\boldsymbol{H}\boldsymbol{b}_\ell|^2 = \langle \mathrm{diag}(\delta\boldsymbol{H}), \delta\boldsymbol{H}\rangle + 2\|\delta\boldsymbol{H}\|_F^2$. We also made use of the fact that $\mathcal{Q} \perp \mathcal{N}$ and therefore $\langle \tilde{\boldsymbol{H}}, \delta\boldsymbol{H}\rangle - \langle \tilde{\boldsymbol{M}}, \delta\boldsymbol{M}\rangle = 0$, or equivalently, $\tilde{\boldsymbol{h}}^*\delta\boldsymbol{H}\tilde{\boldsymbol{h}} = \tilde{\boldsymbol{m}}^*\delta\boldsymbol{M}\tilde{\boldsymbol{m}}$.

It is easy to conclude from (40) now that

$$\begin{aligned}&\mathbb{E}\left||\boldsymbol{c}_\ell^*\tilde{\boldsymbol{m}}|^2\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \delta\boldsymbol{H}\rangle + |\boldsymbol{b}_\ell^*\tilde{\boldsymbol{h}}|^2\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \delta\boldsymbol{M}\rangle\right|^2 \geq 6(\|\tilde{\boldsymbol{h}}\|_2^4\|\delta\boldsymbol{H}\|_F^2 + \|\tilde{\boldsymbol{m}}\|_2^4\|\delta\boldsymbol{M}\|_F^2)\\ &\geq c\min(\|\tilde{\boldsymbol{h}}\|_2^2, \|\tilde{\boldsymbol{m}}\|_2^2)(\|\delta\boldsymbol{H}\|_F^2 + \|\delta\boldsymbol{M}\|_F^2) = c\max(\|\tilde{\boldsymbol{h}}\|_2^2, \|\tilde{\boldsymbol{m}}\|_2^2)(\|\delta\boldsymbol{H}\|_F^2 + \|\delta\boldsymbol{M}\|_F^2),\end{aligned}$$

where the last equality uses the fact that $\mathrm{Tr}(\tilde{\boldsymbol{H}}) = \mathrm{Tr}(\tilde{\boldsymbol{M}})$ from (10), which is equivalent to $\|\tilde{\boldsymbol{h}}\|_2^2 = \tilde{\boldsymbol{m}}\|_2^2$. This directly means, we can take $\tau = c\max(\|\tilde{\boldsymbol{h}}\|_2^2, \|\tilde{\boldsymbol{m}}\|_2^2)$, where c is an absolute constant.

The complexity estimate in (38), value of τ computed above, and $p_\tau(\mathcal{Q})$ stated in (39) together with an application of Lemma 4 prove Theorem 2.

5. Proof of Lemma 4

The proof is based on small ball method developed in Koltchinskii and Mendelson (2015); Mendelson (2014) and further studied in Lecué et al. (2018) and Lecué and Mendelson (2017).

Introduce a one sided loss function:

$$\mathcal{L}(\boldsymbol{H}, \boldsymbol{M}) := \frac{1}{m}\sum_{\ell=1}^m [f_\ell(\boldsymbol{H}, \boldsymbol{M})]_+, \tag{41}$$

where $(\cdot)_+$ denotes the positive side, and $f_\ell(\boldsymbol{H}, \boldsymbol{M})$ is a convex function as defined in (27). Using this definition, we rewrite (29) compactly as

$$\begin{aligned}&\text{minimize } \mathcal{J}(\boldsymbol{H}, \boldsymbol{M})\\ &\text{subject to } \mathcal{L}(\boldsymbol{H}, \boldsymbol{M}) \leq 0.\end{aligned} \tag{42}$$

Our objective is to show that any feasible perturbation $(\delta\boldsymbol{H}, \delta\boldsymbol{M}) \in \mathcal{Q}$ around any member $(\boldsymbol{H}^\dagger, \boldsymbol{M}^\dagger)$ of the linearized global scaling set $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ has a small Frobenius norm. Feasibility of the perturbation implies that

$$\mathcal{L}\big(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M}\big) \leq 0. \tag{43}$$

Expand the summands $[f_\ell(\boldsymbol{H}^\dagger+\delta\boldsymbol{H}, \boldsymbol{M}^\dagger+\delta\boldsymbol{M})]_+$ of the loss function $\mathcal{L}\big(\boldsymbol{H}^\dagger+\delta\boldsymbol{H}, \boldsymbol{M}^\dagger+\delta\boldsymbol{M}\big)$ to obtain

$$\begin{aligned}&[f_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M})]_+ =\\ &\gamma_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M})\Bigg[\sqrt{4y_\ell^2 - \frac{1}{m}\big(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}^\dagger + \delta\boldsymbol{H}\rangle - \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}^\dagger + \delta\boldsymbol{M}\rangle\big)^2}\\ &\qquad - \frac{1}{\sqrt{m}}\big(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}^\dagger + \delta\boldsymbol{H}\rangle + \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}^\dagger + \delta\boldsymbol{M}\rangle\big)\Bigg]_+.\end{aligned}$$

Recall that the noisy measurements y_ℓ^2, defined in (5), are related to the noiseless measurements $\tilde{y}_\ell^2$ through $y_\ell^2 = \tilde{y}_\ell^2(1+\xi_\ell)$. Using this relation together with triangle inequality gives

$$\begin{aligned}&[f_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M})]_+\\ &= \gamma_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M})\Big[\sqrt{4\tilde{y}_\ell{}^2 - \tfrac{1}{m}\big(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}^\dagger + \delta\boldsymbol{H}\rangle - \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}^\dagger + \delta\boldsymbol{M}\rangle\big)^2}\\ &\qquad - \tfrac{1}{\sqrt{m}}\big(\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}^\dagger + \delta\boldsymbol{H}\rangle + \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}^\dagger + \delta\boldsymbol{M}\rangle\big) - \sqrt{[-4\tilde{y}_\ell^2\xi_\ell]_+}\Big]_+\\ &\geq [\tilde{f}_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M})]_+ - 2\gamma_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M})\sqrt{[-\tilde{y}_\ell^2\xi_\ell]_+}\\ &\geq [\tilde{f}_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M})]_+ - 2\sqrt{[-\tilde{y}_\ell^2\xi_\ell]_+}\\ &\geq [\tilde{f}_\ell(\tilde{\boldsymbol{H}} - \beta\tilde{\boldsymbol{H}} + \delta\boldsymbol{H}, \tilde{\boldsymbol{M}} + \beta\tilde{\boldsymbol{M}} + \delta\boldsymbol{M})]_+ - 2\sqrt{[-\tilde{y}_\ell^2\xi_\ell]_+}\\ &\geq [\langle \nabla\tilde{f}_\ell, (-\beta\tilde{\boldsymbol{H}} + \delta\boldsymbol{H}, \beta\tilde{\boldsymbol{M}} + \delta\boldsymbol{M})\rangle]_+ - 2\sqrt{[-\tilde{y}_\ell^2\xi_\ell]_+}\\ &= [\langle \nabla\tilde{f}_\ell, (-\beta\tilde{\boldsymbol{H}} + \delta\boldsymbol{H}, \beta\tilde{\boldsymbol{M}} + \delta\boldsymbol{M})\rangle]_+ - 2\sqrt{[-\tilde{y}_\ell^2\xi_\ell]_+}\\ &= [\langle \nabla\tilde{f}_\ell, (\delta\boldsymbol{H}, \delta\boldsymbol{M})\rangle]_+ - 2\sqrt{[-\tilde{y}_\ell^2\xi_\ell]_+},\end{aligned} \tag{44}$$

where in the first inequality follows from the fact that if $a \geq 0$, and $b < 0$ with $a+b \geq 0$, then $\sqrt{a+b} \geq \sqrt{a} - \sqrt{-b}$ holds, and if $a \geq 0$, and $b \geq 0$, $\sqrt{a+b} \geq \sqrt{a}$ holds, the second inequality uses the fact that $\tilde{f}_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M}) \leq 0$ as $(\delta\boldsymbol{H}, \delta\boldsymbol{M})$ is a feasible perturbation of $(\boldsymbol{H}^\dagger, \boldsymbol{M}^\dagger)$, and hence using the definition (28) it holds that $\gamma_\ell(\boldsymbol{H}^\dagger + \delta\boldsymbol{H}, \boldsymbol{M}^\dagger + \delta\boldsymbol{M}) \leq 1$, the third simply uses the fact that $(\boldsymbol{H}^\dagger, \boldsymbol{M}^\dagger) \in (\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ is of the form $(\boldsymbol{H}^\dagger, \boldsymbol{M}^\dagger) = ((1-\beta)\tilde{\boldsymbol{H}}, (1+\beta)\tilde{\boldsymbol{M}})$ for some $\beta \in \mathbb{R}$ (More precisely, the scalar $\beta \in [-1, 1]$, as by feasibility $\boldsymbol{H}^\dagger$, and $\boldsymbol{M}^\dagger$ are PSD), and finally the last inequality uses the definition of sub-gradient (30) of the convex function $\tilde{f}_\ell$. The last equality uses the fact that $\langle \nabla\tilde{f}_\ell, (-\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\rangle = 0$.

Plugging the lower bound (44) in (43) produces

$$\begin{aligned}\sum_{\ell=1}^m [\langle \nabla\tilde{f}_\ell, (\delta\boldsymbol{H}, \delta\boldsymbol{M})\rangle]_+ &\leq 2\sum_{\ell=1}^m \sqrt{|\tilde{y}_\ell^2\xi_\ell|} \leq \frac{2}{\sqrt{m}}\sqrt{\|\boldsymbol{\xi}\|_\infty}\sum_{\ell=1}^m |\boldsymbol{b}_\ell^*\tilde{\boldsymbol{h}}||\boldsymbol{c}_\ell^*\tilde{\boldsymbol{m}}|\\ &\leq 2\sqrt{\frac{\|\boldsymbol{\xi}\|_\infty}{m}}\|\boldsymbol{B}\tilde{\boldsymbol{h}}\|_2\|\boldsymbol{C}\tilde{\boldsymbol{m}}\|_2 \leq 18\sqrt{m\|\boldsymbol{\xi}\|_\infty}\|\tilde{\boldsymbol{h}}\|_2\|\tilde{\boldsymbol{m}}\|_2,\end{aligned} \tag{45}$$

where the second last display follows just by using the fact that $\tilde{y}_\ell^2 = \frac{1}{m}\langle \boldsymbol{b}_\ell \boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}}\rangle\langle \boldsymbol{c}_\ell \boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}}\rangle$, where $\tilde{\boldsymbol{H}} = \tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^*$, and $\tilde{\boldsymbol{M}} = \tilde{\boldsymbol{m}}\tilde{\boldsymbol{m}}^*$, and the last display simply employs Cauchy Schwartz, and $\|\boldsymbol{B}\| \le 3\sqrt{m}$, and $\|\boldsymbol{C}\| \le 3\sqrt{m}$, which holds with probability at least $1 - \mathrm{e}^{-m/2}$.

Let $\psi_t(s) := (s)_+ - (s-t)_+$. Using the fact that $\psi_t(s) \le (s)_+$, and that for every $\alpha, t \ge 0$, and $s \in \mathbb{R}$, $\psi_{\alpha t}(s) = t\psi_\alpha(\frac{s}{t})$, we have

$$\begin{aligned}\frac{1}{\sqrt{m}}\sum_{\ell=1}^m \big[\langle \nabla \tilde{f}_\ell, (\delta\boldsymbol{H}, \delta\boldsymbol{M})\rangle\big]_+ &\ge \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \psi_{\tau\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}(\langle \nabla \tilde{f}_\ell, (\delta\boldsymbol{H}, \delta\boldsymbol{M})\rangle)\\ &= \|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \cdot \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big)\\ &= \|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \cdot \frac{1}{\sqrt{m}}\Bigg[\sum_{\ell=1}^m \mathbb{E}\psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big) - \\ &\quad \sum_{\ell=1}^m \Bigg[\mathbb{E}\psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big) - \psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big)\Bigg]\Bigg].\end{aligned} \tag{46}$$

Define a centred random process $\mathcal{R}(\boldsymbol{B}, \boldsymbol{C})$ as follows

$$\mathcal{R}(\boldsymbol{B}, \boldsymbol{C}) := \sup_{(\delta\boldsymbol{H},\delta\boldsymbol{M})\in\mathcal{Q}} \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \Bigg[\mathbb{E}\psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big) - \psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big)\Bigg]$$

and an application of bounded difference inequality (McDiarmid, 1989) yields that $\mathcal{R}(\boldsymbol{B}, \boldsymbol{C}) \le \mathbb{E}\mathcal{R}(\boldsymbol{B}, \boldsymbol{C}) + t\tau/\sqrt{m}$ with probability at least $1 - \mathrm{e}^{-2mt^2}$. It remains to evaluate $\mathbb{E}\mathcal{R}(\boldsymbol{B}, \boldsymbol{C})$, which after using a simple symmetrization inequality (van der Vaart and Wellner, 1997) yields

$$\mathbb{E}\mathcal{R}(\boldsymbol{B}, \boldsymbol{C}) \le 2\mathbb{E} \sup_{(\delta\boldsymbol{H},\delta\boldsymbol{M})\in\mathcal{Q}} \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \varepsilon_\ell \psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big), \tag{47}$$

where $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ are independent Rademacher random variables. Using the fact that $\psi_t(0) = 0$, and $\psi_t(s)$ is a contraction: $|\psi_t(\alpha_1) - \psi_t(\alpha_2)| \le |\alpha_1 - \alpha_2|$ for all $\alpha_1, \alpha_2 \in \mathbb{R}$, we have from the Rademacher contraction inequality (Ledoux and Talagrand, 2013) that

$$\begin{aligned}\mathbb{E} \sup_{(\delta\boldsymbol{H},\delta\boldsymbol{M})\in\mathcal{Q}} \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \varepsilon_\ell \psi_\tau\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\big) &\le \mathbb{E} \sup_{(\delta\boldsymbol{H},\delta\boldsymbol{M})\in\mathcal{Q}} \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \varepsilon_\ell \langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\\ &= \mathbb{E} \sup_{(\delta\boldsymbol{H},\delta\boldsymbol{M})\in\mathcal{Q}} \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \varepsilon_\ell \langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle,\end{aligned} \tag{48}$$

where the last equality is the result of the fact that a global sign change of a sequence of Rademacher random variables does not change their distribution. In addition, using the facts that $t\mathbf{1}(s \ge t) \le \psi_t(s)$, and that random vectors $\nabla \tilde{f}_1, \nabla \tilde{f}_2, \ldots, \nabla \tilde{f}_m$ are identically distributed and the distribution is symmetric, it follows

$$\frac{\tau}{\sqrt{m}}\mathbb{P}\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle \ge \tfrac{\tau}{\sqrt{m}}\big) = \frac{\tau}{\sqrt{m}}\mathbb{E}\Bigg(\mathbf{1}\Bigg[\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle \ge \tfrac{\tau}{\sqrt{m}}\Bigg]\Bigg)$$

$$\le \mathbb{E}\psi_{\tau/\sqrt{m}}\Big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\Big) = \frac{1}{\sqrt{m}}\mathbb{E}\psi_\tau\Big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\Big). \tag{49}$$

Plugging (49), and (48) in (46), we have

$$\begin{aligned}\frac{1}{\sqrt{m}}\sum_{\ell=1}^m \big[\langle \nabla \tilde{f}_\ell, (\delta\boldsymbol{H}, \delta\boldsymbol{M})\rangle\big]_+ &\ge \tau\|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \cdot \mathbb{P}\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle \ge \tfrac{\tau}{\sqrt{m}}\big)\\ &- 2\|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \mathbb{E} \sup_{(\delta\boldsymbol{H},\delta\boldsymbol{M})\in\mathcal{Q}} \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \varepsilon_\ell \langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle - 2\|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \frac{t\tau}{\sqrt{m}}.\end{aligned}$$

Using this lower bound in (45), we obtain

$$\begin{aligned}&\|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \Bigg[\tau\mathbb{P}\big(\langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle \ge \tfrac{\tau}{\sqrt{m}}\big) - 2\mathbb{E} \sup_{(\delta\boldsymbol{H},\delta\boldsymbol{M})\in\mathcal{Q}} \frac{1}{\sqrt{m}}\sum_{\ell=1}^m \varepsilon_\ell \langle \nabla \tilde{f}_\ell, \tfrac{(\delta\boldsymbol{H},\delta\boldsymbol{M})}{\|(\delta\boldsymbol{H},\delta\boldsymbol{M})\|_F}\rangle\Bigg]\\ &\quad - 2\|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \frac{t\tau}{\sqrt{m}} \le 18\sqrt{\|\boldsymbol{\xi}\|_\infty}\|\tilde{\boldsymbol{h}}\|_2\|\tilde{\boldsymbol{m}}\|_2.\end{aligned}$$

Using the definitions in (35), and (36), we can write

$$\|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F \left(\tau p_\tau(\mathcal{Q}) - \frac{(2\mathfrak{C}(\mathcal{Q}) + t\tau)}{\sqrt{m}}\right) \le 18\sqrt{\|\boldsymbol{\xi}\|_\infty}\|\tilde{\boldsymbol{h}}\|_2\|\tilde{\boldsymbol{m}}\|_2.$$

It is clear that choosing $m \ge \left(\frac{2\mathfrak{C}(\mathcal{Q})+t\tau}{\tau p_\tau(\mathcal{Q})}\right)^2$ implies that any feasible direction $(\delta\boldsymbol{H}, \delta\boldsymbol{M}) \in \mathcal{N}_\perp$ is bounded by

$$\|(\delta\boldsymbol{H}, \delta\boldsymbol{M})\|_F^2 \le 18^2\|\boldsymbol{\xi}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F \tag{50}$$

with probability at least $1 - \mathrm{e}^{-c_t m}$, where $c_t = ct^2$ for a universal constant c, where we used the fact that $\tilde{\boldsymbol{H}} = \tilde{\boldsymbol{h}}\tilde{\boldsymbol{h}}^*$, and $\tilde{\boldsymbol{M}} = \tilde{\boldsymbol{m}}\tilde{\boldsymbol{m}}^*$. Since $(\delta\boldsymbol{H}, \delta\boldsymbol{M}) \in \mathcal{N}_\perp$, the last display only gives us that an element, $((1-\beta_0)\tilde{\boldsymbol{H}}, (1+\beta_0)\tilde{\boldsymbol{M}})$ for some $\beta_0 \in \mathbb{R}$, of the set $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ obeys

$$\|(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}}) - ((1-\beta_0)\tilde{\boldsymbol{H}}, (1+\beta_0)\tilde{\boldsymbol{M}})\|_F^2 \le 18^2\|\boldsymbol{\xi}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F. \tag{51}$$

That is, the solution $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ cannot wander too far away from the line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$. We call this norm cylinder constraint as the solution must lie within a cylinder, centred at a line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ and of radius given by the rhs of the last display above. Equivalently, a displacement $((1-\beta_0)\tilde{\boldsymbol{H}}, (1+\beta_0)\tilde{\boldsymbol{M}})$ of the ground truth $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ is sufficiently close to $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$. Using this fact together with the fact that the feasible hyperbolic set diverges away from the $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ for larger displacement β, we will conclude in the remaining proof that the displacement β_0 cannot be too large, and hence the Euclidean distance between $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$, and the ground truth $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ is also bounded.

Case 1: Assume that noise $\boldsymbol{\xi}$ is such that $\xi_\ell \in [-1, 0]$ for every $\ell \in [m]$, and $\exists\ \ell' \in [m]$ such that $\xi_{\ell'} = 0$.

Trivially, the minimizer $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ must lie somewhere in the feasible set specified by the ℓ' constraint: $\frac{1}{m}\langle \boldsymbol{b}_{\ell'}\boldsymbol{b}_{\ell'}^*, \boldsymbol{H}\rangle\langle \boldsymbol{c}_{\ell'}\boldsymbol{c}_{\ell'}^*, \boldsymbol{M}\rangle \geq \tilde{y}_{\ell'}^2$. Define the boundary $\mathcal{B}$ of the feasible set above as follows

$$\mathcal{B} := \{(\boldsymbol{H}, \boldsymbol{M}) : \tfrac{1}{m}\langle \boldsymbol{b}_{\ell'}\boldsymbol{b}_{\ell'}^*, \boldsymbol{H}\rangle\langle \boldsymbol{c}_{\ell'}\boldsymbol{c}_{\ell'}^*, \boldsymbol{M}\rangle = \tilde{y}_{\ell'}^2\}. \tag{52}$$

The line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ only touches the feasible set at $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$. Define a plane $\mathcal{P} := \text{span}\{(\tilde{\boldsymbol{H}}, \boldsymbol{0}), (\boldsymbol{0}, \tilde{\boldsymbol{M}})\}$. Clearly, the line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ is contained in $\mathcal{P}$. Moreover, the intersection $\mathcal{P} \cap \mathcal{B}$ only happens at a set of bilinear scaling ambiguity, that is, $\mathcal{P} \cap \mathcal{B} = \{(\gamma\tilde{\boldsymbol{H}}, \frac{1}{\gamma}\tilde{\boldsymbol{M}})$ for some $\gamma \in \mathbb{R} - \{0\}\}$. Observe now that a point $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ leads to the largest displacement β_0 when $\mathcal{P} \cap \mathcal{B}$ is feasible, $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}}) \in \mathcal{P} \cap \mathcal{B}$, and $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ achieves the rhs bound in (51). For all other feasible points $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$, there exists a point $((1-\beta_0)\tilde{\boldsymbol{H}}, (1+\beta_0)\tilde{\boldsymbol{M}})$ for a smaller $\beta_0 \in \mathbb{R}$ such that (51) is satisfied. Use this fact to conclude that in the worst case (largest β_0) solution point $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ must be equal to $(\gamma\tilde{\boldsymbol{H}}, \frac{1}{\gamma}\tilde{\boldsymbol{M}})$ for some $\gamma \in \mathbb{R} - \{0\}$. In general, the Euclidean distance between a point $(\gamma\tilde{\boldsymbol{H}}, \frac{1}{\gamma}\tilde{\boldsymbol{M}}) \in \mathcal{P} \cap \mathcal{B}$ and its orthogonal projection $((1-\beta)\tilde{\boldsymbol{H}}, (1+\beta)\tilde{\boldsymbol{M}})$ onto the line $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}}) \oplus \mathcal{N}$ is given by

$$\|(\gamma\tilde{\boldsymbol{H}}, \tfrac{1}{\gamma}\tilde{\boldsymbol{M}}) - ((1-\beta)\tilde{\boldsymbol{H}}, (1+\beta)\tilde{\boldsymbol{M}})\|_F^2 = \left[\left(\frac{2\beta + \sqrt{\beta^2+2}}{\sqrt{2}} - 1\right)^2 + \left(\frac{-2\beta + \sqrt{\beta^2+2}}{\sqrt{2}} - 1\right)^2\right] \|(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F^2. \tag{53}$$

In light of (51), we then have

$$\left[\left(\frac{2\beta_0 + \sqrt{\beta_0^2+2}}{\sqrt{2}} - 1\right)^2 + \left(\frac{-2\beta_0 + \sqrt{\beta_0^2+2}}{\sqrt{2}} - 1\right)^2\right] \|(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F^2 \leq 18^2\|\boldsymbol{\xi}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F$$
$$\implies 4\beta_0^2\|(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F^2 \leq 18^2\|\boldsymbol{\xi}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F,$$

where the implication follows by using the fact that

$$\left(\frac{2\beta + \sqrt{\beta^2+2}}{\sqrt{2}} - 1\right)^2 + \left(\frac{-2\beta + \sqrt{\beta^2+2}}{\sqrt{2}} - 1\right)^2 \geq 4\beta^2$$

holds for any $\beta \in \mathbb{R}$. No $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ can achieve a larger displacement β_0. This allows us to then conclude that $(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}})$ cannot deviate too far not only from the line of linearized scaling ambiguity but also from the ground truth using a simple triangle inequality and (51) as follows

$$\|(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}}) - (\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F - \beta_0\|(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F \leq 18\sqrt{\|\boldsymbol{\xi}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F}$$
$$\|(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}}) - (\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F^2 \leq 405\|\boldsymbol{\xi}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F.$$

Case 2: We now consider general case of noise when $\boldsymbol{\xi}$ is such that $\xi_\ell \geq -1$.

The key idea is that the measurements with noise as in Case 2 can be converted to the measurements with noise as in Case 1. To see this, define

$$s := \max_{\ell \in [m]} \frac{y_\ell}{\tilde{y}_\ell} = 1 + \|\boldsymbol{\xi}\|_\infty, \;\; \eta_\ell := \frac{1}{s}(1 - s + \xi_\ell).$$

From the definitions above, it can be easily verified that $s = 1 + \|\boldsymbol{\xi}\|_\infty$, and that $(1+\xi_\ell) = s(1+\eta_\ell)$. Using this relation allows us to rewrite measurements $y_\ell = \tilde{y}_\ell(1+\xi_\ell)$ contaminated with noise ξ_ℓ equivalently as $y_\ell = s\tilde{y}_\ell(1+\eta_\ell)$, where η_ℓ is now interpreted as noise, and the new scaled noiseless measurements are interpreted as $s y_\ell$. We will now show that $\eta_\ell \in [-1, 0]$.

By definition, $s \geq 1 + \xi_\ell$, this implies that $\eta_\ell \leq (1/s)(s - s) = 0$. Also note that $\eta_\ell = 0$ for a ξ_ℓ that achieves maximum $\|\boldsymbol{\xi}\|_\infty$. Moreover, using the definition of η_ℓ, and the fact that $\xi_\ell \geq -1$ gives $\eta_\ell \geq -1$.

Firstly, the new noise η_ℓ obeys all the conditions in Case 1 above. Secondly, since the noiseless measurements $\tilde{y}_\ell^2$ of $(\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})$ are $\tilde{y}_\ell^2 = \frac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \tilde{\boldsymbol{H}}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \tilde{\boldsymbol{M}}\rangle$, we can interpret $s\tilde{y}_\ell^2$ as the noiseless measurements of $(\sqrt{s}\tilde{\boldsymbol{H}}, \sqrt{s}\tilde{\boldsymbol{M}})$. We can now directly invoke result of Case 1 here to obtain

$$\|(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}}) - (\sqrt{s}\tilde{\boldsymbol{H}}, \sqrt{s}\tilde{\boldsymbol{M}})\|_F^2 = \|\widehat{\boldsymbol{H}} - \sqrt{s}\tilde{\boldsymbol{H}}\|_F^2 + \|\widehat{\boldsymbol{M}} - \sqrt{s}\tilde{\boldsymbol{M}}\|_F^2$$
$$\leq 405\|\boldsymbol{\eta}\|_\infty^2\|\sqrt{s}\tilde{\boldsymbol{H}}\|_F\|\sqrt{s}\tilde{\boldsymbol{M}}\|_F \leq 405 s\|\boldsymbol{\eta}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F \leq 810\|\boldsymbol{\xi}\|_\infty\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F, \tag{54}$$

where the last inequality is obtained using the inequality derived below

$$s\eta_\ell = 1 - s + \xi_\ell = \xi_\ell - \|\boldsymbol{\xi}\|_\infty \implies s|\eta_\ell| = \|\boldsymbol{\xi}\|_\infty - \xi_\ell$$
$$\implies s\|\boldsymbol{\eta}\|_\infty = \|\boldsymbol{\xi}\|_\infty - \min_{\ell \in [m]} \xi_\ell \leq 2\|\boldsymbol{\xi}\|_\infty.$$

Observe that

$$\|(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{M}}) - (\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{M}})\|_F = \sqrt{\|\widehat{\boldsymbol{H}} - \tilde{\boldsymbol{H}}\|_F^2 + \|\widehat{\boldsymbol{M}} - \tilde{\boldsymbol{M}}\|_F^2}$$
$$= \sqrt{\|\widehat{\boldsymbol{H}} - \sqrt{s}\tilde{\boldsymbol{H}} + \sqrt{s}\tilde{\boldsymbol{H}} - \tilde{\boldsymbol{H}}\|_F^2 + \|\widehat{\boldsymbol{M}} - \sqrt{s}\tilde{\boldsymbol{M}} + \sqrt{s}\tilde{\boldsymbol{M}} - \tilde{\boldsymbol{M}}\|_F^2}$$
$$\leq 2\left(\sqrt{\|\widehat{\boldsymbol{H}} - \sqrt{s}\tilde{\boldsymbol{H}}\|_F^2 + \|\widehat{\boldsymbol{M}} - \sqrt{s}\tilde{\boldsymbol{M}}\|_F^2} + (\sqrt{s} - 1)\sqrt{\|\tilde{\boldsymbol{H}}\|_F^2 + \|\tilde{\boldsymbol{M}}\|_F^2}\right)$$
$$\leq 2\sqrt{810}\sqrt{\|\boldsymbol{\xi}\|_\infty}\sqrt{\|\tilde{\boldsymbol{H}}\|_F\|\tilde{\boldsymbol{M}}\|_F} + 2(\sqrt{s} - 1)\sqrt{\|\tilde{\boldsymbol{H}}\|_F^2 + \|\tilde{\boldsymbol{M}}\|_F^2}$$
$$\leq 44\sqrt{\|\boldsymbol{\xi}\|_\infty}\sqrt{\|\tilde{\boldsymbol{H}}\|_F^2 + \|\tilde{\boldsymbol{M}}\|_F^2},$$

where the second last inequality follows by using (54). The proof is complete.

Appendix A. Proof of Lemma 1

The objective of (8) is simply linear, we focus on the constraints. For a fixed ℓ, let $S_\ell := \{(\boldsymbol{H}, \boldsymbol{M}) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{m\times m} \mid \frac{1}{m}\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle\langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle \geq \tilde{y}_\ell^2, \boldsymbol{H} \succcurlyeq \boldsymbol{0}, \boldsymbol{M} \succcurlyeq \boldsymbol{0}\}$, $S_{\ell,1} := \{(u_\ell, v_\ell) \in \mathbb{R}^2 \mid \frac{1}{m}u_\ell v_\ell \geq \tilde{y}_\ell^2, u_\ell \geq 0, v_\ell \geq 0\}$, and

$$S_{\ell,2} := \{(\boldsymbol{H}, \boldsymbol{M}) \in \mathcal{H}^{k\times k} \times \mathcal{H}^{m\times m} \mid (\langle \boldsymbol{b}_\ell\boldsymbol{b}_\ell^*, \boldsymbol{H}\rangle, \langle \boldsymbol{c}_\ell\boldsymbol{c}_\ell^*, \boldsymbol{M}\rangle) \in S_{\ell,1}\}.$$

To show that S_ℓ is convex, it suffices to show that $S_{\ell,1}$, and $S_{\ell,2}$ are convex.

Fix $(u_1, v_1), (u_2, v_2) \in S_{\ell,1}$, and let $\alpha \in [0, 1]$. Note that $u_1 > 0$, and $u_2 > 0$ as $y_\ell > 0$. Consider

$$\begin{aligned}
&\frac{1}{m}(\alpha u_1 + (1-\alpha)u_2)(\alpha v_1 + (1-\alpha)v_2) \\
&\quad = \frac{1}{m}\left((\alpha^2 u_1 v_1 + (1-\alpha)^2 u_2 v_2) + \alpha(1-\alpha)(u_1 v_2 + u_2 v_1)\right) \\
&\quad \geq (\alpha^2 \tilde{y}_\ell^2 + (1-\alpha)^2 \tilde{y}_\ell^2) + \alpha(1-\alpha)(\frac{\tilde{y}_\ell^2 u_1}{u_2} + \frac{\tilde{y}_\ell^2 u_2}{u_1}) \\
&\quad = \tilde{y}_\ell^2 \left(1 + \frac{2\alpha^2 u_1 u_2 - 2\alpha u_1 u_2 + \alpha(1-\alpha)(u_1^2 + u_2^2)}{u_1 u_2}\right) \\
&\quad = \tilde{y}_\ell^2 \left(1 + \frac{(\alpha - \alpha^2)(u_1 - u_2)^2}{u_1 u_2}\right) \geq \tilde{y}_\ell^2,
\end{aligned}$$

where the last inequality follows form the fact that $\alpha \in [0, 1]$, and $u_1 u_2 > 0$. This shows that $S_{\ell,1}$ is convex.

The set $S_{\ell,2}$ is convex as the inverse image of a convex set of a linear map is convex. This implies that S_ℓ is convex. Finally, the intersection of any number of convex sets is convex means that the constraint of (8) is convex. This proves that (8) is a convex program.

Acknowledgments

Ali Ahmed acknowledges support from NRPU 6856, HEC. Paul Hand acknowledges support from NSF DMS 1464525.

References

Alireza Aghasi, Ali Ahmed, and Paul Hand. Branchhull: Convex bilinear inversion from the entrywise product of signals with known signs. *arXiv preprint arXiv:1702.04342*, 2017a.

Alireza Aghasi, Ali Ahmed, and Paul Hand. Convex inversion of the entrywise product of real signals with known signs. In *Signals, Systems, and Computers, 2017 51st Asilomar Conference on*, pages 1622–1626. IEEE, 2017b.

Alireza Aghasi, Ali Ahmed, Paul Hand, and Babhru Joshi. A convex program for bilinear inversion of sparse vectors. In *Advances in Neural Information Processing Systems*, pages 8557–8567, 2018.

Ali Ahmed, Benjamin Recht, and Justin Romberg. Blind deconvolution using convex programming. *IEEE Transactions on Information Theory*, 60(3):1711–1732, 2014.

Ali Ahmed, Alireza Aghasi, and Paul Hand. Blind deconvolutional phase retrieval via convex programming. In *Advances in Neural Information Processing Systems 31*, pages 10051–10061. Curran Associates, Inc., 2018.

Ahmad Helmi Azhar, Tuan-Anh Tran, and Dominic O'Brien. Demonstration of high-speed data transmission using mimo-ofdm visible light communications. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pages 1052–1056. IEEE, 2010.

Ahmad Helmi Azhar, Thomas Tran, and Dominic O'Brien. A gigabit/s indoor wireless transmission using mimo-ofdm visible-light communications. *IEEE photonics technology letters*, 25(2):171–174, 2013.

Sohail Bahmani and Justin Romberg. Anchored regression: Solving random convex equations via convex programming. *arXiv preprint arXiv:1702.05327*, 2017a.

Sohail Bahmani and Justin Romberg. Phase Retrieval Meets Statistical Learning Theory: A Flexible Convex Relaxation. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 252–260, Fort Lauderdale, FL, USA, 20–22 Apr 2017b. PMLR. URL `http://proceedings.mlr.press/v54/bahmani17a.html`.

Srinadh Bhojanapalli, Nicolas Boumal, Prateek Jain, and Praneeth Netrapalli. Smoothed analysis for low-rank solutions to semidefinite programs in quadratic penalty form. *arXiv preprint arXiv:1803.00186*, 2018.

Oliver Bunk, Ana Diaz, Franz Pfeiffer, Christian David, Bernd Schmitt, Dillip K Satapathy, and J Friso van der Veen. Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels. *Acta Crystallographica Section A: Foundations of Crystallography*, 63(4):306–314, 2007.

Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2): 329–357, 2003.

Samuel Burer and Renato DC Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

Emmanuel J Candes, Thomas Strohmer, and Vladislav Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251, 2015a.

Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval from coded diffraction patterns. *Applied and Computational Harmonic Analysis*, 39(2):277–299, 2015b.

Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

Veit Elser, Ti-Yen Lan, and Tamir Bendory. Benchmark problems for phase retrieval. *arXiv preprint arXiv:1706.00399*, 2017.

C Fienup and J Dainty. Phase retrieval and image reconstruction for astronomy. *Image Recovery: Theory and Application*, 231:275, 1987.

Tom Goldstein and Christoph Studer. Phasemax: Convex phase retrieval via basis pursuit. *IEEE Transactions on Information Theory*, 2018.

Joseph Goodman. Introduction to fourier optics. 2008.

Robert W Harrison. Phase problem in crystallography. *JOSA a*, 10(5):1046–1055, 1993.

Vladimir Koltchinskii and Shahar Mendelson. Bounding the smallest singular value of a random matrix without concentration. *International Mathematics Research Notices*, 2015 (23):12991–13008, 2015.

Han-Wen Kuo, Yenson Lau, Yuqian Zhang, and John Wright. Geometry and symmetry in short-and-sparse deconvolution. *arXiv preprint arXiv:1901.00256*, 2019.

Guillaume Lecué and Shahar Mendelson. Regularization and the small-ball method ii: complexity dependent error rates. *The Journal of Machine Learning Research*, 18(1): 5356–5403, 2017.

Guillaume Lecué, Shahar Mendelson, et al. Regularization and the small-ball method i: sparse recovery. *The Annals of Statistics*, 46(2):611–641, 2018.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1): 148–188, 1989.

Shahar Mendelson. Learning without concentration. In *Conference on Learning Theory*, pages 25–39, 2014.

Jianwei Miao, Tetsuya Ishikawa, Qun Shen, and Thomas Earnest. Extending x-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes. *Annu. Rev. Phys. Chem.*, 59:387–410, 2008.

Rick P Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.

José Ramón Durán Retamal, Hassan Makine Oubei, Bilal Janjua, Yu-Chieh Chi, Huai-Yung Wang, Cheng-Ting Tsai, Tien Khee Ng, Dan-Hua Hsieh, Hao-Chung Kuo, Mohamed-Slim Alouini, et al. 4-gbit/s visible light communication link based on 16-qam ofdm transmission over remote phosphor-film converted white light by using blue laser diode. *Optics express*, 23(26):33656–33666, 2015.

Mark Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab. *Software available at http://www. cs. ubc. ca/˜ schmidtm/Software/minFunc. htm*, 2005.

Joel A Tropp. Convex recovery of a structured signal from independent random linear measurements. In *Sampling Theory, a Renaissance*, pages 67–101. Springer, 2015.

Sara van de Geer and Johannes Lederer. The bernstein–orlicz norm and deviation inequalities. *Probability theory and related fields*, 157(1-2):225–250, 2013.

Aad W van der Vaart and Jon A Wellner. Weak convergence and empirical processes with applications to statistics. *Journal of the Royal Statistical Society-Series A Statistics in Society*, 160(3):596–608, 1997.

Yuqian Zhang, Han-Wen Kuo, and John Wright. Structured local optima in sparse blind deconvolution. *arXiv preprint arXiv:1806.00338*, 2018.

Journal of Machine Learning Research 20 (2019) 1-7 Submitted 6/19; Revised 8/19; Published 9/19

GraSPy: Graph Statistics in Python

Jaewon Chung[1,†] J1C@JHU.EDU
Benjamin D. Pedigo[1,†] BPEDIGO@JHU.EDU
Eric W. Bridgeford[2] EBRIDGE2@JHU.EDU
Bijan K. Varjavand[1] BVARJAV1@JHU.EDU
Hayden S. Helm[3] HH@JHU.EDU
Joshua T. Vogelstein[1,3,4,⋆] JOVO@JHU.EDU

[1] *Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218*
[2] *Department of Biostatistics, Johns Hopkins University, Baltimore, MD 21218*
[3] *Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218*
[4] *Kavli Neuroscience Discovery Institute, Institute for Computational Medicine, Johns Hopkins University, Baltimore, MD 21218*
[†] *Denotes equal contribution*
[⋆] *Corresponding author*

Editor: Alexandre Gramfort

Abstract

We introduce `GraSPy`, a Python library devoted to statistical inference, machine learning, and visualization of random graphs and graph populations. This package provides flexible and easy-to-use algorithms for analyzing and understanding graphs with a `scikit-learn` compliant API. `GraSPy` can be downloaded from Python Package Index (PyPi), and is released under the Apache 2.0 open-source license. The documentation and all releases are available at `https://neurodata.io/graspy`.

Keywords: Python, graph analysis, network analysis, statistical inference, machine learning

1. Introduction

Graphs, or networks, are a mathematical representation of data that consists of discrete objects (nodes or vertices) and relationships between these objects (edges). For example, a brain can be represented with nodes corresponding to each brain region and edges representing the presence of a structural connection between regions (Vogelstein et al., 2019). Since graphs necessarily deal with relationships between nodes, classical statistical assumptions about independence are violated. Thus, novel methodology is required for performing statistical inference on graphs and populations of graphs (Athreya et al., 2018). While the theory for inference on graphs is highly developed, to date, there has not existed a numerical package implementing these methods. `GraSPy` fills this gap by providing implementations of algorithms with strong statistical guarantees, such as graph and multi-graph embedding methods, two-graph hypothesis testing, and clustering of vertices of graphs. Many of the algorithms implemented in `GraSPy` are flexible and can operate on graphs that are weighted or unweighted, as well as directed or undirected. Table 1 provides a comparison of `GraSPy`

©2019 Jaewon Chung, Benjamin D. Pedigo, Eric W. Bridgeford, Bijan K. Varjavand, Hayden S. Helm, Joshua T. Vogelstein.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/19-490.html`.

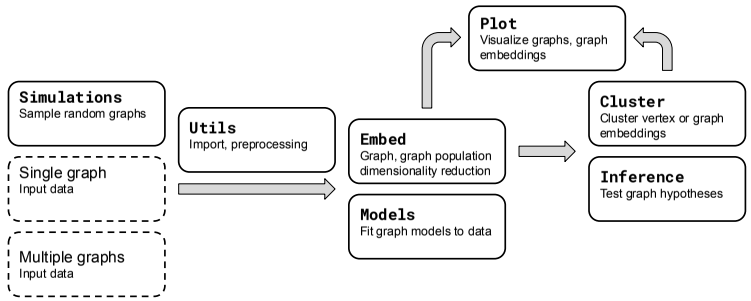


Figure 1: Illustration of modules and procedure for statistical inference on graphs, populations of graphs, or simulated data. A detailed description of each module is given in Section 2.

to other existing graph analysis packages (Hagberg et al., 2008; Peixoto, 2014; Leskovec and Sosič, 2016).

2. Library Overview

`GraSPy` includes functionality for fitting and sampling from random graph models, performing dimensionality reduction on graphs or populations of graphs (embedding), testing hypotheses on graphs, and plotting of graphs and embeddings. The following provides brief overview of different modules of `GraSPy`. An example workflow using these modules is shown in Figure 1. More detailed overview and code usage can be found in the tutorial section of `GraSPy` documentation at `https://graspy.neurodata.io/tutorial`. All descriptions here correspond to `GraSPy` version 0.1.1.

Simulations Several random graph models are implemented in `GraSPy`, including the Erdős-Rényi (ER) model, stochastic block model (SBM), degree-corrected Erdős-Rényi (DCER) model, degree-corrected stochastic block model (DCSBM), and random dot product graph (RDPG) (Holland et al., 1983; Karrer and Newman, 2011; Young and Scheinerman, 2007). The simulations module allows the user to sample random graphs given the parameters of one of these models. Additionally, the user can specify a distribution on the weights of graph edges.

Utils `GraSPy` includes a variety of utility functions for graph and graph population importing and preprocessing. `GraSPy` can import graphs represented as `NetworkX` objects or NumPy arrays. Preprocessing examples include finding the largest connected component of a graph, finding the intersection or union of connected components across multiple graphs, or checking whether a graph is directed.

Embed Inferences on random graphs can leverage low-dimensional Euclidean representations of the vertices, known as *latent positions*. Adjacency spectral embedding (ASE)

	Graph Theory			Statistical Modeling					Other	
package	*traversal*	*network stats.*	*communities*	*embed*	*mult. embed*	*model fit*	*simulations*	*hyp. test*	*plotting*	*pip install*
`GraSPy` 0.1.1	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
`NetworkX` 2.3	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓
`graph-tool` 2.29	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗
`Snap.py` 4.1	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗

Table 1: Qualitative comparison of Python graph analysis packages. `GraSPy` is largely complementary to existing graph analysis packages in Python. `GraSPy` does not implement many of the essential algorithms for operating on graphs (rather, it leverages `NetworkX` for these implementations). The focus of `GraSPy` is on statistical modeling of populations of networks, with features such as multiple graph embeddings, model fitting, and hypothesis testing. A ✓ is given for packages that implement the respective feature, a ✓ for packages that partially implement the respective feature, and a ✗ is given for packages that do not implement the respective feature. Note that while a ✓ shows that the feature exists in the corresponding package, it does not imply that the specific algorithms are the same for all packages.

and Laplacian spectral embedding (LSE) are methods for embedding a single graph (Athreya et al., 2018). Omnibus embedding and multiple adjacency spectral embedding (MASE) allows for embedding multiple graphs into the same subspace such that the embeddings can be meaningfully compared (Levin et al., 2017; Arroyo et al., 2019). `GraSPy` includes a method for choosing the number of embedding dimensions automatically (Zhu and Ghodsi, 2006).

Models `GraSPy` includes classes for fitting random graph models to an input graph (Figure 2). Currently, ER, SBM, DCER, DCSBM, and RDPG are supported for model estimation. After fitting a model to data, the model class can also output goodness-of-fit metrics (mean squared error, likelihood) and the number of estimated model parameters, allowing for model selection. The model classes can also be used to sample new simulated graphs based on the fit model.

Inference Given two graphs, a natural question to ask is whether these graphs are both random samples from the same generative distribution. `GraSPy` provides two types of test for this null hypothesis: a latent position test and a latent distribution test. Both tests are framed under the RDPG model, where the generative distribution for the graph can be modeled as a set of latent positions. The latent position test can only be performed on two graphs of the same size and with known correspondence between the vertices of the two graphs (Tang et al., 2017). The latent distribution test

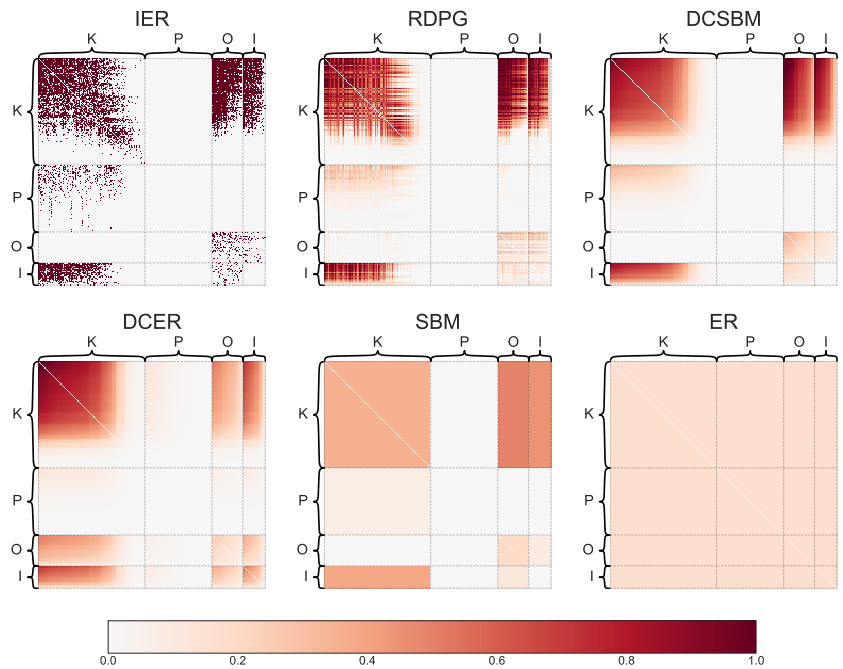


Figure 2: Connectome model fitting using `GraSPy`. Heatmaps show the probability of potential edges for models of graphs fit to the *Drosophila* larva right mushroom body connectome (unweighted, directed) (Eichler et al., 2017). The node labels correspond to cell types: P) projection neurons, O) mushroom body output neurons, I) mushroom body input neurons. The graph models are: inhomogeneous Erdős-Rényi (IER) model in which all potential edges are specified, random dot product graph (RDPG), degree-corrected stochastic block model (DCSBM), degree-corrected Erdős-Rényi (DCER), stochastic block model (SBM), and Erdős-Rényi (ER). Blocks (cell types) are sorted by number of member vertices and nodes are sorted by degree within blocks. The code used to generate the figure is a tutorial section at `https://neurodata.io/graspy`.

can be performed on graphs without vertex alignment, or even with slightly different numbers of vertices (Tang et al., 2014).

Cluster `GraSPy` extends Gaussian mixture models (GMM) and k-means from `scikit-learn` to sweep over a specified range of parameters and choose the best clustering (Pedregosa et al., 2011). The number of clusters and covariance structure for each GMM is chosen by Bayesian information criterion (BIC), which is a penalized likelihood function

to evaluate the quality of estimators (Schwarz et al., 1978). Silhouette score is used to choose the number of clusters for k-means (Rousseeuw, 1987). Clustering is often useful for computing the the community structure of vertices after embedding.

Plot `GraSPy` extends `seaborn` to visualize graphs as adjacency matrices and embedded graphs as paired scatter plots (Waskom et al., 2018). Individual graphs can be visualized using heatmap function, and multiple graphs can be overlaid on top of each other using gridplot. The nodes in both graph visualizations can be sorted by various node metadata, such as node degree or assigned node labels. Pairplot can visualize high dimensional data, such as embeddings, as a pairwise scatter plot.

3. Code example

Given the connectomes of the *Drosophila* larva left and right mushroom bodies, one natural question to ask is: how similar are these graphs (Eichler et al., 2017)? We can frame this question as whether these graphs are generated from the same distribution of latent positions (Tang et al., 2014). We can use the latent distribution test to test this hypothesis:

```
from graspy.datasets import load_drosophila_left, load_drosophila_right
from graspy.inference import LatentDistributionTest

# Load data
left_graph = load_drosophila_left()
right_graph = load_drosophila_right()

# Initialize hypothesis test object and compute p-value
ldt = LatentDistributionTest(n_components=3, n_bootstraps=500)
p_value = ldt.fit(left_graph, right_graph)
print("p-value: " + str(p_value))
p-value: 0.002
```

4. Conclusion

`GraSPy` is an open-source Python package to perform statistical analysis on graphs and graph populations. Its compliance with the `scikit-learn` API makes it an easy-to-use tool for anyone familiar with machine learning in Python (Buitinck et al., 2013). In addition, `GraSPy` is implemented with an extensible class structure, making it easy to modify and add new algorithms to the package. As `GraSPy` continues to grow and add functionality, we believe it will accelerate statistically principled discovery in any field of study concerned with graphs or populations of graphs.

Acknowledgments

This work is graciously supported by the DARPA, under agreement numbers FA8650-18-2-7834 and FA8750-17-2-0112. We thank all the contributors for assisting with writing `GraSPy`. We thank the NeuroData Design class, the NeuroData lab, and Carey E. Priebe at Johns Hopkins University for helpful feedback.

References

Jesús Arroyo, Avanti Athreya, Joshua Cape, Guodong Chen, Carey E Priebe, and Joshua T Vogelstein. Inference for multiple heterogeneous networks with a common invariant subspace. *arXiv preprint arXiv:1906.10026*, 2019.

Avanti Athreya, Donniell E. Fishkind, Minh Tang, Carey E. Priebe, Youngser Park, Joshua T. Vogelstein, Keith Levin, Vince Lyzinski, Yichen Qin, and Daniel L Sussman. Statistical inference on random dot product graphs: a survey. *Journal of Machine Learning Research*, 18(226):1–92, 2018. URL `http://jmlr.org/papers/v18/17-448.html`.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Katharina Eichler, Feng Li, Ashok Litwin-Kumar, Youngser Park, Ingrid Andrade, Casey M Schneider-Mizell, Timo Saumweber, Annina Huser, Claire Eschbach, Bertram Gerber, et al. The complete connectome of a learning and memory centre in an insect brain. *Nature*, 548(7666):175, 2017.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

Keith Levin, Avanti Athreya, Minh Tang, Vince Lyzinski, and Carey E Priebe. A central limit theorem for an omnibus embedding of multiple random dot product graphs. pages 964–967, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014. doi: 10.6084/m9.figshare.1164194. URL `http://figshare.com/articles/graph_tool/1164194`.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2): 461–464, 1978.

Minh Tang, Avanti Athreya, Daniel L. Sussman, Vince Lyzinski, and Carey E. Priebe. A nonparametric two-sample hypothesis testing problem for random dot product graphs. *Journal of Computational and Graphical Statistics*, art. arXiv:1409.2344, September 2014.

Minh Tang, Avanti Athreya, Daniel L Sussman, Vince Lyzinski, Youngser Park, and Carey E Priebe. A semiparametric two-sample hypothesis testing problem for random graphs. *Journal of Computational and Graphical Statistics*, 26(2):344–354, 2017.

Joshua T Vogelstein, Eric W Bridgeford, Benjamin D Pedigo, Jaewon Chung, Keith Levin, Brett Mensh, and Carey E Priebe. Connectal coding: discovering the structures linking cognitive phenotypes to individual histories. *Current Opinion in Neurobiology*, 55:199–212, 2019.

Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Joel Ostblom, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Thomas Brunner, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, and Adel Qalieh. mwaskom/seaborn: v0.9.0 (july 2018), July 2018. URL `https://doi.org/10.5281/zenodo.1313201`.

Stephen J Young and Edward R Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer, 2007.

Mu Zhu and Ali Ghodsi. Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51(2):918–930, 2006.

Journal of Machine Learning Research 20 (2019) 1-31 Submitted 6/19; Revised 9/19; Published 9/19

Optimal Convergence Rates for Convex Distributed Optimization in Networks

Kevin Scaman KEVIN.SCAMAN@HUAWEI.COM
Huawei Noah's Ark Lab, Paris, France

Francis Bach FRANCIS.BACH@INRIA.FR
INRIA, Ecole Normale Supérieure, PSL Research University, Paris, France

Sébastien Bubeck SEBUBECK@MICROSOFT.COM
Microsoft Research, Redmond, United States

Yin Tat Lee YILE@MICROSOFT.COM
University of Washington, Seattle, United States

Laurent Massoulié LAURENT.MASSOULIE@INRIA.FR
MSR-INRIA Joint Center, Paris, France

Editor: Kilian Weinberger

Abstract

This work proposes a theoretical analysis of distributed optimization of convex functions using a network of computing units. We investigate this problem under two communication schemes (centralized and decentralized) and four classical regularity assumptions: Lipschitz continuity, strong convexity, smoothness, and a combination of strong convexity and smoothness. Under the decentralized communication scheme, we provide matching upper and lower bounds of complexity along with algorithms achieving this rate up to logarithmic constants. For non-smooth objective functions, while the dominant term of the error is in $O(1/\sqrt{t})$, the structure of the communication network only impacts a second-order term in $O(1/t)$, where t is time. In other words, the error due to limits in communication resources decreases at a fast rate even in the case of non-strongly convex objective functions. Such a convergence rate is achieved by the novel *multi-step primal-dual* (MSPD) algorithm. Under the centralized communication scheme, we show that the naive distribution of standard optimization algorithms is optimal for smooth objective functions, and provide a simple yet efficient algorithm called *distributed randomized smoothing* (DRS) based on a local smoothing of the objective function for non-smooth functions. We then show that DRS is within a $d^{1/4}$ multiplicative factor of the optimal convergence rate, where d is the underlying dimension.

Keywords: distributed optimization, convex optimization, first-order methods

1. Introduction

Distributed optimization finds many applications in machine learning, for example when the data set is large and training is achieved using a cluster of computing units. As a result, many algorithms were recently introduced to minimize the average $\bar{f} = \frac{1}{n}\sum_{i=1}^n f_i$ of functions f_i, $i = 1, \dots, n$, which are respectively accessible by separate nodes in a network (Nedic and Ozdaglar, 2009; Boyd et al., 2011; Duchi et al., 2012a; Shi et al., 2015). Most

©2019 Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee and Laurent Massoulié.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-543.html.

often, these algorithms alternate between local and incremental improvement steps (such as gradient steps) with communication steps between nodes in the network, and come with a variety of convergence rates (see, for example, Shi et al., 2014, 2015; Jakovetić et al., 2015; Nedic et al., 2017).

The main contribution of this paper is to provide optimal convergence rates and their corresponding optimal algorithms for distributed convex problems under a large panel of regularity assumptions and communication schemes. More specifically, we consider two communication schemes and four regularity assumptions. Communication is either achieved through a *master/slave* approach that we refer to as *centralized*, or in a *decentralized* scheme in which communication is performed using the *gossip* algorithm (Boyd et al., 2006). Moreover, the four regularity assumptions that we investigate in this paper are: Lipschitz continuity, strong convexity, smoothness, and both strong convexity and smoothness. These settings are summarized in Table 1 along with their corresponding upper and lower complexity bounds. These bounds depend on the time τ necessary to communicate a vector between two neighbors in the communication graph, the diameter of the communication graph Δ, its *mixing time* $\widetilde{\Delta}$ (see Section 4), the dimension of the optimization problem d and distance to optimum R, and the Lipschitz, strong convexity, smoothness constants, and condition number of the objective function (L_g, α_g, β_g and κ_g, respectively, for the *global* objective function, and L_ℓ, α_ℓ, β_ℓ and κ_ℓ for an aggregation of the *local* individual functions of each computing unit, see Section 2).

Under decentralized communication, we provide in Section 4 matching upper and lower bounds of complexity.[1] Moreover, we propose the first optimal algorithm for non-smooth decentralized optimization, called *multi-step primal-dual* (MSPD). Under centralized communication, we show in Section 3 that, for smooth objectives, a naïve distribution of accelerated gradient descent is optimal, while for non-smooth objectives, distributing the simple smoothing approach introduced by Duchi et al. (2012b) yields fast convergence rates with respect to communication. This algorithm, called *distributed randomized smoothing* (DRS), achieves a convergence rate matching the lower bound up to a $d^{1/4}$ multiplicative factor, where d is the dimensionality of the problem.

The analysis of non-smooth optimization differs from the smooth setting in two major aspects: (1) the naïve *master/slave* distributed algorithm is in this case not optimal, and (2) the convergence rates differ between communication and local computations. More specifically, errors due to limits in communication resources enjoy fast convergence rates, as we establish by formulating the optimization problem as a composite saddle-point problem with a smooth term for communication and non-smooth term for the optimization of the local functions (see Section 4 and Equation 11 for more details).

1.1. Related Work

Many algorithms were proposed to solve the decentralized optimization of an average of functions, from early decentralized algorithms (Nedic and Ozdaglar, 2009; Boyd et al., 2011; Duchi et al., 2012a; Wei and Ozdaglar, 2012; Jakovetić et al., 2014; Shi et al., 2014, 2015;

1. The lower bounds for decentralized communication for strongly convex, smooth, and both strongly convex and smooth settings, are met up to logarithmic factors using the PSTM and R-Sliding algorithms of Dvinskikh and Gasnikov (2019).

Setting	Lower bound	Upper bound	Algorithm
Lipschitz continuity + convexity			
Centr.	$\Omega\left(\left(\frac{L_g R}{\varepsilon}\right)^2 + \frac{L_g R}{\varepsilon}\Delta\tau\right)$	$O\left(\left(\frac{L_g R}{\varepsilon}\right)^2 + \frac{L_g R}{\varepsilon}\Delta\tau d^{1/4}\right)$	DRS (Alg. 1)
Decentr.	$\Omega\left(\left(\frac{L_\ell R}{\varepsilon}\right)^2 + \frac{L_\ell R}{\varepsilon}\widetilde{\Delta}\tau\right)$	matching lower bound	MSPD (Alg. 3)
Lipschitz continuity + strong convexity			
Centr.	$\Omega\left(\frac{L_g^2}{\alpha_g \varepsilon} + \sqrt{\frac{L_g^2}{\alpha_g \varepsilon}}\Delta\tau\right)$	$O\left(\frac{L_g^2}{\alpha_g \varepsilon} + \sqrt{\frac{L_g^2}{\alpha_g \varepsilon}}\Delta\tau d^{1/4}\log\left(\frac{1}{\varepsilon}\right)\right)$	DRS (Alg. 2)
Decentr.	$\Omega\left(\frac{L_\ell^2}{\alpha_\ell \varepsilon} + \sqrt{\frac{L_\ell^2}{\alpha_\ell \varepsilon}}\widetilde{\Delta}\tau\right)$	matching lower bound[1]	R-Sliding[1]
Smoothness + convexity			
Centr.	$\Omega\left(R\sqrt{\frac{\beta_g}{\varepsilon}}(1+\Delta\tau)\right)$	matching lower bound	AGD
Decentr.	$\Omega\left(R\sqrt{\frac{\beta_\ell}{\varepsilon}}(1+\widetilde{\Delta}\tau)\right)$	matching lower bound[1]	PSTM[1]
Smoothness + strong convexity			
Centr.	$\Omega\left(\sqrt{\kappa_g}\ln\left(\frac{1}{\varepsilon}\right)(1+\Delta\tau)\right)$	matching lower bound	AGD
Decentr.	$\Omega\left(\sqrt{\kappa_\ell}\ln\left(\frac{1}{\varepsilon}\right)(1+\widetilde{\Delta}\tau)\right)$	matching lower bound[1]	PSTM[1]

Table 1: Upper and lower bounds on the minimax optimal time needed to reach an $\varepsilon > 0$ precision for four regularization assumptions and two communication settings, along with corresponding optimal algorithms (AGD stands for the naïve distribution of accelerated gradient descent, see Section 3 for more details).

Mokhtari and Ribeiro, 2016) to more recent improvements (Qu and Li, 2016; Jiang et al., 2017; He et al., 2018; Pu et al., 2018; Uribe et al., 2018; Koloskova et al., 2019; Xin et al., 2019; Jakovetić, 2019). Another line of work considers more restrictive assumptions on the optimization problems to obtain fast convergence rates (Lee et al., 2017; Jesús Arroyo, 2016; Tian and Gu, 2017). A sheer amount of work was devoted to improving the convergence rate of these algorithms (Shi et al., 2014; Jakovetić et al., 2015). In the case of non-smooth optimization, fast communication schemes were developed by Lan et al. (2017); Jaggi et al. (2014), although precise optimal convergence rates were not obtained. Our decentralized algorithm is closely related to the recent primal-dual algorithm of Lan et al. (2017) which enjoys fast communication rates in a decentralized and stochastic setting. Unfortunately, their algorithm lacks gossip acceleration to reach optimality with respect to communication time. Finally, optimal convergence rates for distributed algorithms were investigated by Scaman et al. (2017) for smooth and strongly convex objective functions, Shamir (2014); Arjevani and Shamir (2015) for totally connected networks, and more recently by Dvinskikh and Gasnikov (2019) who derive optimal algorithms (up to logarithmic multiplicative

factors) for decentralized optimization, yet without the corresponding complexity lower bounds. Our analysis proves that these algorithms are indeed, up to logarithmic multiplicative factors, optimal in a minimax sense. A preliminary version of this work with only the analysis for the Lipschitz-continuous case (first line in Table 1) is available as a conference paper (Scaman et al., 2018).

2. Distributed Optimization Setting

In this section, we provide a detailed presentation of the distributed optimization setting considered in this work. The definitions are analogous to that of Scaman et al. (2017).

2.1. Optimization Problem

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected simple (i.e., undirected) graph of n computing units and diameter Δ, each having access to a convex function f_i over a convex set $\mathcal{K} \subset \mathbb{R}^d$. We consider minimizing the average of the local functions

$$\min_{\theta \in \mathcal{K}} \bar{f}(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)\,, \tag{1}$$

in a distributed setting. More specifically, we assume that each computing unit can compute a subgradient $\nabla f_i(\theta)$ of its own function in one unit of time, and communicate values (i.e., vectors in $\mathbb{R}^d$) to its neighbors in $\mathcal{G}$. A *direct* communication along the edge $(i,j) \in \mathcal{E}$ requires a time $\tau \geq 0$. These actions may be performed asynchronously and in parallel, and each machine i possesses a local version of the parameter, which we refer to as $\theta_i \in \mathcal{K}$.

2.2. Regularity Assumptions

Optimal convergence rates depend on the precise set of assumptions applied to the objective function. In our case, we will consider four different regularity assumptions:

1. **Convexity:** a function f is convex if, for all $\theta, \theta' \in \mathcal{K}$ and $a \in [0,1]$,
$$f(a\theta + (1-a)\theta') \leq af(\theta) + (1-a)f(\theta')\,.$$
2. **Strong convexity:** a function f is α-strongly convex if $f(\theta) - \frac{\alpha}{2}\|\theta\|_2^2$ is convex.
3. **Lipschitz continuity:** a function f is L-Lipschitz continuous if, for all $\theta, \theta' \in \mathcal{K}$,
$$|f(\theta) - f(\theta')| \leq L\|\theta - \theta'\|_2\,.$$
4. **Smoothness:** a function f is β-smooth if its gradient ∇f is β-Lipschitz continuous.

Moreover, when a function is both α-strongly convex and β-smooth, we denote by $\kappa = \beta/\alpha$ its condition number. In our analysis, we will denote by L_g, α_g, β_g and κ_g the characteristics of the (global) function $\bar{f}$, while, for $i \in [\![1,n]\!]$, we will refer to L_i, α_i, β_i and κ_i for characteristics of each (local) function f_i. Finally, for the sake of clarity, we will aggregate

these local characteristics into the following values: $L_\ell = \sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2}$, $\alpha_\ell = \min_i \alpha_i$, $\beta_\ell = \max_i \beta_i$, and $\kappa_\ell = \beta_\ell/\alpha_\ell$.

Global regularity is always *weaker* than local regularity, as we always have $L_g \leq L_\ell$, $\alpha_g \geq \alpha_\ell$, $\beta_g \leq \beta_\ell$ and $\kappa_g \leq \kappa_\ell$. Moreover, we may have $L_g = 0$ and L_ℓ arbitrarily large, for example with two linear functions $f_1(x) = -f_2(x) = ax$ and $a \to +\infty$. We will see in the following sections that the local regularity assumption is easier to analyze and leads to matching upper and lower bounds. For the global regularity assumption, we only provide algorithms with a $d^{1/4}$ competitive ratio, where d is the dimension of the problem. Finding an optimal distributed algorithm for global regularity is, to our understanding, a much more challenging task and is left for future work.

Finally, we assume that the feasible region $\mathcal{K}$ is convex and bounded, and denote by R the radius of a ball containing $\mathcal{K}$, i.e.,

$$\forall \theta \in \mathcal{K}, \;\; \|\theta - \theta_0\|_2 \leq R\,,$$

where $\theta_0 \in \mathcal{K}$ is the initial value of the algorithm, that we set to $\theta_0 = 0$ without loss of generality.

2.3. Black-box Optimization Procedure

The lower bounds provided hereafter depend on the notion of black-box optimization procedures for the problem in Equation 1. The definition is analogous to that of Scaman et al. (2017), except that gradients of the Fenchel conjugate of local functions are considered too expensive to compute. This allows us to focus on primal methods which are generally easier to apply (less assumptions required on the objective function) and faster to compute (the Fenchel conjugate is, except for very specific functions, hard to approximate). A black-box optimization procedure is a distributed algorithm verifying the following constraints:

1. **Local memory:** each node i can store past values in a (finite) internal memory $\mathcal{M}_{i,t} \subset \mathbb{R}^d$ at time $t \geq 0$. These values can be accessed and used at time t by the algorithm run by node i, and are updated either by local computation or by communication (defined below), that is, for all $i \in [\![1, n]\!]$,

$$\mathcal{M}_{i,t} \subset \mathcal{M}_{i,t}^{\text{comp}} \cup \mathcal{M}_{i,t}^{\text{comm}}.$$

2. **Local computation:** each node i can, at time t, compute a subgradient of its local function $\nabla f_i(\theta)$ for a value $\theta \in \mathcal{M}_{i,t-1}$ in the node's internal memory before the computation.

$$\mathcal{M}_{i,t}^{\text{comp}} = \text{Span}\left(\{\theta, \nabla f_i(\theta) : \theta \in \mathcal{M}_{i,t-1}\}\right).$$

3. **Local communication:** each node i can, at time t, share a value to all or part of its neighbors, that is, for all $i \in [\![1, n]\!]$,

$$\mathcal{M}_{i,t}^{\text{comm}} = \text{Span}\Big(\bigcup_{(j,i)\in\mathcal{E}} \mathcal{M}_{j,t-\tau}\Big).$$

4. **Output value:** each node i must, at time t, specify one vector in its memory as local output of the algorithm, that is, for all $i \in [\![1, n]\!]$,

$$\theta_{i,t} \in \mathcal{M}_{i,t}\,.$$

Hence, a black-box procedure will return n output values—one for each computing unit—and our analysis will focus on ensuring that *all local output values* are converging to the optimal parameter of Equation 1. For simplicity, we assume that all nodes start with the simple internal memory $\mathcal{M}_{i,0} = \{0\}$. Note that communications and local computations may be performed in parallel and asynchronously.

3. Centralized Optimization under Global Regularity

The most standard approach for distributing a first-order optimization method consists in computing a gradient (or subgradient when the objective function is non-smooth) of the average function

$$\nabla \bar{f}(\theta) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\theta)\,,$$

where $\nabla f_i(\theta)$ is a gradient (or any subgradient of f_i at θ), by sending the current parameter θ_t to all nodes, performing the computation of all local subgradients in parallel and averaging them on a master node. Since each iteration requires communicating twice to the whole network (once for θ_t and once for sending the local subgradients to the master node, which both take a time $\Delta\tau$ where Δ is the diameter of the network) and one gradient computation (on each node and performed in parallel), the time to reach a precision ε with such a distributed optimization algorithm is upper-bounded by

$$O\Big(g(\varepsilon)(1 + \Delta\tau)\Big), \tag{2}$$

where $g(\varepsilon)$ is the number of iterations required for the single-machine optimization algorithm to achieve a precision ε. Note that, as the single-machine optimization algorithm is directly used for $\bar{f}$, this convergence rate depends on the global characteristics of objective function.

For smooth objective functions, we will see in Section 3.2 that distributing in such a way Nesterov's accelerated gradient descent leads to an optimal convergence rate. For non-smooth objective functions, however, the simple scheme is not optimal: the number of subgradient computations in Equation 2 (i.e., the term not proportional to τ) cannot be improved, since it is already optimal for objective functions defined on only one machine—see for example Theorem 3.13 p. 280 from Bubeck (2015). However, quite surprisingly, the error due to communication time may benefit from fast convergence rates in $O(RL_g/\varepsilon)$. This result is already known under the local regularity assumption (i.e., replacing L_g with L_ℓ or even $\max_i L_i$) in the case of decentralized optimization (Lan et al., 2017) or distributed optimization on a totally connected network (Arjevani and Shamir, 2015). To our knowledge, the case of global regularity has not been investigated by prior work.

We first describe algorithms in Section 3.1, and then study their optimality according in Section 3.2.

3.1. Fast Communication Rates for Non-Smooth Objective Functions

We now show that the simple smoothing approach introduced by Duchi et al. (2012b) can lead to fast rates for error due to communication time. Let $\gamma > 0$ and $f : \mathbb{R}^d \to \mathbb{R}$ be a real function. We denote as *smoothed version of* f the following function:

$$f^\gamma(\theta) = \mathbb{E}\left[f(\theta + \gamma X)\right], \tag{3}$$

where $X \sim \mathcal{N}(0, I)$ is a standard Gaussian random variable. The following lemma shows that f^γ is both smooth and a good approximation of f.

Lemma 1 (Lemma $E.3$ **of Duchi et al., 2012b)** *If $\gamma > 0$, then f^γ is $(\frac{L_g}{\gamma})$-smooth and, for all $\theta \in \mathbb{R}^d$,*

$$f(\theta) \quad \leq \quad f^\gamma(\theta) \quad \leq \quad f(\theta) + \gamma L_g \sqrt{d}\,.$$

Hence, smoothing the objective function allows the use of accelerated optimization algorithms and provides faster convergence rates. Of course, the price to pay is that each computation of the smoothed gradient $\nabla \bar{f}^\gamma(\theta) = \frac{1}{n}\sum_{i=1}^n \nabla f_i^\gamma(\theta)$ now requires, at each iteration m, to sample a sufficient amount of subgradients $\nabla f_i(\theta + \gamma X_{m,k})$ to approximate Equation 3, where $X_{m,k}$ are K i.i.d. Gaussian random variables. At first glance, this algorithm requires all computing units to synchronize on the choice of $X_{m,k}$, which would require to send to all nodes each $X_{m,k}$ and thus incur a communication cost proportional to the number of samples. Fortunately, computing units only need to share one random seed $s \in \mathbb{R}$ and then use a random number generator initialized with the provided seed to generate the same random variables $X_{m,k}$ without the need to communicate any vector. While the theoretical results assume that the randomness is shared, in practice this is done approximately by sharing random seeds and using pseudo-random generators. The overall algorithm, denoted *distributed randomized smoothing* (DRS), uses the randomized smoothing optimization algorithm of Duchi et al. (2012b) adapted to a distributed setting, and is summarized in Alg. 1. The computation of a spanning tree $\mathcal{T}$ in step 1 allows efficient communication to the whole network in time at most $\Delta\tau$. Most of the algorithm (i.e., steps $2, 4, 6, 7, 9, 10$ and 11) are performed on the root of the spanning subtree $\mathcal{T}$, while the rest of the computing units are responsible for computing the smoothed gradient (step 8). The seed s of step 2 is used to ensure that every $X_{m,k}$, although random, is the *same on every node*. Finally, step 10 is a simple orthogonal projection of the gradient step on the convex set $\mathcal{K}$.

3.1.1. Convex Case

We now show that the DRS algorithm converges to the optimal parameter under global Lipschitz regularity.

Theorem 2 *Under global Lipschitz-continuity, Alg. 1 achieves an approximation error $\mathbb{E}\left[\bar{f}(\theta_T)\right] - \bar{f}(\theta^*)$ of at most $\varepsilon > 0$ in a time T_ε upper-bounded by*

$$O\left(\left(\frac{RL_g}{\varepsilon}\right)^2 + \frac{RL_g}{\varepsilon}(1 + \Delta\tau)d^{1/4}\right). \tag{4}$$

Algorithm 1 distributed randomized smoothing (convex case)

Input: approximation error $\varepsilon > 0$, communication graph $\mathcal{G}$, $\alpha_0 = 1$, $\alpha_{t+1} = 2/(1 + \sqrt{1 + 4/\alpha_t^2})$
$T = \left\lceil \frac{20RL_g d^{1/4}}{\varepsilon} \right\rceil$, $K = \left\lceil \frac{5RL_g d^{-1/4}}{\varepsilon} \right\rceil$, $\gamma_t = Rd^{-1/4}\alpha_t$, $\eta_t = \frac{R\alpha_t}{2L_g(d^{1/4} + \sqrt{\frac{t+1}{K}})}$.

Output: optimizer θ_T

1: Compute a spanning tree $\mathcal{T}$ on $\mathcal{G}$.
2: Send a random seed s to every node in $\mathcal{T}$.
3: Initialize the random number generator of each node using s.
4: $x_0 = 0$, $z_0 = 0$, $G_0 = 0$
5: **for** $t = 0$ to $T - 1$ **do**
6: $\quad y_t = (1 - \alpha_t)x_t + \alpha_t z_t$
7: $\quad$ Send y_t to every node in $\mathcal{T}$.
8: $\quad$ Each node i computes $g_i = \frac{1}{K}\sum_{k=1}^K \nabla f_i(y_t + \gamma_t X_{t,k})$, where $X_{t,k} \sim \mathcal{N}(0, I)$
9: $\quad G_{t+1} = G_t + \frac{1}{n\alpha_t}\sum_i g_i$
10: $\quad z_{t+1} = \operatorname{argmin}_{x \in \mathcal{K}} \|x + \eta_{t+1} G_{t+1}\|_2^2$
11: $\quad x_{t+1} = (1 - \alpha_t)x_t + \alpha_t z_{t+1}$
12: **end for**
13: **return** $\theta_T = x_T$

More specifically, Alg. 1 completes its T iterations by time

$$T_\varepsilon \quad \leq \quad 100\left\lceil \frac{RL_g d^{1/4}}{\varepsilon} \right\rceil \left\lceil \frac{RL_g d^{-1/4}}{\varepsilon} \right\rceil + 40\left\lceil \frac{RL_g d^{1/4}}{\varepsilon} \right\rceil \Delta\tau\,. \tag{5}$$

Comparing Equation 4 to Equation 2, we can see that our algorithm benefits from faster convergence rates than the naïve distribution of gradient descent (leading to $g(\varepsilon) = (RL_g/\varepsilon)^2$ in Equation 2, which is optimal for single machine optimization of convex non-smooth problems) when the dimension is not too large, and more specifically

$$d \leq \left(\frac{RL_g}{\varepsilon}\right)^4.$$

In practice, this condition is easily met, as $\varepsilon \leq 10^{-2}$ already leads to the condition $d \leq 10^8$ (assuming that the product RL_g is close to 1). Moreover, for problems of moderate dimension, the term $d^{1/4}$ remains a small multiplicative factor (e.g. for $d = 1000$, $d^{1/4} \approx 6$). Finally, note that DRS achieves a linear speedup when communication through the whole network requires a constant time, i.e., $\Delta\tau = O(1)$.

Remark 3 *Several other smoothing methods exist in the literature, notably the* Moreau envelope *(Moreau, 1965) enjoying a dimension-free approximation guarantee. However, the Moreau envelope of an average of functions is difficult to compute (requires a different oracle than computing a subgradient), and unfortunately leads to convergence rates with respect to local Lipschitz characteristics instead of L_g. Moreover, the above analysis can be extended to any rotation invariant probability measure (e.g. uniform on a ball) for the smoothing noise. Such an analysis shows that the convergence rate obtained by using Gaussian random*

Algorithm 2 distributed randomized smoothing (strongly-convex case)

Input: approximation error $\varepsilon > 0$, communication graph $\mathcal{G}$,

$c_{d,\varepsilon} = d^{1/4} \log\left(\frac{8\left(\bar{f}(\theta_0)-\bar{f}(\theta^*)\right)}{\varepsilon}\right)$, $T = 2\left\lceil \frac{\sqrt{2}L_g c_{d,\varepsilon}}{\sqrt{\alpha_g \varepsilon}}\right\rceil$, $K = \left\lceil \frac{70 L_g c_{d,\varepsilon}^{-1}}{\sqrt{\alpha_g \varepsilon}}\right\rceil$, $\gamma = \frac{\varepsilon}{2L_g\sqrt{d}}$,

$n_0 = -1$, $n_1 = \left\lceil \sqrt{\frac{L_g}{\alpha_g \gamma}} \log\left(\frac{4\left(\bar{f}(\theta_0)-\bar{f}(\theta^*)\right)}{\varepsilon}\right)\right\rceil$, $\forall t > 1$, $n_t = 2^t \left\lceil \sqrt{\frac{L_g}{\alpha_g\gamma}}\log(8)\right\rceil$,

$\eta_1 = \frac{\gamma}{L_g}$, $\forall t > 1$, $\eta_t = \frac{\gamma}{2^{2t} L_g}$, $\mu_t = \frac{1-\sqrt{\alpha_g \eta_t}}{1+\sqrt{\alpha_g \eta_t}}$.

Output: optimizer θ_T

Compute a spanning tree $\mathcal{T}$ on $\mathcal{G}$.
Send a random seed s to every node in $\mathcal{T}$.
Initialize the random number generator of each node using s.
for $t = 1$ to T **do**
  $x_0^t = x_1^t = x_{n_{t-1}+1}^{t-1}$
  for $m = 1$ to n_t **do**
    $y_m^t = (1+\mu_t)x_m^t - \mu_t x_{m-1}^t$
    Send y_m^t to every node in $\mathcal{T}$.
    Each node i computes $g_i = \frac{1}{K}\sum_{k=1}^K \nabla f_i(y_m^t + \gamma X_k)$, where $X_k \sim \mathcal{N}(0, I)$
    $x_{m+1}^t = y_m^t - \frac{\eta_t}{n}\sum_i g_i$
  end for
end for
return $\theta_T = x_T$

noise is unimprovable by modifying the noise distribution. This is in accordance with recent theoretical results of Bubeck et al. (2019) that imply that the $d^{1/4}$ multiplicative factor in Equation 4 is unimprovable in the regime $T \leq \sqrt{d}$ where T is the number of iterations of the algorithm.

Remark 4 *Due to its random nature, Alg. 1 is not* per se *a black-box procedure, and Theorems 8 and 9 of Section 3.2 do not apply to it. Lower bounds for random algorithms are more challenging and left for future work.*

Remark 5 *Only the convexity of the* global *function $\bar{f}$ is necessary for Alg. 1 to converge. For example, the local functions f_i may be non convex differentiable functions whose average is convex.*

3.1.2. Strongly Convex Case

The same smoothing strategy is also applicable for strongly convex objectives, and yields similar improvements in the convergence rate. Unfortunately, the analysis of randomized smoothing for strongly convex objectives is yet missing in the literature, and we thus need to use another convergence result. Alg. 2 applies Nesterov's accelerated gradient descent to the smoothed objective function $\bar{f}^\gamma$, leading to fast communication rates for strongly convex objectives. More specifically, we use the recent algorithm M-ASG by Serhat Aybat et al. (2019) that provides tight convergence rates for accelerated gradient descent under additive noise on the gradient.

Theorem 6 *Under global Lipschitz-continuity and strong convexity and for $\varepsilon < \bar{f}(\theta_0) - \bar{f}(\theta^*)$, Alg. 2 achieves an approximation error $\mathbb{E}\left[\bar{f}(\theta_T)\right] - \bar{f}(\theta^*)$ of at most ε in a time T_ε upper-bounded by*

$$O\left(\frac{L_g^2}{\alpha_g \varepsilon} + \sqrt{\frac{L_g^2}{\alpha_g \varepsilon}}(1+\Delta\tau)d^{1/4}\log\left(\frac{1}{\varepsilon}\right)\right).$$

More specifically, Alg. 2 completes its T iterations by time

$$T_\varepsilon \;\leq\; 2\left\lceil \frac{\sqrt{2}L_g c_{d,\varepsilon}}{\sqrt{\alpha_g\varepsilon}}\right\rceil \left\lceil \frac{70 L_g c_{d,\varepsilon}^{-1}}{\sqrt{\alpha_g\varepsilon}}\right\rceil + 4\left\lceil \frac{\sqrt{2}L_g c_{d,\varepsilon}}{\sqrt{\alpha_g\varepsilon}}\right\rceil \Delta\tau\,, \tag{6}$$

where $c_{d,\varepsilon} = d^{1/4}\log\left(\frac{8\left(\bar{f}(\theta_0)-\bar{f}(\theta^*)\right)}{\varepsilon}\right)$. The proof is available in Appendix A.

While Alg. 2 relies on the two-level nested iterations of M-ASG, we believe that more simple algorithms may also lead to similar convergence rates, e.g. directly using Nesterov's gradient descent and its recent stochastic analysis by Vaswani et al. (2019). Note that Theorem 6, as well as all lower bounds for the strongly convex case, is an addition compared to our preliminary work (Scaman et al., 2018).

Remark 7 *The Lipschitz and convex setting of Section 3.1.1 can also be solved by Alg. 2 by adding a quadratic regularization $\frac{\alpha}{2}\|\theta\|_2^2$ with a sufficiently small α, although this would incur a multiplicative logarithmic factor compared to the convergence rate of Alg. 1.*

3.2. Optimal Convergence Rates

The following results provide oracle complexity lower bounds for centralized algorithms, and are proved in Appendix B. These lower bounds extend the communication complexity lower bounds for totally connected communication networks of Arjevani and Shamir (2015). Note that, in all following results, we consider that the dimension d is larger than both the number of communication and computation steps. This assumption is relatively standard in non-smooth optimization (Nesterov, 2004, Theorem 3.2.1) and valid for either small times or large dimensional problems. When the number of iterations is larger than the dimension d, linearly convergent algorithms exist even for convex non-smooth problems (Bubeck, 2015).

3.2.1. Non-Smooth Objective Functions

Theorem 8 *Let $\mathcal{G}$ be a network of computing units of size $n > 0$, and $L_g, R > 0$. There exists n convex functions $f_i : \mathbb{R}^d \to \mathbb{R}$ such that $\bar{f}$ is L_g-Lipschitz and, for any $t < \frac{d-2}{2}\min\{1, \Delta\tau\}$ and any black-box procedure one has, for all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \min_{\theta \in B_2(R)} \bar{f}(\theta) \geq \frac{RL_g}{36}\sqrt{\frac{1}{(1+\frac{t}{2\Delta\tau})^2} + \frac{1}{1+t}}\,. \tag{7}$$

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d > 2 + 2\max\{t, t/\Delta\tau\}$), Theorem 8 implies that, under global Lipschitz

continuity, the time to reach a precision $\varepsilon > 0$ with any black-box procedure is lower bounded by

$$\Omega\left(\left(\frac{RL_g}{\varepsilon}\right)^2 + \frac{RL_g}{\varepsilon}\Delta\tau\right),$$

where the notation $g(\varepsilon) = \Omega(f(\varepsilon))$ stands for $\exists C > 0$ s.t. $\forall \varepsilon > 0, g(\varepsilon) \geq Cf(\varepsilon)$. This lower bound proves that the convergence rate of DRS in Equation 4 is optimal with respect to computation time and within a $d^{1/4}$ multiplicative factor of the optimal convergence rate with respect to communication.

Theorem 9 *Let $\mathcal{G}$ be a network of computing units of size $n > 0$, and $L_g, \alpha_g > 0$. There exists a convex set $\Theta \subset \mathbb{R}^d$ and n convex functions $f_i : \Theta \to \mathbb{R}$ such that $\bar{f}$ is L_g-Lipschitz and α_g-strongly convex and, for any $t < \frac{d-2}{2}\min\{1, \Delta\tau\}$ and any black-box procedure one has, for all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \min_{\theta\in\Theta} \bar{f}(\theta) \geq \frac{L_g^2}{108\alpha_g}\left(\frac{1}{(1+\frac{t}{2\Delta\tau})^2} + \frac{1}{1+t}\right). \tag{8}$$

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d > 2 + 2\max\{t, t/\Delta\tau\}$), Theorem 9 implies that, under global strong convexity and Lipschitz continuity, the time to reach a precision $\varepsilon > 0$ with any black-box procedure is lower bounded by

$$\Omega\left(\frac{L_g^2}{\alpha_g\varepsilon} + \sqrt{\frac{L_g^2}{\alpha_g\varepsilon}}\Delta\tau\right).$$

This lower bound proves that the convergence rate of the strongly convex version of DRS in Equation 6 is, up to logarithmic factors, optimal with respect to computation time and within a $d^{1/4}$ multiplicative factor of the optimal convergence rate with respect to communication.

The proof of Theorem 8 and Theorem 9 relies on the use of two objective functions: first, the standard worst case function used for single machine convex optimization (see, e.g., Bubeck, 2015) is used to obtain a lower bound on the local computation time of individual machines. Then, a second function first introduced by Arjevani and Shamir (2015) is split on the two most distant machines to obtain worst case communication times. By aggregating these two functions, a third one is obtained with the desired lower bound on the convergence rate. The complete proof is available in Appendix B.1. Finally, note that, due to its random nature, Alg. 1 is not *per se* a black-box procedure, and Theorems 8 and 9 do not apply to it. Lower bounds for random algorithms are more challenging and left for future work.

Remark 10 *The lower bounds also hold for the average of local parameters $\frac{1}{n}\sum_{i=1}^n \theta_i$, and more generally any parameter that can be computed using any vector of the local memories at time t: in Theorem 8, $\theta_{i,t}$ may be replaced by any θ_t such that*

$$\theta_t \in \mathrm{Span}\left(\bigcup_{i\in\mathcal{V}} \mathcal{M}_{i,t}\right).$$

3.2.2. Smooth Objective Functions

The following theorems provide lower bounds for smooth centralized optimization. The first is a novel result of this work, while the second was first derived by Scaman et al. (2017, Theorem 1) and is included in this section for the sake of completeness.

Theorem 11 *Let $\mathcal{G}$ be a network of computing units of size $n > 0$, and $\beta_g > 0$. There exists n convex functions $f_i : \mathbb{R}^d \to \mathbb{R}$ such that $\bar{f}$ is β_g-smooth and, for any $t \leq 1 + \frac{d-3}{2}(1+\Delta\tau)$ and any black-box procedure one has, for all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \min_{\theta\in\mathbb{R}^d} \bar{f}(\theta) \geq \frac{3\beta_g\|\theta_{i,0} - \theta^*\|^2}{32\left(\frac{t-1}{1+\Delta\tau}+1\right)^2}, \tag{9}$$

where θ^* is a minimizer of $\bar{f}$.

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d > 3+2(t-1)/(1+\Delta\tau)$), Theorem 11 implies that, under global smoothness, the time to reach a precision $\varepsilon > 0$ with any black-box procedure is lower bounded by

$$\Omega\left(\sqrt{\frac{\beta_g\|\theta_{i,0} - \theta^*\|^2}{\varepsilon}}\left(1 + \Delta\tau\right)\right).$$

In order to simplify the proofs of the following theorem, and following the approach of Bubeck (2015), we will consider the limiting situation $d \to +\infty$. More specifically, we now assume that we are working in $\ell_2 = \{\theta = (\theta_k)_{k\in\mathbb{N}} \ : \ \sum_k \theta_k^2 < +\infty\}$ rather than $\mathbb{R}^d$.

Theorem 12 *Let $\mathcal{G}$ be a network of computing units of size $n > 0$, and $\beta_g \geq \alpha_g > 0$. There exists n convex functions $f_i : \ell_2 \to \mathbb{R}$ such that $\bar{f}$ is α_g strongly convex and β_g smooth, and for any $t \geq 0$ and any black-box procedure one has, for all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \bar{f}(\theta^*) \geq \frac{\alpha_g}{2}\left(1 - \frac{4}{\sqrt{\kappa_g}}\right)^{1+\frac{t}{1+\Delta\tau}}\|\theta_{i,0} - \theta^*\|^2, \tag{10}$$

where $\kappa_g = \beta_g/\alpha_g$.

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d \gg \max\{t, t/\Delta\tau\}$), Theorem 12 implies that, under global strong convexity and smoothness, the time to reach a precision $\varepsilon > 0$ with any black-box procedure is lower bounded by

$$\Omega\left(\sqrt{\kappa_g}\ln\left(\frac{1}{\varepsilon}\right)\left(1 + \Delta\tau\right)\right).$$

These two lower bounds prove that the naïve distribution of Nesterov's gradient descent is optimal for both smooth and smooth and strongly convex objective functions. Their proofs rely on splitting the functions used by Nesterov to prove oracle complexities for smooth optimization (Nesterov, 2004; Bubeck, 2015) on two nodes at distance Δ. One can show that most dimensions of the parameters $\theta_{i,t}$ will remain zero, and local gradient computations may only increase the number of non-zero dimensions by one. Finally, at least Δ communication rounds are necessary in-between every gradient computation, in order to share information between the two nodes. The complete proofs are available in Appendix B.2.

4. Decentralized Optimization under Local Regularity

In many practical scenarios, the network may be unknown or changing through time, and a local communication scheme is preferable to the *master/slave* approach of Alg. 1. Decentralized algorithms tackle this problem by replacing targeted communication by *local averaging* of the values of neighboring nodes (Boyd et al., 2006). More specifically, we now consider that, during a communication step, each machine i broadcasts a vector $x_i \in \mathbb{R}^d$ to its neighbors, then performs a weighted average of the values received from its neighbors:

$$\text{node } i \text{ sends } x_i \text{ to his neighbors and receives } \textstyle\sum_j W_{ji} x_j \,.$$

In order to ensure the efficiency of this communication scheme, we impose standard assumptions on the matrix $W \in \mathbb{R}^{n \times n}$, called the *gossip* matrix (Boyd et al., 2006; Scaman et al., 2017):

1. W is symmetric and positive semi-definite.
2. The kernel of W is the set of constant vectors: $\mathrm{Ker}(W) = \mathrm{Span}(\mathbb{1})$, where $\mathbb{1} = (1, ..., 1)^\top$.
3. W is defined on the edges of the network: $W_{ij} \neq 0$ only if $i = j$ or $(i, j) \in \mathcal{E}$.

This procedure, known as gossip algorithm (Boyd et al., 2006), is a standard method for averaging values across a network when its connectivity may vary through time. This approach was shown to be robust against machine failures, non-uniform latencies and asynchronous or time-varying graphs, and a large body of literature extended this algorithm to distributed optimization (Nedic and Ozdaglar, 2009; Duchi et al., 2012a; Wei and Ozdaglar, 2012; Shi et al., 2015; Jakovetić et al., 2015; Nedic et al., 2017; Aryan Mokhtari and Ribeiro, 2016). Note that these assumptions are implied by symmetry, stochasticity and positive eigengap on $I - W$. The convergence analysis of decentralized algorithms usually relies on the spectrum of the *gossip matrix* W used for communicating values in the network, and more specifically on the ratio between the second smallest and the largest eigenvalue of W, denoted $\gamma(W) = \lambda_{n-1}(W)/\lambda_1(W)$ and sometimes called *normalized eigengap*. More precisely, our lower bounds on the optimal convergence rate are obtained by replacing the diameter of the network with the quantity $\widetilde{\Delta} = 1/\sqrt{\gamma(W)}$, that we refer to as the *mixing time* due to its relation to the mixing time of random walks when W is the Laplacian matrix of the communication network.

To ease the presentation of this section, and contrary to Section 3, we first present the lower bounds for decentralized optimization in Section 4.1, before describing the corresponding optimal algorithms in Section 4.2.

Remark 13 *When communication is asymmetric, or the assumptions on the gossip matrix do not hold, traditional gossip methods fail and more advanced communication methods are required. We refer the reader to Nedic and Olshevsky (2015); Xi and Khan (2017); Nedic and Olshevsky (2016); Nedic et al. (2016) for asymmetric decentralized optimization algorithms.*

4.1. Optimal Convergence Rates

The following results provide oracle complexity lower bounds for decentralized algorithms, and are proved in Appendix C. Similarly to Section 3.2, we consider that the dimension d is larger than both the number of communication and computation steps. This assumption is relatively standard in non-smooth optimization (Nesterov, 2004, Theorem 3.2.1) and valid for either small times or large dimensional problems. When the number of iterations is larger than the dimension d, linearly convergent algorithms exist even for convex non-smooth problems (Bubeck, 2015).

4.1.1. Non-Smooth Objective Functions

Theorem 14 *Let $L_\ell, R > 0$ and $\widetilde{\Delta} \geq 1$. There exists a matrix W of mixing time $\widetilde{\Delta}$, and n convex and L_i-Lipschitz functions f_i, where n is the size of W, such that for all $t < \frac{d-2}{2}\min(1, \widetilde{\Delta}\tau)$ and all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \min_{\theta \in B_2(R)} \bar{f}(\theta) \geq \frac{RL_\ell}{108} \sqrt{\frac{1}{\left(1 + \frac{2t}{\widetilde{\Delta}\tau}\right)^2} + \frac{1}{1+t}}\,,$$

where $L_\ell = \sqrt{\frac{1}{n}\sum_i L_i^2}$.

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d > 2 + 2\max\{t, t/\widetilde{\Delta}\tau\}$), Theorem 14 implies that, under local Lipschitz continuity and for a gossip matrix W with mixing time $\widetilde{\Delta}$, the time to reach a precision $\varepsilon > 0$ with any *decentralized* black-box procedure is lower-bounded by

$$\Omega\left(\left(\frac{RL_\ell}{\varepsilon}\right)^2 + \frac{RL_\ell}{\varepsilon}\widetilde{\Delta}\tau\right).$$

Theorem 15 *Let $L_\ell, \alpha_\ell > 0$ and $\widetilde{\Delta} \geq 1$. There exists a matrix W of size n and mixing time $\widetilde{\Delta}$, a convex set $\Theta \subset \mathbb{R}^d$ and n α_i-strongly convex and L_i-Lipschitz functions $f_i : \Theta \to \mathbb{R}$, such that for all $t < \frac{d-2}{2}\min(1, \widetilde{\Delta}\tau)$ and all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \min_{\theta \in \Theta} \bar{f}(\theta) \geq \frac{L_\ell^2}{648\alpha_\ell} \sqrt{\frac{1}{\left(1 + \frac{2t}{\widetilde{\Delta}\tau}\right)^2} + \frac{1}{1+t}}\,,$$

where $\alpha_\ell = \min_i \alpha_i$ and $L_\ell = \sqrt{\frac{1}{n}\sum_i L_i^2}$.

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d > 2+2\max\{t, t/\widetilde{\Delta}\tau\}$), Theorem 15 implies that, under local strong convexity and Lipschitz continuity, and for a gossip matrix W with mixing time $\widetilde{\Delta}$, the time to reach a precision $\varepsilon > 0$ with any *decentralized* black-box procedure is lower-bounded by

$$\Omega\left(\frac{L_\ell^2}{\alpha_\ell \varepsilon} + \sqrt{\frac{L_\ell^2}{\alpha_\ell \varepsilon}}\widetilde{\Delta}\tau\right).$$

The proof of Theorem 14 relies on linear graphs (whose diameter is proportional to $\widetilde{\Delta}$ when the Laplacian matrix is used as gossip matrix) and Theorem 8. More specifically, a technical aspect of the proof consists in splitting the functions used in Theorem 8 on multiple nodes to obtain a dependency in L_ℓ instead of L_g. The complete derivation is available in Appendix C.1.

4.1.2. Smooth Objective Functions

The following theorems provide lower bounds for smooth centralized optimization. The first is a novel result of this work, while the second was first derived by Scaman et al. (2017, Theorem 2) and is included in this section for the sake of completeness.

Theorem 16 *Let $\beta_\ell > 0$ and $\widetilde{\Delta} \geq 1$. There exists a matrix W of mixing time $\widetilde{\Delta}$, and n convex and β_i-smooth functions f_i, where n is the size of W, such that for all $t \leq 1 + \frac{d-3}{2}(1+\widetilde{\Delta}\tau)$ and all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \min_{\theta \in \mathbb{R}^d} \bar{f}(\theta) \geq \frac{3\beta_\ell \|\theta_{i,0} - \theta^*\|^2}{64\left(\frac{3(t-1)}{3+\widetilde{\Delta}\tau} + 1\right)^2},$$

where θ^* is a minimizer of $\bar{f}$ and $\beta_\ell = \max_i \beta_i$.

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d > 3 + 2(t-1)/(1+\widetilde{\Delta}\tau)$), Theorem 16 implies that, under local smoothness, and for a gossip matrix W with mixing time $\widetilde{\Delta}$, the time to reach a precision $\varepsilon > 0$ with any *decentralized* black-box procedure is lower-bounded by

$$\Omega\left(\sqrt{\frac{\beta_\ell \|\theta_{i,0} - \theta^*\|^2}{\varepsilon}}\left(1 + \widetilde{\Delta}\tau\right)\right).$$

Theorem 17 *Let $\alpha, \beta > 0$ and $\widetilde{\Delta} \geq 1$. There exists a gossip matrix W of mixing time $\widetilde{\Delta}$, and α-strongly convex and β-smooth functions $f_i : \ell_2 \to \mathbb{R}$ such that, for any $t \geq 0$ and any black-box procedure using W one has, for all $i \in [\![1, n]\!]$,*

$$\bar{f}(\theta_{i,t}) - \bar{f}(\theta^*) \geq \frac{\alpha}{2}\left(1 - \frac{16}{\sqrt{\kappa_\ell}}\right)^{1+\frac{3t}{3+\widetilde{\Delta}\tau}} \|\theta_{i,0} - \theta^*\|^2,$$

where $\kappa_\ell = \beta/\alpha$ is the local condition number.

Assuming that the dimension d is large compared to the characteristic values of the problem (i.e., $d \gg \max\{t, t/\widetilde{\Delta}\tau\}$), Theorem 17 implies that, under local strong convexity and smoothness, and for a gossip matrix W with mixing time $\widetilde{\Delta}$, the time to reach a precision $\varepsilon > 0$ with any *decentralized* black-box procedure is lower-bounded by

$$\Omega\left(\sqrt{\kappa_l} \ln\left(\frac{1}{\varepsilon}\right)\left(1 + \widetilde{\Delta}\tau\right)\right).$$

The proofs of Theorem 16 and Theorem 17 rely on the same technique as that of Theorem 14 and Theorem 15. The complete proof is available in Appendix C.2.

We will see in the next section that this lower bound is met for a novel decentralized algorithm called *multi-step primal-dual* (MSPD) and based on the dual formulation of the optimization problem. Note that these results provide optimal convergence rates with respect to κ_l and $\widetilde{\Delta}$, but do not imply that $\widetilde{\Delta}$ is the right quantity to consider on general graphs. The quantity $\widetilde{\Delta}$ may indeed be very large compared to Δ, for example for star networks, for which $\Delta = 2$ and $\widetilde{\Delta} = \sqrt{n}$. However, on many simple networks, the diameter Δ and the eigengap of the Laplacian matrix are tightly connected, and $\Delta \approx \widetilde{\Delta}$. For example, for linear graphs, $\Delta = n - 1$ and $\widetilde{\Delta} \approx 2n/\pi$, for totally connected networks, $\Delta = 1$ and $\widetilde{\Delta} = 1$, and for regular networks, $\widetilde{\Delta} \geq \frac{\Delta}{2\sqrt{2}\ln_2 n}$ (Alon and Milman, 1985). Finally, note that the case of totally connected networks corresponds to a previous complexity lower bound on communications proven by Arjevani and Shamir (2015), and is equivalent to our result for centralized algorithms with $\Delta = 1$.

4.2. Optimal Decentralized Algorithms

In this section, we present optimal algorithms for the decentralized setting.

4.2.1. Non-Smooth Objective Functions

We now provide an optimal decentralized optimization algorithm under (A2). This algorithm is closely related to the primal-dual algorithm proposed by Lan et al. (2017), which we modify by the use of accelerated gossip using Chebyshev polynomials as done by Scaman et al. (2017).

First, following Jakovetić et al. (2015), we formulate our optimization problem in Equation 1 as the saddle-point problem in Equation 11 below, by considering the equivalent problem of minimizing $\frac{1}{n}\sum_{i=1}^n f_i(\theta_i)$ over $\Theta = (\theta_1, \dots, \theta_n) \in \mathcal{K}^n$ with the constraint that $\theta_1 = \dots = \theta_n$, or equivalently $\Theta A = 0$, where A is a square root of the symmetric matrix W. Through Lagrangian duality, we therefore get the equivalent problem:

$$\min_{\Theta \in \mathcal{K}^n} \max_{\Lambda \in \mathbb{R}^{d \times n}} \frac{1}{n}\sum_{i=1}^n f_i(\theta_i) - \operatorname{tr} \Lambda^\top \Theta A. \tag{11}$$

We solve it by applying the algorithm of Chambolle and Pock (2011), which is both simple and well tailored to our problem—we could alternatively apply composite Mirror-Prox (He et al., 2015): (a) it is an accelerated method for saddle-point problems, (b) it allows for composite problems with a sum of non-smooth and smooth terms, (c) it provides a primal-dual gap that can easily be extended to the case of approximate proximal operators. At each iteration t, with initialization $\Lambda^0 = 0$ and $\Theta^0 = \Theta^{-1} = (\theta_0, \dots, \theta_0)$:

$$\begin{aligned}
(a) \quad \Lambda^{t+1} &= \Lambda^t - \sigma(2\Theta^t - \Theta^{t-1})A \\
(b) \quad \Theta^{t+1} &= \operatorname*{argmin}_{\Theta \in \mathcal{K}^n} \frac{1}{n}\sum_{i=1}^n f_i(\theta_i) - \operatorname{tr} \Theta^\top \Lambda^{t+1} A^\top + \frac{1}{2\eta}\operatorname{tr}(\Theta - \Theta^t)^\top(\Theta - \Theta^t),
\end{aligned}$$

where the gain parameters η, σ are required to satisfy $\sigma\eta\lambda_1(W) \leq 1$. We implement the algorithm with the variables Θ^t and $Y^t = \Lambda^t A^\top = (y_1^t, \dots, y_n^t) \in \mathbb{R}^{d \times n}$, for which all updates

Algorithm 3 multi-step primal-dual algorithm

Input: approximation error $\varepsilon > 0$, gossip matrix $W \in \mathbb{R}^{n\times n}$,
$T = \lceil \frac{2RL_\ell \widetilde{\Delta}}{\varepsilon} \rceil$, $M = \lceil \frac{2RL_\ell}{\varepsilon \widetilde{\Delta}} \rceil$, $\eta = \frac{nR}{L_\ell \widetilde{\Delta}}$, $\sigma = \frac{1}{\eta \lambda_1(W)}$.

Output: optimizer $\bar{\theta}_T$

$\Theta_0 = 0$, $\Theta_{-1} = 0$, $Y_0 = 0$
for $t = 0$ to $T-1$ **do**
  $Y^{t+1} = Y^t - \sigma(2\Theta^t - \Theta^{t-1})W$
  $\tilde{\Theta}^0 = \Theta^t$
  for $m = 0$ to $M-1$ **do**
    $\tilde{\theta}_i^{m+1} = \frac{m}{m+2}\tilde{\theta}_i^m - \frac{2}{m+2}[\frac{\eta}{n}\nabla f_i(\tilde{\theta}_i^m) - \eta y_i^{t+1} - \theta_i^t]$, $\forall i \in [\![1,n]\!]$
  end for
  $\Theta^{t+1} = \tilde{\Theta}^M$
end for
return $\bar{\theta}_T = \frac{1}{T}\frac{1}{n}\sum_{t=1}^T \sum_{i=1}^n \theta_i^t$

can be made locally: since $AA^\top = W$, they now become

$$\begin{aligned} (a') \quad Y^{t+1} &= Y^t - \sigma(2\Theta^t - \Theta^{t-1})W \\ (b') \quad \theta_i^{t+1} &= \operatorname*{argmin}_{\theta_i \in \mathcal{K}} \frac{1}{n} f_i(\theta_i) - \theta_i^\top y_i^{t+1} + \frac{1}{2\eta}\|\theta_i - \theta_i^t\|^2, \forall i \in [\![1,n]\!]\,. \end{aligned} \tag{12}$$

The step (b') still requires a proximal step for each function f_i. We approximate it by the outcome of the subgradient method run for M steps, with a step-size proportional to $2/(m+2)$ as suggested by Lacoste-Julien et al. (2012). That is, initialized with $\tilde{\theta}_i^0 = \theta_i^t$, it performs the iterations

$$\tilde{\theta}_i^{m+1} = \frac{m}{m+2}\tilde{\theta}_i^m - \frac{2}{m+2}\left[\frac{\eta}{n}\nabla f_i(\tilde{\theta}_i^m) - \eta y_i^{t+1} - \theta_i^t\right], \quad m = 0, \dots, M-1. \tag{13}$$

We thus replace the step (b') by running M steps of the subgradient method to obtain $\tilde{\theta}_i^M$.

Theorem 18 *Under local Lipschitz continuity, the approximation error with the iterative algorithm of Equation 12 and 13 after T iterations and using M subgradient steps per iteration is bounded by*

$$\bar{f}(\bar{\theta}_T) - \min_{\theta \in \mathcal{K}} \bar{f}(\theta) \le RL_\ell\Big(\frac{\widetilde{\Delta}}{T} + \frac{1}{M\widetilde{\Delta}}\Big).$$

Theorem 18 implies that the proposed algorithm achieves an error of at most ε in a time no larger than

$$O\left(\left(\frac{RL_\ell}{\varepsilon}\right)^2 + \frac{RL_\ell}{\varepsilon}\widetilde{\Delta}\tau\right), \tag{14}$$

which is optimal due to Theorem 14. This algorithm, called *multi-step primal-dual* (MSPD), is described in Alg. 3.

Remark 19 *It is clear from the algorithm's description that it completes its T iterations by time*

$$T_\varepsilon \le \left\lceil \frac{2RL_\ell \widetilde{\Delta}}{\varepsilon} \right\rceil \left\lceil \frac{2RL_\ell}{\varepsilon \widetilde{\Delta}} \right\rceil + \left\lceil \frac{2RL_\ell \widetilde{\Delta}}{\varepsilon} \right\rceil \tau\,. \tag{15}$$

To obtain the average of local parameters $\bar{\theta}_T = \frac{1}{nT}\sum_{t=1}^T \sum_{i=1}^n \theta_i$, one can then rely on accelerated gossip (Auzinger, 2011; Arioli and Scott, 2014) to average over the network the individual nodes' time averages. This leads to a time $O(\ln(\frac{RL_\ell}{\varepsilon})\widetilde{\Delta}\tau)$ to ensure that each node reaches a precision ε on the objective function (see Boyd et al., 2006, for more details on the linear convergence of gossip), which is negligible compared to Equation 14.

Remark 20 *A stochastic version of the algorithm is also possible by considering stochastic oracles on each f_i and using stochastic subgradient descent instead of the subgradient method.*

Remark 21 *In the more general context where node compute times ρ_i are not necessarily all equal to 1, we may still apply Alg. 3, where now the number of subgradient iterations performed by node i is M/ρ_i rather than M. The proof of Theorem 18 also applies, and now yields the modified upper bound on time to reach precision ε:*

$$O\left(\left(\frac{RL_c}{\varepsilon}\right)^2 + \frac{RL_\ell}{\varepsilon}\widetilde{\Delta}\tau\right), \tag{16}$$

where $L_c^2 = \frac{1}{n}\sum_{i=1}^n \rho_i L_i^2$.

4.2.2. Other Regularity Assumptions

While this primal-dual approach could in theory be applied to all remaining three settings (strongly convex, smooth, and both strongly convex and smooth), the convergence rates of the Chambolle-Pock algorithm are weaker in such settings, and lead to convergence rates with respect to the distance to the optimum in squared norm $\|\theta_t - \theta^*\|^2$ (see Chambolle and Pock, 2011, Theorems 2 and 3). These issues could be avoided using the stronger results of composite Mirror-Prox (He et al., 2015), but would require substantial modifications to the algorithm. Fortunately, these cases were recently investigated by Dvinskikh and Gasnikov (2019), who designed optimal algorithms up to logarithmic factors for each decentralized communication setting (*R-Sliding* for strongly convex objectives and *PSTM* for smooth objectives). We thus refer to their work for the remaining cases.

5. Conclusion

In this paper, we provide optimal convergence rates for convex distributed optimization for two communication schemes (centralized and decentralized) and under four regularity assumptions on the objective function: Lipschitz continuity, strong convexity, smoothness, and both strong convexity and smoothness. Under decentralized communication, we provide optimal convergence rates that depend on the *local* regularity characteristics and *mixing time* of the gossip matrix. Moreover, we also provide the first optimal decentralized algorithm, called *multi-step primal-dual* (MSPD).

Under centralized communication, we first show that the naïve distribution of accelerated gradient descent is optimal for smooth functions. Then, we provide lower complexity bounds for non-smooth functions that depends on the (global) characteristics of the objective function, as well as a distributed version of the smoothing approach of Duchi et al. (2012b) and show that this algorithm is within a $d^{1/4}$ multiplicative factor of the optimal convergence rate.

In both settings, the optimal convergence rate for non-smooth distributed optimization exhibits two different speeds: a slow rate in $\Theta(1/\sqrt{t})$ with respect to local computations and a fast rate in $\Theta(1/t)$ due to communication. Intuitively, communication is the limiting factor in the initial phase of optimization. However, its impact decreases with time and, for the final phase of optimization, local computation time is the main limiting factor.

The analysis presented in this paper allows several natural extensions, including time-varying communication networks, asynchronous algorithms, in the line of Hendrikx et al. (2019b), stochastic settings (see, e.g., Hendrikx et al., 2019a; Stich, 2018), and an analysis of unequal node compute speeds going beyond Remark 21. Moreover, despite the efficiency of DRS, finding an optimal algorithm under the global regularity assumption remains an open problem and would make a notable addition to this work.

Acknowledgments

We acknowledge support from the European Research Council (grant SEQUOIA 724063).

Appendix A. Proof of the Convergence Rate of DRS (Theorems 2 and 6)

In this appendix, we provide detailed proofs of Theorems 2 and 6, that is, for the convex and strongly-convex cases.

A.1. Convex Case

Corollary 2.4 of Duchi et al. (2012b) gives, with the appropriate choice of gradient step η_t and smoothing γ_t,

$$\mathbb{E}\left[\bar{f}(\theta_T)\right] - \min_{\theta\in\mathcal{K}} \bar{f}(\theta) \leq \frac{10RL_g d^{1/4}}{T} + \frac{5RL_g}{\sqrt{TK}}.$$

Thus, to reach a precision $\varepsilon > 0$, we may set $T = \left\lceil \frac{20RL_g d^{1/4}}{\varepsilon} \right\rceil$ and $K = \left\lceil \frac{5RL_g d^{-1/4}}{\varepsilon} \right\rceil$, leading to the desired bound on the time $T_\varepsilon = T(2\Delta\tau + K)$ to reach ε.

A.2. Strongly Convex Case

Using Corollary 3.9 of Serhat Aybat et al. (2019) to optimize $\bar{f}^\gamma$, the number of iterations n_ε to reach a precision $\varepsilon/2$ is bounded by

$$n_\varepsilon \leq \left\lceil \sqrt{\frac{2L_g^2}{\alpha_g \varepsilon}} c_{d,\varepsilon} \right\rceil + 32(1+\log(8)) \frac{L_g^2}{\alpha_g K \varepsilon},$$

where $c_{d,\varepsilon} = d^{1/4} \log\left(\frac{8(\bar{f}(\theta_0) - \bar{f}(\theta^*))}{\varepsilon}\right)$. Since $\bar{f}^\gamma$ is an $L_g\gamma\sqrt{d}$-approximation of $\bar{f}$ (see Lemma 1), setting $\gamma = \frac{\varepsilon}{2L_g\sqrt{d}}$ leads to a precision ε on $\bar{f}$ after n_ε iterations. Finally, each iteration takes a time $n_\varepsilon(K + 2\Delta\tau)$, and setting $K = \left\lceil \frac{70 L_g c_{d,\varepsilon}^{-1}}{\sqrt{\alpha_g \varepsilon}} \right\rceil$ leads to the desired result.

Appendix B. Proof of the Centralized Lower Bounds (Theorems 8, 9, 11 and 12)

The proofs of lower bounds all rely on the fact that, for well-chosen local functions, most of the coordinates of the vectors in the memory of any node will remain equal to 0.

Definition 22 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph of n nodes, $I_0, I_1 \subset \mathcal{V}$ and $\|x\|_{nz} = \max\{k \in \mathbb{N} : x_k \neq 0\}$ the largest non-zero coordinate of x. Local functions $(f_i)_{i\in[\![1,n]\!]}$ are called* (I_0, I_1)-zero preserving *if, for all $x \in \mathbb{R}^d$ and $i \in [\![1, n]\!]$,*

$$\|\nabla f_i(x)\|_{nz} \leq \begin{cases} \|x\|_{nz} + \mathbb{1}\{\|x\|_{nz} \equiv 0 \ mod\ 2\} & \text{if } i \in I_0, \\ \|x\|_{nz} + \mathbb{1}\{\|x\|_{nz} \equiv 1 \ mod\ 2\} & \text{if } i \in I_1, \\ \|x\|_{nz} & \text{otherwise.} \end{cases}$$

In other words, local gradients can only increase even dimensions for nodes in I_0 and odd dimensions for nodes in I_1. With such objective functions, information needs to travel from I_0 to I_1 at each iteration in order to increase the number of non-zero coordinates.

Lemma 23 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph of n nodes, $I_0, I_1 \subset \mathcal{V}$ and $k_{i,t} = \max\{k \in \mathbb{N} : \exists \theta \in \mathcal{M}_{i,t} \text{ s.t. } \theta_k \neq 0\}$ be the last non-zero coordinate of a vector in the memory of node i at time t. If the local functions are (I_0, I_1)-zero preserving, then*

$$\forall i \in \mathcal{V}, k_{i,t} \leq \left\lfloor \frac{t-1}{1 + d(I_0, I_1)\tau} \right\rfloor + 1, \tag{17}$$

where $d(I_0, I_1)$ is the shortest-path distance between I_0 and I_1 in $\mathcal{G}$.

Proof Since the local functions are zero preserving, we have, under any black-box procedure and for any local computation step,

$$k_{i,t+1} \leq \begin{cases} k_{i,t} + \mathbb{1}\{k_{i,t} \equiv 0 \bmod 2\} & \text{if } i \in I_0, \\ k_{i,t} + \mathbb{1}\{k_{i,t} \equiv 1 \bmod 2\} & \text{if } i \in I_1, \\ k_{i,t} & \text{otherwise.} \end{cases}$$

Thus, in order to reach the third coordinate, algorithms must first perform one local computation in I_0, then $d(I_0, I_1)$ communication steps in order for a node in I_1 to have a non-zero

second coordinate, and finally, one local computation in I_1. Accordingly, one must perform at least k local computation steps and $(k-1)d(I_0, I_1)$ communication steps to achieve $k_{i,t} \geq k$ for at least one node $i \in \mathcal{V}$, and thus, for any $k \in \mathbb{N}^*$,

$$\forall t < 1 + (k-1)(1 + d(I_0, I_1)\tau), \;\; k_{i,t} \leq k - 1 \,,$$

which leads to the desired result. ■

B.1. Non-Smooth Lower Bounds

Let $i_0 \in \mathcal{V}$ and $i_1 \in \mathcal{V}$ be two nodes at distance Δ. Following Nesterov (2004), the function used by Bubeck (2015) to prove the oracle complexity for non-smooth convex functions is

$$g_1(\theta) = \delta \max_{i \in [\![1,t]\!]} \theta_i + \frac{\alpha}{2}\|\theta\|_2^2.$$

By considering this function on a single node (e.g. i_0), at least $O\left(\left(\frac{RL}{\varepsilon}\right)^2\right)$ subgradients will be necessary to obtain a precision ε. Moreover, we also split the difficult function used by Arjevani and Shamir (2015)

$$g_2(\theta) = \gamma \sum_{i=1}^{t} |\theta_{i+1} - \theta_i| - \beta\theta_1 + \frac{\alpha}{2}\|\theta\|_2^2,$$

on the two extremal nodes i_0 and i_1 in order to ensure that communication is necessary between the most distant nodes of the network. The final function that we consider is, for all $j \in [\![1, n]\!]$,

$$f_j(\theta) = \begin{cases} \frac{\alpha}{2}\|\theta\|_2^2 + n\left[\gamma \sum_{i=1}^{k} |\theta_{2i+1} - \theta_{2i}| - \beta\theta_1\right] & \text{if } j = i_0, \\ \frac{\alpha}{2}\|\theta\|_2^2 + n\left[\gamma \sum_{i=1}^{k} |\theta_{2i} - \theta_{2i-1}| + \delta \max_{i \in [\![2k+2, 2k+1+l]\!]} \theta_i\right] & \text{if } j = i_1, \\ \frac{\alpha}{2}\|\theta\|_2^2 & \text{otherwise,} \end{cases} \tag{18}$$

where $\gamma, \delta, \beta, \alpha > 0$ and $k, l \geq 0$ are parameters of the function satisfying $2k + l < d$. The objective function is thus

$$\bar{f}(\theta) = \gamma \sum_{i=1}^{2k} |\theta_{i+1} - \theta_i| - \beta\theta_1 + \delta \max_{i \in [\![2k+2, 2k+1+l]\!]} \theta_i + \frac{\alpha}{2}\|\theta\|_2^2.$$

First, note that reordering the coordinates of θ between θ_2 and θ_{2k+1} in a decreasing order can only decrease the value function $\bar{f}(\theta)$. Hence, the optimal value θ^* verifies this constraint and

$$\bar{f}(\theta^*) = -\gamma\theta^*_{2k+1} - (\beta - \gamma)\theta^*_1 + \delta \max_{i \in [\![2k+2, 2k+1+l]\!]} \theta^*_i + \frac{\alpha}{2}\|\theta^*\|_2^2.$$

Moreover, at the optimum, all the coordinates between θ_2 and θ_{2k+1} are equal, all the coordinates between θ_{2k+2} and θ_{2k+1+l} are also equal, and all the coordinates after θ_{2k+1+l} are zero. Hence

$$\bar{f}(\theta^*) = -\gamma\theta^*_{2k+1} - (\beta - \gamma)\theta^*_1 + \delta\theta^*_{2k+2} + \frac{\alpha}{2}\left({\theta^*_1}^2 + 2k{\theta^*_{2k+1}}^2 + l{\theta^*_{2k+2}}^2\right),$$

and optimizing over $\theta^*_1 \geq \theta^*_{2k+1} \geq 0 \geq \theta^*_{2k+2}$ leads to, when $\beta \geq \gamma(1 + \frac{1}{2k})$,

$$\bar{f}(\theta^*) = \frac{-1}{2\alpha}\left[(\beta - \gamma)^2 + \frac{\gamma^2}{2k} + \frac{\delta^2}{l}\right].$$

Now note that the objective function is the sum of two functions, one function on $(\theta_1, ..., \theta_{2k+1})$ that is (i_0, i_1)-zero preserving, and another function on $(\theta_{2k+2}, ..., \theta_{2k+1+l})$ that is (i_1, i_1)-zero preserving. As a result, Lemma 23 implies that, if $l > \lfloor t-1 \rfloor + 1$, we have $\theta_{t,2k+1+l} = 0$ and thus

$$\max_{i \in [\![2k+2, 2k+1+l]\!]} \theta_{t,i} \geq 0 \,.$$

Moreover, if $2k + 1 > \left\lfloor \frac{t-1}{1+\Delta\tau} \right\rfloor + 1$, then $\theta_{t,2k+1} = 0$, and thus

$$\begin{aligned} \bar{f}(\theta_t) &\geq \min_{\theta \in \mathbb{R}^d} -(\beta - \gamma)\theta_1 + \tfrac{\alpha}{2}\|\theta\|_2^2 \\ &\geq \tfrac{-(\beta-\gamma)^2}{2\alpha} \,. \end{aligned}$$

Hence, we have,

$$\bar{f}(\theta_t) - \bar{f}(\theta^*) \geq \frac{1}{2\alpha}\left[\frac{\gamma^2}{2k} + \frac{\delta^2}{l}\right].$$

Optimizing $\bar{f}$ over a ball of radius $R \geq \|\theta^*\|_2$ thus leads to the previous approximation error bound, and we choose

$$R^2 = \|\theta^*\|_2^2 = \frac{1}{\alpha^2}\left[(\beta - \gamma)^2 + \frac{\gamma^2}{2k} + \frac{\delta^2}{l}\right].$$

Finally, the Lipschitz constant of the objective function $\bar{f}$ on a ball of radius R is

$$L_g = \beta + 2\sqrt{2k+1}\gamma + \delta + \alpha R,$$

and setting the parameters of $\bar{f}$ to $\beta = \gamma(1 + \frac{1}{\sqrt{2k}})$, $\delta = \frac{L_g}{9}$, $\gamma = \frac{L_g}{9\sqrt{k}}$, $l = \lfloor t \rfloor + 1$, and $k = \left\lfloor \frac{t}{2\Delta\tau} \right\rfloor + 1$ leads to $l > \lfloor t-1 \rfloor + 1$ and $2k + 1 > \left\lfloor \frac{t-1}{1+\Delta\tau} \right\rfloor + 1$, and

$$\bar{f}(\theta_t) - \bar{f}(\theta^*) \geq \frac{RL_g}{36}\sqrt{\frac{1}{(1 + \frac{t}{2\Delta\tau})^2} + \frac{1}{1+t}},$$

while $\bar{f}$ is L_g-Lipschitz and $\|\theta^*\|_2 \leq R$.

Moreover, $\bar{f}$ is α-strongly convex and rewriting the radius R is terms of $\alpha_g = \alpha$ and L_g gives

$$R = \frac{L_g}{3\alpha_g}\sqrt{\frac{1}{k^2} + \frac{1}{l}},$$

and thus

$$\bar{f}(\theta_t) - \bar{f}(\theta^*) \geq \frac{L_g^2}{108\alpha}\left(\frac{1}{(1 + \frac{t}{2\Delta\tau})^2} + \frac{1}{1+t}\right).$$

B.2. Smooth Lower Bounds

Both proofs of smooth lower bounds rely on splitting the function used by Nesterov to prove oracle complexities for smooth optimization (Nesterov, 2004; Bubeck, 2015).

B.2.1. Convex and Smooth Case

Let $\beta > 0$, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a graph and $i_0 \in \mathcal{V}$ and $i_1 \in \mathcal{V}$ two nodes at distance Δ. Let, for all $j \in \mathcal{V}$, $f_j : \mathbb{R}^d \to \mathbb{R}$ be the functions defined as:

$$f_j(\theta) = \begin{cases} \frac{n\beta}{8}\left[\sum_{i=1}^{k}(\theta_{2i+1} - \theta_{2i})^2 + \theta_1^2 - 2\theta_1\right] & \text{if } j = i_0 \\ \frac{n\beta}{8}\left[\sum_{i=1}^{k}(\theta_{2i} - \theta_{2i-1})^2 + \theta_{2k+1}^2\right] & \text{if } j = i_1 \\ 0 & \text{otherwise} \end{cases}, \tag{19}$$

where $k \in \mathbb{N}$ is a parameter of the function. The objective function is thus

$$\bar{f}(\theta) = \frac{\beta}{8}\left[\sum_{i=1}^{2k}(\theta_{i+1} - \theta_i)^2 + \theta_1^2 + \theta_{2k+1}^2 - 2\theta_1\right].$$

First, note that $\nabla \bar{f}(\theta) = \frac{\beta}{8}(M\theta - 2e_1)$ where $M = \begin{pmatrix} M' & 0 \\ 0 & 0 \end{pmatrix}$ and $M' \in \mathbb{R}^{(2k+1)\times(2k+1)}$ is a tridiagonal matrix with 2 on the diagonal and -1 on the upper and lower diagonals. A simple calculation shows that $0 \preceq M \preceq 4I$, and thus $\bar{f}$ is β-smooth. The optimum of $\bar{f}$ is obtained for $\theta_i^* = 1 - \frac{i}{2k+2}$, and

$$\bar{f}(\theta^*) = -\frac{\beta}{8}\left(1 - \frac{1}{2k+2}\right).$$

Moreover, since the local functions are (i_0, i_1)-zero preserving, Lemma 23 implies that, for all $i > k_t = \left\lfloor \frac{t-1}{1+\Delta\tau} \right\rfloor + 1$, one has $\theta_{t,i} = 0$, and thus

$$\bar{f}(\theta_t) \geq -\frac{\beta}{8}\left(1 - \frac{1}{k_t+1}\right).$$

Choosing $k = k_t$ directly implies

$$\bar{f}(\theta_t) - \bar{f}(\theta^*) \geq \frac{\beta}{16(k_t+1)} \geq \frac{\beta}{16\left(\frac{t-1}{1+\Delta\tau} + 2\right)}.$$

Finally, noting that $\|\theta_0 - \theta^*\|^2 = \|\theta^*\|^2 \leq \frac{2(k+1)}{3}$ leads to the desired result.

B.2.2. Strongly Convex and Smooth Case

Let $\beta \geq \alpha > 0$, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a graph and $i_0 \in \mathcal{V}$ and $i_1 \in \mathcal{V}$ two nodes at distance Δ. Let, for all $j \in \mathcal{V}$, $f_j : \ell_2 \to \mathbb{R}$ be the functions defined as:

$$f_j(\theta) = \begin{cases} \frac{\alpha}{2}\|\theta\|_2^2 + n\frac{\beta-\alpha}{8}(\theta^\top M_1\theta - 2\theta_1) & \text{if } j = i_0, \\ \frac{\alpha}{2}\|\theta\|_2^2 + n\frac{\beta-\alpha}{8}\theta^\top M_2\theta & \text{if } j = i_1, \\ \frac{\alpha}{2}\|\theta\|_2^2 & \text{otherwise,} \end{cases} \tag{20}$$

where $M_2 : \ell_2 \to \ell_2$ is the infinite block diagonal matrix with $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ on the diagonal, and $M_1 = \begin{pmatrix} 1 & 0 \\ 0 & M_2 \end{pmatrix}$. First, note that, since $0 \preceq M_1 + M_2 \preceq 4I$, $\bar{f} = \frac{1}{n}\sum_{i=1}^n f_i$ is α-strongly convex and β-smooth. Since the local functions are (i_0, i_1)-zero preserving, Lemma 23 implies that, for all $i > k_t = \left\lfloor \frac{t-1}{1+\Delta\tau} \right\rfloor + 1$, one has $\theta_{t,i} = 0$, and thus

$$\|\theta_t - \theta^*\|_2^2 \geq \sum_{i=k_t+1}^{+\infty} \theta_i^{*2}. \tag{21}$$

Since $\bar{f}$ is α-strongly convex, we also have

$$\bar{f}(\theta_t) - \bar{f}(\theta^*) \geq \frac{\alpha}{2}\|\theta_t - \theta^*\|_2^2. \tag{22}$$

Finally, the solution of the global problem $\min_\theta \bar{f}(\theta)$ is $\theta_i^* = \left(\frac{\sqrt{\beta}-\sqrt{\alpha}}{\sqrt{\beta}+\sqrt{\alpha}}\right)^i$. Combining this result with Equations 17, 21 and 22 leads to the desired inequality.

Appendix C. Proof of Decentralized Lower Bounds (Theorems 14, 15, 16 and 17)

Lower bounds for decentralized optimization are proved using the centralized setting on linear graphs and splitting the worst-case objective functions on multiple nodes. This allows to have both local characteristics approximately equal to global characteristics (i.e., $L_\ell \approx L_g$, $\alpha_\ell \approx \alpha_g$, $\beta_\ell \approx \beta_g$ and $\kappa_\ell \approx \kappa_g$) and the mixing time approximately equal to the diameter of the graph.

Lemma 24 *Let $\widetilde{\Delta} \geq 1$. There exists a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of size n and a gossip matrix $W \in \mathbb{R}^{n\times n}$ on this graph such that its mixing time is $\widetilde{\Delta}$. Moreover, there exists two subsets of nodes $I_0, I_1 \subset \mathcal{V}$ of size $|I_0| = |I_1| \geq n/4$ and such that*

$$d(I_0, I_1) \geq \frac{\sqrt{2}}{3}\widetilde{\Delta}, \tag{23}$$

where $d(I_0, I_1)$ is the shortest-path distance in $\mathcal{G}$ between the two sets.

Proof Let $\gamma = 1/\widetilde{\Delta}^2$ be the (desired) eigengap of the gossip matrix. First of all, when $\gamma \geq 1/3$, we consider the totally connected network of 3 nodes, reweight only the edge (v_1, v_3) by $a \in [0,1]$, and let W_a be its Laplacian matrix. If $a = 1$, then the network is totally connected and $\gamma(W_a) = 1$. If, on the contrary, $a = 0$, then the network is a linear graph and $\gamma(W_a) = 1/3$. Thus, by continuity of the eigenvalues of a matrix, there exists a value $a \in [0,1]$ such that $\gamma(W_a) = \gamma$ and Equation 23 is trivially verified. Otherwise, let $x_n = \frac{1-\cos(\frac{\pi}{n})}{1+\cos(\frac{\pi}{n})}$ be a decreasing sequence of positive numbers. Since $x_3 = 1/3$ and $\lim_n x_n = 0$, there exists $n \geq 3$ such that $x_n \geq \gamma > x_{n+1}$. Let $\mathcal{G}$ be the linear graph of size n ordered from node v_1 to v_n, and weighted with $w_{i,i+1} = 1 - a\mathbb{1}\{i = 1\}$. If we take W_a as the Laplacian of the weighted graph $\mathcal{G}$, a simple calculation gives that, if $a = 0$, $\gamma(W_a) = x_n$ and, if $a = 1$, the network is disconnected and $\gamma(W_a) = 0$. Thus, there exists a value $a \in [0,1]$ such that $\gamma(W_a) = \gamma$. By definition of n, one has

$$\gamma > x_{n+1} \geq \frac{2}{(n+1)^2}\,. \tag{24}$$

We now consider $I_0 = [\![1, m]\!]$ and $I_1 = [\![n-m+1, n]\!]$ where $m = \lfloor \frac{n+1}{3} \rfloor$. When $\gamma < 1/3$, $\mathcal{G}$ is linear and the distance $d(I_0, I_1)$ between the two sets I_0 and I_1 is thus bounded by

$$d(I_0, I_1) = n - 2m + 1 \geq \frac{n+1}{3},$$

and using Equation 24 leads to the desired inequality. For $\gamma \geq 1/3$, $\mathcal{G}$ is a triangle and Equation 23 also trivially holds. Finally, since, for all $\gamma \in (0,1]$, $n \geq 3$, we have $m = \lfloor \frac{n+1}{3} \rfloor \geq \frac{n}{4}$. ■

Since all centralized lower bounds depend on splitting the objective function $\bar{f}(\theta) = \frac{f_1(\theta)+f_2(\theta)}{2}$ on two nodes at distance Δ, Lemma 24 allows to replace Δ by $\frac{\sqrt{2}}{3}\widetilde{\Delta}$. Finally, all lower bounds are obtained by splitting the functions f_1 and f_2 on all I_0 and I_1 nodes, respectively, in order to decrease local regularity characteristics.

C.1. Non-Smooth Lower Bounds

We now consider the local functions of Equation 18 split on I_0 and I_1 of Lemma 24:

$$f_j(\theta) = \begin{cases} \frac{\alpha}{2}\|\theta\|_2^2 + \frac{n}{m}\left[\gamma \sum_{i=1}^k |\theta_{2i} - \theta_{2i-1}| + \delta \max_{i \in [\![2k+2, 2k+1+l]\!]} \theta_i\right] & \text{if } j \in I_0, \\ \frac{\alpha}{2}\|\theta\|_2^2 + \frac{n}{m}\left[\gamma \sum_{i=1}^k |\theta_{2i+1} - \theta_{2i}| - \beta\theta_1\right] & \text{if } j \in I_1, \\ \frac{\alpha}{2}\|\theta\|_2^2 & \text{otherwise.} \end{cases}$$

The average function $\bar{f}$ remains unchanged and the time to communicate a vector between a node of I_0 and a node of I_1 is at least $d(I_0, I_1)\tau$. Thus, the same result as Theorem 8 holds with $\Delta = d(I_0, I_1)$. Moreover, $\alpha_\ell = \alpha = \alpha_g$ and, since $m \geq n/4$, the local Lipschitz constant L_ℓ is bounded by

$$L_\ell \leq \sqrt{\frac{2n}{m}} L_g \leq 3L_g\,.$$

Replacing L_g, α_g and Δ by, respectively, $L_\ell/3$, α_ℓ and $\frac{\sqrt{2}}{3}\widetilde{\Delta}$ in Equation 7 and Equation 8 leads to the desired result.

C.2. Smooth Lower Bounds

For the smooth and convex setting, we consider the local functions of Equation 19 split on I_0 and I_1 of Lemma 24:

$$f_j(\theta) = \begin{cases} \frac{n\beta}{8m}\left[\sum_{i=1}^k (\theta_{2i+1} - \theta_{2i})^2 + \theta_1^2 - 2\theta_1\right] & \text{if } j \in I_0, \\ \frac{n\beta}{8m}\left[\sum_{i=1}^k (\theta_{2i} - \theta_{2i-1})^2 + \theta_{2k+1}^2\right] & \text{if } j \in I_1, \\ 0 & \text{otherwise.} \end{cases}$$

The average function $\bar{f}$ remains unchanged and the time to communicate a vector between a node of I_0 and a node of I_1 is at least $d(I_0, I_1)\tau$. Thus, the same result as Theorem 11 holds with $\Delta = d(I_0, I_1)$. Moreover, since $m \geq n/4$, the local smoothness β_ℓ is bounded by

$$\beta_\ell = \max_i \beta_i = \frac{n}{2m}\beta_g \leq 2\beta_g\,.$$

Replacing β_g and Δ by, respectively, $\beta_\ell/2$ and $\frac{\sqrt{2}}{3}\widetilde{\Delta}$ in Equation 9 leads to the desired result.

For the smooth and strongly convex setting, we consider the local functions of Equation 20 split on I_0 and I_1 of Lemma 24:

$$f_j(\theta) = \begin{cases} \frac{\alpha}{2}\|\theta\|_2^2 + \frac{n}{m}\frac{\beta-\alpha}{8}(\theta^\top M_1 \theta - 2\theta_1) & \text{if } j \in I_0, \\ \frac{\alpha}{2}\|\theta\|_2^2 + \frac{n}{m}\frac{\beta-\alpha}{8}\theta^\top M_2 \theta & \text{if } j \in I_1, \\ \frac{\alpha}{2}\|\theta\|_2^2 & \text{otherwise.} \end{cases}$$

The average function $\bar{f}$ remains unchanged and the time to communicate a vector between a node of I_0 and a node of I_1 is at least $d(I_0, I_1)\tau$. Thus, the same result as Theorem 12 holds with $\Delta = d(I_0, I_1)$. Moreover, $\alpha_\ell = \alpha = \alpha_g$ and, since $m \geq n/4$, the local condition number κ_ℓ is bounded by

$$\kappa_\ell = \frac{\max_i \beta_i}{\min_i \alpha_i} = \frac{\alpha + \frac{n}{2m}(\beta - \alpha)}{\alpha} \leq 2\kappa_g - 1\,.$$

Replacing κ_g and Δ by, respectively, $\kappa_\ell/2$ and $\frac{\sqrt{2}}{3}\widetilde{\Delta}$ in Equation 10 leads to the desired result.

Appendix D. Proof of the Convergence Rate of MSPD (Theorem 18)

Theorem 1 (b) in Chambolle and Pock (2011) implies that, provided $\tau\sigma\lambda_1(W) < 1$, the algorithm with exact proximal step leads to a restricted primal-dual gap

$$\sup_{\|\Lambda'\|_F \leq c} \left\{\frac{1}{n}\sum_{i=1}^n f_i(\theta_i) - \operatorname{tr} \Lambda'^\top \Theta A\right\} - \inf_{\Theta' \in \mathcal{K}^n} \left\{\frac{1}{n}\sum_{i=1}^n f_i(\theta_i') - \operatorname{tr} \Lambda^\top \Theta' A\right\}$$

of

$$\varepsilon = \frac{1}{2t}\Big(\frac{nR^2}{\eta} + \frac{c^2}{\sigma}\Big).$$

This implies that our candidate Θ is such that

$$\frac{1}{n}\sum_{i=1}^n f_i(\theta_i) + c\|\Theta A\|_F \leq \inf_{\Theta' \in \mathcal{K}^n} \Big\{ \frac{1}{n}\sum_{i=1}^n f_i(\theta'_i) + c\|\Theta' A\|_F + \varepsilon \Big\}.$$

Let θ be the average of all θ_i. We have:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n f_i(\theta) &\leq \frac{1}{n}\sum_{i=1}^n f_i(\theta_i) + \frac{1}{n}\sum_{i=1}^n L_i\|\theta_i - \theta\| \\
&\leq \frac{1}{n}\sum_{i=1}^n f_i(\theta_i) + \frac{1}{\sqrt{n}}\sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2} \cdot \|\Theta(I - 11^\top/n)\|_F \\
&\leq \frac{1}{n}\sum_{i=1}^n f_i(\theta_i) + \frac{1}{\sqrt{n}}\sqrt{\frac{\frac{1}{n}\sum_{i=1}^n L_i^2}{\lambda_{n-1}(W)}} \cdot \|\Theta A\|_F.
\end{aligned}$$

Thus, if we take $c = \frac{1}{\sqrt{n}}\sqrt{\frac{\frac{1}{n}\sum_{i=1}^n L_i^2}{\lambda_{n-1}(W)}}$, we obtain

$$\frac{1}{n}\sum_{i=1}^n f_i(\theta) \leq \frac{1}{n}\sum_{i=1}^n f_i(\theta_*) + \varepsilon,$$

and we thus obtain an ε-minimizer of the original problem.

We have

$$\varepsilon \leq \frac{1}{2T}\Big(\frac{nR^2}{\eta} + \frac{\frac{1}{\lambda_{n-1}(W)}\frac{1}{n^2}\sum_{i=1}^n L_i^2}{\sigma}\Big)$$

with the constraint $\sigma\eta\lambda_1(W) < 1$. This leads to, with the choice

$$\eta = nR\sqrt{\frac{\lambda_{n-1}(W)/\lambda_1(W)}{\sum_{i=1}^n L_i^2/n}}$$

and taking σ to the limit $\sigma\eta\lambda_1(W) = 1$, to a convergence rate of

$$\varepsilon = \frac{1}{T}R\sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2}\sqrt{\frac{\lambda_1(W)}{\lambda_{n-1}(W)}}\,.$$

Since we cannot use the exact proximal operator of f_i, we instead approximate it. If we approximate (with the proper notion of gap (Chambolle and Pock, 2011, Eq. (11))) each $\operatorname{argmin}_{\theta_i \in \mathcal{K}} f_i(\theta_i) + \frac{n}{2\eta}\|\theta_i - z\|^2$ up to δ_i, then the overall added gap is $\frac{1}{n}\sum_{i=1}^n \delta_i$. If we do M steps of subgradient descent then the associated gap is $\delta_i = \frac{L_i^2\eta}{nM}$ (standard result for strongly convex subgradient (Lacoste-Julien et al., 2012)). Therefore the added gap is

$$\frac{1}{M}R\sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2}\sqrt{\frac{\lambda_{n-1}(W)}{\lambda_1(W)}}\,.$$

Therefore after T communication steps, i.e., communication time $T\tau$ plus MT subgradient evaluations, i.e., time MT, we get an error of

$$\Big(\frac{\widetilde{\Delta}}{T} + \frac{1}{M\widetilde{\Delta}}\Big)RL_\ell\,,$$

where $\widetilde{\Delta} = 1/\sqrt{\gamma(W)} = \sqrt{\lambda_1(W)/\lambda_{n-1}(W)}$. Thus to reach ε, it takes

$$\left\lceil \frac{2RL_\ell\widetilde{\Delta}}{\varepsilon} \right\rceil \left\lceil \frac{2RL_\ell}{\varepsilon\widetilde{\Delta}} \right\rceil + \left\lceil \frac{2RL_\ell\widetilde{\Delta}}{\varepsilon} \right\rceil \tau\,,$$

which leads to the desired result.

References

Noga Alon and Vitali D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, series B*, 38:73–88, 1985.

Mario Arioli and Jennifer Scott. Chebyshev acceleration of iterative refinement. *Numerical Algorithms*, 66(3):591–608, 2014.

Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1756–1764, 2015.

Qing Ling Aryan Mokhtari, Wei Shi and Alejandro Ribeiro. A decentralized second-order method with exact linear convergence rate for consensus optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(4):507–522, 2016.

Winfried Auzinger. *Iterative Solution of Large Linear Systems*. Lecture notes, TU Wien, 2011.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Complexity of highly parallel non-smooth convex optimization. *arXiv e-prints*, 2019.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40 (1):120–145, May 2011.

John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012a.

John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012b.

Darina Dvinskikh and Alexander Gasnikov. Decentralized and Parallelized Primal and Dual Accelerated Methods for Stochastic Convex Programming Problems. *arXiv e-prints*, 2019.

Lie He, An Bian, and Martin Jaggi. Cola: Decentralized linear learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4536–4546, 2018.

Niao He, Anatoli Juditsky, and Arkadi Nemirovski. Mirror prox algorithm for multi-term composite minimization and semi-separable problems. *Computational Optimization and Applications*, 61(2):275–319, 2015.

Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. An accelerated decentralized stochastic proximal algorithm for finite sums. *arXiv e-prints*, 2019a.

Hadrien Hendrikx, Laurent Massoulié, and Francis Bach. Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019b.

Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3068–3076, 2014.

Dušan Jakovetić. A unification and generalization of exact distributed first-order methods. *IEEE Transactions on Signal and Information Processing over Networks*, 5(1):31–46, 2019.

Dušan Jakovetić, Joao Xavier, and José M. F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

Dušan Jakovetić, José M. F. Moura, and Joao Xavier. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60(4):922–936, 2015.

Elizabeth Hou Jesús Arroyo. Efficient distributed estimation of inverse covariance matrices. In *IEEE Statistical Signal Processing Workshop (SSP)*, pages 1–5, 2016.

Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5904–5914, 2017.

Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning (ICML)*, pages 3478–3487, 2019.

Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an $O(1/t)$ convergence rate for the projected stochastic subgradient method. *arXiv e-prints*, 2012.

Guanghui Lan, Soomin Lee, and Yi Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *arXiv e-prints*, 2017.

Jason D. Lee, Qiang Liu, Yuekai Sun, and Jonathan E. Taylor. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(5):1–30, 2017.

Aryan Mokhtari and Alejandro Ribeiro. DSA: Decentralized double stochastic averaging gradient algorithm. *Journal of Machine Learning Research*, 17(1):2165–2199, 2016.

Jean-Jacques Moreau. Proximité et dualité dans un espace Hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

Angelia Nedic and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

Angelia Nedic and Alex Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Transactions on Automatic Control*, 61(12):3936–3947, 2016.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Angelia Nedic, Alex Olshevsky, and Wei Shi. Linearly convergent decentralized consensus optimization over directed networks. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 485–489, 2016.

Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4): 2597–2633, 2017.

Yurii Nesterov. *Introductory Lectures on Convex Optimization : a Basic Course*. Kluwer Academic Publishers, 2004.

Shi Pu, Wei Shi, Jinming Xu, and Angelia Nedic. A push-pull gradient method for distributed optimization in networks. In *IEEE Conference on Decision and Control (CDC)*, pages 3385–3390, 2018.

Guannan Qu and Na Li. Accelerated distributed nesterov gradient descent for smooth and strongly convex functions. In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 209–216, 2016.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning (ICML)*, pages 3027–3036, 2017.

Kevin Scaman, Francis Bach, Sebastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2740–2749, 2018.

Necdet Serhat Aybat, Alireza Fallah, Mert Gurbuzbalaban, and Asuman Ozdaglar. A Universally Optimal Multistage Accelerated Stochastic Gradient Method. *arXiv e-prints*, 2019.

Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 163–171, 2014.

Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Sebastian U. Stich. Local SGD converges fast and communicates little. *arXiv e-prints*, 2018.

Lu Tian and Quanquan Gu. Communication-efficient Distributed Sparse Linear Discriminant Analysis. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1178–1187, 2017.

César A. Uribe, Soomin Lee, Alexander Gasnikov, and Angelia Nedic. A dual approach for optimal algorithms in distributed optimization over networks. *arXiv e-prints*, 2018.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1195–1204, 2019.

Ermin Wei and Asuman Ozdaglar. Distributed alternating direction method of multipliers. In *IEEE Conference on Decision and Control (CDC)*, pages 5445–5450, 2012.

Chenguang Xi and Usman A. Khan. Dextra: A fast algorithm for optimization over directed graphs. *IEEE Transactions on Automatic Control*, 62(10):4980–4993, Oct 2017.

Ran Xin, Dušan Jakovetić, and Usman A. Khan. Distributed nesterov gradient methods over arbitrary graphs. *IEEE Signal Processing Letters*, 26(8):1247–1251, Aug 2019.

Journal of Machine Learning Research 20 (2019) 1-56 Submitted 12/18; Revised 7/19; Published 10/19

Learning by Unsupervised Nonlinear Diffusion

Mauro Maggioni MAURO.MAGGIONI@JHU.EDU
Department of Mathematics, Department of Applied Mathematics and Statistics,
Mathematical Institute of Data Sciences, Institute of Data Intensive Engineering and Science,
Johns Hopkins University, Baltimore, MD 21218, USA

James M. Murphy JM.MURPHY@TUFTS.EDU
Department of Mathematics
Tufts University, Medford, MA 02155, USA

Editor: Aapo Hyvärinen

Abstract

This paper proposes and analyzes a novel clustering algorithm, called *learning by unsupervised nonlinear diffusion (LUND)*, that combines graph-based diffusion geometry with techniques based on density and mode estimation. LUND is suitable for data generated from mixtures of distributions with densities that are both multimodal and supported near nonlinear sets. A crucial aspect of this algorithm is the use of time of a data-adapted diffusion process, and associated diffusion distances, as a scale parameter that is different from the local spatial scale parameter used in many clustering algorithms. We prove estimates for the behavior of diffusion distances with respect to this time parameter under a flexible nonparametric data model, identifying a range of times in which the mesoscopic equilibria of the underlying process are revealed, corresponding to a gap between within-cluster and between-cluster diffusion distances. These structures may be missed by the top eigenvectors of the graph Laplacian, commonly used in spectral clustering. This analysis is leveraged to prove sufficient conditions guaranteeing the accuracy of LUND. We implement LUND and confirm its theoretical properties on illustrative data sets, demonstrating its theoretical and empirical advantages over both spectral and density-based clustering.

Keywords: unsupervised learning, clustering, spectral graph theory, manifold learning, diffusion geometry

1. Introduction

Unsupervised learning is a central problem in machine learning, requiring that data be analyzed without a priori knowledge of any class labels. A common unsupervised problem is *clustering*, in which the data is to be partitioned into clusters so that each

©2019 Mauro Maggioni and James M. Murphy.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-873.html.

cluster contains similar points and distinct clusters are sufficiently separated. Even with suitable definitions of "similarity" and "separation", this problem is typically ill-posed, requiring various geometric, analytic, topological, and statistical assumptions on the data and measurement method be imposed to make it tractable. Feature extraction is often combined with these standard methods (e.g. K-means) to improve clustering performance.

In particular, *spectral methods* construct graphs representing data, and use the spectral properties of the resulting graph Laplacian to produce structure-revealing features in the data. Graphs often encode pairwise similarities between points, typically at a local "spatial" scale, often determined by a parameter σ. For example only points x_i, x_j within distance 4σ of each other may be connected, with weight $\exp(-\|x_i - x_j\|_2^2/\sigma^2)$. From the graph, global features on the data may be derived, for example by considering the eigenfunctions of the random walk on the graph. Alternatively, graphs may be used to introduce data-adaptive distances, such as *diffusion distances*, which are associated to random walks and diffusion processes on graphs. Diffusion distances do not depend only on the graph itself, but also on a time parameter t that determines a scale *on the graph* at which these distances are considered, related to the time of diffusion or random walk. Choosing σ in graph-based algorithms, and both σ and t in the case of diffusion distances, is important in both theory and applications. However, their role is well-understood only in certain regimes (e.g. asymptotically for $\sigma, t \to 0^+$) which are of interest in some problems (e.g. manifold learning) but not necessarily for clustering.x

We propose the *Learning by Unsupervised Nonlinear Diffusion (LUND)* scheme for clustering, which combines diffusion distances and density estimation to efficiently cluster data generated from a nonparametric model. At the same time, we advance the understanding of the relationship between the local "spatial" scale parameter σ and the diffusion time parameter t in the context of clustering, demonstrating how the role of t can be exploited to successfully cluster data sets for which K-means, spectral clustering, or density-based clustering methods fail. We provide quantitative bounds and guarantees on the performance of the proposed clustering algorithm for data that may be highly nonlinear (i.e. non-convex, elongated, ellipsoidal, etc.) and of variable density.

1.1 Major Contributions and Outline

This article makes two major contributions. First, *explicit estimates on diffusion distances for nonparametric clustered data* are proved: we obtain lower bounds for the diffusion distance (see Definition 2.1, or Coifman et al. (2005)) between clusters, and upper bounds on the diffusion distance within clusters, as a function of the time parameter t and suitable properties of the clusters. These bounds yield a mesoscopic—not too small, not too large—diffusion time-scale at which diffusion distances separate clusters clearly and cohere points in the same cluster. These results, among other things, show how the role of the time parameter, which controls the scale "on the

data" of the diffusion distances, is very different from the commonly-used scaling parameter σ in the construction of the underlying graph, which is a local spatial scale measured in the ambient space. Relationships between t and σ are well-understood in the asymptotic case of $n \to +\infty$, $\sigma \to 0^+$ (at an appropriate rate with n; see Coifman et al. (2005), Lafon et al. (2006), and Von Luxburg (2007)) and $t \to 0^+$ (essentially Varadhan's lemma applied to diffusions on a manifold; see Den Hollander (2008), Jones et al. (2008), and references therein). These asymptotic relationships at small scales imply that the choice of t is essentially irrelevant, since in these limits diffusion distances are essentially geodesic distances. However, the clustering phenomena we are interested in are far from this regime, and we show that the interplay between t, σ, and n becomes crucial.

Second, *the LUND clustering scheme* is proposed and shown to enjoy performance guarantees for clustering of certain non-parametric mixture models. We prove sufficient conditions for LUND to correctly determine the number of clusters in the data and to have low clustering error. Computationally, we present an efficient algorithm implementing LUND, which scales near-linearly in the number of points n, in the ambient dimension D, and exponentially in the intrinsic dimension of the data. We verify the properties of the LUND scheme and algorithm on synthetic data, studying the relationships between the different parameters in LUND, in particular between σ and t, and compare with popular and related the graph-based *spectral clustering* and *fast search and find of density peaks clustering (FSFDPC)* (Rodriguez and Laio, 2014) algorithms, illustrating weaknesses of these methods and corresponding advantages of LUND. LUND may be understood as a combination of these two methods, in that it integrates diffusion distances (which are graph-based) and an outlier robustness procedure into the FSFDPC framework, which uses Euclidean distances. Indeed, our experiments illustrate how LUND combines the benefits of graph-based and density-based methods.

The outline of the article is as follows. Background is presented in Section 2. In Section 3, motivational data sets and a summary of the theoretical results are presented and discussed. Theoretical comparisons with related clustering methods are also made in Section 3. Estimates on diffusion distances are proved in Section 4. Performance guarantees for the LUND algorithm are proved in Section 5. Numerical experiments and computational complexity are discussed in Section 6. Conclusions and future research directions are given in Section 7.

2. Background

2.1 Background on Clustering

Given the wide variety of data of interest to scientific practitioners, many approaches to clustering have been developed, whose performance is often wildly variable and data-dependent.

2.1.1 K-Means

A classical and popular clustering algorithm is *K-means* (Steinhaus, 1957; Friedman et al., 2001) and its variants (Ostrovsky et al., 2006; Arthur and Vassilvitskii, 2007; Park and Jun, 2009), which is often used in conjunction with feature extraction methods. K-means partitions the data into K (a parameter) groups, $\{C_k\}_{k=1}^K$, chosen to minimize within-cluster dissimilarity: $C^* = \arg\min_{\{C_k\}_{k=1}^K} \sum_{k=1}^K \sum_{x \in C_k} \|x - \bar{x}_k\|_2^2$, where $\bar{x}_k$ is the mean of the kth cluster (for a given partition, it is the minimizer of the least squares cost in the inner sum). While popular, K-means and its variants may perform poorly for data sets that are not the union of well-separated, near-spherical clusters, and are sensitive to outliers.

2.1.2 Hierarchical Clustering Methods

Hierarchical methods iteratively merge or split clusters in order to produce a multiscale family of partitions known as a dendrogram (Friedman et al., 2001). More precisely, a dendrogram for n data points is a family of clusterings $\{\mathcal{C}_i\}_{i=1}^n$ such that $\mathcal{C}_1$ is the clustering of n singleton clusters, and $\mathcal{C}_i$ is related to $\mathcal{C}_{i+1}$ in that the two clusters minimizing some linkage function in $\mathcal{C}_i$ are merged in $\mathcal{C}_{i+1}$. *Single linkage clustering (SLC)* (Sneath, 1957; Gower and Ross, 1969; Friedman et al., 2001) is a particular hierarchical clustering method that iteratively merges clusters according to the linkage function $\mathcal{L}_{\text{SLC}}(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} \|x_1 - x_2\|_2$; metrics other than the ℓ^2 norm may be used. For clusters that are well-separated, single linkage clustering is known to perform well (Arias-Castro, 2011), despite lack of strong statistical consistency in dimensions greater than 1 (Hartigan, 1981).

2.1.3 Density and Mode-Based Methods

Density and mode-based clustering methods detect regions of high-density and low-density to determine clusters. The *DBSCAN* (Ester et al., 1996) and *DBCLASD* (Xu et al., 1998) algorithms assign to the same cluster points that are close and have many near neighbors, and flag as outliers points that lie alone in low-density regions. The *mean-shift* algorithm and variants (Fukunaga and Hostetler, 1975; Comaniciu and Meer, 2002; Chacón, 2012; Genovese et al., 2016) push points towards regions of high-density, and associate clusters with these high-density points, sometimes called *modes*. Both DBSCAN and mean-shift clustering suffer from a lack of robustness to outliers and depend strongly on parameter choices.

The *fast search and find of density peaks clustering algorithm* (FSFDPC) (Rodriguez and Laio, 2014) proposes to address these weaknesses. This method characterizes cluster modes as points that are far in Euclidean distance from points of higher density. FSFDPC has been widely applied (Spitzer et al., 2015; Wiwie et al., 2015; Sun et al., 2015; Rossant et al., 2016; Wang et al., 2016; Jia et al., 2016), and correctly clusters the data in Figure 1. However, we show that the standard FSFDPC method, while very popular in scientific applications, does not correctly cluster data unless

strong geometric and statistical assumptions on the data are satisfied. The main reason is that Euclidean distances are used to find modes, which is inappropriate for data drawn from mixtures of distributions supported near nonlinear sets (see, for example, Figure 18). Moreover, FSFDPC is not robust to outliers, which may be far from other points but be of low-density.

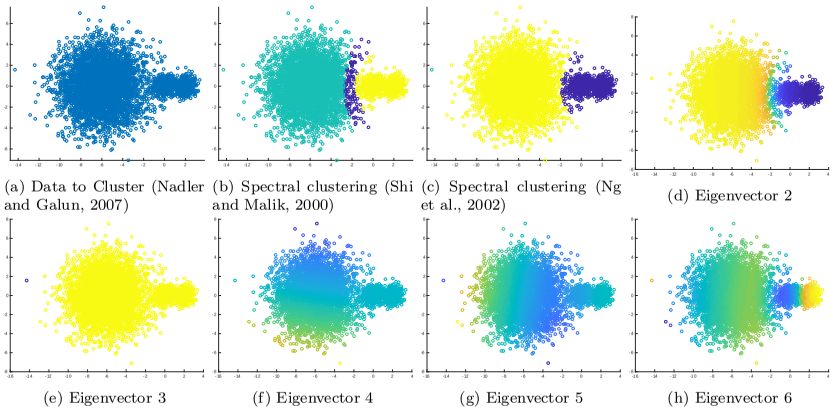
2.1.4 Spectral Methods

Spectral clustering methods compute features that reveal the structure of data that may deviate from the spherical, Gaussian shapes ideal for K-means, and in particular may be nonlinear or elongated in shape. This is done by building local connectivity graphs on the data that encode pairwise similarities between points, then computing a spectral decomposition of adjacency or random walk or Laplacian operators defined on this graph.

Let $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^D$ be a set of points to cluster. Let $\mathcal{G}$ be a graph with vertices corresponding to points of X and edges stored in an $n \times n$ symmetric weight matrix $\mathbf{W}$. Often one chooses $\mathbf{W}_{ij} = \mathcal{K}(x_i, x_j)$ for some (symmetric, often radial and rapidly decaying) nonnegative *kernel* $\mathcal{K} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, such as $\mathcal{K}(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2)$ for some choice of scaling parameter $\sigma > 0$. The graph $\mathcal{G}$ may be fully connected, or it may be a nearest neighbors graph with respect to some metric. Let $\mathbf{D}$ be the diagonal matrix $\mathbf{D}_{ii} := \sum_{j=1}^n \mathbf{W}_{ij}$. The *graph Laplacian* is constructed as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. One then normalizes $\mathbf{L}$ to acquire either the *random walk Laplacian* $\mathbf{L}_{\mathrm{RW}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$ or the *symmetric normalized Laplacian* $\mathbf{L}_{\mathrm{SYM}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\,\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$. We focus on $\mathbf{L}_{\mathrm{SYM}}$ in what follows. It can be shown that $\mathbf{L}_{\mathrm{SYM}}$ has real eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n \leq 2$ and corresponding eigenvectors $\{\phi_i\}_{i=1}^n$. The original data X can be clustered by clustering the embedded data $x_i \mapsto (\phi_1(x_i), \phi_2(x_i), \ldots, \phi_M(x_i))$ for an appropriate choice of $M \leq n$. In this step typically K-means is used, though Gaussian mixture models may (and perhaps should) be used, as they enjoy, unlike K-means, a suitably-defined statistical consistency guarantee in the infinite sample limit (Athreya et al., 2017).

Spectral clustering relaxes a graph-cut problem: for a collection of subsets $X_1, \ldots, X_K \subset X$, the corresponding *normalized cut* is $\mathrm{Ncut}(X_1, \ldots, X_K) = \sum_{k=1}^K \mathrm{cut}(X_k, X_k^c)/\mathrm{vol}(X_k)$,where $\mathrm{cut}(A, B) = \sum_{x_i \in A, x_j \in B} \mathbf{W}_{ij}$, $\mathrm{vol}(A) = \sum_{x_i \in A} \sum_{j=1}^n \mathbf{W}_{ij}$. Minimizing Ncut yields clusters that are simultaneously separated and balanced (Shi and Malik, 2000). This NP-hard problem may be relaxed by analyzing the first K eigenvectors of $\mathbf{L}_{\mathrm{SYM}}$ (Shi and Malik, 2000; Ng et al., 2002), or via a semidefinite programming problem (Ling and Strohmer, 2018).

Weaknesses of spectral clustering were scrutinized by Nadler and Galun (2007). They show the top eigenvectors of the random walk matrix $\mathbf{P}$—defined on $\{x_i\}_{i=1}^n$ sampled from $p(x)$ proportional to $e^{-U(x)/2}$ for some potential function $U(x)$—converge under a suitable scaling as $n \to \infty$ to the top eigenfunctions of the Fokker-Planck operator $\mathcal{L}\psi(x) = \Delta\psi - \nabla\psi \cdot \nabla U = -\mu\psi(x)$. The characteristic time scales of the stochastic differential equation (SDE) $\dot{x}(t) = -\nabla U(x) + \sqrt{2}\dot{w}(t)$, w a Wiener process, determine



(a) Data to Cluster (Nadler and Galun, 2007) (b) Spectral clustering (Shi and Malik, 2000) (c) Spectral clustering (Ng et al., 2002) (d) Eigenvector 2

(e) Eigenvector 3 (f) Eigenvector 4 (g) Eigenvector 5 (h) Eigenvector 6

Figure 1: In (a), three Gaussians of essentially the same density are shown. Results of spectral clustering are shown in (b) (Shi and Malik, 2000) and (c) (Ng et al., 2002). In (d) - (h), the first five non-trivial eigenvectors are shown. As noted by Nadler and Galun (2007), the underlying density for this data yields a Fokker-Planck operator whose low-energy eigenfunctions cannot distinguish between the two smaller clusters, thus preventing spectral clustering from succeeding: higher energy eigenfunctions are required. For this example, the sixth non-trivial eigenvector localizes sufficiently on the small clusters to allow for correct determination of the cluster structure; this eigenvector is not used in traditional spectral clustering algorithms.

the structure of the leading eigenfunctions of $\mathcal{L}$ (Gardiner, 2009): they correspond to the time scales of the slowest transitions between different clusters and the equilibrium times within clusters. The relationships between these quantities determine which eigenfunctions of $\mathcal{L}$ (or $\mathbf{P}$) reveal the cluster structure in the data. Gavish and Nadler (2013) further analyze related connections between normalized cuts and cluster exit times. Nadler and Galun (2007) present data which cannot be clustered with spectral clustering (Shi and Malik, 2000; Ng et al., 2002); see Figure 1.

2.2 Background on Diffusion Distances

One of the primary tools in the proposed clustering algorithm is *diffusion distances*, a class of data-dependent distances computed by constructing Markov processes on data that capture its intrinsic structure (Coifman et al., 2005; Coifman and Lafon, 2006; Lafon et al., 2006; Coifman et al., 2008; Singer and Coifman, 2008; Rohrdanz et al., 2011; Zheng et al., 2011; Lederman and Talmon, 2018; Lederman et al., 2015; Czaja et al., 2016; Li et al., 2017). We consider diffusion on the point cloud $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^D$ via a Markov chain (Levin et al., 2009) with state space X. Let $\mathbf{P}$ be the corresponding $n \times n$ transition matrix. The following shall be referred to as the *usual assumptions* on $\mathbf{P}$: $\mathbf{P}$ is reversible, irreducible, aperiodic, and therefore ergodic. A common construction for $\mathbf{P}$, and the one we consider in the algorithmic sections of this article, is to first compute a *weight matrix* $\mathbf{W}$, where $\mathbf{W}_{ij} = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$, $i \neq$

j for some appropriate scale parameter $\sigma \in (0, \infty)$. The parameter σ encodes the interaction radius of each point: σ large allows for long-range interactions between points that are ℓ^2-far, while σ small allows only for short-range interactions. Then $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$, where $\mathbf{D}$ is the diagonal degree matrix with $\mathbf{D}_{ii} = \sum_{\ell=1}^{n} \mathbf{W}_{ij}$. This[1] row-normalizes $\mathbf{P}$: $\sum_{j=1}^{n} P_{ij} = 1,\ \forall i = 1, \ldots, n$. Since it is ergodic, $\mathbf{P}$ has a unique stationary distribution $\boldsymbol{\pi}$ satisfying $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$, given by $\boldsymbol{\pi}_i = \mathbf{D}_{ii} / \sum_{j=1}^{n} \mathbf{D}_{jj}$.

Diffusion processes on graphs lead to a data-dependent notion of distance, known as *diffusion distance* (Coifman et al., 2005; Coifman and Lafon, 2006). While the focus of the construction is on diffusion distances and the diffusion process itself, we mention that *diffusion maps* provide a way of efficiently computing and visualizing large-time diffusion distances in Euclidean space, and at the same time may be understood as a type of nonlinear dimension reduction, in which data in a high number of dimensions may be embedded in a low-dimensional space by a nonlinear coordinate transformation. In this regard, diffusion maps are related to nonlinear dimension reduction techniques such as Isomap (Tenenbaum et al., 2000), Laplacian eigenmaps (Belkin and Niyogi, 2003), and local linear embedding (Roweis and Saul, 2000), kernel PCA (with a data-adapted kernel), among many others.

Definition 2.1 *Let $X = \{x_i\}_{i=1}^{n} \subset \mathbb{R}^D$ and let $\mathbf{P}$ be a Markov process on X satisfying the usual assumptions and with stationary distribution $\boldsymbol{\pi}$. Let $\boldsymbol{\pi}_0$ be a probability distribution on X. For points $x_i, x_j \in X$, let $p_t(x_i, x_j) = (\mathbf{P}^t)_{ij}$, for some $t \in [0, \infty)$. The* diffusion distance at time t *between $x, y \in X$ is defined, for $\nu = \boldsymbol{\pi}_0 / \boldsymbol{\pi}$, by*

$$D_t(x, y) = \sqrt{\sum_{u \in X} \left(p_t(x, u) - p_t(y, u)\right)^2 \nu(u)} = ||p_t(x, \cdot) - p_t(y, \cdot)||_{\ell^2(\nu)} .$$

If the underlying graph is generated from data sampled from a low-dimensional manifold, then diffusion distance parametrizes this low-dimensional structure (Coifman et al., 2005; Jones et al., 2008; Singer et al., 2009; Singer and Wu, 2012, 2016; Talmon and Wu, 2018). Indeed, diffusion distances admit a formulation in terms of the (right) eigenfunctions of $\mathbf{P}$:

$$D_t(x, y) = \sqrt{\sum_{\ell=1}^{n} \lambda_\ell^{2t} (\psi_\ell(x) - \psi_\ell(y))^2}, \tag{2.2}$$

where $\{(\psi_\ell, \lambda_\ell)\}_{\ell=1}^{n}$ are the right eigenpairs of $\mathbf{P}$, ordered so that $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n > -1$, and noting that ψ_1 is constant by construction.

Diffusion distances are parametrized by t, which measures how long the diffusion process on $\mathcal{G}$ has run. Small t allows a small amount of diffusion, which may prevent the interesting geometry of X from being discovered, but provides detailed, fine scale information. Large t allows the diffusion process to run for so long that the fine geometry may be washed out, leaving only coarse scale information. We will relate properties of clustered data X to t.

1. Note that with some abuse of notation we denote the entries of $\mathbf{P}$ by P_{ij}, reserving the notation $\mathbf{P}_{ij}$ for block submatrices of $\mathbf{P}$ that will be introduced and used later.

3. Data Model and Overview of Main Results

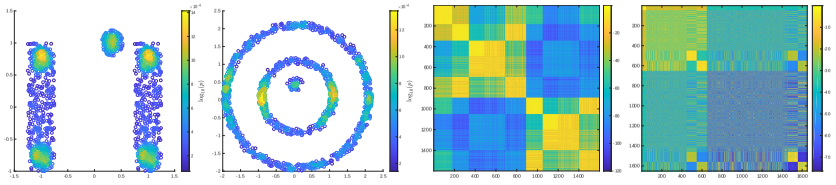
Among the main results of this article are sufficient conditions for clustering certain discrete data $X \subset \mathbb{R}^D$. The data X is modeled as a realization from a probability distribution

$$\mu = \sum_{k=1}^{K} w_k \mu_k,\ w_k \geq 0,\ \sum_{k=1}^{K} w_k = 1, \tag{3.1}$$

where each μ_k is a probability measure. Intuitively, our results require *separation and cohesion* conditions on $\{\mu_k\}_{k=1}^{K}$. That is, each μ_k is far from $\mu_{k'}$, $k \neq k'$ and connections are strong (in a suitable sense) within each μ_k. $X = \{x_i\}_{i=1}^{n}$ is generated by drawing, for each i, one of the K clusters, say k_i, according to the multinomial distribution with parameters $(w_1, \ldots, w_K)$, and then drawing x_i from μ_{k_i}. The clusters in the data are defined as the subsets of X whose samples were drawn from a particular μ_k, that is, we define the cluster $X_k := \{x_i \in X : k_i = k\}$. Given X, the goal of clustering is to estimate these X_k, and to estimate K. Throughout the theoretical analysis of this article, we will define the accuracy of a set of labels $\{Y_i\}_{i=1}^{n}$, $Y_i \in \{1, \ldots, K\}$, learned from an unsupervised algorithm to be $|\{i \mid Y_i = k_i\}|/n$, i.e. the proportion of points correctly labeled.

The model (3.1) is *nonparametric* and makes few explicit assumptions on μ. We will allow μ_k to be supported near a non-linear set (e.g. a nonconvex subset, or a submanifold in $\mathbb{R}^D$) and be multimodal (i.e. with multiple high-density regions). These features may cause prominent clustering methods to fail, e.g. K-means, which requires near-spherical or well-separated clusters (Dasgupta and Schulman, 2007; Mixon et al., 2017); Gaussian mixture models, which handle well spherical and ellipsoidal clusters, but may struggle with clusters exhibiting different, non-elliptical geometries (for example those shown in Figure 2 (b)); spectral clustering, which can fail for highly elongated clusters or clusters of different sizes and densities; density-based methods, which are sensitive to noise and clusters of very different densities, and may require care in setting parameters or implementing adaptive parameters. Two simple, motivating examples are in Figure 2. They feature variable densities, variable levels of connectivity, both within and across clusters, and (for the second example) nonlinear cluster shapes.

The estimates for the behavior of diffusion distances that we derive will then be leveraged to prove that the LUND clustering scheme correctly labels the points and also correctly estimates the number of clusters, while other clustering schemes fail to cluster these data sets correctly.



(a) Bottleneck data (b) Nonlinear data (c) $\mathbf{P}$ for data in (a). (d) $\mathbf{P}$ for data in (b).

Figure 2: Subfigures (a) and (b) show two data sets—linear and nonlinear—colored by $\log_{10}(p(x))$, where $p(x)$ is the empirical density. In (c), (d), we show the corresponding Markov transition matrices $\mathbf{P}$, with entry magnitudes shown in $\log_{10}$ scale. We sorted the rows and columns so that the structure in these matrices become apparent (of course, the algorithms are independent of the sorting). The Markov chains are ergodic, but close to being reducible. Indeed, these transition matrices show hierarchical structure, with large within-cluster transitions and uniformly small probabilities of transition between clusters. The analysis of Section 4 makes this intuition precise. The transition matrices were constructed using the Gaussian kernel as in Section 2.2, with Euclidean distances and $\sigma = .25$

3.1 Summary of Main Results

Our first result shows that the within-cluster and between-cluster diffusion distances can be controlled, as soon as $\mathbf{P}^t$ is approximately block constant in some sense. Define the (worst-case) within-cluster and between-cluster diffusion distances as:

$$D_t^{\text{in}} = \max_k \max_{x,y \in X_k} D_t(x,y), \quad D_t^{\text{btw}} = \min_{k \neq k'} \min_{x \in X_k, y \in X_{k'}} D_t(x,y). \tag{3.2}$$

Our results guarantee that D_t^{in} is small and D_t^{btw} is large in terms of a constant ϵ, for all t in some time interval $\mathcal{T}$. The interval $\mathcal{T}$ depends on both data-driven quantities of the matrix $\mathbf{P}$ (which may be understood as geometrically intrinsic to the underlying cluster structure), as well as ϵ. This simplification of Theorem 4.12 holds:

Theorem 3.3 *Let $X = \bigcup_{k=1}^K X_k$ and let $\mathbf{P}$ be a corresponding Markov transition matrix on X, inducing diffusion distances D_t. Then there exist non-negative constants $\{C_i\}_{i=1}^5$ such that the following holds: for any $\epsilon > 0$ we have*

$$C_1 \ln\left(\frac{C_2}{\epsilon}\right) < t < C_3\epsilon \quad \Longrightarrow \quad D_t^{in} \leq C_4\epsilon, \quad D_t^{btw} \geq C_5 - C_4\epsilon.$$

The constants $\{C_i\}_{i=1}^5$ are defined precisely in Section 4, but to get a sense of them, let $\mathbf{S}$ be an idealized version of $\mathbf{P}$ in which all edges between clusters are deleted and redirected back into the cluster; this will be made precise by the notion of stochastic complement in Definition 4.7. Let $\mathbf{S}^\infty$ be a block diagonal matrix consisting of K rank 1 blocks, corresponding to the equilibrium transition matrices of the the blocks of $\mathbf{S}$. The constants $\{C_i\}_{i=1}^5$ may be interpreted as follows:

- C_1, C_2 are related to the compactness or irreducibility of the clusters X_k, in terms of the rank of each block of $\mathbf{S}$. In particular, if $\mathbf{S}$ is constant on each diagonal block corresponding to a cluster, then $C_1 = 0$ and $C_2 = O(1)$, independently of n.
- C_3 is related to the transition probabilities between clusters in $\mathbf{P}$. In the idealized case that there are no transitions between any of the clusters, $C_3 = \infty$. More generally, if the probabilities of transitions between clusters are small, then C_3 will be large.
- C_4 is related to the balance of $\mathbf{S}^\infty$ relative to $\mathbf{P}$. If each row of $\mathbf{S}^\infty - \mathbf{P}$ is constant, then $C_4 = 1/\sqrt{n}$.
- C_5 is related to how balanced the equilibrium distributions are on each cluster. If $\mathbf{S}^\infty$ is block constant with blocks of the same size, then $C_5 = 1/\sqrt{n}$.

The time interval $\mathcal{T} := [C_1 \ln\left(\frac{C_2}{\epsilon}\right), C_3\epsilon]$ depends on ϵ, and there is tension in the role of ϵ between the condition $t \in \mathcal{T}$ and the conclusion that $D_t^{\text{in}} \leq C_4\epsilon, \quad D_t^{\text{btw}} \geq C_5 - C_4\epsilon$. For large ϵ, $\mathcal{T}$ may be quite wide, but the conclusion on diffusion distances is weak (or even trivial if $\epsilon > C_5/C_4$). On the other hand, ϵ small induces strong separation between within-cluster diffusion distances and between-cluster diffusion distances, at the expense of shrinking $\mathcal{T}$. Indeed, for $0 < C_1, C_2, C_3 < \infty$ fixed, $\mathcal{T}$ shrinks to the empty set as $\epsilon \to 0^+$.
Suppose indeed that the each cluster is compact (C_1 small, $C_2 = O(1)$), the clusters are well-separated (C_3 large), and balanced in the sense that $\mathbf{S}^\infty - \mathbf{P}$ is constant and $\mathbf{S}^\infty$ is block-constant with blocks of the same size ($C_4 = C_5 = 1/\sqrt{n}$). Then for $\epsilon > 0$,

$$t \in \left[C_1 \ln\left(\frac{C_2}{\epsilon}\right), C_3\epsilon\right] \implies \frac{D_t^{\text{btw}}}{D_t^{\text{in}}} \geq \frac{C_5 - C_4\epsilon}{C_4\epsilon} = O\left(\frac{1}{\epsilon}\right),$$

which suggests strong separation with diffusion distances independently of n when ϵ is small. In particular, as the clusters become more separated (C_3 increases) the time interval on which the $D_t^{\text{btw}}/D_t^{\text{in}}$ remains large widens on the right. Similarly, the more compact or irreducible the clusters become (C_1 becomes smaller), the wider the interval becomes on the left. In the ideal case that $C_1 = 0$ (clusters are localized at single points), $C_3 = \infty$ (infinite separation between clusters), the ratio $D_t^{\text{btw}}/D_t^{\text{in}} = \infty$ for all t, due to the fact that D_t^{btw} is bounded away from 0 for all t while $D_t^{\text{in}} = 0$. We remark that as long as ϵ can be taken sufficiently small, due to the geometric properties of the data as determined by C_1, C_2, C_3, the lower bound on D_t^{btw} is positive. In the idealized case when $C_4 = C_5 = 1/\sqrt{n}$, then the lower bound is positive as long as $\epsilon < 1$. We note that C_1, C_2, C_3 are close to geometrically intrinsic, as will be discussed in Section 4.1.
The *LUND scheme* characterizes modes of the clusters $\{X_k\}_{k=1}^K$ as high-density points that are far in diffusion distance from other points of high-density, regardless of the shape of the support of the distribution. Once modes are learned, remaining points are subsequently assigned to a mode in an iterative fashion. To detect the modes,

we introduce two quantities: $p(x)$ is related to data density, while $\rho_t(x)$ is related to diffusion geometry.

Let p be a kernel density estimator (KDE) on X, for example $p(x) = \frac{1}{Z}\sum_{y\in NN(x)} \exp(-\|x-y\|_2^2/\sigma_0^2)$, for some choice of σ_0 and set of nearest neighbors $NN(x)$, normalized by Z so that $\sum_{x\in X} p(x) = 1$. Given D_t defined on X, let

$$\rho_t(x) = \begin{cases} \min_{y\in X}\{D_t(x,y) \mid p(y) \geq p(x)\}, & x \neq \arg\max_{y\in X} p(y), \\ \max_{y\in X} D_t(x,y), & x = \arg\max_{y\in X} p(y). \end{cases} \tag{3.4}$$

The function ρ_t measures the diffusion distance of a point to its D_t-nearest neighbor of higher empirical density. LUND proceeds by analyzing

$$\mathcal{D}_t(x) = p(x)\rho_t(x),$$

which is large only for high-density points that are far from their nearest diffusion neighbor of higher density. The function $\mathcal{D}_t$ serves two important purposes. First, its decay estimates the number of clusters in the data. Indeed, as will be shown in Section 5, under a flexible data model, $\mathcal{D}_t$ has K very large values with the rest very small, where K is the number of latent clusters. Second, the modes of the data are estimated as the maximizers of $\mathcal{D}_t$. Once these modes have been learned, they are given unique labels. Then, in order of decreasing density, points are assigned the same label as their D_t-nearest neighbor of higher density. In this sense, the labels of the modes are distributed—from high to low density—to the rest of the data. The LUND algorithm is detailed in Algorithm 1. A simpler variant of LUND, when K is known a priori, is detailed in Algorithm 2.

Functions of $p(x), \rho_t(x)$ other than multiplication could be used to construct $\mathcal{D}_t(x)$. The primary reason to consider the multiplication of these factors is to gain robustness to outliers. Indeed, an outlying point x_o may be very far from all other points in diffusion distance, simply because its Euclidean coordinates are very far from the rest of X. In this case, one would have $\rho_t(x_o)$ very large, and $p(x_o)$ very small. By constructing $\mathcal{D}_t$ as the product of the density and diffusion geometric measurements, $\mathcal{D}_t(x_o) = p(x_o)\rho_t(x_o)$ is not large (under a suitable regime of variation between $p(x_o)$ and $\rho_t(x_o)$), ensuring that a far outlier is not be selected as a data mode. More precisely, suppose that diffusion distances and $p(x)$ are computed with the same choice of scaling parameter σ and collection of nearest neighbors, so that the stationary distribution of $\mathbf{P}$ is equal to p: for all $x \in X$, $p(x) = \boldsymbol{\pi}(x)$. Suppose that X is fixed except for the outlier point x_o, which we assume to be the element of X with lowest empirical density. Letting x_o^{NN} be the nearest neighbor of x_o of higher empirical density and $\boldsymbol{\pi}_0$ an arbitrary initial distribution,

$$\mathcal{D}_t(x_o) = \rho_t(x_o)p(x_o) = \sqrt{\sum_{\ell=1}^{n} \left(p_t(x_o,x_\ell) - p_t(x_o^{NN},x_\ell)\right)^2 \frac{\boldsymbol{\pi}_0(x_\ell)\boldsymbol{\pi}(x_o)^2}{\boldsymbol{\pi}(x_\ell)}} \leq \sqrt{n\boldsymbol{\pi}(x_o)}.$$

Algorithm 1 Learning by Unsupervised Nonlinear Diffusion (LUND) Algorithm

Input: X (data), σ_0 (kernel density bandwidth), σ (diffusion scaling parameter), t (time parameter), τ (threshold)

Output: Y (cluster assignments), $\hat{K}$ (estimated number of clusters)

Build Markov transition matrix $\mathbf{P}$ using scale parameter σ.
Compute KDE $p(x)$ for all $x \in X$ using kernel bandwidth σ_0.
Compute $\rho_t(x)$ for all $x \in X$.
Compute $\mathcal{D}_t(x) = \rho_t(x)p(x)$ for all $x \in X$.
Sort X according to $\mathcal{D}_t(x)$ in descending order as $\{x_{m_i}\}_{i=1}^n, n = |X|$.
Compute $\hat{K} = \inf\left\{k \,\middle|\, \frac{\mathcal{D}_t(x_{m_k})}{\mathcal{D}_t(x_{m_{k+1}})} > \tau\right\}$.
Assign $Y(x_{m_i}) = i,\ i = 1,\dots,\hat{K}$, and $Y(x_{m_i}) = 0,\ i = \hat{K}+1,\dots,n$.
Sort X according to $p(x)$ in decreasing order as $\{x_{\ell_i}\}_{i=1}^n$.
for $i = 1 : n$ **do**
  if $Y(x_{\ell_i})$=0 **then**
    $Y(x_{\ell_i}) = Y(x^*),\ x^* = \arg\min_y\{D_t(x_{\ell_i},y) \mid p(y) \geq p(x_{\ell_i})$ and y is labeled$\}$.
  end if
end for

Noting $\lim_{\|x_o\|_2\to\infty} \boldsymbol{\pi}(x_o) = 0$, it follows that $\lim_{\|x_o\|_2\to\infty} \mathcal{D}_t(x_o) = 0$, so that as outliers move farther away from the rest of the data, they become less likely to be detected as modes. We remark that if outlier detection is performed on the data as a pre-processing step, this problem is less significant, since densities become more comparable across points. In this case, other constructions for $\mathcal{D}_t$ may be sufficiently robust, for example constructions that are additive in p, ρ_t.

The LUND algorithm combines density estimation (as captured by p) with diffusion geometry (as captured by ρ_t). The crucial parameter of LUND is the time parameter, which determines the diffusion distance D_t used. Theorem 3.3 may be used to show that there is a range of t for which applying the proposed LUND algorithm is provably accurate. The first concern is to understand conditions guaranteeing these modes are estimated accurately, the second that all other points are consequently labeled correctly. The following result summarizes Corollaries 5.4, 5.5, corresponding to the case when K is unknown a priori and must be estimated (as in Algorithm 1) or is known a priori (as in Algorithm 2).

Theorem 3.5 *Suppose $X = \bigcup_{k=1}^K X_k$ as above. Let $\mathcal{M}$ be the set of cluster density maxima:*

$$\mathcal{M} = \{p(x) \mid \exists k \in \{1,2,\dots,K\} \text{ such that } x = \arg\max_{y\in X_k} p(y)\}.$$

(a) Let $\{x_{m_i}\}_{i=1}^n$ be the points $\{x_i\}_{i=1}^n$, sorted so that $\mathcal{D}_t(x_{m_1}) \geq \mathcal{D}_t(x_{m_2}) \geq \cdots \geq \mathcal{D}_t(x_{m_n})$. Then Algorithm 1 correctly estimates K and labels all points correctly

Algorithm 2 LUND Algorithm, K Known

Input: X (data), σ_0 (kernel density bandwidth), σ (diffusion scaling parameter), t (time parameter), K (number of clusters)
Output: Y (cluster assignments)

Build Markov transition matrix $\mathbf{P}$ using scale parameter σ.
Compute a KDE $p(x)$ for all $x \in X$ using kernel bandwidth σ_0.
Compute $\rho_t(x)$ for all $x \in X$.
Compute $\mathcal{D}_t(x) = \rho_t(x)p(x)$ for all $x \in X$.
Sort X according to $\mathcal{D}_t(x)$ in descending order as $\{x_{m_i}\}_{i=1}^n, n = |X|$.
Assign $Y(x_{m_i}) = i,\ i = 1, \ldots, K$, and $Y(x_{m_i}) = 0,\ i = K+1, \ldots, n$.
Sort X according to $p(x)$ in decreasing order as $\{x_{\ell_i}\}_{i=1}^n$.
for $i = 1 : n$ **do**
  if $Y(x_{\ell_i})$=0 **then**
    $Y(x_{\ell_i}) = Y(x^*),\ x^* = \arg\min_y \{D_t(x_{\ell_i}, y) \mid p(y) \geq p(x_{\ell_i})$ and y is labeled$\}$.
  end if
end for

for any τ satisfying

$$\frac{\max(\mathcal{M})}{\min(\mathcal{M})} \frac{\max_{i=1,\cdots,K} \rho_t(x_{m_i})}{\min_{i=1,\cdots,K} \rho_t(x_{m_i})} < \tau < \frac{\min(\mathcal{M})}{\max(\mathcal{M})} \frac{D_t^{btw}}{D_t^{in}}. \tag{3.6}$$

(b) If K is known a priori, then Algorithm 2 labels all points accurately provided that

$$\frac{D_t^{in}}{D_t^{btw}} < \frac{\min(\mathcal{M})}{\max(\mathcal{M})}. \tag{3.7}$$

Theorem 3.3 suggests that the conditions (3.6), (3.7) will hold for a wide range of t, depending on $\max(\mathcal{M})/\min(\mathcal{M})$ and the underlying data geometry. Indeed, in the case that $C_4 = C_5 = 1/\sqrt{n}$, and the clusters are irreducible (C_1 small) and well-separated (C_3 large), the time interval guaranteeing $D_t^{\text{in}}/D_t^{\text{btw}} < \epsilon$ is $[C_1 \ln(\frac{C_2}{\epsilon}), C_3\epsilon]$, which is wide even for ϵ small. Indeed, setting $\epsilon = \min(\mathcal{M})/(2\max(\mathcal{M}))$ is sufficient to guarantee (3.7) holds for a wide range of t (always for C_1 sufficiently small and C_3 sufficiently large). Of course, the smaller the ratio $\min(\mathcal{M})/\max(\mathcal{M})$, the harder (3.7) is to satisfy.

Together with Theorem 3.5, this implies the proposed method (Algorithm 1) correctly labels the data and estimates the number of clusters K correctly. Note that (3.7) implicitly relates the density of the separate clusters to their geometric properties. Indeed, if the clusters are well-separated and cohesive enough, then $D_t^{\text{in}}/D_t^{\text{btw}}$ is very small, and a large discrepancy in the density of the clusters can be tolerated in inequality (3.7). Note that $D_t^{\text{in}}, D_t^{\text{btw}}, \min(\mathcal{M})$, and $\max(\mathcal{M})$ are invariant to increasing n, as long as the scale parameter in the kernel used for constructing diffusion distances and the KDE adjusts according to standard convergence results for graph

Laplacians (Belkin and Niyogi, 2005, 2007; Garcia Trillos et al., 2016, 2018). In this sense these quantities are properties of the mixture model (3.1), and neither of n nor the scale parameter of the diffusion process σ.

We note moreover that Theorem 3.5 suggests t must be taken in a mesoscopic range, that is, sufficiently far from 0 but also bounded. Indeed, for t small, D_t^{in} is not necessarily small, as the Markov process has not mixed locally yet. For t large, $\mathbf{P}^t$ is close to global stationarity, and the ratio $D_t^{\text{in}}/D_t^{\text{btw}}$ is not necessarily small, since D_t^{btw} will be small. In this case, clusters would only be detectable based on density, requiring thresholding, which is susceptible to spurious identification of regions around local density maxima as clusters.

We remark that a LUND prototype adapted to image data was proposed for the empirical study of high-dimensional hyperspectral images by Murphy and Maggioni (2018a,b, 2019b), where it is shown to enjoy competitive performance with state-of-the-art clustering algorithms on specific data sets. The LUND algorithm presented in the present work is more general and, in contrast with earlier related methods, appropriate for general point cloud data, not just images.

3.2 Comparisons with Related Clustering Algorithms

LUND combines graph-based methods with density-based methods, and it is therefore natural to compare it with spectral clustering and FSFDPC among other methods.

3.2.1 Comparison with Spectral Clustering

The normalized graph-cut problem in spectral clustering is related to the probability of transitioning between clusters in *one* time step (Meila and Shi, 2001). LUND uses intermediate time scales to separate clusters, namely the time scale at which the random walk has almost reached the stationary distribution conditioned on not leaving a cluster, and has not yet transitioned (with sizable probability) to a different cluster.

Spectral clustering enjoys performance guarantees under a range of model assumptions (Chen and Lerman, 2009a,b; Arias-Castro, 2011; Arias-Castro et al., 2011; Vidal, 2011; Zhang et al., 2012; Elhamifar and Vidal, 2013; Wang et al., 2015; Soltanolkotabi et al., 2014; Arias-Castro et al., 2017; Little et al., 2017). Under nonparametric assumptions on (3.1) with $K = 2$, Shi et al. (2009) show that the principal eigenfunctions and eigenvalues of the associated kernel operator $\mathcal{K}(f)(x) = \int K(x,y)f(y)d\mu(y)$ are closely approximated by the principal spectra of the kernel operators $\mathcal{K}_i(f)(x) = \int K(x,y)f(y)d\mu_i(y), i = 1, 2$, possibly mixed up, depending on the spectra of $\mathcal{K}_1, \mathcal{K}_2$ and the weights w_1, w_2. This allows for the number of classes to be estimated accurately in some situations, and for points to be labeled by determining which distribution certain eigenvectors come from.

The related work of Schiebinger et al. (2015) provides sufficient conditions under the nonparametric model (3.1) for the low-dimensional embedding of spectral clus-

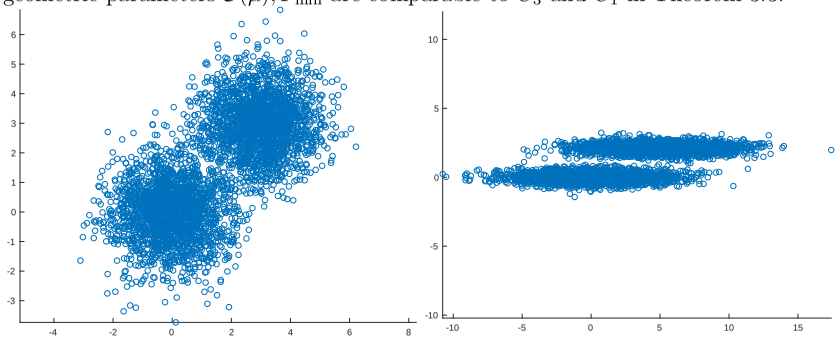
tering to map well-separated, coherent regions in input space to approximately orthogonal regions in the embedding space. This in turn implies that K-means clustering succeeds with high probability, thereby yielding guarantees on the accuracy of spectral clustering. These results depend on two quantities: with μ as in (3.1) and $\mathcal{K}$ a kernel, they define separation and cohesion quantities, respectively, as $\mathcal{S}(\mu) = \max_{i \neq j} \mathcal{S}(\mu_i, \mu_j), \Gamma_{\min}(\mu) = \min_{i=1,\ldots,K} \Gamma(\mu_i)$, where

$$\mathcal{S}(\mu_i, \mu_j) = \frac{1}{p(X)} \int_X \int_X \mathcal{K}(x,y) d\mu_i(x) d\mu_j(y), \Gamma(\mu_i) = \inf_{S \subset X} \frac{p(X)}{p(S)p(S^c)} \int_S \int_{S^c} \mathcal{K}(x,y) d\mu_i(x) d\mu_i(y),$$

$p(S) = \int_S \int_X \mathcal{K}(x,y) d\mu(x) d\mu(y)$. A major result of Schiebinger et al. (2015) is that spectral clustering is accurate with high probability depending on a confidence parameter β and the number of data samples n if

$$\frac{\sqrt{K(\mathcal{S}(\mu) + \mathcal{C}(\mu))}}{\min_{i=1,\ldots K} w_i} + \left(\frac{1}{\sqrt{n}} + \beta \right) \lesssim \Gamma^4_{\min}(\mu)\,, \tag{3.8}$$

where $\mathcal{C}(\mu)$ is a "coupling parameter" that is not germane to the present discussion. Condition (3.8) holds when the within-cluster coherence $\Gamma_{\min}(\mu)$ is large relative to the similarity between clusters $\mathcal{S}(\mu)$. Fixing the separation $\Gamma_{\min}(\mu)$, (3.8) is more likely to hold if the clusters are relatively spherical in shape. For example, in Figure 3 we represent two data sets, each consisting of two clusters, with comparable $\mathcal{S}(\mu)$, but substantially different $\Gamma_{\min}(\mu)$. Also note that in the finite sample case when $\frac{1}{\sqrt{n}}$ in (3.8) is non-negligible, the importance of $\Gamma_{\min}$ being not too small increases. The geometric parameters $\mathcal{S}(\mu), \Gamma_{\min}$ are comparable to C_3 and C_1 in Theorem 3.3.



(a) $\mathcal{S}(\mu) = 0.0533, \Gamma_{\min}(\mu) \approx 0.9550$. (b) $\mathcal{S}(\mu) = 0.0523,\ \Gamma_{\min}(\mu) \approx 0.2560$.

Figure 3: In (a) and (b) two different mixtures of Gaussians are shown. The two mixtures have roughly the same measure of between-cluster distance $\mathcal{S}(\mu)$, but significantly different within-cluster coherence $\Gamma_{\min}(\mu)$. Spectral clustering will enjoy much stronger performance guarantees, according to Schiebinger et al. (2015), for the data in (a) compared to the data in (b), for a range of relevant choices of the parameter σ.

It is of related interest to compare LUND to spectral clustering by recalling (2.2). In the generic case that $\lambda_2 > \lambda_3$, the $(\psi_2(x) - \psi_2(y))^2$ term dominates asymptotically as $t \to \infty$. Hence, as $t \to \infty$, LUND bears resemblance to spectral clustering with the second eigenvector alone (Shi and Malik, 2000). On the other extreme, for $t = 0$, diffusion distances depend on all eigenvectors equally. Using the first K or the 2nd through $(K+1)^{st}$ eigenvectors ψ_l is the basis for many spectral clustering algorithms (Ng et al., 2002; Schiebinger et al., 2015), and is comparable to LUND for $t = 0$, combined with a truncation of (2.2). Note that clustering with the kernel $\mathcal{K}$ alone relates to using all eigenvectors and $t = 1$. By allowing t to be a tunable parameter, LUND interpolates between the extremes of the K principal eigenvectors equally ($t = 0$ and truncating the eigendecomposition after the Kth or $(K+1)^{st}$ eigenvector), using the kernel matrix ($t = 1$), and using only the second eigenvector ($t \to \infty$). The results of Section 6 validate the importance of this flexibility.

An additional challenge when using spectral clustering is to robustly estimate K. The *eigengap* $\hat{K} = \arg\max_i \lambda_{i+1} - \lambda_i$ is a commonly used heuristic, but is often ineffective when Euclidean distances are used in the case of non-spherical clusters (Arias-Castro, 2011; Little et al., 2017). In contrast, Theorem 3.5 suggests LUND can robustly estimate K, which is shown empirically for synthetic data in Section 6.

It is also of interest to compare the guarantees of Theorem 3.3 to the analysis of spectral clustering of Nadler and Galun (2007); see Section 2.1.4. In particular, the guarantees of LUND require balancing two quantities: the within cluster mixing times and the between-cluster transition probabilities. These are analyzed precisely in Section 4, and are quantified in Theorem 3.3 by C_1 (within cluster mixing propensity) and C_3 (between cluster transition propensity). In the framework of continuous Fokker-Planck equations, these notions are closely related to relaxation time and first exit time, respectively. Nadler and Galun (2007) argue that as long as the first passage exit time is greater than the relaxation time within a cluster, for all clusters, then spectral clustering has a hope of achieving good results. The LUND algorithm relies on a similar fundamental observation, and the delicate balance between these two notions (within cluster mixing and between cluster transitions) are analyzed in a precise, quantitative sense for discrete Markov chains in Section 4, leading in Theorem 4.12 to a guarantee on the behavior of diffusion distances in terms of these quantities.

Computationally, LUND and spectral methods are essentially the same, with the bottleneck in complexity being either the spectral decomposition of a dense $n \times n$ matrix ($O(Mn^2)$ where M is the number of eigenvectors sought), or the computation of nearest neighbors when using a sparse diffusion operator or Laplacian (using an indexing structure for a fast nearest neighbors search, this is $O(C^d Dn \log(n))$, where d is the intrinsic dimension of the data).

3.2.2 Comparison With Local Graph Cutting Algorithms

The LUND algorithm bears some resemblance to local graph cutting algorithms (Spielman and Teng, 2004; Andersen et al., 2006, 2008; Andersen and Peres, 2009;

Spielman and Teng, 2013, 2014; Yin et al., 2017; Fountoulakis et al., 2017). These methods compute a cluster C around a given vertex v such that the conductance of C is high (see Definition 4.3), and which can be computed in sublinear time with respect to the total number of vertices in the graph n, and in linear time with respect to $|C|$. In order to avoid an algorithm that scales linearly (or worse) in n, global features—such as eigenvectors of a Markov transition matrix or graph Laplacian defined on the data—must be avoided. The Nibble algorithm (Spielman and Teng, 2013) and related methods (Andersen and Peres, 2009) compute approximate random walks for points nearby v, and truncate steps that take the random walker too far from already explored points. This accounts for the most important steps a random walker would take, and avoids considering all n vertices of the graph. In this sense, Nibble and related methods focus on local diffusion in order to compute a local cluster around a prioritized vertex v, while LUND focuses on both finding good starting points and a globally consistent partition of the whole graph, using nonlinear, typically large-time diffusion to uncover multitemporal structure.

3.2.3 Comparison with FSFDPC

The FSFDPC algorithm (Rodriguez and Laio, 2014) learns the modes of clusters in a manner similar to the method proposed in this article. In FSFDPC, the diffusion distance-based quantity ρ_t is replaced with a corresponding Euclidean distance-based quantity:

$$\rho^{\mathrm{Euc}}(x) = \begin{cases} \min_{y\in X}\{\|x-y\|_2 \mid p(y) \geq p(x)\}, & x \neq \arg\max_{y\in X} p(y), \\ \max_{y\in X} \|x-y\|_2, & x = \arg\max_{y\in X} p(y). \end{cases}$$

Moreover, the modes are estimated using only $\rho^{\mathrm{Euc}}(x)$, rather than $\mathcal{D}_t(x) = p(x)\rho_t(x)$ as proposed in the LUND algorithm. As in LUND, FSFDPC iteratively assigns points the same label as their nearest Euclidean neighbor (LUND uses diffusion nearest neighbor) of higher density.

The differences between LUND and FSFDPC are fundamental. Theoretical guarantees for the FDFDPC using Euclidean distances do not accommodate a rich class of distributions and the guarantees proved in this article fail when using $\rho^{\mathrm{Euc}}(x)$ (as in Rodriguez and Laio (2014)) or $\mathcal{D}^{\mathrm{euc}}(x) = p(x)\rho^{\mathrm{Euc}}(x)$ for computing modes. This is because for clusters that are multimodal or supported near non-spherical sets, there is no reason for high-density regions of one cluster to be well-separated in Euclidean distance. In Section 6, we shall see FSFDPC fails for the motivating data in Section 3. Moreover, the use of the product $\mathcal{D}_t(x) = p(x)\rho_t(x)$ to determine modes gives LUND robustness to outliers that FSFDPC lacks. Indeed, outlying points may admit very large $\rho_t(x)$ or $\rho^{\mathrm{Euc}}(x)$ values, but very small $p(x)$ values. In this sense, the density factor in $\mathcal{D}_t$ ensures that outlying points are not labeled as modes. This tension between geometry and density is highlighted in Theorem 3.5.

In addition, the LUND algorithm is able to correctly estimate the number of clusters in the data, even for nonlinear or elongated clusters, using the ratio (or decay) of $\mathcal{D}_t(x)$. A similar criterion for FSFDPC is not available for clusters that are nonlinear or elongated, due to the fact that high density regions connected by many paths in the data may be very far apart in Euclidean distance, leading heuristics based on the decay of $\rho^{\mathrm{Euc}}(x)$ or $\mathcal{D}^{\mathrm{euc}}(x)$ to fail; see Section 6. We remark that for simple, spherical data sets, using these heuristics for FSFDPC may work well; this is observed for isotropic Gaussian data in Section 6.3.

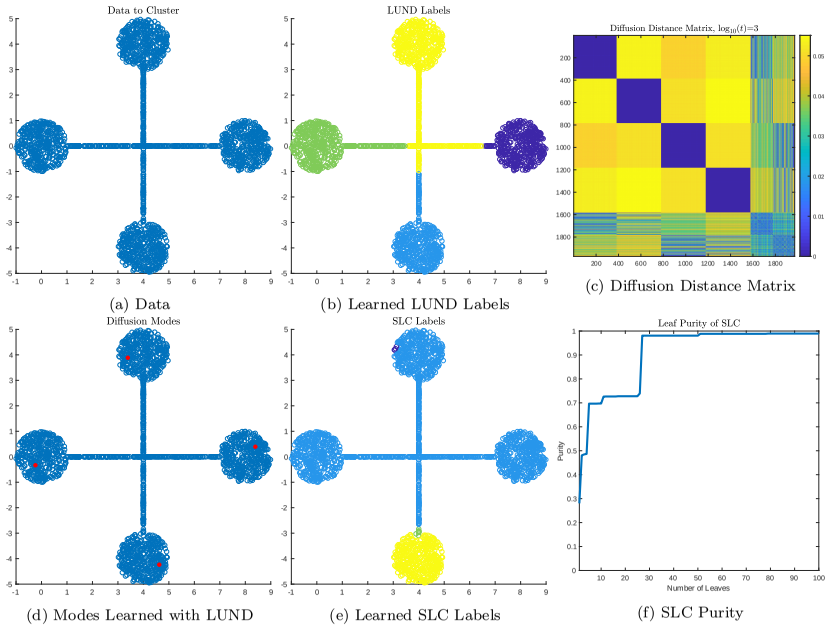
3.2.4 Comparison with Single Linkage Clustering

LUND is related to SLC in the sense that the underlying density of the data is an important determinant in the clusterings. However, LUND also incorporates geometric structure in the data when determining clusters, which can be especially powerful when density is uninformative. In Figure 4, data with constant density but significant geometric structure is analyzed. LUND succeeds in learning an accurate clustering with respect to the latent data geometry, while SLC fails. A theoretical analysis of the bottleneck phenomenon is presented in Section 4.3.

4. Analysis of Diffusion Processes on Data

In this section, we derive estimates for diffusion distances. Let $D_t^{\mathrm{in}}, D_t^{\mathrm{btw}}$ be as in (3.2). The main result of this section is to show there exists a time interval $\mathcal{T} \subset [0,\infty]$ so that $\forall t \in \mathcal{T},\ D_t^{\mathrm{btw}} > D_t^{\mathrm{in}}$, that is, for $t \in \mathcal{T}$, within cluster diffusion distance is smaller than between cluster diffusion distance. Showing that within-cluster distances are small and between-cluster distances are large is essential for any clustering problem. The benefit of using diffusion distance is its adaptability to the geometry of the data: it is possible that within cluster diffusion distance is less than between cluster diffusion distance, even in the case that the clusters are highly elongated and nonlinear. This property does not hold when points are compared with Euclidean distances or many other data-independent distances.

Compared to existing methods for analyzing diffusion distances, the proposed method does not require an analysis of the localization properties of eigenfunctions of $\mathbf{P}$, which may be challenging for small or elongated clusters (Shi et al., 2009). The proposed method has analogy to the analysis of Fokker-Planck equations of Nadler and Galun (2007), in that both approaches attempt to characterize the tension between within-cluster similarity and between-cluster separation for clustering with spectral methods. Unlike that analysis of the continuum stochastic PDE, our analysis provides explicit estimates on the behaviors of the discrete diffusion distances for finite data. Our analysis is based on the notions of *near-reducibility* and *stochastic complement*, which make precise the tension between within-cluster mixing and between-cluster transitions.



(a) Data (b) Learned LUND Labels (c) Diffusion Distance Matrix

(d) Modes Learned with LUND (e) Learned SLC Labels (f) SLC Purity

Figure 4: Geometric data with roughly constant empirical density is shown in (a). The spherical clusters are connected with thin bottlenecks. In (d), the modes learned by LUND are shown in red, indicating that one mode per cluster is learned. This is because although the density is roughly constant, the thin bottlenecks force the four circular clusters to be far apart in diffusion distance. The corresponding labels learned by LUND are in (b), showing that the cluster structure is accurately inferred. In (c), the matrix of pairwise diffusion distances is shown with diffusion time $t = 10^3$ and diffusion scaling parameter $\sigma = .5$. The four blue blocks on the diagonal indicate that within-cluster diffusion distances are small, while between cluster diffusion distances are large. The connecting bottlenecks correspond to the final rows and columns of this matrix, where the diffusion distances are less informative. In (e), labels learned from pruning the single linkage dendrogram at the third highest merge (producing four clusters) is shown, indicating this method is not appropriate for this data. Indeed, in (f), the purity of the hierarchical clustering is shown as a function of how many leaves (i.e. clusters) are used from the dendrogram. The purity remains low until ~ 25 clusters are used, indicating that the single linkage dendrogram is unable to efficiently separate the four spherical clusters until ~ 25 clusters are used. The fundamental reason for the failure of SLC on this data is the fact that the density is essentially constant and the data set is connected, which are the properties driving SLC. Unlike LUND, SLC does not incorporate geometric information, e.g. bottlenecks, which is quite discriminative for this data set.

4.1 Near Reducibility of Diffusion Processes

Let $\mathbf{P}$ be a Markov chain defined on points X satisfying the usual assumptions with unique stationary distribution $\boldsymbol{\pi}$. We will sometimes consider $\boldsymbol{\pi}$ as a function with domain X, other times as a vector with indices $\{1, \ldots, |X|\}$.

For any initial distribution $\boldsymbol{\pi}_0$, $\lim_{t\to\infty} \boldsymbol{\pi}_0 \mathbf{P}^t = \boldsymbol{\pi}$ and moreover for any choice of $\nu = \boldsymbol{\pi}_0/\boldsymbol{\pi}$, $D_t(x, y) \to 0$ uniformly as $t \to \infty$. One can quantify the rate of this convergence by estimating the convergence rate of $\mathbf{P}$ to its stationary distribution.

Definition 4.1 *For a discrete Markov chain with transition matrix* $\mathbf{P}$ *and stationary distribution* $\boldsymbol{\pi}$*, the* relative pointwise distance to $\boldsymbol{\pi}$ at time t *is* $\Delta(t) = \max_{i,j\in\{1,\ldots,n\}} |P^t_{ij} - \boldsymbol{\pi}_j|/\boldsymbol{\pi}_j$.

The decay of $\Delta(t)$ is regulated by the spectrum of $\mathbf{P}$ (Jerrum and Sinclair, 1989; Sinclair and Jerrum, 1989). Indeed, let $1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n > -1$ be the eigenvalues of $\mathbf{P}$; note that $\lambda_2 < 1$ follows from $\mathbf{P}$ irreducible and $\lambda_n > -1$ follows from $\mathbf{P}$ aperiodic (Chung, 1997). Let $\lambda_* = \max_{i=2,\ldots,n} |\lambda_i| = \max(|\lambda_2|, |\lambda_n|)$, $\boldsymbol{\pi}_{\min} = \min_{x\in X} \boldsymbol{\pi}(x)$.

Theorem 4.2 *(Jerrum and Sinclair, 1989; Sinclair and Jerrum, 1989) Let* $\mathbf{P}$ *be the transition matrix of a Markov chain on state space* X *satisfying the usual assumptions. Then* $\Delta(t) \leq \lambda_*^t/\boldsymbol{\pi}_{\min}$.

Instead of analyzing λ_*, the *conductance* of X may be used to bound $\Delta(t)$.

Definition 4.3 *Let* $\mathcal{G}$ *be a weighted graph on* X *and let* $S \subset X$*. The* conductance of S *is* $\Phi_X(S) = \sum_{x_i\in S, x_j\in S^c} \boldsymbol{\pi}_i P_{ij} / \min\left(\sum_{x_i\in S} \boldsymbol{\pi}_i, \sum_{x_i\in S^c} \boldsymbol{\pi}_i\right)$*. The* conductance of $\mathcal{G}$ *is* $\Phi(\mathbf{P}) = \min_{S\subseteq\mathcal{G}} \Phi_X(S)$.

Methods for estimating the conductance of certain graphs include *Poincaré estimates* (Diaconis and Stroock, 1991; Diaconis and Saloff-Coste, 1993) and the method of *canonical paths* (Jerrum and Sinclair, 1989; Sinclair and Jerrum, 1989; Aldous and Fill, 2002). These approaches estimate $\Phi(\mathbf{P})$ by showing that certain simple paths may be used as surrogates for generic paths in the graph. The conductance is related to λ_2; see Chung (1997):

Theorem 4.4 *(Cheeger's Inequality) Let* $\mathcal{G}$ *be a weighted, undirected graph with transition matrix* $\mathbf{P}$*. Then the second eigenvalue* λ_2 *of* $\mathbf{P}$ *satisfies* $\Phi(\mathbf{P})^2/2 \leq 1 - \lambda_2 \leq 2\Phi(\mathbf{P})$.

Combining Theorem 4.2 and Cheeger's inequality relates $\Delta(t)$ to $\Phi(\mathbf{P})$.

Theorem 4.5 *(Jerrum and Sinclair, 1989; Sinclair and Jerrum, 1989) Let* $\mathbf{P}$ *be the transition matrix for a Markov chain on* X *satisfying the usual assumptions. Suppose* $P_{ii} \geq \frac{1}{2}$, $\forall i = 1, \ldots, n$*. Then* $\Delta(t) \leq (1 - \frac{1}{2}\Phi(\mathbf{P})^2)^t/\boldsymbol{\pi}_{\min}$.

Note that any Markov chain can be made to satisfy $P_{ii} \geq \frac{1}{2}, \forall i = 1, \ldots, n$, simply by replacing $\mathbf{P}$ with $\frac{1}{2}(\mathbf{P} + \mathbf{I})$. This keeps the same stationary distribution and reduces the conductance by a factor of $\frac{1}{2}$.

Whether Theorem 4.2 or 4.5 is used, the convergence of $\mathbf{P}$ towards its stationary distribution is exponential, with rate determined by λ_* or $\Phi(\mathbf{P})$, that is, to how close to being reducible the chain is. This yields estimates on diffusion distances; indeed, for $x, y \in X$ and any initial distribution $\boldsymbol{\pi}_0$,

$$\begin{aligned} D_t(x,y) =& ||p_t(x,\cdot) - p_t(y,\cdot)||_{\ell^2(\nu)} \leq ||p_t(x,\cdot) - \boldsymbol{\pi}(\cdot)||_{\ell^2(\nu)} + ||p_t(y,\cdot) - \boldsymbol{\pi}(\cdot)||_{\ell^2(\nu)} \\ \leq& 2\sqrt{\sum_{u\in X} \max_{z\in X} \frac{|p_t(z,u) - \boldsymbol{\pi}(u)|^2}{\boldsymbol{\pi}(u)^2} \boldsymbol{\pi}(u)\boldsymbol{\pi}_0(u)} \\ \leq& 2\Delta(t)\sqrt{\sum_{u\in X} \boldsymbol{\pi}(u)\boldsymbol{\pi}_0(u)} \leq 2\Delta(t) \leq \frac{2(1-\frac{1}{2}\Phi(\mathbf{P})^2)^t}{\boldsymbol{\pi}_{\min}}. \end{aligned}$$

Thus, as $t \to \infty$, $D_t \to 0$ uniformly at an exponential rate depending on the conductance of the underlying graph; a similar result holds for λ_* in place of $\Phi(\mathbf{P})$. This gives a global estimate on the diffusion distance in terms of λ_* and $\Phi(\mathbf{P})$. Note that a similar conclusion holds by analyzing (2.2), recalling that ψ_1 is constant and $\lambda_2 = \max_{i\neq 1} |\lambda_i| = \lambda_*$.

Unfortunately, a global estimate on diffusion distances may be too coarse for unsupervised clustering. To obtain the desired separation of $D_t^{\text{in}}, D_t^{\text{btw}}$, we need to study not the global mixing time, but the *mesoscopic* mixing times, corresponding to the time it takes for convergence of points in each cluster towards their mesoscopic equilibria, before reaching the global equilibrium. For this purpose, we use results from the theory of *nearly reducible Markov processes* (Simon and Ando, 1961; Meyer, 1989). Suppose the matrix $\mathbf{P}$ is irreducible; write $\mathbf{P}$, possibly after a permutation of the indices of the points, in block decomposition as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \ldots & \mathbf{P}_{1m} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \ldots & \mathbf{P}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{m1} & \mathbf{P}_{m2} & \ldots & \mathbf{P}_{mm} \end{bmatrix}, \tag{4.6}$$

where each $\mathbf{P}_{ii}$ is square and $m \leq n$. Let I_i be the indices of the points corresponding to $\mathbf{P}_{ii}$. Recall that if the graph corresponding to $\mathbf{P}$ is disconnected, then $\mathbf{P}$ is a *reducible* Markov chain. Recall that $\|\mathbf{A}\|_\infty = \max_i \sum_j |A_{ij}|$ is the maximal row sum of $\mathbf{A} = (A_{ij})$. Suppose that $\|\mathbf{P}_{ij}\|_\infty$ is small but nonzero for $i \neq j$, that is, most of the interactions for points in I_i are contained within $\mathbf{P}_{ii}$. This suggests diffusion on the blocks $\mathbf{P}_{ii}$ have dynamics that converge to their own, mesoscopic equilibria before the entire chain converges to a global equilibrium, depending on the weakness of connection between blocks. Interpreting the support sets I_i as corresponding to the clusters of X, this suggests there will be a time range for which points within

each cluster are close in diffusion distance but far in diffusion distance from points in other clusters; such a state corresponds to a mesoscopic equilibrium. To make this precise, consider the notion of *stochastic complement*.

Definition 4.7 *Let $\mathbf{P}$ be an $n \times n$ irreducible Markov matrix partitioned into square block matrices as in (4.6). For a given index $i \in \{1, \ldots, m\}$, let $\mathbf{P}_i$ denote the principal block submatrix generated by deleting the ith row and i^{th} column of blocks from (4.6), and let $\mathbf{P}_{*i} = \left[\mathbf{P}_{1i}\mathbf{P}_{2i} \ldots \mathbf{P}_{i-1,i}\mathbf{P}_{i+1,i} \ldots \mathbf{P}_{mi}\right]^T$ and $\mathbf{P}_{i*} = \left[\mathbf{P}_{i1}\ \mathbf{P}_{i2}\ \ldots\ \mathbf{P}_{i,i-1}\ \mathbf{P}_{i,i+1}\ \ldots\ \mathbf{P}_{im}\right]$. The* stochastic complement *of $\mathbf{P}_{ii}$ is the matrix $\mathbf{S}_{ii} = \mathbf{P}_{ii} + \mathbf{P}_{i*}(\mathbf{I} - \mathbf{P}_i)^{-1}\mathbf{P}_{*i}$.*

One can interpret the stochastic complement $\mathbf{S}_{ii}$ as the transition matrix for a reduced Markov chain obtained from the original chain, but in which transitions into or out of I_i are masked. More precisely, in the reduced chain $\mathbf{S}_{ii}$, a transition is either direct in $\mathbf{P}_{ii}$ or indirect by moving first through points outside of $\mathbf{P}_{ii}$, then back into $\mathbf{P}_{ii}$ at some future time. Indeed, the term $\mathbf{P}_{i*}(\mathbf{I} - \mathbf{P}_i)^{-1}\mathbf{P}_{*i}$ in the definition of $\mathbf{S}_{ii}$ accounts for leaving I_i (the factor $\mathbf{P}_{i*}$), traveling for some time in I_i^c (the factor $(\mathbf{I} - \mathbf{P}_i)^{-1}$), then re-entering I_i (the factor $\mathbf{P}_{*i}$). Note that the factor $(\mathbf{I} - \mathbf{P}_i)^{-1}$ may be expanded in Neumann sum as $(\mathbf{I} - \mathbf{P}_i)^{-1} = \sum_{t=0}^{\infty} \mathbf{P}_i^t$, showing that it accounts for exiting from I_i and returning to it after an arbitrary number of steps outside of it.

The notion of stochastic complement quantifies the interplay between the mesoscopic and global equilibria of $\mathbf{P}$. We say $\mathbf{P}$ is *primitive* if it is non-negative, irreducible and aperiodic. The following theorem indicates how $\mathbf{P}$ may be analyzed when it is derived from cluster data $\{X_k\}_{k=1}^K$ sampled according to (3.1); a proof appears in the Appendix for completeness. This result, which produces estimates related to the diffusion operator $\mathbf{P}$ in the ℓ^1 norm, is used to prove results on diffusion distances, which are defined in an ℓ^2 sense, partially in order to take advantage of spectral decompositions for fast computations. This discrepancy will be discussed and controlled in Section 4.2.

Theorem 4.8 *(Meyer, 1989) Let $\mathbf{P}$ be an $n \times n$ irreducible row-stochastic matrix partitioned into K^2 square block matrices, and let $\mathbf{S}$ be the reducible row-stochastic matrix consisting of the stochastic complements of the diagonal blocks of $\mathbf{P}$:*

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \ldots & \mathbf{P}_{1K} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \ldots & \mathbf{P}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{K1} & \mathbf{P}_{K2} & \ldots & \mathbf{P}_{KK} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{22} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{S}_{KK} \end{bmatrix}.$$

Suppose each $\mathbf{S}_{ii}$ is primitive, so that the eigenvalues of $\mathbf{S}$ satisfy $\lambda_1 = \lambda_2 = \cdots = \lambda_K = 1 > \lambda_{K+1} \geq \lambda_{K+2} \geq \cdots > -1$. Let $\mathbf{Z}$ diagonalize $\mathbf{S}$, and let

$$\mathbf{S}^\infty = \lim_{t\to\infty} \mathbf{S}^t = \begin{bmatrix} \mathbf{1}\boldsymbol{\pi}^1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}\boldsymbol{\pi}^2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{1}\boldsymbol{\pi}^K \end{bmatrix},$$

where $\boldsymbol{\pi}^i$ is the stationary distribution for $\mathbf{S}_{ii}$. Then $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty \leq t\delta + \kappa|\lambda_{K+1}|^t$, where $\delta = 2\max_i \|\mathbf{P}_{i*}\|_\infty$ and $\kappa = \|\mathbf{Z}\|_\infty \|\mathbf{Z}^{-1}\|_\infty$. Moreover, for any initial distribution $\boldsymbol{\pi}_0$ and $\mathbf{s} = \lim_{t\to\infty} \boldsymbol{\pi}_0 \mathbf{S}^t = \boldsymbol{\pi}_0 \mathbf{S}^\infty$, $\|\boldsymbol{\pi}_0 \mathbf{P}^t - \mathbf{s}\|_1 \leq t\delta + \kappa|\lambda_{K+1}|^t$.

Note that this result does not require the Markov chain to be reversible, and hence applies to diffusion processes defined on *directed* graphs. The assumption that $\mathbf{S}$ is diagonalizable is not strictly necessary, and similar estimates hold more generally (Meyer, 1989).
The estimate $t\delta + \kappa|\lambda_{K+1}|^t$ consists of two terms. The $t\delta$ term corresponds to $\|\mathbf{P}^t - \mathbf{S}^t\|_\infty$, which accounts for the approximation of $\mathbf{P}^t$ by the reducible Markov chain $\mathbf{S}^t$. In the context of clustering, this term accounts for the between-cluster connections in $\mathbf{P}$. The term $\kappa|\lambda_{K+1}|^t$ corresponds to $\|\mathbf{S}^t - \mathbf{S}^\infty\|_\infty$, which accounts for propensity of mixing within a cluster. In the clustering context, this term quantifies the within-cluster distances.
It follows from Theorem 4.8 that, given ϵ sufficiently large, there is a range of t for which the dynamics of $\mathbf{P}^t$ are ϵ-close to the dynamics of the reducible, low-rank Markov chain $\mathbf{S}^\infty$.

Corollary 4.9 *Let $\lambda_{K+1}, \delta, \kappa$ be as in Theorem 4.8. Suppose that for some $\epsilon > 0$, $\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{|\lambda_{K+1}|}\right) < t < \frac{\epsilon}{2\delta}$. Then $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty < \epsilon$, and for every initial distribution $\boldsymbol{\pi}_0$, $\|\boldsymbol{\pi}_0 \mathbf{P}^t - \mathbf{s}\|_1 < \epsilon$.*

In contrast with t, the values $\lambda_{K+1}, \delta, \kappa$ may be understood as fixed geometric parameters of the data set which determine the range of times t at which mesoscopic equilibria are reached. More precisely, as $n \to \infty$, δ, κ converge to natural continuous quantities independent of n, and Garcia Trillos et al. (2018) proved that as $n \to \infty$, there is a natural scaling for $\sigma \to 0^+$ in which the (random) empirical eigenvalues of $\mathbf{P}$ converge in a precise sense to the (deterministic) eigenvalues of a corresponding continuous operator defined on the support of μ as in (3.1). Thus, the parameters of Theorem 4.9 may be understood as random fluctuations of geometrically intrinsic quantities depending on μ. In the context of the proposed data model, these quantities may be interpreted as follows:

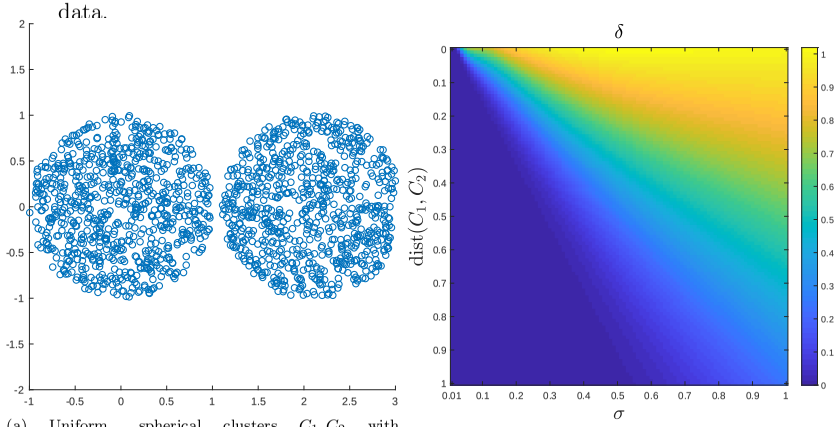
- λ_{K+1} is the largest eigenvalue of $\mathbf{S}$ not equal to 1. Since $\mathbf{S}$ is block diagonal and each $\mathbf{S}_{kk}$ is primitive, it follows that $\lambda_{K+1} = \max_{k=1,\dots,K} \lambda_2(\mathbf{S}_{kk})$. As discussed above, $\{\lambda_2(\mathbf{S}_{kk})\}_{k=1}^K$ is related to the conductance $\Phi(\mathbf{S}_{kk})$ and the mixing time of the random walk restricted to $\mathbf{S}_{kk}$. If the entries of $\mathbf{S}_{kk}$ are very close to the entries of $\mathbf{P}_{kk}$, then a perturbative argument yields $\lambda_2(\mathbf{S}_{kk}) \approx \lambda_2(\mathbf{P}_{kk})$.

- The quantity $\delta = 2\max_{k=1,\dots,K} \|\mathbf{P}_{k*}\|_\infty$ is controlled by the largest interaction between clusters. If the separation between the $\{X_k\}_{k=1}^K$ is large enough, δ will be small. To get a sense of this parameter, let C_1, C_2 be clusters with $n/2$ points uniformly sampled from the balls $B_1(0,0), B_1(2+\eta, 0) \subset \mathbb{R}^2$. Then, for

n sufficiently large, $\mathrm{dist}(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} \|x_1 - x_2\|_2 \approx \eta$. Note that the point in $B_1(0,0)$ nearest $B_1(2+\eta, 0)$ is $(1,0)$. Then modulo the variance from the random sampling,

$$\delta \approx \frac{2\int_{B_1(2+\eta,0)} \exp(-\|(x,y)-(1,0)\|_2^2/\sigma^2)dxdy}{\int_{B_1(2+\eta,0)} \exp(-\|(x,y)-(1,0)\|_2^2/\sigma^2)dxdy + \int_{B_1(0,0)} \exp(-\|(x,y)-(1,0)\|_2^2/\sigma^2)dxdy} \leq \frac{2\exp(-\eta^2/\sigma^2)}{\exp(-(\eta^2+4\eta)/\sigma^2)+1}. \tag{4.10}$$

Figure 5 illustrates empirically how δ depends on σ and $\mathrm{dist}(C_1, C_2)$ for such data.



(a) Uniform, spherical clusters C_1, C_2 with $\mathrm{dist}(C_1, C_2) \approx .1$

(b) δ as a function of $\mathrm{dist}(C_1, C_2)$ and σ.

Figure 5: In (a), data drawn uniformly at random from the union of two balls in $\mathbb{R}^2$ is shown. In (b), it is shown that for such equal sized, spherical, constant density clusters, the separation parameter exhibits rapid decay as $\sigma \to 0^+$, for fixed separation. The experiments confirm that the decay of δ is essentially logistic in $-(\eta^2/\sigma^2)$, as estimated in (4.10).

- The quantity $\kappa = \|\mathbf{Z}\|_\infty \|\mathbf{Z}^{-1}\|_\infty$, with $\mathbf{Z} = (\phi_1 | \dots | \phi_n)$, is a measure of the condition number of diagonalizing $\mathbf{S}$. If $\mathbf{Z}, \mathbf{Z}^{-1}$ are orthogonal matrices, then each row of $\mathbf{Z}, \mathbf{Z}^{-1}$ have ℓ^2 norm 1, hence $\kappa \leq n$. We remark that κ is bounded independently of n in the case that all the data live on a common manifold, using convergence of heat kernels and low-frequency eigenfunctions together with heat kernel estimates on manifolds. In the clustering setting, if each cluster is a manifold, similar results would hold in this case, albeit this analysis is a topic of ongoing research.

4.2 Diffusion Distance Estimates

Returning to the proposed data model $X = \bigcup_{k=1}^{K} X_k \sim \mu$ as per (3.1), let $\mathbf{P}$ be a corresponding Markov chain on X satisfying the usual assumptions. We estimate the dependence of diffusion distances on the parameters $\delta, \lambda_{K+1}, \kappa$ above. We also introduce a *balance quantity* that quantifies the difference between the ℓ^1 norm (the setting of Theorem 4.8) and the ℓ^2 norm (the setting of diffusion distances). Throughout this section, let $p_t(x_i, x_j) = \mathbf{P}^t_{ij},\ \mathbf{s}^\infty(x_i, x_j) = \mathbf{S}^\infty_{ij}$.

Definition 4.11 *Let $\mathbf{P}, \mathbf{S}^\infty \in \mathbb{R}^{n\times n}$ be as in Theorem 4.8. Define*

$$\gamma(t) = \max_{x\in X}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u) - \mathbf{s}^\infty(x,u)|}{\|p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)\|_2} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}.$$

Botelho-Andrade et al. (2019) show that for any vector $v \in \mathbb{R}^n$, $\|v\|_2 = \frac{c_v}{\sqrt{n}}\|v\|_1$, where

$$c_v = \left(1 - \frac{1}{2}\sum_{i=1}^{n}\left|\frac{|v_i|}{\|v\|_2} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}.$$

In this sense, $\gamma(t)$ measures how the ℓ^1 norm differs from the ℓ^2 norm across all rows of $\mathbf{P}^t - \mathbf{S}^\infty$. In particular, when each row of $\mathbf{P}^t - \mathbf{S}^\infty$ is close to uniform, $\gamma(t)$ is close to 1; when some row of $\mathbf{P}^t - \mathbf{S}^\infty$ concentrates all its mass around one index, then $\gamma(t) = \sqrt{n}$. Note that $1 \le \gamma(t) \le \sqrt{n}$ for all t.

Theorem 4.12 *Let $X = \bigcup_{k=1}^{K} X_k$ and let $\mathbf{P}$ be a corresponding Markov transition matrix on X. Let $\delta, \lambda_{K+1}, \kappa, \mathbf{S}^\infty$ be as in Theorem 4.8. Let D_t be the diffusion distance associated to $\mathbf{P}$ and counting measure ν. If t, ϵ satisfy*

$$\frac{\ln\left(\frac{2\kappa}{\epsilon}\right)}{\ln\left(\frac{1}{\lambda_{K+1}}\right)} < t < \frac{\epsilon}{2\delta},$$

then

(a) $D_t^{in} \le 2\frac{\epsilon}{\sqrt{n}}\gamma(t)$.

(b) $D_t^{btw} \ge 2\min_{y\in X}\|\mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)} - 2\frac{\epsilon}{\sqrt{n}}\gamma(t)$.

Proof By Corollary 4.9, $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty < \epsilon$, that is, $\max_{x\in X}\sum_{u\in X}|p_t(x,u) - \mathbf{s}^\infty(x,u)|\nu(u) < \epsilon$. To see (a), let k be arbitrary and let $x, y \in X_k$. Then:

$$\begin{aligned}
&||p_t(x,\cdot) - p_t(y,\cdot)||_{\ell^2(\nu)}\\
\le&||p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)||_{\ell^2(\nu)} + ||p_t(y,\cdot) - \mathbf{s}^\infty(y,\cdot)||_{\ell^2(\nu)} + ||\mathbf{s}^\infty(y,\cdot) - \mathbf{s}^\infty(x,\cdot)||_{\ell^2(\nu)}
\end{aligned}$$

$$\begin{aligned}
=&\frac{1}{\sqrt{n}}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u) - \mathbf{s}^\infty(x,u)|}{\|p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}||p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)||_{\ell^1(\nu)}\\
+&\frac{1}{\sqrt{n}}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(y,u) - \mathbf{s}^\infty(y,u)|}{\|p_t(y,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}||p_t(y,\cdot) - \mathbf{s}^\infty(y,\cdot)||_{\ell^1(\nu)}\\
+&||\mathbf{s}^\infty(y,\cdot) - \mathbf{s}^\infty(x,\cdot)||_{\ell^2(\nu)}\\
\le&\frac{2\epsilon}{\sqrt{n}}\max_{x\in X}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u) - \mathbf{s}^\infty(x,u)|}{\|p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1} + ||\mathbf{s}^\infty(y,\cdot) - \mathbf{s}^\infty(x,\cdot)||_{\ell^2(\nu)},
\end{aligned}$$

where t satisfies $\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{\lambda_{K+1}}\right) < t < \epsilon/(2\delta)$. The line relating the norm in $\ell^1(\nu)$ and $\ell^2(\nu)$ follows from Theorem 1 in Botelho-Andrade et al. (2019). Note that $\mathbf{S}^\infty$ has constant columns on each cluster, and in particular for $x, y \in X_k$, $\mathbf{s}^\infty(x,u) = \mathbf{s}^\infty(y,u) = \boldsymbol{\pi}^k(u)$ for all $u \in X$, so that $||\mathbf{s}^\infty(y,\cdot) - \mathbf{s}^\infty(x,\cdot)||_{\ell^2(\nu)} = 0$. Statement (a) follows.
To see (b), suppose that $x \in X_k, y \in X_\ell,\ k \ne \ell$. Then

$$\begin{aligned}
&\|p_t(x,\cdot) - p_t(y,\cdot)\|_{\ell^2(\nu)}\\
=&\|p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot) + \mathbf{s}^\infty(x,\cdot) - \mathbf{s}^\infty(y,\cdot) + \mathbf{s}^\infty(y,\cdot) - p_t(y,\cdot)\|_{\ell^2(\nu)}\\
\ge&\|\mathbf{s}^\infty(x,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)} - \|p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)\|_{\ell^2(\nu)} - \|p_t(y,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)}\\
=&\|\mathbf{s}^\infty(x,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)}\\
-&\frac{1}{\sqrt{n}}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u) - \mathbf{s}^\infty(x,u)|}{\|p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}||p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)||_{\ell^1(\nu)}\\
-&\frac{1}{\sqrt{n}}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(y,u) - \mathbf{s}^\infty(y,u)|}{\|p_t(y,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}||p_t(y,\cdot) - \mathbf{s}^\infty(y,\cdot)||_{\ell^1(\nu)}\\
\ge&\|\mathbf{s}^\infty(x,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)} - \frac{\epsilon}{\sqrt{n}}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u) - \mathbf{s}^\infty(x,u)|}{\|p_t(x,\cdot) - \mathbf{s}^\infty(x,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}\\
-&\frac{\epsilon}{\sqrt{n}}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(y,u) - \mathbf{s}^\infty(y,u)|}{\|p_t(y,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}\\
\ge&\|\mathbf{s}^\infty(x,\cdot) - \mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)} - 2\frac{\epsilon}{\sqrt{n}}\max_{z\in X_\ell\cup X_k}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(z,u) - \mathbf{s}^\infty(z,u)|}{\|p_t(z,\cdot) - \mathbf{s}^\infty(z,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}\\
\ge&2\min_{w\in X_k\cup X_\ell}\|\mathbf{s}^\infty(w,\cdot)\|_{\ell^2(\nu)} - 2\frac{\epsilon}{\sqrt{n}}\max_{z\in X_\ell\cup X_k}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(z,u) - \mathbf{s}^\infty(z,u)|}{\|p_t(z,\cdot) - \mathbf{s}^\infty(z,\cdot)\|_{\ell^2(\nu)}} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1},
\end{aligned}$$

where in the last step, to lower bound the first term we used that $\mathbf{s}^\infty(y,\cdot) = \boldsymbol{\pi}^l(\cdot)$, $\mathbf{s}^\infty(x,\cdot) = \boldsymbol{\pi}^k(\cdot)$, and recalled that since $k \ne l$ the supports of $\mathbf{s}^\infty(x,\cdot)$ and $\mathbf{s}^\infty(y,\cdot)$

are disjoint. Minimizing this lower bound over all clusters X_k, X_ℓ yields the desired result. ■

Heuristically, if ϵ is small and the reduced equilibrium distribution $\mathbf{s}^\infty$ is roughly constant on each cluster, there will be a range of t for which $D_t^{\text{in}} \ll D_t^{\text{btw}}$. The notion of $\mathbf{s}^\infty$ being roughly constant on each cluster is equivalent to nodes in the same cluster having roughly constant degree. These theoretical estimates are compared to empirical bounds computed numerically in Section 6.
If $\mathbf{P}$ is very close to $\mathbf{S}$ in Frobenius norm, then $p_t(x,y)$ is very close to $\mathbf{s}^\infty(x,y)$ and ϵ may be taken close to 0. In particular, for the ideal case $\epsilon = 0$, the estimates of Theorem 4.12 reduce to

$$D_t^{\text{in}} = 0 \qquad , \qquad D_t^{\text{btw}} \geq 2 \min_{y \in X} \|\mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)}. \tag{4.13}$$

One can define a natural notion of diffusion distance between disjoint clusters in a reducible Markov chain as the sum of the ℓ^2 norms of their respective stationary distribution, which agrees with both the definition of diffusion distances upon taking the limit $t \to +\infty$ and with the lower bound (b) in Theorem 4.12 when $\epsilon \to 0^+$. Hence, while the estimates in the proof of Theorem 4.12 may not be optimal, they are quite natural for $\epsilon \to 0^+$.
Away from the asymptotic regime $\epsilon \to 0^+$, the estimates of Theorem 4.12 may be further simplified by placing additional assumptions on the data. Indeed, if the equilibrium distributions in $\mathbf{S}^\infty$ are balanced and uniform, the following result holds:

Corollary 4.14 *Suppose that $\mathbf{s}^\infty$ is uniform on each X_k, and the cardinality of each X_k is n/K. Then for any t, ϵ satisfying $\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{\lambda_{K+1}}\right) < t < \frac{\epsilon}{2\delta}$,*

$$D_t^{in} \leq \frac{2}{\sqrt{n}}\epsilon\gamma(t) \qquad , \qquad D_t^{btw} \geq \frac{2}{\sqrt{n}}\left(\sqrt{K} - \epsilon\gamma(t)\right).$$

Proof If $\mathbf{S}^\infty$ has constant rows on each cluster (i.e. the stationary distribution on each cluster of the reduced Markov chain is uniform), and the clusters are of constant size n/K, then $2\min_{y\in X}\|\mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)} = 2\sqrt{K/n}$. Then Theorem 4.12 yields

$$\begin{aligned} D_t^{\text{btw}} \geq & 2\min_{y\in X}\|\mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)} - 2\frac{\epsilon}{\sqrt{n}}\gamma(t) \\ = & 2\sqrt{\frac{K}{n}} - 2\frac{\epsilon}{\sqrt{n}}\gamma(t) \\ = & \frac{2}{\sqrt{n}}\left(\sqrt{K} - \epsilon\gamma(t)\right). \end{aligned}$$

■

In particular, if $\epsilon \ll \sqrt{K}/(2\gamma(t))$, within cluster distances will be small since $D_t^{\text{in}} \ll \sqrt{K/n}$, and also there will be clear separation between clusters since $D_t^{\text{btw}} = \Omega(\sqrt{K/n})$. Note that when $\mathbf{P}^t - \mathbf{S}^\infty$ is balanced, $\gamma(t)$ is $O(1)$ with respect to n, so that the assumption on ϵ is independent of n.
We remark that in general if $\mathbf{S}^\infty$ and $\mathbf{P}^t - \mathbf{S}^\infty$ are balanced in the sense of having approximately uniform rows, then $\min_{y\in X}\|\mathbf{s}^\infty(y,\cdot)\|_{\ell^2(\nu)}$ scales like $1/\sqrt{n}$ and $\gamma(t) = O(1)$, respectively. In this (generic) case, the bounds of Theorem 4.12 are on the order $1/\sqrt{n}$. This is natural, since diffusion distances are computed in the ℓ^2 sense, while $\mathbf{P}^t$ and $\mathbf{S}^\infty$ have rows with ℓ^1 norm equal to 1.

4.3 Relaxing Separation Between Clusters

The analysis of diffusion distances in Section 4.2 depends on δ, a parameter characterizing separation between clusters. When δ is large, the estimates on D_t^{in} and D_t^{btw} from Theorem 4.12 are poor. However, δ is defined as the *maximum over all points* of the probability of transitioning to a new cluster in one time step. The formulation in terms of the maximum is convenient for analysis, but is pessimistic in the case of clusters cores that are separated by bottlenecks or low density noise regions. Indeed, for these data, δ may be large—there are some points quite close to the cluster boundaries—but the probability of transition between clusters cores is low.
The conservative estimates based on δ may be improved by considering some data points not as part of a cluster, but rather as noise or transition points. Let $X = \bigcup_{k=1}^K X_k \cup \mathcal{N}$ be a decomposition of X into cluster points ($\bigcup_{k=1}^K X_k$) and noise points ($\mathcal{N}$). Decompose $\mathbf{P}$ as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1K} & \tilde{\mathbf{P}}_{1,\mathcal{N}} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2K} & \tilde{\mathbf{P}}_{2,\mathcal{N}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}_{K1} & \mathbf{P}_{K2} & \dots & \mathbf{P}_{KK} & \tilde{\mathbf{P}}_{K,\mathcal{N}} \\ \tilde{\mathbf{P}}_{\mathcal{N},1} & \tilde{\mathbf{P}}_{\mathcal{N},2} & \dots & \tilde{\mathbf{P}}_{\mathcal{N},K} & \tilde{\mathbf{P}}_{\mathcal{N},\mathcal{N}} \end{bmatrix} \tag{4.15}$$

in a manner analogous to (4.6). Interpreting the final block rows and columns as noise or bottleneck points that are not part of clusters, Theorem 4.12 may be generalized to allow for clusters that are not well-separated, by accounting for the geometric properties of noise and bottlenecks which do not admit transitions between clusters over short time scales. This more generalized analysis does not make use of the reduced Markov chain $\mathbf{S}$, and some slightly new notation is required.

Definition 4.16 *For $\mathbf{P}$ as in (4.15) define*

$$\gamma_{min}(t) = \min_{x,y\in X, x\neq y}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u) - p_t(y,u)|}{\|p_t(x,\cdot) - p_t(y,\cdot)\|_2} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1},$$

$$\gamma_{max}(t) = \max_{x,y\in X}\left(1-\frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u)-p_t(y,u)|}{\|p_t(x,\cdot)-p_t(y,\cdot)\|_2}-\frac{1}{\sqrt{n}}\right|^2\right)^{-1}.$$

The following result is more general than Theorem 4.12, in that it allows for bottlenecks in the data, but is not interpretable in terms of the near reducibility of $\mathbf{P}$. Let $D_t^{\text{in}}, D_t^{\text{btw}}$ be as in (3.2); note that these quantities ignore $\mathcal{N}$ and only consider distances within and between the clusters $\{X_k\}_{k=1}^K$.

Theorem 4.17 *Let $X = \bigcup_{k=1}^K X_k \cup \mathcal{N}$ and let $\mathbf{P}$ be a corresponding Markov transition matrix on X, decomposed as in (4.15), and let ν be the counting measure. Let*

$$\begin{aligned}
\delta(t) =& 2\max_{1\le k\le K}\|(\mathbf{P}^t)_{k*}\|_\infty, \text{ where } \mathbf{P}_{k*} = [\mathbf{P}_{k1}\mathbf{P}_{k2}\dots\mathbf{P}_{k,k-1}\mathbf{P}_{k,k+1}\dots\mathbf{P}_{kK}],\\
\alpha(t) =& \max_{1\le k\le K}\max_{x,y\in X_k}\sum_{u\in X_k}|p_t(x,u)-p_t(y,u)|,\\
\beta(t) =& 2\min_{1\le k\le K}\min_{x\in X_k}\sum_{u\in X_k}|p_t(x,u)|,\\
\zeta(t) =& \max_{1\le k\le K}\max_{x,y\in X_k}\sum_{u\in \mathcal{N}}|p_t(x,u)-p_t(y,u)|.
\end{aligned}$$

Then:

(a) $D_t^{in} \le \gamma_{max}(t)(\alpha(t)+\delta(t)+\zeta(t))$.

(b) $D_t^{btw} \ge \gamma_{min}(t)(\beta(t)-\delta(t))$.

Proof To see (a), suppose $x, y \in X_k$. Then

$$\begin{aligned}
&\|p_t(x,\cdot)-p_t(y,\cdot)\|_{\ell^2(\nu)}\\
\le&\gamma_{\max}(t)\|p_t(x,\cdot)-p_t(y,\cdot)\|_{\ell^1(\nu)}\\
=&\gamma_{\max}(t)\left(\sum_{u\in X_k}|p_t(x,u)-p_t(y,u)|+\sum_{\ell\ne k}\sum_{u\in X_\ell}|p_t(x,u)-p_t(y,u)|+\sum_{u\in\mathcal{N}}|p_t(x,u)-p_t(y,u)|\right)\\
\le&\gamma_{\max}(t)(\alpha(t)+\delta(t)+\zeta(t)).
\end{aligned}$$

On the other hand, suppose $x \in X_k, y \in X_\ell, k \ne \ell$. Then

$$\begin{aligned}
\|p_t(x,\cdot)-p_t(y,\cdot)\|_{\ell^2(\nu)} \ge&\gamma_{\min}(t)\|p_t(x,\cdot)-p_t(y,\cdot)\|_{\ell^1(\nu)}\\
\ge&\gamma_{\min}(t)\left(\sum_{u\in X_k}|p_t(x,u)-p_t(y,u)|+\sum_{u\in X_\ell}|p_t(x,u)-p_t(y,u)|\right)\\
\ge&\gamma_{\min}(t)\left(\sum_{u\in X_k}|p_t(x,u)|-|p_t(y,u)|+\sum_{u\in X_\ell}|p_t(y,u)|-|p_t(x,u)|\right)
\end{aligned}$$

$$\ge\gamma_{\min}(t)(\beta(t)-\delta(t)).$$

■

The quantities $\delta(t), \alpha(t), \beta(t)$ in Theorem 4.17 are analogous to those in Theorem 4.12, albeit without the aid of the analysis in terms of $\mathbf{S}^\infty$. The quantity $\zeta(t)$ pertains to the noise $\mathcal{N}$, and hence has no analogy in Theorem 4.12.

- The quantity $\delta(t)$ is an upper bound on the probability mass that transitions between clusters at time t. It does not consider the mass transfer from clusters to the noise region. In Theorem 4.8, when the clusters are well-separated and $\mathcal{N} = \varnothing$, $\delta(t) \le \delta(1)t = \delta t$ is a worst-case, though intuitive, upper bound. Generally, $\delta(t)$ may depend on time nonlinearly and in particular it may stay small for a long time, before the global equilibrium time of the underlying Markov chain $\mathbf{P}$ is approached, at which point it grows essentially linearly in t before stabilizing around the global mixing time. This occurs, for example, in data with bottlenecks between clusters. We remark that considering $\delta(t)$ in these more general terms is possible even in the noiseless case of Theorem 4.8. Indeed, this may allow for a better estimate than the worst-case estimate of $\delta(t) \le \delta(1)t = \delta t$, though at some loss of simplicity.

- The quantity $\alpha(t)$ controls how close the diagonal blocks are to having constant rows, that is, how close to mesoscopic equilibria the random walker is on each individual cluster. In the case that all individual clusters are near their mesoscopic equilibria by time t, $\alpha(t)$ is close to 0. In the simpler context of Theorem 4.8, $\alpha(t)$ is governed by λ_{K+1}^t, that is, by the second largest eigenvalue of the blocks of the stochastic complement of $\mathbf{P}$. This is because the second smallest eigenvalue on each block controls the rate of convergence to local equilibrium of that block in the stochastic complement $\mathbf{S}$.

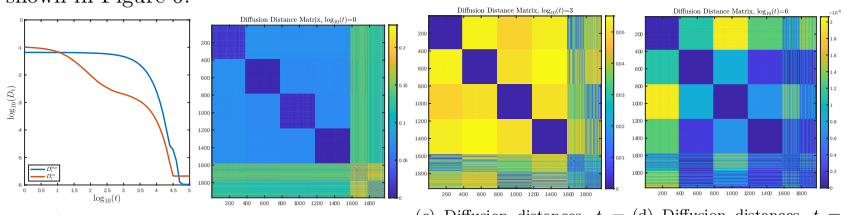
- The quantity $\beta(t)$ measures the row sums of the transition matrices localized to each cluster. It is analogous to the term involving the norm of the stationary distribution in the lower bound on D_t^{btw} in Theorem 4.12.

- The quantity $\zeta(t)$ is a measure of the impact of the noise region in the computation of diffusion distances. When the number of noise points is small relative to the number of cluster points, this term is negligible. It is also negligible when the noise region impacts a cluster in a uniform way, in particular when the cluster has reached a mesoscopic equilibrium.

Theorem 4.17 suggests that for data in which, with high probability, the clusters reach mesoscopic equilibria before the random walker exits the clusters and crosses the noise, there exists a range of parameters t such that D_t^{in} is small and D_t^{btw} is large. At this time scale, $\alpha(t)$ is small since the individual clusters are near mesoscopic equilibria,

and $\delta(t)$ is small and $\beta(t)$ large because most of the mass is still localized on the distinct clusters.

An important class of examples exhibiting this behavior is clusters connected by narrow bottlenecks, as shown for example in Figure 4 (a). In that example, consider the four balls as clusters $\{X_k\}_{k=1}^4$ and the bottlenecks as $\mathcal{N}$. The behavior of D_t^{in} and D_t^{btw} as a function of t, as well as several transition matrices for different t values are shown in Figure 6.



(a) $D_t^{\text{btw}}, D_t^{\text{in}}$ versus t. (b) Diffusion distances, $t = 1$ (c) Diffusion distances, $t = 10^3$ (d) Diffusion distances, $t = 10^6$

Figure 6: Plots illustrating that the bottleneck data in Figure 4 exhibits mesoscopic equilibria, despite the lack of any strict separation between the four clusters. Indeed, (a) shows that for a large time range, D_t^{in} is much less (note that the vertical axis is in $\log_{10}$ scale) than D_t^{btw}. This is further illustrated by the fact that for different ranges of time t, the blocks in the diffusion distance matrix corresponding to the cluster sets become essentially all 0—see (b), (c). For large time, the diffusion distances start to converge uniformly to 0, as shown in (d). Note that the last rows and columns of $\mathbf{P}$ correspond to the bottleneck points.

Theorem 4.12 does not provide useful estimates for this bottleneck data, because of the lack of strict separation between the clusters. Indeed, if bottleneck points are assigned to their nearest clusters, then δ is nearly 1, rendering Theorem 4.12 essentially useless. However, Theorem 4.17 supplies useful estimates on the diffusion distances within and between clusters (e.g. that $D_t^{\text{in}} \ll D_t^{\text{btw}}$). This is because $\delta(t)$ is small and $\beta(t)$ large for long time scales, until the random walk has explored from one cluster to another. That it takes a long time for the random walk to reach global stationarity due to the bottleneck can be argued in terms of graph conductances (Chung, 1997). We sketch a combinatorial argument below, in order to observe how the data dimensionality affects the bottlenecks. For simplicity, we consider just two clusters.

Consider the following model of bottleneck data in $\mathbb{R}^D$. Let $L > 0$ and $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{N}$, where $\mathcal{X}_1 = [0,1]^D, \mathcal{X}_2 = [0,1]^D + (L, 0, 0, \ldots, 0)$ are clusters connected at the centers of faces by a cylindrical tube $\mathcal{N}$ of length L and $D-1$ dimensional radius ϵ. Let X be generated by sampling uniformly at random from $\mathcal{X}$. To simplify the analysis, consider a random walk on X such that $x \in X$ transitions to points in $B_\sigma(x)$ with equal probability, and not to any other points, for some $\sigma > 0$.

For $\epsilon < \sigma$, a point x_* at the boundary interface between $\mathcal{X}_1$ and $\mathcal{N}$ is such that most of its near neighbors lie inside the cluster, rather than in the bottleneck. Indeed, up to geometric constants, $\text{vol}_D(X_1 \cap B_\sigma(x_*)) \asymp \sigma^D$, while $\text{vol}_D(\mathcal{N} \cap B_\sigma(x_*)) \asymp$

$\sigma\epsilon^{D-1}$. In particular, the probability of a random walker transitioning from x_* into the bottleneck in one time step is upper bounded by $(\epsilon/\sigma)^{D-1}$, so that the expected number of times a random walk reaches the boundary interface before exiting X_1 is lower bounded by $(\sigma/\epsilon)^{D-1}$. Supposing that the random walker spends time T exploring X_1 before returning to the boundary interface, the expected time it takes for a random walker to exit from X_1 into the bottleneck is $(\sigma/\epsilon)^{D-1}T$. Moreover, once the random walker transitions from X_1 to $\mathcal{N}$, the expected number of steps it takes to cross the bottleneck to get to X_2 may be lower bounded by $\lfloor L/\sigma \rfloor$. Indeed, if all the one-step transitions are optimal in the sense of being distance σ exactly in the direction parallel to $(1, 0, \ldots, 0)$, then at least $\lfloor L/\sigma \rfloor$ steps are required to cross; better estimates on the expected number of steps to cross are also possible (Levin et al., 2009).

In particular, the random walker will, with high probability, exit X_1 then return to X_1 $\Omega(\lfloor L/\sigma \rfloor)$ times, before exiting X_1 and making it all the way to X_2. Thus, the expected time to transition from an arbitrary starting point in X_1 to any point in X_2 may be crudely estimated to be $\Omega(\lfloor L/\sigma \rfloor (\sigma/\epsilon)^{D-1}T)$, which is large when L is large and ϵ is small, i.e. the bottleneck is long and thin. In particular, D_t^{btw} will remain large for $t \lesssim \lfloor L/\sigma \rfloor (\sigma/\epsilon)^{D-1}T$. Noting that T is comparable to the mixing time on X_1 (and thus X_2) alone, this suggests that there are time scales at which the random walker is at equilibrium on the distinct clusters (albeit with small leakage into the bottleneck at each time step) but at which global equilibrium is not reached. At such time scales, diffusion distances within a cluster are small, and between the two clusters are large.

We remark that this property of diffusion distances for bottlenecks also allows LUND to correctly learn the clusters in the data, since the density is constant on this data set. Indeed, in Figure 4, LUND learns four modes in the four distinct clusters, and consequently labels all points correctly. It is interesting to note that depending on the random sample, LUND may learn three modes in distinct clusters and a fourth mode either in the fourth cluster (as in Figure 4 (a)) or at the intersection of the bottlenecks (a possibility not shown in Figure 4). Whether the fourth mode is in a cluster or in the middle of the bottleneck depends on random fluctuations in the empirical density of the data. Regardless of if fourth mode is in the bottleneck region or in a cluster, the consequent LUND labeling of the data is correct. Indeed, if one of the modes is the bottleneck, then the cluster without a mode will have points assigned the label associated to the bottleneck mode, leading all the clusters to be given correct labels.

4.4 Example: A Simple Gaussian Mixture Model

The major parameters controlling the estimates in Theorem 4.12 are δ, λ_{K+1}, and κ. To illustrate the key quantities of this theorem, we consider the simple example of a mixture of Gaussians $\mu_G = \frac{1}{2}\mathcal{N}(\mathbf{x}_1, \Sigma) + \frac{1}{2}\mathcal{N}(\mathbf{x}_2, \Sigma)$ in $\mathbb{R}^2$ with diagonal isotropic covariance matrix $\Sigma = \frac{1}{10}\mathbf{I}$. We construct the diffusion transition matrix $\mathbf{P}$ using the Gaussian kernel with $\sigma = .2$, as described in Section 2.2.

As $\|\mathbf{x}_1 - \mathbf{x}_2\|_2$ increases, Theorem 4.12 becomes more informative. In Figure 7, samples are drawn from μ_G with different amounts of separation (and hence different $\delta, \lambda_{K+1}, \kappa$ values) and we show the dependence on ϵ of the bounds

$$\overline{D_t^{\text{in}}} = 2\frac{\epsilon}{\sqrt{n}}\gamma(t), \underline{D_t^{\text{btw}}} = 2\min_{y\in X}\|\mathbf{s}^{\infty}(y,\cdot)\|_{\ell^2(\nu)} - 2\frac{\epsilon}{\sqrt{n}}\gamma(t).$$

and the permissible time interval $[\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{\lambda_{K+1}}\right), \frac{\epsilon}{2\delta}]$. For Theorem 4.12 to be meaningful, ϵ must be such that simultaneously $\overline{D_t^{\text{in}}} < \underline{D_t^{\text{btw}}}$ and $\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{\lambda_{K+1}}\right) < \frac{\epsilon}{2\delta}$. As $\epsilon \to 0^+$, $\overline{D_t^{\text{in}}} < \underline{D_t^{\text{btw}}}$ holds if the clusters are internally well-connected and separated, as articulated in (4.13), while $\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{\lambda_{K+1}}\right) \to \infty$ and $\frac{\epsilon}{2\delta} \to 0^+$; a similar but reversed dichotomy occurs as $\epsilon \to \infty$. Figure 7 illustrates this tension between the hypotheses of Theorem 4.12 and the strength of its conclusion.

4.5 Robustness to Geometric Deformations

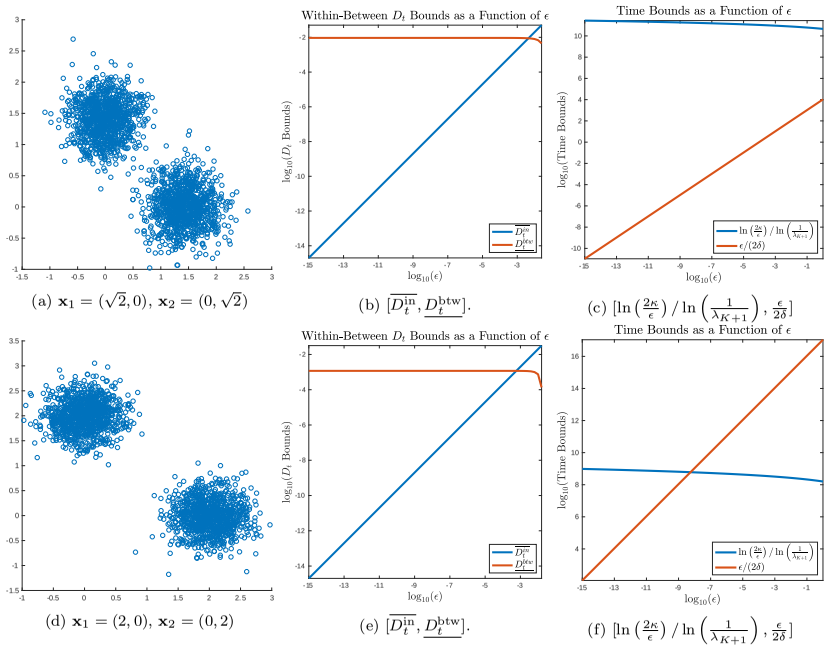
LUND depends mainly on the intrinsic geometric constants δ and λ_{K+1}, so clustering performance with LUND is robust to small amounts of geometric deformation, for example the action of a bi-Lipschitz map. This is illustrated in Figure 8, in which two circular clusters in $\mathbb{R}^2$ are distorted by stretching the second coordinate. The results of LUND as a function of t and the amount of stretching are shown. It can be seen that the more the clusters are stretched, the larger t needs to be to achieve good accuracy. This is because as the clusters are stretched, the parameter λ_{K+1} increases towards 1, meaning that t must be larger to ensure that mesoscopic equilibria on the distinct clusters are reached. Note that the relationship between domain geometry and the eigenvalues of the Laplacian is well-studied in classical settings (Szegö, 1954; Weinberger, 1956), and its role in graph-based clustering is noted in contexts other than LUND (Ng et al., 2002; Arias-Castro, 2011; Schiebinger et al., 2015; Little et al., 2017).

We note that transformations that push the clusters closer together can strongly impact LUND, since this will dramatically change the δ value. This is true of all clustering algorithms; the primary benefit of LUND compared to a range of existing methods is its robustness to distorting the shape of the clusters.

4.6 Relationship Between Time and Scaling Parameter in Diffusion Distances

The Markov chain underlying diffusion distances is typically constructed using the heat kernel $\mathcal{K}(x,y) = \exp(-\|x-y\|_2^2/\sigma^2)$, for some choice of (spatial) scale parameter σ. Once $\mathbf{P} = \mathbf{P}_\sigma$ is constructed, the time parameter t enters. For data sampled from a common manifold, there exists an asymptotic relationship between t and σ as $n \to \infty$:

$$\lim_{\sigma\to 0}\mathbf{P}_\sigma^{T_0/\sigma} = e^{-T_0\Delta}, \tag{4.18}$$

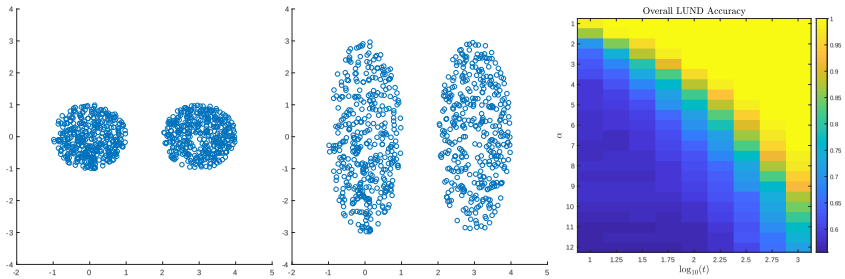


(a) $\mathbf{x}_1 = (\sqrt{2}, 0)$, $\mathbf{x}_2 = (0, \sqrt{2})$ (b) $[\overline{D_t^{\text{in}}}, \underline{D_t^{\text{btw}}}]$. (c) $[\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{\lambda_{K+1}}\right), \frac{\epsilon}{2\delta}]$

(d) $\mathbf{x}_1 = (2, 0)$, $\mathbf{x}_2 = (0, 2)$ (e) $[\overline{D_t^{\text{in}}}, \underline{D_t^{\text{btw}}}]$. (f) $[\ln\left(\frac{2\kappa}{\epsilon}\right)/\ln\left(\frac{1}{\lambda_{K+1}}\right), \frac{\epsilon}{2\delta}]$

Figure 7: 2000 data points sampled from μ_G, for two sets of means are shown in (a), (d). In (b), (e), the range between $\overline{D_t^{\text{in}}}$ and $\underline{D_t^{\text{btw}}}$—as a function of ϵ—is shown. Plots (c), (f) show the t interval guaranteed by Theorem 4.12, indicating the existence of a non-empty range of t for which the conclusions of Theorem 4.12 apply whenever the red curve is above the blue curve. As the means $\mathbf{x}_1, \mathbf{x}_2$ move apart, the time interval in which Theorem 4.12 guarantees good separation between the clusters expands. This is makes sense intuitively, since as the clusters move apart, the unsupervised learning problem becomes easier. We remark that the separation for the data in (a) is insufficient for Theorem 4.12 to guarantee a large time range in which $D_t^{\text{in}} < D_t^{\text{btw}}$. This is because the δ constant is large for this data set, since δ is determined by the points in distinct clusters that are nearest, and in particular is a worst-case bound. Relaxing the δ separation condition is discussed in Section 4.3.

where $e^{-T_0\Delta}$ is the infinitesimal generator corresponding to continuous diffusion with canonical time T_0 (Lafon et al., 2006). So, asymptotically as $n \to \infty$, and requiring $\sigma \to 0^+$ and $t = \frac{T_0}{\sigma}$, using (t, σ) is equivalent to using $(Ct, \sigma/C)$ for any constant $C > 0$. This suggests that asymptotically as $n \to \infty$, the performance of LUND with respect to σ, t should be constant if $t\sigma$ is constant as $\sigma \to 0^+$. As we shall see observe in Section 6, working with *finite* data in the *cluster* setting, rather than the asymptotic regime on a common manifold, may lead to more subtle relationships between t and σ.



(a) Circular Data (b) Geometrically Deformed Data (c) LUND Accuracy

Figure 8: In (a), uniform spherical data is shown. In (b), the data in (a) has been geometrically deformed by dilating the second coordinate by a factor of 3. We consider the robustness of LUND to the natural geometric transformation $(x, y) \mapsto (x, \alpha y)$, where α is understood as a stretching parameter. In (c), the results of LUND are shown as a function of t and the dilation factor α. As α increases, the cluster stretches and taking larger t allows LUND to accurately label the data. This is because the random walker requires more time to explore a high elongated cluster than a compact, spherical one.

5. Performance Guarantees for Unsupervised Learning

We consider now how the LUND algorithm (Algorithm 1) performs on data $X = \bigcup_{k=1}^{K} X_k$. Let $p(x)$ be a KDE for $x \in X$, let ρ_t be as in (3.4), and recall $\mathcal{D}_t(x) = p(x)\rho_t(x)$. The LUND algorithm sets the maximizers of $\mathcal{D}_t$ to be the modes of the clusters. Requiring potential modes to have large ρ_t values enforces that modes should be far in diffusion distance from other high-density points, and incorporating $p(x)$ downweighs outliers, which may be far in diffusion distance from their nearest neighbor of higher empirical density.

Theorem 5.1 *Suppose $X = \bigcup_{k=1}^{K} X_k$. If $D_t^{in}/D_t^{btw} < \min(\mathcal{M})/\max(\mathcal{M})$, then the K maximizers $\{\boldsymbol{x}_i^*\}_{i=1}^{K}$ of $\mathcal{D}_t(x)$ are such that $\boldsymbol{x}_i^*$ is a highest empirical density point of X_{k_i} for some permutation $(k_1, \ldots, k_K)$ of $(1, \ldots, K)$.*

Proof We proceed by induction on $1 \le m \le K$. Clearly $\mathbf{x}_1^* = \arg\max_{y \in X} p(y)$ is a highest empirical density point of some X_k. Then suppose $\mathbf{x}_1^*, \ldots, \mathbf{x}_m^*, m < K$, have been determined, and are highest empirical density points of distinct classes $X_{k_1}, \ldots, X_{k_m}$. We show that $\mathbf{x}_{m+1}^*$ must be a highest density point among the remaining X_k, $k \notin \{k_1, \ldots, k_m\}$.
First, suppose $x \in X_{k_r}$ for some $r \in \{1, \ldots, m\}$ is any point in the classes already discovered, not of maximal within-class density. Then $\rho_t(x) \le D_t^{\text{in}}$, since $\mathbf{x}_r^* \in X_{k_r}$ has $p(x) < p(\mathbf{x}_r^*)$, and hence for any $\mathbf{x}$ a highest density point in a cluster not already discovered,

$$\mathcal{D}_t(x) < p(\mathbf{x}_{k_r}^*)\rho_t(x) \le \max(\mathcal{M}) D_t^{\text{in}} \le \frac{\max(\mathcal{M})}{\min(\mathcal{M})} \frac{D_t^{\text{in}}}{D_t^{\text{btw}}} p(\mathbf{x})\rho_t(\mathbf{x}) < p(\mathbf{x})\rho_t(\mathbf{x}) = \mathcal{D}_t(\mathbf{x}).$$

Thus, $\mathbf{x}_{m+1}^* \neq x$.
Now, suppose $x \in X_k, k \neq k_1, \ldots k_m$. If $x \neq \mathbf{x}$ an empirical density maximizer of X_k, then:

$$\mathcal{D}_t(x) = p(x)\rho_t(x) < p(\mathbf{x})\rho_t(x) \le p(\mathbf{x}) D_t^{\text{in}} < p(\mathbf{x}) D_t^{\text{btw}} \le p(\mathbf{x})\rho_t(\mathbf{x}) = \mathcal{D}_t(\mathbf{x}).$$

Hence, $\mathbf{x}_{m+1}^* \neq x$, and thus $\mathbf{x}_{m+1}^*$ must be among the classwise empirical density maximizers of X_k, $k \notin \{k_1, \ldots, k_m\}$. ■

A similar method proves that the ratios of the sorted $\mathcal{D}_t$ values determine the number of clusters K. The problem of estimating the number of clusters is a crucial one, but few methods admit theoretical guarantees; see Little and Byrd (2015) for an overview.

Corollary 5.2 *Let $\{x_{m_i}\}_{i=1}^{n}$ be the points $\{x_i\}_{i=1}^{n}$, sorted in non-increasing order: $\mathcal{D}_t(x_{m_1}) \ge \mathcal{D}_t(x_{m_2}) \ge \cdots \ge \mathcal{D}_t(x_{m_n})$. Then:*

(a) $\frac{\mathcal{D}_t(x_{m_j})}{\mathcal{D}_t(x_{m_{j+1}})} \le \frac{\max(\mathcal{M})}{\min(\mathcal{M})} \frac{\max_{i=1,\cdots,K} \rho_t(x_{m_i})}{\min_{i=1,\cdots,K} \rho_t(x_{m_i})}$ *for $j < K$.*

(b) $\frac{\mathcal{D}_t(x_{m_K})}{\mathcal{D}_t(x_{m_{K+1}})} \ge \frac{\min(\mathcal{M})}{\max(\mathcal{M})} \frac{D_t^{btw}}{D_t^{in}}$.

Proof Statement (a) is immediate from the definition. To see (b), we compute

$$\mathcal{D}_t(x_{m_K}) \ge \min(\mathcal{M}) D_t^{\text{btw}} = \frac{\min(\mathcal{M})}{\max(\mathcal{M})} \frac{D_t^{\text{btw}}}{D_t^{\text{in}}} D_t^{\text{in}} \max(\mathcal{M}) \ge \frac{\min(\mathcal{M})}{\max(\mathcal{M})} \frac{D_t^{\text{btw}}}{D_t^{\text{in}}} \mathcal{D}_t(x_{m_{K+1}}).$$

■

Hence if $\frac{D_t^{\text{in}}}{D_t^{\text{btw}}} \frac{\max_{i=1,\cdots,K} \rho_t(x_{m_i})}{\min_{i=1,\cdots,K} \rho_t(x_{m_i})} \ll \left(\frac{\min(\mathcal{M})}{\max(\mathcal{M})}\right)^2$, there will be a sharp drop-off in the values of $\mathcal{D}_t$ after the first K maximizers. Note that the ratio $\min_{i=1,\cdots K} \rho_t(x_{m_i}) / \max_{i=1,\cdots K} \rho_t(x_{m_i})$ will be insignificant unless the clusters are arranged at different scales (i.e. some clusters are very close to each other but far from others). Similarly, $(\min(\mathcal{M})/\max(\mathcal{M}))^2$ will be nearly 1 if the maximal densities of the clusters are comparable. These observations suggest to estimate the number of clusters by selecting a cutoff τ and setting

$$\hat{K} = \inf\left\{ k \,\middle|\, \frac{\mathcal{D}_t(x_{m_k})}{\mathcal{D}_t(x_{m_{k+1}})} > \tau \right\}, \tag{5.3}$$

as in Algorithm 1. Once the modes have been learned correctly, points may be clustered simply by labeling each mode as belonging to its own class, then requiring that every point has the same label as its nearest neighbor in diffusion distance of higher density.

Corollary 5.4 *Suppose $X = \bigcup_{k=1}^K X_k$. Let $\{x_{m_i}\}_{i=1}^n$ be the points $\{x_i\}_{i=1}^n$, sorted so that $\mathcal{D}_t(x_{m_1}) \geq \mathcal{D}_t(x_{m_2}) \geq \cdots \geq \mathcal{D}_t(x_{m_n})$. Then Algorithm 1 labels all points correctly for any τ satisfying*

$$\frac{\max(\mathcal{M})}{\min(\mathcal{M})} \frac{\max_{i=1,\cdots,K} \rho_t(x_{m_i})}{\min_{i=1,\cdots,K} \rho_t(x_{m_i})} < \tau < \frac{\min(\mathcal{M})}{\max(\mathcal{M})} \frac{D_t^{btw}}{D_t^{in}}.$$

Proof By Corollary 5.2, the algorithm correctly estimates $\hat{K}$. Then, by Theorem 5.1, the algorithm correctly learns the empirical density maximizers of each of the $\{X_k\}_{k=1}^K$. It remains to show that the subsequent labeling of all points is accurate. For an unlabeled point $x \in X_k$, its nearest diffusion neighbor of higher density, x^*, must be in the same cluster X_k, since $D_t^{\text{in}} < D_t^{\text{btw}}$. Moreover, that point is already labeled as $Y(x^*) = k$, since $p(x^*) \geq p(x)$. Hence, $Y(x) = k$ and by induction, all points are labeled correctly. ■

The dependence on τ is somewhat unsatisfying, and in practice, this quantity can be removed from the inputs of Algorithm 1 by instead setting $\hat{K} = \arg\max_k \mathcal{D}_t(x_{m_k})/\mathcal{D}_t(x_{m_{k+1}})$. This provably detects K accurately by noting that the ratios $\mathcal{D}_t(x_{m_{j+1}})/\mathcal{D}_t(x_{m_j})$ are small for $j > K$ under a range of reasonable assumptions, for example the assumptions that the density of each cluster is bounded away from 0 and the ratio of the minimal and maximal within-cluster diffusion distance is bounded.
Algorithm 2 describes the LUND algorithm in the simpler case that K is known a priori. This algorithm achieves perfect accuracy under milder conditions than Algorithm 1.

Corollary 5.5 *Suppose $X = \bigcup_{k=1}^K X_k$ and K is known. If $\frac{D_t^{in}}{D_t^{btw}} < \frac{\min(\mathcal{M})}{\max(\mathcal{M})}$, then Algorithm 2, labels all points correctly.*

Proof This follows from Theorem 5.1, along with $D_t^{\text{in}} < D_t^{\text{btw}}$. ■

6. Numerical Experiments

We return to the motivating data sets of Section 3. The diffusion distances are computed by truncating (2.2) to sum only over the largest (in terms of modulus of the eigenvalues) $M = 100 \ll n$ eigenpairs, and the KDE $p(x)$ uses 100 nearest neighbors with $\sigma_0 = 1$.
We compute a number of statistics on the data to test our theoretical estimates and to verify the efficacy of the proposed algorithm. For the first two data sets we examine, we plot $D_t^{\text{in}}, D_t^{\text{btw}}$ as functions of t, to observe the multitemporal nature of our clustering algorithm. We also compute the theoretical estimates on $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$ as guaranteed by Theorem 4.8. The tightness of the theoretical estimates is evaluated

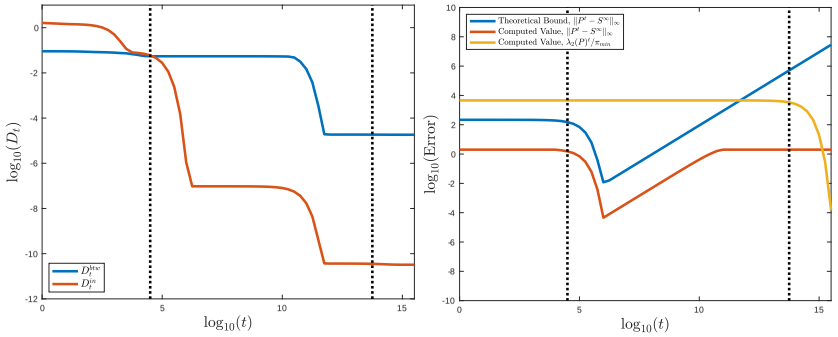
by comparing to the empirical values. We moreover plot the diffusion distances from a fixed point for a variety of t values, to illustrate the multitemporal behavior of these distances.
After these evaluations, we cluster the data with the proposed LUND algorithm and compute the accuracy, comparing with spectral clustering and the FSFDPC algorithm. We moreover compute the estimates of K with both the proposed method $\hat{K} = \arg\max_k \mathcal{D}_t(x_{m_k})/\mathcal{D}_t(x_{m_k+1})$ where $\{x_{m_i}\}_{i=1}^n$ are the points $\{x_i\}_{i=1}^n$ sorted so that $\mathcal{D}_t(x_{m_1}) \geq \mathcal{D}_t(x_{m_2}) \geq \cdots \geq \mathcal{D}_t(x_{m_n})$, and spectral clustering eigengap $\hat{K} = \arg\max_i \lambda_{i+1} - \lambda_i$, as a function of the crucial parameters of the respective algorithms. In particular, we evaluate the robustness of spectral clustering methods with respect to σ, and LUND with respect to σ and t. For spectral clustering, we consider the variant in which just the second eigenvector ψ_2 is used (Shi and Malik, 2000), as well as the variant in which the first K eigenvectors $\{\psi_i\}_{i=1}^K$ are used (Ng et al., 2002). All experiments are conducted on randomly generated data, with results averaged over 100 trials.

6.1 Bottleneck Data

We first analyze the linear, multimodal data set of Figure 2, in which two of the clusters feature two high-density regions, connected by a lower density bottleneck region. Theorem 4.8 upper bounds $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty < \epsilon$ in terms of $\delta, \lambda_{K+1}, \kappa$, which for this data have values $\delta = 6.2697 \times 10^{-8}$, $\lambda_{K+1} = 1 - 1.7563 \times 10^{-4}$, $\kappa = 2.6738 \times 10^2$ when $\mathbf{P}$ is constructed with $\sigma = .15$. As shown in Figure 9, the theoretical estimate correctly illustrates the overall behavior of the transition from initial distribution, to mesoscopic equilibria, then to a global equilibrium.
The distance from a high-density point across time scales appears in Figure 10. For small time values, the diffusion distance scales similarly to Euclidean distance. However, by time $t = 10^8$, a mesoscopic equilibrium has been reached, and all points in the cluster are rather close together. By $t = 10^{16}$, a global equilibrium has been reached.

6.1.1 Bottleneck Data Clustering Evaluation

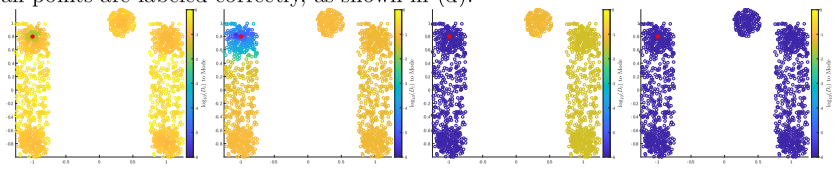
Comparisons with spectral clustering appear in Figures 11 and 12. In Figure 12 (a), it is shown that for all values of the spatial scale parameter σ, the eigengap estimated number of clusters $\hat{K}$ is 1, i.e. always incorrect. On the other hand, Figure 12 (b) shows that there is a range of (σ, t) values—mesoscopic in t—for which LUND achieves perfect accuracy. Indeed, after an initial phase in which the number of clusters is estimated as 1, the LUND estimate for K is decreasing in t, corresponding to the mixing of different clusters over time.
The LUND algorithm and FSFDPC are compared in Figure 13. Due to the non-spherical shapes of the clusters, FSFDPC is unable to learn the modes of the data correctly, and consequently assigns modes to the same cluster: the modes learned



(a) Plot of $D_t^{\text{btw}}, D_t^{\text{in}}$ against t. (b) Quantities related to mesoscopic and global equilibria.

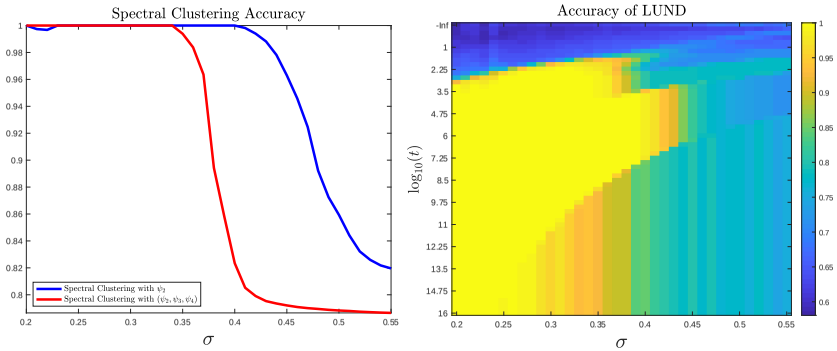
Figure 9: In (a), $D_t^{\text{btw}}, D_t^{\text{in}}$ are plotted against t. For $t < 10^{4.5}$, $D_t^{\text{in}} > D_t^{\text{btw}}$, since for small time, D_t is essentially the same as Euclidean distance. Around $t = 10^{4.5}$, there is a transition, in which $D_t^{\text{in}} \ll D_t^{\text{btw}}$. This corresponds to the Markov chain reaching mesoscopic equilibria in which the chain is well-mixed on each cluster, but not well-mixed globally. The approximate times of convergence towards the mesoscopic and global equilibrium are denoted with dotted black vertical lines. In (b), we plot three quantities against t: the theoretical bound on $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$ guaranteed by Theorem 4.8; the empirical quantity $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$; and the empirically computed quantity $\lambda_2(\mathbf{P})^t/\boldsymbol{\pi}_{\min}$, which estimates the distance to the stationary distribution. Notice that $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$ gets small, both the theoretical bound and the empirical value, around $t = 10^{4.5}$. It then increases. Around $t = 10^{13.75}$, $\lambda_2(\mathbf{P})^t/\boldsymbol{\pi}_{\min}$ decays exponentially to 0, indicating that the global equilibrium has been reached. Note that the theoretical estimate on $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$ is not tight, though it accurately captures the overall behavior of the quantity with respect to t. Moreover, the global decay estimate $\lambda_2(\mathbf{P})^t/\boldsymbol{\pi}_{\min}$ is somewhat conservative in estimating the mixing time of towards the equilibrium distribution, as can be seen by comparing the second vertical lines in (a) and (b). In order to keep the plots at the same scale, the plots in (a) are extended to be constant for roughly $t > 12$.

by FSFDPC and subsequent labels appear in subfigures (a), (b), respectively. In contrast, LUND learns one mode from each cluster, as shown in (c). Consequently, all points are labeled correctly, as shown in (d).



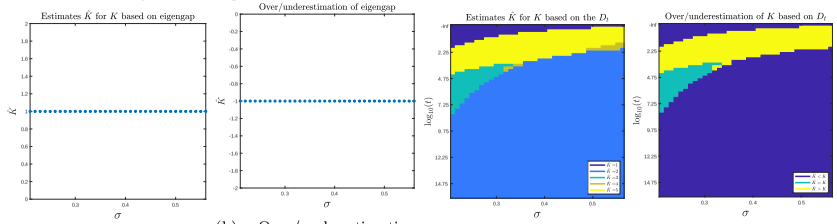
(a) $t = 0$ (b) $t = 10^4$ (c) $t = 10^8$ (d) $t = 10^{16}$

Figure 10: A high-density point is shown in red, and all other points are colored by the D_t distance from this point in $\log_{10}$ scale. The transition from initial distribution (a), to mesoscopic equilibrium (c), to global equilibrium (d), is illustrated as t grows.



(a) Accuracy spectral clustering (b) Accuracy of LUND

Figure 11: Accuracy of two variations of spectral clustering compared to LUND as functions of σ. While spectral clustering with ψ_2 performs nearly perfectly for $\sigma < .4$, its performance degrades as σ increases. Classical spectral clustering using ψ_2, ψ_2, ψ_4 achieves perfect clustering of the data for roughly $\sigma < .35$. LUND is able to achieve perfect clustering accuracy for a wide range of (σ, t) pairs, mainly for those σ values which allows spectral clustering with just ψ_2 to succeed. As σ increases, the mesoscopic regime in which perfect accuracy is achieved shrinks before disappearing entirely around $\sigma = .45$. In this data, spectral clustering with just ψ_2 performs about as well as LUND in terms of accuracy, assuming K is known.

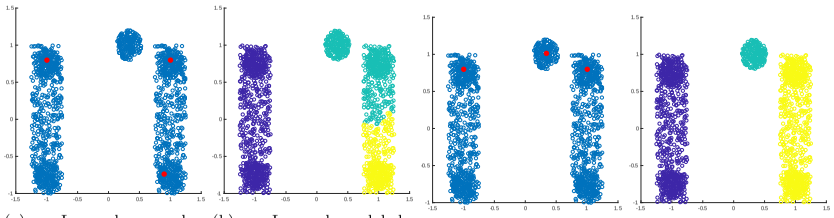


(a) Estimates of $\hat{K}$ using eigengap statistic. (b) Over/underestimation of $\hat{K}$ using eigengap statistic. (c) Estimates of $\hat{K}$ using $\mathcal{D}_t$ statistic. (d) Over/underestimation of $\hat{K}$ using $\mathcal{D}_t$ statistic.

Figure 12: In (a), we see the estimates of $\hat{K}$ using the eigengap statistic, as a function of spatial scale parameter σ. The eigengap consistently estimates $\hat{K} = 1 < 3 = K$, indicating that the multimodal nature of this data is too complicated for the spectral clustering eigengap to handle. A quantized version of these estimates is shown in (b), in which entry 0 indicates correct estimation, -1 indicates $\hat{K} < K$ and 1 indicates $\hat{K} > K$. There is a regime of (σ, t) values in which LUND correctly estimates the number of clusters, as shown in (c) and (d). After an initial time, this regime is essentially monotonic decreasing in t, and the mesoscopic region in which $\hat{K} = K$ is decreasing in σ.

6.2 Nonlinear Data

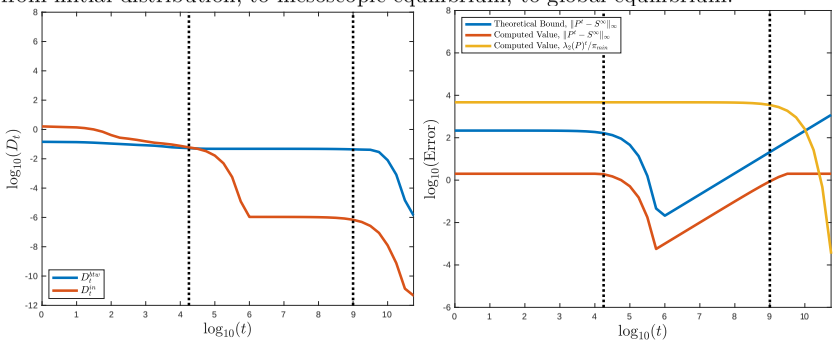
We now consider the nonlinear multimodal data of Figure 2 (b). The innermost circle is filled-in, and has only one high-density region. It is surrounded by two circles, each with two high-density regions connected by low-density regions. The paths connecting



(a) Learned modes, FSFDPC (b) Learned labels, FSFDPC (c) Learned modes, LUND (d) Learned labels, LUND

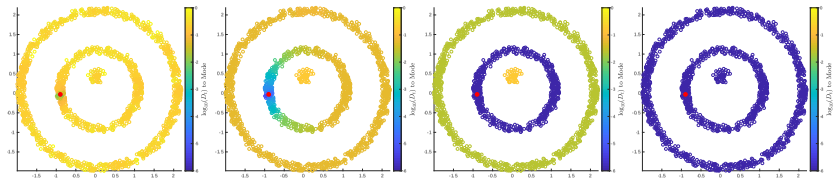
Figure 13: Comparison of FSFDPC to LUND. In (a), the modes learned from FSFDPC—with Euclidean distances—are plotted. Due to the eloganted, non-spherical nature of the data, the modes are learned incorrectly. The subsequent labels, shown in (b), illustrate FSFDPC is not able to capture the structure of this data. In (c), the modes learned from LUND are shown. One mode is learned from each cluster, which allows for a correct labeling of all data points with LUND, as shown in (d). LUND used parameters $(\sigma, t) = (.15, 10^6)$ for these data.

antipodal points on the outer circles are long, which suggests these sets will have low conductance. In the context of Theorem 4.8, the parameters for this data have values $\delta = 1.7225 \times 10^{-4}, 1 - \lambda_{K+1} = 6.8350 \times 10^{-5}, \kappa = 2.655 \times 10^2$ with $\sigma = .2$. Comparison of theoretical and empirical estimates appear in Figure 14, and the diffusion distances from one of the high-density points appear in Figure 15, illustrating the transition from initial distribution, to mesoscopic equilibrium, to global equilibrium.



(a) Plot of $D_t^{\text{btw}}, D_t^{\text{in}}$ against t. (b) Quantities related to mesoscopic and global equilibria.

Figure 14: In (a), we plot $D_t^{\text{btw}}, D_t^{\text{in}}$ against t. For roughly $t < 10^{4.25}$, $D_t^{\text{in}} > D_t^{\text{btw}}$; around $t = 10^{4.25}$, there is a transition, in which $D_t^{\text{in}} \ll D_t^{\text{btw}}$. This corresponds to the Markov chain reaching mesoscopic equilibria in which the chain is well-mixed on each cluster, but not well-mixed globally. The approximate times of convergence towards the mesoscopic and global equilibrium are denoted with dotted black vertical lines. In (b), we plot three quantities against t: the theoretical bound on $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$ guaranteed by Theorem 4.8; the empirically computed quantity $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$; and the empirically computed quantity $\lambda_2(\mathbf{P})^t/\boldsymbol{\pi}_{\min}$, which estimates the distance to the stationary distribution. Notice that $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty$ gets small, both the theoretical bound and the empirical value, around $t = 10^{4.25}$. It then increases. Around $t = 10^9$, $\lambda_2(\mathbf{P})^t/\boldsymbol{\pi}_{\min}$ decays exponentially to 0, indicating that the global equilibrium has been reached.

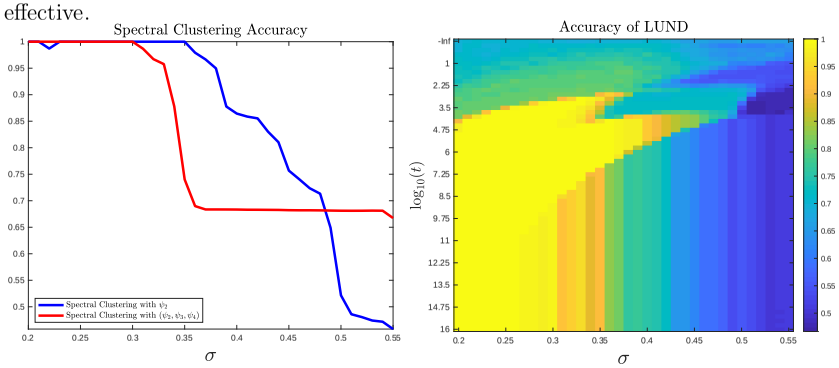


(a) $t = 0$ (b) $t = 10^4$ (c) $t = 10^8$ (d) $t = 10^{16}$

Figure 15: A high-density point is shown in red, and all other points are colored by D_t distance from this point in $\log_{10}$ scale. The transition from initial distribution (a), to mesoscopic equilibrium (c), to global equilibrium (d), is illustrated as t grows.

6.2.1 Nonlinear Data Clustering Evaluation

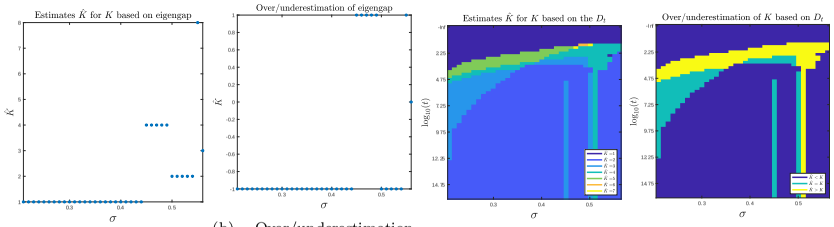
LUND is compared with the two spectral clustering variants in Figures 16 and 17. In terms of overall accuracy, LUND with the correct choice of t outperforms both methods of spectral clustering—using ψ_2 only and using ψ_2, ψ_3, ψ_4—for a range of σ values. The strong performance of LUND in the mesoscopic range, away from $t = 0, t = \infty$, confirms the theoretical results, and demonstrates LUND's flexibility compared to classical spectral methods. Beyond accuracy, the LUND estimator for K is empirically effective for a range of (σ, t) values, while the eigengap is much less effective.



(a) Accuracy spectral clustering (b) Accuracy of LUND

Figure 16: Accuracy of two variations of spectral clustering compared to LUND as functions of σ. While spectral clustering with ψ_2 performs well for very small σ, its performance degrades as σ increases; classical spectral clustering using ψ_2, ψ_3, ψ_4 performs similarly though for a smaller range of σ. LUND is able to achieve perfect clustering accuracy for a wide range of (σ, t) pairs, in particular for pairs (σ, t) such that spectral clustering fails. As σ increases, the mesoscopic regime in which perfect accuracy is achieved shrinks before disappearing entirely around $\sigma = .4$. LUND outperforms spectral clustering with (ψ_2, ψ_3, ψ_4) roughly for $\sigma \in (.3, .4)$, and outperforms spectral clustering with ψ_2 alone roughly for $\sigma \in (.35, .4)$.

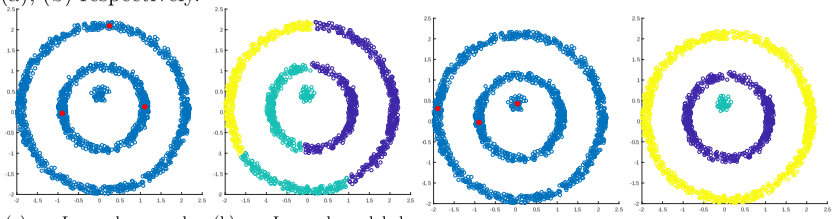
In Figure 18, LUND is compared to FSFDPC. LUND correctly learns the modes of the data and labels points correctly, as shown in (c), (d). FSFDPC, however, fails to



(a) Estimates of $\hat{K}$ using eigengap statistic. (b) Over/underestimation of $\hat{K}$ using eigengap statistic. (c) Estimates of $\hat{K}$ using $\mathcal{D}_t$ statistic. (d) Over/underestimation of $\hat{K}$ using $\mathcal{D}_t$ statistic.

Figure 17: In (a) and (b), we see the eigengap consistently misestimates the number of clusters in the data, first estimating $\hat{K} = 1$, before oscillating between various numbers of clusters, including ending on $\hat{K} = 3$ for one value of σ. LUND is able to achieve an accurate estimate for $\hat{K}$ for a range of (σ, t) values, with generally more t values yielding a correct estimate for smaller σ. We note that the vertical lines in (c), (d) around $\sigma = .45$ and $\sigma = .5$ that deviate from the surrounding values are due to using the mode as the summary statistic across the trials. If the mean is used, a much smoother plot emerges, though one that is less meaningful in terms of estimating the number of clusters.

learn the modes of the data correctly, leading to erroneous labeling—see subfigures (a), (b) respectively.



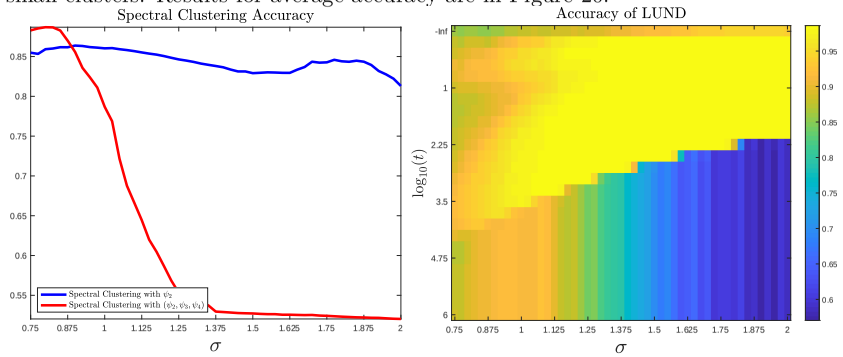
(a) Learned modes, FSFDPC (b) Learned labels, FSFDPC (c) Learned modes, LUND (d) Learned labels, LUND

Figure 18: In (a), the modes learned from FSFDPC—with Euclidean distances—are plotted. The nonlinear nature of the data causes the modes to be learned incorrectly. The subsequent labels, shown in (b), illustrate FSFDPC is not able to capture the structure of this data. In (c), the modes learned from LUND are shown. One mode is learned from each cluster, which allows for a correct labeling of all data points with LUND, as shown in (d). LUND used parameters $(\sigma, t) = (.175, 10^5)$ for these data.

6.3 Gaussian Data

As a final synthetic example, we consider the Gaussians of Figure 1, which were constructed by Nadler and Galun (2007) to be data on which both variants of spectral clustering fail. These data are not sufficiently separated for Theorem 4.12 to apply, but LUND still is able to perform well, owing to the incorporation of density, which allows to easily estimate the modes of the data. Comparisons to spectral clustering in terms of overall accuracy are reported in Figure 19. It is also enlightening to consider

performances of LUND and spectral clustering in terms of *average accuracy*, in which the overall accuracy on each of the clusters is computed separately, and these class-wise accuracies are then averaged. Compared to the overall accuracy measure, the average accuracy measure discounts large clusters and increases the significance of small clusters. Results for average accuracy are in Figure 20.



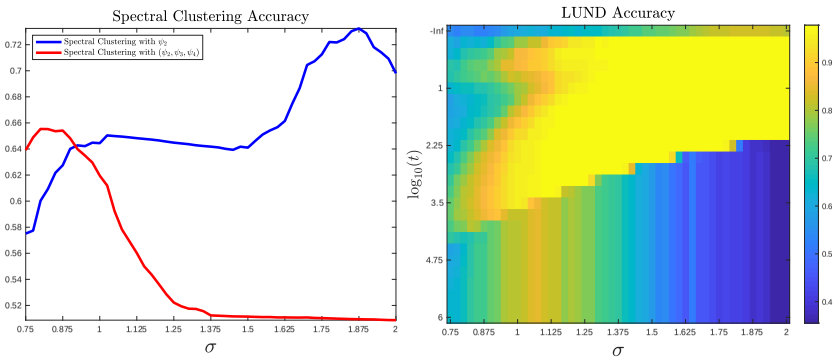
(a) Accuracy of spectral clustering (b) Accuracy of LUND

Figure 19: The *overall accuracy* of the two spectral clustering variants, as well as LUND, are shown for the Gaussian data. In terms of overall accuracy, LUND is able to achieve near-perfect results for a range of parameter values. Nearly all errors made were due to a point being generated from one Gaussian and landing very close to another Gaussian, which is essentially an unavoidable identifiability issue from which any unsupervised clustering method would suffer. Neither of the spectral clustering methods is able to match LUND's performance, which can be attributed to fundamental issues with the use of only the first small number of eigenvectors when performing spectral clustering, as shown by Nadler and Galun (2007) and illustrated in Figure 1.

In Figure 21, LUND is compared to FSFDPC. LUND correctly learns the modes of the data and labels points correctly, as shown in (c), (d). FSFDPC also learns the modes correctly, due to the unimodality of the Gaussian clusters and their isotropic covariance matrices.

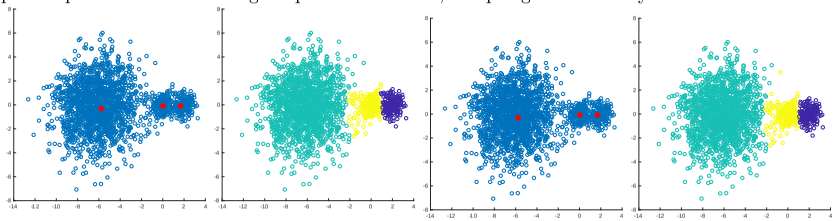
6.4 Experimental Conclusions

In all three synthetic examples, LUND performs well. On the bottleneck data, it gives the same accuracy as spectral clustering with ψ_2 but better estimates on $\hat{K}$; on the nonlinear data, it gives the best range of accuracies with respect to σ, while also giving the best estimates of $\hat{K}$; on the Gaussian data, LUND performs as well as FSFDPC while both spectral methods fail. These results suggest that LUND combines the best properties of spectral clustering with density-based clustering, while enjoying theoretical guarantees. We remark that extensive experiments with a variant of LUND adapted to high-dimensional images were performed on real hyperspectral image data (Murphy and Maggioni, 2018a,b, 2019b), demonstrating the competitiveness of LUND with a range of benchmark and state-of-the-art clustering algorithms.



(a) Accuracy of spectral clustering (b) Accuracy of LUND

Figure 20: The *average accuracy* of the two spectral clustering variants, as well as LUND, are shown for the Gaussian data. The results are qualitatively similar to overall accuracy, but with reduced performance for spectral clustering, since most of the errors made by the spectral clustering variants are on the small cluster, which is washed out by spectral clustering. LUND achieves essentially perfect performance for a range of parameter values, excepting identifiability issues.



(a) Learned modes, FSFDPC (b) Learned labels, FSFDPC (c) Learned modes, LUND (d) Learned labels, LUND

Figure 21: For the Gaussian data, both LUND and FSFDPC learn the data modes correctly, and are subsequently able to label the data with high accuracy. The lack of difference in their comparative performances is attributed to the fact that the data in this case are Gaussians with isotropic covariance matrices, and in particular have simple spherical supports, which confers diffusion distances little advantage compared to Euclidean distances. LUND used parameters $(\sigma, t) = (1, 10^3)$ for these data.

As shown in Figures 11, 12, 16, 17, 19, 20, the relationship between σ and t is not as simple as suggested by equation (4.18). Indeed, in the non-asymptotic case, and in particular in the case of well-separated clusters, the relationship between t and σ does not obey a strict exponential relationship, as suggested by (4.18). Instead, t appears to interact with scales *locally* on each cluster, as can be seen by the bifurcations in these plots. Gaining a complete understanding of the relationship between σ and t in the cluster case is a topic of ongoing research.

6.5 Computational Complexity

The proposed algorithm enjoys essentially linear computational complexity. This is achieved through the indexing structure *cover trees* (Beygelzimer et al., 2006), which allows for efficient nearest neighbor searches under the assumption that data has low-dimensional structure. Indeed, for data $\{x_i\}_{i=1}^n \subset \mathbb{R}^D$, the computation of each of the n data point's k_{nn} nearest neighbors can be achieved at a cost of $O(k_{\text{nn}} DC^d n \log(n))$, where d is the intrinsic dimension of the data. This allows for the computation of each point's empirical density estimate $p(x)$ in $O(k_{\text{nn}} DC^d n \log(n))$, where k_{nn} is the number of nearest neighbors used in the density estimate. In addition, if the Markov chain is computed not on a fully connected graph, but on a k_{nn} nearest neighbors graph, the cost of computing diffusion distances with the eigenvector approximation is $O(k_{\text{nn}} DC^d n \log(n) + k_{\text{nn}} M^2 n)$, where M eigenvectors are used in the approximation. In the case that $k_{\text{nn}} = O(\log(n)), M = O(1)$, these complexities simplify to $O(DC^d n \log(n)^2)$. Computing all $\rho_t(x)$ values is $O(n \log(n))$, under the assumption that, except for the $O(\log(n))$ class modes, each point has among its $O(\log(n))$ nearest neighbors a point of higher empirical density. Subsequent sorting of p and D_t are $O(n \log(n))$, so the overall algorithm is $O(DC^d n \log^2(n))$, which is linear in n, up to logarithmic factors.

6.6 Discussion and Tuning of Parameters

The LUND algorithm depends crucially on the parameters σ, t, and also potentially on the cutoff parameter τ used to estimate K as in (5.3). Our experiments have demonstrated that LUND is robust to σ and t, and that the parameter τ may be dispensed with by estimating $\hat{K} = \arg\max_k \mathcal{D}_t(x_{m_k})/\mathcal{D}_t(x_{m_{k+1}})$.

We remark that σ may be set *locally* in an automated way by replacing a global σ with a point-dependent σ based on distances to near neighbors (Zelnik-Manor and Perona, 2005). This approach modifies the construction of $\mathbf{P}$ by setting $\mathbf{W}_{ij} = \exp(-\|x_i - x_j\|_2^2/\sigma_i\sigma_j)$, where $\sigma_i = \|x_i - NN(x_i, k_{\text{NN}})\|_2$, where $NN(x_i, k_{\text{NN}})$ is the k_{NN}^{th} Euclidean nearest neighbor of x_i. Which nearest neighbor to use, k_{NN}, is a function of the data dimension, but the approach is practically quite robust to this parameter. This has the effect of adapting to the local density near each point. Moreover, it suggests that if the underlying density is smooth and bounded away from 0, then for all x_i, there is a natural scaling $\sigma_i \to 0^+$ as $n \to \infty$, with rate depending on the data dimensionality.

Regarding t, this parameter determines the time scale at which the diffusion is run. The theoretical results guarantee that if a certain clustering of the data admits suitable separation and geometric properties, then there will be choices of t for which LUND accurately labels the data. In practice, choosing t without supervision can be performed in several ways. One approach is to choose the t which maximizes the

ratio between successive values of $\mathcal{D}_t(x)$:

$$t^* = \arg\max_t \left\{ \arg\max_k \mathcal{D}_t(x_k)/\mathcal{D}_t(x_{k+1}) \right\}.$$

This is analogous to the method for selecting the optimal parameters for ultrametric spectral clustering in Little et al. (2017). A different approach is to select t^* such that clustering with LUND at time t^* is "stable" (Wan and Meila, 2016; Meila, 2019) with respect to small perturbations in the underlying data. Relatedly, it is natural to select t such that the clustering results are stable to small perturbations in t.

We remark that in some sense, multiple choices of t, leading to different clusterings, may be reasonable. This is because there may be hierarchical cluster structure in the data, parametrized by t. Developing extensions and theoretical analyses of LUND in this context is the subject of ongoing research.

7. Conclusions and Future Work

In this article, new methods for bounding diffusion distances, based on nearly-reducible Markov chains, are deployed to provide sufficient conditions under which clustering of data can be guaranteed. The theoretical results rigorously show that diffusion distances exhibit multitemporal behavior, even in the case that clusters have multiple regions of high-density or nonlinear support. These estimates on diffusion distance allow to prove performance guarantees on the LUND algorithm. This may be interpreted as a critique of the popular FSFDPC algorithm, for which theoretical guarantees require unrealistic assumptions on the shapes of the clusters. Numerical experiments on bottleneck, nonlinear, and Gaussian data indicate that the theoretical results correspond with empirical performance, and that LUND enjoys advantages of both spectral clustering and FSFDPC while tempering their weaknesses.

While the results presented in this paper for diffusion distances are, we believe, novel and useful for developing performance guarantees for unsupervised learning, they fall short of a full finite sample analysis. It is of interest to understand how the estimates on δ and λ_{K+1}, which govern the estimates of Theorem 4.12, and consequently the performance guarantees for clustering, scale with the number of sample points. Developing such precise estimates would require new mathematical methods for analyzing the spectra of random operators on graphs. Such an analysis is suggested by recent works in discrete-to-continuum spectral analysis (Garcia Trillos et al., 2018), though handling the factor $(\mathbf{I} - \mathbf{P}_{ii})^{-1}$ may provide for new challenges.

As remarked in Section 6, diffusion distances are Euclidean distances in a new coordinate basis, given by the (right) eigenvectors of $\mathbf{P}$. A different approach to proving the clustering properties of D_t with respect to time would be to show that different eigenvectors localize on particular clusters, and show that there are gaps in the eigenvalues λ_ℓ which account for the emergence of mesoscopic equilibria. This approach is related to the analysis of eigenvectors corresponding to small eigenvalues for the symmetric Laplacian (Shi et al., 2009), and may provide new insights.

In Section 4.6, the asymptotic relationship between σ and t for diffusion distances is discussed (Lafon et al., 2006). As shown in Section 6, LUND exhibits a more delicate relationship between σ, t in the finite sample case. This is potentially due to the incorporation of empirical density into the diffusion geometry framework that LUND uses to estimate the data modes. Analyzing this phenomenon mathematically is a topic of future research.

In LUND, the modes are learned sequentially, by selecting the K maximizers of $\mathcal{D}_t$. It is of interest to consider whether jointly selecting the modes, based on a criterion that optimizes over K points simultaneously, leads to different and in some cases improved clustering results in particular situations. Such a formulation may force the modes to be better spread throughout the data, and in particular would force the modes in the bottleneck data in Figure 4 to localize on the clusters, regardless of the fluctuations in empirical density. For this data, it does not affect the clustering accuracy if the fourth mode is in the middle of the bottleneck or in a cluster, but it is intuitive that the modes should localize on the four cluster cores rather than anywhere along the bottleneck. Developing a joint mode selection criterion may enforce this.

The proposed method also lends itself to the semisupervised setting of *active learning* (Chapelle et al., 2006; Dasgupta, 2011), in which the user is allowed to query a small number of points for labels. By estimating which points are most likely to be modes of clusters, the LUND algorithm presents natural candidates to query for labels. Recent work has suggested this approach is feasible (Murphy and Maggioni, 2018b, 2019a; Maggioni and Murphy, 2019), and it is an ongoing research effort to optimally integrate diffusion geometry into an active sampling scheme.

Acknowledgments

We thank the three anonymous reviewers, whose comments and insights greatly improved the manuscript. This project was partially funded by NSF-DMS-125012, NSF-DMS-1724979, NSF-DMS-1708602, NSF-ATD-1737984, AFOSR FA9550-17-1-0280, NSF-IIS-1546392, NSF-DMS 1912737, and NSF-DMS 1924513.

Appendix A. Proof of Theorem 4.8

Notice $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty \leq \|\mathbf{P}^t - \mathbf{S}^t\|_\infty + \|\mathbf{S}^t - \mathbf{S}^\infty\|_\infty$. For all $t \geq 0$, $\mathbf{P}^t - \mathbf{S}^t = \sum_{i=1}^t \mathbf{S}^{t-i}(\mathbf{P} - \mathbf{S})\mathbf{P}^{i-1}$, so that

$$\|\mathbf{P}^t - \mathbf{S}^t\|_\infty = \left\| \sum_{i=1}^t \mathbf{S}^{t-i}(\mathbf{P} - \mathbf{S})\mathbf{P}^{i-1} \right\|_\infty \leq \sum_{i=1}^t \|\mathbf{S}^{t-i}\|_\infty \|(\mathbf{P} - \mathbf{S})\|_\infty \|\mathbf{P}^{i-1}\|_\infty = t\|(\mathbf{P} - \mathbf{S})\|_\infty \leq t\delta.$$

To bound $\|\mathbf{S}^t - \mathbf{S}^\infty\|_\infty$, notice that after diagonalizing $\mathbf{S}$,

$$\mathbf{S}^t = \mathbf{Z} \begin{bmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^t \end{bmatrix} \mathbf{Z}^{-1}, \quad \mathbf{S}^\infty = \mathbf{Z} \begin{bmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{Z}^{-1},$$

where $\mathbf{D}$ is a diagonal matrix with $\lambda_{K+1}, \lambda_{K+2}, \ldots, \lambda_n$ on the diagonal. Hence, $\|\mathbf{S}^t - \mathbf{S}^\infty\|_\infty \leq \|\mathbf{Z}\|_\infty \lambda_{K+1}^t \|\mathbf{Z}^{-1}\|_\infty = \kappa \lambda_{K+1}^t$, as desired. The second result of the theorem follows similarly.

References

D. Aldous and J. Fill. Reversible Markov chains and random walks on graphs, 2002.

R. Andersen and Y. Peres. Finding sparse cuts locally using evolving sets. In *Symposium on Theory of Computing (STOC)*, pages 235–244. ACM, 2009.

R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Symposium on Foundations of Computer Science (FOCS)*, pages 475–486. IEEE, 2006.

R. Andersen, F. Chung, and K. Lang. Local partitioning for directed graphs using pagerank. *Internet Mathematics*, 5(1-2):3–22, 2008.

E. Arias-Castro. Clustering based on pairwise distances when the data is of mixed dimensions. *IEEE Transactions on Information Theory*, 57(3):1692–1706, 2011.

E. Arias-Castro, G. Chen, and G. Lerman. Spectral clustering based on local linear approximations. *Electronic Journal of Statistics*, 5:1537–1587, 2011.

E. Arias-Castro, G. Lerman, and T. Zhang. Spectral clustering based on local PCA. *Journal of Machine Learning Research*, 18(9):1–57, 2017.

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Symposium on Discrete Algorithms (SODA)*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

A. Athreya, D.E. Fishkind, M. Tang, C.E. Priebe, Y. Park, J.T. Vogelstein, K. Levin, V. Lyzinski, and Y. Qin. Statistical inference on random dot product graphs: a survey. *The Journal of Machine Learning Research*, 18(1):8393–8484, 2017.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. In *International Conference on Computational Learning Theory (COLT)*, pages 486–500. Springer, 2005.

M. Belkin and P. Niyogi. Convergence of Laplacian eigenmaps. In *Advances in Neural Information Processing Systems (NIPS)*, pages 129–136, 2007.

A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *International Conference on Machine Learning (ICML)*, pages 97–104. ACM, 2006.

S. Botelho-Andrade, P.G. Casazza, D. Cheng, and R. Tran. The exact constant for the ℓ_1—ℓ_2 norm inequality. *Mathematical Inequalities & Applications*, 22(1):59–64, 2019.

J.E. Chacón. Clusters and water flows: a novel approach to modal clustering through morse theory. *arXiv preprint arXiv:1212.1384*, 2012.

O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised Learning*. MIT Press, 2006.

G. Chen and G. Lerman. Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Foundations of Computational Mathematics*, 9(5):517–558, 2009a.

G. Chen and G. Lerman. Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3):317–330, 2009b.

F. Chung. *Spectral Graph Theory*, volume 92. American Mathematical Soc., 1997.

R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

R.R Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S.W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–7431, 2005.

R.R. Coifman, I.G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5): 603–619, 2002.

W. Czaja, B. Manning, L. McLean, and J.M. Murphy. Fusion of aerial gamma-ray survey and remote sensing data for a deeper understanding of radionuclide fate after radiological incidents: examples from the Fukushima Dai-Ichi response. *Journal of Radioanalytical and Nuclear Chemistry*, 307(3):2397–2401, 2016.

S. Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412(19): 1767–1781, 2011.

S. Dasgupta and L. Schulman. A probabilistic analysis of EM for mixtures of separated, spherical Gaussians. *Journal of Machine Learning Research*, 8(Feb):203–226, 2007.

F. Den Hollander. *Large Deviations*, volume 14. American Mathematical Soc., 2008.

P. Diaconis and L. Saloff-Coste. Comparison theorems for reversible Markov chains. *The Annals of Applied Probability*, pages 696–730, 1993.

P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, pages 36–61, 1991.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (11):2765–2781, 2013.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, volume 96, pages 226–231, 1996.

K. Fountoulakis, D.F., and M.W. Mahoney. An optimization approach to locally-biased graph algorithms. *Proceedings of the IEEE*, 105(2):256–272, 2017.

J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in Statistics Springer, Berlin, 2001.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

N Garcia Trillos, D. Slepcev, J. Von Brecht, T. Laurent, and X. Bresson. Consistency of Cheeger and ratio graph cuts. *Journal of Machine Learning Research*, 17(181): 1–46, 2016.

N. Garcia Trillos, M. Gerlach, M. Hein, and D. Slepcev. Error estimates for spectral convergence of the graph Laplacian on random geometric graphs towards the Laplace–Beltrami operator. *arXiv preprint arXiv:1801.10108*, 2018.

C. Gardiner. *Stochastic Methods*, volume 4. springer Berlin, 2009.

M. Gavish and B. Nadler. Normalized cuts are approximately inverse exit times. *SIAM Journal on Matrix Analysis and Applications*, 34(2):757–772, 2013.

C.R. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Non-parametric inference for density modes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):99–126, 2016.

J.C. Gower and G.J.S. Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 18 (1):54–64, 1969.

J.A. Hartigan. Consistency of single linkage for high-density clusters. *Journal of the American Statistical Society*, 76(374):388–394, 1981.

M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989.

S. Jia, G. Tang, J. Zhu, and Q. Li. A novel ranking-based clustering approach for hyperspectral band selection. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):88–102, 2016.

P.W. Jones, M. Maggioni, and R. Schul. Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels. *Proceedings of the National Academy of Sciences*, 105(6):1803–1808, 2008.

S. Lafon, Y. Keller, and R.R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.

R.R. Lederman and R. Talmon. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3): 509–536, 2018.

R.R. Lederman, R. Talmon, H. Wu, Y.-L. Lo, and R.R. Coifman. Alternating diffusion for common manifold learning with application to sleep stage assessment. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5758–5762. IEEE, 2015.

D.A. Levin, Y. Peres, and E.L. Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.

R. Li, M.G. Frasch, and H. Wu. Efficient fetal-maternal ECG signal separation from two channel maternal abdominal ECG via diffusion-based channel selection. *Frontiers in physiology*, 8:277, 2017.

S. Ling and T. Strohmer. Certifying global optimality of graph cuts via semidefinite relaxation: A performance guarantee for spectral clustering. *Foundations of Computational Mathematics*, pages 1–55, 2018.

A. Little and A. Byrd. A multiscale spectral method for learning number of clusters. In *International Conference on Machine Learning and Applications (ICMLA)*, pages 457–460. IEEE, 2015.

A. Little, M. Maggioni, and J.M. Murphy. Path-based spectral clustering: Guarantees, robustness to outliers, and fast algorithms. *ArXiv:1712.06206*, 2017.

M. Maggioni and J.M. Murphy. Learning by active nonlinear diffusion. *Foundations of Data Science*, 1(3):271–291, 2019.

M. Meila. Good (k-means) clusterings are unique (up to small perturbations). *Journal of Multivariate Analysis*, 173:1–17, 2019.

M. Meila and J. Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 873–879, 2001.

C.D Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2):240–272, 1989.

D.G. Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, 6(4):389–415, 2017.

J.M. Murphy and M. Maggioni. Diffusion geometric methods for fusion of remotely sensed data. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXIV*, volume 10644, page 106440I. International Society for Optics and Photonics, 2018a.

J.M. Murphy and M. Maggioni. Iterative active learning with diffusion geometry for hyperspectral images. In *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5. IEEE, 2018b.

J.M. Murphy and M. Maggioni. Spectral-spatial diffusion geometry for hyperspectral image clustering. *arXiv preprint arXiv:1902.05402*, 2019a.

J.M. Murphy and M. Maggioni. Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1829–1845, 2019b.

B. Nadler and M. Galun. Fundamental limitations of spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1017–1024, 2007.

A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.

R. Ostrovsky, Y. Rabani, L.J. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-means problem. In *Symposium on Foundations of Computer Science (FOCS)*, pages 165–176. IEEE, 2006.

H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.

A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

M.A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of Chemical Physics*, 134 (12):03B624, 2011.

C. Rossant, S.N. Kadir, D.F.M. Goodman, J. Schulman, M.L.D. Hunter, A.N. Saleem, A. Grosmark, M. Belluscio, G.H. Denfield, A.S. Ecker, et al. Spike sorting for large, dense electrode arrays. *Nature Neuroscience*, 19(4):634, 2016.

S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

G. Schiebinger, M.J. Wainwright, and B. Yu. The geometry of kernelized spectral clustering. *The Annals of Statistics*, 43(2):819–846, 2015.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

T. Shi, M. Belkin, and B. Yu. Data spectroscopy: Eigenspaces of convolution operators and clustering. *The Annals of Statistics*, pages 3960–3984, 2009.

H.A. Simon and A. Ando. Aggregation of variables in dynamic systems. *Econometrica: Journal of the Econometric Society*, pages 111–138, 1961.

A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, 1989.

A. Singer and R.R. Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226–239, 2008.

A. Singer and H. Wu. Vector diffusion maps and the connection Laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012.

A. Singer and H. Wu. Spectral convergence of the connection Laplacian from random samples. *Information and Inference: A Journal of the IMA*, 6(1):58–123, 2016.

A. Singer, R. Erban, I.G. Kevrekidis, and R.R. Coifman. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. *Proceedings of the National Academy of Sciences*, 106(38):16090–16095, 2009.

P.H.A. Sneath. The application of computers to taxonomy. *Microbiology*, 17(1): 201–226, 1957.

M. Soltanolkotabi, E. Elhamifar, and E.J. Candes. Robust subspace clustering. *Annals of Statistics*, 42(2):669–699, 2014.

D.A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Symposium on Theory of Computing (STOC)*, pages 81–90. ACM, 2004.

D.A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.

D.A. Spielman and S.-H. Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

M.H. Spitzer, P.F. Gherardini, G.K. Fragiadakis, N. Bhattacharya, R.T. Yuan, A.N. Hotson, R. Finck, Y. Carmi, E.R. Zunder, W.J. Fantl, et al. An interactive reference framework for modeling a dynamic immune system. *Science*, 349(6244):1259425, 2015.

H. Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci.*, 4(12):801–804, 1957.

K. Sun, X. Geng, and L. Ji. Exemplar component analysis: A fast band selection method for hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 12(5):998–1002, 2015.

G. Szegö. Inequalities for certain eigenvalues of a membrane of given area. *Journal of Rational Mechanics and Analysis*, 3:343–356, 1954.

R. Talmon and H. Wu. Latent common manifold learning with alternating diffusion: analysis and applications. *Applied and Computational Harmonic Analysis*, 2018.

J.B. Tenenbaum, V. De Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007.

Y. Wan and M. Meila. Graph clustering: block-models and model free results. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2478–2486, 2016.

P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, 174:806–814, 2016.

X. Wang, K. Slavakis, and G. Lerman. Multi-manifold modeling in non-Euclidean spaces. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1023–1032, 2015.

H.F. Weinberger. An isoperimetric inequality for the n-dimensional free membrane problem. *Journal of Rational Mechanics and Analysis*, 5(4):633–636, 1956.

C. Wiwie, J. Baumbach, and R. Röttger. Comparing the performance of biomedical clustering methods. *Nature Methods*, 12(11):1033, 2015.

X. Xu, M. Ester, H.-P. Kriegel, and J. Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *International Conference on Data Engineering (ICDE)*, pages 324–331. IEEE, 1998.

H. Yin, A.R. Benson, J. Leskovec, and D.F. Gleich. Local higher-order graph clustering. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 555–564. ACM, 2017.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1601–1608, 2005.

T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012.

W. Zheng, M.A. Rohrdanz, M. Maggioni, and C. Clementi. Polymer reversal rate calculated via locally scaled diffusion map. *The Journal of Chemical Physics*, 134 (14):144109, 2011.

Journal of Machine Learning Research 20 (2019) 1-44 Submitted 3/13; Revised 8/18; Published 10/19

Sparse Kernel Regression with Coefficient-based ℓ_q-regularization

Lei Shi MASTONE1983@GMAIL.COM
Shanghai Key Laboratory for Contemporary Applied Mathematics
School of Mathematical Sciences, Fudan University
Shanghai, P. R. China
Xiaolin Huang XIAOLINHUANG@SJTU.EDU.CN
Institute of Image Processing and Pattern Recognition
Institute of Medical Robotics, Shanghai Jiao Tong University
MOE Key Laboratory of System Control and Information Processing
Shanghai, P. R. China
Yunlong Feng YLFENG@ALBANY.EDU
Department of Mathematics and Statistics, State University of New York at Albany
New York, USA
Johan A.K. Suykens JOHAN.SUYKENS@ESAT.KULEUVEN.BE
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10, Leuven, B-3001, Belgium

Editor: Mikhail Belkin

Abstract

In this paper, we consider the ℓ_q-regularized kernel regression with $0 < q \leq 1$. In form, the algorithm minimizes a least-square loss functional adding a coefficient-based ℓ_q-penalty term over a linear span of features generated by a kernel function. We study the asymptotic behavior of the algorithm under the framework of learning theory. The contribution of this paper is two-fold. First, we derive a tight bound on the ℓ_2-empirical covering numbers of the related function space involved in the error analysis. Based on this result, we obtain the convergence rates for the ℓ_1-regularized kernel regression which is the best so far. Second, for the case $0 < q < 1$, we show that the regularization parameter plays a role as a trade-off between sparsity and convergence rates. Under some mild conditions, the fraction of non-zero coefficients in a local minimizer of the algorithm will tend to 0 at a polynomial decay rate when the sample size m becomes large. As the concerned algorithm is non-convex, we also discuss how to generate a minimizing sequence iteratively, which can help us to search a local minimizer around any initial point.

Keywords: Learning Theory, Kernel Regression, Coefficient-based ℓ_q-regularization ($0 < q \leq 1$), Sparsity, ℓ_2-empirical Covering Number

1. Introduction and Main Results

The *regression* problem aims at estimating the function relations from random samples and occurs in various statistical inference applications. An output estimator of regression algorithms is usually expressed as a linear combination of features, i.e., a collection of candidate functions. As an important issue in learning theory and methodologies, *sparsity* focuses on studying the sparse representations of such linear combinations resulted from the algorithms. It is widely known that an ideal way to obtain the sparsest representations is to penalize the combinatorial coefficients by the ℓ_0-norm. However, the algorithms based on ℓ_0-norm often lead to an NP-hard discrete optimization problem (see e.g., Natarajan (1995)), which motives the researchers to consider the ℓ_q-norm ($0 < q \leq 1$) as the substitution. In particular, the ℓ_1-norm constrained or penalized algorithms have achieved great success in a wide range of areas from signal recovery (Candès et al. (2006)) to variable selection in statistics (Tibshirani (1996)). Recently, several theoretical and experimental results (see e.g., Candès et al. (2008); Chartrand (2007); Fan and Li (2001); Saad and Ö. Yılmaz (2010); Rakotomamonjy et al. (2011); Xu et al. (2012)) suggest that ℓ_q-norm with $q \in (0, 1)$ yields sparser solutions than the ℓ_1-norm to produce accurate estimation. Due to the intensive study on compressed sensing (see e.g., Donoho (2006)), the algorithms involving the ℓ_q-norm ($0 < q \leq 1$) have drawn much attention in the last few years and been used for various applications, including image denoising, medical reconstruction and database updating.

In this paper, we focus on the ℓ_q-regularized kernel regression. In form, the algorithms minimize a least-square loss functional adding a coefficient-based ℓ_q-penalty term over a linear span of features generated by a kernel function. We shall establish a rigorous mathematical analysis on the asymptotic behavior of the algorithm under the framework of learning theory.

Let X be a compact subset of $\mathbb{R}^d$ and $Y \subset \mathbb{R}$, ρ be a Borel probability distribution on $Z = X \times Y$. For $f : X \to Y$ and $(x, y) \in Z$, the *least-square loss* $(f(x) - y)^2$ gives the error with f as a model for the process producing y from x. Then the resulting target function is called *regression function* and satisfies

$$f_\rho = \arg\min \left\{ \int_Z (f(x) - y)^2 d\rho \middle| f : X \to Y, \text{measurable} \right\}.$$

From Proposition 1.8 in Cucker and Zhou (2007), the regression function can be explicitly given by

$$f_\rho(x) = \int_Y y d\rho(y|x), \qquad x \in X, \tag{1}$$

where $\rho(\cdot|x)$ is the conditional probability measure induced by ρ at x. In the supervised learning framework, ρ is unknown and one estimates f_ρ based on a set of observations $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m \in Z^m$ which is assumed to be drawn independently according to ρ. We additionally suppose that $\rho(\cdot|x)$ is supported on $[-M, M]$, for some $M \geq 1$ and each $x \in X$. This uniform boundedness assumption for the output is standard in most literature in learning theory (see e.g., Zhang (2003); Smale and Zhou (2007); Mendelson and Neeman (2010); Wu et al. (2006)). Throughout the paper, we will use these assumptions without any further reference. Usually one may get an estimator of f_ρ by minimizing the empirical loss functional $\frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2$ over a *hypothesis space*, i.e., a pre-selected function set on X.

In kernel regression, the hypothesis space is generated by a kernel function $K : X \times X \to \mathbb{R}$. Recall that $\{x_i\}_{i=1}^m$ is the input data of the observations. The hypothesis space considered here is taken to be the linear span of the set $\{K_{x_i}\}_{i=1}^m$. For $t \in X$, we denote by K_t the

©2019 Lei Shi, Xiaolin Huang, Yunlong Feng, and Johan A.K. Suykens.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/13-124.html.

function

$$K_t: \quad X \to \mathbb{R}$$
$$x \mapsto K(x,t).$$

Let $0 < q \leq 1$, the output estimator of ℓ_q-regularized kernel regression is given by $\hat{f}_q = \sum_{i=1}^m c_{q,i}^{\mathbf{z}} K_{x_i}$, where its coefficient sequence $\mathbf{c}_q^{\mathbf{z}} = (c_{q,i}^{\mathbf{z}})_{i=1}^m$ satisfies

$$\mathbf{c}_q^{\mathbf{z}} = \arg\min_{\mathbf{c}\in\mathbb{R}^m} \left\{ \frac{1}{m}\sum_{j=1}^m \left(y_j - \sum_{i=1}^m c_i K(x_j, x_i) \right)^2 + \gamma \|\mathbf{c}\|_q^q \right\}. \tag{2}$$

Here $\gamma > 0$ is called a *regularization parameter* and $\|\mathbf{c}\|_q$ denotes the ℓ_q-norm of $\mathbf{c}$. Recall that for any $0 < q \leq 1$ and any sequence $\mathbf{w} = (w_n)_{n=1}^\infty$, the ℓ_0-norm and ℓ_q-norm are defined respectively as

$$\|\mathbf{w}\|_0 = \sum_{n=1}^\infty I(w_n \neq 0) \text{ and } \|\mathbf{w}\|_q = \left(\sum_{n\in \mathrm{supp}(\mathbf{w})} |w_n|^q \right)^{1/q},$$

where $I(\cdot)$ is the indicator function and $\mathrm{supp}(\mathbf{w}) := \{n \in \mathbb{N} : w_n \neq 0\}$ denotes the support set of $\mathbf{w}$. Strictly speaking, $\|\cdot\|_0$ is not a real norm and $\|\cdot\|_q$ merely defines a quasi-norm when $0 < q < 1$ (e.g., see Conway (2000)).

From an approximation viewpoint, we are in fact seeking for a function to approximate f_ρ from the function set spanned by the kernelized dictionary $\{K_{x_i}\}_{i=1}^m$. The kernelized dictionary together with its induced learning models has been previously considered in literature. In a supervised regression setting, to pursue a sparse nonlinear regression machine, Roth (2004) proposed the ℓ_1-norm regularized learning model induced by the kernelized dictionary, namely, the kernelized Lasso. It is indeed a basis pursuit method, the idea of which can be dated back to Chen and Donoho (1994); Girosi (1998). It was in Wu and Zhou (2008) that a framework of analyzing the generalization bounds for learning models induced by the kernelized dictionary was proposed. The idea behind is controlling the complexity of the hypothesis space and then investigating the approximation ability as well as the data-fitting risk of functions in this hypothesis space via approximation and concentration techniques, which is a typical learning theory approach. Following this line, a series of interesting studies have been expanded for various learning models induced by the kernelized dictionary. For example, probabilistic generalization bounds for different models were derived in Shi et al. (2011); Wang et al. (2012); Shi (2013); Lin et al. (2014); Feng et al. (2016) and many others. However, it is worth pointing out that one is not suggested to simply treat the kernelized dictionary as commonly used dictionaries. This is because learning models induced by the kernelized dictionary may possess more flexibility. For the kernelized dictionary, the positive semi-definite constraint on the kernel function is removed. The removing of the positive semi-definite constraints allows us to utilize some specific indefinite kernels to cope with real-world applications, see e.g., Schleif and Tino (2015). Moreover, $\{K_{x_i}\}_{i=1}^m$ is a data-dependent dictionary. In a nonlinear regression setting, comparing with models induced by basis functions, having seen enough observations, the data-dependent dictionary can provide adequate information. Consequently, the local information of the regression function can be captured with this redundant dictionary. An illustrative example

of this observation can be found in Ando et al. (2008). Recently, learning with indefinite kernels has drawn much attention. Most of work focused on the algorithmic study, e.g., Loosli et al. (2016); Huang et al. (2017). The algorithm under consideration can provide a simple scenario for regularized indefinite kernel regression. However, the theoretical work on this aspect is still limited so far.

Compared with the algorithms involving ℓ_0-penalty, the ℓ_1-regularized algorithms can be efficiently solved by convex programming methods. When $0 < q < 1$, the problem (2) is a non-convex optimization problem. Many efficient approaches have been developed to solve the ℓ_q-minimization problem of this type, e.g., Candès et al. (2008); Chen et al. (2010); Huang et al. (2008); Lai and Wang (2011); Xu et al. (2012); but there is no approach that guarantees to find a global minimizer. Most proposed approaches are *descent-iterative* in nature. To illustrate the principle of the minimization process, we define the objective functional of algorithm (2) as

$$\mathscr{T}_{\gamma,q}(\mathbf{c}) = \|\mathbf{y} - \mathbb{K}\mathbf{c}\|_2^2 + \gamma\|\mathbf{c}\|_q^q, \qquad \forall \mathbf{c} \in \mathbb{R}^m, \tag{3}$$

where $\mathbf{y} = \left(\frac{y_1}{\sqrt{m}}, \cdots, \frac{y_m}{\sqrt{m}}\right) \in \mathbb{R}^m$ and $\mathbb{K} \in \mathbb{R}^{m\times m}$ with entries $\mathbb{K}_{i,j} = \frac{K(x_i,x_j)}{\sqrt{m}}, 1 \leq i,j \leq m$. Given $\gamma > 0$ and an initial point $\mathbf{c}_0$, the descent-iterative minimization approach generates a minimizing sequence $\{\mathbf{c}_k\}_{k=1}^\infty$ such that $\mathscr{T}_{\gamma,q}(\mathbf{c}_k)$ are strictly decreasing along the sequence. Thus any local minimizer, including the global minimizer, that a descent approach may find must be in the level set $\{\mathbf{c} \in \mathbb{R}^m : \mathscr{T}_{\gamma,q}(\mathbf{c}) < \mathscr{T}_{\gamma,q}(\mathbf{c}_0)\}$. Therefore, in both theory and practice, one may be only interested in the local minimizer around some pre-given initial point. Specifically, for our problem, a reasonable choice of the initial point would be the solution in the case $q = 1$.

Assumption 1 *For $\gamma > 0$ and $0 < q < 1$, we assume that the coefficient sequence $\mathbf{c}_q^{\mathbf{z}}$ of the estimator $\hat{f}_q$ is a local minimizer of the problem (2) and satisfies $\mathscr{T}_{\gamma,q}(\mathbf{c}_q^{\mathbf{z}}) < \mathscr{T}_{\gamma,q}(\mathbf{c}_1^{\mathbf{z}})$, where $\mathbf{c}_1^{\mathbf{z}}$ is the global minimizer of the problem (2) at $q = 1$.*

In Section 2.1, we shall present a scheme for searching a local minimizer of problem (2) by constructing a descent-iterative minimization process. The previous theoretical analysis about least-square regression with a coefficient-based penalty term is valid only for a convex learning model, e.g., the ℓ_q-regularized regression with $q \geq 1$. To enhance the sparsity, one expects to use a non-convex ℓ_q penalty, i.e., $0 < q < 1$, but no optimization approach guarantees to find a global minimizer of the induced optimization problem. There is still a gap between the existing theoretical analysis and the optimization process: the estimator needs to be globally optimal in the theoretical analysis while the optimization method can not ensure the global optimality of its solutions. Up to our knowledge, due to the non-convexity of ℓ_q term, there still lacks a rigorous theoretical demonstration to support its efficiency in non-parametric regression. In this paper, we aim to fill this gap by developing an elegant theoretical analysis on the asymptotic performances of estimators $\hat{f}_q$ satisfying Assumption 1, where these estimators can be generated by the scheme in Section 2.1 or another descent-iterative minimization process. Here we would like to point out that the established convergence analysis of $\hat{f}_q$ only requires $\hat{f}_q$ to be a stationary point around $\hat{f}_1$.

One aim of this work is to discuss the sparseness of the algorithms (2), which is characterized by the upper bounds on the fraction of the non-zero coefficients in the expression

$\hat{f}_q = \sum_{i=1}^m c_{q,i}^{\mathbf{z}} K_{x_i}$. In general, the total number of coefficients is as large as the sample size m. But for some kind of kernels such as polynomial kernels, the representation of $\hat{f}_q$ is not unique whenever the sample size becomes large. For the sake of simplicity, we restrict our discussion on a special class of kernels.

Definition 1 *A function $K : X \times X \to \mathbb{R}$ is called an admissible kernel if it is continuous and for any $k \in \mathbb{N}$, $(c_1, \cdots, c_k) \in \mathbb{R}^k$ and distinct set $\{t_1, \cdots, t_k\} \subset X$, $\sum_{j=1}^k c_j K_{t_j} = 0$ for all $x \in X$ implies $c_j = 0$, $j = 1, \cdots, k$.*

It should be noticed that an admissible kernel here is not necessarily symmetric or positive semi-definite. Several widely used kernel functions from multi-variate approximation satisfy the condition of the admissible kernels, e.g., exponential kernels, Gaussian kernels, inverse multi-quadric kernels, B-spline kernels and compact supported radial basis function kernels including Wu's functions (Wu (1995)) and Wendland's functions (Wendland (1995)). One may see Wendland (2005) and references therein for more details of these kernels. For the same reason, the kernel function is required to be universal in Steinwart (2003) when discussing the sparseness of support vector machines. It is noticed that most universal kernels (see Micchelli et al. (2006)) are also admissible kernels. If K is admissible, as long as the input data are mutually different, the representation of $\hat{f}_q$ is unique and the number of non-zero coefficients is given by $\|\mathbf{c}_q^{\mathbf{z}}\|_0$. In the followings, when we refer to the linear combination of $\{K_{x_i}\}_{i=1}^m$, we always suppose that $\{x_i\}_{i=1}^m$ is pairwise distinct. In our setting, this assumption can be almost surely satisfied if the data generating distribution ρ is continuous.

Except sparsity, another purpose of this paper is to investigate how the estimator $\hat{f}_q$ given by (2) approximates the regression function f_ρ. Let ρ_X be the marginal distribution of ρ on X. With a suitable choice of γ depending on the sample size m, we show that the estimator $\hat{f}_q$ (or $\pi_M(\hat{f}_q)$ see Definition 2) converges to f_ρ in the function space $L^2_{\rho_X}(X)$ as m tends to infinity. Here, for a Borel measure Q on X, the space $L^2_Q(X)$ consists of all the square-integrable functions with respect to Q and the norm is given by $\|f\|_{L^2_Q} = \left(\int_X |f(x)|^2 dQ\right)^{1/2}$.

In order to state our results, we further recall some notations used in this paper. We say that K is a *Mercer kernel* if it is continuous, symmetric and positive semi-definite on $X \times X$. Such a kernel can generate a *reproducing kernel Hilbert space* (RKHS) $\mathscr{H}_K$ (e.g., see Aronszajn (1950)). For a continuous kernel function K, define

$$\widetilde{K}(u, v) = \int_X K(u, x) K(v, x) d\rho_X(x). \tag{4}$$

Then one can verify that $\widetilde{K}$ is a Mercer kernel.

Being an important convex approach for pursuing sparsity, regularizing with ℓ_1-norm deserves special attention in its own right. The following theorem illustrates our general error analysis for ℓ_1-regularized kernel regression. The result is stated in terms of properties of the input space X, the measure ρ and the kernel K.

Theorem 1 *Assume that X is a compact convex subset of $\mathbb{R}^d$ with Lipschitz boundary, $K \in \mathscr{C}^s(X \times X)$ with $s > 0$ is an admissible kernel and $f_\rho \in \mathscr{H}_{\widetilde{K}}$ with $\widetilde{K}$ defined by (4).*

Let $0 < \delta < 1$ and

$$\Theta = \begin{cases} \frac{d+2s}{2d+2s}, & \text{if } 0 < s < 1, \\ \frac{d+2\lfloor s \rfloor}{2d+2\lfloor s \rfloor}, & \text{otherwise}, \end{cases} \tag{5}$$

where $\lfloor s \rfloor$ denotes the integral part of s. Take $\gamma = m^{\epsilon - \Theta}$ with $0 < \epsilon \le \Theta - \frac{1}{2}$. Then with confidence $1 - \delta$, there holds

$$\|\hat{f}_1 - f_\rho\|^2_{L^2_{\rho_X}} \le C_\epsilon \log(6/\delta) \left(\log(2/\delta) + 1\right)^6 m^{\epsilon - \Theta}, \tag{6}$$

where $C_\epsilon > 0$ is a constant independent of m or δ.

We shall prove Theorem 1 in Section 4.3 with the constant C_ϵ given explicitly. The convergence rate presented here improves the existing ones obtained in Wang et al. (2012); Shi et al. (2011); Guo and Shi (2012). In particular, for a $\mathscr{C}^\infty$ kernel (such as Gaussian kernels), the rate can be arbitrarily close to m^{-1}. To see the improvement, recall that the best convergence rates so far are given by Guo and Shi (2012). It was proved that if $f_\rho \in \mathscr{H}_{\widetilde{K}}$ and $K \in \mathscr{C}^\infty$, then with confidence $1 - \delta$,

$$\|\hat{f}_1 - f_\rho\|^2_{L^2_{\rho_X}} = \mathcal{O}\left(\left(\log(24/\delta) + \log\log_2\left(\frac{16(1-\epsilon)}{\epsilon}\right)\right)^{\frac{8}{\epsilon}-7} m^{\epsilon - \frac{1}{2}}\right).$$

We improve this result in Theorem 1, as the convergence rate (6) is always faster than $m^{-1/2}$ even for Lipschitz continuous kernels. Next, we will further illustrate the optimality of the convergence rates obtained in Theorem 1 when f_ρ belongs to some specific function space. Let $X = [0, 1]^d$, ρ_X be uniform distribution on $[0, 1]^d$ and $K \in \mathscr{C}^s(X \times X)$ with $s := s' - d/2 > 0$ being an integer. Then the RKHS $\mathscr{H}_{\widetilde{K}} \subset \mathscr{C}^s([0, 1]^d)$ and the Sobolev space $\mathscr{W}_2^{s'}([0, 1]^d)$ consists of functions belonging to the Hölder space $\mathscr{C}^{s-1,\alpha}([0, 1]^d)$ with an arbitrary $\alpha \in (0, 1)$ (see e.g., Adams and Fournier (2003)). If $f_\rho \in \mathscr{H}_{\widetilde{K}} \cap \mathscr{W}_2^{s'}([0, 1]^d)$, the claimed rate in (6) can be arbitrarily close to $\mathcal{O}(m^{-2s'/(2s'+d)})$ which is proven to be mini-max optimal in Fisher and Steinwart (2017).

Our refined result is mainly due to the following reasons. Firstly, when $K \in \mathscr{C}^s$ with $s \ge 2$ and the input space X satisfies some regularity condition, we obtain a tight upper bound on the empirical covering numbers of the related hypothesis space (see Theorem 11). Secondly, we apply the *projection operator* in the error analysis to get better estimates.

Definition 2 *For $M \ge 1$, the projection operator π_M on $\mathbb{R}$ is defined as*

$$\pi_M(t) = \begin{cases} -M & \text{if } t < -M, \\ t & \text{if } -M \le t \le M, \\ M & \text{if } t > M. \end{cases}$$

The projection of a function $f : X \to \mathbb{R}$ is given by $\pi_M(f)(x) = \pi_M(f(x)), \forall x \in X$.

The projection operator was introduced in the literature of Chen et al. (2004); Steinwart and Christmann (2008). It helps to improve the $\|\cdot\|_\infty-$bounds in the convergence analysis, which is very critical for sharp estimation.

In fact, under the uniform boundedness assumption, the performance of the algorithm (2) can be measured by the error $\|\pi_M(\hat{f}) - f_\rho\|_{L^2_{\rho_X}}$, where $\hat{f}$ is a resulting estimator. To

explain the details, we recall the definition of regression function f_ρ given by (1). As the conditional distribution $\rho(\cdot|x)$ is supported on $[-M, M]$ for every $x \in X$, the target function f_ρ takes value in $[-M, M]$ on X. So to see how an estimator $\hat{f}$ approximates f_ρ, it is natural to project values of $\hat{f}$ onto the same interval by the projection operator $\pi_M(\cdot)$. Due to the analysis in this paper, one can always expect better estimates by projecting the output estimator onto the interval $[-M, M]$. So if we consider the estimator $\pi_M(\hat{f}_1)$ in Theorem 1, the obtained result can be further improved. However, in order to make comparisons with previous results, we just give the error analysis for $\hat{f}_1$. But for the case $0 < q < 1$, we shall only consider the error $\|\pi_M(\hat{f}_q) - f_\rho\|_{L^2_{\rho_X}}$. To illustrate the sparseness of the algorithm, we also derive an upper bound on the quantity $\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m}$, where $\mathbf{c}_q^{\mathbf{z}}$ denotes the coefficient sequence of $\hat{f}_q$.

Theorem 3 *Assume that X is a compact convex subset of $\mathbb{R}^d$ with Lipschitz boundary, $K \in \mathscr{C}^\infty(X \times X)$ is an admissible kernel and $f_\rho \in \mathscr{H}_{\widetilde{K}}$ with $\widetilde{K}$ defined by (4). For $0 < q < 1$, the estimator $\hat{f}_q$ is given by algorithm (2) and satisfies Assumption 1. Let $0 < \delta < 1$ and $\gamma = m^{-\tau}$ with $1 - q < \tau < 1$. With confidence $1 - \delta$, there hold*

$$\|\pi_M(\hat{f}_q) - f_\rho\|^2_{L^2_{\rho_X}} \leq \widetilde{C} \left(\log(18/\delta) + \log\log \frac{8}{q(1-\tau)} \right)^3 m^{-(\tau-(1-q))}, \tag{7}$$

and

$$\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m} \leq \widetilde{C}' \left(q(1-q)\right)^{-q/(2-q)} \left(\log(2/\delta) + \log\log \frac{8}{q(1-\tau)} \right)^6 m^{-q(1-\frac{\tau}{2-q})}, \tag{8}$$

where $\widetilde{C}$ and $\widetilde{C}'$ are positive constants independent of m or δ.

From Theorem 3, one can see that under the restrictions on τ and q, the quantity $\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m}$ converges to 0 at a polynomial rate when the sample size m becomes large. The regularization parameter γ plays an important role as a trade-off between sparsity and convergence rates. Thus one can obtain a sparser solution at the price of lower estimation accuracy. Due to Theorem 3, when $\frac{3-\sqrt{5}}{2} < q < 1$, we may take $\tau = (2-q)(1-q)$, then the quantity $\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m}$ behaves like $\mathcal{O}(m^{-q^2})$ and the corresponding convergence rate is $\mathcal{O}(m^{-(1-q)^2})$. In our sparsity analysis (see Section 5), the regularization parameter γ also plays a role as a thresholding for the value of non-zero coefficient in $\hat{f}_q = \sum_{i=1}^m c_{q,i}^{\mathbf{z}} K_{x_i}$. Due to our analysis, a lower bound for non-zero coefficients is given by $O(\gamma^{2/(2-q)})$, which implies that a small q will lead to more zero coefficients in the kernel expansion for a fixed $\gamma < 1$. It should be mentioned that our sparsity analysis is only valid for $0 < q < 1$.

For the RKHS-based regularization algorithms, it is well known that a classical way to obtain sparsity is to introduce the $\epsilon-$insensitive zone in the loss function. A theoretical result for this approach in Steinwart and Christmann (2009) shows that the fraction of non-zero coefficients in the kernel expansion is asymptotically lower and upper bounded by constants. From this point of view, regularizing the combinatorial coefficients by the ℓ_q-norm is a more powerful way to produce sparse solutions. As our theoretical analysis only gives results for the worst case situations, one can expect better performance of the ℓ_q-regularized kernel regression in practice.

At the end of this section, we point out that the mathematical analysis for the case $0 < q < 1$ is far from optimal. It is mainly because of our analysis is based on Assumption 1. Under this assumption, one needs to bound $\|\mathbf{c}_1^{\mathbf{z}}\|_q^q$ which is critical in the error analysis, where $\mathbf{c}_1^{\mathbf{z}}$ denotes the coefficient sequence of the estimator $\hat{f}_1$. In fact, due to the discussion in Section 2.1, we can construct a minimizing sequence from any point. Besides the solution of ℓ_1-regularized kernel regression, we may consider some other choices of the starting point, e.g, the solution of RKHS-based regularized least-square regression. We believe that how to bound the $\|\cdot\|_q-$norm of these initial vectors is still a problem when one considers other possible starting points. In this paper, we use the reverse Hölder inequality to handle this term and the bound is too loose especially when q is small. Actually, even if $\hat{f}_q$ is a global minimizer of problem (2), we can not give an effective approach to conduct the error analysis. Additionally, we do not assume any sparsity condition on the target function. One possible condition that one may consider is that the regression function belongs to the closure of the linear span of $\{K_x | x \in X\}$ under the ℓ_q-constraint. Compared with the *hard sparsity* introduced by the ℓ_0-norm, such kind of sparsity assumption is referred as to *soft sparsity* (see Raskutti et al. (2011)), which is based on imposing a certain decay rate on the entries of the coefficient sequence. Developing the corresponding mathematical analysis under the soft sparsity assumption will help us to understand the role of the ℓ_q-regularization in feature selections in an infinite-dimensional hypothesis space. We shall consider this topic in future work. However, the sparsity analysis in this paper is still valid to derive the asymptotical bound for $\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m}$ and will lead to better estimates if a more elaborate error analysis can be given.

The paper is organized as follows. The next section presents a descent-iterative minimization process for algorithm (2) and establishes the framework of error analysis. In Section 3, we derive a tight bound on empirical covering numbers of the hypothesis space under the ℓ_1-constraint. In Section 4 and Section 5, we derive the related results on the error analysis and sparseness of ℓ_q-regularized kernel regression.

2. Preliminaries

This section is devoted to generating the minimizing sequences and establishing the framework of mathematical analysis for ℓ_q-regularized kernel regression.

2.1. Minimizing sequences for ℓ_q-regularized kernel regression

In this part, we present a descent-iterative minimization process for algorithm (2), which can probably search a local minimizer starting from any initial point. Motivated by recent work on $\ell_{1/2}-$regularization in Xu et al. (2012), we generalize their strategy to the case $0 < q < 1$.

Let $\mathrm{sgn}(x)$ be given by $\mathrm{sgn}(x) = 1$ for $x \geq 0$ and $\mathrm{sgn}(x) = -1$ for $x < 0$. Define a function $\psi_{\eta,q}$ for $\eta > 0$ and $0 < q < 1$ as

$$\psi_{\eta,q}(x) = \begin{cases} \mathrm{sgn}(x) t_{\eta,q}(|x|), & |x| > a_q \eta^{1/(2-q)}, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $a_q = (1-\frac{q}{2})(1-q)^{\frac{q-1}{2-q}}$ and $t_{\eta,q}(|x|)$ denotes the solution of the equation

$$2t + \eta q t^{q-1} - 2|x| = 0 \tag{10}$$

on the interval $[(q(1-q)\eta/2)^{1/(2-q)}, \infty)$ with respect to the variable t. We further define a map $\Psi_{\eta,q} : \mathbb{R}^m \to \mathbb{R}^m$, which is given by

$$\Psi_{\eta,q}(\mathbf{d}) = (\psi_{\eta,q}(d_1), \cdots, \psi_{\eta,q}(d_m)), \qquad \forall \mathbf{d} = (d_1, \cdots, d_m) \in \mathbb{R}^m. \tag{11}$$

Then we have the following important lemma.

Lemma 4 *For any $\eta > 0$, $0 < q < 1$ and $\mathbf{d} = (d_1, \cdots, d_m) \in \mathbb{R}^m$, the map $\Psi_{\eta,q} : \mathbb{R}^m \to \mathbb{R}^m$ given by (11) is well-defined and $\Psi_{\eta,q}(\mathbf{d})$ is a global minimizer of the problem*

$$\min_{\mathbf{c}\in\mathbb{R}^m} \left\{ \|\mathbf{c} - \mathbf{d}\|_2^2 + \eta \|\mathbf{c}\|_q^q \right\}. \tag{12}$$

We shall leave the proof to the Appendix. The function $\psi_{\eta,q}$ defines the ℓ_q-*thresholding function* for $0 < q < 1$. According to the proof of Lemma 4, given a $x \in \mathbb{R}$ and $\eta > 0$, the value of $\psi_{\eta,q}(x)$ in equation (9) is essentially a global minimizer of the problem

$$\min_{t\in\mathbb{R}} \left\{ |t - x|^2 + \eta |t|^q \right\}.$$

When $q = 1/2$, the function $\psi_{\eta,1/2}$ is exactly the half thresholding function obtained in Xu et al. (2012). We also observe that though the analysis in Lemma 4 is based on the fact $0 < q < 1$, the expression of $\psi_{\eta,q}$ is coherent for $q \in [0, 1]$. Concretely, as $\lim_{q\to 1-} a_q = \frac{1}{2}$, by letting $q \to 1-$ in the definition of $\psi_{\eta,q}$, one may obtain the soft thresholding function for ℓ_1-regularization, which is given by (e.g., see Daubechies et al. (2004))

$$\psi_{\eta,1}(x) = \begin{cases} x - \mathrm{sgn}(x)\eta/2, & |x| > \eta/2, \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, if taking $q = 0$ in the expression of $\psi_{\eta,q}$, one may also derive the hard thresholding function for ℓ_0-regularization, which is defined as (e.g., see Blumensath and Davies (2008))

$$\psi_{\eta,0}(x) = \begin{cases} x, & |x| > \eta^{1/2}, \\ 0, & \text{otherwise.} \end{cases}$$

The expression of $\psi_{\eta,q}$ is very useful as we can establish a descent-iterative minimization process for algorithm (2) based on the idea in Daubechies et al. (2004). Recall the definitions of $\mathbb{K}$ and $\mathbf{y}$ in the objective functional $\mathscr{T}_{\gamma,q}$ given by (3). For any $(\lambda, \gamma) \in (0, \infty)^2$ and $\mathbf{c}_0 \in \mathbb{R}^m$, we iteratively define a sequence $\{\mathbf{c}_n\}_{n=1}^\infty$ as

$$\mathbf{c}_{n+1} = \Psi_{\lambda\gamma,q}\left(\mathbf{c}_n + \lambda \mathbb{K}^T(\mathbf{y} - \mathbb{K}\mathbf{c}_n)\right). \tag{13}$$

Then $\{\mathbf{c}_n\}_{n=1}^\infty$ is a minimizing sequence with a suitable chosen $\lambda > 0$.

Proposition 5 *Let $0 < q < 1$, $\gamma > 0$ and $0 < \lambda \le \frac{q}{2}\|\mathbb{K}\|_2^{-2}$, where $\|\cdot\|_2$ denotes the spectral norm of the matrix. If the sequence $\{\mathbf{c}_n\}_{n=0}^\infty$ is generated by the iteration process (13), then the following statements are true.*

(i) *If $\mathbf{c}^* \in \mathbb{R}^m$ is a local minimizer of the objective functional $\mathscr{T}_{\gamma,q}(\mathbf{c})$, then c^* is a stationary point of the iteration process (13), i.e.,*

$$\mathbf{c}^* = \Psi_{\lambda\gamma,q}\left(\mathbf{c}^* + \lambda \mathbb{K}^T(\mathbf{y} - \mathbb{K}\mathbf{c}^*)\right).$$

(ii) *The sequence $\{\mathbf{c}_n\}_{n=0}^\infty$ is a minimizing sequence such that the sequence $\{\mathscr{T}_{\gamma,q}(\mathbf{c}_n)\}_{n=0}^\infty$ is monotonically decreasing.*

(iii) *The sequence $\{\mathbf{c}_n\}_{n=0}^\infty$ converges to a stationary point of the iteration process (13) whenever λ is sufficiently small.*

We also prove this proposition in the Appendix. The properties of $\psi_{\eta,q}$ play an important role in the proof. When $q = 1/2$ and $q = 2/3$, the equation $2t + \eta q t^{q-1} - 2|x| = 0$ can be analytically solved, i.e., the corresponding thresholding function can be explicit expressed as an analytical function. This motivated people to develop efficient algorithms based on (13) for these two special cases. In particular, the $\ell_{1/2}-$regularization problem has been intensively studied in some literature (see Xu et al. (2012) and references therein). Since a general formula for ℓ_q-thresholding function is given by (9), it is also interesting to develop corresponding iterative algorithms and compare their empirical performances for different values of q.

2.2. Framework of error analysis

In this subsection, we establish the framework of convergence analysis. Because of the least-square nature, one can see Cucker and Zhou (2007) that

$$\|f - f_\rho\|_{L^2_{\rho_X}}^2 = \mathscr{E}(f) - \mathscr{E}(f_\rho), \qquad \forall f : X \to \mathbb{R},$$

where $\mathscr{E}(f) = \int_{X\times Y}(f(x) - y)^2 d\rho$. Let $\hat{f}$ be the estimator produced by algorithm (2). Particularly, the estimator $\hat{f}$ under our consideration is $\hat{f}_1$ or $\pi_M(\hat{f}_q)$ with $0 < q \le 1$. To estimate $\|\hat{f} - f_\rho\|_{L^2_{\rho_X}}^2$, we only need to bound $\mathscr{E}(\hat{f}) - \mathscr{E}(f_\rho)$. This will be done by applying the *error decomposition* approach which has been developed in the literature for regularization schemes (e.g., see Cucker and Zhou (2007); Steinwart and Christmann (2008)). In this paper, we establish the error decomposition formula based on the first author's previous work (Guo and Shi (2012)).

To this end, we still need to introduce some notations. For any continuous function $K : X \times X \to \mathbb{R}$, define an integral operator on $L^2_{\rho_X}(X)$ as

$$L_K f(x) = \int_X K(x, t) f(t) d\rho_X(t), \quad x \in X.$$

Since X is compact and K is continuous, L_K and its adjoint L_K^* are both compact operators. If K is a Mercer kernel, the corresponding integral operator L_K is a self-adjoint positive operator on $L^2_{\rho_X}$, and its $r-$th power L_K^r is well-defined for any $r > 0$. From Cucker and Zhou (2007), we know that the RKHS $\mathscr{H}_K$ is in the range of $L_K^{\frac{1}{2}}$. Recalling the Mercer kernel $\widetilde{K}$ defined as (4) for a continuous kernel K, it is easy to check that $L_{\widetilde{K}} = L_K L_K^*$.

Following the same idea in Guo and Shi (2012), we use the RKHS $\mathscr{H}_{\widetilde{K}}$ with the norm denoted by $\|\cdot\|_{\widetilde{K}}$ to approximate f_ρ. The approximation behavior is characterized by the *regularization error*

$$\mathcal{D}(\gamma) = \min_{f\in\mathscr{H}_{\widetilde{K}}} \left\{ \mathscr{E}(f) - \mathscr{E}(f_\rho) + \gamma\|f\|_{\widetilde{K}}^2 \right\}.$$

The following assumption is standard in the literature of learning theory (e.g., see Cucker and Zhou (2007); Steinwart and Christmann (2008)).

Assumption 2 *For some $0 < \beta \le 1$ and $c_\beta > 0$, the regularization error satisfies*

$$\mathcal{D}(\gamma) \le c_\beta \gamma^\beta, \qquad \forall \gamma > 0. \tag{14}$$

The decay of $\mathcal{D}(\gamma)$ as $\gamma \to 0$ measures the approximation ability of the function space $\mathscr{H}_{\widetilde{K}}$. Next, for $\gamma > 0$, we define the *regularizing function* as

$$f_\gamma = \arg\min_{f\in\mathcal{H}_{\widetilde{K}}} \left\{ \mathscr{E}(f) - \mathscr{E}(f_\rho) + \gamma\|f\|_{\widetilde{K}}^2 \right\}. \tag{15}$$

The regularizing function uniquely exists and is given by $f_\gamma = \left(\gamma I + L_{\widetilde{K}}\right)^{-1} L_{\widetilde{K}} f_\rho$ (e.g., see Proposition 8.6 in Cucker and Zhou (2007)), where I denotes the identity operator on $\mathscr{H}_{\widetilde{K}}$.

Now we are in a position to establish the error decomposition for algorithm (2). Recall that for $0 < q \le 1$, $\mathbf{c}_q^{\mathbf{z}}$ denotes the coefficient sequence of the estimator $\hat{f}_q$ and $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m \in Z^m$ is the sample set. The empirical loss functional $\mathcal{E}_{\mathbf{z}}(f)$ is defined for $f : X \to \mathbb{R}$ as

$$\mathcal{E}_{\mathbf{z}}(f) = \frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2.$$

Proposition 6 *For $\gamma > 0$, the regularization function f_γ is given by (15) and $f_{\mathbf{z},\gamma} = \frac{1}{m}\sum_{i=1}^m K_{x_i} g_\gamma(x_i)$ with*

$$g_\gamma = L_K^* \left(\gamma I + L_{\widetilde{K}}\right)^{-1} f_\rho.$$

Let $\hat{f}$ be an estimator under consideration, we define

$$\begin{aligned}
\mathscr{S}_1 &= \{\mathscr{E}(\hat{f}) - \mathscr{E}_{\mathbf{z}}(\hat{f})\} + \{\mathscr{E}_{\mathbf{z}}(f_{\mathbf{z},\gamma}) - \mathscr{E}(f_{\mathbf{z},\gamma})\},\\
\mathscr{S}_2 &= \{\mathscr{E}(f_{\mathbf{z},\gamma}) - \mathscr{E}(f_\gamma)\} + \gamma\{\frac{1}{m}\sum_{i=1}^m |g_\gamma(x_i)| - \|g_\gamma\|_{L^1_{\rho_X}}\},\\
\mathscr{S}_3 &= \mathscr{E}(f_\gamma) - \mathscr{E}(f_\rho) + \gamma\|g_\gamma\|_{L^2_{\rho_X}}.
\end{aligned}$$

If $\hat{f}_q$ satisfies Assumption 1 with $0 < q < 1$, for the estimator $\hat{f} = \pi_M(\hat{f}_q)$, there holds

$$\mathscr{E}(\hat{f}) - \mathscr{E}(f_\rho) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \le \mathscr{S}_1 + \mathscr{S}_2 + \mathscr{S}_3 + \gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q. \tag{16}$$

When $q = 1$, for $\hat{f} = \hat{f}_1$ or $\pi_M(\hat{f}_1)$, there holds

$$\mathscr{E}(\hat{f}) - \mathscr{E}(f_\rho) + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_1 \le \mathscr{S}_1 + \mathscr{S}_2 + \mathscr{S}_3. \tag{17}$$

To save space, we shall leave the proof to the Appendix. In fact, Proposition 6 presents three error decomposition formulas. When $\hat{f} = \hat{f}_1$, the inequality (17) is exactly the error decomposition introduced in Guo and Shi (2012) for ℓ_1-regularized kernel regression. Note that the bound (16) involves an additional term $\gamma m^{1-q}\|\hat{\mathbf{c}}_1\|_1^q$. Therefore, the asymptotic behavior of the global estimator $\hat{f}_1$ plays a significant part in the convergence analysis of $\hat{f}_q$ with $0 < q < 1$.

With the help of Proposition 6, one can estimate the total error by bounding $\mathscr{S}_i$ $(i = 1, 2, 3)$ and $\gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q$ respectively. The terms $\mathscr{S}_2$ and $\mathscr{S}_3$ are well estimated in Guo and Shi (2012) by fully utilizing the structure of the functions $f_{\mathbf{z},\gamma}$ and g_γ. Here we directly quote the following bound for $\mathscr{S}_2 + \mathscr{S}_3$. One may see Guo and Shi (2012) for the detailed proof.

Lemma 7 *For any $(\gamma, \delta) \in (0, 1)^2$, with confidence $1 - \delta$, there holds*

$$\begin{aligned}
\mathscr{S}_2 + \mathscr{S}_3 \le{}& 8\kappa^2\left(2\kappa^2 + 1\right)\log^2(4/\delta)\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2 m^2} + \frac{\mathcal{D}(\gamma)}{\gamma m}\right\}\\
&+ \frac{(2\kappa+1)\sqrt{\mathcal{D}(\gamma)}\log(4/\delta)}{m} + \frac{3}{2}\sqrt{\gamma\mathcal{D}(\gamma)} + 2\mathcal{D}(\gamma),
\end{aligned} \tag{18}$$

where $\kappa = \|K\|_{\mathscr{C}(X\times X)}$.

Therefore, our error analysis mainly focuses on bounding $\mathscr{S}_1$ and $\gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q$. The first term $\mathscr{S}_1$ can be estimated by uniform concentration equalities. These inequalities quantitatively characterize the convergence behavior of the empirical processes over a function set by various capacity measures, such as VC dimension, covering number, entropy integral and so on. For more details, one may refer to Van der Vaart and Wellner (1996) and references therein. In this paper, we apply the concentration technique involving the ℓ_2-*empirical covering numbers* to obtain bounds on $\mathscr{S}_1$. The ℓ_2-empirical covering number is defined by means of the normalized ℓ_2-metric d_2 on the Euclidian space $\mathbb{R}^l$ given by

$$d_2(\mathbf{a}, \mathbf{b}) = \left(\frac{1}{l}\sum_{i=1}^l |a_i - b_i|^2\right)^{1/2}, \qquad \mathbf{a} = (a_i)_{i=1}^l, \mathbf{b} = (b_i)_{i=1}^l \in \mathbb{R}^l.$$

Definition 8 *Let $(\mathcal{M}, d_{\mathcal{M}})$ be a pseudo-metric space and $S \subset \mathcal{M}$ a subset. For every $\epsilon > 0$, the covering number $\mathscr{N}(S, \epsilon, d_{\mathcal{M}})$ of S with respect to ϵ and the pseudo-metric $d_{\mathcal{M}}$ is defined to be the minimal number of balls of radius ϵ whose union covers S, that is,*

$$\mathscr{N}(S, \epsilon, d) = \min\left\{\iota \in \mathbb{N} : \ S \subset \bigcup_{j=1}^{\iota} B(s_j, \epsilon) \text{ for some } \{s_j\}_{j=1}^{\iota} \subset \mathcal{M}\right\},$$

where $B(s_j, \epsilon) = \{s \in \mathcal{M} : d_{\mathcal{M}}(s, s_j) \le \epsilon\}$ is a ball in $\mathcal{M}$. For a set $\mathscr{F}$ of functions on X and $\epsilon > 0$, the ℓ_2-empirical covering number of $\mathscr{F}$ is given by

$$\mathscr{N}_2(\mathscr{F}, \epsilon) = \sup_{l\in\mathbb{N}} \sup_{\mathbf{u}\in X^l} \mathscr{N}(\mathscr{F}|_{\mathbf{u}}, \epsilon, d_2),$$

where for $l \in \mathbb{N}$ and $\mathbf{u} = (u_i)_{i=1}^l \subset X^l$, we denote the covering number of the subset $\mathscr{F}|_{\mathbf{u}} = \{(f(u_i))_{i=1}^l : f \in \mathscr{F}\}$ of the metric space $(\mathbb{R}^l, d_2)$ as $\mathscr{N}(\mathscr{F}|_{\mathbf{u}}, \varepsilon, d_2)$.

As for the last term, we will derive a tighter bound for $\|\mathbf{c}_1^{\mathbf{z}}\|_1$ by an iteration technique based on the convergence analysis for the estimator $\pi_M(\hat{f}_1)$. The application of the projection operator will lead to better estimates.

3. Capacity of the hypothesis space under ℓ_1-constraint

In this section, by means of the ℓ_2-empirical covering numbers, we study the capacity of the hypothesis space generated by the kernel function. For $R > 0$, define $\mathscr{B}_R$ to be the linear combination of functions $\{K_x | x \in X\}$ under the ℓ_1-constraint

$$\mathscr{B}_R = \left\{ \sum_{i=1}^{k} \mu_i K_{u_i} : k \in \mathbb{N}, u_i \in X, \mu_i \in \mathbb{R} \text{ and } \sum_{i=1}^{k} |\mu_i| \leq R \right\}. \tag{19}$$

In this paper, we assume that the following capacity assumption for $\mathscr{B}_1$ comes into existence.

Assumption 3 *For a kernel function K, there exists an exponent p with $0 < p < 2$ and a constant $c_{K,p} > 0$ such that*

$$\log_2 \mathscr{N}_2(\mathscr{B}_1, \epsilon) \leq c_{K,p} \left(\frac{1}{\epsilon}\right)^p, \qquad \forall 0 < \epsilon \leq 1. \tag{20}$$

It is strictly proved in Shi et al. (2011) that, for a compact subset X of $\mathbb{R}^d$ and $K \in \mathscr{C}^s$ with some $s > 0$, the power index p can be given by

$$p = \begin{cases} 2d/(d+2s), & \text{when } 0 < s \leq 1, \\ 2d/(d+2), & \text{when } 1 < s \leq 1 + d/2, \\ d/s, & \text{when } s > 1 + d/2. \end{cases} \tag{21}$$

We will present a much tighter bound on the logarithmic ℓ_2-empirical covering numbers of $\mathscr{B}_1$. This bound holds for a general class of input space satisfying an *interior cone condition*.

Definition 9 *A subset X of $\mathbb{R}^d$ is said to satisfy an interior cone condition if there exist an angle $\theta \in (0, \pi/2)$, a radius $R_X > 0$, and a unit vector $\xi(x)$ for every $x \in X$ such that the cone*

$$C(x, \xi(x), \theta, R_X) = \left\{ x + ty : y \in \mathbb{R}^d, |y| = 1, y^T \xi(x) \geq \cos\theta, 0 \leq t \leq R_X \right\}$$

is contained in X.

Remark 10 *The interior cone condition excludes those sets X with cusps. It is valid for any convex subset of $\mathbb{R}^d$ with Lipschitz boundary (see e.g., Adams and Fournier (2003)).*

Now we are in a position to give our refined result on the capacity of $\mathscr{B}_1$.

Theorem 11 *Let X be a compact subset of $\mathbb{R}^d$. Suppose that X satisfies an interior cone condition and $K \in \mathscr{C}^s(X \times X)$ with $s \geq 2$ is an admissible kernel. Then there exists a constant $C_{X,K}$ that depends on X and K only, such that*

$$\log_2 \mathscr{N}_2(\mathscr{B}_1, \epsilon) \leq C_{X,K} \epsilon^{-\frac{2d}{d+2\lfloor s \rfloor}} \log_2 \left(\frac{2}{\epsilon}\right), \qquad \forall\, 0 < \epsilon \leq 1, \tag{22}$$

where $\lfloor s \rfloor$ denotes the integral part of s.

Recall the asymptotical bound obtained in Shi et al. (2011) with p given by (21). It asserts that for $K \in \mathscr{C}^s$ with $s \geq 2$, the quantity $\log_2 \mathscr{N}_2(\mathscr{B}_1, \epsilon)$ grows at most of the order $\epsilon^{-\min\left\{\frac{2d}{d+2}, \frac{d}{s}\right\}}$. Our stated result in Theorem 11 improves the previous bound a lot, as $\log_2 \mathscr{N}_2(\mathscr{B}_1, \epsilon)$ can be bounded by $\mathcal{O}(\epsilon^{-s_1})$ for any $s_1 > \frac{2d}{d+2\lfloor s \rfloor}$. Besides the ℓ_2-empirical covering number, another way to measure the capacity is the uniform covering number $\mathscr{N}(\mathscr{B}_1, \epsilon, \|\cdot\|_\infty)$ of $\mathscr{B}_1$ as a subset of the metric space $(\mathscr{C}(X), \|\cdot\|_\infty)$ of bounded continuous functions on X. From a classical result in function spaces (see Edmunds and Triebel (1996)), for $X = [0,1]^d$ and $K \in \mathscr{C}^s(X \times X)$, the unit ball $\mathscr{B}_1$ satisfies

$$c_s \left(\frac{1}{\epsilon}\right)^{d/s} \leq \log \mathscr{N}(\mathscr{B}_1, \epsilon, \|\cdot\|_\infty) \leq c_s' \left(\frac{1}{\epsilon}\right)^{d/s}, \qquad \forall \epsilon > 0.$$

When d is large or s is small, this estimate is rough. Moreover, the estimate above is asymptotic optimal and cannot be improved, which implies that the uniform covering number is not a suitable measurement for the capacity of $\mathscr{B}_1$. The ℓ_2-emprical covering number was first investigated in the field of empirical process (e.g., see Dudley (1987); Van der Vaart and Wellner (1996) and references therein). One usually assumes that, there exist $0 < p < 2$ and $c_p > 0$ such that

$$\log \mathscr{N}_2(\mathscr{F}, \epsilon) \leq c_p \epsilon^{-p}, \qquad \forall \epsilon > 0, \tag{23}$$

which guarantees the convergence of Dudley's entropy integral, i.e., $\int_0^1 \sqrt{\log \mathscr{N}_2(\mathscr{F}, \epsilon)} d\epsilon < \infty$. This fact is very important, since function class $\mathscr{F}$ with bounded entropy integrals satisfy the uniform central limit Theorem (see Dudley (1987) for more details). The classical result on ℓ_2-empirical covering number only asserts that if $\mathscr{N}_2(\mathscr{F}, \epsilon) = \mathcal{O}(\epsilon^{-p'})$ for some $p' > 0$, then the convex hull of $\mathscr{F}$ satisfies (23) with $p = \frac{2p'}{p'+2} < 2$. In this paper, we further clarify the relation between the smoothness of the kernel and the capacity of the hypothesis space. That is, we establish a more elaborate estimate for the power index p in Assumption 3 by using the prior information on the smoothness of the kernel. It should be pointed out that, from the relation $\mathscr{N}_2(\mathscr{F}, \epsilon) \leq \mathscr{N}(\mathscr{F}, \epsilon, \|\cdot\|_\infty)$, capacity assumption (23) for an RKHS generated by a positive semi-definite kernel K can be verified by bounding the uniform covering number. It is proved in Zhou (2003) that, when $\mathscr{F}$ is taken to be the unit ball in the RKHS $\mathscr{H}_K$, there holds $\log \mathscr{N}(\mathscr{F}, \epsilon, \|\cdot\|_\infty) = \mathcal{O}(\epsilon^{-2d/s})$ provided that $X \subset \mathbb{R}^d$ and $K \in \mathscr{C}^s(X \times X)$. Therefore, a sufficiently smooth kernel with $s > d$ can guarantee that capacity assumption (23) comes into existence. When X is a Euclidean ball in $\mathbb{R}^d$, the Sobolev space $\mathscr{W}_2^s(X)$ with $s > d/2$ is an RKHS. Birman and Solomyak (1967) proved that $\log \mathscr{N}_2(\mathscr{W}_2^s(X), \epsilon)$ is upper and lower bounded by $\mathcal{O}(\epsilon^{-p})$ with $p = d/s < 2$. However, up to our best knowledge, how to demonstrate capacity assumption (23) for a general RKHS is still widely open. From Theorem 11, one can see that even for positive semi-definite kernels,

the function space (19) is more suitable to be the hypothesis space than the classical RKHS, as its capacity can be well-estimated by the ℓ_2-empirical covering number.

We will prove Theorem 11 after a few lemmas. The improvement is mainly due to a local polynomial reproduction formula from the literature of multivariate approximation (see Wendland (2001); Jetter et al. (1999)). A point set $\Omega = \{\omega_1, \cdots, \omega_n\} \subset X$ is said to be Δ-*dense* if

$$\sup_{x\in X} \min_{\omega_j \in \Omega} |x - \omega_j| \leq \Delta,$$

where $|\cdot|$ is the standard Euclidean norm on $\mathbb{R}^d$. Denote the space of polynomials of degree at most s on $\mathbb{R}^d$ as $\mathcal{P}_s^d$. The following lemma is a formula for local polynomial reproduction (see Theorem 3.10 in Wendland (2001)).

Lemma 12 *Suppose X satisfies an interior cone condition with radius $R_X > 0$ and angle $\theta \in (0, \pi/2)$. Fix $s \in \mathbb{N}$ with $s \geq 2$. There exists a constant c_0 depending on θ, d and s such that for any Δ-dense point set $\Omega = \{\omega_1, \cdots, \omega_n\}$ in X with $\Delta \leq \frac{R_X}{c_0}$ and every $u \in X$, we can find real numbers $b_i(u)$, $1 \leq i \leq n$, satisfying*

(1) $\sum_{i=1}^n b_i(u)p(\omega_i) = p(u) \qquad \forall p \in \mathcal{P}_s^d$,

(2) $\sum_{i=1}^n |b_i(u)| \leq 2$,

(3) $b_i(u) = 0$ *provided that* $|u - \omega_i| > c_0\Delta$.

This formula was first introduced by Wang et al. (2012) to learning theory for investigating the approximation property of the kernel-based hypothesis space. For any function set $\mathscr{F}$ on X, we use $absconv\mathscr{F}$ to denote the *absolutely convex hull* of $\mathscr{F}$, which is given by

$$absconv\mathscr{F} = \left\{ \sum_{i=1}^k \lambda_i f_i : k \in \mathbb{N}, f_i \in \mathscr{F} \text{ and } \sum_{i=1}^k |\lambda_i| \leq 1 \right\}.$$

In order to prove our result, we also need the following lemma.

Lemma 13 *Let Q be a probability measure on X and $\mathscr{F}$ be a class of n measurable functions of finite L_Q^2-diameter $diam\mathscr{F}$. Then for every $\epsilon > 0$,*

$$\mathscr{N}\left(absconv\mathscr{F}, ediam\mathscr{F}, d_{2,Q}\right) \leq \left(e + e(2n+1)\epsilon^2\right)^{2/\epsilon^2}, \tag{24}$$

where $d_{2,Q}$ is the metric induced by the norm $\|\cdot\|_{L_Q^2}$.

This lemma can be proved following the same idea of Lemma 2.6.11 in Van der Vaart and Wellner (1996). Now we can concentrate our efforts on deriving our conclusion in Theorem 11.

Proof [Proof of Theorem 11]. A set is called $\Delta-$*separated* if the distance between any two elements of the set is larger than Δ. We take $\{\Delta_n\}_{n=1}^\infty$ to be a positive sequence decreasing to 0 with Δ_n explicitly given later. Let the set $X_n = \{v_1, v_2, \cdots, v_{|X_n|}\}$ be an increasing family of finite sets, where $|X_n|$ denotes the cardinality of X_n. For every n, X_n is a maximal Δ_n-separated set in X with respect to inclusion, i.e., each X_n is

Δ_n-separated and if $X_n \subset W \subset X$ then W is not Δ_n-separated. Note that, if X_n is a maximal Δ_n-separated set in X, then it is Δ_n-dense in X. Based on $\{X_n\}_{n=1}^\infty$, we create a family of sets $\mathscr{A}_n = \{K_v | v \in X_n\}$. Similarly, let $\mathscr{A} = \{K_v | v \in X\}$, then $\mathscr{A}_1 \subset \mathscr{A}_2 \subset \cdots \subset \mathscr{A}$.

We first limit our discussion to the case that $s \geq 2$ is an integer. Motivated by the proof of Proposition 1 in Wang et al. (2012), we will show that for any $t_0 \in X$, the function K_{t_0} can be approximated by a linear combination of $\mathscr{A}_n$ whenever Δ_n is sufficiently small. For given $x \in X$, we let $g_x(t) = K(x,t)$. Then $g_x \in \mathscr{C}^s(X)$. We consider the Taylor expansion of g_x at a fixed point t_0 of degree less than s, which is given by

$$P_x(t) = K(x, t_0) + \sum_{|\alpha|=1}^{s-1} \frac{D^\alpha g_x(t_0)}{\alpha!}(t - t_0)^\alpha.$$

Here $\alpha = (\alpha^1, \cdots, \alpha^d) \in \mathbb{Z}_+^d$ is a multi-index with $|\alpha| = \sum_{j=1}^d |\alpha^j|$, $\alpha! = \prod_{j=1}^d (\alpha^j)!$, and $D^\alpha g_x(t_0)$ denotes the partial derivatives of g_x at the point t_0. Due to Lemma 12, if X satisfies an interior cone condition, we take a constant $\Delta_{X,s} := \frac{R_X}{c_0}$ depending only on X and s. Then for any $\Delta-$dense set given by $\{\omega_1, \cdots, \omega_n\}$ with $\Delta \leq \Delta_{X,s}$, we have

$$P_x(t_0) = \sum_{i \in I(t_0)} b_i(t_0) P_x(\omega_i),$$

where $\sum_{i\in I(t_0)} |b_i(t_0)| \leq 2$ and $I(t_0) = \{i \in \{1, \cdots, n\} : |\omega_i - t_0| \leq c_0\Delta\}$. Note that $K(x, t_0) = P_x(t_0)$ and $\max_{i\in I(t_0)} |K(x, \omega_i) - P_x(\omega_i)| \leq c_0^s \|K\|_{\mathscr{C}^s} \Delta^s$. It follows that

$$\begin{aligned} \left| K(x,t_0) - \sum_{i\in I(t_0)} b_i(t_0) K(x,\omega_i) \right| &= \left| \sum_{i\in I(t_0)} b_i(t_0) \left(P_x(\omega_i) - K(x,\omega_i)\right) \right| \\ &\leq 2c_0^s \|K\|_{\mathscr{C}^s} \Delta^s. \end{aligned} \tag{25}$$

It is important that the right hand side of the above inequality is independent of x or t_0.

For any function f belonging to $\mathscr{B}_1$, there exist $k \in \mathbb{N}$ and $\{u_i\}_{i=1}^k \subset X$, such that $f(x) = \sum_{i=1}^k \mu_i K(x, u_i)$ with $\sum_{i=1}^k |\mu_i| \leq 1$. Recall that $X_n = \{v_1, v_2, \cdots, v_{|X_n|}\}$ is Δ_n-dense in X. If $\Delta_n \leq \Delta_{X,s}$, we set

$$f_n(x) = \sum_{i=1}^k \mu_i \sum_{j\in I_n(u_i)} b_j(u_i) K(x, v_j),$$

where the index set $I_n(u)$ is defined for any $n \in \mathbb{N}$ and $u \in X$ as

$$I_n(u) = \{j \in \{1, \cdots, |X_n|\} : |v_j - u| \leq c_0\Delta_n\}.$$

Hence, we have

$$\|f - f_n\|_\infty = \sup_{x\in X} \left| \sum_{i=1}^k \mu_i \left(K(x,u_i) - \sum_{j\in I_n(u_i)} b_j(u_i) K(x,v_j) \right) \right| \leq 2c_0^s \|K\|_{\mathscr{C}^s} \Delta_n^s, \tag{26}$$

where the last inequality is from (25). Obviously, we can rewrite f_n as $f_n(x) = \sum_{j=1}^{|X_n|} \nu_j K(x, v_j)$ where

$$\sum_{j=1}^{|X_n|} |\nu_j| \le \sum_{i=1}^{k} |\mu_i| \sum_{j \in I_n(u_i)} |b_j(u_i)| \le 2.$$

Therefore, we have $f_n \in 2absconv\mathscr{A}_n$.

For a compact subset X of $\mathbb{R}^d$, there exists a constant $c_X > 0$ depending only on X, such that

$$\mathscr{N}(X, \epsilon, \tilde{d}_2) \le c_X \left(\frac{1}{\epsilon}\right)^d, \qquad \forall\, 0 < \epsilon \le 1,$$

where $\tilde{d}_2$ denotes the metric induced by the standard Euclidean norm on $\mathbb{R}^d$ (see Theorem 5.3 in Cucker and Zhou (2007)). Now we let $\Delta_n = 2c_X^{\frac{1}{d}} n^{-\frac{1}{d}}$. Recall that $|X_n|$ is the maximal cardinality of an Δ_n-separated set in X. Thus we have $|X_n| \le \mathscr{N}(X, \Delta_n/2, \tilde{d}_2) \le n$.

For any given $\epsilon > 0$, we choose $N := N(\epsilon)$ be to the smallest integer which is larger than $C'_{X,K}\left(\frac{1}{\epsilon}\right)^{\frac{d}{s}}$ with $C'_{X,K} = 2^{(s+2)d/s} c_0^d \|K\|_{\mathscr{C}^s}^{d/s} c_X$. Then the set $2absconv\mathscr{A}_N$ is $\frac{\epsilon}{2}-$dense in $\mathscr{B}_1$ due to (26). Recall that for any probability measure Q on X, $d_{2,Q}$ denotes the metric induced by the norm $\|\cdot\|_{L_Q^2}$. Therefore, we have

$$\mathscr{N}(\mathscr{B}_1, \epsilon, d_{2,Q}) \le \mathscr{N}(2absconv\mathscr{A}_N, \epsilon/2, d_{2,Q}) = \mathscr{N}(absconv\mathscr{A}_N, \epsilon/4, d_{2,Q}).$$

To bound the latter, let $N_1 := N_1(\epsilon)$ be the integer part of $\epsilon^{-\frac{2d}{d+2s}}$. We choose a sufficiently small $\epsilon > 0$ satisfying

$$0 < \epsilon \le \min\left\{(C'_{X,K})^{s(d+2s)/d^2}, \left(2^{-1} c_X^{-1/d} \Delta_{X,s}\right)^{s+\frac{d}{2}}, \left(\frac{1}{2}\right)^{(d+2s)/2d}\right\} := \epsilon_{X,K}, \tag{27}$$

then $X_{N_1} \subset X_N$ and $\Delta_{N_1} \le \Delta_{X,s}$.

For any $g \in absconv\mathscr{A}_N$, there exists $\{\nu_i\}_{i=1}^{|X_N|} \subset \mathbb{R}^{|X_N|}$ with $\sum_{i=1}^{|X_N|} |\nu_i| \le 1$, such that

$$g(x) = \sum_{i=1}^{|X_N|} \nu_i K(x, v_i) := g_1(x) + g_2(x),$$

where

$$\begin{aligned} g_1(x) &= \sum_{i=1}^{|X_{N_1}|} \nu_i K(x, v_i) + \sum_{i=|X_{N_1}|+1}^{|X_N|} \nu_i \left(\sum_{j \in I_{N_1}(v_i)} b_j(v_i) K(x, v_j)\right), \\ g_2(x) &= \sum_{i=|X_{N_1}|+1}^{|X_N|} \nu_i \left(K(x, v_i) - \sum_{j \in I_{N_1}(v_i)} b_j(v_i) K(x, v_j)\right). \end{aligned}$$

From the expression of g_1, we see that it is a linear combination of $\{K(x, v_i) | v_i \in X_{N_1}\}$. Similarly, one can check that the summation of the absolute value of the combinational

coefficients is still bounded by 2. Hence, we have $g_1 \in 2absconv\mathscr{A}_{N_1}$ and $g_2 \in absconv\mathscr{K}_{N_1,N}$ with

$$\mathscr{K}_{N_1,N} = \left\{K(x, v_i) - \sum_{j \in I_{N_1}(v_i)} b_j(v_i) K(x, v_j)\right\}_{i=|X_{N_1}|+1}^{|X_N|}.$$

Therefore,

$$absconv\mathscr{A}_N \subset 2absconv\mathscr{A}_{N_1} + absconv\mathscr{K}_{N_1,N}.$$

And it follows that

$$\begin{aligned} &\mathscr{N}\left(absconv\mathscr{A}_N, \epsilon/4, d_{2,Q}\right) \\ &\quad \le \mathscr{N}\left(absconv\mathscr{A}_{N_1}, \epsilon/16, d_{2,Q}\right) \cdot \mathscr{N}\left(absconv\mathscr{K}_{N_1,N}, \epsilon/8, d_{2,Q}\right). \end{aligned} \tag{28}$$

To estimate the first term, we choose some suitable $\epsilon_1 > 0$ and $N_2 := N_2(\epsilon_1)$, such that the function set $\{f_j\}_{j=1}^{N_2}$ is the maximal ϵ_1-separated set in $absconv\mathscr{A}_{N_1}$ with respect to the distance $d_{2,Q}$. Hence, for any $f \in absconv\mathscr{A}_{N_1}$, f must belong to one of the N_2 balls of radius ϵ_1 centered at f_j. We denote these balls by $\mathscr{B}(f_j, \epsilon_1), j = 1, \cdots, N_2$. Moreover, as K is an admissible kernel, any function in $absconv\mathscr{A}_{N_1}$ has an unique expression as $f(x) = \sum_{i=1}^{|X_{N_1}|} \nu_i^f K(x, v_i)$ with $\sum_{i=1}^{|X_{N_1}|} |\nu_i^f| \le 1$. We define a mapping by $\Phi(f) = (\nu_1^f, \cdots, \nu_{|X_{N_1}|}^f) \in \mathbb{R}^{|X_{N_1}|}$. Under this mapping, the image of $\mathscr{B}(f_j, \epsilon_1)$ is given by

$$\begin{aligned} Im(\mathscr{B}(f_j, \epsilon_1)) &= \left\{\{\nu_i\}_{i=1}^{|X_{N_1}|} \middle| \sum_{i=1}^{|X_{N_1}|} |\nu_i| \le 1 \text{ and } \sqrt{\sum_{i=1}^{|X_{N_1}|} (\nu_i - \nu_i^{f_j})^2} \le \epsilon_1\right\} \\ &\subset \mathscr{B}_{\mathbb{R}^{|X_{N_1}|}}(\nu^{f_j}, \epsilon_1), \end{aligned} \tag{29}$$

where $\mathscr{B}_{\mathbb{R}^{|X_{N_1}|}}(\nu^{f_j}, \epsilon_1)$ denotes the ball in $\mathbb{R}^{|X_{N_1}|}$ of radius ϵ_1 centered at $\nu^{f_j} = (\nu_1^{f_j}, \cdots, \nu_{|X_{N_1}|}^{f_j})$. Furthermore, for $\forall f, g \in absconv\mathscr{A}_{N_1}$, there holds

$$\|f - g\|_{L_Q^2} \le \kappa \sqrt{\sum_{i=1}^{|X_{N_1}|} (\nu_i^f - \nu_i^g)^2}, \tag{30}$$

where $\kappa = \|K\|_{\mathscr{C}(X \times X)}$.

Next for any $\epsilon_2 > 0$, from (30) and (29), we obtain

$$\begin{aligned} \mathscr{N}(\mathscr{B}(f_j, \epsilon_1), \epsilon_2, d_{2,Q}) &\le \mathscr{N}(Im(\mathscr{B}(f_j, \epsilon_1)), \epsilon_2/\kappa, \tilde{d}_2) \\ &\le \mathscr{N}(\mathscr{B}_{\mathbb{R}^{|X_{N_1}|}}(\nu^{f_j}, \epsilon_1), \epsilon_2/\kappa, \tilde{d}_2) \\ &= \mathscr{N}(\epsilon_1 \mathscr{B}_{\mathbb{R}^{|X_{N_1}|}}, \epsilon_2/\kappa, \tilde{d}_2), \end{aligned} \tag{31}$$

where $\mathscr{B}_{\mathbb{R}^{|X_{N_1}|}}$ denotes the unit ball in $\mathbb{R}^{|X_{N_1}|}$. Recall that N_2 is the cardinality of the maximal ϵ_1-separated set in $absconv\mathscr{A}_{N_1}$. Then $N_2 \le \mathscr{N}\left(absconv\mathscr{A}_{N_1}, \epsilon_1/2, d_{2,Q}\right)$. Hence,

for any positive ϵ_1 and ϵ_2, there holds

$$\begin{aligned}&\mathscr{N}\left(absconv\mathscr{A}_{N_1},\epsilon_2,d_{2,Q}\right)\\&\leq\sum_{j=1}^{N_2}\mathscr{N}(\mathscr{B}(f_j,\epsilon_1),\epsilon_2,d_{2,Q})\\&\leq\mathscr{N}(\epsilon_1\mathscr{B}_{\mathbb{R}^{|X_{N_1}|}},\epsilon_2/\kappa,\tilde{d}_2)\cdot\mathscr{N}\left(absconv\mathscr{A}_{N_1},\epsilon_1/2,d_{2,Q}\right),\end{aligned}\tag{32}$$

where the second inequality is from (31).

Now we set $\epsilon_2=\epsilon/16$ and $\epsilon_1=1/(16\kappa\sqrt{N_1})$, then by equation (32), there holds

$$\begin{aligned}&\mathscr{N}\left(absconv\mathscr{A}_{N_1},\epsilon/16,d_{2,Q}\right)\\&\leq\mathscr{N}(\mathscr{B}_{\mathbb{R}^{|X_{N_1}|}},\epsilon\sqrt{N_1},\tilde{d}_2)\cdot\mathscr{N}\left(absconv\mathscr{A}_{N_1},1/(32\kappa\sqrt{N_1}),d_{2,Q}\right).\end{aligned}$$

And from equation (28), we further find that

$$\begin{aligned}&\mathscr{N}\left(absconv\mathscr{A}_{N},\epsilon/4,d_{2,Q}\right)\leq\mathscr{N}(\mathscr{B}_{\mathbb{R}^{|X_{N_1}|}},\epsilon\sqrt{N_1},\tilde{d}_2)\\&\cdot\mathscr{N}\left(absconv\mathscr{A}_{N_1},1/(32\kappa\sqrt{N_1}),d_{2,Q}\right)\cdot\mathscr{N}\left(absconv\mathscr{K}_{N_1,N},\epsilon/8,d_{2,Q}\right).\end{aligned}\tag{33}$$

It is noticed that $\mathbb{R}^{|X_{N_1}|}$ is a finite dimensional space and $|X_{N_1}|\leq N_1$, then

$$\mathscr{N}(\mathscr{B}_{\mathbb{R}^{|X_{N_1}|}},\epsilon\sqrt{N_1},\tilde{d}_2)\leq(1+2N_1^{-1/2}\epsilon^{-1})^{|X_{N_1}|}\leq 3^{N_1}(N_1)^{-N_1/2}\epsilon^{-N_1}.\tag{34}$$

Next, we use Lemma 13 to estimate the rest two terms. For the function set $\mathscr{A}_{N_1}$, it is easy to check that $diam\mathscr{A}_{N_1}\leq 2\kappa$. Then by (24), there holds

$$\mathscr{N}\left(absconv\mathscr{A}_{N_1},1/(32\kappa\sqrt{N_1}),d_{2,Q}\right)\leq\left(e+\frac{e}{1368\kappa^2}\right)^{8192\kappa^2N_1}.\tag{35}$$

Recall that $\Delta_{N_1}\leq\Delta_{X,s}$, then we have $Diam\mathscr{K}_{N_1,N}\leq 4c_0^s\|K\|_{\mathscr{C}^s}\Delta_{N_1}^s$ and the cardinality of $\mathscr{K}_{N_1,N}$ satisfies that $|\mathscr{K}_{N_1,N}|=|X_N|-|X_{N_1}|\leq N-1$. Also by (24), there holds

$$\begin{aligned}\mathscr{N}\left(absconv\mathscr{K}_{N_1,N},\epsilon/8,d_{2,Q}\right)&=\mathscr{N}\left(absconv\mathscr{K}_{N_1,N},\frac{\epsilon}{32c_0^s\|K\|_{\mathscr{C}^s}\Delta_{N_1}^s}\cdot 4c_0^s\|K\|_{\mathscr{C}^s}\Delta_{N_1}^s,L_Q^2\right)\\&\leq\left(e+e(2N-1)\frac{\epsilon^2}{1024c_0^{2s}\|K\|_{\mathscr{C}^s}^2\Delta_{N_1}^{2s}}\right)^{\frac{2048c_0^{2s}\|K\|_{\mathscr{C}^s}^2\Delta_{N_1}^{2s}}{\epsilon^2}}.\end{aligned}$$

Due to the choice of N_1, we have $\epsilon^{-\frac{2d}{d+2s}}-1<N_1\leq\epsilon^{-\frac{2d}{d+2s}}$. Combining the restriction (27) for ϵ, we have $\Delta_{N_1}^{2s}=2^{2s}c_X^{\frac{2s}{d}}(N_1)^{-\frac{2s}{d}}\leq 2^{4s}c_X^{\frac{2s}{d}}\epsilon^{\frac{4s}{d+2s}}$. It follows that $\Delta_{N_1}^{2s}\epsilon^{-2}\leq 2^{4s}c_X^{\frac{2s}{d}}\epsilon^{-\frac{2d}{d+2s}}$. On the other hand, one can bound $\epsilon^2\Delta_{N_1}^{-2s}$ by $2^{-2s}c_X^{-\frac{2s}{d}}\epsilon^{\frac{2d}{d+2s}}$. Recalling that $N\leq C'_{X,K}\left(\frac{1}{\epsilon}\right)^{\frac{d}{s}}+1$, we then have

$$\begin{aligned}&\mathscr{N}\left(absconv\mathscr{K}_{N_1,N},\epsilon/8,d_{2,Q}\right)\\&\leq\left(e+\frac{eC'_{X,s}\left(\frac{1}{\epsilon}\right)^{\frac{d^2}{s(d+2s)}}}{2^{2s+8}c_X^{2s/d}c_0^{2s}\|K\|_{\mathscr{C}^s}^2}\right)^{2^{4s+11}c_X^{2s/d}c_0^{2s}\|K\|_{\mathscr{C}^s}^2\epsilon^{-\frac{2d}{d+2s}}}.\end{aligned}\tag{36}$$

When $0<\epsilon\leq\epsilon_{X,K}$, substituting the bounds (34), (35) and (36) into the inequality (33), we obtain

$$\log_2\mathscr{N}(\mathscr{B}_1,\epsilon,d_{2,Q})\leq C''_{X,s}\epsilon^{-\frac{2d}{d+2s}}\log_2\left(\frac{1}{\epsilon}\right),$$

where

$$\begin{aligned}C''_{X,K}&=2+d^{-1}+8192\kappa^2\log_2\left(e+\frac{e}{1368\kappa^2}\right)\\&+2^{4s+11}c_X^{2s/d}ds^{-1}c_0^{2s}\|K\|_{\mathscr{C}^s}^2\log_2\left(e+\frac{eC'_{X,s}}{2^{2s+8}c_X^{2s/d}c_0^{2s}\|K\|_{\mathscr{C}^s}^2}\right).\end{aligned}$$

Note that $\mathscr{B}_1$ can be considered as a subset of $\mathscr{C}(X)$. We use d_∞ to denote the metric induced by $\|\cdot\|_\infty$. Then for $\epsilon_{X,s}<\epsilon\leq 1$, we find that

$$\log_2\mathscr{N}(\mathscr{B}_1,\epsilon,d_{2,Q})\leq\log_2\mathscr{N}(\mathscr{B}_1,\epsilon_{X,s}/\kappa,d_\infty)\leq\log_2\mathscr{N}(\mathscr{B}_1,\epsilon_{X,s}/\kappa,d_\infty)\epsilon^{-\frac{2d}{d+2s}}\log_2\left(\frac{2}{\epsilon}\right).$$

We take $C_{X,K}=\max\{C''_{X,K},\log_2\mathscr{N}(\mathscr{B}_1,\epsilon_{X,s}/\kappa,d_\infty)\}$, then

$$\log_2\mathscr{N}(\mathscr{B}_1,\epsilon,d_{2,Q})\leq C_{X,K}\epsilon^{-\frac{2d}{d+2s}}\log_2\left(\frac{2}{\epsilon}\right),\qquad\forall\ 0<\epsilon\leq 1.$$

Notice that the above bound is independent of the distribution Q. Then for any sample $\mathbf{x}=\{x_i\}_{i=1}^\ell\in X^\ell$ with $\ell\in\mathbb{N}$, the above estimate holds true for $Q=\frac{1}{\ell}\sum_{i=1}^\ell\delta_{x_i}$. Consequently, we prove our conclusion if s is an integer. When s is not an integer, one can check that the proof is also valid if we replace s by its integral part $\lfloor s\rfloor$. Thus we complete the proof of Theorem 11. ■

4. Convergence analysis

In this section, we investigate the convergence behavior of ℓ_q-kernel regression based on concentration techniques involving ℓ_2-empirical covering numbers. Our error analysis presented in this part is under the capacity assumption (20). For an admissible kernel $K\in\mathscr{C}^s$ with $s>0$, recall the previous result obtained in Shi et al. (2011) for $0<s<2$. Then under the assumption of Theorem 11, one can check that the assumption (20) can be satisfied with $0<p<2$. Concretely, when $0<s<2$, $p=\frac{2d}{d+2\min\{1,s\}}$; when $s\geq 2$, p can be chosen to be any constant satisfying $p>\frac{2d}{d+2\lfloor s\rfloor}$.

4.1. Deriving the estimator for the total error

Recall the quantity $\mathscr{S}_1$ defined for an estimator $\hat{f}$ in Proposition 6, we can rewrite it as

$$\mathscr{S}_1=\mathcal{S}_{\mathbf{z}}(\hat{f})-\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma}),$$

where the quantity $\mathcal{S}_{\mathbf{z}}$ is defined for $f\in\mathscr{C}(X)$ by

$$\mathcal{S}_{\mathbf{z}}(f)=\{\mathscr{E}(f)-\mathscr{E}(f_\rho)\}-\{\mathscr{E}_{\mathbf{z}}(f)-\mathscr{E}_{\mathbf{z}}(f_\rho)\}.$$

We use the following proposition to estimate $\mathscr{S}_1$. Recall that $\mathscr{B}_R$ is defined as (19).

Proposition 14 *If $\mathscr{B}_1$ satisfies the capacity condition (20) with $0 < p < 2$, then for any $R \geq 1$ and $0 < \delta < 1$, with confidence $1-\delta$, there hold*

$$\mathcal{S}_{\mathbf{z}}(\pi_M(f)) \leq \frac{1}{2}\left\{\mathscr{E}(\pi_M(f)) - \mathscr{E}(f_\rho)\right\} + \frac{176M^2\log(1/\delta)}{m} + \tilde{c}_{K,p}M^2 m^{-\frac{2}{2+p}} R^{\frac{2p}{2+p}}, \qquad \forall f \in \mathscr{B}_R, \tag{37}$$

and

$$\mathcal{S}_{\mathbf{z}}(f) \leq \frac{1}{2}\left\{\mathscr{E}(f) - \mathscr{E}(f_\rho)\right\} + \frac{20(3M+\kappa)^2R^2\log(1/\delta)}{m} + \tilde{c}_{K,p}(3M+\kappa)^2 m^{-\frac{2}{2+p}} R^2, \qquad \forall f \in \mathscr{B}_R. \tag{38}$$

Here $\tilde{c}_{K,p} > 0$ is a constant depending only on p and the kernel function K and $\kappa = \|K\|_{\mathscr{C}(X\times X)}$. The above bounds also hold true for $-\mathcal{S}_{\mathbf{z}}(\pi_M(f))$ and $-\mathcal{S}_{\mathbf{z}}(f)$.

We shall prove Proposition 14 in the Appendix by using the entropy integral based on ℓ_2-empirical covering numbers. Now we can derive the estimator for the total error. For $R \geq 1$ and $0 < q \leq 1$, denote

$$\mathcal{W}_q(R) = \left\{\mathbf{z} \in Z^m : \ \|\mathbf{c}_q^{\mathbf{z}}\|_q^q \leq R\right\}. \tag{39}$$

Proposition 15 *Assume that approximation assumption (14) with $0 < \beta \leq 1$ and capacity condition (20) with $0 < p < 2$ are valid, and the estimator $\hat{f}_q$ satisfies Assumption 1 with $0 < q < 1$. If $0 < \gamma \leq 1$, $0 < \delta < 1$ and $R \geq 1$, then there is a subset $V_R^{(q)}$ of Z^m with measure at most δ such that*

$$\begin{aligned}
&\mathscr{E}(\pi_M(\hat{f}_q)) - \mathscr{E}(f_\rho) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \\
&\leq 2\tilde{c}_{K,p}M^2 m^{-\frac{2}{2+p}} R^{\frac{2p}{q(2+p)}} + C_1\log^3(18/\delta)\max\left\{\gamma^{\beta-2}m^{-2}, \gamma^{\beta-1}m^{-\frac{2}{2+p}}\right\} \\
&\quad + (3\sqrt{c_\beta} + 6c_\beta)\gamma^\beta + 2\gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q, \qquad \forall \mathbf{z} \in \mathcal{W}_q(R)\backslash V_R^{(q)}.
\end{aligned} \tag{40}$$

Moreover, when $q = 1$, there are subsets $V_R^{(1)}$ and $\widetilde{V}_R^{(1)}$ of Z^m with each measure at most δ such that

$$\begin{aligned}
&\mathscr{E}(\pi_M(\hat{f}_1)) - \mathscr{E}(f_\rho) + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_1 \\
&\leq 2\tilde{c}_{K,p}M^2 m^{-\frac{2}{2+p}} R^{\frac{2p}{2+p}} + C_1\log^3(18/\delta)\max\left\{\gamma^{\beta-2}m^{-2}, \gamma^{\beta-1}m^{-\frac{2}{2+p}}\right\} \\
&\quad + (3\sqrt{c_\beta} + 6c_\beta)\gamma^\beta, \qquad \forall \mathbf{z} \in \mathcal{W}_1(R)\backslash V_R^{(1)},
\end{aligned} \tag{41}$$

and

$$\begin{aligned}
\mathscr{E}(\hat{f}_1) - \mathscr{E}(f_\rho) + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_1 &\leq 2(20 + \tilde{c}_{K,p})(3M+\kappa)^2\log(3/\delta)m^{-\frac{2}{2+p}}R^2 \\
&\quad + \widetilde{C}_1\log^3(18/\delta)\max\left\{\gamma^{\beta-2}m^{-2}, \gamma^{\beta-1}m^{-\frac{2}{2+p}}, \gamma^\beta\right\}, \forall \mathbf{z} \in \mathcal{W}_1(R)\backslash\widetilde{V}_R^{(1)}.
\end{aligned} \tag{42}$$

Here C_1 and $\widetilde{C}_1$ are positive constants depending on $\kappa, \tilde{c}_{K,p}$, M and c_β.

Proof Let $\hat{f}$ be the estimator under consideration. We estimate $\mathscr{S}_1$ by bounding $\mathcal{S}_{\mathbf{z}}(\hat{f})$ and $-\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma})$ respectively. Due to Proposition 14, the quantities $-\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma})$ and $\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma})$ have the same upper bound. We thus turn to estimate $\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma})$. For $(\gamma,\delta) \in (0,1)^2$, let

$$R(\gamma,\delta) := \frac{2\kappa\sqrt{\mathcal{D}(\gamma)}\log(6/\delta)}{\gamma m} + \sqrt{\frac{2\mathcal{D}(\gamma)\log(6/\delta)}{\gamma m}} + \sqrt{\frac{\mathcal{D}(\gamma)}{\gamma}}.$$

From the proof of Proposition 5 in Guo and Shi (2012), there exist two subsets of Z^m denoted by U_1 and U_2 with $\rho(U_i) \leq \delta/3, i = 1,2$, such that

$$f_{\mathbf{z},\gamma} \in \mathscr{B}_{R(\gamma,\delta)}, \qquad \forall\ \mathbf{z} \in Z^m\backslash U_1, \tag{43}$$

and

$$\mathscr{E}(f_{\mathbf{z},\gamma}) - \mathscr{E}(f_\rho) \leq 8\kappa^2\left(2\kappa^2+1\right)\log^2(6/\delta)\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2m^2} + \frac{\mathcal{D}(\gamma)}{\gamma m}\right\} + 2\mathcal{D}(\gamma), \forall\ \mathbf{z} \in Z^m\backslash U_2. \tag{44}$$

We use inequality (38) to bound $\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma})$. Now (43) and (44) are both valid for $f_{\mathbf{z},\gamma}$ with $\mathbf{z} \in Z^m\backslash(U_1 \cup U_2)$. Then there exists a subset U_3 of Z^m with measure at most $\delta/3$ such that

$$\begin{aligned}
\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma}) &\leq 4\kappa^2\left(2\kappa^2+1\right)\log^2(6/\delta)\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2m^2} + \frac{\mathcal{D}(\gamma)}{\gamma m}\right\} \\
&\quad + \mathcal{D}(\gamma) + \frac{20(3M+\kappa)^2(R(\gamma,\delta))^2\log(3/\delta)}{m} \\
&\quad + \tilde{c}_{K,p}(3M+\kappa)^2 m^{-\frac{2}{2+p}}(R(\gamma,\delta))^2, \forall \mathbf{z} \in Z^m\backslash(U_1\cup U_2\cup U_3).
\end{aligned}$$

Note that $\rho(U_1 \cup U_2 \cup U_3) \leq \delta$ and

$$(R(\gamma,\delta))^2 \leq (8\kappa^2+4)\log^2(6/\delta)\left(\frac{\mathcal{D}(\gamma)}{\gamma^2m^2} + \frac{\mathcal{D}(\gamma)}{\gamma m} + \frac{\mathcal{D}(\gamma)}{\gamma}\right).$$

Therefore, with confidence $1-\delta$, there holds

$$\begin{aligned}
\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma}) &\leq C_\kappa\log^2(6/\delta)\max\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2m^2}, \frac{\mathcal{D}(\gamma)}{\gamma m}\right\} + \mathcal{D}(\gamma) \\
&\quad + (20 + \tilde{c}_{K,p})C_{\kappa,M}\log^3(6/\delta)m^{-\frac{2}{2+p}}\max\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2m^2}, \frac{\mathcal{D}(\gamma)}{\gamma}\right\},
\end{aligned} \tag{45}$$

where $C_\kappa = 8\kappa^2\left(2\kappa^2+1\right)$ and $C_{\kappa,M} = 3(3M+\kappa)^2(8\kappa^2+4)$.

Next, for $\hat{f} = \hat{f}_1$ or $\pi_M(\hat{f}_q)$ with $0 < q \leq 1$, we consider bounding the quantity $\mathcal{S}_{\mathbf{z}}(\hat{f})$ with $\mathbf{z} \in \mathcal{W}(R)$. It is easy if we notice that $\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \leq R$ implies the corresponding estimator $\hat{f}_q \in \mathscr{B}_{R^{1/q}}$. Thus we can bound $\mathcal{S}_{\mathbf{z}}(\hat{f})$ by directly applying Proposition 14.

When $\hat{f} = \pi_M(\hat{f}_q)$ with $0 < q < 1$, combining the bounds on $\mathscr{S}_1 + \mathscr{S}_2$ given in Lemma 7, we are able to derive the estimator for the total error bound. Due to (37), there exist subsets $V_1^{(q)}$ with measure at most $\delta/3$ such that

$$\begin{aligned}
\mathcal{S}_{\mathbf{z}}(\pi_M(\hat{f}_q)) &\leq \frac{1}{2}\left\{\mathscr{E}(\pi_M(\hat{f}_q)) - \mathscr{E}(f_\rho)\right\} + \frac{176M^2\log(3/\delta)}{m} \\
&\quad + \tilde{c}_{K,p}M^2 m^{-\frac{2}{2+p}} R^{\frac{2p}{q(2+p)}}, \qquad \forall \mathbf{z} \in \mathcal{W}_q(R)\backslash V_1^{(q)}.
\end{aligned}$$

Similarly, from (45) and Lemma 7, we can find subsets V_2 and V_3 such that

$$-\mathcal{S}_{\mathbf{z}}(f_{\mathbf{z},\gamma}) \le (C_\kappa + C_{K,p}) \log^3(18/\delta) \max\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2 m^2}, \frac{\mathcal{D}(\gamma)}{\gamma m^{2/(2+p)}}\right\} + \mathcal{D}(\gamma), \forall \mathbf{z} \in Z^m \backslash V_2,$$

and

$$\mathscr{S}_2 + \mathscr{S}_3 \le 16\kappa^2 \left(2\kappa^2+1\right) \log^2(12/\delta) \max\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2 m^2}, \frac{\mathcal{D}(\gamma)}{\gamma m}\right\} + \frac{(2\kappa+1)\sqrt{\mathcal{D}(\gamma)}\log(12/\delta)}{m} + \frac{3}{2}\sqrt{\gamma \mathcal{D}(\gamma)} + 2\mathcal{D}(\gamma), \forall \mathbf{z} \in Z^m \backslash V_3,$$

where $C_{K,p} = (20 + \tilde{c}_{K,p}) C_{\kappa,M}$ and $\rho(V_i) \le \delta/3$ for $i = 2, 3$.

Let $V_R^{(q)} = V_1^{(q)} \cup V_2 \cup V_3$, then $\rho(V_R^{(q)}) \le \delta$. Recall the error decomposition formula (16). We combine the above three bounds and obtain

$$\begin{aligned}
&\mathscr{E}(\pi_M(\hat{f}_q)) - \mathscr{E}(f_\rho) + \gamma \|\mathbf{c}_q^{\mathbf{z}}\|_q^q \\
&\le \frac{1}{2}\left\{\mathscr{E}(\pi_M(\hat{f}_q)) - \mathscr{E}(f_\rho)\right\} + (176M^2 + 16\kappa^2(2\kappa^2+1)) \log^2(12/\delta) \max\left\{\frac{1}{m}, \frac{\mathcal{D}(\gamma)}{\gamma m}\right\} \\
&+ \tilde{c}_{K,p} M^2 m^{-\frac{2}{2+p}} R^{\frac{2p}{q(2+p)}} + (C_\kappa + C_{K,p} + 16\kappa^2\left(2\kappa^2+1\right)) \log^3(18/\delta) \max\left\{\frac{\mathcal{D}(\gamma)}{\gamma^2 m^2}, \frac{\mathcal{D}(\gamma)}{\gamma m^{2/(2+p)}}\right\} \\
&+ \frac{(2\kappa+1)\sqrt{\mathcal{D}(\gamma)}\log(12/\delta)}{m} + \frac{3}{2}\sqrt{\gamma\mathcal{D}(\gamma)} + 3\mathcal{D}(\gamma) + \gamma m^{1-q} \|\mathbf{c}_1^{\mathbf{z}}\|_1^q, \quad \forall \mathbf{z} \in \mathcal{W}_q(R) \backslash V_R^{(q)}.
\end{aligned}$$

Due to this inequality, we use Approximation condition (14) and find that the inequality (40) holds true with

$$C_1 = 2((20 + \tilde{c}_{K,p}) C_{\kappa,M} + 176M^2 + 40\kappa^2\left(2\kappa^2+1\right)) c_\beta + (4\kappa+2)\sqrt{c_\beta}.$$

Following the same method, when $\hat{f} = \pi_M(\hat{f}_1)$, we can prove inequality (41) by using the error decomposition formula (17) .

We next focus on the case $\hat{f} = \hat{f}_1$. Due to inequality (38), for $R \ge 1$, with confidence $1 - \delta/3$, there holds

$$\mathcal{S}_{\mathbf{z}}(\hat{f}_1) \le \frac{1}{2}\left\{\mathscr{E}(\hat{f}_1) - \mathscr{E}(f_\rho)\right\} + (20 + \tilde{c}_{K,p})(3M+\kappa)^2 \log(3/\delta) m^{-\frac{2}{2+p}} R^2, \forall \mathbf{z} \in \mathcal{W}_1(R).$$

Also by the same analysis above, we obtain inequality (42) with

$$\widetilde{C}_1 = 6((20 + \tilde{c}_{K,p})(3M+\kappa)^2(8\kappa^2+4) + 8\kappa^2\left(2\kappa^2+1\right) + 1) c_\beta + (4\kappa+5)\sqrt{c_\beta}.$$

Thus we complete our proof. ■

4.2. Bounding $\|\mathbf{c}_q^{\mathbf{z}}\|_q$ by iteration

To apply Proposition 15 for error analysis, we need to determine some $R \ge 1$ for the set $\mathcal{W}_q(R)$ given by (39). To this end, we shall apply an iteration technique to obtain a tight

bound for $\|\mathbf{c}_q^{\mathbf{z}}\|_q^q$. This technique can be found in Steinwart and Scovel (2007); Wu et al. (2006). Take the case $q = 1$ for example. Recall that $\mathbf{c}_1^{\mathbf{z}}$ is the global minimizer of the target functional $\mathscr{T}_{\gamma,1}$ defined by (3). Hence $\mathscr{T}_{\gamma,1}(\mathbf{c}_1^{\mathbf{z}}) \le \mathscr{T}_{\gamma,1}(\mathbf{0})$. This inequality leads to a trivial bound on $\|\mathbf{c}_1^{\mathbf{z}}\|_1$, which is given by

$$\|\mathbf{c}_1^{\mathbf{z}}\|_1 \le \frac{M^2}{\gamma}, \qquad \forall\, \mathbf{z} \in Z^m. \tag{46}$$

By applying inequality (41) iteratively, we are able to improve the above bound to the order $O(\gamma^{\beta-1})$ with some suitable choice of $\gamma = \gamma(m)$. Similarly, when $0 < q < 1$, a better estimates for $\|\mathbf{c}_q^{\mathbf{z}}\|_q^q$ can be derived based on the inequality (40). The following proposition illustrates the detailed process of iteration.

Proposition 16 *Under the assumptions of Proposition 15, let $(\delta, \tau) \in (0,1)^2$ and $J(\tau, p, q)$ be a constant given by*

$$J(\tau, p, q) = \max\left\{2, \frac{\log \frac{(2-(p+2)\tau)p}{(q-p\tau)(p+2)}}{\log \frac{2p}{q(p+2)}}\right\}. \tag{47}$$

For $q = 1$, take $\gamma = m^{-\tau}$ with $0 < \tau < \frac{2}{2+p}$. Then with confidence $1 - \delta$, there holds

$$\|\mathbf{c}_1^{\mathbf{z}}\|_1 \le C_2 \left(\log(1/\delta) + \log(J(\tau, p, 1))\right)^3 m^{(1-\beta)\tau}. \tag{48}$$

For $0 < q < 1$ satisfying $q > \max\left\{\frac{2p}{p+2}, \frac{p}{p+2\beta}\right\}$, take $\gamma = m^{-\tau}$ with $\frac{1-q}{1-q(1-\beta)} < \tau < \frac{2}{2+p}$. Then with confidence $1 - \delta$, there holds

$$\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \le \widetilde{C}_2 \left(\log J(\tau, p, 1)\right)^{3q} \left(\log(1/\delta) + \log(J(\tau, p, q) + 1)\right)^3 m^{1-q+q(1-\beta)\tau}. \tag{49}$$

Here C_2 and $\widetilde{C}_2$ are positive constants depending only on $\kappa, \tilde{c}_{K,p}$, M and c_β.

Proof Take $0 < \delta < 1$ and $\gamma = m^{-\tau}$ under the restriction $0 < \tau < \frac{2}{p+2}$. For $0 < q \le 1$, we shall verify the following inequality derived from Proposition 15, that is

$$\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \le \max\{a_m R^\theta, b_m\}, \qquad \forall \mathbf{z} \in \mathcal{W}_q(R) \setminus V_R^{(q)}, \tag{50}$$

where $V_R^{(q)}$ is a subset of Z^m with measure at most δ, $\theta = \frac{2}{q(p+2)}$, $a_m = 4\tilde{c}_{K,p} M^2 m^{\tau - \frac{2}{2+p}}$ and b_m will be given explicitly in accordance with specific situations. This inequality ensures that

$$\mathcal{W}_q(R) \subset \mathcal{W}_q\left(\max\{a_m R^\theta, b_m\}\right) \cup V_R^{(q)} \tag{51}$$

with $\rho(V_R^{(q)}) \le \delta$. We then apply the inclusion (51) for a sequence of radiuses $\{R^{(j)}\}_{j\in\mathbb{N}}$ defined by $R^{(0)} = M_\delta m^\tau$ with a suitable chosen $M_\delta > 0$ and

$$R^{(j)} = \max\left\{a_m (R^{(j-1)})^\theta, b_m\right\}, \qquad j \in \mathbb{N}. \tag{52}$$

Then (51) holds for each $R^{(j)}$, which implies $\mathcal{W}_q(R^{(j-1)}) \subseteq \mathcal{W}_q(R^{(j)}) \cup V^{(q)}_{R^{(j-1)}}$ with $\rho(V^{(q)}_{R^{(j-1)}}) \leq \delta$. Applying this inclusion for $j = 1, 2, \ldots, J$ with J to be determined later, we see that

$$\mathcal{W}_q(R^{(0)}) \subseteq \mathcal{W}_q(R^{(1)}) \cup V^{(q)}_{R^{(0)}} \subseteq \cdots \subseteq \mathcal{W}_q(R^{(J)}) \cup \left(\cup_{j=0}^{J-1} V^{(q)}_{R^{(j)}}\right).$$

By the definition of the sequence $\{R^{(j)}\}_j$, we thus have

$$R^{(J)} = \max\left\{a_m^{1+\theta+\theta^2+\ldots+\theta^{J-1}}(R^{(0)})^{\theta^J}, a_m^{1+\theta+\theta^2+\ldots+\theta^{J-2}} b_m^{\theta^{J-1}}, \ldots, a_m b_m^{\theta}, b_m\right\}. \tag{53}$$

Recall that $R^{(0)} = M_\delta m^\tau$. When $\theta \neq 1$, the first term on the right-hand side of equation (53) can be bounded as

$$\begin{aligned} a_m^{1+\theta+\theta^2+\ldots+\theta^{J-1}}(R^{(0)})^{\theta^J} &= a_m^{(\theta^J-1)/(\theta-1)}(R^{(0)})^{\theta^J} \\ &\leq (4\tilde{c}_{K,p}M^2)^{(\theta^J-1)/(\theta-1)} M_\delta^{\theta^J} m^{\frac{\tau(\theta^{J+1}-1)}{\theta-1} - \frac{2(\theta^J-1)}{(p+2)(\theta-1)}}. \end{aligned}$$

For the remaining terms, we find that

$$\begin{aligned} &\max\left\{a_m^{1+\theta+\theta^2+\ldots+\theta^{J-2}} b_m^{\theta^{J-1}}, \ldots, a_m b_m^{\theta}, b_m\right\} \\ &= \max\left\{(a_m)^{(\theta^{J-1}-1)/(\theta-1)}(b_m)^{\theta^{J-1}}, \ldots, a_m b_m^{\theta}, b_m\right\} \\ &= \max\left\{a_m^{(\theta^{J-1}-1)/(\theta-1)} b_m^{\theta^{J-1}}, b_m\right\}. \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} R^{(J)} \leq \max\Big\{&A_{\theta,J} M_\delta^{\theta^J} m^{\frac{\tau(\theta^{J+1}-1)}{\theta-1} - \frac{2(\theta^J-1)}{(p+2)(\theta-1)}}, \\ &A_{\theta,J-1} m^{(\tau - \frac{2}{2+p})(\theta^{J-1}-1)/(\theta-1)} b_m^{\theta^{J-1}}, b_m\Big\}, \end{aligned} \tag{54}$$

where $A_{\theta,J}$ is defined as $A_{\theta,J} = (4\tilde{c}_{K,p}M^2)^{(\theta^J-1)/(\theta-1)}$ for any $J \in \mathbb{N}$ and $\theta \neq 1$.

Now we go back to our concrete examples. Due to the above analysis, we need to determine b_m, M_δ and J depending on the concerned cases.

When $q = 1$, then $\theta = \frac{2p}{p+2} < 1$ and we can take $b_m = C_1' \log^3(18/\delta) m^{(1-\beta)\tau}$ with $C_1' = 2C_1 + 6\sqrt{c_\beta} + 12c_\beta$. One can check that the inequality (50) is valid due to (41). Recall the trivial bound (46) on $\|\mathbf{c}_1^{\mathbf{z}}\|_1$, which implies $\|\mathbf{c}_1^{\mathbf{z}}\|_1 \leq M^2/\gamma = M^2 m^\tau$ for $\forall \mathbf{z} \in Z^m$. We thus take $M_\delta = M^2$ and $R^{(0)} = M^2 m^\tau$. From (54), we further find that

$$R^{(J)} \leq \max\left\{\left(4\tilde{c}_{K,p}M^2\right)^{\frac{2+p}{2-p}} M^2, C_1' \log^3(18/\delta)\left(4\tilde{c}_{K,p}M^2\right)^{\frac{2+p}{2-p}}, C_1' \log^3(18/\delta)\right\} m^{\theta_1},$$

where the power index θ_1 is given by

$$\begin{aligned} \theta_1 = \max\Big\{&\left(\frac{2p}{p-2}\left(\frac{2p}{p+2}\right)^J - \frac{p+2}{p-2}\right)\tau - \frac{2}{p-2}\left(\left(\frac{2p}{p+2}\right)^J - 1\right), \\ &(1-\beta)\tau, (1-\beta)\tau + \left(\left(\frac{2p}{p+2}\right)^{J-1} - 1\right)\left((1-\beta)\tau + \frac{(p+2)\tau - 2}{p-2}\right)\Big\}. \end{aligned}$$

Let J be the smallest integer such that

$$\left(\frac{2p}{p-2}\left(\frac{2p}{p+2}\right)^J - \frac{p+2}{p-2}\right)\tau - \frac{2}{p-2}\left(\left(\frac{2p}{p+2}\right)^J - 1\right) \leq 0$$

i.e., J is the integer satisfying

$$\max\left\{1, \frac{\log\frac{2-(p+2)\tau}{2-2p\tau}}{\log\frac{2p}{p+2}}\right\} \leq J < \max\left\{2, \frac{\log\frac{(2-(p+2)\tau)p}{(1-p\tau)(p+2)}}{\log\frac{2p}{p+2}}\right\}.$$

Then $\theta_1 = (1-\beta)\tau$ and

$$R^{(J)} \leq \left(\left(4\tilde{c}_{K,p}M^2\right)^{\frac{2+p}{2-p}} M^2 + C_1'\right)\log^3(18/\delta) m^{(1-\beta)\tau}.$$

Since $Z^m = \mathcal{W}_1(R^{(0)})$ and $\rho\left(\cup_{j=0}^{J-1} V^{(1)}_{R^{(j)}}\right) \leq J\delta$, the measure of the set $\mathcal{W}_1(R^{(J)})$ is at least $1 - J\delta$. By scaling $J\delta$ to δ, we derive the bound (48) with

$$C_2 = 64\left(\left(4\tilde{c}_{K,p}M^2\right)^{\frac{2+p}{2-p}} M^2 + C_1'\right).$$

Next we turn to the case $0 < q < 1$ with $q > \max\left\{\frac{2p}{p+2}, \frac{p}{p+2\beta}\right\}$. The estimation in this case is more involved. In order to determine b_m and $R^{(0)}$ in this case, we need to use the result just obtained for $\|\mathbf{c}_1^{\mathbf{z}}\|_1$. In fact, for any $0 < \delta < 1$ and $0 < \tau < \frac{2}{p+2}$, with confidence $1 - \delta/2$ there holds

$$\|\mathbf{c}_1^{\mathbf{z}}\|_1 \leq C_2\left(\log J(\tau, p, 1) + \log(2/\delta)\right)^3 m^{(1-\beta)\tau}, \tag{55}$$

where $J(\tau, p, 1)$ is given by (47) at $q = 1$. The inequality (40) asserts that, with confidence $1 - \delta/2$, there holds

$$\begin{aligned} \mathscr{E}(\pi_M(\hat{f}_q)) - \mathscr{E}(f_\rho) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q &\leq 2\tilde{c}_{K,p}M^2 m^{-\frac{2}{2+p}} R^{\frac{2p}{q(2+p)}} \\ &+ \frac{1}{2}C_1' \log^3(36/\delta)\gamma^\beta + 2\gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q, \qquad \forall \mathbf{z} \in \mathcal{W}_q(R). \end{aligned} \tag{56}$$

It is easy to check under the restriction $q > \max\left\{\frac{2p}{p+2}, \frac{p}{p+2\beta}\right\}$, there holds $\frac{1-q}{1-q(1-\beta)} < \frac{2}{2+p}$ and $\frac{2p}{q(2+p)} < 1$. Then we can take $\gamma = m^{-\tau}$ with $\frac{1-q}{1-q(1-\beta)} < \tau < \frac{2}{2+p}$. Combining (55) and (56), we find that (50) is satisfied with $b_m = B_\delta m^{\tilde{\alpha}}$, where

$$B_\delta = \left(C_1' + 4C_2^q\right)\left(\log J(\tau, p, 1)\right)^{3q}\log^3(36/\delta) \text{ and } \tilde{\alpha} = 1 - q + q(1-\beta)\tau.$$

We further define

$$R^{(0)} = M_\delta m^\tau \text{ and } M_\delta = M^2 + C_2^q\left(\log J(\tau, p, 1) + \log(1/\delta)\right)^{3q}. \tag{57}$$

Recall that the estimator $\hat{f}_q$ satisfies Assumption 1. Then we have

$$\gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \leq \mathscr{T}_{\gamma,q}(\mathbf{c}_q^{\mathbf{z}}) \leq \mathscr{T}_{\gamma,q}(\mathbf{c}_1^{\mathbf{z}}) \leq \mathscr{T}_{\gamma,1}(\mathbf{c}_1^{\mathbf{z}}) + \gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q \leq \mathscr{T}_{\gamma,1}(\mathbf{0}) + \gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q.$$

Due to this inequality and the bound on $\|\mathbf{c}_1^{\mathbf{z}}\|_1$, one can easily check that with confidence $1-\delta$, there holds

$$\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \le M^2 m^\tau + m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q \le M_\delta m^\tau,$$

where the last inequality is due to $\tau > \frac{1-q}{1-q(1-\beta)}$. Hence the measure of the set $\mathcal{W}_q(R^{(0)})$ is at least $1-\delta$. We thus can iteratively define $R^{(j)}$ as (52) with $R^{(0)}$ given by (57). Now we can follow the same fashion to derive our desire bounds.

Since $\theta = \frac{2p}{q(p+2)} < 1$, due to the inequality (54), we have

$$R^{(J)} \le \max\left\{A_{\theta,J} M_\delta^{\theta^J}, A_{\theta,J-1} B_\delta^{\theta^{J-1}}, B_\delta\right\} m^{\theta_2},$$

where θ_2 is given by

$$\max\left\{\left(\frac{2p}{2p-q(p+2)}\left(\frac{2p}{q(p+2)}\right)^J - \frac{q(p+2)}{2p-q(p+2)}\right)\tau - \frac{2q}{2p-q(p+2)}\left(\left(\frac{2p}{q(p+2)}\right)^J - 1\right),\right.$$
$$\left.\tilde\alpha, \tilde\alpha + \left(\left(\frac{2p}{q(p+2)}\right)^{J-1} - 1\right)\left(\tilde\alpha + \frac{q(p+2)}{2p-q(p+2)}\tau - \frac{2q}{2p-q(p+2)}\right)\right\}.$$

We chose J to be the smallest integer such that

$$\max\left\{1, \frac{\log\frac{2q-q(p+2)\tau}{2q-2p\tau}}{\log\frac{2p}{q(p+2)}}\right\} \le J < \max\left\{2, \frac{\log\frac{(2-(p+2)\tau)p}{(q-p\tau)(p+2)}}{\log\frac{2p}{q(p+2)}}\right\}.$$

Under the choice of J, we have $\theta_2 = \tilde\alpha$,

$$R^{(J)} \le (M^2 + C_1' + 4C_2^q)\left(4\tilde c_{K,p} M^2\right)^{\frac{2+p}{2-p}} (\log J(\tau,p,1))^{3q} \log^3(36/\delta) m^{\tilde\alpha}$$

and the measure of the set $\mathcal{W}_1(R^{(J)})$ is at least $1-(J+1)\delta$. We thus can derive the inequality (49) with

$$\widetilde{C}_2 = 64(M^2 + C_1' + 4C_2^q)\left(4\tilde c_{K,p} M^2\right)^{\frac{2+p}{2-p}}$$

by scaling $(J+1)\delta$ to δ. ■

4.3. Deriving the convergence rates

In this subsection, we estimate the total error for algorithm (2) and derive the explicit convergence rates. Recall that for $0 < q \le 1$, $\hat f_q$ denotes the estimator given by the algorithm (2).

Theorem 17 *Assume that approximation assumption (14) with $0 < \beta \le 1$ and capacity condition (20) with $0 < p < 2$ are valid. Let $(\delta,\tau,q) \in (0,1]^3$ and $J(\tau,p,q)$ be a constant*

given by (47). When $q=1$, take $\gamma = m^{-\tau}$ with $0 < \tau < \frac{2}{2+p}$. Then with confidence $1-\delta$, there holds

$$\|\hat f_1 - f_\rho\|_{L^2_{\rho_X}}^2 \le C_3 \log(6/\delta)\left(\log(2/\delta) + \log(J(\tau,p,1))\right)^6 m^{-\widetilde\Theta}, \tag{58}$$

where

$$\widetilde\Theta = \min\left\{\frac{2}{2+p} - 2(1-\beta)\tau, \beta\tau\right\}. \tag{59}$$

Additionally, if an estimator $\hat f_q$ satisfies Assumption 1 with $\max\left\{\frac{2p}{p+2}, \frac{p}{p+2\beta}\right\} < q < 1$. Take $\gamma = m^{-\tau}$ with $\frac{1-q}{1-q(1-\beta)} < \tau < \frac{2}{2+p}$. Then with confidence $1-\delta$, there holds

$$\|\pi_M(\hat f_q) - f_\rho\|_{L^2_{\rho_X}}^2 \le \widetilde C_3 \left(\log(9/\delta) + \log(J(\tau,p,1)) + \log(J(\tau,p,q)+1)\right)^3 m^{-(\tau-\tilde\alpha)}, \tag{60}$$

where $\tilde\alpha = 1-q+q(1-\beta)\tau$. Here C_3 and $\widetilde C_3$ are positive constants independent of m or δ.

Proof Recall that for any estimator $\hat f$ under consideration, there holds

$$\|\hat f - f_\rho\|_{L^2_{\rho_X}}^2 = \mathscr{E}(\hat f) - \mathscr{E}(f_\rho).$$

Therefore, due to inequality (42) in Proposition 15, there is a subset $\widetilde V_R$ of Z^m with measure at most δ such that

$$\|\hat f - f_\rho\|_{L^2_{\rho_X}}^2 \le 2(20+\tilde c_{K,p})(3M+\kappa)^2 \log(3/\delta) m^{-\frac{2}{2+p}} R^2$$
$$+\widetilde C_1 \log^3(18/\delta)\max\left\{\gamma^{\beta-2}m^{-2}, \gamma^{\beta-1}m^{-\frac{2}{2+p}}, \gamma^\beta\right\}, \forall \mathbf{z} \in \mathcal{W}_1(R)\backslash\widetilde V_R.$$

Now we choose R to be the right-hand side of inequality (48), i.e.,

$$R = C_2\left(\log(1/\delta) + \log(J(\tau,p,1))\right)^3 m^{(1-\beta)\tau}.$$

Then the measure of the set $\mathcal{W}_1(R)\setminus\widetilde V_R$ is at least $1-2\delta$. So with confidence at least $1-2\delta$, there holds

$$\|\hat f_1 - f_\rho\|_{L^2_{\rho_X}}^2 \le C_3 \max\left\{\log^3(3/\delta), \log(3/\delta)\left(\log(1/\delta) + \log(J(\tau,p,1))\right)^6\right\} m^{-\widetilde\Theta},$$

where $\widetilde\Theta$ is given by (59) and

$$C_3 = 54(20+\tilde c_{K,p})(3M+\kappa)^2 C_2^2 + 27\widetilde C_1.$$

Then we obtain the error bound (58) by scaling 2δ to δ.

Next we prove the error bound (60). Recalling the total error bound (40), for some $R \ge 1$, there is a subset $V_R^{(q)}$ with measure at most δ such that

$$\|\pi_M(\hat f_q) - f_\rho\|_{L^2_{\rho_X}}^2 \le 2\tilde c_{K,p} M^2 m^{-\frac{2}{2+p}} R^{\frac{2p}{q(2+p)}}$$
$$+\widetilde C_1' \log^3(18/\delta)\max\left\{\gamma^{\beta-2}m^{-2}, \gamma^{\beta-1}m^{-\frac{2}{2+p}}, \gamma^\beta\right\} + 2\gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q, \ \forall \mathbf{z} \in \mathcal{W}_q(R)\backslash V_R^{(q)},$$

where $\widetilde{C}_1' = C_1 + 3\sqrt{c_\beta} + 6c_\beta$. Using the same argument, the inequality (48) ensures the existence of a subset V_1' with measure at most δ such that

$$\|\mathbf{c}_1^{\mathbf{z}}\|_1 \le C_2 \left(\log(1/\delta) + \log(J(\tau,p,1))\right)^3 m^{(1-\beta)\tau}, \qquad \forall \mathbf{z} \in Z^m \backslash V_1'.$$

If we take R to be the right-hand side of the inequality (49), i.e.,

$$R = \widetilde{C}_2 \left(\log J(\tau,p,1)\right)^{3q} \left(\log(1/\delta) + \log(1 + J(\tau,p,q))\right)^3 m^{\tilde{\alpha}}$$

with $\tilde{\alpha} = 1 - q + q(1-\beta)\tau$. Then the measure of the set $\mathcal{W}_q(R)$ is at least $1-\delta$. Combining these bounds, for $\forall \mathbf{z} \in \mathcal{W}_q(R) \backslash (V_R^{(q)} \cup V_1')$, there holds

$$\begin{aligned}
&\|\pi_M(\hat{f}_q) - f_\rho\|_{L^2_{\rho_X}}^2 \\
&\qquad \le \widetilde{C}_3 \log^3(18/\delta) \left(\log(1/\delta) + \log(J(\tau,p,1)) + \log(1 + J(\tau,p,q))\right)^{\max\left\{\frac{6p}{q(2+p)}, 3q\right\}} m^{-\widetilde{\Theta}_1},
\end{aligned}$$

where $\widetilde{C}_3 = 2\tilde{c}_{K,p} M^2 \widetilde{C}_2^{\frac{2p}{q(2+p)}} + \widetilde{C}_1' + 2C_2^q$ and

$$\widetilde{\Theta}_1 = \min\left\{\frac{2}{2+p} - \frac{2p}{q(p+2)}\tilde{\alpha}, \frac{2}{2+p} - (1-\beta)\tau, \beta\tau, \tau - \tilde{\alpha}\right\}$$

Note that the measure of the set $\mathcal{W}_q(R) \backslash (V_R^{(q)} \cup V_1')$ is at least $1 - 3\delta$. And under the restrictions on τ, p and q, we have $\widetilde{\Theta}_1 = \tau - \tilde{\alpha}$ and $\max\left\{\frac{6p}{q(2+p)}, 3q\right\} \le 3$. Then the error bound (60) follows by scaling 3δ to δ. Thus we complete the proof. ■

Now we can prove Theorem 1 based on the error bound (58).

Proof [Proof of Theorem 1]. Recall that the function space $\mathscr{H}_{\widetilde{K}}$ is in the range of the integral operator $L_{\widetilde{K}}^{1/2}$. Hence the assumption $f_\rho \in \mathscr{H}_{\widetilde{K}}$ implies the approximation condition (14) is valid with $\beta = 1$ (see Proposition 8.5 in Cucker and Zhou (2007)). The assumption on the input space X ensures that X satisfies an interior cone condition (see Definition 9). Then for an admissible kernel $K \in \mathscr{C}^s$ with $s > 0$, due to the previous result obtained in Shi et al. (2011) and Theorem 11, one can check that the capacity assumption (20) is achieved. Concretely, when $0 < s < 2$, $p = \frac{2d}{d+2\min\{1,s\}}$; when $s \ge 2$, p can be chosen to be any constant satisfying $p > \frac{2d}{d+2\lceil s \rceil}$.

We just focus on the case $s \ge 2$. For any $0 < \epsilon \le \Theta - \frac{1}{2}$ with Θ given by (5), the capacity assumption is achieved for $p > \frac{2d}{d+2\lceil s \rceil}$ satisfying $\frac{2}{2+p} = \Theta - \frac{\epsilon}{2}$. We thus set $\tau = \Theta - \epsilon$. Next we need to bound the quantity $J(\tau,p,1)$. In fact, as $\frac{(2-(p+2)\tau)p}{(1-p\tau)(p+2)} = \frac{1-\frac{p+2}{2}\tau}{1-p\tau} \cdot \frac{2p}{p+2}$, we further have

$$\frac{\log \frac{(2-(p+2)\tau)p}{(1-p\tau)(p+2)}}{\log \frac{2p}{p+2}} = \frac{\log \frac{1-p\tau}{1-\frac{p+2}{2}\tau}}{\log \frac{p+2}{2p}} + 1.$$

Then we substitute $p = \frac{2}{\Theta - \epsilon/2} - 2$ and $\tau = \Theta - \epsilon$ into the above equation. Combining the restrictions for Θ and ϵ, we obtain

$$\frac{\log \frac{(2-(p+2)\tau)p}{(1-p\tau)(p+2)}}{\log \frac{2p}{p+2}} \le \frac{\log \frac{4}{\epsilon(1-\epsilon)}}{\log \frac{1}{1-\epsilon}}$$

and

$$J(\tau,p,1) = \max\left\{2, \frac{\log \frac{(2-(p+2)\tau)p}{(1-p\tau)(p+2)}}{\log \frac{2p}{p+2}}\right\} \le \frac{\log \frac{4}{\epsilon(1-\epsilon)}}{\log \frac{1}{1-\epsilon}} := J_\epsilon.$$

Now recall the general bound (58). Since $\beta = 1$, one can check that $\widetilde{\Theta} = \tau = \Theta - \epsilon$ and the bound (6) is valid with $C_\epsilon = C_3(J_\epsilon + 1)^6$. As the error bound for the case $0 < s < 2$ can be derived following the same way, we thus complete the proof of Theorem 1. ■

5. Sparsity analysis

In this section, we shall derive an asymptotical upper bound on $\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m}$, where $\mathbf{c}_q^{\mathbf{z}}$ denotes the coefficient sequence of the estimator $\hat{f}_q$ with $0 < q < 1$. In order to do so, we need a lower bound on the value of the non-zero elements in $\mathbf{c}_q^{\mathbf{z}}$. Recall that for any vector $\mathbf{c} = (c_1, \cdots, c_m) \in \mathbb{R}^m$, the support set of $\mathbf{c}$ is given by $\mathrm{supp}(\mathbf{c}) := \{j \in \{1, \cdots, m\} : c_j \ne 0\}$.

Proposition 18 *Let $I \subseteq \{1, 2, \cdots, m\}$ be a non-empty index set, $0 < q < 1$ and $\mathbf{c}_q^* = (c_{q,1}^*, \cdots, c_{q,m}^*)$ be a local minimizer of the following optimization problem*

$$\min_{\mathbf{c} \in \mathbb{R}^m, supp(\mathbf{c}) \subseteq I} \left\{\frac{1}{m}\sum_{j=1}^m \left(y_j - \sum_{i=1}^m c_i K(x_j, x_i)\right)^2 + \gamma \|\mathbf{c}\|_q^q\right\}. \tag{61}$$

Then for $i \in supp(\mathbf{c}_q^*)$, there holds

$$|c_{q,i}^*| > \left(\frac{q(1-q)}{2\kappa^2}\right)^{1/(2-q)} \gamma^{1/(2-q)}, \tag{62}$$

where $\kappa = \|K\|_{\mathscr{C}(X \times X)}$. Moreover, if $\mathbf{c}_q^*$ is a global minimizer of the optimization problem (61), the above bound can be improved to

$$|c_{q,i}^*| \ge \left(\frac{1-q}{\kappa^2}\right)^{1/(2-q)} \gamma^{1/(2-q)}. \tag{63}$$

Proof Recall the target functional (3) to be optimized with $\mathbf{c} = (c_1, \cdots, c_m) \in \mathbb{R}^m$, which can be expressed as

$$\mathscr{T}_{\gamma,q}(\mathbf{c}) = \frac{1}{m}\sum_{j=1}^m \left(y_j - \sum_{i=1}^m c_i K(x_j, x_i)\right)^2 + \gamma \sum_{i=1}^m |c_i|^q.$$

For $i \in \mathrm{supp}(\mathbf{c}_q^*)$, we define an univariate function as

$$h_i(t) = \mathscr{T}_{\gamma,q}(\mathbf{c}_q^{*\backslash i}(t)), \qquad \forall t \in \mathbb{R},$$

where $\mathbf{c}_q^{*\backslash i}(t) = (c_{q,1}^*, \cdots, c_{q,i-1}^*, t, c_{q,i+1}^*, \cdots, c_{q,m}^*)$. As $\mathbf{c}_q^*$ is a local minimizer of the optimization problem (61), thus $c_{q,i}^*$ is a local minimizer of $h_i(t)$. We compute the first and the

second derivative of $h_i(t)$ on the interval $(c^*_{q,i}-\varrho, c^*_{q,i}+\varrho)$ for some $\varrho>0$. Since $c^*_{q,i}\neq 0$, we can choose a sufficiently small ϱ such that 0 is not contained in the interval $(\hat{c}_i-\varrho, \hat{c}_i+\varrho)$. Then we obtain

$$h_i^{'}(t)=\frac{2}{m}\sum_{j=1}^{m}\left(\sum_{k\neq i}c^*_{q,k}K(x_j,x_k)+tK(x_j,x_i)-y_j\right)K(x_j,x_i)+\gamma q\mathrm{sgn}(t)|t|^{q-1}$$

and

$$h_i^{''}(t)=\frac{2}{m}\sum_{j=1}^{m}K^2(x_i,x_j)-\gamma q(1-q)|t|^{q-2}.$$

Recall that $c^*_{q,i}$ is a local minimizer of $h_i(t)$. Due to the optimality condition, we must have $h_i^{'}(c^*_{q,i})=0$ and $h_i^{''}(c^*_{q,i})>0$, i.e.,

$$\frac{2}{m}\sum_{j=1}^{m}\left(\sum_{k=1}^{m}c^*_{q,k}K(x_j,x_k)-y_j\right)K(x_j,x_i)+\gamma q\mathrm{sgn}(c^*_{q,i})|c^*_{q,i}|^{q-1}=0 \tag{64}$$

and

$$\frac{2}{m}\sum_{j=1}^{m}K^2(x_j,x_i)-\gamma q(1-q)|c^*_{q,i}|^{q-2}>0. \tag{65}$$

From the inequality (65), we have

$$\gamma q(1-q)|c^*_{q,i}|^{q-2}<\frac{2}{m}\sum_{j=1}^{m}K^2(x_j,x_i)\leq 2\kappa^2,$$

which leads to bound (62).

If $\mathbf{c}^*_q$ is the global minimizer of the optimization problem (61), the global optimality of $\mathbf{c}^*_q$ implies $c^*_{q,i}$ is a global minimizer of $h_i(t)$. We thus have $h_i^{'}(c^*_{q,i})=0$. Due to equation (64), there holds

$$\frac{2}{m}\sum_{j=1}^{m}\left(y_j-\sum_{k=1}^{m}c^*_{q,k}K(x_j,x_k)\right)K(x_j,x_i)=\gamma q\mathrm{sgn}(c^*_{q,i})|c^*_{q,i}|^{q-1}.$$

We multiply both sides of the above equation by $c^*_{q,i}$ and find that

$$\frac{2}{m}\sum_{j=1}^{m}\left(y_j-\sum_{k=1}^{m}c^*_{q,k}K(x_j,x_k)\right)K(x_j,x_i)c^*_{q,i}=\gamma q|c^*_{q,i}|^{q}. \tag{66}$$

Now we consider a new vector given by

$$\mathbf{c}_q^{*\backslash i}(0)=(c^*_{q,1},\cdots,c^*_{q,i-1},0,c^*_{q,i+1},\cdots,c^*_{q,m}).$$

We compare $\mathscr{T}_{\gamma,q}(\mathbf{c}_q^{*\backslash i}(0))$ with $\mathscr{T}_{\gamma,q}(\mathbf{c}^*_q)$ and find that

$$\begin{aligned}
&\mathscr{T}_{\gamma,q}(\mathbf{c}^*_q)-\mathscr{T}_{\gamma,q}(\mathbf{c}_q^{*\backslash i}(0))\\
&=-\frac{1}{m}\sum_{j=1}^{m}\left\{2y_j-2\sum_{k=1}^{m}c^*_{q,k}K(x_j,x_k)+c^*_{q,i}K(x_j,x_i)\right\}c^*_{q,i}K(x_j,x_i)+\gamma|c^*_{q,i}|^q\\
&=\frac{2}{m}\sum_{j=1}^{m}\left\{\sum_{k=1}^{m}c^*_{q,k}K(x_j,x_k)-y_j\right\}K(x_j,x_i)c^*_{q,i}-\frac{1}{m}\sum_{j=1}^{m}K^2(x_j,x_i)(c^*_{q,i})^2+\gamma|c^*_{q,i}|^q.
\end{aligned}$$

From the equality (66), it follows that

$$\begin{aligned}
\mathscr{T}_{\gamma,q}(\mathbf{c}^*_q)-\mathscr{T}_{\gamma,q}(\mathbf{c}_q^{*\backslash i}(0)) &= -\gamma q|c^*_{q,i}|^q-\frac{1}{m}\sum_{j=1}^{m}K^2(x_j,x_i)(c^*_{q,i})^2+\gamma|c^*_{q,i}|^q\\
&\geq \gamma(1-q)|c^*_{q,i}|^q-\kappa^2(c^*_{q,i})^2.
\end{aligned}$$

Recall that $\mathbf{c}^*_q$ is a global minimizer of problem (61). The above inequality implies $\gamma(1-q)|c^*_{q,i}|^q-\kappa^2(c^*_{q,i})^2\leq 0$, which leads to the lower bound (63). Thus we complete our proof. ■

Remark 19 *We use the second order optimality condition to derive the lower bound on the non-zero coefficients of a local minimizer in (61). Based on the same idea, a lower bound estimation which is similar to (62) has been derived in Chen et al. (2010). However, our analysis gives another lower bound when the solution is a global minimizer, which indicates that the global optimality will help to improve the sparseness of the solutions.*

It should be noticed that for a given kernel function K, the lower bounds presented in Proposition 18 only depend on γ and q. Thus we can use these bounds to obtain some universal results. When $\mathbf{c}^{\mathbf{z}}_q$ is a local minimizer of optimization problem (2) satisfying Assumption 1, which implies that $\mathbf{c}^{\mathbf{z}}_q$ satisfies the condition of Proposition 18 for $I=\{1,\cdots,m\}$. We can derive the upper bound on $\frac{\|\mathbf{c}^{\mathbf{z}}_q\|_0}{m}$ based on the lower bound (62).

Theorem 20 *Assume that approximation assumption (14) with $0<\beta\leq 1$ and capacity condition (20) with $0<p<2$ are valid, and the estimator $\hat{f}_q$ satisfies Assumption 1 with*

$$\max\left\{\frac{2p}{p+2},\frac{p}{p+2\beta}\right\}<q<1.$$

Let $(\delta,\tau)\in(0,1)^2$ and $J(\tau,p,q)$ be a constant given by (47). Take $\gamma=m^{-\tau}$ with $\frac{1-q}{1-q(1-\beta)}<\tau<\frac{2}{2+p}$. Then with confidence $1-\delta$, there holds

$$\frac{\|\mathbf{c}^{\mathbf{z}}_q\|_0}{m}\leq C_q\left(\log J(\tau,p,1)\right)^{3q}\left(\log(1/\delta)+\log(1+J(\tau,p,q))\right)^3 m^{q\left(\frac{\tau}{2-q}-1+(1-\beta)\tau\right)}, \tag{67}$$

where $\mathbf{c}^{\mathbf{z}}_q$ denotes the coefficient sequence of the estimator $\hat{f}_q$ and C_q is a constant positive constants depending on $q,\kappa,\tilde{c}_{K,p}$, M and c_β.

Proof From Proposition 18, if $\mathbf{c}_q^{\mathbf{z}}$ is a local minimizer of problem (2), then for $i \in \mathrm{supp}(\mathbf{c}_q^{\mathbf{z}})$, there holds

$$|c_{q,i}^{\mathbf{z}}| > \left(\frac{q(1-q)}{2\kappa^2}\right)^{1/(2-q)} \gamma^{1/(2-q)}.$$

Equivalently, we have

$$1 < \left(\frac{q(1-q)}{2\kappa^2}\right)^{-q/(2-q)} \gamma^{-q/(2-q)} |c_{q,i}^{\mathbf{z}}|^q.$$

Note that the above inequality holds true for every $c_{q,i}^{\mathbf{z}}$ with $i \in \mathrm{supp}(\mathbf{c}_q^{\mathbf{z}})$. Therefore, we take a summation on both sides of the inequality according to $i \in \mathrm{supp}(\mathbf{c}_q^{\mathbf{z}})$ and find that

$$\|\mathbf{c}_q^{\mathbf{z}}\|_0 < \left(\frac{q(1-q)}{2\kappa^2}\right)^{-q/(2-q)} \gamma^{-q/(2-q)} \|\mathbf{c}_q^{\mathbf{z}}\|_q^q.$$

Now taking $\gamma = m^{-\tau}$ with $\frac{1-q}{1-q(1-\beta)} < \tau < \frac{2}{2+p}$ and recalling the bound on $\|\mathbf{c}_q^{\mathbf{z}}\|_q^q$ given by (49), then the desired result is obtained with $C_q = \widetilde{C}_2 \left(\frac{q(1-q)}{2\kappa^2}\right)^{-q/(2-q)}$. Thus we complete the proof. ■

If $\mathbf{c}_q^{\mathbf{z}}$ is a global minimizer of algorithm (2), one can derive an upper bound on $\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m}$ from (63) following the same method. From the upper bound (67), we can see that if $\tau < \frac{2-q}{1+(2-q)(1-\beta)}$, the quantity $\frac{\|\mathbf{c}_q^{\mathbf{z}}\|_0}{m}$ converges to 0 at a rate of polynomial decay as m tends to infinity. At the end of this section, we give the proof of Theorem 3.

Proof [Proof of Theorem 3]. From the proof of Theorem 1, we know that under the assumption of Theorem 3, the approximation condition (14) is valid with $\beta = 1$ and the capacity assumption (20) can be satisfied with an arbitrarily small $p > 0$. We derive our conclusions based on Theorem 17 and Theorem 20. We first check the restrictions for q and τ. As $\beta = 1$, we thus find $1 - q < \tau < \frac{2}{p+2}$ and $\frac{2p}{2+p} < q < 1$. Note that p can be arbitrarily closed to 0. Therefore, we can take τ and q to be any value in the interval $(0, 1)$ by choosing a small enough p. By the same argument, we can bound $J(\tau, p, q)$ by $\log \frac{4}{q(1-\tau)}$ for $1 - q < \tau < 1$ and $0 < q \leq 1$. Therefore, from bound (60), the inequality (7) holds with confidence $1 - \delta$ and $\widetilde{C} = 8\widetilde{C}_3$. Similarly, as a direct corollary of the bound (67), the inequality (8) holds with confidence $1 - \delta$ and $\widetilde{C}' = \widetilde{C}_2(1 + 2\kappa^2)$. Finally, if the two bounds hold together, we need to scale 2δ to δ. Thus we complete the proof. ■

6. Conclusion

In this paper, we investigate the sparse kernel regression with ℓ_q-regularization, where $0 < q \leq 1$. The data dependence nature of the kernel-based hypothesis space provides flexibility for this algorithm. The regularization scheme is essentially different from the standard one in an RKHS: the kernel is not necessarily symmetric or positive semi-definite and the regularizer is the ℓ_q-norm of a function expansion involving samples. When the underlying hypothesis space is finite-dimensional, the ℓ_q-regularization with $0 < q \leq 1$ is well understood in theory and widely used in various applications such as compressed sensing

and sparse recovery. However, its role in the non-parametric regression within an infinite-dimensional hypothesis space has not been fully understood yet. In this paper, we develop the mathematical analysis for the asymptotic convergence and sparseness of ℓ_q-regularized kernel regression. We first present a tight bound on the ℓ_2-empirical covering numbers of the kernel-based hypothesis space under ℓ_1-constrain, which is interesting on its own right. We thus demonstrate that, compared with classical RKHS, the hypothesis space involved in the error analysis induced by the non-symmetric kernel has nice behaviors in terms of the ℓ^2-empirical covering numbers of its unit ball. Moreover, the empirical covering number estimates developed in this paper can also be applied to obtain distribution-free error analysis for other sparse approximation schemes, for example Guo, Fan and Zhou (2016); Guo et al. (2017). Our theoretical analysis is based on the concentration estimates with ℓ^2-empirical covering numbers, a refined iteration technique for ℓ_q-regularization and a descent-iterative minimization process which can be realized by the ℓ_q-threshholding function. Based on our analysis, we show that the ℓ_q-regularization term plays a role as a trade-off between sparsity and convergence rates. We also prove that regularizing the combinatorial coefficients by the ℓ_q-norm can produce strong sparse solutions, i.e., the fraction of non-zero coefficients converges to 0 at a polynomial rate when the sample size m becomes large. Our mathematical analysis established in this paper can shed some lights on understanding the role of the ℓ_q-regularization in feature selections in an infinite-dimensional hypothesis space.

Acknowledgments

We thank the anonymous referees for their constructive suggestions. The work described in this paper is supported in part by the National Science Foundation of China (Grants No.11571078, 11631015, 61603248, 61977046); the Joint Research Fund by National Natural Science Foundation of China and Research Grants Council of Hong Kong (Project No. 11461161006 and Project No. CityU 104012); Program of Shanghai Subject Chief Scientist (Project No.18XD1400700). The research leading to these results has also received funding from the European Research Council ERC AdG A-DATADRIVE-B (290923) and ERC AdG E-DUALITY (787960) under the European Union's Horizon 2020 research and innovation programme. The corresponding author is Xiaolin Huang.

Appendix A.

This appendix includes some detailed proofs.

A.1. Proof of Lemma 4

In this subsection, we shall prove Lemma 4.

Proof [Proof of Lemma 4]. It is easy to verify that the vector $\Psi_{\eta,q}(\mathbf{d})$ is a global minimizer of the problem (12) if and only if $(\Psi_{\eta,q}(\mathbf{d}))_i$ is a global minimizer of $\min_{c\in\mathbb{R}} \left\{|c - d_i|^2 + \eta|c|^q\right\}$, where $(\Psi_{\eta,q}(\mathbf{d}))_i$ denotes the $i-$th coordinate value of the vector $\Psi_{\eta,q}(\mathbf{d})$. In the following proof, we shall use x to denote a given constant. In order to prove our conclusion, we need to find the global minimizer of the univariate function $h(t) = t^2 - 2xt + \eta|t|^q$, i.e., the solution

of the minimization problem $\min_{t\in\mathbb{R}}\left\{|t-x|^2+\eta|t|^q\right\}$. Additionally, to ensure the well-definedness of the map $\Psi_{\eta,q}$, we also need to show that under the restriction $|x|>a_q\eta^{1/(2-q)}$, the equation (10) has only one solution on the interval $[(q(1-q)\eta/2)^{1/(2-q)},\infty)$.

When $x=0$, as $\eta>0$, the function $h(t)$ achieves its minimum at $t=0$.

We first limit our discussion to the case $x>0$. In this case, the function h is strictly decreasing on $(-\infty,0]$, hence all its possible minimizers are achieved on $[0,\infty)$. Let us consider the difference between $h(t)$ and $h(0)$ for $t>0$, i.e., $h(t)-h(0)=t(t-2x+\eta t^{q-1})$. It is noticed that the function $g(t)=t-2x+\eta t^{q-1}$ is continuously differentiable on $(0,\infty)$ and $\lim_{t\to 0+}g(t)=\lim_{t\to\infty}g(t)=\infty$. So we take its derivative on $(0,\infty)$ and find that its unique minimizer is given by $t_1=((1-q)\eta)^{1/(2-q)}$. It follows that $g(t)\geq g(t_1)=2a_q\eta^{\frac{1}{2-q}}-2x$ for all $t>0$.

When $0<x\leq a_q\eta^{1/(2-q)}$, then $\min_{t>0}g(t)=g(t_1)\geq 0$ which implies $h(t)-h(0)=tg(t)\geq 0$ for $t>0$. Hence, $t=0$ is a global minimizer of h in this case.

And then we consider the case $x>a_q\eta^{1/(2-q)}$. One may see that $t=0$ is not a global minimizer of h, because $g(t_1)<0$ in this case, which implies $h(t_1)<h(0)$. In addition, we have $\lim_{t\to\infty}h(t)=\infty$, thus the function $h(t)$ achieves its minimum on the interval $(0,\infty)$. We claim that this minimizer is given by t_2, where t_2 is the solution of the equation $h^{'}(t)=0$ on the interval $[(q(1-q)\eta/2)^{1/(2-q)},\infty)$. It should be noticed that $h^{'}(t)=0$ is exactly the equation given by (10) with $x>0$. We thus consider the second derivative of h given by $h^{''}(t)=2+\eta q(q-1)t^{q-2}$. We first prove the existence and uniqueness of t_2. In fact, a direct computation shows that

$$\begin{aligned}
h^{'}((q(1-q)\eta/2)^{1/(2-q)}) &= 2(\eta q)^{\frac{1}{2-q}}\left(\frac{1-q}{2}\right)^{\frac{1}{2-q}}+(\eta q)^{\frac{1}{2-q}}\left(\frac{1-q}{2}\right)^{\frac{q-1}{2-q}}-2x\\
&= (2-q)\left(\frac{1-q}{2}\right)^{\frac{q-1}{2-q}}(\eta q)^{\frac{1}{2-q}}-2x\\
&\leq (2-q)\left(\frac{1-q}{2}\right)^{\frac{q-1}{2-q}}q^{\frac{1}{2-q}}a_q^{-1}x-2x\\
&= \left(2^{\frac{3-2q}{2-q}}q^{\frac{1}{2-q}}-2\right)x<0,
\end{aligned}$$

the first inequality is from $x>a_q\eta^{1/(2-q)}$ and the last inequality holds as $q<2^{q-1}$ for $0<q<1$. We also observe that $h^{''}(t)\geq 0$ on $[(q(1-q)\eta/2)^{1/(2-q)},\infty)$, which implies $h^{'}(t)$ is strictly increasing on this interval. Since $h^{'}(t)$ is continuous on $(0,\infty)$ and $\lim_{t\to\infty}h^{'}(t)=\infty$, the equation $h^{'}(t)=0$ has a unique solution t_2 on $[(q(1-q)\eta/2)^{1/(2-q)},\infty)$. Because $h^{'}(t_2)=0$ and $h^{''}(t_2)>0$, we also conclude that t_2 is the only minimizer of $h(t)$ on $[(q(1-q)\eta/2)^{1/(2-q)},\infty)$. We further prove that t_2 is actually the minimizer of $h(t)$ on $(0,\infty)$. We just need to show $h(t)$ has no local minimizer on $(0,(q(1-q)\eta/2)^{1/(2-q)})$. This conclusion can be easily drawn from the fact that $h^{''}(t)<0$ on $(0,(q(1-q)\eta/2)^{1/(2-q)})$.

When $x<0$, one can easily find that $h(t)$ achieves its minimum on $(-\infty,0]$. Then for $t\in(-\infty,0]$, we can rewrite the function $h(t)$ as $(-t)^2+2x(-t)+\eta(-t)^q$. Due to the same analysis as above, we find that, when $-a_q\eta^{1/(2-q)}\leq x<0$, the global minimizer of $h(t)$ is $t=0$; when $x<-a_q\eta^{1/(2-q)}$, the minimizer of $h(t)$ is given by the unique solution of the equation $2(-t)+\eta q(-t)^{q-1}+2x=0$ on the interval $(-\infty,-(q(1-q)\eta/2)^{1/(2-q)}]$.

Finally, we combine all the cases together and find that for a given $x\in\mathbb{R}$, a global minimizer of $h(t)$ is $\psi_{\eta,q}(x)$ given by (9). Hence according to our analysis, if we evaluate $(\Psi_{\eta,q}(\mathbf{d}))_i$ as (11), the vector $\Psi_{\eta,q}(\mathbf{d})$ is a global minimizer of the optimization problem (12).

It is also easy to check that when $|x|=a_q\eta^{1/(2-q)}$, the univariate function $h(t)$ has two global minimizers given by 0 and $\frac{2-2q}{2-q}|x|$ respectively. Thus the vector $\Psi_{\eta,q}(\mathbf{d})$ indeed gives one global minimizer of problem (12). Finally we complete the proof. ■

Remark 21 *A similar discussion about the global minimizer of $h(t)$ can be found in Knight and Fu (2000).*

A.2. Proof of Proposition 5

In this subsection, we shall prove Proposition 5.

Proof [Proof of Proposition 5]. We first prove conclusion (i). Recall that $\mathbf{c}^*=(c_1^*,\cdots,c_m^*)\in\mathbb{R}^m$ is a local minimizer of the objective functional $\mathscr{T}_{\gamma,q}(\mathbf{c})$ defined by (3) and $0<\lambda\leq\frac{q}{2}\|\mathbb{K}\|_2^{-2}$. It is sufficient to prove that for $i\in\mathrm{supp}(\mathbf{c}^*)$, i.e., $c_i^*\neq 0$, there holds

$$\left|\left(\mathbf{c}^*+\lambda\mathbb{K}^T(\mathbf{y}-\mathbb{K}\mathbf{c}^*)\right)_i\right|>a_q(\lambda\gamma)^{1/(2-q)}$$

and $c_i^*=\mathrm{sgn}\left(\left(\mathbf{c}^*+\lambda\mathbb{K}^T(\mathbf{y}-\mathbb{K}\mathbf{c}^*)\right)_i\right)t_i^*$, where t_i^* is the solution of the equation

$$2t+\lambda\gamma qt^{q-1}-2\left|\left(\mathbf{c}^*+\lambda\mathbb{K}^T(\mathbf{y}-\mathbb{K}\mathbf{c}^*)\right)_i\right|=0$$

on the interval $[(q(1-q)\lambda\gamma/2)^{1/(2-q)},\infty)$. Here $(\cdot)_i$ denotes the $i-$th coordinate value of a vector.

By the same argument in the proof of Proposition 18, for $i\in\mathrm{supp}(\mathbf{c}^*)$, we have

$$2\left(\mathbb{K}^T(\mathbb{K}\mathbf{c}^*-\mathbf{y})\right)_i+\gamma q\mathrm{sgn}(c_i^*)|c_i^*|^{q-1}=0 \tag{68}$$

and

$$2\sum_{j=1}^m\mathbb{K}_{j,i}^2>\gamma q(1-q)|c_i^*|^{q-2}, \tag{69}$$

where $\mathbb{K}_{j,i}$ denotes the $(j,i)-$th entry of the matrix $\mathbb{K}$. We multiply both sides of the inequality (69) by a positive λ. Since $\lambda\leq\frac{q}{2}\|\mathbb{K}\|_2^{-2}$, we obtain

$$\lambda\gamma q(1-q)|c_i^*|^{q-2}<2\lambda\sum_{j=1}^m\mathbb{K}_{j,i}^2\leq q,$$

which implies $|c_i^*|>((1-q)\lambda\gamma)^{1/(2-q)}$.

We consider the case $c_i^*>0$. By equation (68), for $c_i^*>0$, there holds

$$2c_i^*+\lambda\gamma q(c_i^*)^{q-1}=2\left(\mathbf{c}^*+\lambda\mathbb{K}^T(\mathbf{y}-\mathbb{K}\mathbf{c}^*)\right)_i,$$

which verifies that $\left(\mathbf{c}^*+\lambda\mathbb{K}^T(\mathbf{y}-\mathbb{K}\mathbf{c}^*)\right)_i$ is positive and c_i^* is the zero point of

$$f(t)=2t+\lambda\gamma qt^{q-1}-2\left(\mathbf{c}^*+\lambda\mathbb{K}^T(\mathbf{y}-\mathbb{K}\mathbf{c}^*)\right)_i.$$

Note that $c_i^* > ((1-q)\lambda\gamma)^{1/(2-q)} > (q(1-q)\lambda\gamma/2)^{1/(2-q)}$. Recalling the analysis in Lemma 4, we know that $f(t)$ is monotonically increasing on the interval $[(q(1-q)\lambda\gamma/2)^{1/(2-q)}, \infty)$. We thus have $f((\lambda\gamma(1-q))^{1/(2-q)}) < 0$ which implies

$$\left(\mathbf{c}^* + \lambda\mathbb{K}^T(\mathbf{y} - \mathbb{K}\mathbf{c}^*)\right)_i > a_q(\lambda\gamma)^{1/(2-q)}.$$

Therefore, we verify conclusion (i) for $c_i^* > 0$. When $c_i^* < 0$, we can prove it following the same method.

Next, we shall prove the function $\psi_{\eta,q}(x)$ defined by (9) is Lipschitz continuous and strictly increasing for any $|x| > a_q\eta^{1/(2-q)}$. Then one can check that the proof of the rest two conclusions are almost the same as the proof of Theorem 3 in Xu et al. (2012). Since $\psi_{\eta,q}(x)$ is an odd function, we only need to prove our desired result for $x > a_q\eta^{1/(2-q)}$. From (9), when $x > a_q\eta^{1/(2-q)}$, $\psi_{\eta,q}(x)$ is defined to be the solution of the equation $2t+\eta q t^{q-1}-2x = 0$ on the interval $[(q(1-q)\eta/2)^{1/(2-q)}, \infty)$. Since the bivariate function $F(t,x) = 2t+\eta q t^{q-1} - 2x$ is continuously differentiable inside $[(q(1-q)\eta/2)^{1/(2-q)}, \infty) \times (a_q\eta^{1/(2-q)}, \infty)$, we have $\psi'_{\eta,q}(x) = \frac{2}{2-\eta q(1-q)(\psi_{\eta,q}(x))^{q-2}}$ due to the implicit function theorem. Obviously $\psi'_{\eta,q}(x) > 1$ which implies $\psi_{\eta,q}(x)$ is strictly increasing. In order to verify the Lipschitz continuity, we still need an upper bound for $\psi'_{\eta,q}(x)$. To this end, we show that $\psi_{\eta,q}(x) > (q(1-q)\eta)^{1/(2-q)}$ for $x > a_q\eta^{1/(2-q)}$. Also following the analysis of Lemma 4, it is equivalent to check that when $x > a_q\eta^{1/(2-q)}$, there holds $2\left(q(1-q)\eta\right)^{\frac{1}{2-q}} + \eta q\left(q(1-q)\eta\right)^{\frac{q-1}{2-q}} - 2x < 0$. Since when $x > a_q\eta^{1/(2-q)}$, a simple calculation show that

$$2\left(q(1-q)\eta\right)^{\frac{1}{2-q}} + \eta q\left(q(1-q)\eta\right)^{\frac{q-1}{2-q}} - 2x < 2a_q\eta^{1/(2-q)}\left(\frac{3-2q}{2-q}q^{1/(2-q)} - 1\right).$$

The left hand side of the above inequality can be further bounded by 0 as $\frac{3-2q}{2-q}q^{1/(2-q)} < 1$ for $0 < q < 1$. Hence $\psi_{\eta,q}(x) > (q(1-q)\eta)^{1/(2-q)}$ for $x > a_q\eta^{1/(2-q)}$, which implies $\psi'_{\eta,q}(x) < 2$. We thus verify the Lipschitz continuity of $\psi_{\eta,q}(x)$ for $x > a_q\eta^{1/(2-q)}$ and the proof is completed. ■

A.3. Proof of Proposition 6

In this subsection, we shall prove Proposition 6.

Proof [Proof of Proposition 6]. We just prove the inequality (16), the bound (17) can be derived by the same approach. Recall the definition of the projection operator π_M (see Definition 2). As $\rho(\cdot|x)$ is supported on $[-M, M]$ at every $x \in X$, it obviously implies $|y_i| \le M$ for $i = 1, \cdots, m$ and $\mathscr{E}_{\mathbf{z}}(\pi_M(\hat{f}_q)) \le \mathscr{E}_{\mathbf{z}}(\hat{f}_q)$.

When $0 < q < 1$, since the estimator $\hat{f}_q$ satisfies Assumption 1, we have

$$\mathscr{E}_{\mathbf{z}}(\hat{f}_q) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \le \mathscr{E}_{\mathbf{z}}(\hat{f}_1) + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_q^q.$$

Therefore,

$$\begin{aligned}\mathscr{E}_{\mathbf{z}}(\pi_M(\hat{f}_q)) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q &\le \mathscr{E}_{\mathbf{z}}(\hat{f}_q) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \\ &\le \mathscr{E}_{\mathbf{z}}(\hat{f}_1) + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_q^q \\ &\le \mathscr{E}_{\mathbf{z}}(\hat{f}_1) + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_1 + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_q^q. \end{aligned} \tag{70}$$

Note that the coefficient sequence of $f_{\mathbf{z},\gamma}$ is given by $\{\frac{1}{m}g_\gamma(x_i)\}_{i=1}^m$ and $\hat{f}_1$ is the global minimizer of problem (2) at $q = 1$. Hence, there holds

$$\mathscr{E}_{\mathbf{z}}(\hat{f}_1) + \gamma\|\mathbf{c}_1^{\mathbf{z}}\|_1 \le \mathscr{E}_{\mathbf{z}}(f_{\mathbf{z},\gamma}) + \gamma m^{-1}\sum_{i=1}^m |g_\gamma(x_i)|. \tag{71}$$

A direct computation shows that

$$\begin{aligned}&\mathscr{E}(\pi_M(\hat{f}_q)) - \mathscr{E}(f_\rho) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q \\ &= \{\mathscr{E}(\pi_M(\hat{f}_q)) - \mathscr{E}_{\mathbf{z}}(\pi_M(\hat{f}_q))\} + \{\mathscr{E}_{\mathbf{z}}(\pi_M(\hat{f}_q)) + \gamma\|\mathbf{c}_q^{\mathbf{z}}\|_q^q - \mathscr{E}_{\mathbf{z}}(f_{\mathbf{z},\gamma}) - \gamma m^{-1}\sum_{i=1}^m |g_\gamma(x_i)|\} \\ &+\{\mathscr{E}_{\mathbf{z}}(f_{\mathbf{z},\gamma}) - \mathscr{E}(f_{\mathbf{z},\gamma})\} + \{\gamma m^{-1}\sum_{i=1}^m |g_\gamma(x_i)| - \gamma\|g_\gamma\|_{L^1_{\rho_X}}\} + \{\gamma\|g_\gamma\|_{L^1_{\rho_X}} - \gamma\|g_\gamma\|_{L^2_{\rho_X}}\} \\ &+\{\mathscr{E}(f_{\mathbf{z},\gamma}) - \mathscr{E}(f_\gamma)\} + \{\mathscr{E}(f_\gamma) - \mathscr{E}(f_\rho) + \gamma\|g_\gamma\|_{L^2_{\rho_X}}\}.\end{aligned}$$

From inequalities (70) and (71), the second term on the right hand side of the above equation is at most $\gamma\|\mathbf{c}_1^{\mathbf{z}}\|_q^q$, which can be further bounded by $\gamma m^{1-q}\|\mathbf{c}_1^{\mathbf{z}}\|_1^q$ due to the reverse Hölder inequality. The inequality $\|g_\gamma\|_{L^1_{\rho_X}} \le \|g_\gamma\|_{L^2_{\rho_X}}$ implies the fifth term is at most zero, thus we obtain bound (16). Then the proof is completed. ■

A.4. Proof of Proposition 14

Now we concentrate our efforts on the proof of Proposition 14.

Definition 22 *A function $\psi : \mathbb{R}^+ \to \mathbb{R}^+$ is sub-root if it is non-negative, non-decreasing, and if $\psi(r)/\sqrt{r}$ is non-increasing.*

It is easy to see for a sub-root function ψ and any $D > 0$, the equation $\psi(r) = r/D$ has unique positive solution. Proposition 14 can be proved based on the following lemma, which is given in Blanchard et al. (2008).

Lemma 23 *Let $\mathcal{F}$ be a class of measurable, square-integrable functions such that $\mathbb{E}(f) - f \le b$ for all $f \in \mathcal{F}$. Let ψ be a sub-root function, D be some positive constant and r^* be the unique positive solution of $\psi(r) = r/D$. Assume that*

$$\forall r \ge r^*, \quad \mathbb{E}\left[\max\left\{0, \sup_{f\in\mathcal{F}, \mathbb{E}f^2 \le r}\left(\mathbb{E}(f) - \frac{1}{m}\sum_{i=1}^m f(z_i)\right)\right\}\right] \le \psi(r). \tag{72}$$

Then for all $x > 0$ and all $T > D/7$, with probability at least $1 - e^{-x}$ there holds

$$\mathbb{E}f - \frac{1}{m}\sum_{i=1}^n f(X_i) \le \frac{\mathbb{E}f^2}{D} + \frac{50T}{D^2}r^* + \frac{(T+9b)x}{m}, \quad \forall f \in \mathcal{F}. \tag{73}$$

Proof [Proof of Proposition 14]. We assume that the function g in the concerned function set satisfies $\|g\|_\infty \le B$ and $\mathbb{E}g^2 \le c\mathbb{E}g$ for some $c, B > 0$. In order to prove the conclusion, we define the function set for $R \ge 1$ as

$$\mathcal{G}_R = \left\{g(z) = (\pi_M(f)(x) - y)^2 - (f_\rho(x) - y)^2 : f \in \mathscr{B}_R\right\}.$$

We will apply Lemma 23 to $\mathcal{G}_R$ and find the sub-root function ψ in our setting. To this end, let $\sigma_1, \cdots, \sigma_n$ be the independent *Rademacher random variables*, that is, independent random variables for which $\text{Prob}(\sigma_i = 1) = \text{Prob}(\sigma_i = -1) = 1/2$. One can see Van der Vaart and Wellner (1996) that

$$\mathbb{E}\left[\sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\left|\mathbb{E}g - \frac{1}{m}\sum_{i=1}^m g(z_i)\right|\right] \le 2\mathbb{E}\left[\sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\left|\frac{1}{m}\sum_{i=1}^m \sigma_i g(z_i)\right|\right]. \tag{74}$$

Next, we will bound the right-hand side of the above inequality by using the ℓ_2-empirical covering numbers and entropy integral. Due to Van der Vaart and Wellner (1996), there is an universal absolute constant C such that

$$\frac{1}{\sqrt{m}}\mathbb{E}_\sigma \sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\left|\sum_{i=1}^m \sigma_i g(z_i)\right| \le C\int_0^{\sqrt{V}} \log_2^{\frac{1}{2}} \mathscr{N}_2(\mathcal{G}_R,\nu)d\nu, \tag{75}$$

where $V = \sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\frac{1}{m}\sum_{i=1}^m g^2(z_i)$ and $\mathbb{E}_\sigma$ denotes the expectation with respect to the random variables $\{\sigma_1,\cdots,\sigma_m\}$ conditioned on all of the other random variables. Now we use the capacity condition (20) for $\mathscr{B}_1$. It asserts that

$$\log_2 \mathscr{N}_2(\mathscr{B}_1,\epsilon) \le c_{K,p}\epsilon^{-p}, \qquad \forall\ 0<\epsilon\le 1.$$

For $g_1, g_2 \in \mathcal{G}_R$, we have

$$|g_1(z) - g_2(z)| = |\,(y-\pi_M(f_1)(x))^2 - (y-\pi_M(f_2)(x))^2\,| \le 4M|f_1(x)-f_2(x)|.$$

Therefore, it follows that

$$\mathscr{N}_2\left(\mathcal{G}_R,\epsilon\right) \le \mathscr{N}_2\left(\mathscr{B}_R,\frac{\epsilon}{4M}\right) \le \mathscr{N}_2\left(\mathscr{B}_1,\frac{\epsilon}{4MR}\right) \le \mathscr{N}_2\left(\mathscr{B}_1,\frac{\epsilon}{R_1}\right)$$

where $R_1 = \max\{B, 4MR\}$. Then

$$\log_2 \mathscr{N}_2\left(\mathcal{G}_R,\epsilon\right) \le c_{K,p}R_1^p\epsilon^{-p}, \qquad \forall\ 0<\epsilon\le R_1.$$

Using equation (75), since $V = \sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\frac{1}{m}\sum_{i=1}^m g^2(z_i)$ and $\sqrt{V}\le B\le R_1$, one easily gets

$$\begin{aligned}&\frac{1}{\sqrt{m}}\mathbb{E}_\sigma \sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\left|\sum_{i=1}^m \sigma_i g(z_i)\right| \le Cc_{K,p}^{1/2}R_1^{p/2}\int_0^{\sqrt{V}}\nu^{-p/2}d\nu \\ &= \tilde{c}_{K,p}R_1^{p/2}\left(\sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\frac{1}{m}\sum_{i=1}^m g^2(z_i)\right)^{\frac{1}{2}-\frac{p}{4}},\end{aligned} \tag{76}$$

where $\tilde{c}_{K,p} = 2Cc_{K,p}^{1/2}(2-p)^{-1}$ depending on p and the kernel function K. For simplicity, the constant $\tilde{c}_{K,p}$ may change from line to line in the following derivations. Due to Talagrand (1994), one see that

$$\mathbb{E}\sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\frac{1}{m}\sum_{i=1}^m g^2(z_i) \le \frac{8B}{m}\mathbb{E}\mathbb{E}_\sigma \sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\left|\sum_{i=1}^m \sigma_i g(z_i)\right| + r. \tag{77}$$

Therefore, from (76) and (77), we have

$$\mathcal{R}_m := \frac{1}{\sqrt{m}}\mathbb{E}\mathbb{E}_\sigma \sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\left|\sum_{i=1}^m \sigma_i g(z_i)\right| \le \tilde{c}_{K,p}\left(\frac{\mathcal{R}_m B}{\sqrt{m}} + r\right)^{\frac{1}{2}-\frac{p}{4}} R_1^{p/2}.$$

Solving the above inequality for $\mathcal{R}_m$ and substituting it to the equation (74), we have

$$\begin{aligned}\mathbb{E}\left[\sup_{g\in\mathcal{G}_R,\mathbb{E}g^2\le r}\left|\mathbb{E}g - \frac{1}{m}\sum_{i=1}^m g(z_i)\right|\right] &\le \tilde{c}_{K,p}\max\left\{B^{\frac{2-p}{2+p}}m^{-\frac{2}{2+p}}R_1^{\frac{2p}{2+p}}, r^{\frac{1}{2}-\frac{p}{4}}m^{-\frac{1}{2}}R_1^{\frac{p}{2}}\right\} \\ &\le \tilde{c}_{K,p}R_1^{\frac{2p}{2+p}}\max\left\{B^{\frac{2-p}{2+p}}m^{-\frac{2}{2+p}}, r^{\frac{1}{2}-\frac{p}{4}}m^{-\frac{1}{2}}\right\},\end{aligned}$$

where the last inequality is due to $R_1 \ge 1$ and $0 < p < 2$. According to Lemma 23, one may take $\psi(r)$ to be the right-hand side of the above inequality. Then the solution r^* to the equation $\psi(r) = r/D$ satisfies

$$r^* \le \tilde{c}_{K,p}\max\left\{D^{\frac{4}{2+p}}, DB^{\frac{2-p}{2+p}}\right\}m^{-\frac{2}{2+p}}R_1^{\frac{2p}{2+p}}.$$

Recalling that $\mathbb{E}g^2 \le c\mathbb{E}g$, now we apply Lemma 23 to the function set $\mathcal{G}_R$ by let $T = D = 2c$, then with probability $1-e^{-x}$, there holds

$$\begin{aligned}\mathbb{E}g - \frac{1}{m}\sum_{i=1}^m g(z_i) &\le \frac{1}{2}\mathbb{E}g + \frac{(2c+9b)x}{m} \\ &+\tilde{c}_{K,p}\max\left\{c^{\frac{2-p}{2+p}}, B^{\frac{2-p}{2+p}}\right\}m^{-\frac{2}{2+p}}R_1^{\frac{2p}{2+p}}, \quad \forall g\in\mathcal{G}_R.\end{aligned}$$

For $g\in\mathcal{G}_R$, it is easy to verify that $b = c = 16M^2$, $B = 8M^2$ and $R_1 = \max\{4MR, 8M^2\} \le 8M^2R$. From the above inequality, we obtain

$$\mathbb{E}g - \frac{1}{m}\sum_{i=1}^m g(z_i) \le \frac{1}{2}\mathbb{E}g + \frac{176M^2x}{m} + \tilde{c}_{K,p}M^2m^{-\frac{2}{2+p}}R^{\frac{2p}{2+p}}, \forall g\in\mathcal{G}_R,$$

which is the exactly the inequality given by (37).

Next, we consider the function set defined as

$$\mathcal{G}_R' = \left\{g(z) = (f(x)-y)^2 - (f_\rho(x)-y)^2 : f\in\mathscr{B}_R\right\}.$$

Note that $\mathscr{B}_R$ is a subset of $\mathscr{C}(X)$ and

$$\|f\|_{\mathscr{C}(X)} \le \kappa R, \qquad \forall f\in\mathscr{B}_R,$$

where $\kappa = \|K\|_{\mathscr{C}(X\times X)}$. Therefore, we have $B = c = (3M+\kappa R)^2$ and $b = 2B$ for $g\in\mathcal{G}_R'$. Moreover, $\forall g_1, g_2 \in \mathcal{G}_R'$, there holds

$$\begin{aligned}|g_1(z) - g_2(z)| &= |\,(y-f_1(x))^2 - (y-f_2(x))^2\,| \\ &\le |2y - f_1(x) - f_2(x)||f_1(x)-f_2(x)| \\ &\le 2(M+\kappa R)|f_1(x)-f_2(x)|.\end{aligned}$$

Following the same analysis as above, we find that with probability $1-e^{-x}$,

$$\mathbb{E}g - \frac{1}{m}\sum_{i=1}^{m} g(z_i) \le \frac{1}{2}\mathbb{E}g + \frac{20(3M+\kappa R)^2 x}{m} + \tilde{c}_{K,p}(3M+\kappa)^{\frac{4-2p}{2+p}} R^{\frac{4-2p}{2+p}} m^{-\frac{2}{2+p}} R_2^{\frac{2p}{2+p}}, \quad \forall g \in \mathcal{G}'_R,$$

where $R_2 = \max\left\{(3M+\kappa R)^2, 2(M+\kappa R)R\right\} \le (3M+\kappa)^2 R^2$. Hence, we obtain

$$\mathbb{E}g - \frac{1}{m}\sum_{i=1}^{m} g(z_i) \le \frac{1}{2}\mathbb{E}g + \frac{20(3M+\kappa R)^2 x}{m} + \tilde{c}_{K,p}(3M+\kappa)^2 m^{-\frac{2}{2+p}} R^2, \quad \forall g \in \mathcal{G}'_R,$$

which leads to the inequality (38).

Finally, we can derive the same bounds for $-\mathcal{S}(\pi_M(f))$ and $-\mathcal{S}(f)$ by considering the function set $-\mathcal{G}_R$ and $-\mathcal{G}'_R$. Thus we complete the proof. ■

References

R. Adams and J. Fournier. *Sobolev Spaces*. Academic press, 2003.

T. Ando, S. Konishi, and S. Imoto. Nonlinear regression modeling via regularized radial basis function networks. *Journal of Statistical Planning and Inference*, 138:3616–3633, 2008.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

G. Blanchard, O. Bousquet, and P. Massart. Statistical performance of support vector machines. *Annals of Statistics*, 36:489–531, 2008.

T. Blumensath and M. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14:629–654, 2008.

M. S. Birman and M. Z. Solomyak. Piecewise-polynomial approximations of functions of the classes W_p^α (Russian). *Matematicheskii Sbornik*, 73:331–355, 1967.

E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2006.

E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2008.

R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *Signal Processing Letters, IEEE*, 14:707–710, 2007.

D. Chen, Q. Wu, Y. Ying, and D. X. Zhou. Support vector machine soft margin classifiers: error analysis. *Journal of Machine Learning Research*, 5:1143–1175, 2004.

S. Chen and D. Donoho. Basis pursuit. In *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, volume 1, pages 41–44. IEEE, 1994.

X. Chen, F. Xu, and Y. Ye. Lower bound theory of nonzero entries in solutions of $\ell_2 - \ell_p$ minimization. *SIAM Journal on Scientific Computing*, 32:2832–2852, 2010.

J. B. Conway. *A Course in Operator Theory*. American Mathematical Society, 2000.

F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57:1413–1457, 2004.

D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006.

R. M. Dudley. Universal Donsker classes and metric entropy. *Annals of Probability*, 15:1306–1326, 1987.

D. Edmunds and H. Triebel. *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge University Press, 1996.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

Y. Feng, S. G. Lv, H. Hang, and J. A. K. Suykens. Kernelized elastic net regularization: generalization bounds, and sparse recovery. *Neural Computation*, 28:525–562, 2016.

S. Fischer and I. Steinwart. Sobolev norm learning rates for regularized least-squares algorithm. *arXiv preprint*, arXiv:1702.07254, 2017.

F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10:1455–1480, 1998.

X. Guo, J. Fan, and D. X. Zhou. Sparsity and error analysis of empirical feature-based regularization schemes. *Journal of Machine Learning Research*, 17:3058–3091, 2017.

Z. C. Guo and L. Shi. Learning with coefficient-based regularization and ℓ_1-penalty. *Advances in Computational Mathematics*, 39:493–510, 2013.

Z. C. Guo, D. H. Xiang, X. Guo, and D. X. Zhou. Thresholded spectral algorithms for sparse approximations. *Analysis and Applications*, 15:433–455, 2017.

J. Huang, J. Horowitz, and S. Ma. Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *Annals of Statistics*, 36:587–613, 2008.

X. Huang, A. Maier, J. Hornegger, and J. A. K. Suykens. Indefinite kernels in least squares support vector machines and principal component analysis. *Applied and Computational Harmonic Analysis*, 43:162–172, 2017.

K. Jetter, J. Stöckler, and J. D. Ward. Error estimates for scattered data interpolation on spheres. *Mathematics of Computation*, 68:733–747, 1999.

K. Knight and W. Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, 28:1356–1378, 2000.

M. J. Lai and J. Wang. An unconstrained ℓ_q minimization with $0 < q \leq 1$ for sparse solution of underdetermined linear systems. *SIAM Journal on Optimization*, 21:82–101, 2011.

S. B. Lin, J. Zeng, J. Fang, and Z. Xu. Learning rates of ℓ^q-coefficient regularization learning with gaussian kernel. *Neural Computation*, 26:2350–2378, 2014.

G. Loosli, S. Canu, and C. S. Ong. Learning SVM in Krĕin spaces. *IEEE transactions on pattern analysis and machine intelligence*, 38:1204–1216, 2016.

S. Mendelson and J. Neeman. Regularization in kernel learning. *Annals of Statistics*, 38:526–565, 2010.

C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

B. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24:227–234, 1995.

A. Rakotomamonjy, R. Flamary, G. Gasso, and S. Canu. $\ell_p - \ell_q$ penalty for sparse linear and sparse multiple kernel multi-task learning. *IEEE Transations on Neural Networks*, 22:1307–1320, 2011.

G. Raskutti, M. Wainwright, and B. Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Transactions on Information Theory*, 57:6976–6994, 2011.

V. Roth. The generalized Lasso. *IEEE transactions on Neural Networks*, 15:16–28, 2004.

R. Saad and Ö. Yılmaz. Sparse recovery by non-convex optimization–instance optimality. *Applied and Computational Harmonic Analysis*, 29:30–48, 2010.

F. Schleif and P. Tino. Indefinite proximity learning: a review. *Neural Computation*, 27:2039–2096, 2015.

L. Shi, Y. L. Feng, and D. X. Zhou. Concentration estimates for learning with ℓ^1-regularizer and data dependent hypothesis spaces. *Applied and Computational Harmonic Analysis*, 31:286-302, 2011.

L. Shi. Learning theory estimates for coefficient-based regularized regression. *Applied and Computational Harmonic Analysis*, 34:252–265, 2013.

S. Smale and D. X. Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26:153–172, 2007.

I. Steinwart. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4:1071–1105, 2003.

I. Steinwart and A. Christmann. *Support Vector Machines*. New York: Springer, 2008.

I. Steinwart and A. Christmann. Sparsity of SVMs that use the ϵ-insensitive loss. In *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), 1569–1576, 2009.

I. Steinwart and C. Scovel. Fast rates for support vector machines using Gaussian kernels. *Annals of Statistics*, 35:575–607, 2007.

M. Talagrand. Sharper bounds for Gaussian and empirical processes. *Annals of Probability*, 22:28–76, 1994.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

H. Y. Wang, Q. W. Xiao, and D. X. Zhou. An approximation theory approach to learning with ℓ^1 regularization. *Journal of Approximation Theory*, 167:240–258, 2013.

H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematic*, 4:389–396, 1995.

H. Wendland. Local polynomial reproduction and moving least squares approximation. *IMA Journal of Numercial Analysis*, 21:285–300, 2001.

H. Wendland. *Scattered data approximation*. Cambridge University Press Cambridge, Cambridge, 2005.

Q. Wu, Y. Ying, and D. X. Zhou. Learning rates of least-square regularized regressioin. *Foundations of Computational Mathematics*, 6:171–192, 2006.

Q. Wu and D. X. Zhou. Learning with sample dependent hypothesis spaces. *Computers & Mathematics with Applications*, 56:2896–2907, 2008.

Z. Wu. Compactly supported positive definite radial functions. *Advances in Computational Mathematics*, 4:283–292, 1995.

A. Van der Vaart and J. Wellner. *Weak Convergence and Emprical Processes*. Springer-Verlag, New York, 1996.

Z. Xu, X. Chang, F. Xu, and H. Zhang. $L_{1/2}$ regularization: a thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems*, 23:1013–1027, 2012.

T. Zhang. Leave-one-out bounds for kernel methods. *Neural Computation*, 15:1397–1437, 2003.

D. X. Zhou. Capacity of reproducing kernel spaces in learning theory. *IEEE Transactions on Information Theory*, 49:1743–1752, 2003.

Journal of Machine Learning Research 20 (2019) 1-56 Submitted 3/16; Revised 9/19; Published 10/19

A Kernel Multiple Change-point Algorithm via Model Selection

Sylvain Arlot SYLVAIN.ARLOT@U-PSUD.FR
Université Paris-Saclay, Univ. Paris-Sud, CNRS, Inria, Laboratoire de mathématiques d'Orsay, 91405, Orsay, France.

Alain Celisse CELISSE@MATH.UNIV-LILLE1.FR
Laboratoire de Mathématiques Paul Painlevé
UMR 8524 CNRS-Université Lille 1
MODAL *Project-Team*
F-59 655 Villeneuve d'Ascq Cedex, France

Zaid Harchaoui ZAID@UW.EDU
Department of Statistics
University of Washington
Seattle, WA, USA

Editor: Kenji Fukumizu

Abstract

We consider a general formulation of the multiple change-point problem, in which the data is assumed to belong to a set equipped with a positive semidefinite kernel. We propose a model-selection penalty allowing to select the number of change points in Harchaoui and Cappé's kernel-based change-point detection method. The model-selection penalty generalizes non-asymptotic model-selection penalties for the change-in-mean problem with univariate data. We prove a non-asymptotic oracle inequality for the resulting kernel-based change-point detection method, whatever the unknown number of change points, thanks to a concentration result for Hilbert-space valued random variables which may be of independent interest. Experiments on synthetic and real data illustrate the proposed method, demonstrating its ability to detect subtle changes in the distribution of data.

Keywords: model selection, kernel methods, change-point detection, concentration inequality

1. Introduction

The change-point problem has been considered in numerous papers in the statistics and machine learning literature (Brodsky and Darkhovsky, 1993; Carlstein et al., 1994; Tartakovsky et al., 2014; Truong et al., 2019). Given a time series, the goal is to split it into homogeneous segments, in which the marginal distribution of the observations —their mean or their variance, for instance— is constant. When the number of change points is known, this problem reduces to estimating the change-point locations as precisely as possible; in general, the number of change points itself must be estimated. This problem arises in a wide range of applications, such as bioinformatics (Picard et al., 2005; Curtis et al., 2012), neuroscience (Park et al., 2015), audio signal processing (Wu and Hsieh, 2006), temporal video segmentation (Koprinska and Carrato, 2001), hacker-attacks detection (Wang et al., 2014), social sciences (Kossinets and Watts, 2006) and econometrics (McCulloh, 2009).

©2019 Sylvain Arlot, Alain Celisse and Zaid Harchaoui.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/16-155.html.

1.1. Related Work

A large part of the literature on change-point detection deals with observations in $\mathbb{R}$ or $\mathbb{R}^d$ and focuses on detecting changes arising in the mean and/or the variance of the signal (Gijbels et al., 1999; Picard et al., 2005; Arlot and Celisse, 2011; Bertin et al., 2017). To this end, parametric models are often involved to derive change-point detection procedures. For instance, Comte and Rozenholc (2004), Lebarbier (2005), Picard et al. (2011) and Geneus et al. (2015) make a Gaussian assumption, while Frick et al. (2014) and Cleynen and Lebarbier (2014) consider an exponential family.

The challenging problem of detecting abrupt changes in the full distribution of the data has been recently addressed in the nonparametric setting. However, the corresponding procedures suffer several limitations since they are limited to real-valued data or they assume that the number of true change points is known. For instance, Zou et al. (2014) design a strategy based on empirical cumulative distribution functions that allows to recover an unknown number of change points by use of BIC, but only applies to $\mathbb{R}$-valued data. The strategy of Matteson and James (2014) applies to multivariate data, but it is time-consuming due to an intensive permutation use, and fully justified only in an asymptotic setting when there is a single change point (Biau et al., 2016). The kernel-based procedure proposed by Harchaoui and Cappé (2007) enables to deal not only with vectorial data but also with structured data in the sense of Gärtner (2008). The question of model selection, that is the adaptation to unknown number of change points, was raised by Harchaoui and Cappé (2007, Section 4.1); the present article solves this issue. Finally, many of these procedures cited above were analyzed and justified from a classical asymptotic viewpoint, which may be misleading in particular when the series of datapoints is not long.

Other attempts have been made to design change-point detection procedures allowing to deal with complex data (that are not necessarily vectors). However, the resulting procedures do not allow to detect more than one or two changes arising in particular features of the distribution. For instance, Chen and Zhang (2015) describe a strategy based on a dissimilarity measure between individuals to compute a graph from which a statistical test allows to detect only one or two change points. For a graph-valued time series, Wang et al. (2014) design specific scan statistics to test whether one change arises in the connectivity matrix.

1.2. Main Contributions

We first describe a multiple change-point detection procedure (KCP) allowing to deal with univariate, multivariate or complex data (DNA sequences or graphs, for instance) as soon as a positive semidefinite kernel can be defined for them. Among several assets, this procedure is nonparametric and does not require to know the true number of change points in advance. Furthermore, it allows to detect abrupt changes arising in the full distribution of the data by using a characteristic kernel; it can also focus on changes in specific features of the distribution by choosing an appropriate kernel.

Secondly, the procedure (KCP) is theoretically grounded with a finite-sample optimality result, namely an oracle inequality in terms of quadratic risk, stating that its performance is almost the same as that of the best one within the class we consider (Theorem 2). As argued by Lebarbier (2005) for instance, such a guarantee is what we want for a change-point detection procedure. It means that the procedure detects only changes that are "large enough" given the noise level and the amount of data available, which is necessary to avoid having many false positives. A crucial point is that Theorem 2 holds true for any value of the sample size n; in particular it can be smaller than the dimensionality of the data. Note that contrary to previous oracle inequalities in the change-point detection framework, the result we prove requires neither the variance to be constant nor the data to be Gaussian.

Thirdly, we settle a new concentration inequality for the quadratic norm of sums of independent Hilbert-valued vectors with exponential tails, which is a key result to derive the non-asymptotic oracle inequality for a large collection of candidate segmentations. The exponential concentration inequality may be of independent interest for other statistical problems involving model selection within a large collection of models. Let us finally mention that since the first version of the present work (Arlot et al., 2012), KCP has been successfully applied on different practical examples. Celisse et al. (2018) illustrate that KCP outperforms state-of-the-art approaches on biological data. Cabrieto et al. (2017) show that KCP with a Gaussian kernel outperforms three nonparametric methods for detecting correlation changes in synthetic multivariate time series, and provide an application to some data from behavioral sciences. Applying KCP to running empirical correlations (Cabrieto et al., 2018b) or to the autocorrelations of a multivariate time series (Cabrieto et al., 2018a) can make it focus on a specific kind of change —in the covariance between coordinates or in the autocorrelation structure of each coordinate, respectively—, as illustrated on synthetic data experiments and two real-world data sets from psychology.

1.3. Outline and Notation

Motivating examples are first provided in Section 2 to highlight the wide applicability of the procedure to various important settings. A comprehensive description of the proposed kernel change-point detection algorithm (KCP, or Algorithm 1) is provided in Section 3, where we also discuss algorithmic aspects as well as the practical choice of influential parameters (Section 3.3). Section 4 exposes some important ideas underlying KCP and then states the main theoretical results of the paper (Proposition 1 and Theorem 2). Proofs of these main results are collected in Section 5, while technical details are deferred to Appendices A and B. The practical performance of the kernel change-point detection algorithm is illustrated by experiments on synthetic data in Section 6 and on real data in Section 7. Section 8 concludes the paper by a short discussion.

For any $a < b$, we denote by $[\![a, b]\!] := [a, b] \cap \mathbb{N}$ the set of integers between a and b.

2. The Change-point Problem

Let $\mathcal{X}$ be some measurable set and $X_1, \ldots, X_n \in \mathcal{X}$ a sequence of independent $\mathcal{X}$-valued random variables. For any $i \in \{1, \ldots, n\}$, we denote by P_{X_i} the distribution of X_i. The change-point problem can then be summarized as follows: Given $(X_i)_{1 \leqslant i \leqslant n}$, the goal is to

find the locations of the abrupt changes along the sequence $P_{X_1}, \ldots, P_{X_n}$. Note that the case of dependent time series is often considered in the change-point literature (Lavielle and Moulines, 2000; Bardet and Kammoun, 2008; Bardet et al., 2012; Chang et al., 2018); as a first step, this paper focuses on the independent case for simplicity.

An important example to have in mind is when X_i corresponds to the observation at time $t_i = i/n$ of some random process on $[0, 1]$, and we assume that this process is stationary over $[t^\star_\ell, t^\star_{\ell+1})$, $\ell = 0, \ldots, D^\star - 1$, for some fixed sequence $0 = t^\star_0 < t^\star_1 < \cdots < t^\star_{D^\star} = 1$. Then, the change-point problem is equivalent to localizing the change points $t^\star_1, \ldots, t^\star_{D^\star-1} \in [0, 1]$, which should be possible as the sample size n tends to infinity. Note that we never make such an asymptotic assumption in the paper, where all theoretical results are non-asymptotic.

Let us now detail some motivating examples of the change-point problem.

Example 1 *The set $\mathcal{X}$ is $\mathbb{R}$ or $\mathbb{R}^d$, and the sequence $(P_{X_i})_{1 \leqslant i \leqslant n}$ changes only through its mean. This is the most classical setting, for which numerous methods have been proposed and analyzed in the one-dimensional setting (Comte and Rozenholc, 2004; Zhang and Siegmund, 2007; Boysen et al., 2009; Korostelev and Korosteleva, 2011; Fryzlewicz, 2014) as well as the multi-dimensional case (Picard et al., 2011; Bleakley and Vert, 2011; Hocking et al., 2013; Soh and Chandrasekaran, 2017; Collilieux et al., 2019).*

Example 2 *The set $\mathcal{X}$ is $\mathbb{R}$ or $\mathbb{R}^d$, and the sequence $(P_{X_i})_{1 \leqslant i \leqslant n}$ changes only through its mean and/or its variance (or covariance matrix). This setting is rather classical, at least in the one-dimensional case, and several methods have been proposed for it (Andreou and Ghysels, 2002; Picard et al., 2005; Fryzlewicz and Subba Rao, 2014; Cabrieto et al., 2017).*

Example 3 *The set $\mathcal{X}$ is $\mathbb{R}$ or $\mathbb{R}^d$, and no assumption is made on the changes in the sequence $(P_{X_i})_{1 \leqslant i \leqslant n}$. For instance, when data are centered and normalized, as in the audio-track example (Rabiner and Schäfer, 2007), the mean and the variance of the X_i can be constant, and only higher-order moments of $(P_{X_i})_{1 \leqslant i \leqslant n}$ are changing. Only a few recent papers deal with (an unknown number of) multiple change points in a fully nonparametric framework: Zou et al. (2014) for $\mathcal{X} = \mathbb{R}$, Matteson and James (2014) for $\mathcal{X} = \mathbb{R}^d$. Note that assuming $\mathcal{X} = \mathbb{R}$ and adding some further restrictions on the maximal order of the moments for which a change can arise in the sequence $(P_{X_i})_{1 \leqslant i \leqslant n}$, it is nevertheless possible to consider the multivariate sequence $((p_j(X_i))_{0 \leqslant j \leqslant d})_{1 \leqslant i \leqslant n}$, where p_j is a polynomial of degree j for $j \in \{0, \ldots, d\}$, and to use a method made for detecting changes in the mean (Example 1). For instance with $\mathbb{R}$-valued data, as proposed by Lajugie et al. (2014), one can take $p_j(X) = X^j$ for every $1 \leqslant j \leqslant d$, or p_j equal to the j-th Hermite polynomial.*

Example 4 *The set $\mathcal{X}$ is $\{(p_1, \ldots, p_d) \in [0, 1]^d$ such that $p_1 + \cdots + p_d = 1\}$ the d-dimensional simplex. For instance, audio and video data are often represented by histogram features (Oliva and Torralba, 2001; Lowe, 2004; Rabiner and Schäfer, 2007), as done in Section 7. In such cases, it is a bad idea to do as if $\mathcal{X}$ were $\mathbb{R}^d$-valued, since the Euclidean norm on $\mathbb{R}^d$ is usually a bad distance measure between histogram data.*

Example 5 *The set $\mathcal{X}$ is a set of graphs. For instance, the X_i can represent a social network (Kossinets and Watts, 2006) or a biological network (Curtis et al., 2012) that is changing over time (Chen and Zhang, 2015). Then, detecting meaningful changes in*

the structure of a time-varying network is a change-point problem. In the case of social networks, this can be used for detecting the rise of an economic crisis (McCulloh, 2009).

Example 6 *The set $\mathcal{X}$ is a set of texts (strings). For instance, text analysis can try to localize possible changes of authorship within a given text (Chen and Zhang, 2015).*

Example 7 *The set $\mathcal{X}$ is a subset of $\{A, T, C, G\}^{\mathbb{N}}$, the set of DNA sequences. For instance, an important question in phylogenetics is to find recombination events from the genome of individuals of a given species (Knowles and Kubatko, 2010; Ané, 2011). This can be achieved from a multiple alignment of DNA sequences (Schölkopf et al., 2004) by detecting abrupt changes (change points) in the phylogenetic tree at each DNA position, that is, by solving a change-point problem.*

Example 8 *The set $\mathcal{X}$ is a set of images. For instance, video-shot boundary detection (Cotsaces et al., 2006) or scene detection in videos (Allen et al., 2017) can be cast as change-point detection problems.*

Example 9 *The set $\mathcal{X}$ is an infinite-dimensional functional space. Such functional data arise in various fields (see for instance Ferraty and Vieu, 2006, Chapter 2), and the problem of testing whether there is a change or not in a functional time series has been considered recently (Ferraty and Vieu, 2006; Berkes et al., 2009; Sharipov et al., 2016).*

Other kinds of data could be considered, such as counting data (Cleynen and Lebarbier, 2014; Alaya et al., 2015), qualitative descriptors, as well as composite data, that is, data X_i that are mixing several above examples.

The goal of the paper is to describe a change-point detection procedure that is (i) general enough to handle all these situations (up to the choice of an appropriate similarity measure on $\mathcal{X}$), (ii) in a nonparametric framework, (iii) with an unknown number of change points, and (iv) that can be theoretically analyzed in all these examples simultaneously.

Note also that this procedure is required to output a set of change points that are "close to" the true ones, at least when n is large enough. But in settings where the signal-to-noise ratio is not strong enough to recover all true change points (for a fixed n), false positives are to be avoided. This motivates the non-asymptotic analysis of the procedure based on the (kernel) quadratic risk as a performance measure (see Eq. (9) in Section 4.5). Since the change-point detection procedure relies on model selection, an oracle inequality is proved in Section 4 —Eq. (12) in Theorem 2—, as usually done in non-asymptotic model-selection theory. It establishes that the procedure enjoys a quadratic risk close to the smallest possible value.

3. Detecting Changes in the Distribution With Kernels

We consider a general change-point problem where the data belongs to a set equipped with a positive semidefinite kernel.

3.1. Kernel Change-point Algorithm

For any integer $D \in [\![1, n+1]\!]$, the set of sequences of $(D-1)$ change points is defined by

$$\mathcal{T}_n^D := \left\{ (\tau_0, \dots, \tau_D) \in \mathbb{N}^{D+1} \,/\, 0 = \tau_0 < \tau_1 < \tau_2 < \dots < \tau_D = n \right\} \tag{1}$$

where $\tau_1, \dots, \tau_{D-1}$ are the change points, and τ_0, τ_D are just added for notational convenience. Any $\tau \in \mathcal{T}_n^D$ is called a *segmentation* (of $\{1, \dots, n\}$) into $D_\tau := D$ segments.

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive semidefinite kernel, that is, a measurable function $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for any $x_1, \dots, x_n \in \mathcal{X}$, the $n \times n$ matrix $(k(x_i, x_j))_{1 \leqslant i,j \leqslant n}$ is positive semidefinite. Examples of such kernels are given in Section 3.2. Then, we measure the quality of any candidate segmentation $\tau \in \mathcal{T}_n^D$ with the *kernel least-squares criterion* introduced by Harchaoui and Cappé (2007):

$$\widehat{\mathcal{R}}_n(\tau) := \frac{1}{n} \sum_{i=1}^{n} k(X_i, X_i) - \frac{1}{n} \sum_{\ell=1}^{D} \left[\frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} \sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} k(X_i, X_j) \right] . \tag{2}$$

In particular when $\mathcal{X} = \mathbb{R}$ and $k(x, y) = xy$, we recover the usual least-squares criterion

$$\widehat{\mathcal{R}}_n(\tau) = \frac{1}{n} \sum_{\ell=1}^{D} \sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} \left(X_i - \overline{X}_{[\![\tau_{\ell-1}+1, \tau_\ell]\!]} \right)^2 \quad \text{where} \quad \overline{X}_{[\![\tau_{\ell-1}+1, \tau_\ell]\!]} := \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} X_j \,.$$

Note that Eq. (6) in Section 4.1 provides an equivalent formula for $\widehat{\mathcal{R}}_n(\tau)$, which is helpful for understanding its meaning. Given the criterion (2), we cast the choice of τ as a model-selection problem (as thoroughly detailed in Section 4), which leads to Algorithm 1 below, that we now briefly comment on.

Input: observations: $X_1, \dots, X_n \in \mathcal{X}$,
kernel: $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,
constants: $c_1, c_2 > 0$ and $D_{\max} \in [\![1, n-1]\!]$.

Step 1: $\forall D \in [\![1, D_{\max}]\!]$, compute (by dynamic programming):
$$\widehat{\tau}(D) \in \operatorname{argmin}_{\tau \in \mathcal{T}_n^D} \left\{ \widehat{\mathcal{R}}_n(\tau) \right\} \quad \text{and} \quad \widehat{\mathcal{R}}_n\big(\widehat{\tau}(D)\big) .$$

Step 2: find:
$$\widehat{D} \in \operatorname{argmin}_{1 \leqslant D \leqslant D_{\max}} \left\{ \widehat{\mathcal{R}}_n\big(\widehat{\tau}(D)\big) + \frac{1}{n} \left[c_1 \log \binom{n-1}{D-1} + c_2 D \right] \right\} .$$

Output: sequence of change points: $\widehat{\tau} = \widehat{\tau}\big(\widehat{D}\big)$.

Algorithm 1: Kernel change-point algorithm (KCP)

- Step 1 of KCP consists in choosing the "best" segmentation with D segments — that is, the minimizer of the kernel least-squares criterion $\widehat{\mathcal{R}}_n(\cdot)$ over $\mathcal{T}_n^D$—, for every $D \in [\![1, D_{\max}]\!]$.

- Step 2 of KCP chooses D by model selection, using a penalized empirical criterion. A major contribution of this paper lies in the building and theoretical justification of the penalty $n^{-1}[c_1 \log \binom{n-1}{D-1} + c_2 D]$, see Sections 4–5; a simplified penalty, of the form $\frac{D}{n}(c_1 \log(\frac{n}{D}) + c_2)$, would also be possible, see Section 4.5.
- Practical issues (computational complexity and choice of constants $c_1, c_2, D_{\max}$) are discussed in Section 3.3. Let us only emphasize here that KCP is computationally tractable; its most expensive part is the minimization problem of Step 1, which can be done by dynamic programming (see Harchaoui and Cappé, 2007; Celisse et al., 2018). Efficient implementations of KCP in Python can be found in the *ruptures* package (Truong et al., 2018) and in the *Chapydette* package (Jones and Harchaoui, 2019).

3.2. Examples of Kernels

KCP can be used with various sets $\mathcal{X}$ (not necessarily vector spaces) as long as a positive semidefinite kernel on $\mathcal{X}$ is available. An important issue is to design relevant kernels, that are able to capture important features of the data for a given change-point problem, including non-vectorial data —for instance, simplicial data (histograms), texts or graphs (networks), see Section 2. The question of choosing a kernel is discussed in Section 8.2.

Classical kernels can be found in the books by Schölkopf and Smola (2001), Shawe-Taylor and Cristianini (2004) and Schölkopf et al. (2004) for instance. Let us mention a few of them:

- When $\mathcal{X} = \mathbb{R}^d$, $k^{\text{lin}}(x, y) = \langle x, y \rangle_{\mathbb{R}^d}$ defines the *linear kernel*. When $d = 1$, KCP then coincides with the algorithm proposed by Lebarbier (2005).
- When $\mathcal{X} = \mathbb{R}^d$, $k_h^{\text{G}}(x, y) = \exp[-\|x - y\|^2 / (2h^2)]$ defines the *Gaussian kernel* with bandwidth $h > 0$, which is used in the experiments of Section 6.
- When $\mathcal{X} = \mathbb{R}^d$, $k_h^L(x, y) = \exp[-\|x - y\|/h]$ defines the *Laplace kernel* with bandwidth $h > 0$.
- When $\mathcal{X} = \mathbb{R}^d$, $k_h^{\text{e}}(x, y) = \exp(\langle x, y \rangle_{\mathbb{R}^d}/h)$ defines the *exponential kernel* with bandwidth $h > 0$. Note that, unlike the Gaussian and Laplace kernels, the exponential kernel is not translation invariant.
- When $\mathcal{X} = \mathbb{R}$, $k_h^{\text{H}}(x, y) = \sum_{j=1}^{5} H_{j,h}(x) H_{j,h}(y)$, corresponds to the Hermite kernel, where $H_{j,h}(x) = 2^{j+1}\sqrt{\pi j!}\mathrm{e}^{-x^2/(2h^2)}(-1)^j \mathrm{e}^{-x^2/2}(\partial/\partial x)^j(\mathrm{e}^{-x^2/2})$ denotes the j-th Hermite function with bandwidth $h > 0$. This kernel is used in Section 6. It can of course be generalized to maximal-degree values different from 5.
- When $\mathcal{X}$ is the d-dimensional simplex as in Example 4, the χ^2 kernel can be defined by $k_h^{\chi^2}(x, y) = \exp\left(-\frac{1}{h \cdot d}\sum_{i=1}^{d} \frac{(x_i - y_i)^2}{x_i + y_i}\right)$ for some bandwidth $h > 0$. An illustration of its behavior is provided in the simulation experiments of Sections 6 and 7.

Note that more generally, Sejdinovic et al. (2013) prove that positive semidefinite kernels can be defined on any set $\mathcal{X}$ for which a semimetric of negative type is used to measure

closeness between points. The so-called *energy distance* between probability measures is an example (Matteson and James, 2014). In addition, specific kernels have been designed for various kinds of structured data, including all the examples of Section 2 (Cuturi et al., 2005; Rakotomamonjy and Canu, 2005; Shervashidze, 2012; Vedaldi and Zisserman, 2012). Convolutional kernels can also be designed to mimic the feature maps defined by convolutional networks, with successful applications in computer vision (Mairal et al., 2014; Paulin et al., 2017).

Let us finally remark that KCP can also be used when k is not a positive semidefinite kernel; its computational complexity remains unchanged, but we might loose the theoretical guarantees of Section 4.

3.3. Practical Issues

Let us now discuss the main practical issues when applying KCP.

3.3.1. Computational Complexity

The discrete optimization problem at Step 1 of KCP is apparently hard to solve since, for each D, there are $\binom{n-1}{D-1}$ segmentations of $\{1, \ldots, n\}$ into D segments. Fortunately, as shown by Harchaoui and Cappé (2007), this optimization problem can be solved efficiently by dynamic programming. In the special case of a linear kernel, we recover the classical dynamic-programming algorithm for detecting changes in mean (Fisher, 1958; Auger and Lawrence, 1989; Kay, 1993).

Denoting by $\mathcal{C}_k$ the cost of computing $k(x, y)$ for some given $x, y \in \mathcal{X}$, the computational cost of a naive implementation of Step 1 —computing each coefficient (i, j) of the *cost matrix* independently— then is $\mathcal{O}(\mathcal{C}_k n^2 + D_{\max} n^4)$ in time and $\mathcal{O}(D_{\max} n + n^2)$ in space. The computational complexity can actually be $\mathcal{O}((\mathcal{C}_k + D_{\max})n^2)$ in time and $\mathcal{O}(D_{\max} n)$ in space as soon as one either uses the summed area table or integral image technique as in (Potapov et al., 2014) or optimizes the interplay of the dynamic-programming recursions and cost-matrix computations (Celisse et al., 2018). For given constants $D_{\max}$ and c_1, c_2, Step 2 is straightforward since it consists in a minimization problem among $D_{\max}$ terms already stored in memory. Therefore, the overall complexity of KCP is at most $\mathcal{O}((\mathcal{C}_k + D_{\max})n^2)$ in time and $\mathcal{O}(D_{\max} n)$ in space.

3.3.2. Setting the Constants c_1 and c_2

At Step 2 of KCP, two constants $c_1, c_2 > 0$ appear in the penalty term. Theoretical guarantees (Theorem 2 in Section 4) suggest to take $c_1 = c_2 = c$ large enough, but the lower bound on c in Theorem 2 is pessimistic, and the optimal value of c certainly depends on unknown features of the data such as their "variance", as discussed after Theorem 2. In practice the constants c_1, c_2 must be chosen from data. To do so, we propose a fully data-driven method, based upon the "slope heuristics" (Baudry et al., 2012; Arlot, 2019), that is explained in Section 6.2. Another way of choosing c_1, c_2 is described in Appendix B.3.

3.3.3. Setting the Constant $D_{\max}$

KCP requires to specify the maximal dimension $D_{\max}$ of the segmentations considered, a choice that has three main consequences. First, the computational complexity of KCP is affine in $D_{\max}$, as discussed above. Second, if $D_{\max}$ is too small —smaller than the number of true change points that can be detected—, the segmentation $\widehat{\tau}$ provided by the algorithm will necessarily be too coarse. Third, when the slope heuristics is used for choosing c_1, c_2, taking $D_{\max}$ larger than the true number of change points might not be sufficient: better values for c_1, c_2 can be obtained by taking $D_{\max}$ larger, up to n. From our experiments, it seems that $D_{\max} \approx n/\sqrt{\log n}$ is large enough to provide good results.

3.4. Related Change-point Algorithms

In addition to the references given in the Introduction, let us mention a few change-point algorithms to which KCP is more closely related.

First, some two-sample (or homogeneity) tests based on kernels have been suggested. They tackle a simpler problem than the general change-point problem described in Section 2. Among them, Gretton et al. (2012a) propose a two-sample test based on a U-statistic of order two, called the maximum mean discrepancy (MMD). A related family of two-sample tests, called B-tests, is proposed by Zaremba et al. (2013); B-tests are also used by Li et al. (2015, 2019) for localizing a single change point. Harchaoui et al. (2008) propose a studentized kernel-based test statistic for testing homogeneity. Resampling methods —(block) bootstrap and permutations— have also been proposed for choosing the threshold of several kernel two-sample tests (Fromont et al., 2012; Chwialkowski et al., 2014; Sharipov et al., 2016).

Second, Harchaoui and Cappé (2007) consider a kernel-based method for multiple change-point detection, focusing on the case where the true number of segments $D^\star$ is known. Step 1 of KCP builds off the method of Harchaoui and Cappé (2007). The present paper proposes a data-driven choice of D supported by non-asymptotic theoretical guarantees.

Third, when $\mathcal{X} = \mathbb{R}$ and $k(x, y) = xy$, $\widehat{\mathcal{R}}_n(\tau)$ is the usual least-squares risk and Step 2 of KCP is similar to the penalization procedures proposed by Comte and Rozenholc (2004) and Lebarbier (2005) for detecting changes in the mean of a one-dimensional signal. We refer readers familiar with model-selection techniques to Section 4.1 for an equivalent formulation of KCP —in more abstract terms— that clearly emphasizes the links between KCP and these penalization procedures.

4. Theoretical Analysis

We now provide theoretical guarantees for KCP. We start by reformulating it in an abstract way, which enlightens how it works.

4.1. Abstract Formulation of KCP

Let $\mathcal{H} = \mathcal{H}_k$ denote the reproducing kernel Hilbert space (RKHS) associated with the positive semidefinite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The canonical feature map $\Phi : \mathcal{X} \mapsto \mathcal{H}$ is then defined by $\Phi(x) = k(x, \cdot) \in \mathcal{H}$ for every $x \in \mathcal{X}$. A detailed presentation of positive

semidefinite kernels and related notions can be found in several books (Schölkopf and Smola, 2001; Cucker and Zhou, 2007; Steinwart and Christmann, 2008).

Let us define $Y_i = \Phi(X_i) \in \mathcal{H}$ for every $i \in \{1, \dots, n\}$, $Y = (Y_i)_{1 \leqslant i \leqslant n} \in \mathcal{H}^n$, the set of segmentations $\mathcal{T}_n := \bigcup_{D=1}^n \mathcal{T}_n^D$ where $\mathcal{T}_n^D$ is defined by Eq. (1), and for every $\tau \in \mathcal{T}_n$,

$$F_\tau := \left\{ f = (f_1, \dots, f_n) \in \mathcal{H}^n \text{ s.t. } f_{\tau_{\ell-1}+1} = \dots = f_{\tau_\ell} \quad \forall 1 \leqslant \ell \leqslant D_\tau \right\}, \tag{3}$$

which is a linear subspace of $\mathcal{H}^n$. We also define on $\mathcal{H}^n$ the canonical scalar product by $\langle f, g \rangle := \sum_{i=1}^n \langle f_i, g_i \rangle_{\mathcal{H}}$ for $f, g \in \mathcal{H}^n$, and we denote by $\|\cdot\|$ the corresponding norm. Then, for any $g \in \mathcal{H}^n$,

$$\Pi_\tau g := \operatorname{argmin}_{f \in F_\tau} \left\{ \|f - g\|^2 \right\} \tag{4}$$

is the orthogonal projection of $g \in \mathcal{H}^n$ onto F_τ, and satisfies

$$\forall g \in \mathcal{H}^n, \forall 1 \leqslant \ell \leqslant D_\tau, \forall i \in [\![\tau_{\ell-1} + 1, \tau_\ell]\!], \quad (\Pi_\tau g)_i = \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} g_j \,. \tag{5}$$

This statement is proved in Appendix A.1.

Following Harchaoui and Cappé (2007), the empirical risk $\widehat{\mathcal{R}}_n(\tau)$ defined by Eq. (2) can be rewritten as

$$\widehat{\mathcal{R}}_n(\tau) = \frac{1}{n} \|Y - \widehat{\mu}_\tau\|^2 \qquad \text{where} \qquad \widehat{\mu}_\tau = \Pi_\tau Y \,, \tag{6}$$

as proved in Appendix A.1.

For each $D \in [\![1, D_{\max}]\!]$, Step 1 of KCP consists in finding a segmentation $\widehat{\tau}(D)$ in D segments such that

$$\widehat{\tau}(D) \in \operatorname{argmin}_{\tau \in \mathcal{T}_n^D} \left\{ \left\|Y - \widehat{\mu}_\tau\right\|^2 \right\} = \operatorname{argmin}_{\tau \in \mathcal{T}_n^D} \left\{ \inf_{f \in F_\tau} \sum_{i=1}^n \left\|\Phi(X_i) - f_i\right\|^2 \right\},$$

which is the "kernelized" version of the classical least-squares change-point algorithm (Lebarbier, 2005). Since the penalized criterion of Step 2 is similar to that of Comte and Rozenholc (2004) and Lebarbier (2005), we can see KCP as a "kernelization" of these penalized least-squares change-point procedures.

Let us emphasize that building a theoretically-grounded penalty for such a kernel least-squares change-point algorithm is not straightforward. For instance, we cannot apply the model-selection results by Birgé and Massart (2001) that were used by Comte and Rozenholc (2004) and Lebarbier (2005). Indeed, a Gaussian homoscedastic assumption is not realistic for general Hilbert-valued data, and we have to consider possibly heteroscedastic data for which we assume only that $Y_i = \Phi(X_i)$ is bounded in $\mathcal{H}$ —see Assumption (**Db**) in Section 4.3. Note that unbounded data X_i can satisfy Assumption (**Db**), for instance by choosing a bounded kernel such as the Gaussian or Laplace ones. In addition, dealing with Hilbert-valued random variables instead of (multivariate) real variables requires a new concentration inequality, see Proposition 1 in Section 4.4.

4.2. Intuitive Analysis

Section 4.1 shows that KCP can be seen as a kernelization of change-point algorithms focusing on changes of the mean of the signal (Lebarbier, 2005, for instance). Therefore, KCP is looking for changes in the "mean" of $Y_i = \Phi(X_i) \in \mathcal{H}$, provided that such a notion can be defined.

If $\mathcal{H}$ is separable and $\mathbb{E}[\sqrt{k(X_i, X_i)}] < +\infty$, we can define the (Bochner) mean $\mu_i^\star \in \mathcal{H}$ of $\Phi(X_i)$ (Ledoux and Talagrand, 1991), also called the mean element of P_{X_i}, by

$$\forall g \in \mathcal{H}, \qquad \langle \mu_i^\star, g \rangle_{\mathcal{H}} = \mathbb{E}\big[g(X_i)\big] = \mathbb{E}\big[\langle Y_i, g \rangle_{\mathcal{H}}\big]\,. \tag{7}$$

Then, we can write

$$\forall 1 \leqslant i \leqslant n, \qquad Y_i = \mu_i^\star + \varepsilon_i \in \mathcal{H} \qquad \text{where} \qquad \varepsilon_i := Y_i - \mu_i^\star\,.$$

The variables $(\varepsilon_i)_{1 \leqslant i \leqslant n}$ are independent and centered —that is, $\forall g \in \mathcal{H}$, $\mathbb{E}[\langle \varepsilon_i, g \rangle_{\mathcal{H}}] = 0$. So, we can understand $\widehat{\mu}_\tau$ as the least-squares estimator over F_τ of $\mu^\star = (\mu_1^\star, \ldots, \mu_n^\star) \in \mathcal{H}^n$.

An interesting case is when k is a *characteristic kernel* (Fukumizu et al., 2008), or equivalently, when $\mathcal{H}_k$ is *probability-determining* (Fukumizu et al., 2004a,b). Then any change in the distribution P_{X_i} induces a change in the mean element $\mu_i^\star$. In such settings, we can expect KCP to be able to detect *any change* in the distribution P_{X_i}, at least asymptotically. For instance the Gaussian kernel is characteristic (Fukumizu et al., 2004b, Theorem 4), and general sufficient conditions for k to be characteristic are known (Sriperumbudur et al., 2010, 2011).

Note that Sharipov et al. (2016) suggest to use $k_\leqslant(x, y) = \mathbf{1}_{x \leqslant y}$ as a "kernel" within a two-sample test, in order to look for any change of the distribution of real-valued data X_i (Example 3). This idea is similar to our proposal of using KCP with a characteristic kernel for tackling Example 3, even if we do not advise to take $k = k_\leqslant$ within KCP. Indeed, when $k = k_\leqslant$, $\widehat{\mathcal{R}}_n(\tau) = \frac{1}{2} - \frac{D_\tau}{2n}$ as soon as the X_i are all different so that KCP becomes useless. This illustrates that using a kernel which is not symmetric positive definite should be done cautiously.

4.3. Notation and Assumptions

Throughout the paper, we assume that $\mathcal{H}$ *is separable*, which is kind of a minimal assumption for two reasons: it allows to define the mean element —see Eq. (7)—, and most reasonable examples satisfy this requirement (Dieuleveut and Bach, 2014, p. 4). Let us further assume

$$\exists M \in (0, +\infty)\,, \quad \forall i \in \{1, \ldots, n\}, \qquad \|Y_i\|_{\mathcal{H}}^2 = \|\Phi(X_i)\|_{\mathcal{H}}^2 = k(X_i, X_i) \leqslant M^2 \quad \text{a.s.} \quad \textbf{(Db)}$$

For every $1 \leqslant i \leqslant n$, we also define the "variance" of Y_i by

$$v_i := \mathbb{E}\Big[\big\|\Phi(X_i) - \mu_i^\star\big\|_{\mathcal{H}}^2\Big] = \mathbb{E}\big[k(X_i, X_i)\big] - \big\|\mu_i^\star\big\|_{\mathcal{H}}^2 = \mathbb{E}\big[k(X_i, X_i) - k(X_i, X_i')\big] \tag{8}$$

where X_i' is an independent copy of X_i, and $v_{\max} := \max_{1 \leqslant i \leqslant n} v_i$. Let us make a few remarks.

- If (**Db**) holds true, then the mean element $\mu_i^\star$ exists since $\mathbb{E}[\sqrt{k(X_i, X_i)}] < \infty$, the variances v_i are finite and smaller than $v_{\max} \leqslant M^2$.

- If (**Db**) holds true, then Y_i admits a covariance operator Σ_i that is trace-class and $v_i = \mathrm{tr}(\Sigma_i)$.
- If k is translation invariant, that is, $\mathcal{X}$ is a vector space and $k(x, x') = \overline{k}(x - x')$ for every $x, x' \in \mathcal{X}$, and some measurable function $\overline{k} : \mathcal{X} \to \mathbb{R}$, then (**Db**) holds true with $M^2 = \overline{k}(0)$ and $v_i = \overline{k}(0) - \|\mu_i^\star\|_{\mathcal{H}}^2$. For instance the Gaussian and Laplace kernels are translation invariant (see Section 3.2).
- Let us consider the case of the linear kernel $(x, y) \mapsto \langle x, y \rangle$ on $\mathcal{X} = \mathbb{R}^d$. If we assume $\mathbb{E}[\|X_i\|_{\mathbb{R}^d}^2] < \infty$, then, $v_i = \mathrm{tr}(\Sigma_i)$ where Σ_i is the covariance matrix of X_i. In addition, (**Db**) holds true if and only if $\|X_i\|_{\mathbb{R}^d} \leqslant M$ a.s. for all i.

4.4. Concentration Inequality for Some Quadratic Form of Hilbert-valued Random Variables

Our main theoretical result, stated in Section 4.5, relies on two concentration inequalities for some linear and quadratic functionals of Hilbert-valued vectors. Here we state the concentration result that we prove for the quadratic term, which is significantly different from existing results and can be of independent interest.

Proposition 1 (Concentration of the quadratic term) *Let $\tau \in \mathcal{T}_n$ and recall that Π_τ is the orthogonal projection onto F_τ in $\mathcal{H}^n$ defined by Eq. (4). Let $X_1, \ldots, X_n$ be independent $\mathcal{X}$-valued random variables and assume that* (**Db**) *holds true, so that $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n) \in \mathcal{H}^n$ is defined as in Section 4.1. Then for every $x > 0$, with probability at least $1 - \mathrm{e}^{-x}$,*

$$\|\Pi_\tau \varepsilon\|^2 - \mathbb{E}\big[\|\Pi_\tau \varepsilon\|^2\big] \leqslant \frac{14M^2}{3}\Big(x + 2\sqrt{2xD_\tau}\Big)\,.$$

Proposition 1 is proved in Section 5.4. The proof relies on a combination of Bernstein's and Pinelis-Sakhanenko's inequalities. Note that the proof of Proposition 1 also shows that for every $x > 0$, with probability at least $1 - \mathrm{e}^{-x}$,

$$\|\Pi_\tau \varepsilon\|^2 - \mathbb{E}\big[\|\Pi_\tau \varepsilon\|^2\big] \geqslant \frac{-14M^2}{3}\Big(x + 2\sqrt{2xD_\tau}\Big)\,.$$

Previous concentration results for quantities such as $\|\Pi_\tau \varepsilon\|^2$ or $\|\Pi_\tau \varepsilon\|$ do not imply Proposition 1 —even up to numerical constants. Indeed, they either assume that ε is a Gaussian vector, or they involve much larger deviation terms (see Section 5.4.3 for a detailed discussion of these results).

4.5. Oracle Inequality for KCP

Similarly to the results of Comte and Rozenholc (2004) and Lebarbier (2005) in the one-dimensional case, we state below a non-asymptotic oracle inequality for KCP. First, we define the (kernel) quadratic risk of any $\mu \in \mathcal{H}^n$ as an estimator of $\mu^\star$ by

$$\mathcal{R}(\mu) = \frac{1}{n}\|\mu - \mu^\star\|^2 = \frac{1}{n}\sum_{i=1}^n \|\mu_i - \mu_i^\star\|_{\mathcal{H}}^2\,. \tag{9}$$

Theorem 2 *We consider the framework and notation introduced in Sections 2–4. Let $C \geqslant 0$ be some constant. Assume that* (**Db**) *holds true and that* $\mathrm{pen} : \mathcal{T}_n \to \mathbb{R}$ *is some penalty function satisfying*

$$\forall \tau \in \mathcal{T}_n, \qquad \mathrm{pen}(\tau) \geqslant \frac{CM^2}{n}\left[\log\binom{n-1}{D_\tau - 1} + D_\tau\right]. \tag{10}$$

Then, some numerical constant $L_1 > 0$ exists such that the following holds: if $C \geqslant L_1$, for every $y \geqslant 0$, an event of probability at least $1 - \mathrm{e}^{-y}$ exists on which, for every

$$\widehat{\tau} \in \mathrm{argmin}_{\tau \in \mathcal{T}_n}\left\{\widehat{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau)\right\}, \tag{11}$$

we have

$$\mathcal{R}(\widehat{\mu}_{\widehat{\tau}}) \leqslant 2 \inf_{\tau \in \mathcal{T}_n}\left\{\mathcal{R}(\widehat{\mu}_\tau) + \mathrm{pen}(\tau)\right\} + \frac{83yM^2}{n}. \tag{12}$$

Theorem 2 is proved in Section 5.5.

Theorem 2 applies to the segmentation $\widehat{\tau}$ output by KCP when c_1 and c_2 are larger than L_1M^2. It shows that $\widehat{\mu}_{\widehat{\tau}}$ estimates well the "mean" $\mu^\star \in \mathcal{H}^n$ of the transformed time series $Y_1 = \Phi(X_1), \ldots, Y_n = \Phi(X_n)$. Such an oracle inequality, namely Eq. (12), is the classical means for theoretically validating any model-selection procedure in a non-asymptotic way (Birgé and Massart, 2001, for instance). Since the present change-point detection procedure (KCP) is based on model selection, Eq. (12) can serve as theoretical validation. In addition, such a non-asymptotic optimality result is necessary for taking into account situations where some changes are too small to be detected —they are "below the noise level" (Lebarbier, 2005, for instance). By defining the performance of $\widehat{\tau}$ as the quadratic risk of $\widehat{\mu}_{\widehat{\tau}}$ (seen as an estimator of $\mu^\star$), the oracle inequality (12) proves that KCP works well for finite sample size and for a set $\mathcal{X}$ that can have a large dimensionality (possibly much larger than the sample size n). The consistency of KCP for estimating the change-point locations, which is outside the scope of this paper, is discussed in Section 8.1.

Why is a penalty satisfying Eq. (10) a good choice? As detailed in Section 5.1, the oracle inequality in Eq. (12) results from taking a penalty such that, for every $\tau \in \mathcal{T}_n$, the (penalized) empirical criterion $\widehat{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau)$ in Eq. (11) approximates the (oracle) performance measure $\mathcal{R}(\widehat{\mu}_\tau)$. At least, the penalty must be *large enough* so that

$$\widehat{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau) \geqslant \mathcal{R}(\widehat{\mu}_\tau)$$

holds true *simultaneously* for all $\tau \in \mathcal{T}_n$ (up to technical details exposed in Section 5). Then, the core of the proof of Theorem 2 is to show that Eq. (10) implies that such a set of inequalities holds true on an event of high probability.

The constant 2 in front of the first term in Eq. (12) has no special meaning, and could be replaced by any quantity strictly larger than 1, at the price of enlarging 83 in the right-hand side of Eq. (12) and the constant L_1.

The value L_1M^2 suggested by Theorem 2 for the constants c_1, c_2 within KCP should not be used in practice because it is likely to lead to a conservative choice for two reasons. First, the minimal value L_1 for the constant C suggested by the proof of Theorem 2 depends on the numerical constants appearing in the deviation bounds of Propositions 1 and 3, which

probably are not optimal. Second, the constant M^2 in the penalty is probably pessimistic in several frameworks. For instance, with the linear kernel and Gaussian data belonging to $\mathcal{X} = \mathbb{R}$, (**Db**) is not satisfied, but other similar oracle inequalities have been proved with M^2 replaced by the residual variance (Lebarbier, 2005). In practice, as we do in the experiments of Sections 6–7, we recommend to use a data-driven value for the leading constant C in the penalty, as explained in Section 3.3.

Theorem 2 also applies to KCP with simplified penalty shapes. Indeed,

$$\forall D \in \{1, \ldots, n\}, \qquad \binom{n-1}{D-1} = \frac{D}{n}\binom{n}{D} \leqslant \binom{n}{D} \leqslant \frac{n^D}{D!} \leqslant \left(\frac{\mathrm{e}n}{D}\right)^D$$

so that Theorem 2 applies to the penalty $\frac{D}{n}[c_1 \log(\frac{n}{D}) + c_2]$ —similar to the one of Lebarbier (2005)— as soon as $c_1, c_2 \geqslant L_1M^2$. A BIC-type penalty $CD\log(n)/n$ is also covered by Theorem 2 provided that $C \geqslant 2.5L_1M^2$ and $n \geqslant 2$, even if we do not recommend to use it —see Section 6.3.

A nice feature of Theorem 2 is that it holds under mild assumptions: we only need the data X_i to be independent and to have (**Db**) satisfied. As noticed in Section 4.3, (**Db**) holds true for any translation-invariant kernel, such as the Gaussian and Laplace kernels. Compared to previous results (Comte and Rozenholc, 2004; Lebarbier, 2005), we do not need the data to be Gaussian or homoscedastic. Furthermore, the independence assumption can certainly be relaxed: to do so, it would be sufficient to prove concentration inequalities similar to Propositions 1 and 3 for some dependent X_i.

In the particular setting where $\mathcal{X} = \mathbb{R}$ and k is the linear kernel $(x, y) \mapsto xy$, Theorem 2 provides an oracle inequality similar to the one proved by Lebarbier (2005) for Gaussian and homoscedastic real-valued data. The price to pay for extending this result to heteroscedastic Hilbert-valued data is rather mild: we only assume (**Db**) and replace the residual variance by M^2.

Apart from the results already mentioned, a few oracle inequalities have been proved for change-point procedures, for real-valued data with a multiplicative penalty (Baraud et al., 2009), for discrete data (Akakpo, 2011), for counting data with a total-variation penalty (Alaya et al., 2015), for counting data with a penalized maximum-likelihood procedure (Cleynen and Lebarbier, 2014) and for data distributed according to an exponential family (Cleynen and Lebarbier, 2017). Among these oracle inequalities, only the result by Akakpo (2011) is more precise than Theorem 2 —there is no $\log(n)$ factor compared to the oracle loss—, at the price of using a smaller (dyadic) collection of possible segmentations, hence a worse oracle performance in general.

5. Main Proofs

We now prove the main results of the paper, Theorem 2 and Proposition 1.

5.1. Outline of the Proof of Theorem 2

As usual for proving an oracle inequality (see Arlot, 2014, Section 2.2), we remark that by Eq. (11), for every $\tau \in \mathcal{T}_n$,

$$\widehat{\mathcal{R}}_n(\widehat{\tau}) + \mathrm{pen}(\widehat{\tau}) \leqslant \widehat{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau).$$

Therefore,

$$\mathcal{R}(\widehat{\mu}_{\widehat{\tau}}) + \mathrm{pen}(\widehat{\tau}) - \mathrm{pen}_{\mathrm{id}}(\widehat{\tau}) \leqslant \mathcal{R}(\widehat{\mu}_\tau) + \mathrm{pen}(\tau) - \mathrm{pen}_{\mathrm{id}}(\tau) \tag{13}$$

$$\text{where} \qquad \forall \tau \in \mathcal{T}, \qquad \mathrm{pen}_{\mathrm{id}}(\tau) := \mathcal{R}(\widehat{\mu}_\tau) - \widehat{\mathcal{R}}_n(\tau) + \frac{1}{n}\|\varepsilon\|^2 . \tag{14}$$

The idea of the proof is that if we prove that $\mathrm{pen}(\tau) \geqslant \mathrm{pen}_{\mathrm{id}}(\tau)$ for every $\tau \in \mathcal{T}_n$, we get an oracle inequality similar to Eq. (12). What remains to obtain is a deterministic upper bound on the *ideal penalty* $\mathrm{pen}_{\mathrm{id}}(\tau)$ that holds true simultaneously for all $\tau \in \mathcal{T}_n$ on a large probability event. To this aim, we compute $\mathbb{E}[\mathrm{pen}_{\mathrm{id}}(\tau)]$ and show that $\mathrm{pen}_{\mathrm{id}}(\tau)$ concentrates around its expectation for every $\tau \in \mathcal{T}_n$ (Sections 5.2–5.4). Then we use a union bound as detailed in Section 5.5. A similar strategy is used for instance by Comte and Rozenholc (2004) and Lebarbier (2005) in the specific context of change-point detection.

Note that we prove below a slightly weaker result than $\mathrm{pen}(\tau) \geqslant \mathrm{pen}_{\mathrm{id}}(\tau)$, which is nevertheless sufficient to obtain Eq. (12). Remark also that Eq. (13) would be true if the constant $n^{-1}\|\varepsilon\|^2$ in the definition (14) of $\mathrm{pen}_{\mathrm{id}}$ was replaced by any quantity independent from τ; the reasons for this specific choice appear in the computations below.

5.2. Computation of the Ideal Penalty

From Eq. (14) it results that for every $\tau \in \mathcal{T}_n$,

$$\begin{aligned} n \times \mathrm{pen}_{\mathrm{id}}(\tau) &= \|\widehat{\mu}_\tau - \mu^\star\|^2 - \|\widehat{\mu}_\tau - Y\|^2 + \|\varepsilon\|^2 \\ &= \|\widehat{\mu}_\tau - \mu^\star\|^2 - \|\widehat{\mu}_\tau - \mu^\star - \varepsilon\|^2 + \|\varepsilon\|^2 \\ &= 2\,\langle \widehat{\mu}_\tau - \mu^\star,\, \varepsilon\rangle \\ &= 2\big\langle \Pi_\tau(\mu^\star + \varepsilon) - \mu^\star,\, \varepsilon\big\rangle \\ &= 2\,\langle \Pi_\tau\mu^\star - \mu^\star,\, \varepsilon\rangle + 2\,\langle \Pi_\tau\varepsilon,\, \varepsilon\rangle \\ &= 2\,\langle \Pi_\tau\mu^\star - \mu^\star,\, \varepsilon\rangle + 2\,\|\Pi_\tau\varepsilon\|^2 \end{aligned} \tag{15}$$

since Π_τ is an orthogonal projection. The next two sections focus separately on the two terms appearing in Eq. (15).

5.3. Concentration of the Linear Term

We prove in Section A.2 the following concentration inequality for the linear term in Eq. (15), mostly by applying Bernstein's inequality.

Proposition 3 (Concentration of the linear term) *If* (**Db**) *holds true, then for every* $x > 0$*, with probability at least* $1 - 2\mathrm{e}^{-x}$*,*

$$\forall \theta > 0\,, \quad \Big|\big\langle (I - \Pi_\tau)\mu^\star,\, \Phi(\mathbf{X}) - \mu^\star\big\rangle\Big| \leqslant \theta\,\|\Pi_\tau\mu^\star - \mu^\star\|^2 + \left(\frac{v_{\max}}{2\theta} + \frac{4M^2}{3}\right) x\,. \tag{16}$$

5.4. Dealing With the Quadratic Term

We now focus on the quadratic term in the right-hand side of Eq. (15).

5.4.1. Preliminary Computations

We start by providing a useful closed-form formula for $\|\Pi_\tau\varepsilon\|^2$ and by computing its expectation. First, a straightforward consequence of Eq. (5) is that

$$\|\Pi_\tau\varepsilon\|^2 = \sum_{\ell=1}^{D_\tau}\left[\frac{1}{\tau_\ell - \tau_{\ell-1}}\left\|\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\varepsilon_i\right\|_{\mathcal{H}}^2\right] \tag{17}$$

$$= \sum_{\ell=1}^{D_\tau}\left[\frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{\tau_{\ell-1}+1\leqslant i,j\leqslant\tau_\ell}\langle\varepsilon_i,\,\varepsilon_j\rangle_{\mathcal{H}}\right]. \tag{18}$$

Second, we remark that for every $i, j \in \{1, \ldots, n\}$,

$$\begin{aligned} \mathbb{E}\left[\langle\varepsilon_i,\,\varepsilon_j\rangle_{\mathcal{H}}\right] &= \mathbb{E}\left[\langle\Phi(X_i),\,\Phi(X_j)\rangle_{\mathcal{H}}\right] - \mathbb{E}\left[\langle\mu_i^\star,\,\Phi(X_j)\rangle_{\mathcal{H}}\right] - \mathbb{E}\left[\langle\Phi(X_i),\,\mu_j^\star\rangle_{\mathcal{H}}\right] + \langle\mu_i^\star,\,\mu_j^\star\rangle_{\mathcal{H}} \\ &= \mathbb{E}\left[\langle\Phi(X_i),\,\Phi(X_j)\rangle_{\mathcal{H}}\right] - \langle\mu_i^\star,\,\mu_j^\star\rangle_{\mathcal{H}} \\ &= \mathbf{1}_{i=j}\left(\mathbb{E}\left[k(X_i, X_i)\right] - \|\mu_i^\star\|_{\mathcal{H}}^2\right) = \mathbf{1}_{i=j}v_i\,. \end{aligned} \tag{19}$$

Combining Eq. (18) and (19), we get

$$\mathbb{E}\left[\|\Pi_\tau\varepsilon\|^2\right] = \sum_{\ell=1}^{D_\tau}\left[\frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} v_i\right] = \sum_{\ell=1}^{D_\tau} v_\ell^\tau\,, \tag{20}$$

where $v_\ell^\tau := \frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} v_i$.

5.4.2. Concentration: Proof of Proposition 1

This proof is inspired from that of a concentration inequality by Sauvé (2009) in the context of regression with real-valued non-Gaussian noise. Let us define

$$T_\ell := \frac{1}{\tau_\ell - \tau_{\ell-1}}\left\|\sum_{j=\tau_{\ell-1}+1}^{\tau_\ell}\varepsilon_j\right\|_{\mathcal{H}}^2, \qquad \text{so that} \qquad \|\Pi_\tau\varepsilon\|^2 = \sum_{1\leqslant\ell\leqslant D_\tau} T_\ell$$

by Eq. (17). Since the real random variables $(T_\ell)_{1\leqslant\ell\leqslant D_\tau}$ are independent, we get a concentration inequality for their sum $\|\Pi_\tau\varepsilon\|^2$ via Bernstein's inequality (Proposition 6 in Appendix B.1) as long as T_ℓ satisfies some moment conditions. The rest of the proof consists in showing such moment bounds by using Pinelis-Sakhanenko's deviation inequality (Proposition 7 in Appendix B.1).

First, note that (**Db**) implies that $\|\varepsilon_i\|_{\mathcal{H}} \leqslant 2M$ almost surely for every i by Lemma 5, hence $\|\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\varepsilon_i\|_{\mathcal{H}} \leqslant 2(\tau_\ell - \tau_{\ell-1})M$ a.s. for every $1 \leqslant \ell \leqslant D_\tau$. Then for every $q \geqslant 2$ and $1 \leqslant \ell \leqslant D_\tau$,

$$\mathbb{E}\left[T_\ell^q\right] = \frac{1}{(\tau_\ell - \tau_{\ell-1})^q}\int_0^{2(\tau_\ell - \tau_{\ell-1})M} 2qx^{2q-1}\mathbb{P}\left(\left\|\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\varepsilon_i\right\|_{\mathcal{H}} \geqslant x\right)\mathrm{d}x\,. \tag{21}$$

Second, since $\|\varepsilon_i\|_{\mathcal{H}} \leqslant 2M$ almost surely and $\mathbb{E}\big[\|\varepsilon_i\|_{\mathcal{H}}^2\big] = v_i \leqslant M^2$ for every i, we get that for every $p \geqslant 2$ and $1 \leqslant \ell \leqslant D_\tau$,

$$\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} \mathbb{E}\left[\|\varepsilon_i\|_{\mathcal{H}}^p\right] \leqslant \frac{p!}{2}\left(\sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} v_j\right)\left(\frac{2M}{3}\right)^{p-2} \leqslant \frac{p!}{2} \times (\tau_\ell - \tau_{\ell-1})M^2 \times \left(\frac{2M}{3}\right)^{p-2} .$$

Hence, the assumptions of Pinelis-Sakhanenko's deviation inequality (Pinelis and Sakhanenko, 1986) —which is recalled by Proposition 7 in Appendix B.1— are satisfied with $c = 2M/3$ and $\sigma^2 = (\tau_\ell - \tau_{\ell-1})M^2$, and we get that for every $x \in [0,\ 2(\tau_\ell - \tau_{\ell-1})M]$

$$\begin{aligned}\mathbb{P}\left(\left\|\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} \varepsilon_i\right\|_{\mathcal{H}} \geqslant x\right) &\leqslant 2\exp\left(-\frac{x^2}{2\big[(\tau_\ell - \tau_{\ell-1})M^2 + \frac{2Mx}{3}\big]}\right)\\ &\leqslant 2\exp\left(-\frac{3x^2}{14(\tau_\ell - \tau_{\ell-1})M^2}\right).\end{aligned}$$

Together with Eq. (21), we obtain that

$$\begin{aligned}\mathbb{E}\left[T_\ell^q\right] &\leqslant \frac{4q}{(\tau_\ell - \tau_{\ell-1})^q}\int_0^{2(\tau_\ell-\tau_{\ell-1})M} x^{2q-1}\exp\left(-\frac{3x^2}{14(\tau_\ell - \tau_{\ell-1})M^2}\right)\mathrm{d}x\\ &\leqslant 4q\left(\frac{7M^2}{3}\right)^q \int_0^{+\infty} u^{2q-1}\exp\left(-\frac{u^2}{2}\right)\mathrm{d}u\\ &= 2^{q-1}(q-1)! \times 4q\left(\frac{7M^2}{3}\right)^q\\ &= 2 \times (q!)\left[\frac{14M^2}{3}\right]^q ,\end{aligned} \tag{22}$$

since for every $q \geqslant 1$,

$$\int_0^{+\infty} u^{2q-1}\exp(-u^2/2)\,\mathrm{d}u = 2^{q-1}(q-1)!\,.$$

Finally, summing Eq. (22) over $1 \leqslant \ell \leqslant D_\tau$, it comes

$$\begin{aligned}\sum_{1\leqslant\ell\leqslant D_\tau} \mathbb{E}\left[T_\ell^q\right] &\leqslant 2 \times (q!)\left[\frac{14M^2}{3}\right]^q D_\tau\\ &= \frac{q!}{2} \times D_\tau \left[\frac{28M^2}{3}\right]^2 \times \left[\frac{14M^2}{3}\right]^{q-2} .\end{aligned}$$

Then, condition (34) of Bernstein's inequality holds true with

$$v = D_\tau\left[\frac{28M^2}{3}\right]^2 \quad \text{and} \quad c = \frac{14M^2}{3}\,.$$

Therefore, Bernstein's inequality (Massart, 2007, Proposition 2.9) —which is recalled by Proposition 6 in Appendix B.1— shows that for every $x > 0$, with probability at least

$1 - \mathrm{e}^{-x}$,

$$\begin{aligned}\|\Pi_\tau\varepsilon\|^2 - \mathbb{E}\left[\|\Pi_\tau\varepsilon\|^2\right] &\leqslant \sqrt{2vx} + cx\\ &= \sqrt{2D_\tau x}\frac{28M^2}{3} + \frac{14M^2}{3}x\\ &= \frac{14M^2}{3}\left(2\sqrt{2D_\tau x} + x\right) .\end{aligned}$$

■

5.4.3. Why Do We Need a New Concentration Inequality?

We now review previous concentration results for quantities such as $\|\Pi_\tau\varepsilon\|^2$ or $\|\Pi_\tau\varepsilon\|$, showing that they are not sufficient for our needs, hence requiring a new result such as Proposition 1.

First, when $\varepsilon \in \mathbb{R}^n$ is a Gaussian isotropic vector, $\|\Pi_\tau\varepsilon\|^2$ is a chi-square random variable for which concentration tools have been developed. Such results are used by Birgé and Massart (2001) and by Lebarbier (2005) for instance. They cannot be applied here since ε cannot be assumed Gaussian, and the ε_j do not necessarily have the same variance.

Second, Eq. (18) shows that $\|\Pi_\tau\varepsilon\|^2$ is a U-statistic of order 2. Some tight exponential concentration inequalities exist for such quantities when $\varepsilon_j \in \mathbb{R}$ (Houdré and Reynaud-Bouret, 2003) and when ε_j belongs to a general measurable set (Giné and Nickl, 2016, Theorem 3.4.8). In both results, a term of order M^2x^2 appears in the deviations, which is too large because the proof of Theorem 2 relies on Proposition 1 with $x \gg D_\tau$: we really need a smaller deviation term, as in Proposition 1 where it is proportional to M^2x.

Third, since

$$\|\Pi_\tau\varepsilon\| = \sup_{f\in\mathcal{H}^n, \|f\|=1}\big|\langle f,\, \Pi_\tau\varepsilon\rangle\big| = \sup_{f\in\mathcal{H}^n, \|f\|=1}\Big|\sum_{i=1}^n \langle f_i,\, (\Pi_\tau\varepsilon)_i\rangle_{\mathcal{H}}\Big|\,,$$

Talagrand's inequality (Boucheron et al., 2013, Corollary 12.12) provides a concentration inequality for $\|\Pi_\tau\varepsilon\|$ around its expectation. More precisely, we can get the following result, which is proved in Appendix B.2.

Proposition 4 *If* (**Db**) *holds true, then for every $x > 0$ with probability at least $1 - 2\mathrm{e}^{-x}$,*

$$\big|\|\Pi_\tau\varepsilon\| - \mathbb{E}\big[\|\Pi_\tau\varepsilon\|\big]\big| \leqslant \sqrt{2x\left(4M\mathbb{E}\big[\|\Pi_\tau\varepsilon\|\big] + \max_{1\leqslant\ell\leqslant D_\tau} v_\ell^\tau\right)} + \frac{2Mx}{3}\,. \tag{23}$$

Therefore, in order to get a concentration inequality for $\|\Pi_\tau\varepsilon\|^2$, we have to square Eq. (23) and we necessarily get a deviation term of order M^2x^2. As with the U-statistics approach, this is too large for our needs.

Fourth, given Eq. (17), it is also natural to think of Pinelis-Sakhanenko's inequality (Pinelis and Sakhanenko, 1986), but this result alone is not precise enough because it is a *deviation* inequality, and not a *concentration* inequality. It is nevertheless a key ingredient in the proof of Proposition 1.

5.5. Oracle Inequality: Proof of Theorem 2

We now end the proof of Theorem 2 as explained in Section 5.1.

5.5.1. Upper Bound on $\mathrm{pen}_{\mathrm{id}}(\tau)$ for Every $\tau \in \mathcal{T}_n$

First, by Eq. (15) for every $\tau \in \mathcal{T}_n$,

$$\mathrm{pen}_{\mathrm{id}}(\tau) = \frac{1}{n}\left(\|\widehat{\mu}_\tau - \mu^\star\|^2 - \|\widehat{\mu}_\tau - Y\|^2 + \|\varepsilon\|^2\right) = \frac{2}{n}\|\Pi_\tau\varepsilon\|^2 - \frac{2}{n}\big\langle (I-\Pi_\tau)\mu^\star,\, \varepsilon\big\rangle . \quad (24)$$

In other words, $\mathrm{pen}_{\mathrm{id}}(\tau)$ is the sum of two terms, for which Propositions 1 and 3 provide concentration inequalities.

On the one hand, by Proposition 1 under (**Db**), for every $\tau \in \mathcal{T}_n$ and $x \geqslant 0$, with probability at least $1 - \mathrm{e}^{-x}$ we have

$$\frac{2}{n}\|\Pi_\tau\varepsilon\|^2 \leqslant \frac{2}{n}\left(\mathbb{E}\left[\|\Pi_\tau\varepsilon\|^2\right] + \frac{14M^2}{3}\left(x + 2\sqrt{2xD_\tau}\right)\right) \quad (25)$$

$$\leqslant \frac{2M^2}{n}\left(D_\tau + \frac{14x}{3} + \frac{28}{3}\sqrt{2xD_\tau}\right) \quad (26)$$

since

$$\mathbb{E}\left[\|\Pi_\tau\varepsilon\|^2\right] = \sum_{j=1}^{D_\tau} v_j^\tau \leqslant D_\tau M^2$$

by Eq. (20). On the other hand, by Proposition 3 under (**Db**), for every $\tau \in \mathcal{T}_n$ and $x \geqslant 0$, with probability at least $1 - 2\mathrm{e}^{-x}$ we have

$$\forall \theta > 0, \qquad \frac{2}{n}\Big|\big\langle (I-\Pi_\tau)\mu^\star,\, \varepsilon\big\rangle\Big| \leqslant \frac{2\theta}{n}\|\Pi_\tau\mu^\star - \mu^\star\|^2 + \frac{2}{n}\left(\frac{v_{\max}}{2\theta} + \frac{4M^2}{3}\right)x$$

$$\leqslant \frac{2\theta}{n}\|\Pi_\tau\mu^\star - \mu^\star\|^2 + \frac{xM^2}{n}\left(\theta^{-1} + \frac{8}{3}\right). \quad (27)$$

For every $\tau \in \mathcal{T}_n$ and $x \geqslant 0$, let Ω_x^τ be the event on which Eq. (26) and (27) hold true. A union bound shows that $\mathbb{P}(\Omega_x^\tau) \geqslant 1 - 3\mathrm{e}^{-x}$. Furthermore, combining Eq. (24), (26) and (27) shows that on Ω_x^τ, for every $\theta > 0$,

$$\mathrm{pen}_{\mathrm{id}}(\tau) \leqslant \frac{2M^2}{n}\left(D_\tau + \frac{14x}{3} + \frac{28}{3}\sqrt{2xD_\tau}\right) + \frac{2\theta}{n}\|\Pi_\tau\mu^\star - \mu^\star\|^2 + \frac{xM^2}{n}\left(\theta^{-1} + \frac{8}{3}\right)$$

$$\leqslant 2\theta\mathcal{R}(\widehat{\mu}_\tau) + \frac{M^2}{n}\left[2D_\tau + \left(\theta^{-1} + \frac{36}{3}\right)x + \frac{56}{3}\sqrt{2xD_\tau}\right] \quad (28)$$

using that $n^{-1}\|\Pi_\tau\mu^\star - \mu^\star\|^2 = \mathcal{R}(\Pi_\tau\mu^\star) \leqslant \mathcal{R}(\widehat{\mu}_\tau)$ by definition of the orthogonal projection Π_τ, and

$$\mathrm{pen}_{\mathrm{id}}(\tau) \geqslant -\frac{2}{n}\big\langle (I-\Pi_\tau)\mu^\star,\, \varepsilon\big\rangle$$

$$\geqslant -\frac{2\theta}{n}\|\Pi_\tau\mu^\star - \mu^\star\|^2 - \frac{xM^2}{n}\left(\theta^{-1} + \frac{8}{3}\right)$$

$$\geqslant -2\theta\mathcal{R}(\widehat{\mu}_\tau) - \frac{xM^2}{n}\left(\theta^{-1} + \frac{8}{3}\right). \quad (29)$$

5.5.2. Union Bound Over the Models and Conclusion

Let $y \geqslant 0$ be fixed and let us define the event $\Omega_y = \bigcap_{\tau\in\mathcal{T}_n}\Omega^\tau_{x(\tau,y)}$ where for every $\tau \in \mathcal{T}_n$,

$$x(\tau, y) := y + \log\left(\frac{3}{\mathrm{e}-1}\right) + D_\tau + \log\binom{n-1}{D_\tau - 1}.$$

Then, since

$$\mathrm{Card}\left\{\tau \in \mathcal{T}_n \,|\, D_\tau = D\right\} = \binom{n-1}{D-1}$$

for every $D \in \{1, \ldots, n\}$, a union bound shows that

$$\mathbb{P}(\Omega_y) \geqslant 1 - \sum_{\tau\in\mathcal{T}_n}\mathbb{P}\big(\overline{\Omega}^\tau_{x(\tau,y)}\big) \geqslant 1 - 3\sum_{D=1}^{n}\mathrm{e}^{-y-\log\left(\frac{3}{\mathrm{e}-1}\right)-D} = 1 - (\mathrm{e}-1)\mathrm{e}^{-y}\sum_{D=1}^{n}\mathrm{e}^{-D}$$

$$\geqslant 1 - \mathrm{e}^{-y}.$$

In addition, on Ω_y, for every $\tau \in \mathcal{T}_n$, Eq. (28) and (29) hold true with $x = x(\tau, y) \geqslant D_\tau$, hence, taking $\theta = 1/6$, we get that

$$-\frac{26}{3}\frac{M^2x(\tau,y)}{n} - \frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) \leqslant \mathrm{pen}_{\mathrm{id}}(\tau) \leqslant \frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) + \left(20 + \frac{56\sqrt{2}}{3}\right)\frac{M^2x(\tau,y)}{n}.$$

Let us define

$$\kappa_1 := 20 + \frac{56\sqrt{2}}{3} \qquad \text{and} \qquad \kappa_2 := \frac{26}{3},$$

and assume that $C \geqslant \kappa_1$. Then, using Eq. (10), we have

$$\mathrm{pen}_{\mathrm{id}}(\tau) \leqslant \frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) + \mathrm{pen}(\tau) + \frac{\kappa_1 M^2\left[y + \log\big(3/(\mathrm{e}-1)\big)\right]}{n}$$

$$\mathrm{pen}_{\mathrm{id}}(\tau) \geqslant -\frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) - \frac{\kappa_2}{C}\,\mathrm{pen}(\tau) - \frac{\kappa_2 M^2\left[y + \log\big(3/(\mathrm{e}-1)\big)\right]}{n}.$$

Therefore, by Eq. (13), on Ω_y, for every $\tau \in \mathcal{T}_n$,

$$\frac{2}{3}\mathcal{R}(\widehat{\mu}_{\widehat{\tau}}) - \frac{\kappa_1 M^2\left[y + \log\big(3/(\mathrm{e}-1)\big)\right]}{n}$$

$$\leqslant \frac{4}{3}\mathcal{R}(\widehat{\mu}_\tau) + \left(1 + \frac{\kappa_2}{C}\right)\mathrm{pen}(\tau) + \frac{\kappa_2 M^2\left[y + \log\big(3/(\mathrm{e}-1)\big)\right]}{n}$$

hence

$$\frac{2}{3}\mathcal{R}(\widehat{\mu}_{\widehat{\tau}}) \leqslant \frac{4}{3}\mathcal{R}(\widehat{\mu}_\tau) + \left(1 + \frac{\kappa_2}{C}\right)\mathrm{pen}(\tau) + \frac{(\kappa_1+\kappa_2)M^2\left[y + \log\big(3/(\mathrm{e}-1)\big)\right]}{n}$$

$$\leqslant \frac{4}{3}\mathcal{R}(\widehat{\mu}_\tau) + \left(1 + \frac{\kappa_2 + (\kappa_1+\kappa_2)\log\big(3/(\mathrm{e}-1)\big)}{C}\right)\mathrm{pen}(\tau) + (\kappa_1+\kappa_2)\frac{M^2y}{n}$$

since $\mathrm{pen}(\tau) \geqslant CM^2/n$ for every $\tau \in \mathcal{T}_n$. Multiplying both sides by 3/2, we get that if $C \geqslant \kappa_1$, on Ω_y,

$$\mathcal{R}(\widehat{\mu}_{\widehat{\tau}}) \leqslant \inf_{\tau \in \mathcal{T}_n} \left\{ 2\mathcal{R}(\widehat{\mu}_\tau) + \frac{3}{2}\left(1 + \frac{\kappa_2 + (\kappa_1+\kappa_2)\log\big(3/(\mathrm{e}-1)\big)}{C}\right)\mathrm{pen}(\tau)\right\} + \frac{3(\kappa_1+\kappa_2)}{2}\frac{M^2 y}{n}.$$

Let us finally define

$$L_1 := 3\Big[\kappa_2 + (\kappa_1+\kappa_2)\log\big(3/(\mathrm{e}-1)\big)\Big] \geqslant \kappa_1$$

so that

$$\frac{3}{2}\left(1 + \frac{\kappa_2 + (\kappa_1+\kappa_2)\log\big(3/(\mathrm{e}-1)\big)}{L_1}\right) = 2.$$

Then, we get that if $C \geqslant L_1$, on Ω_y,

$$\mathcal{R}(\widehat{\mu}_{\widehat{\tau}}) \leqslant 2 \inf_{\tau \in \mathcal{T}_n} \big\{\mathcal{R}(\widehat{\mu}_\tau) + \mathrm{pen}(\tau)\big\} + \frac{3(\kappa_1+\kappa_2)}{2}\frac{M^2 y}{n}$$

and the result follows. ■

6. Experiments on Synthetic Data

This section reports the results of some experiments on synthetic data that illustrate the performance of KCP.

6.1. Data-generation Process

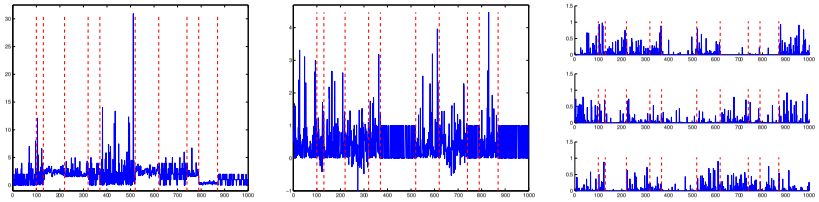
Three scenarios are considered: (1) real-valued data with a changing (mean,variance), (2) real-valued data with constant mean and variance, and (3) histogram-valued data as in Example 4.

In the three scenarios, the sample size is $n = 1\,000$ and the true segmentation $\tau^\star$ is made of $D^\star = 11$ segments, with change-points $\tau_1^\star = 100$, $\tau_2^\star = 130$, $\tau_3^\star = 220$, $\tau_4^\star = 320$, $\tau_5^\star = 370$, $\tau_6^\star = 520$, $\tau_7^\star = 620$, $\tau_8^\star = 740$, $\tau_9^\star = 790$, $\tau_{10}^\star = 870$ (see Figure 1). For each sample, we choose randomly the distribution of the X_i within each segment of $\tau^\star$ as detailed below; note that we always make sure that the distribution of X_i does change at each $\tau_\ell^\star$.

For each scenario, we generate $N = 500$ independent samples, from which we estimate all quantities that are reported in Section 6.3.

Scenario 1: Real-valued data with changing (mean, variance). The distribution of $X_i \in \mathbb{R}$ is randomly picked out from: $\mathcal{B}(10, 0.2)$ (binomial), $\mathcal{NB}(3, 0.7)$ (negative-binomial), $\mathcal{H}(10, 5, 2)$ (hypergeometric), $\mathcal{N}(2.5, 0.25)$ (Gaussian), $\gamma(0.5, 5)$ (gamma), $\mathcal{W}(5, 2)$ (Weibull) and $\mathcal{P}ar(1.5, 3)$ (Pareto). Note that the pair (mean, variance) in each segment changes from that of its neighbors. Table B.1 summarizes its values.

The distribution within segment $\ell \in \{1, \dots, D^\star\}$ is given by the realization of a random variable $S_\ell \in \{1, \dots, 7\}$, each integer representing one of the seven possible distributions.



(a) Scenario 1: changes in the pair (mean, variance)

(b) Scenario 2: constant mean and variance

(c) Scenario 3: histogram-valued data (first three coordinates)

Figure 1: Examples of generated signals (blue plain curve) in the three scenarios. Red vertical dashed lines visualize the true change-points locations.

The variables S_ℓ are generated as follows: S_1 is uniformly chosen among $\{1, \dots, 7\}$, and for every $\ell \in \{1, \dots, D^\star - 1\}$, given S_ℓ, $S_{\ell+1}$ is uniformly chosen among $\{1, \dots, 7\} \backslash \{S_\ell\}$. Figure 1a shows one sample generated according to this scenario.

Scenario 2: Real-valued data with constant mean and variance. The distribution of $X_i \in \mathbb{R}$ is randomly chosen among (1) $\mathcal{B}(0.5)$ (Bernoulli), (2) $\mathcal{N}(0.5, 0.25)$ (Gaussian) and (3) $\mathcal{E}(0.5)$ (exponential). These three distributions have a mean 0.5 and a variance 0.25.

The distribution within segment $\ell \in \{1, \dots, D^\star\}$ is given by the realization of a random variable $S_\ell \in \{1, 2, 3\}$, similarly to what is done in scenario 1 (replacing 7 by 3). Figure 1b shows one sample generated according to this scenario.

Scenario 3: Histogram-valued data. The observations X_i belong to the d-dimensional simplex with $d = 20$ (Example 4), that is, $X_i = (a_1, \dots, a_d) \in [0, 1]^d$ with $\sum_{j=1}^d a_j = 1$. For each $\ell \in \{1, \dots, D^\star\}$, we randomly generate d parameter values $p_1^\ell, \dots, p_d^\ell$ independently with uniform distribution over $[0, c_3]$ with $c_3 = 0.2$. Then, within the ℓ-th segment of $\tau^\star$, X_i follows a Dirichlet distribution with parameter $(p_1^\ell, \dots, p_d^\ell)$. Figure 1c displays the first three coordinates of one sample generated according to this scenario.

6.2. Parameters of KCP

For each sample, we apply the kernel change-point procedure (KCP, that is, Algorithm 1) with the following choices for its parameters. We always take $D_{\max} = 100$.

For the first two scenarios, we consider three kernels:

(i) The linear kernel $k^{\mathrm{lin}}(x, y) = xy$.

(ii) The Hermite kernel given by $k^{\mathrm{H}}_{\sigma_H}(x, y)$ defined in Section 3.2. In scenario 1, $\sigma_H = 1$. In scenario 2, $\sigma_H = 0.1$.

(iii) The Gaussian kernel $k^{\mathrm{G}}_{\sigma_G}$ defined in Section 3.2. In scenario 1, $\sigma_G = 0.1$. In scenario 2, $\sigma_G = 0.16$.

For scenario 3, we consider the χ^2 kernel $k^{\chi^2}_{0.1}(x, y)$ defined in Section 3.2, and the Gaussian kernel $k^{\mathrm{G}}_{\sigma_G}$ with $\sigma_G = 1$.

In each scenario several candidate values have been explored for the bandwidth parameters of the above kernels. We have selected the ones with the most representative results.

For choosing the constants c_1, c_2 arising from Step 2 of KCP, we use the "slope heuristics" method, and more precisely a variant proposed by Lebarbier (2002, Section 4.3.2) for the calibration of two constants for change-point detection. We first perform a linear regression of $\widehat{\mathcal{R}}_n(\widehat{\tau}(D))$ against $1/n \cdot \log\binom{n-1}{D-1}$ and D/n for $D \in [0.6 \times D_{\max}, D_{\max}]$. Then, denoting by $\widehat{s}_1, \widehat{s}_2$ the coefficients obtained, we define $c_i = -\alpha \widehat{s}_i$ for $i = 1, 2$, with $\alpha = 2$. The slope heuristics is justified theoretically in various settings (for instance by Arlot and Massart, 2009, for regressograms; see Arlot, 2019, for a survey) and is supported by numerous experiments (Baudry et al., 2012), including for change-point detection (Lebarbier, 2002, 2005). A partial theoretical justification has been obtained recently for change-point detection (Sorba, 2017). The intuition behind the slope heuristics is that the minimal amount of penalization needed for avoiding to overfit with $\widehat{\tau} \in \operatorname{argmin}_\tau\{\widehat{\mathcal{R}}_n(\tau) + \operatorname{pen}(\tau)\}$ and the optimal penalty (approximately) are proportional:

$$\operatorname{pen}_{\text{optimal}}(\tau) \approx \alpha \operatorname{pen}_{\text{minimal}}(\tau)$$

for some constant $\alpha > 1$, equal to 2 in several settings. The linear regression step described above corresponds to estimating the minimal penalty:

$$\operatorname{pen}_{\text{minimal}}(\tau) \approx -\widehat{s}_1 \cdot \frac{1}{n} \log\binom{n-1}{D_\tau - 1} - \widehat{s}_2 \frac{D_\tau}{n} \cdot$$

Then, multiplying it by α leads to an estimation of the optimal penalty. In the experiments, we considered several values of $\alpha \in [0.8, 2.5]$. Remarkably, the performance of the procedure is not too sensitive to the value of α provided $\alpha \in [1.7, 2.2]$. We only report the results for $\alpha = 2$ because it corresponds to the classical advice when using the slope heuristics, and it is among the best choices for α according to the experiments.

6.3. Results

We now summarize the results of the experiments.

6.3.1. Distance Between Segmentations

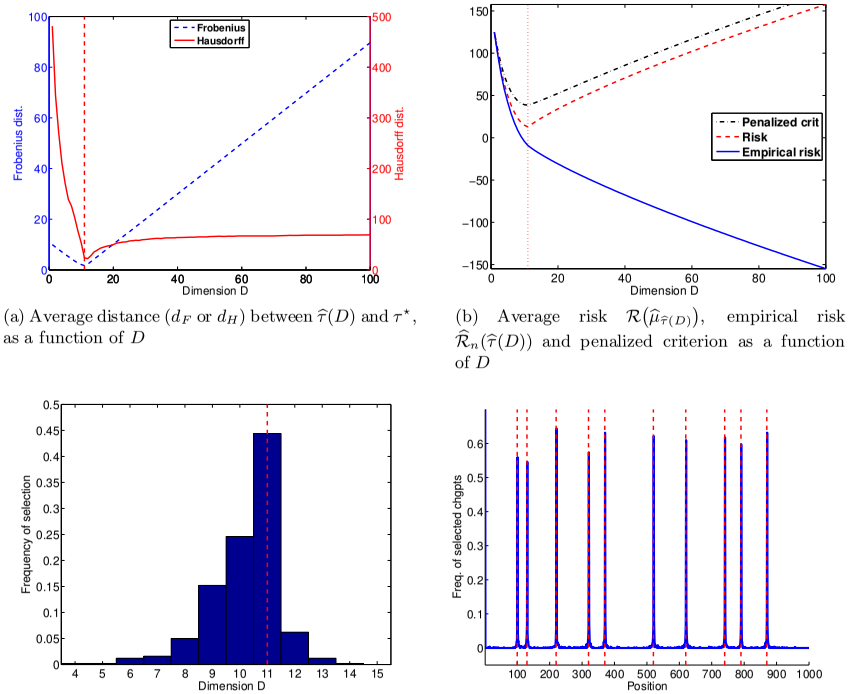
In order to assess the quality of the segmentation $\widehat{\tau}$ as an estimator of the true segmentation $\tau^\star$, we consider two measures of distance between segmentations. For any $\tau, \tau' \in \mathcal{T}_n$, we define the Hausdorff distance between τ and τ' by

$$d_H(\tau, \tau') := \max\left\{ \max_{1 \leqslant i \leqslant D_\tau - 1} \min_{1 \leqslant j \leqslant D_{\tau'} - 1} \left|\tau_i - \tau'_j\right|, \max_{1 \leqslant j \leqslant D_{\tau'} - 1} \min_{1 \leqslant i \leqslant D_\tau - 1} \left|\tau_i - \tau'_j\right| \right\}$$

and the Frobenius distance between τ and τ' (Lajugie et al., 2014) by

$$d_F(\tau, \tau') := \left\| M^\tau - M^{\tau'} \right\|_F = \sqrt{\sum_{1 \leqslant i,j \leqslant n} (M^\tau_{i,j} - M^{\tau'}_{i,j})^2},$$

$$\text{where} \qquad M^\tau_{i,j} = \frac{\mathbf{1}_{\{i \text{ and } j \text{ belong to the same segment of } \tau\}}}{\operatorname{Card}(\text{segment of } \tau \text{ containing } i \text{ and } j)}.$$



(a) Average distance (d_F or d_H) between $\widehat{\tau}(D)$ and $\tau^\star$, as a function of D

(b) Average risk $\mathcal{R}(\widehat{\mu}_{\widehat{\tau}(D)})$, empirical risk $\widehat{\mathcal{R}}_n(\widehat{\tau}(D))$ and penalized criterion as a function of D

(c) Distribution of $\widehat{D}$

(d) Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau} = \widehat{\tau}(\widehat{D})$ puts a change point at i

Figure 2: Scenario 1: $\mathcal{X} = \mathbb{R}$, variable (mean, variance). Performance of KCP with kernel $k^G_{0.1}$. The value $D^\star$ and the localization of the true change points in $\tau^\star$ are materialized by vertical red lines.

Note that $M^\tau = \Pi_\tau$ the projection matrix onto F_τ when $\mathcal{H} = \mathbb{R}$, that is, for the linear kernel on $\mathcal{X} = \mathbb{R}$. The Hausdorff distance is probably more classical in the change-point literature, but Figure 2a shows that the Frobenius distance is more informative for comparing $(\widehat{\tau}(D))_{D > D^\star}$. Indeed, when D is already a bit larger than $D^\star$, adding false change points makes the segmentation worse without increasing much d_H; on the contrary, d_F^2 readily takes into account these additional false change points.

6.3.2. Illustration of KCP

Figure 2 illustrates the typical behaviour of KCP when k is well-suited to the change-point problem we consider. It summarizes results obtained in scenario 1 with $k = k^{\mathrm{G}}$ the Gaussian kernel.

Figure 2a shows the expected distance between the true segmentation $\tau^\star$ and the segmentations $(\widehat{\tau}(D))_{1 \leqslant D \leqslant D_{\max}}$ produced at Step 1 of KCP. As expected, the distance is clearly minimal at $D = D^\star$, for both Hausdorff and Frobenius distances. Note that for each individual sample, $d(\widehat{\tau}(D), \tau^\star)$ behaves exactly as the expectation shown on Figure 2a, up to minor fluctuations. Moreover, the minimal value of the distance is small enough to suggest that $\widehat{\tau}(D^\star)$ is indeed close to $\tau^\star$. For instance, $\mathbb{E}[d_F(\widehat{\tau}(D^\star), \tau^\star)] \approx 1.71$, with a 95% error bar smaller than 0.11. The closeness between $\widehat{\tau}(D^\star)$ and $\tau^\star$ when $k = k^{\mathrm{G}}$ can also be visualized on Figure B.9c in Appendix B.4.

As a comparison, when $k = k^{\mathrm{lin}}$ in the same setting, $\widehat{\tau}(D^\star)$ is much further from $\tau^\star$ since $\mathbb{E}[d_F(\widehat{\tau}(D^\star), \tau^\star)] \approx 10.39 \pm 0.24$, and a permutation test shows that the difference is significant, with a p-value smaller than 10^{-13}. See also Figures B.7 and B.9a in Appendix B.4.

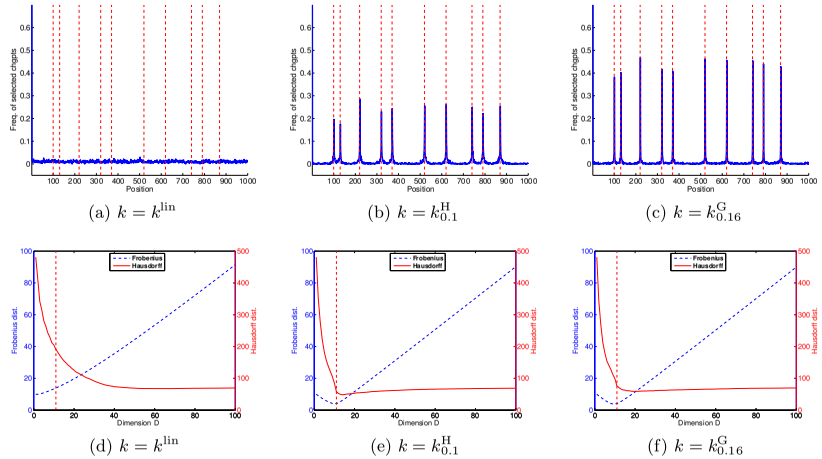
Step 2 of KCP is illustrated by Figures 2b and 2c. The expectation of the penalized criterion is minimal at $D = D^\star$ (as well as for the risk of $\widehat{\mu}_{\widehat{\tau}(D)}$), and takes significantly larger values when $D \neq D^\star$ (Figure 2b). As a result, KCP often selects a number of change points $\widehat{D} - 1$ close to its true value $D^\star - 1$ (Figure 2c). Overall, this suggests that the model-selection procedure used at Step 2 of KCP works fairly well.

The overall performance of KCP as a change-point detection procedure is illustrated by Figure 2d. Each true change-point has a probability larger than 0.5 to be recovered *exactly* by $\widehat{\tau}$. If one groups the positions i by blocks of six elements $\{6j, 6j+1, \ldots, 6j+5\}$, $j \geqslant 1$, the frequency of detection of a change point by $\widehat{\tau}$ in each block containing a true change point is between 79 and 89%. Importantly, such figures are obtained without overestimating much the number of change points, according to Figure 2c. Figures B.10a and B.10b in Appendix B.4 show that more standard change-point detection algorithms —that is, KCP with $k = k^{\mathrm{lin}}$ or k^{H}— have a slightly worse performance.

6.3.3. Comparison of Three Kernels in Scenario 2

Scenario 2 proposes a more challenging change-point problem with real-valued data: the distribution of the X_i changes while the mean *and* the variance remain constant. The performance of KCP with three kernels —k^{lin}, k^{H} and k^{G}— is shown on Figure 3. The linear kernel k^{lin} corresponds to the classical least-squares change-point algorithm (Lebarbier, 2005), which is designed to detect changes in the mean, hence it should fail in scenario 2. KCP with the Hermite kernel k^{H} is a natural "hand-made" extension of this classical approach, since it corresponds to applying the least-squares change-point algorithm to the feature vectors $(H_{j,h}(X_i))_{1 \leqslant j \leqslant 5}$. By construction, it should be able to detect changes in the first five moments on the X_i. On the contrary, taking $k = k^{\mathrm{G}}$ the Gaussian kernel fully relies on the versatility of KCP, which makes possible to consider (virtually) infinite-dimensional feature vectors $k^{\mathrm{G}}(X_i, \cdot)$. Since k^{G} is characteristic, it should be able to detect any change in the distribution of the X_i.

In order to compare these three kernels within KCP, let us first assume that the number of change points is known, hence we can estimate $\tau^\star$ with $\widehat{\tau}(D^\star)$, where D^* is the true



(a) $k = k^{\mathrm{lin}}$ (b) $k = k^{\mathrm{H}}_{0.1}$ (c) $k = k^{\mathrm{G}}_{0.16}$

(d) $k = k^{\mathrm{lin}}$ (e) $k = k^{\mathrm{H}}_{0.1}$ (f) $k = k^{\mathrm{G}}_{0.16}$

Figure 3: Scenario 2: $\mathcal{X} = \mathbb{R}$, constant mean and variance. Performance of KCP with three different kernels k. The value $D^\star$ and the localization of the true change points in $\tau^\star$ are materialized by vertical red lines. **Top:** Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau}(D^\star)$ puts a change point at i. **Bottom:** Average distance (d_F or d_H) between $\widehat{\tau}(D)$ and $\tau^\star$, as a function of D.

number of segments. Then, Figures 3a, 3b and 3c show that k^{lin}, k^{H} and k^{G} behave as expected: k^{lin} seems to put the change points of $\widehat{\tau}(D^\star)$ uniformly at random over $\{1, \ldots, n\}$, while k^{H} and k^{G} are able to localize the true change points with a rather large probability of success. The Gaussian kernel here shows a significantly better detection power, compared to k^{H}: the frequency of exact detection of the true change points is between 38 and 47% with k^{G}, and between 17 and 29% with k^{H}. The same holds when considering blocks of size 6: k^{G} then detects the change points with probability 70 to 79%, while k^{H} exhibits probabilities between 58 and 62%.

Figures 3d, 3e and 3f show that a similar comparison between k^{lin}, k^{H}, and k^{G} holds over the whole set of segmentations $(\widehat{\tau}(D))_{1 \leqslant D \leqslant D_{\max}}$ provided by Step 1 of KCP. With the linear kernel (Figure 3d), the Frobenius distance between $\widehat{\tau}(D)$ and $\tau^\star$ is almost minimal for $D = 1$, which suggests that $\widehat{\tau}(D)$ is not far from random guessing for all D. The shape of the Hausdorff distance —first decreasing fastly, then almost constant— also supports this interpretation: A small number of purely random guesses do lead to a fast decrease of d_H ; and for large dimensions, adding a new random guess does not move away $\widehat{\tau}(D)$ from $\tau^\star$ if $\widehat{\tau}(D)$ already contains all the worst possible candidate change points (which are the furthest from the true change points). The Hermite kernel does much better according to Figure 3e: the Frobenius distance from $\widehat{\tau}(D)$ to $\tau^\star$ is minimal for D close to $D^\star$, and the minimal expected distance, $\inf_D \mathbb{E}[d_F(\widehat{\tau}(D), \tau^\star)] \approx 4.12 \pm 0.6$ (with confidence 95%), is

much smaller than when $k = k^{\mathrm{lin}}$ (in which case $\inf_D \mathbb{E}[d_F(\widehat{\tau}(D), \tau^\star)] \approx 10$); this difference is significant (a permutation test yields a p-value smaller than 10^{-15}). Nevertheless, we still obtain slightly better performance for $(\widehat{\tau}(D))_{1 \leqslant D \leqslant D_{\max}}$ with $k = k^{\mathrm{G}}$, for which the minimal distance to $\tau^\star$ is achieved at $D = 9$, with a minimal expected value equal to 3.83 ± 0.49 (the difference between k^{H} and k^{G} is not statistically significant). The Hausdorff distance suggests that both k^{H} and k^{G} lead to include false change points among true ones as long as $D \leqslant D^\star$. However, the smaller Frobenius distance achieved by k^{G} at $D = 9$ (rather than $D = 11$ for k^{H}) indicates that the corresponding change points are closer to the true ones than those provided by k^{H} (which include more false positives).

When $D = \widehat{D}$ is chosen by KCP, k^{G} clearly leads to the best performance in terms of recovering the exact change points compared to k^{lin} and k^{H}, as illustrated by Figures B.11a, B.11b and B.11c in Appendix B.4.

Overall, the best performance for KCP in scenario 2 is clearly obtained with k^{G}, while k^{lin} completely fails and k^{H} yields a decent but suboptimal procedure.

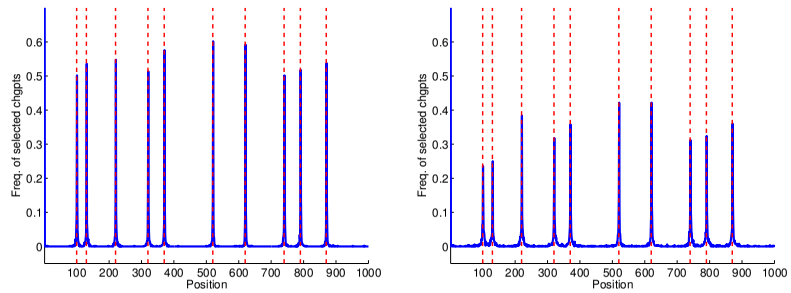
We can notice that other settings can lead to different behaviours. For instance, in scenario 1, according to Figure B.10a in Appendix B.4, k^{lin} can detect fairly well the true change points —as expected since the mean (almost) always changes in this scenario, see Table B.1 in Appendix B.4—, but this is at the price of a strong overestimation of the number of change points (Figure B.8a). In the same setting, k^{H} provides fairly good results (Figure B.10b), while k^{G} remains the best choice (Figure 2d).

Since k^{G} is a characteristic kernel, these results suggest that KCP with a characteristic kernel k might be more versatile than classical least-squares change-point algorithms and their extensions. A more detailed simulation experiment would nevertheless be needed to confirm this hypothesis. We also refer to Section 8.2 for a discussion on the choice of k for a given change-point problem.

6.3.4. Structured Data

Figure 4 illustrates the performance of KCP on some histogram-valued data (scenario 3). Since a d-dimensional histogram is also an element of $\mathbb{R}^d$, we can analyze such data either with a kernel taking into account the histogram structure (such as k^{χ^2}) or with a usual kernel on $\mathbb{R}^d$ (such as k^{lin} or k^{G}; here, we consider k^{G}, which seems more reliable according to our experiments in scenarios 1 and 2). Assuming that the number of change points is known, taking $k = k^{\chi^2}$ yields quite good results according to Figure 4a, at least in comparison with $k = k^{\mathrm{G}}$ (Figure 4b). Similar results hold with a fully data-driven number of change points, as shown by Figures B.13a and B.13b in Appendix B.4. Hence, choosing a kernel such as k^{χ^2}, which takes into account the histogram structure of the X_i, can improve much the change-point detection performance, compared to taking a kernel such as k^{G}, which ignores the structure of the X_i.

Let us emphasize that scenario 3 is quite challenging —changes are hard to distinguish on Figure 1c—, which has been chosen on purpose. Preliminary experiments have been done with larger values of c_3 —which makes the change-point problem easier, see Section 6.1—, leading to an almost perfect localization of all change points by KCP with $k = k_{0.1}^{\chi^2}$.



(a) $k = k_{0.1}^{\chi^2}$ (b) $k = k_1^{\mathrm{G}}$

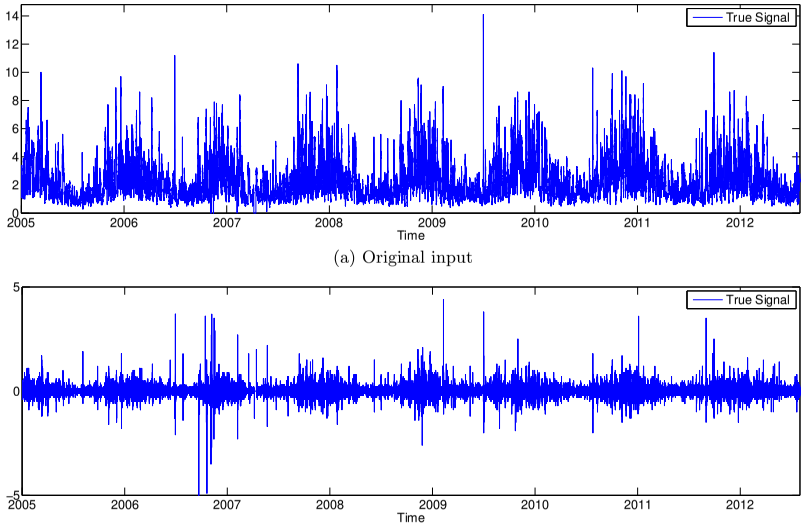
Figure 4: Scenario 3: histogram-valued data. Performance of KCP with two different kernels k. Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau}(D^\star)$ puts a change point at i. Vertical red lines show the true change-points locations.

6.3.5. Comparison to AIC/BIC-type Penalty

Figures B.14 and B.15 in Appendix B.4 show the results of KCP with a linear penalty —that is, of the form CD/n, $C > 0$— in step 2, similarly to AIC (which would correspond to $C = \sigma^2$) and BIC (for which $C = \log(n)\sigma^2/2$). Since σ^2 is unknown, we use the slope heuristics for choosing C from data, as explained in Section 6.2. The performance is comparable to the one of KCP, except that a linear penalty leads to overfitting —by detecting too many change points (including false positives)— with a large probability in scenarios 1 and 2 (compare Figures B.14a and B.8c for scenario 1, and Figures B.14b and B.12c for scenario 2). Therefore, a linear penalty seems less reliable than the refined shape proposed in the definition of KCP, so we do not recommend to use a linear penalty in practice.

6.3.6. Comparison to the E-divisive Procedure (ED)

We finally consider the e-divisive procedure (ED) designed by Matteson and James (2014), focusing on scenarios 1–2 since this procedure is made for $\mathcal{X} = \mathbb{R}^d$ only. We use the `e.divisive` function from the `R`-package `ecp` described by James and Matteson (2015), with recommended parameters `sig.lvl` $= 0.05$ (significance level to test any new change point), $\alpha = 1$, and $R = 199$ permutations. Detailed results are shown on Figures B.16, B.17 and B.18 in Appendix B.4. In both scenarios, ED provides much more conservative results —that is, it strongly underestimates the number of change points— compared to KCP with $k = k^{\mathrm{G}}$. This drawback of ED is particularly clear in scenario 2 (more difficult) from the comparison of Figures B.12c and B.16b. As a result, ED's detection power is much smaller than the one of KCP, with detection frequencies 2 to 5 times lower for ED in scenario 2 (see Figures B.11c and B.17b). The performance of ED improves when $D^\star$ is given to the algorithm, but KCP remains significantly better than ED in terms of detection power (see



(a) Original input

(b) First-difference based signal

Figure 5: Wave-heights time-series collected between January 2005 and September 2012

Figures 3c and B.18b, for instance). Overall, KCP with $k = k^{\mathrm{G}}$ clearly outperforms ED in scenarios 1–2, which can be explained by at least two reasons: (i) ED uses a different similarity measures than ours; (ii) ED relies on a greedy strategy, in which $\widehat{\tau}(D+1)$ is obtained from $\widehat{\tau}(D)$ by adding one change point, so that any mistake at the beginning of the process impacts the final segmentation.

7. Real-data Experiment

This section reports the results of KCP applied on some real data set.

7.1. Data Description

In this section, we illustrate the behavior of KCP on a publicly available data set corresponding to wave heights hourly-measured between January 2005 and September 2012 at a location in Northern Atlantic (Killick et al., 2012, Section 4.2). This leads to a large sample of length $n = 63\,651$.

This data set exhibits a strong difference between the wave heights during winters (high level) and summers (low level), as one can see on Figure 5a. Plotting the first-difference based signal, following Killick et al. (2012, Figure 4), highlights strong changes in the variance of the signal (Figure 5b). Automatically detecting the change points between such successive periods is of primary interest for analyzing the environmental conditions of offshore wind farms for instance.

7.2. Procedures Compared

This section details the parameters used for the different procedures compared.

7.2.1. KCP

We apply KCP (Algorithm 1) on the original data (Figure 5a) with the Gaussian kernel and bandwidth parameter equal to the empirical standard deviation of the data $\sigma_G = 1.3526$ — we here take the same data-driven bandwidth choice as Celisse et al. (2018). The maximum number of segments is set to $D_{\max} = 50$, which seems to be large enough since about 14 changes only are expected along this period of almost 7 years. The numerical constants $c_1, c_2 \geqslant 0$ are estimated by using the slope heuristics as detailed in Section 6.2.

7.2.2. ED

The e-divisive procedure (ED, see Section 6.3.6) from Matteson and James (2014) is applied on the original data (Figure 5a) with its default parameter values: `sig.lvl` $= 0.05$, $\alpha = 1$, and $R = 199$ permutations.

7.2.3. PELT

The so-called PELT procedure (Killick et al., 2012) is considered by means of the function `cpt.var` implemented in the `R` package `changepoint` (Killick and Eckley, 2014). It is applied to the first-difference based signal (Figure 5b) as done by Killick et al. (2012), because this procedure is built for detecting changes in the variance of a zero-mean Gaussian signal.

7.3. Results

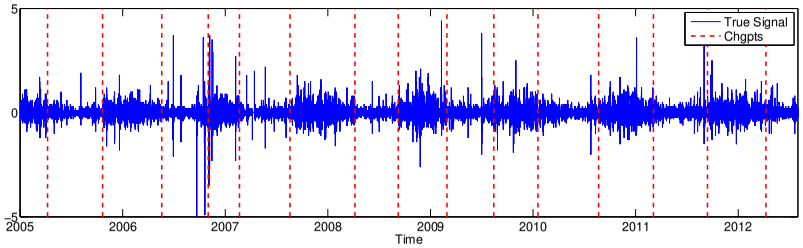
Figure 6 displays the estimated change points (red vertical dashed lines) output by KCP (Figure 6a) and PELT (Figure 6b). The segmentation output by ED only contains one segment (no change point), which is consistent with the trend of ED towards underestimating the number of changes.

KCP outputs 16 homogeneous segments which do not coincide with changes of the mean as one could have feared. PELT outputs 17 segments which are mainly similar to the ones of KCP. Both results are realistic, as explained by Killick et al. (2012).

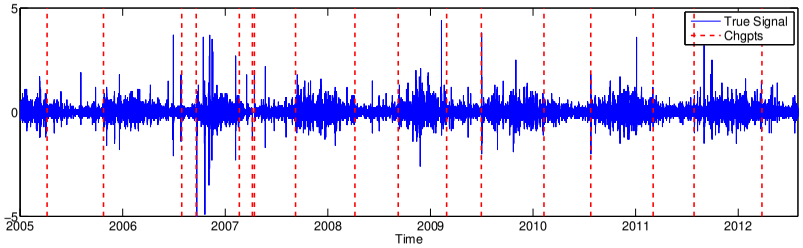
Nevertheless there are a few differences around the year 2007 where PELT detects very narrow segments, which are likely related to outliers. It still arises that the fourth change-point location estimated by KCP is somewhat questionable, compared to the one output by PELT, since it coincides with a strong change in the variance which seems, by eye, to have started a bit sooner.

A striking feature of KCP remains that no *a priori* specification has been made about the type of changes we are looking for. This strongly contrasts with the PELT procedure, which makes a Gaussian assumption and relies on the prior knowledge that changes occur in the variance of the signal only (in this example).

The overall conclusion is that KCP here provides reliable results without requiring any side information —apart from the choice of a kernel, see Section 8.2— about the data distribution and the nature of its changes. This turns out to be a strong asset when



(a) KCP



(b) PELT

Figure 6: Segmentations of the wave-heights time-series output by KCP and PELT, respectively (red vertical dashed lines) and first-difference based signal (blue). Note that KCP is applied to the *original* data (see text).

analyzing real data, for which any distributional assumption is misleading when it happens to be violated.

8. Conclusion

This paper describes a kernel change-point algorithm (KCP, that is, Algorithm 1), based upon a penalization procedure generalizing the one of Comte and Rozenholc (2004) and Lebarbier (2005) to RKHS-valued data. Such an extension significantly broadens the range of possible applications of the algorithm, since it can deal with complex or structured data, and it can detect changes in the full distribution of the data —not only the mean or the variance. The new theoretical tools developed in the paper —mostly, a concentration inequality for some function of RKHS-valued random variables (Proposition 1)— may be useful for related statistical problems such as clustering in reproducing kernel Hilbert spaces or functional-data analysis. Let us now end the paper with three questions about KCP: one that has been solved while this paper was under review, and two that are still open.

8.1. Identification of the Change-point Locations

A natural question for a change-point algorithm is its consistency for estimating the true change-point locations $\tau^\star$. More precisely, let us assume that some $\tau^\star \in \mathcal{T}_n$ exists such that

$$P_{X_{\tau^\star_{\ell-1}+1}} = \cdots = P_{X_{\tau^\star_\ell}} \qquad \text{for } 1 \leqslant \ell \leqslant D_{\tau^\star}\,, \qquad P_{X_{\tau^\star_\ell}} \neq P_{X_{\tau^\star_\ell+1}} \qquad \text{for } 1 \leqslant \ell \leqslant D_{\tau^\star} - 1$$

and $D_{\tau^\star}$ is fixed as n tends to infinity (even if $\tau^\star$ necessarily depends on n). The goal is to prove that $d(\widehat{\tau}, \tau^\star)$ tends to zero almost surely as n tends to infinity, where d is some distance on $\mathcal{T}_n$, for instance $n^{-1}d_F$ or $n^{-1}d_H$ as defined in Section 6.3. Many papers prove such a consistency result for other change-point algorithms in various settings (for instance, Yao, 1988; Lavielle and Moulines, 2000; Frick et al., 2014; Matteson and James, 2014). Answering this question for KCP is beyond the scope of the paper. The change-point estimation consistency of KCP has been proved by Garreau and Arlot (2018) under mild assumptions, after the first version of this work appeared as a preprint.

8.2. Choosing the Kernel k

A major practical and theoretical question about KCP is the choice of the kernel k. Fully answering this question is beyond the scope of the paper, but we can already provide a few guidelines, based upon the theoretical and experimental results that we already have, and review some previous works tackling a related question.

First, simulation experiments in Section 6 show that the performance of KCP can strongly vary with k. They suggest that using a characteristic kernel —such as the Gaussian kernel k^{G}— yields a more versatile procedure when the goal is to detect changes in the full distribution of the data. Nevertheless, for a given change-point problem, all characteristic kernels certainly are not equivalent. For instance, unshown experimental results suggest that k^{G}_h with a clearly bad choice of the bandwidth h —say, smaller than 10^{-4} or larger than 10^4 in settings similar to scenario 1— leads to a poor performance of KCP, despite the fact that k^{G}_h is characteristic for any $h > 0$.

Furthermore, for a given setting, a non characteristic kernel can be a good choice: when the goal is to detect changes in the mean of $X_i \in \mathbb{R}^d$, k^{lin} is known to work very well (Lebarbier, 2005). Cabrieto et al. (2018b,a) also show that KCP can be used for focusing on changes in the correlation (resp. autocorrelation) structure of multivariate time series.

Second, the theoretical interpretation of KCP in Section 4.2 suggests how the performance of KCP depends on k, hence on which basis k should be chosen. Indeed, KCP focuses on changes in the mean $\mu^\star_1, \ldots, \mu^\star_n$ of the time series $Y_1, \ldots, Y_n \in \mathcal{H}$. A change between P_{X_i} and $P_{X_{i+1}}$ should be detected more easily when

$$\|\mu^\star_{i+1} - \mu^\star_i\|^2_{\mathcal{H}} = \mathbb{E}\big[k(X_{i+1}, X_{i+1})\big] - 2\mathbb{E}\big[k(X_{i+1}, X_i)\big] + \mathbb{E}\big[k(X_i, X_i)\big]$$

is larger, compared to the "noise level" $\max\{v_i, v_{i+1}\}$. When $P_{X_i} \neq P_{X_{i+1}}$, we know that $\|\mu^\star_{i+1} - \mu^\star_i\|_{\mathcal{H}}$ is positive for any characteristic kernel k, while it might be equal to zero when k is not characteristic. But the fact that k is characteristic or not is not sufficient to guess whether k will work well or not, according to the above heuristic.

The problem of choosing a kernel has been considered for many different tasks in the machine-learning literature. Let us only mention here some references that are tackling this question in a framework close to change-point detection: choosing the best kernel for a two-sample or an homogeneity test.

For choosing the bandwidth h of a Gaussian kernel, a classical heuristic —called the median heuristic— is to take h equal to some median of $(\|X_i - X_j\|)_{i<j}$, see Gretton et al. (2012a, Section 8, and references therein) and Garreau et al. (2018).

A procedure for choosing the best convex combination of a finite number of kernels is proposed by Gretton et al. (2012b), with the goal of building a powerful two-sample test. Another idea for combining several kernels, for instance the family $\{k_h^{\mathrm{G}} : h > 0\}$, is studied by Sriperumbudur et al. (2009) for homogeneity and independence tests. Roughly, the idea is to replace the MMD test statistics —which depends on a kernel k— by its supremum over the considered family of kernels. Nevertheless, the extension of these two ideas to change-point detection with KCP does not seem straightforward.

Let us now discuss the choice of the bandwidth of a Gaussian kernel for KCP. If n is large, the median heuristic can require a large computation time. When $X_i \in \mathbb{R}$, the empirical standard deviation of $\{X_1, \ldots, X_n\}$ is a good proxy to it, easy to compute on large data sets. It is used successfully with KCP by Celisse et al. (2018), which is the reason why we use it in Section 7.

Nevertheless, using the median heuristic (or a proxy) with KCP may be questionable in general, since two-sample test and multiple change-point detection are different tasks. For instance, when the mean of the $X_i \in \mathbb{R}$ has large jumps, the median-heuristic bandwidth can be much larger than the standard deviation of the X_i , so that it may not work as well. In such cases, another option to consider would be some median of $(\|X_{i+1} - X_i\|)_{1 \leqslant i \leqslant n-1}$, which could be studied in future works on KCP.

8.3. Heteroscedasticity of Data in $\mathcal{H}$

A possible drawback of KCP is that it does not take into account the fact that the variance v_i of $Y_i = \Phi(X_i)$ can change with i: in general, the Y_i are heteroscedastic. In the case of real-valued data and the linear kernel k^{lin}, Arlot and Celisse (2011) show that heteroscedastic data can make KCP fail, and that this failure cannot be fixed by changing the penalty used at Step 2: all the segmentations $\widehat{\tau}(D)$ produced at Step 1 can be wrong.

We conjecture that, for the Gaussian kernel k_h^{G} at least, when the bandwidth h is well chosen, the variances of the Y_i stay within a reasonably small range of values for most non-degenerate distributions. Indeed, according to Eq. (8),

$$v_i = 1 - \mathbb{E}\left[\exp\left(\frac{-\|X_i - X_i'\|_{\mathcal{H}}^2}{2h^2}\right)\right] \in [0, 1]$$

where X_i' is an independent copy of X_i . If X_i is not deterministic and if h is smaller than the typical order of magnitude of $\|X_i - X_i'\|_{\mathcal{H}}$, then, v_i cannot be much smaller than its maximal value 1. The simulation experiments suggest that "good" values of h for change-point detection are small enough, but this remains to be proved.

When heteroscedasticity is a problem for KCP, which probably occurs for some kernels beyond k^{lin}, we can think of combining KCP with the ideas of Arlot and Celisse (2011),

that is, replacing the empirical risk and the penalized criterion in Steps 1 and 2 of KCP by cross-validation estimators of the risk $\mathcal{R}(\widehat{\mu}_\tau)$.

Acknowledgments

The authors thank Damien Garreau for some discussions that lead to an improvement of the theoretical results —namely, Proposition 1 and Theorem 2, which were stated with the additional assumption that $\min_i v_i \geqslant cM^2 > 0$ in a previous version of this paper (Arlot et al., 2012).
This work was mostly done while Sylvain Arlot was financed by CNRS and member of the Sierra team in the Departement d'Informatique de l'Ecole normale superieure (CNRS/ENS/INRIA UMR 8548), 45 rue d'Ulm, F-75230 Paris Cedex 05, France, and Zaid Harchaoui was a member of the LEAR team of Inria. Sylvain Arlot and Alain Celisse were also supported by Institut des Hautes Études Scientifiques (IHES, Le Bois-Marie, 35, route de Chartres, 91440 Bures-Sur-Yvette, France) at the end of the writing of this paper. Sylvain Arlot is also member of the Celeste project-team of Inria Saclay.
The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR) under reference ANR-09-JCJC-0027-01 (Detect project) and ANR-14-CE23-0003-01 (Macaron project), the GARGANTUA project funded by the Mastodons program of CNRS, the LabEx Persyval-Lab (ANR-11-LABX-0025), the BeFast project funded by the PEPS Fascido program of CNRS, the Moore-Sloan Data Science Environment at NYU, the NSF DMS-1810975 grant, and faculty research awards.

Appendix A. Additional Proofs

This section provides the remaining proofs of the results, except the proof of Proposition 4 which is delayed to Appendix B.2.

A.1. Proofs of Section 4.1

This section proves two results stated in Section 4.1: Eq. (5) and Eq. (6).

A.1.1. Proof of Eq. (5)

Let $f \in F_\tau$ and $g \in \mathcal{H}^n$. For any $\ell \in [\![1, D_\tau]\!]$, we define $I_\ell^\tau := [\![\tau_{\ell-1} + 1, \tau_\ell]\!]$ the ℓ-th interval of τ, $f_{I_\ell^\tau}$ the common value of $(f_i)_{i \in I_\ell^\tau}$ and

$$\overline{g}_{I_\ell^\tau} := \frac{1}{\mathrm{Card}(I_\ell^\tau)} \sum_{i \in I_\ell^\tau} g_i = \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{i \in I_\ell^\tau} g_i \,. \tag{30}$$

Then,

$$\|f-g\|^2 = \sum_{\ell=1}^{D_\tau}\sum_{i\in I_\ell^\tau}\left[\left\|f_{I_\ell^\tau}-\overline{g}_{I_\ell^\tau}\right\|_{\mathcal{H}}^2 + \left\|g_i-\overline{g}_{I_\ell^\tau}\right\|_{\mathcal{H}}^2 + 2\left\langle f_{I_\ell^\tau}-\overline{g}_{I_\ell^\tau},\, \overline{g}_{I_\ell^\tau}-g_i\right\rangle_{\mathcal{H}}\right]$$
$$= \sum_{\ell=1}^{D_\tau}\left[(\tau_\ell-\tau_{\ell-1})\left\|f_{I_\ell^\tau}-\overline{g}_{I_\ell^\tau}\right\|_{\mathcal{H}}^2\right] + \sum_{\ell=1}^{D_\tau}\sum_{i\in I_\ell^\tau}\left\|g_i-\overline{g}_{I_\ell^\tau}\right\|_{\mathcal{H}}^2$$

since $\sum_{i\in I_\ell^\tau}(\overline{g}_{I_\ell^\tau}-g_i)=0$. So, $\|f-g\|^2$ is minimal over $f\in F_\tau$ if and only if $f_{I_\ell^\tau}=\overline{g}_{I_\ell^\tau}$ for every $\ell\in[\![1,D_\tau]\!]$. ■

A.1.2. Proof of Eq. (6)

We use the notation introduced in the proof of Eq. (5). Then,

$$\|Y-\widehat{\mu}_\tau\|^2 = \sum_{\ell=1}^{D_\tau}\sum_{i\in I_\ell^\tau}\|Y_i-\overline{Y}_{I_\ell^\tau}\|_{\mathcal{H}}^2 = \sum_{\ell=1}^{D_\tau}\sum_{i\in I_\ell^\tau}\left(\|Y_i\|_{\mathcal{H}}^2-\|\overline{Y}_{I_\ell^\tau}\|_{\mathcal{H}}^2\right)$$

where we use Eq. (5) for the first equality, and that

$$\sum_{i\in I_\ell^\tau}\left\langle Y_i,\, \overline{Y}_{I_\ell^\tau}\right\rangle_{\mathcal{H}} = \mathrm{Card}(I_\ell^\tau)\left\|\overline{Y}_{I_\ell^\tau}\right\|_{\mathcal{H}}^2$$

for the second equality. Therefore,

$$\|Y-\widehat{\mu}_\tau\|^2 = \sum_{i=1}^{n}\|Y_i\|_{\mathcal{H}}^2 - \sum_{\ell=1}^{D_\tau}\frac{1}{\tau_\ell-\tau_{\ell-1}}\left\|\sum_{i\in I_\ell^\tau}Y_i\right\|_{\mathcal{H}}^2$$
$$= \sum_{i=1}^{n}\|Y_i\|_{\mathcal{H}}^2 - \sum_{\ell=1}^{D_\tau}\frac{1}{\tau_\ell-\tau_{\ell-1}}\sum_{i,j\in I_\ell^\tau}\langle Y_i,\, Y_j\rangle_{\mathcal{H}}$$
$$= \sum_{i=1}^{n}k(X_i,X_i) - \sum_{\ell=1}^{D_\tau}\frac{1}{\tau_\ell-\tau_{\ell-1}}\sum_{i,j\in I_\ell^\tau}k(X_i,X_j)\,,$$

which proves Eq. (6). ■

A.2. Concentration of the Linear Term: Proof of Proposition 3

Let us define $\mu_\tau^\star = \Pi_\tau\mu^\star$ and

$$S_\tau = \langle \mu^\star-\mu_\tau^\star,\, \varepsilon\rangle = \sum_{i=1}^{n}Z_i \quad\text{with}\quad Z_i = \left\langle(\mu^\star-\mu_\tau^\star)_i,\, \varepsilon_i\right\rangle_{\mathcal{H}}.$$

The Z_is are independent and centered, so Eq. (32)–(33) in Lemma 5 below —which requires assumption (**Db**)— show that the conditions of Bernstein's inequality are satisfied (see

Proposition 6 in Appendix B.1). Therefore for every $x\geqslant 0$, with probability at least $1-2\mathrm{e}^{-x}$,

$$\left|\sum_{i=1}^{n}Z_i\right| \leqslant \sqrt{2v_{\max}\|\mu^\star-\mu_\tau^\star\|^2\,x} + \frac{4M^2x}{3}$$
$$\leqslant \theta\,\|\mu^\star-\mu_\tau^\star\|^2 + \left(\frac{v_{\max}}{2\theta}+\frac{4M^2}{3}\right)x$$

for every $\theta>0$, using $2ab\leqslant\theta a^2+\theta^{-1}b^2$. ■

A key argument in the proof is the following lemma.

Lemma 5 *For every $m\in\mathcal{M}_n$, if (**Db**) holds true, the following holds true with probability one:*

$$\forall i\in\{1,\dots,n\}\,,\quad \|\mu_i^\star\|_{\mathcal{H}}\leqslant M\,,\qquad \|\varepsilon_i\|_{\mathcal{H}}\leqslant 2M \tag{31}$$

$$\text{and}\quad \|(\mu^\star-\mu_\tau^\star)_i\|_{\mathcal{H}}\leqslant 2M \quad\text{so that}\quad |Z_i|\leqslant 4M^2\,. \tag{32}$$

$$\text{In addition,}\quad \sum_{i=1}^{n}\mathrm{Var}\,(Z_i)\leqslant v_{\max}\,\|\mu^\star-\mu_\tau^\star\|^2\,. \tag{33}$$

Proof First, remark that for every i,

$$v_i = \mathbb{E}\big[\|\varepsilon_i\|^2\big] = \mathbb{E}\big[k(X_i,X_i)\big] - \|\mu_i^\star\|_{\mathcal{H}}^2 \geqslant 0\,,$$

so that with (**Db**),

$$\|\mu_i^\star\|_{\mathcal{H}}^2 \leqslant \mathbb{E}\big[k(X_i,X_i)\big]\leqslant M^2\,,$$

which proves the first bound in Eq. (31). As a consequence, by the triangular inequality,

$$\|\varepsilon_i\|_{\mathcal{H}}\leqslant\|Y_i\|_{\mathcal{H}}+\|\mu_i^\star\|_{\mathcal{H}}\leqslant 2M\,,$$

that is, the second inequality in Eq. (31) holds true.

Let us now define for every $i\in\{1,\dots,n\}$, the integer $K(i)\in\{1,\dots,D_\tau\}$ such that $I_{K(i)}^\tau=[\![\tau_{K(i)-1}+1,\tau_{K(i)}]\!]$ is the unique interval of the segmentation τ such that $i\in I_{K(i)}^\tau$. Then,

$$(\mu^\star-\mu_\tau^\star)_i = \frac{1}{\tau_{K(i)}-\tau_{K(i)-1}}\sum_{j\in I_{K(i)}^\tau}(\mu_i^\star-\mu_j^\star)\,,$$

so that the triangular inequality and Eq. (31) imply

$$\|(\mu^\star-\mu_\tau^\star)_i\|_{\mathcal{H}}\leqslant\sup_{j\in I_{K(i)}^\tau}\left\|\mu_i^\star-\mu_j^\star\right\|_{\mathcal{H}}\leqslant\sup_{1\leqslant j,k\leqslant n}\left\|\mu_k^\star-\mu_j^\star\right\|_{\mathcal{H}}\leqslant 2\sup_{1\leqslant j\leqslant n}\left\|\mu_j^\star\right\|_{\mathcal{H}}\leqslant 2M\,,$$

that is, the first part of Eq. (32) holds true. The second part of Eq. (32) directly follows from Cauchy-Schwarz's inequality. For proving Eq. (33), we remark that

$$\mathbb{E}\left[Z_i^2\right] = \mathbb{E}\Big[\left\langle(\mu^\star-\mu_\tau^\star)_i,\, \varepsilon_i\right\rangle_{\mathcal{H}}^2\Big]$$
$$\leqslant\left\|(\mu^\star-\mu_\tau^\star)_i\right\|_{\mathcal{H}}^2\mathbb{E}\left[\|\varepsilon_i\|_{\mathcal{H}}^2\right] \qquad\text{by Cauchy-Schwarz's inequality}$$
$$= \left\|(\mu^\star-\mu_\tau^\star)_i\right\|_{\mathcal{H}}^2 v_i\leqslant\left\|(\mu^\star-\mu_\tau^\star)_i\right\|_{\mathcal{H}}^2 v_{\max}\,,$$

so that $\displaystyle\sum_{i=1}^{n}\mathrm{Var}\,(Z_i)\leqslant v_{\max}\,\|\mu^\star-\mu_\tau^\star\|^2\,.$

■

Appendix B. Supplementary Material

This section provides the statements of classical concentration inequalities that are used in the proofs, the proof of an auxiliary result (Proposition 4 in Section 5.4.3), another method for choosing c_1 and c_2 in KCP, and more details about the experiments of Section 6.

B.1. Classical Concentration Inequalities

This section collects a few classical results that are used throughout the paper.

B.1.1. Bernstein's Inequality

We state below Berntein's inequality, as formulated by Massart (2007, Proposition 2.9).

Proposition 6 (Bernstein's inequality) *Let $X_1, \dots, X_n$ be independent real-valued random variables. Assume that some positive constants v and c exist such that, for every $k \geqslant 2$*

$$\sum_{i=1}^{n} \mathbb{E}\Big[|X_i|^k\Big] \leqslant \frac{k!}{2} v c^{k-2} . \tag{34}$$

Then, for every $x > 0$,

$$\mathbb{P}\left(\sum_{i=1}^{n} \big(X_i - \mathbb{E}[X_i]\big) > \sqrt{2vx} + cx\right) \leqslant \mathrm{e}^{-x} .$$

In particular, if for every $i \in \{1, \dots, n\}$, $|X_i| \leqslant 3c$ almost surely, Eq. (34) holds true with $v = \sum_{i=1}^{n} \mathrm{Var}(X_i)$.

B.1.2. Pinelis-Sakhanenko's Inequality

Proposition 7 (Pinelis and Sakhanenko, 1986, Corollary 1) *Let $X_1, \dots, X_n$ be independent random variables with values in some Hilbert space $\mathcal{H}$. Assume the X_i are centered and that constants $\sigma^2, c > 0$ exist such that for every $p \geqslant 2$,*

$$\sum_{i=1}^{n} \mathbb{E}\left[\|X_i\|_{\mathcal{H}}^p\right] \leqslant \frac{p!}{2} \sigma^2 c^{p-2} .$$

Then, for every $x > 0$,

$$\mathbb{P}\left(\left\|\sum_{i=1}^{n} X_i\right\|_{\mathcal{H}} > x\right) \leqslant 2 \exp\left[-\frac{x^2}{2\left(\sigma^2 + cx\right)}\right] .$$

B.1.3. Talagrand's Inequality

The following proposition is a refined version of Talagrand's concentration inequality (Talagrand, 1996), as it is stated by Boucheron et al. (2013, Corollary 12.12).

Proposition 8 (Boucheron et al., 2013, Corollary 12.12) *Let $X_1, \dots, X_n$ be independent vector-valued random variables and let*

$$Z = \sup_{f \in \mathcal{F}} \sum_{i=1}^{n} X_{i,f} .$$

Assume that for all $i \in \{1, \dots, n\}$ and $f \in \mathcal{F}$, $\mathbb{E}[X_{i,f}] = 0$ and $|X_{i,f}| \leqslant 1$. Define

$$\sigma^2 = \sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathbb{E}\left[X_{i,f}^2\right] \qquad \text{and} \qquad v = 2\mathbb{E}[Z] + \sigma^2 .$$

Then, for all $x \geqslant 0$,

$$\mathbb{P}\Big(Z \geqslant \mathbb{E}[Z] + \sqrt{2vx} + \frac{x}{3}\Big) \leqslant \mathrm{e}^{-x} \tag{35}$$

$$\text{and} \qquad \mathbb{P}\Big(Z \leqslant \mathbb{E}[Z] - \sqrt{2vx} - \frac{x}{8}\Big) \leqslant \mathrm{e}^{-x} . \tag{36}$$

B.2. Proof of Proposition 4

The first step is to write $\|\Pi_\tau \varepsilon\|$ of the form of Z in Proposition 8 for some well-chosen $(X_{i,f})_{1 \leqslant i \leqslant n,\, f \in \mathcal{G}_\tau}$. Defining

$$\overline{f}_K = \frac{1}{\tau_K - \tau_{K-1}} \sum_{i=\tau_{K-1}+1}^{\tau_K} f_i \qquad \text{for every } 1 \leqslant K \leqslant D_\tau ,$$

it comes

$$\begin{aligned}
\|\Pi_\tau \varepsilon\| &= \sup_{f \in \mathcal{H}^n,\, \|f\| \leqslant 1} \big|\langle f,\, \Pi_\tau \varepsilon\rangle\big| \\
&= \sup_{f \in \mathcal{H}^n,\, \|f\| \leqslant 1} \big|\langle \Pi_\tau f,\, \varepsilon\rangle\big| \\
&= \sup_{f \in \mathcal{H}^n,\, \sum_{K=1}^{D_\tau} (\tau_K - \tau_{K-1}) \|\overline{f}_K\|^2 \leqslant 1} \left| \sum_{K=1}^{D_\tau} \sum_{i=\tau_{K-1}+1}^{\tau_K} \langle \overline{f}_K,\, \varepsilon_i\rangle_{\mathcal{H}} \right| \\
&= \sup_{f \in \mathcal{G}_\tau} \sum_{i=1}^{n} \overline{X}_{i,f}
\end{aligned}$$

where $\mathcal{G}_\tau$ is some countable dense subset of

$$\left\{ f \in \mathcal{H}^n,\ \sum_{K=1}^{D_\tau} (\tau_K - \tau_{K-1}) \left\|\overline{f}_K\right\|_{\mathcal{H}}^2 \leqslant 1 \right\}$$

(such a set $\mathcal{G}_\tau$ exists since $\mathcal{H}$ is separable), and for every $i \in \{1, \dots, n\}$ and $f \in \mathcal{G}_\tau$,

$$\overline{X}_{i,f} = \left\langle \overline{f}_{K(i)}, \varepsilon_i \right\rangle_{\mathcal{H}}$$

where we recall that $K(i)$ is defined in the proof of Lemma 5 in Appendix A.2.

Let us now check that the assumptions of Proposition 8 are satisfied: $(\overline{X}_{1,f})_{f\in\mathcal{G}_\tau}, \dots, (\overline{X}_{n,f})_{f\in\mathcal{G}_\tau}$ are independent since $\varepsilon_1, \dots, \varepsilon_n$ are assumed independent. For every $i \in \{1, \dots, n\}$ and $f \in \mathcal{G}_\tau$,

$$\mathbb{E}\left[\overline{X}_{i,f}\right] = \mathbb{E}\left[\left\langle \overline{f}_{K(i)}, \varepsilon_i \right\rangle_{\mathcal{H}}\right] = 0$$

since $\overline{f}_{K(i)} \in \mathcal{H}$ is deterministic and for every $f \in \mathcal{G}_\tau$,

$$\left|\overline{X}_{i,f}\right| = \left|\left\langle \overline{f}_{K(i)}, \varepsilon_i \right\rangle_{\mathcal{H}}\right| \leqslant \left\|\overline{f}_{K(i)}\right\|_{\mathcal{H}} \|\varepsilon_i\|_{\mathcal{H}} \leqslant \frac{2M}{\sqrt{\tau_{K(i)} - \tau_{K(i)-1}}} \leqslant 2M$$

by Cauchy-Schwarz's inequality, assumption (**Db**) and Lemma 5. So, we can apply Proposition 8 to

$$Z = \frac{1}{2M} \|\Pi_\tau \varepsilon\| = \sup_{f\in\mathcal{G}_\tau} \sum_{i=1}^n X_{i,f}$$

where $X_{i,f} := (2M)^{-1}\overline{X}_{i,f}$.

Before writing the resulting concentration inequality, let us first compute (and bound) the quantity denoted by σ^2 in the statement of Proposition 8. For every $f \in \mathcal{G}_\tau$,

$$\begin{aligned}
4M^2 \sum_{i=1}^n \mathbb{E}\left[X_{i,f}^2\right] = \sum_{i=1}^n \mathbb{E}\left[\left\langle \overline{f}_{K(i)}, \varepsilon_i \right\rangle_{\mathcal{H}}^2\right] &\leqslant \sum_{i=1}^n \left[\left\|\overline{f}_{K(i)}\right\|_{\mathcal{H}}^2 \mathbb{E}\left[\|\varepsilon_i\|_{\mathcal{H}}^2\right]\right] \\
&= \sum_{K=1}^{D_\tau} \left[\|\overline{f}_K\|_{\mathcal{H}}^2 \sum_{i=\tau_{K-1}+1}^{\tau_K} v_i\right] \\
&= \sum_{K=1}^{D_\tau} \left[(\tau_K - \tau_{K-1}) \left\|\overline{f}_K\right\|_{\mathcal{H}}^2 v_K^\tau\right]
\end{aligned}$$

by Cauchy-Schwarz's inequality. So, by definition of $\mathcal{G}_\tau$ and σ^2,

$$\sigma^2 \leqslant \frac{1}{4M^2} \max_{1\leqslant K\leqslant D_\tau} v_K^\tau .$$

We can now write what Proposition 8 proves about the concentration of $\|\Pi_\tau\varepsilon\|$: for every $x \geqslant 0$, with probability at least $1 - \mathrm{e}^{-x}$,

$$\|\Pi_\tau\varepsilon\| - \mathbb{E}\big[\|\Pi_\tau\varepsilon\|\big] \leqslant 2M\sqrt{2vx} + \frac{2Mx}{3} \leqslant \sqrt{2x\left(4M\mathbb{E}\big[\|\Pi_\tau\varepsilon\|\big] + \max_{1\leqslant K\leqslant D_\tau} v_K^\tau\right)} + \frac{2Mx}{3},$$

and similarly, with probability at least $1 - \mathrm{e}^{-x}$,

$$\|\Pi_\tau\varepsilon\| - \mathbb{E}\big[\|\Pi_\tau\varepsilon\|\big] \geqslant -\sqrt{2x\left(4M\mathbb{E}\big[\|\Pi_\tau\varepsilon\|\big] + \max_{1\leqslant K\leqslant D_\tau} v_K^\tau\right)} - \frac{Mx}{4}.$$

So, using a union bound, we have just proved Eq. (23). ■

B.3. Second Method for Choosing c_1 and c_2 in KCP

We describe an alternative to the slope heuristics for choosing c_1, c_2 in KCP.

When prior information guarantee that the "variance" is almost constant and that no change occurs in some parts of the observed time series —say, at the start and at the end—, we can estimate this "variance" within each of these parts and take $c_1 = c_2$ equal to

$$\widehat{c}_{var} := 2\max\big(\widehat{v}_s, \widehat{v}_e\big), \tag{37}$$

where

$$\widehat{v}_s := \frac{1}{|I_s| - 1} \sum_{i\in I_s} \left[k(X_i, X_i) + \frac{1}{|I_s|^2} \sum_{j,\ell\in I_s} k(X_j, X_\ell) - \frac{2}{|I_s|} \sum_{j\in I_s} k(X_i, X_j)\right]$$

denotes the empirical variance of the start $(X_i)_{i\in I_s}$ of the time series, and $\widehat{v}_e$ is defined similarly from the end $(X_i)_{i\in I_e}$ of the time series. The fact that an estimate of the variance multiplied by 2 is a good choice for $c_1 = c_2$ is justified by the numerical experiments made by Lebarbier (2005) in the case of the linear kernel and one-dimensional data. This strategy was used successfully in the real-data experiments of an earlier version of the present paper (Arlot et al., 2012, Section 6.2).

Distribution	Mean	Variance
$\mathcal{B}(10, 0.2)$	2	1.6
$\mathcal{NB}(3, 0.7)$	$9/7 \approx 1.29$	$90/49 \approx 1.84$
$\mathcal{H}(10, 5, 2)$	1	$4/9 \approx 0.44$
$\mathcal{N}(2.5, 0.25)$	2.5	0.25
$\gamma(0.5, 5)$	2.5	12.5
$\mathcal{W}(5, 2)$	$\frac{5\sqrt{\pi}}{2} \approx 4.43$	$25(1 - \frac{\pi}{4}) \approx 5.37$
$\mathcal{P}ar(1.5, 3)$	$9/4 = 2.25$	$27/16 \approx 1.69$

Table B.1: Scenario 1, mean and variance for the seven distributions considered

B.4. Additional Details About the Synthetic Experiments

This section provides more details about the experiments on synthetic data (Section 6).

B.4.1. Data Generation Process

Table B.1 provides the values of the mean and variance of the seven distribution considered in scenario 1. It shows that the pair (mean, variance) changes at every change point in scenario 1, but the mean sometimes stays constant.

B.4.2. Further Results on Synthetic Data

This section gathers some additional results concerning the experiments of Section 6.

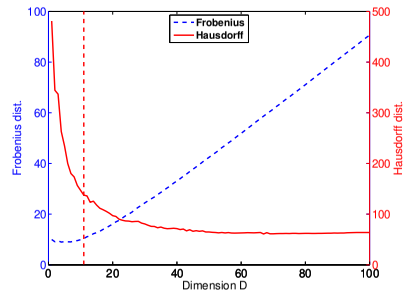
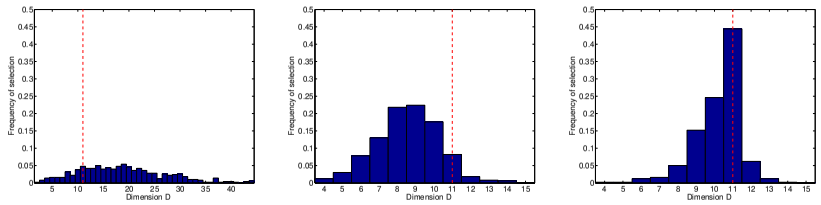
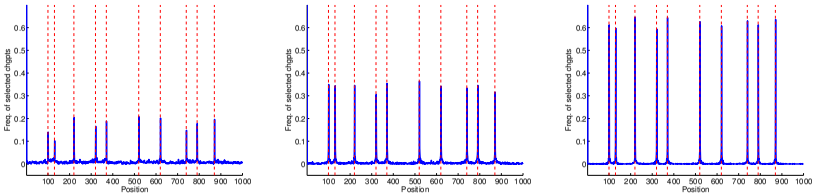


Figure B.7: Scenario 1: $\mathcal{X} = \mathbb{R}$, variable (mean, variance). Performance of KCP with kernel $k = k^{\text{lin}}$. Average distance (d_F or d_H) between $\widehat{\tau}(D)$ and $\tau^\star$, as a function of D.



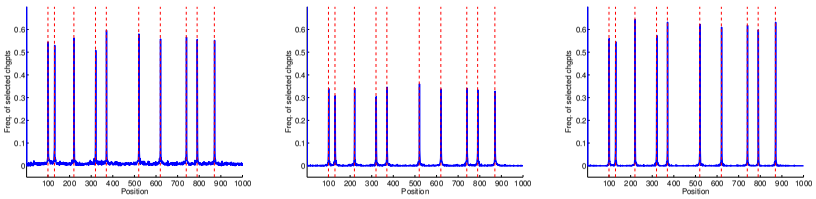
(a) $k = k^{\text{lin}}$ (b) $k = k_1^{\text{H}}$ (c) $k = k_{0.1}^{\text{G}}$

Figure B.8: Scenario 1: $\mathcal{X} = \mathbb{R}$, variable (mean, variance). KCP with three different kernels k. Distribution of $\widehat{D}$.
Figure B.8c is a copy of Figure 2c, that we repeat here for making comparisons easier.



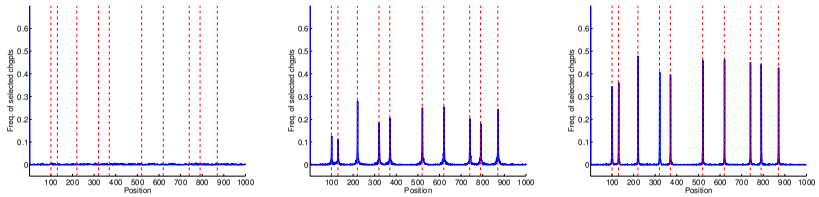
(a) $k = k^{\mathrm{lin}}$ (b) $k = k_1^{\mathrm{H}}$ (c) $k = k_{0.1}^{\mathrm{G}}$

Figure B.9: Scenario 1: $\mathcal{X} = \mathbb{R}$, variable (mean, variance). Performance of KCP with three different kernels k. Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau}(D^\star)$ puts a change point at i.



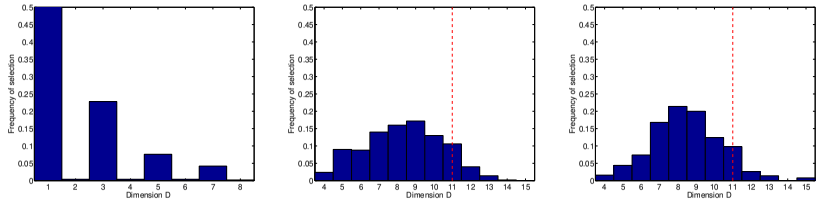
(a) $k = k^{\mathrm{lin}}$ (b) $k = k_1^{\mathrm{H}}$ (c) $k = k_{0.1}^{\mathrm{G}}$

Figure B.10: Scenario 1: $\mathcal{X} = \mathbb{R}$, variable (mean, variance). Performance of KCP with three different kernels k. Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau} = \widehat{\tau}(\widehat{D})$ puts a change point at i.
For $k = k^{\mathrm{lin}}$, notice the high "baseline" level of (wrong) detection of change points, which is due to a frequent overestimation of the number of change points, see Figure B.8a.
Figure B.10c is a copy of Figure 2d, that we repeat here for making comparisons easier.



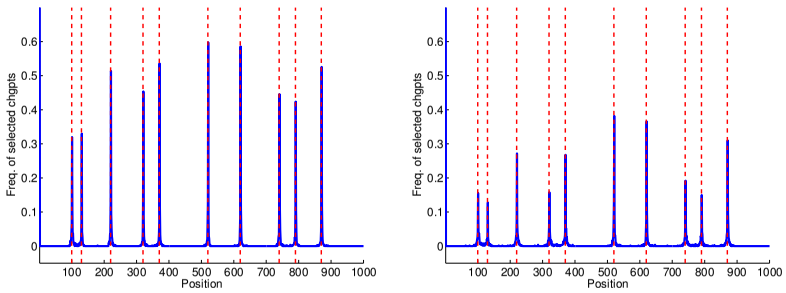
(a) $k = k^{\mathrm{lin}}$ (b) $k = k_{0.1}^{\mathrm{H}}$ (c) $k = k_{0.16}^{\mathrm{G}}$

Figure B.11: Scenario 2: $\mathcal{X} = \mathbb{R}$, constant mean and variance. Performance of KCP with three different kernels k. Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau} = \widehat{\tau}(\widehat{D})$ puts a change point at i.



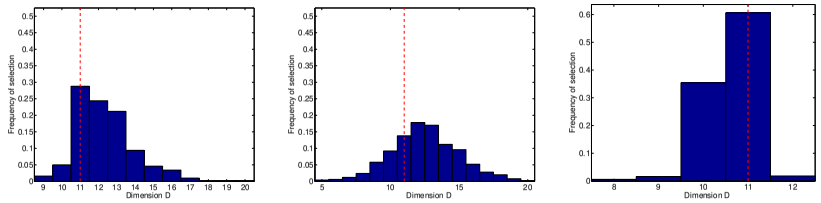
(a) $k = k^{\mathrm{lin}}$ (b) $k = k_{0.1}^{\mathrm{H}}$ (c) $k = k_{0.16}^{\mathrm{G}}$

Figure B.12: Scenario 2: $\mathcal{X} = \mathbb{R}$, constant mean and variance. KCP with three different kernels k. Distribution of $\widehat{D}$.



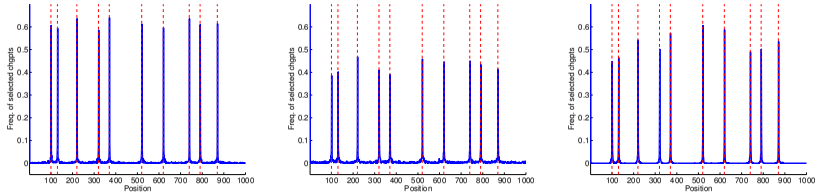
(a) $k = k^{\chi^2}_{0.1}$ (b) $k = k^{\mathrm{G}}_{1}$

Figure B.13: Scenario 3: histogram-valued data. Performance of KCP with two different kernels. Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau} = \widehat{\tau}(\widehat{D})$ puts a change point at i.



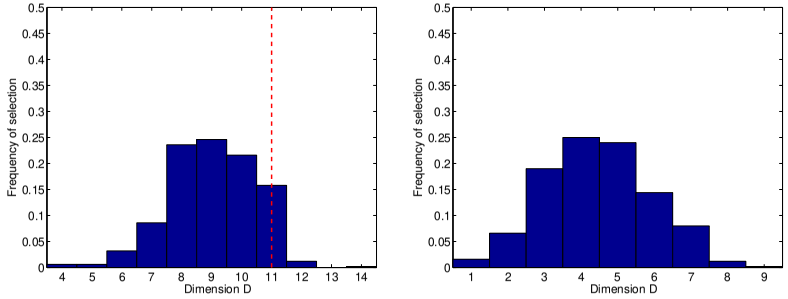
(a) Scenario 1, $k = k^{\mathrm{G}}_{0.1}$ (b) Scenario 2, $k = k^{\mathrm{G}}_{0.16}$ (c) Scenario 3, $k = k^{\chi^2}_{0.1}$

Figure B.14: KCP with a linear penalty (see Section 6.3.5): distribution of $\widehat{D}$



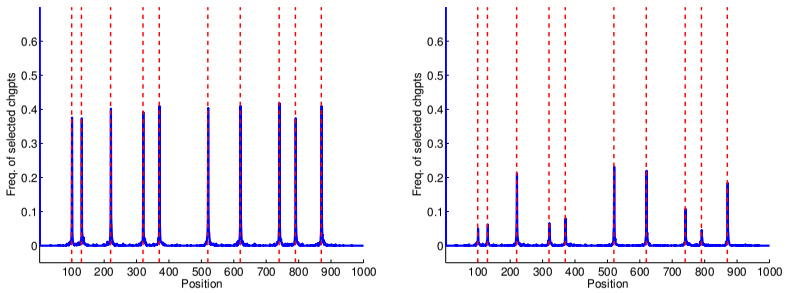
(a) Scenario 1, $k = k^{\mathrm{G}}_{0.1}$ (b) Scenario 2, $k = k^{\mathrm{G}}_{0.16}$ (c) Scenario 3, $k = k^{\chi^2}_{0.1}$

Figure B.15: KCP with a linear penalty (see Section 6.3.5): probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau} = \widehat{\tau}(\widehat{D})$ puts a change point at i



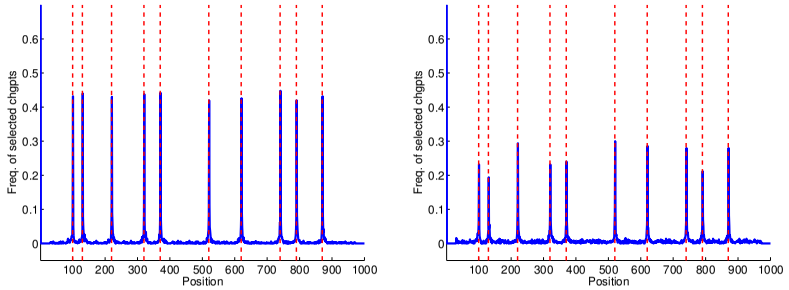
(a) Scenario 1 (b) Scenario 2

Figure B.16: E-divisive procedure (ED, see Section 6.3.6) with type-I error level `sig.lvl` = 0.05, $\alpha = 1$, and $R = 199$: distribution of $\widehat{D}$, the number of segments selected



(a) Scenario 1 (b) Scenario 2

Figure B.17: E-divisive procedure (ED, see Section 6.3.6) with type-I error level `sig.lvl` = 0.05, $\alpha = 1$, and $R = 199$: probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau}_{\mathrm{ED}}$ puts a change point at i



(a) Scenario 1 (b) Scenario 2

Figure B.18: E-divisive procedure (ED, see Section 6.3.6) with $\alpha = 1$ and $D = D^\star = 11$ known: probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau}_{\mathrm{ED}}(D^\star)$ puts a change point at i

References

Nathalie Akakpo. Estimating a discrete distribution via histogram selection. *ESAIM: Probability and Statistics*, 15:1–29, 2011.

Mokhtar Z. Alaya, Stéphane Gaïffas, and Agathe Guilloux. Learning the intensity of time events with change-points. *IEEE Transactions on Information Theory*, 61(9):5148–5171, 2015.

Stephanie Allen, David Madras, Ye Ye, and Greg Zanotti. Change-point detection methods for body-worn video. *SIURO*, 10, 2017. doi: 10.1137/16S015656.

Elena Andreou and Eric Ghysels. Detecting multiple breaks in financial market volatility dynamics. *Journal of Applied Econometrics*, 17(5):579–600, 2002.

Cécile Ané. Detecting phylogenetic breakpoints and discordance from genome-wide alignments for species tree reconstruction. *Genome Biology and Evolution*, 3:246–258, 2011.

Sylvain Arlot. *Contributions to Statistical Learning Theory: Estimator Selection and Change-point Detection*. Habilitation à diriger des recherches, University Paris Diderot, December 2014. Available at `http://tel.archives-ouvertes.fr/tel-01094989`.

Sylvain Arlot. Minimal penalties and the slope heuristics: a survey. *Journal de la Société Française de Statistique*, 160(3):1–106, 2019. With discussion and rejoinder.

Sylvain Arlot and Alain Celisse. Segmentation of the mean of heteroscedastic data via cross-validation. *Statistics and Computing*, 21(4):613–632, 2011.

Sylvain Arlot and Pascal Massart. Data-driven calibration of penalties for least-squares regression. *Journal of Machine Learning*, 10:245–279, 2009.

Sylvain Arlot, Alain Celisse, and Zaïd Harchaoui. Kernel change-point detection, February 2012. arXiv:1202.3878v1.

Ivan E. Auger and Charles E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54, 1989.

Yannick Baraud, Christophe Giraud, and Sylvie Huet. Gaussian model selection with an unknown variance. *The Annals of Statistics*, pages 630–672, 2009.

Jean-Marc Bardet and Imen Kammoun. Detecting abrupt changes of the long-range dependence or the self-similarity of a gaussian process. *Comptes Rendus Mathematique*, 346 (13):789–794, 2008.

Jean-Marc Bardet, William Chakry Kengne, and Olivier Wintenberger. Multiple breaks detection in general causal time series using penalized quasi-likelihood. *Electronic Journal of Statistics*, 6:435–477 (electronic), 2012.

Jean-Patrick Baudry, Cathy Maugis, and Bertrand Michel. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470, 2012.

István Berkes, Robertas Gabrys, Lajos Horváth, and Piotr Kokoszka. Detecting changes in the mean of functional observations. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 71(5):927–946, 2009.

Karine Bertin, Xavier Collilieux, Émilie Lebarbier, and Cristian Meza. Semi-parametric segmentation of multiple series using a DP-Lasso strategy. *Journal of Statistical Computation and Simulation*, 87(6):1255–1268, 2017.

Gérard Biau, Kevin Bleakley, and David Mason. Long signal change-point detection. *Electronic Journal of Statistics*, 10(2):2097–2123, 2016.

Lucien Birgé and Pascal Massart. Gaussian model selection. *Journal of the European Mathematical Society (JEMS)*, 3(3):203–268, 2001.

Kevin Bleakley and Jean-Philippe Vert. The group fused lasso for multiple change-point detection, 2011. arXiv:1106.4199.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, Oxford, 2013.

Leif Boysen, Angela Kempe, Volkmar Liebscher, Axel Munk, and Olaf Wittich. Consistencies and rates of convergence of jump-penalized least squares estimators. *The Annals of Statistics*, 37(1):157–183, 2009.

Boris E. Brodsky and Boris S. Darkhovsky. *Nonparametric Methods in Change-point Problems*, volume 243 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1993.

Jedelyn Cabrieto, Francis Tuerlinckx, Peter Kuppens, Mariel Grassmann, and Eva Ceulemans. Detecting correlation changes in multivariate time series: A comparison of four non-parametric change point detection methods. *Behavior Research Methods*, 49(3):988–1005, June 2017.

Jedelyn Cabrieto, Janne Adolf, Francis Tuerlinckx, Peter Kuppens, and Eva Ceulemans. Detecting long-lived autodependency changes in a multivariate system via change point detection and regime switching models. *Scientific Reports*, 8(15637), 2018a.

Jedelyn Cabrieto, Francis Tuerlinckx, Peter Kuppens, Frank H. Wilhelm, Michael Liedlgruber, and Eva Ceulemans. Capturing correlation changes by applying kernel change point detection on the running correlations. *Information Sciences*, (447):117–139, 2018b.

Edward Carlstein, Hans-Georg Müller, and David Siegmund, editors. *Change-point Problems*. IMS Lect. Notes, 1994.

Alain Celisse, Guillemette Marot, Morgane Pierre-Jean, and Guillem Rigaill. New efficient algorithms for multiple change-point detection with reproducing kernels. *Computational Statistics and Data Analysis*, 128:200–220, 2018.

Jinyuan Chang, Bin Guo, and Qiwei Yao. Principal component analysis for second-order stationary vector time series. *The Annals of Statistics*, 46(5):2094–2124, 2018.

Hao Chen and Nancy Zhang. Graph-based change-point detection. *The Annals of Statistics*, 43(1):139–176, 2015.

Kacper P. Chwialkowski, Dino Sejdinovic, and Arthur Gretton. A wild bootstrap for degenerate kernel tests. In *Advances in Neural Information Processing Systems 27*, pages 3608–3616. Curran Associates, Inc., 2014.

Alice Cleynen and Émilie Lebarbier. Segmentation of the poisson and negative binomial rate models: a penalized estimator. *ESAIM: Probability and Statistics*, 18:750–769, 2014.

Alice Cleynen and Émilie Lebarbier. Model selection for the segmentation of multiparameter exponential family distributions. *Electronic Journal of Statistics*, 11(1):800–842, 2017.

Xavier Collilieux, Emilie Lebarbier, and Stéphane Robin. A factor model approach for the joint segmentation with between-series correlation. *Scandinavian Journal of Statistics*, 46(3):686–705, 2019.

Fabienne Comte and Yves Rozenholc. A new algorithm for fixed design regression and denoising. *Annals of the Institute of Statistical Mathematics*, 56(3):449–473, 2004.

Costas Cotsaces, Nikos Nikolaidis, and Ioannis Pitas. Video shot boundary detection and condensed representation: a review. *IEEE Signal Processing Magazine*, 23(2):28–37, 2006.

Felipe Cucker and Ding Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

Ross E. Curtis, Jing Xiang, Ankur Parikh, Peter Kinnaird, and Eric P. Xing. Enabling dynamic network analysis through visualization in TVNViewer. *BMC Bioinformatics*, 13 (204), 2012.

Marco Cuturi, Kenji Fukumizu, and Jean-Philippe Vert. Semigroup kernels on measures. *Journal of Machine Learning Research (JMLR)*, 6:1169–1198, 2005.

Aymeric Dieuleveut and Francis Bach. Non-parametric stochastic approximation with large step sizes, 2014. arXiv:1408.0361v1.

Frédéric Ferraty and Philippe Vieu. *Nonparametric Functional Data Analysis*. Springer Series in Statistics. Springer, New York, 2006.

Walter D. Fisher. On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53:789–798, 1958.

Klaus Frick, Axel Munk, and Hannes Sieling. Multiscale change point inference. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 76(3):495–580, 2014.

Magalie Fromont, Béatrice Laurent, Matthieu Lerasle, and Patricia Reynaud-Bouret. Kernels based tests with non-asymptotic bootstrap approaches for two-sample problems. In *JMLR W& CP (COLT 2012)*, volume 23, pages 23.1–23.23, 2012.

Piotr Fryzlewicz. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.

Piotr Fryzlewicz and Suhasini Subba Rao. Multiple-change-point detection for auto-regressive conditional heteroscedastic processes. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 76(5):903–924, 2014.

Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research (JMLR)*, 5:73–99, 2004a.

Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Kernel dimensionality reduction for supervised learning. In *Advances in Neural Information Processing Systems 16*, pages 81–88. MIT Press, 2004b.

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20*, pages 489–496. Curran Associates, Inc., 2008.

Damien Garreau and Sylvain Arlot. Consistent change-point detection with kernels. *Electronic Journal of Statistics*, 12(2):4440–4486, 2018. Preliminary versions available at arXiv:1612.04740.

Damien Garreau, Wittawat Jitkrittum, and Motonobu Kanagawa. Large sample analysis of the median heuristic, 2018. arXiv:1707.07269.

Thomas Gärtner. *Kernels for Structured Data*, volume 72 of *Series in Machine Perception and Artificial Intelligence*. WorldScientific, 2008.

Vladimir J. Geneus, Jordan Cuevas, Eric Chicken, and J Pignatiello. A changepoint detection method for profile variance. In *Industrial and Systems Engineering Research Conference*, pages 1–7. 2015. Preliminary version available at arXiv:1408.7000.

Irene Gijbels, Peter Hall, and Aloïs Kneip. On the estimation of jump points in smooth curves. *Annals of the Institute of Statistical Mathematics*, 51(2):231–251, 1999.

Evarist Giné and Richard Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2016.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13(Mar):723–773, 2012a.

Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1205–1213. Curran Associates, Inc., 2012b.

Zaïd Harchaoui and Olivier Cappé. Retrospective change-point estimation with kernels. In *IEEE Workshop on Statistical Signal Processing*, 2007.

Zaïd Harchaoui, Francis Bach, and Éric Moulines. Testing for Homogeneity with Kernel Fisher Discriminant Analysis, April 2008. Available at `http://hal.archives-ouvertes.fr/hal-00270806/`.

Toby Hocking, Guillem Rigaill, Jean-Philippe Vert, and Francis Bach. Learning sparse penalties for change-point detection using max margin interval regression. In *International Conference on Machine Learning (ICML)*, pages 172–180, 2013.

Christian Houdré and Patricia Reynaud-Bouret. Exponential inequalities, with constants, for U-statistics of order two. In *Stochastic Inequalities and Applications*, volume 56 of *Progr. Probab.*, pages 55–69. Birkhäuser, Basel, 2003.

Nicholas A. James and David S. Matteson. ecp: An R package for nonparametric multiple change point analysis of multivariate data. *Journal of Statistical Software*, 62(i07), 2015.

Corinne Jones and Zaid Harchaoui. Chapydette. Available at `https://github.com/cjones6/chapydette`, 2019.

Steven M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice-Hall, Inc., 1993.

Rebecca Killick and Idris Eckley. changepoint: An R package for changepoint analysis. *Journal of Statistical Software*, 58(3):1–19, 2014.

Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107 (500):1590–1598, 2012.

L. Lacey Knowles and Laura S. Kubatko. *Estimating Species Trees: Practical and Theoretical Aspects*. Hobroken, N. J.: Wiley-Blackwell, 2010.

Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477–500, 2001.

Alexander Korostelev and Olga Korosteleva. *Mathematical Statistics. Asymptotic Minimax Theory*. Graduate Studies in Mathematics 119. American Mathematical Society (AMS), 2011.

Gueorg Kossinets and Duncan J. Watts. Empirical analysis of an evolving social network. *Science*, 311:88–90, 2006.

Rémi Lajugie, Sylvain Arlot, and Francis Bach. Large-margin metric learning for constrained partitioning problems. In *International Conference on Machine Learning (ICML)*, volume 32, pages 297–305, 2014. See also arXiv:1303.1280.

Marc Lavielle and Eric Moulines. Least-squares estimation of an unknown number of shifts in a time series. *Journal of Time Series Analysis*, 21(1):33–59, 2000.

Émilie Lebarbier. *Quelques Approches pour la Détection de Ruptures à Horizon Fini*. PhD thesis, Université Paris-Sud, July 2002.

Émilie Lebarbier. Detecting multiple change-points in the mean of a Gaussian process by model selection. *Signal Processing*, 85:717–736, 2005.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Springer-Verlag, Berlin, 1991.

Shuang Li, Yao Xie, Hanjun Dai, and Le Song. M-statistic for kernel change-point detection. In *Advances in Neural Information Processing Systems 28*, pages 3348–3356. Curran Associates, Inc., 2015.

Shuang Li, Yao Xie, Hanjun Dai, and Le Song. Scan b-statistic for kernel change-point detection. *Sequential Analysis*, 2019. Accepted. arXiv:1507.01279.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. Convolutional kernel networks. In *Advances in Neural Information Processing Systems*, pages 2627–2635, 2014.

Pascal Massart. *Concentration Inequalities and Model Selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003.

David S. Matteson and Nicholas A. James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109 (505):334–345, 2014.

Ian McCulloh. *Detecting Changes in a Dynamic Social Network*. PhD thesis, Institute for Software Research, School of Computer Science, Carnegie Mellon University, 2009. CMU-ISR-09-104.

Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

Youngser Park, Heng Wang, Tobias Nöbauer, Alipasha Vaziri, and Carey E. Priebe. Anomaly detection on whole-brain functional imaging of neuronal activity using graph scan statistics. *Neuron*, 2(3,000):4–000, 2015.

Mattis Paulin, Julien Mairal, Matthijs Douze, Zaïd Harchaoui, Florent Perronnin, and Cordelia Schmid. Convolutional patch representations for image retrieval: An unsupervised approach. *International Journal of Computer Vision*, 121(1):149–168, 2017.

Franck Picard, Stéphane Robin, Marc Lavielle, Christian Vaisse, and Jean-Jacques Daudin. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 27(6), 2005.

Franck Picard, Émilie Lebarbier, Eva Budinska, and Stéphane Robin. Joint segmentation of multivariate Gaussian processes using mixed linear models. *Computational Statistics and Data Analysis*, 55(2):1160–70, 2011.

Iosif F. Pinelis and Aleksandr Ivanovich Sakhanenko. Remarks on inequalities for large deviation probabilities. *Theory of Probability and Its Applications*, 30(1):143–148, 1986.

Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European Conference on Computer Vision (ECCV)*, 2014. Preliminary version available at `http://hal.inria.fr/hal-01022967`.

Lawrence R. Rabiner and Ronald W. Schäfer. Introduction to digital signal processing. *Foundations and Trends in Information Retrieval*, 1(1–2):1–194, 2007.

Alain Rakotomamonjy and Stéphane Canu. Frames, reproducing kernels, regularization and learning. *Journal of Machine Learning Research (JMLR)*, 6:1485–1515, December 2005.

Marie Sauvé. Histogram selection in non Gaussian regression. *ESAIM: Probability and Statistics*, 13:70–86, 2009.

Bernard Schölkopf, Koji Tsuda, and Jean-Philippe Vert, editors. *Kernel Methods in Computational Biology*. MIT Press, 2004.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291, 2013.

Olimjon Sharipov, Johannes Tewes, and Martin Wendler. Sequential block bootstrap in a Hilbert space with application to change point analysis. *Canadian Journal of Statistics*, 44(3):300–322, 2016.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Nino Shervashidze. *Scalable Graph Kernels*. PhD thesis, Universität Tübingen, 2012. Available at `http://hdl.handle.net/10900/49731`.

Yong Sheng Soh and Venkat Chandrasekaran. High-dimensional change-point estimation: Combining filtering with convex optimization. *Applied and Computational Harmonic Analysis*, 43(1):122–147, 2017.

Olivier Sorba. *Minimal Penalties for Model Selection*. PhD thesis, Université Paris-Saclay, February 2017. Available at `https://tel.archives-ouvertes.fr/tel-01515957`.

Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert R. G. Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *Advances in Neural Information Processing Systems*, volume 21. NIPS Foundation (http://books.nips.cc), 2009.

Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R. G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research (JMLR)*, 11:1517–1561, 2010.

Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research (JMLR)*, 12:2389–2410, 2011.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Information Science and Statistics. Springer, New York, 2008.

Michel Talagrand. New concentration inequalities in product spaces. *Inventiones Mathematicae*, 126(3):505–563, 1996.

Alexander Tartakovsky, Igor Nikiforov, and Michèle Basseville. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*, volume 136 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, Boca Raton, FL, 2014.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. ruptures: change point detection in python, 2018. arXiv:1801.00826.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 2019. doi: 10.1016/j.sigpro.2019.107299. In press.

Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3), March 2012.

Heng Wang, Minh Tang, Yu-Seop Park, and Carey E. Priebe. Locality statistics for anomaly detection in time series of graphs. *IEEE Transactions on Signal Processing*, 62(3):703–717, 2014.

Chung-Hsien Wu and Chia-Hsin Hsieh. Multiple change-point audio segmentation and classification using an MDL-based Gaussian model. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(2):647–657, 2006.

Yi-Ching Yao. Estimating the number of change-points via Schwarz criterion. *Statistics and Probability Letters*, 6:181–189, 1988.

Wojciech Zaremba, Arthur Gretton, and Matthew Blaschko. B-test: A non-parametric, low variance kernel two-sample test. In *Advances in Neural Information Processing Systems 26*, pages 755–763. Curran Associates, Inc., 2013.

Nancy R. Zhang and David O. Siegmund. A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63 (1):22–32, 2007.

Changliang Zou, Guosheng Yin, Long Feng, and Zhaojun Wang. Nonparametric maximum likelihood approach to multiple change-point problems. *The Annals of Statistics*, 42(3): 970–1002, 2014.

Journal of Machine Learning Research 20 (2019) 1-42 Submitted 8/16; Published 8/19

Two-Layer Feature Reduction for Sparse-Group Lasso via Decomposition of Convex Sets

Jie Wang JIEWANGX@USTC.EDU.CN
Department of Electronic Engineering and Information Science
University of Science and Technology of China
Hefei, Anhui, China

Zhanqiu Zhang ZZQ96@MAIL.USTC.EDU.CN
Department of Electronic Engineering and Information Science
University of Science and Technology of China
Hefei, Anhui, China

Jieping Ye JPYE@UMICH.EDU
Department of Computational Medicine and Bioinformatics
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2218, USA

Editor: Zhihua Zhang

Abstract

Sparse-Group Lasso (SGL) has been shown to be a powerful regression technique for simultaneously discovering group and within-group sparse patterns by using a combination of the ℓ_1 and ℓ_2 norms. However, in large-scale applications, the complexity of the regularizers entails great computational challenges. In this paper, we propose a novel **t**wo-**l**ayer **f**eature **re**duction method (TLFre) for SGL via a decomposition of its dual feasible set. The two-layer reduction is able to quickly identify the inactive groups and the inactive features, respectively, which are guaranteed to be absent from the sparse representation and can be removed from the optimization. Existing feature reduction methods are only applicable to sparse models with one sparsity-inducing regularizer. To our best knowledge, TLFre is *the first one* that is capable of dealing with *multiple* sparsity-inducing regularizers. Moreover, TLFre has a very low computational cost and can be integrated with any existing solvers. We also develop a screening method—called DPC (**d**ecom**p**osition of **c**onvex set)—for nonnegative Lasso. Experiments on both synthetic and real data sets show that TLFre and DPC improve the efficiency of SGL and nonnegative Lasso by several orders of magnitude.

Keywords: Sparse, Sparse Group Lasso, Screening, Fenchel's Dual, Decomposition, Convex Sets, Composite Function Optimization

1. Introduction

Sparse-Group Lasso (SGL) (Friedman et al.; Simon et al., 2013) is a powerful regression technique in identifying important groups and features simultaneously. To yield sparsity at both group and individual feature levels, SGL combines the Lasso (Tibshirani, 1996) and group Lasso (Yuan and Lin, 2006) penalties. In recent years, SGL has found great success in a wide range of applications, including but not limited to machine learning

©2019 Jie Wang, Zhanqiu Zhang, and Jieping Ye.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/16-383.html.

(Vidyasagar, 2014; Yogatama and Smith, 2014), signal processing (Sprechmann et al., 2011), bioinformatics (Peng et al., 2010) etc. Many research efforts have been devoted to developing efficient solvers for SGL (Friedman et al.; Simon et al., 2013; Liu and Ye, 2010; Vincent and Hansen, 2014). However, when the feature dimension is extremely high, the complexity of the SGL regularizers imposes great computational challenges. Therefore, there is an increasingly urgent need for nontraditional techniques to address the challenges posed by the massive volume of the data sources.

Recently, El Ghaoui et al. (2012) proposed a promising feature reduction method, called *SAFE screening*, to screen out the so-called *inactive* features, which have zero coefficients in the solution, from the optimization. Thus, the size of the data matrix needed for the training phase can be significantly reduced, which may lead to substantial improvement in the efficiency of solving sparse models. Inspired by SAFE, various exact and heuristic feature screening methods have been proposed for many sparse models such as Lasso (Wang et al., 2013; Liu et al., 2014; Tibshirani et al., 2012; Xiang and Ramadge, 2012), group Lasso (Wang et al., 2013; Wang et al.; Tibshirani et al., 2012), etc. It is worthwhile to mention that the discarded features by exact feature screening methods such as SAFE (El Ghaoui et al., 2012), DOME (Xiang and Ramadge, 2012) and EDPP (Wang et al., 2013) are guaranteed to have zero coefficients in the solution. However, heuristic feature screening methods like strong rule (Tibshirani et al., 2012) may mistakenly discard features that have nonzero coefficients in the solution. Thus, to compute the exact solutions, the authors propose to check the KKT conditions after the screening pass of strong rules. More recently, the idea of exact feature screening has been extended to exact sample screening, which screens out the nonsupport vectors in SVM (Ogawa et al., 2013; Wang et al., 2014) and LAD (Wang et al., 2014). As a promising data reduction tool, exact feature/sample screening would be of great practical importance because they can effectively reduce the data size without sacrificing the optimality (Ogawa et al., 2014).

However, all of the existing feature/sample screening methods are only applicable for the sparse models with one sparsity-inducing regularizer. In this paper, we propose an exact two-layer feature screening method, called TLFre, for the SGL problem. The first and second layer of TLFre aim to quickly identify the inactive groups and the inactive features, respectively, which are guaranteed to have zero coefficients in the solution. To the best of our knowledge, TLFre is the first screening method which is capable of dealing with multiple sparsity-inducing regularizers.

We note that most of the existing exact feature screening methods involve an estimation of the dual optimal solution. The difficulty in developing screening methods for sparse models with multiple sparsity-inducing regularizers like SGL is that the dual feasible set is the sum of simple convex sets. Thus, to determine the feasibility of a given point, we need to know if it is decomposable with respect to the summands, which is itself a nontrivial problem (see Section 2). One of our major contributions is that we derive an elegant decomposition method of any dual feasible solutions of SGL via the framework of Fenchel's duality (see Section 3). Based on the Fenchel's dual problem of SGL, we motivate TLFre by an in-depth exploration of its geometric properties and the optimality conditions in Section 4. We derive the set of the regularization parameter values corresponding to zero solutions. To develop TLFre, we need to estimate the upper bounds involving the dual optimal solution. To this end, we first give an accurate estimation of the dual optimal solution via the normal

cones. Then, we formulate the estimation of the upper bounds via nonconvex optimization problems. We show that these nonconvex problems admit closed form solutions.

The rest of this paper is organized as follows. In Section 2, we briefly review some basics of the SGL problem. We then derive the Fenchel's dual of SGL with nice geometric properties under the elegant framework of Fenchel's Duality in Section 3. In Section 4, we develop the TLFre screening rule for SGL. To demonstrate the flexibility of the proposed framework, we extend TLFre to the nonnegative Lasso problem in Section 5. Experiments in Section 6 on both synthetic and real data demonstrate that the speedup gained by the proposed screening rules in solving SGL and nonnegative Lasso can be orders of magnitude. Please see the appendix for detailed proofs that are not presented in the main text.

Notation: Let $\|\cdot\|_1$, $\|\cdot\|$ and $\|\cdot\|_\infty$ be the ℓ_1, ℓ_2 and ℓ_∞ norms, respectively. Denote by $\mathcal{B}_1^n$, $\mathcal{B}^n$, and $\mathcal{B}_\infty^n$ the unit ℓ_1, ℓ_2, and ℓ_∞ norm balls in $\mathbb{R}^n$ (we omit the superscript if it is clear from the context). For a set $\mathcal{C}$, let $\operatorname{int}\mathcal{C}$ be its interior. If $\mathcal{C}$ is closed and convex, we define the projection operator as

$$\mathbf{P}_{\mathcal{C}}(\mathbf{w}) := \operatorname{argmin}_{\mathbf{u}\in\mathcal{C}}\|\mathbf{w}-\mathbf{u}\|.$$

We denote the indicator function of $\mathcal{C}$ by

$$\mathbf{I}_{\mathcal{C}}(\mathbf{w}) = \begin{cases} 0, & \text{if } \mathbf{w}\in\mathcal{C}, \\ \infty, & \text{otherwise.} \end{cases}$$

Let $\Gamma_0(\mathbb{R}^n)$ be the class of proper closed convex functions on $\mathbb{R}^n$. For $f\in\Gamma_0(\mathbb{R}^n)$, let ∂f be its subdifferential. The domain of f is the set $\operatorname{dom} f := \{\mathbf{w} : f(\mathbf{w}) < \infty\}$.

For $\mathbf{w}\in\mathbb{R}^n$, let $[\mathbf{w}]_i$ be its ith component. More generally, if $\mathcal{G}\subset\{1,2,\ldots,n\}$ is an index set, we denote the corresponding subvector of $\mathbf{w}$ by $[\mathbf{w}]_{\mathcal{G}}\in\mathbb{R}^{|\mathcal{G}|}$, where $|\mathcal{G}|$ denotes the number of elements in $\mathcal{G}$. For $\gamma\in\mathbb{R}$, let

$$\operatorname{sgn}(\gamma) = \begin{cases} \operatorname{sign}(\gamma), & \text{if } \gamma\neq 0, \\ 0, & \text{otherwise.} \end{cases}$$

We define

$$\operatorname{SGN}(\mathbf{w}) = \left\{ \mathbf{s}\in\mathbb{R}^n : [\mathbf{s}]_i \in \begin{cases} \operatorname{sign}([\mathbf{w}]_i), & \text{if } [\mathbf{w}]_i\neq 0; \\ [-1,1], & \text{if } [\mathbf{w}]_i = 0. \end{cases} \right\}$$

We denote by $\gamma_+ = \max(\gamma, 0)$. Then, for $\gamma\geq 0$, the shrinkage operator $\mathcal{S}_\gamma(\mathbf{w}) : \mathbb{R}^n\to\mathbb{R}^n$ can be written as

$$[\mathcal{S}_\gamma(\mathbf{w})]_i = (|[\mathbf{w}]_i| - \gamma)_+ \operatorname{sgn}([\mathbf{w}]_i),\ i = 1,\ldots,n. \tag{1}$$

2. Basics and Motivation

In this section, we briefly review some basics of SGL. Let $\mathbf{y}\in\mathbb{R}^N$ be the response vector and $\mathbf{X}\in\mathbb{R}^{N\times p}$ be the matrix of features. With the group information available, the SGL problem (Friedman et al.) is

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{2}\left\|\mathbf{y} - \sum\nolimits_{g=1}^{G}\mathbf{X}_g\beta_g\right\|^2 + \lambda_1\sum\nolimits_{g=1}^{G}\sqrt{n_g}\|\beta_g\| + \lambda_2\|\beta\|_1, \tag{2}$$

where n_g is the number of features in the gth group, $\mathbf{X}_g\in\mathbb{R}^{N\times n_g}$ denotes the predictors in that group with the corresponding coefficient vector β_g, and λ_1,λ_2 are positive regularization parameters. Without loss of generality, let $\lambda_1 = \alpha\lambda$ and $\lambda_2 = \lambda$ with $\alpha > 0$. Then, problem (2) becomes:

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{2}\left\|\mathbf{y} - \sum\nolimits_{g=1}^{G}\mathbf{X}_g\beta_g\right\|^2 + \lambda\left(\alpha\sum\nolimits_{g=1}^{G}\sqrt{n_g}\|\beta_g\| + \|\beta\|_1\right). \tag{3}$$

By the Lagrangian multipliers method (Boyd and Vandenberghe, 2004) (see the appendix), we can derive the dual problem of SGL as follows.

$$\begin{aligned} &\sup_{\theta}\ \frac{1}{2}\|\mathbf{y}\|^2 - \frac{1}{2}\left\|\frac{\mathbf{y}}{\lambda} - \theta\right\|^2 \\ &\text{s.t. } \mathbf{X}_g^T\theta\in\mathcal{D}_g^\alpha := \alpha\sqrt{n_g}\mathcal{B} + \mathcal{B}_\infty,\ g = 1,\ldots,G. \end{aligned} \tag{4}$$

It is well-known that the dual feasible set of Lasso is the intersection of closed half spaces (thus a polytope); for group Lasso, the dual feasible set is the intersection of ellipsoids. Surprisingly, the geometric properties of these dual feasible sets play fundamentally important roles in most of the existing screening methods for sparse models with one sparsity-inducing regularizer (Wang et al., 2014; Liu et al., 2014; Wang et al., 2013; El Ghaoui et al., 2012).

When we incorporate multiple sparse-inducing regularizers to the sparse models, problem (4) indicates that the dual feasible set can be much more complicated. Although (4) provides a geometric description of the dual feasible set of SGL, it is not suitable for further analysis. Notice that, *even the feasibility of a given point θ is not easy to determine*, since it is nontrivial to tell if $\mathbf{X}_g^T\theta$ can be decomposed into $\mathbf{b}_1+\mathbf{b}_2$ with $\mathbf{b}_1\in\alpha\sqrt{n_g}\mathcal{B}$ and $\mathbf{b}_2\in\mathcal{B}_\infty$. Therefore, to develop screening methods for SGL, it is desirable to gain deeper understanding of the sum of simple convex sets.

In the next section, we analyze the dual feasible set of SGL in depth via the Fenchel's Duality Theorem. We show that for each $\mathbf{X}_g^T\theta\in\mathcal{D}_g^\alpha$, Fenchel's duality naturally leads to an explicit decomposition $\mathbf{X}_g^T\theta = \mathbf{b}_1+\mathbf{b}_2$, with one belonging to $\alpha\sqrt{n_g}\mathcal{B}$ and the other one belonging to $\mathcal{B}_\infty$. This lays the foundation of the proposed screening method for SGL.

3. The Fenchel's Dual Problem of SGL

In Section 3.1, we derive the Fenchel's dual of SGL via Fenchel's Duality Theorem. We then motivate TLFre in Section 3.2 and sketch our approach by Algorithm 1. In Section 3.3, we explore the geometric properties of the Fenchel's dual of SGL and derive the effective interval of the parameter λ with a fixed value of α—that is the set of λ given α corresponding to nonzero solutions of SGL.

Algorithm 1 Guidelines for developing TLFre.

1: Given a pair of parameter values (λ, α), we estimate a region Θ that contains the dual optimum $\theta^*(\lambda, \alpha)$ of (4).

2: We solve the following two optimization problems:

$$s_g^* = \sup_{\xi_g} \{\|\mathcal{S}_1(\xi_g)\| : \xi_g \in \Xi_g \supseteq \mathbf{X}_g^T\Theta\}, \text{ where } \mathbf{X}_g^T\Theta = \{\mathbf{X}_g^T\theta : \theta \in \Theta\}, \tag{5}$$

$$t_{g_k}^* = \sup_{\theta} \{|\mathbf{x}_{g_k}^T\theta| : \theta \in \Theta\}, \text{ where } \mathbf{x}_{g_k} \text{ is the } k^{th} \text{ column of } \mathbf{X}_g. \tag{6}$$

3: The TLFre screening rules take the form of

$$s_g^* < \alpha\sqrt{n_g} \Rightarrow \beta_g^*(\lambda, \alpha) = 0, \tag{7}$$

$$t_{g_k}^* \leq 1 \Rightarrow [\beta_g^*(\lambda, \alpha)]_k = 0, \tag{8}$$

where $\beta^*(\lambda, \alpha)$ is the optimal solution of SGL in (3).

3.1. The Fenchel's Dual of SGL via Fenchel's Duality Theorem

To derive the Fenchel's dual problem of SGL, we need the Fenchel's Duality Theorem as stated in Theorem 1. We denote the conjugate of $f \in \Gamma_0(\mathbb{R}^n)$ by $f^* \in \Gamma_0(\mathbb{R}^n)$:

$$f^*(\mathbf{z}) = \sup_{\mathbf{w}} \langle \mathbf{w}, \mathbf{z}\rangle - f(\mathbf{w}). \tag{9}$$

Theorem 1 [Fenchel's Duality Theorem] *Let $f \in \Gamma_0(\mathbb{R}^N)$, $\Omega \in \Gamma_0(\mathbb{R}^p)$, and $\mathcal{T}(\beta) = \mathbf{y} - \mathbf{X}\beta$ be an affine mapping from $\mathbb{R}^p$ to $\mathbb{R}^N$. Let $p^*, d^* \in [-\infty, \infty]$ be primal and dual values defined, respectively, by the Fenchel problems:*

$$p^* = \inf_{\beta\in\mathbb{R}^p} f(\mathbf{y} - \mathbf{X}\beta) + \lambda\Omega(\beta); \quad d^* = \sup_{\theta\in\mathbb{R}^N} -f^*(\lambda\theta) - \lambda\Omega^*(\mathbf{X}^T\theta) + \lambda\langle \mathbf{y}, \theta\rangle.$$

One has $p^* \geq d^*$. If, furthermore, f and Ω satisfy the condition

$$0 \in \operatorname{int}\left(\operatorname{dom} f - \mathbf{y} + \mathbf{X}\operatorname{dom}\Omega\right),$$

then $p^* = d^*$, and the supreme is attained in the dual problem if finite.

We omit the proof of Theorem 1 as it is similar to that of Theorem 3.3.5 in (Borwein and Lewis, 2006).

Let $f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$ and $\lambda\Omega(\beta)$ be the second term in (3). We can write SGL as

$$\min_\beta f(\mathbf{y} - \mathbf{X}\beta) + \lambda\Omega(\beta). \tag{10}$$

To derive the Fenchel's dual problem of SGL, Theorem 1 implies that we need to find f^* and Ω^*. It is well-known that $f^*(\mathbf{z}) = \frac{1}{2}\|\mathbf{z}\|^2$. Therefore, we only need to find Ω^*, where the concept *infimal convolution* is needed:

Definition 2 (Bauschke and Combettes, 2011) *Let $h, g \in \Gamma_0(\mathbb{R}^n)$. The infimal convolution of h and g is*

$$(h\Box g)(\xi) = \inf_\eta h(\eta) + g(\xi - \eta), \tag{11}$$

and it is exact at a point ξ if there exists a $\eta^*(\xi)$ such that

$$(h\Box g)(\xi) = h(\eta^*(\xi)) + g(\xi - \eta^*(\xi)). \tag{12}$$

$h\Box g$ is exact if it is exact at each point in its domain, and we denote it by $h \boxdot g$.

With the infimal convolution, we derive Ω^* in the following Lemma.

Lemma 3 *Let $\Omega_1^\alpha(\beta) = \alpha\sum_{g=1}^G \sqrt{n_g}\|\beta_g\|$, $\Omega_2(\beta) = \|\beta\|_1$ and $\Omega(\beta) = \Omega_1^\alpha(\beta) + \Omega_2(\beta)$. Moreover, let $\mathcal{C}_g^\alpha = \alpha\sqrt{n_g}\mathcal{B} \subset \mathbb{R}^{n_g}$, $g = 1, \ldots, G$. Then, the following hold:*

(i) $(\Omega_1^\alpha)^*(\xi) = \sum_{g=1}^G \mathbf{I}_{\mathcal{C}_g^\alpha}(\xi_g)$, $\quad (\Omega_2)^*(\xi) = \sum_{g=1}^G \mathbf{I}_{\mathcal{B}_\infty}(\xi_g)$,

(ii) $\Omega^*(\xi) = ((\Omega_1^\alpha)^* \boxdot (\Omega_2)^*)(\xi) = \sum_{g=1}^G \mathbf{I}_{\mathcal{B}}\left(\frac{\xi_g - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g)}{\alpha\sqrt{n_g}}\right)$,

where $\xi_g \in \mathbb{R}^{n_g}$ is the sub-vector of ξ corresponding to the gth group.

To prove Lemma 3, we first cite the following technical result.

Theorem 4 (Hiriart-Urruty, 2006) *Let $f_1, \cdots, f_k \in \Gamma_0(\mathbb{R}^n)$. Suppose there is a point in $\cap_{i=1}^k \operatorname{dom} f_i$ at which $f_1, \cdots, f_{k-1}$ is continuous. Then, for all $p \in \mathbb{R}^n$:*

$$(f_1 + \cdots + f_k)^*(p) = \min_{p_1+\cdots+p_k=p} [f_1^*(p_1) + \cdots + f_k^*(p_k)].$$

We now give the proof of Lemma 3.

Proof The first part can be derived directly by the definition as follows:

$$\begin{aligned}(\Omega_1^\alpha)^*(\xi) &= \sup_\beta \langle \beta, \xi\rangle - \Omega_1^\alpha(\beta) = \sum_{g=1}^G \alpha\sqrt{n_g}\left(\sup_{\beta_g}\left\langle \beta_g, \frac{\xi_g}{\alpha\sqrt{n_g}}\right\rangle - \|\beta_g\|\right) \\ &= \sum_{g=1}^G \alpha\sqrt{n_g}\mathbf{I}_{\mathcal{B}}\left(\frac{\xi_g}{\alpha\sqrt{n_g}}\right) = \sum_{g=1}^G \mathbf{I}_{\mathcal{B}}\left(\frac{\xi_g}{\alpha\sqrt{n_g}}\right) = \sum_{g=1}^G \mathbf{I}_{\mathcal{C}_g^\alpha}(\xi_g).\end{aligned}$$

$$(\Omega_2)^*(\xi) = \sup_\beta \langle \beta, \xi\rangle - \Omega_2(\beta) = \mathbf{I}_{\mathcal{B}_\infty}(\xi) = \sum_{g=1}^G \mathbf{I}_{\mathcal{B}_\infty}(\xi_g).$$

To show the second part, Theorem 4 indicates that we only need to show $(\Omega_1^\alpha)^*\Box(\Omega_2)^*(\xi)$ is exact (note that Ω_1^α and Ω_2 are continuous everywhere). Let us now compute $(\Omega_1^\alpha)^*\Box(\Omega_2)^*$.

$$\begin{aligned}((\Omega_1^\alpha)^*\Box(\Omega_2)^*)(\xi) &= \inf_\eta (\Omega_1^\alpha)^*(\xi - \eta) + (\Omega_2)^*(\eta) \qquad (13)\\ &= \sum_{g=1}^G \inf_{\eta_g} \mathbf{I}_{\mathcal{B}}\left(\frac{\xi_g - \eta_g}{\alpha\sqrt{n_g}}\right) + \mathbf{I}_{\mathcal{B}_\infty}(\eta_g) \\ &= \sum_{g=1}^G \inf_{\|\eta_g\|_\infty \leq 1} \mathbf{I}_{\mathcal{B}}\left(\frac{\xi_g - \eta_g}{\alpha\sqrt{n_g}}\right)\end{aligned}$$

To solve the optimization problem in (13), i.e.,

$$\mu_g^* = \inf_{\eta_g} \left\{ \mathbf{I}_{\mathcal{B}} \left(\frac{\xi_g - \eta_g}{\alpha\sqrt{n_g}} \right) : \|\eta_g\|_\infty \le 1 \right\}, \tag{14}$$

we can consider the following problem

$$\nu_g^* = \inf_{\eta_g} \left\{ \frac{1}{\alpha\sqrt{n_g}} \|\xi_g - \eta_g\| : \|\eta_g\|_\infty \le 1 \right\}. \tag{15}$$

We can see that the optimal solution of problem (15) must also be an optimal solution of problem (14). Let $\eta_g^*(\xi_g)$ be the optimal solution of (15). We can see that $\eta_g^*(\xi_g)$ is indeed the projection of ξ_g on $\mathcal{B}_\infty$, which admits a closed form solution:

$$[\eta_g^*(\xi_g)]_i = [\mathbf{P}_{\mathcal{B}_\infty}(\xi_g)]_i = \begin{cases} 1, & \text{if } [\xi_g]_i > 1, \\ [\xi_g]_i, & \text{if } |[\xi_g]_i| \le 1, \\ -1, & \text{if } [\xi_g]_i < -1. \end{cases}$$

Thus, problem (14) can be solved as

$$\mu_g^* = \mathbf{I}_{\mathcal{B}} \left(\frac{\xi_g - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g)}{\alpha\sqrt{n_g}} \right).$$

Hence, the infimal convolution in Eq. (13) is exact and Theorem 4 leads to

$$\Omega^*(\xi) = \left((\Omega_1^\alpha)^* \boxdot (\Omega_2)^* \right)(\xi) = \sum_{g=1}^{G} \mathbf{I}_{\mathcal{B}} \left(\frac{\xi_g - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g)}{\alpha\sqrt{n_g}} \right), \tag{16}$$

which completes the proof. ■

Note that $\mathbf{P}_{\mathcal{B}_\infty}(\xi_g)$ admits a closed form solution:

$$[\mathbf{P}_{\mathcal{B}_\infty}(\xi_g)]_i = \operatorname{sgn}([\xi_g]_i) \min(|[\xi_g]_i|, 1).$$

By Theorem 1 and Lemma 3, we derive the Fenchel's dual of SGL in Theorem 5 (see Section B for the proof).

Theorem 5 *For the SGL problem in (3), the following hold:*

(i) *The Fenchel's dual of SGL is given by:*

$$\begin{aligned} &\inf_\theta \ \frac{1}{2}\|\frac{\mathbf{y}}{\lambda} - \theta\|^2 - \frac{1}{2}\|\mathbf{y}\|^2, \\ &\text{s.t.} \ \left\|\mathbf{X}_g^T\theta - \mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_g^T\theta)\right\| \le \alpha\sqrt{n_g}, \ g = 1, \ldots, G. \end{aligned} \tag{17}$$

(ii) *Let $\beta^*(\lambda, \alpha)$ and $\theta^*(\lambda, \alpha)$ be the optimal solutions of problems (3) and (17), respectively. Then,*

$$\lambda\theta^*(\lambda, \alpha) = \mathbf{y} - \mathbf{X}\beta^*(\lambda, \alpha), \tag{18}$$

$$\mathbf{X}_g^T\theta^*(\lambda, \alpha) \in \alpha\sqrt{n_g}\partial\|\beta_g^*(\lambda, \alpha)\| + \partial\|\beta_g^*(\lambda, \alpha)\|_1, \ g = 1, \ldots, G. \tag{19}$$

Eq. (18) and Eq. (19) are the so-called KKT conditions (Boyd and Vandenberghe, 2004) and can also be obtained by the Lagrangian multiplier method (see Section A in the appendix).

Remark 6 *We note that the shrinkage operator can also be expressed by*

$$\mathcal{S}_\gamma(\mathbf{w}) = \mathbf{w} - \mathbf{P}_{\gamma\mathcal{B}_\infty}(\mathbf{w}), \ \ \gamma \ge 0. \tag{20}$$

Therefore, problem (17) can be written more compactly as

$$\begin{aligned} &\inf_\theta \ \frac{1}{2}\|\frac{\mathbf{y}}{\lambda} - \theta\|^2 - \frac{1}{2}\|\mathbf{y}\|^2, \\ &\text{s.t.} \ \left\|\mathcal{S}_1(\mathbf{X}_g^T\theta)\right\| \le \alpha\sqrt{n_g}, \ g = 1, \ldots, G. \end{aligned} \tag{21}$$

The equivalence between the dual formulations For the SGL problem, its Lagrangian dual in (4) and Fenchel's dual in (17) are indeed equivalent to each other. We bridge them together by the following lemma.

Lemma 7 (Bauschke and Combettes, 2011) *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be nonempty subsets of $\mathbb{R}^n$. Then $\mathbf{I}_{\mathcal{C}_1} \boxdot \mathbf{I}_{\mathcal{C}_2} = \mathbf{I}_{\mathcal{C}_1 + \mathcal{C}_2}$.*

In view of Lemmas 3 and 7, and recall that $\mathcal{D}_g^\alpha = \mathcal{C}_g^\alpha + \mathcal{B}_\infty$, we have

$$\Omega^*(\xi) = \left((\Omega_1^\alpha)^* \boxdot (\Omega_2)^* \right)(\xi) = \sum_{g=1}^{G} \left(\mathbf{I}_{\mathcal{C}_g^\alpha} \boxdot \mathbf{I}_{\mathcal{B}_\infty} \right)(\xi_g) = \sum_{g=1}^{G} \mathbf{I}_{\mathcal{D}_g^\alpha}(\xi_g). \tag{22}$$

Combining Eq. (22) and Theorem 1, we obtain the dual formulation of SGL in (4). Therefore, the dual formulations of SGL in (4) and (17) are the same.

Remark 8 *An appealing advantage of the Fenchel's dual in (17) is that we have a natural decomposition of all points $\xi_g \in \mathcal{D}_g^\alpha$: $\xi_g = \mathbf{P}_{\mathcal{B}_\infty}(\xi_g) + \mathcal{S}_1(\xi_g))$ with $\mathbf{P}_{\mathcal{B}_\infty}(\xi_g) \in \mathcal{B}_\infty$ and $\mathcal{S}_1(\xi_g) \in \mathcal{C}_g^\alpha$. As a result, this leads to a convenient way to determine the feasibility of any dual variable θ by checking if $\mathcal{S}_1(\mathbf{X}_g^T\theta) \in \mathcal{C}_g^\alpha$, $g = 1, \ldots, G$.*

3.2. Motivation of the Two-Layer Screening Rules

We motive the two-layer screening rules via the KKT condition in Eq. (19). As implied by the name, there are two layers in our method. The first layer aims to identify the inactive groups, and the second layer detects the inactive features for the remaining groups.

by Eq. (19), we have the following cases by noting $\partial\|\mathbf{w}\|_1 = \mathrm{SGN}(\mathbf{w})$ and

$$\partial\|\mathbf{w}\| = \begin{cases} \left\{ \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\}, & \text{if } \mathbf{w} \ne 0, \\ \{\mathbf{u} : \|\mathbf{u}\| \le 1\}, & \text{if } \mathbf{w} = 0. \end{cases}$$

Case 1. If $\beta_g^*(\lambda, \alpha) \ne 0$, we have

$$[\mathbf{X}_g^T\theta^*(\lambda, \alpha)]_k \in \begin{cases} \alpha\sqrt{n_g}\frac{[\beta_g^*(\lambda,\alpha)]_k}{\|\beta_g^*(\lambda,\alpha)\|} + \operatorname{sign}([\beta_g^*(\lambda, \alpha)]_k), & \text{if } [\beta_g^*(\lambda, \alpha)]_k \ne 0, \\ [-1, 1], & \text{if } [\beta_g^*(\lambda, \alpha)]_k = 0. \end{cases} \tag{23}$$

In view of Eq. (23), we can see that

$$\text{(a): } \mathcal{S}_1(\mathbf{X}_g^T\theta^*(\lambda,\alpha)) = \alpha\sqrt{n_g}\frac{\beta_g^*(\lambda_1,\lambda_2)}{\|\beta_g^*(\lambda_1,\lambda_2)\|} \text{ and } \|\mathcal{S}_1(\mathbf{X}_g^T\theta^*(\lambda,\alpha))\| = \alpha\sqrt{n_g}, \tag{24}$$

$$\text{(b): If } \left|[\mathbf{X}_g^T\theta^*(\lambda,\alpha]_k\right| \leq 1 \text{ then } [\beta_g^*(\lambda,\alpha)]_k = 0. \tag{25}$$

Case 2. If $\beta_g^*(\lambda,\alpha) = 0$, we have

$$[\mathbf{X}_g^T\theta^*(\lambda,\alpha)]_k \in \alpha\sqrt{n_g}[\mathbf{u}_g]_k + [-1,1],\ \|\mathbf{u}_g\| \leq 1. \tag{26}$$

The first layer (group-level) of TLFre From (24) in **Case 1**, we have

$$\left\|\mathcal{S}_1(\mathbf{X}_g^T\theta^*(\lambda,\alpha))\right\| < \alpha\sqrt{n_g} \Rightarrow \beta_g^*(\lambda,\alpha) = 0. \tag{R1}$$

We can see that we can utilize (R1) to identify the inactive groups, and thus it is a group-level screening rule.

The second layer (feature-level) of TLFre Let $\mathbf{x}_{g_k}$ be the kth column of $\mathbf{X}_g$. We have $[\mathbf{X}_g^T\theta^*(\lambda,\alpha)]_k = \mathbf{x}_{g_k}^T\theta^*(\lambda,\alpha)$. In view of (25) and (26), we can see that

$$\left|\mathbf{x}_{g_k}^T\theta^*(\lambda,\alpha)\right| \leq 1 \Rightarrow [\beta_g^*(\lambda,\alpha)]_k = 0. \tag{R2}$$

Different from (R1), (R2) detects the inactive features, and thus it is a feature-level screening rule.

However, we cannot directly apply (R1) and (R2) to identify the inactive groups/features because both need to know $\theta^*(\lambda,\alpha)$. Inspired by the SAFE rules (El Ghaoui et al., 2012), we can first estimate a region Θ containing $\theta^*(\lambda,\alpha)$. Let $\mathbf{X}_g^T\Theta = \{\mathbf{X}_g^T\theta : \theta \in \Theta\}$. Then, (R1) and (R2) can be relaxed as follows:

$$\sup_{\xi_g}\left\{\|\mathcal{S}_1(\xi_g)\| : \xi_g \in \Xi_g \supseteq \mathbf{X}_g^T\Theta\right\} < \alpha\sqrt{n_g} \Rightarrow \beta_g^*(\lambda,\alpha) = 0, \tag{R1*}$$

$$\sup_{\theta}\left\{\left|\mathbf{x}_{g_k}^T\theta\right| : \theta \in \Theta\right\} \leq 1 \Rightarrow [\beta_g^*(\lambda,\alpha)]_k = 0. \tag{R2*}$$

We note that the two optimization problems in (R1*) and (R2*) are the same with (5) and (6), respectively. Therefore, inspired by (R1*) and (R2*), we can develop TLFre via the guidelines as shown in Algorithm 1.

3.3. The Effective Interval of Parameter Values

In this section, we explore the geometric properties of the Fenchel's dual of SGL in depth—based on which we can derive the set of parameter values such that the primal optimum is zero/nonzero. We note that, Simon et al. (Simon et al., 2013) derived similar results for SGL with a different parameterization of the parameter values. However, their approach is based on the primal problem of SGL and the KKT conditions. Our new approach—that is based on the dual perspective—sheds new insights on the geometric properties of SGL. We consider the SGL problem in (3) and (2) in Section 3.3.1 and 3.3.2, respectively.

3.3.1. The Effective Interval of Parameter Values for Problem (3)

Consider the SGL problem in (3). For notational convenience, let

$$\mathcal{F}_g^\alpha = \{\theta : \|\mathcal{S}_1(\mathbf{X}_g^T\theta)\| \leq \alpha\sqrt{n_g}\},\ g = 1,\ldots,G.$$

We denote the feasible set of the Fenchel's dual of SGL by

$$\mathcal{F}^\alpha = \cap_{g=1,\ldots,G}\mathcal{F}_g^\alpha.$$

Problem (17) [or (21)] implies that $\theta^*(\lambda,\alpha)$ is the projection of $\mathbf{y}/\lambda$ on $\mathcal{F}^\alpha$, i.e.,

$$\theta^*(\lambda,\alpha) = \mathbf{P}_{\mathcal{F}^\alpha}(\mathbf{y}/\lambda). \tag{27}$$

Thus, if $\mathbf{y}/\lambda \in \mathcal{F}^\alpha$, we have $\theta^*(\lambda,\alpha) = \mathbf{y}/\lambda$. Moreover, (R1) implies that $\beta^*(\lambda,\alpha) = 0$ if $\mathbf{y}/\lambda$ is an *interior* point of $\mathcal{F}^\alpha$. Indeed, we have the following stronger result.

Theorem 9 *For the SGL problem in (3), let*

$$\lambda_{\max}^\alpha = \max_g\{\rho_g : \left\|\mathcal{S}_1(\mathbf{X}_g^T\mathbf{y}/\rho_g)\right\| = \alpha\sqrt{n_g}\}. \tag{28}$$

Then, the following statements are equivalent:

(i) $\dfrac{\mathbf{y}}{\lambda} \in \mathcal{F}^\alpha$, (ii) $\theta^*(\lambda,\alpha) = \dfrac{\mathbf{y}}{\lambda}$, (iii) $\beta^*(\lambda,\alpha) = 0$, (iv) $\lambda \geq \lambda_{\max}^\alpha$.

Remark 10 *Theorem 9 implies that the primal optimum $\beta^*(\lambda,\alpha) \neq 0$ if and only if $\lambda \in (0,\lambda_{\max}^\alpha)$, namely, the effective interval of the parameter λ with a fixed value of α is $(0,\lambda_{\max}^\alpha)$.*

We note that ρ_g in the definition of $\lambda_{\max}^\alpha$ admits a closed form solution. For notational convenience, let $|\mathbf{w}|$ be the vector by taking absolute value of $\mathbf{w}$ component-wisely and $[\mathbf{w}]^{(k)}$ be the vector consisting of the first k components of $\mathbf{w}$.

Lemma 11 *We sort $0 \neq |\mathbf{X}_g^T\mathbf{y}| \in \mathbb{R}^{n_g}$ in descending order and denote it by $\mathbf{z}$.*

(i) *If there exists $[\mathbf{z}]_k$ such that $\|\mathcal{S}_1(\mathbf{X}_g^T\mathbf{y}/[\mathbf{z}]_k)\| = \alpha\sqrt{n_g}$, then $\rho_g = [\mathbf{z}]_k$.*

(ii) *Otherwise, let $\tau_i = \|\mathcal{S}_1(\mathbf{X}_g^T\mathbf{y}/[\mathbf{z}]_i)\|$, $i = 1,\ldots,n_g$, and $\tau_{n_g+1} = \infty$. There exists a k such that $\alpha\sqrt{n_g} \in (\tau_k,\tau_{k+1})$, and $\rho_g \in ([\mathbf{z}]_{k+1},[\mathbf{z}]_k)$ is the root of*

$$(k - \alpha^2 n_g)\rho^2 - 2\rho\|[\mathbf{z}]^{(k)}\|_1 + \|[\mathbf{z}]^{(k)}\|^2 = 0.$$

We omit the proof of Lemma 11 because it is a direct consequence by noting that

$$\|\mathcal{S}_1(\mathbf{X}_g^T\mathbf{y}/\lambda)\|^2 = \alpha^2 n_g$$

is piecewise quadratic.

3.3.2. The Effective Interval of Parameter Values for Problem (2)

Theorem 9 implies that the optimal solution $\beta^*(\lambda,\alpha)$ is 0 as long as $\mathbf{y}/\lambda \in \mathcal{F}^\alpha$. This geometric property also leads to an explicit characterization of the set of (λ_1,λ_2) such that the corresponding solution of problem (2) is 0. We denote by $\bar{\beta}^*(\lambda_1,\lambda_2)$ the optimal solution of problem (2).

Corollary 12 *For the SGL problem in (2), let*

$$\lambda_1^{\max}(\lambda_2) = \max_g \frac{1}{\sqrt{n_g}} \|\mathcal{S}_{\lambda_2}(\mathbf{X}_g^T\mathbf{y})\|.$$

Then, the following hold.

(i) $\bar{\beta}^*(\lambda_1,\lambda_2) = 0 \Leftrightarrow \lambda_1 \geq \lambda_1^{\max}(\lambda_2)$.

(ii) $\bar{\beta}^*(\lambda_1,\lambda_2) = 0$ *if*

$$\lambda_1 \geq \lambda_1^{\max} := \max_g \frac{1}{\sqrt{n_g}} \|\mathbf{X}_g^T\mathbf{y}\| \text{ or } \lambda_2 \geq \lambda_2^{\max} := \|\mathbf{X}^T\mathbf{y}\|_\infty.$$

By Corollary 12, we can see that the primal optimum $\bar{\beta}^*(\lambda_1,\lambda_2) \neq 0$ if and only if $\lambda_1 \in (0, \lambda_1^{\max}(\lambda_2))$. In other words, the effective interval of the parameter λ_1 with a fixed value of λ_2 is $(0, \lambda_1^{\max}(\lambda_2))$.

4. The Two-Layer Screening Rules for SGL

We follow the guidelines in Algorithm 1 to develop TLFre. In Section 4.1, we give an accurate estimation of $\theta^*(\lambda,\alpha)$ via normal cones (Ruszczyński, 2006). Then, we compute the supreme values in (R1*) and (R2*) by solving nonconvex problems in Section 4.2.

We note that, in many applications, the parameter values that perform the best are usually unknown. To determine appropriate parameter values, commonly used approaches such as cross validation and stability selection involve solving SGL many times over a grip of parameter values. Thus, given $\{\alpha_i\}_{i=1}^{\mathcal{I}}$ and $\lambda_{i,1} > \cdots > \lambda_{i,\mathcal{J}_i}$, we can fix the value of α each time and solve SGL by varying the value of λ. We repeat the process until we solve SGL for all of the parameter values.

We present the TLFre screening rule combined with any solver for solving the SGL problems at a grid of parameters in Algorithm 2 (see Section 4.3 for a detailed explanation). Moreover, Theorem 9 gives the closed form solution of $\beta^*(\lambda,\alpha)$ for any $\lambda \geq \lambda_{\max}^\alpha$. Thus, we assume that the input parameter values in Algorithm 2 satisfy $\lambda_{i,j} < \lambda_{\max}^{\alpha_i}$ for all $i = 1,\ldots,\mathcal{I}$ and $j = 1,\ldots,\mathcal{J}_i$.

Algorithm 2 The TLFre screening rule combined with any solver of SGL.

Input: $\{(\lambda_{i,j},\alpha_i) : i = 1,\ldots,\mathcal{I}, j = 1,\ldots,\mathcal{J}_i\}$, where $\lambda_{\max}^{\alpha_i} > \lambda_{i,1} > \cdots > \lambda_{i,\mathcal{J}_i}$ for $i = 1,\ldots,\mathcal{I}$.

Output: $\beta^*(\lambda_{i,j},\alpha_i)$ and the index set $\mathcal{G}_{i,j}$ such that $[\beta^*(\lambda_{i,j},\alpha_i)]_{\mathcal{G}_{i,j}} = 0$ for $i = 1,\ldots,\mathcal{I}$ and $j = 1,\ldots,\mathcal{J}_i$.

Initialize $\mathcal{G}_{i,j} \leftarrow \emptyset$, $i = 1,\ldots,\mathcal{I}$, $j = 1,\ldots,\mathcal{J}_i$.
for $i = 1$ to $\mathcal{I}$ **do**
  Compute $\lambda_{\max}^{\alpha_i}$ by Eq. (28) and set $\lambda_{i,0} \leftarrow \lambda_{\max}^{\alpha_i}$.
  Set $\theta^*(\lambda_{i,0},\alpha_i) \leftarrow \frac{\mathbf{y}}{\lambda_{i,0}}$ by Theorem 9.
  for $j = 1$ to $\mathcal{J}_i$ **do**
    /* compute the ball Θ that contains $\theta^*(\lambda_{i,j},\alpha_i)$ */
    Compute $\mathbf{v}_{\alpha_i}^\perp(\lambda_{i,j-1},\lambda_{i,j})$ by Theorem 14.
    Set the center of Θ: $\mathbf{o}_{\alpha_i}(\lambda_{i,j-1},\lambda_{i,j}) \leftarrow \theta^*(\lambda_{i,j-1},\alpha_i) + \frac{1}{2}\mathbf{v}_{\alpha_i}^\perp(\lambda_{i,j},\lambda_{i,j-1})$.
    Set the radius of Θ: $r_{\alpha_i}(\lambda_{i,j-1},\lambda_{i,j}) \leftarrow \frac{1}{2}\|\mathbf{v}_{\alpha_i}^\perp(\lambda_{i,j},\lambda_{i,j-1})\|$.
    for $g = 1$ to G **do**
      Compute $s_g^*(\lambda_{i,j},\lambda_{i,j-1};\alpha_i)$ by Theorem 17.
      /* the first layer (group-level screening) of TLFre */
      if $s_g^*(\lambda_{i,j},\lambda_{i,j-1};\alpha_i) < \alpha_i\sqrt{n_g}$ **then**
        $\beta_g^*(\lambda_{i,j},\alpha_i) = 0$.
        Set $\mathcal{G}_{i,j} \leftarrow \mathcal{G}_{i,j} \cup \{g_k : \mathbf{x}_{g_k}$ is the kth column of $\mathbf{X}_g\}$.
      else
        for $k = 1$ to n_g **do**
          Compute $t_{g_k}^*(\lambda_{i,j},\lambda_{i,j-1};\alpha_i)$ by Theorem 18.
          /* the second layer (feature-level screening) of TLFre */
          if $t_{g_k}^*(\lambda_{i,j},\lambda_{i,j-1};\alpha_i) \leq 1$ **then**
            $[\beta_g^*(\lambda_{i,j},\alpha_i)]_k = 0$.
            Set $\mathcal{G}_{i,j} \leftarrow \mathcal{G}_{i,j} \cup \{g_k\}$.
          end if
        end for
      end if
    end for
    Set $\overline{\mathcal{G}}_{i,j} \leftarrow \{k : k = 1,\ldots,p, k \notin \mathcal{G}_{i,j}\}$.
    Compute $[\beta^*(\lambda_{i,j},\alpha_i)]_{\overline{\mathcal{G}}_{i,j}}$ on the reduced data set $\mathbf{X}_{\overline{\mathcal{G}}_{i,j}}$ by any solver.
    Set $\theta^*(\lambda_{i,j},\alpha_i) \leftarrow (\mathbf{y} - \mathbf{X}\beta^*(\lambda_{i,j},\alpha_i))/\lambda_{i,j}$ by Eq. (18).
  end for
end for

4.1. Estimation of the Dual Optimal Solution

Due to the geometric property of the dual problem in (17), i.e., $\theta^*(\lambda,\alpha) = \mathbf{P}_{\mathcal{F}^\alpha}(\mathbf{y}/\lambda)$, we have a very useful characterization of the dual optimal solution via the so-called normal cones (Ruszczyński, 2006).

Proposition 13 (Ruszczyński, 2006; Bauschke and Combettes, 2011) *For a closed convex set $\mathcal{C} \in \mathbb{R}^n$ and a point $\mathbf{w} \in \mathcal{C}$, the normal cone to $\mathcal{C}$ at $\mathbf{w}$ is defined by*

$$N_{\mathcal{C}}(\mathbf{w}) = \{\mathbf{v} : \langle \mathbf{v}, \mathbf{w}' - \mathbf{w} \rangle \leq 0, \forall \mathbf{w}' \in \mathcal{C}\}. \tag{29}$$

Then, the following hold:

(i) $N_{\mathcal{C}}(\mathbf{w}) = \{\mathbf{v} : \mathbf{P}_{\mathcal{C}}(\mathbf{w} + \mathbf{v}) = \mathbf{w}\}$.

(ii) $\mathbf{P}_{\mathcal{C}}(\mathbf{w} + \mathbf{v}) = \mathbf{w}$, $\forall \mathbf{v} \in N_{\mathcal{C}}(\mathbf{w})$.

(iii) *Let $\overline{\mathbf{w}} \notin \mathcal{C}$. Then, $\mathbf{w} = \mathbf{P}_{\mathcal{C}}(\overline{\mathbf{w}}) \Leftrightarrow \overline{\mathbf{w}} - \mathbf{w} \in N_{\mathcal{C}}(\mathbf{w})$.*

(iv) *Let $\overline{\mathbf{w}} \notin \mathcal{C}$ and $\mathbf{w} = \mathbf{P}_{\mathcal{C}}(\overline{\mathbf{w}})$. Then, $\mathbf{P}_{\mathcal{C}}(\mathbf{w} + t(\overline{\mathbf{w}} - \mathbf{w})) = \mathbf{w}$ for all $t \geq 0$.*

By Theorem 9, $\theta^*(\bar{\lambda}, \alpha)$ is known if $\bar{\lambda} = \lambda_{\max}^{\alpha}$. Thus, we can estimate $\theta^*(\lambda, \alpha)$ in terms of $\theta^*(\bar{\lambda}, \alpha)$. Due to the same reason, *we only consider the cases with $\lambda < \lambda_{\max}^{\alpha}$ for $\theta^*(\lambda, \alpha)$* to be estimated.

Theorem 14 *For the SGL problem in (3), suppose that $\theta^*(\bar{\lambda}, \alpha)$ is known with $\bar{\lambda} \leq \lambda_{\max}^{\alpha}$. Let ρ_g, $g = 1, \ldots, G$, be defined by Theorem 9. For $\lambda \in (0, \bar{\lambda})$, let*

$$\mathbf{n}_{\alpha}(\bar{\lambda}) = \begin{cases} \frac{\mathbf{y}}{\bar{\lambda}} - \theta^*(\bar{\lambda}, \alpha), & \text{if } \bar{\lambda} < \lambda_{\max}^{\alpha}, \\ \mathbf{X}_* \mathcal{S}_1\left(\mathbf{X}_*^T \frac{\mathbf{y}}{\lambda_{\max}^{\alpha}}\right), & \text{if } \bar{\lambda} = \lambda_{\max}^{\alpha}, \end{cases} \text{ where } \mathbf{X}_* = \operatorname{argmax}_{\mathbf{X}_g} \rho_g,$$

$$\mathbf{v}_{\alpha}(\lambda, \bar{\lambda}) = \frac{\mathbf{y}}{\lambda} - \theta^*(\bar{\lambda}, \alpha),$$

$$\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda}) = \mathbf{v}_{\alpha}(\lambda, \bar{\lambda}) - \frac{\langle \mathbf{v}_{\alpha}(\lambda, \bar{\lambda}), \mathbf{n}_{\alpha}(\bar{\lambda}) \rangle}{\|\mathbf{n}_{\alpha}(\bar{\lambda})\|^2} \mathbf{n}_{\alpha}(\bar{\lambda}).$$

Then, the following hold:

(i) $\mathbf{n}_{\alpha}(\bar{\lambda}) \in N_{\mathcal{F}^{\alpha}}(\theta^*(\bar{\lambda}, \alpha))$,

(ii) $\|\theta^*(\lambda, \alpha) - (\theta^*(\bar{\lambda}, \alpha) + \frac{1}{2}\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda}))\| \leq \frac{1}{2}\|\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda})\|$.

For notational convenience, we denote

$$\mathbf{o}_{\alpha}(\lambda, \bar{\lambda}) = \theta^*(\bar{\lambda}, \alpha) + \frac{1}{2}\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda}). \tag{30}$$

Theorem 14 shows that $\theta^*(\lambda, \alpha)$ lies inside the ball of radius $\frac{1}{2}\|\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda})\|$ centered at $\mathbf{o}_{\alpha}(\lambda, \bar{\lambda})$.

4.2. Solving for the Supreme Values via Nonconvex Optimization

We solve the optimization problems in (R1*) and (R2*). To simplify notations, let

$$\Theta = \{\theta : \|\theta - \mathbf{o}_{\alpha}(\lambda, \bar{\lambda})\| \leq \frac{1}{2}\|\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda})\|\}, \tag{31}$$

$$\Xi_g = \left\{\xi_g : \|\xi_g - \mathbf{X}_g^T \mathbf{o}_{\alpha}(\lambda, \bar{\lambda})\| \leq \frac{1}{2}\|\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda})\| \|\mathbf{X}_g\|_2\right\}, \; g = 1, \ldots, G. \tag{32}$$

Theorem 14 indicates that $\theta^*(\lambda, \alpha) \in \Theta$. Moreover, we can see that $\mathbf{X}_g^T \Theta \subseteq \Xi_g$, $g = 1, \ldots, G$. To develop the TLFre rule by (R1*) and (R2*), we need to solve the following optimization problems:

$$s_g^*(\lambda, \bar{\lambda}; \alpha) = \sup_{\xi_g} \{\|\mathcal{S}_1(\xi_g)\| : \xi_g \in \Xi_g\}, \; g = 1, \ldots, G, \tag{33}$$

$$t_{g_k}^*(\lambda, \bar{\lambda}; \alpha) = \sup_{\theta} \{|\mathbf{x}_{g_k}^T \theta| : \theta \in \Theta\}, \; k = 1, \ldots, n_g, \; g = 1, \ldots, G. \tag{34}$$

4.2.1. The Solution of Problem (33)

We consider the following equivalent problem of (33):

$$\frac{1}{2}\left(s_g^*(\lambda, \bar{\lambda}; \alpha)\right)^2 = \sup_{\xi_g} \left\{\frac{1}{2}\|\mathcal{S}_1(\xi_g)\|^2 : \xi_g \in \Xi_g\right\}. \tag{35}$$

We can see that the objective function of problem (35) is *continuously differentiable* and the feasible set is a ball. Thus, problem (35) is *nonconvex* because we need to *maximize* a convex function subject to a convex set. We first derive the necessary optimality conditions in Lemma 15 and then deduce the closed form solutions of problems (33) and (35) in Theorem 17.

Lemma 15 *Let Ξ_g^* be the set of optimal solutions of (35) and $\xi_g^* \in \Xi_g^*$. Then, the following hold:*
(i) *Suppose that ξ_g^* is an interior point of Ξ_g. Then, Ξ_g is a subset of $\mathcal{B}_{\infty}$.*
(ii) *Suppose that ξ_g^* is a boundary point of Ξ_g. Then, there exists $\mu^* \geq 0$ such that*

$$\mathcal{S}_1(\xi_g^*) = \mu^* \left(\xi_g^* - \mathbf{X}_g^T \mathbf{o}_{\alpha}(\lambda, \bar{\lambda})\right). \tag{36}$$

(iii) *Suppose that there exists $\xi_g^0 \in \Xi_g$ and $\xi_g^0 \notin \mathcal{B}_{\infty}$. Then, we have*
(iiia) *$\xi_g^* \notin \mathcal{B}_{\infty}$ and ξ_g^* is a boundary point of Ξ_g, i.e.,*

$$\|\xi_g^* - \mathbf{X}_g^T \mathbf{o}_{\alpha}(\lambda, \bar{\lambda})\| = \frac{1}{2}\|\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda})\| \|\mathbf{X}_g\|_2.$$

(iiib) *The optimality condition in Eq. (36) holds with $\mu^* > 0$.*

To show Lemma 15, we need the following proposition.

Proposition 16 (Hiriart-Urruty, 1988) *Suppose that $h \in \Gamma_0$ and $\mathcal{C}$ is a nonempty closed convex set. If $\mathbf{w}^* \in \mathcal{C}$ is a local maximum of h on $\mathcal{C}$, then $\partial h(\mathbf{w}^*) \subseteq N_{\mathcal{C}}(\mathbf{w}^*)$.*

We now present the proof of Lemma 15.
Proof To simplify notations, let

$$\mathbf{c} = \mathbf{X}_g^T \mathbf{o}_{\alpha}(\lambda, \bar{\lambda}) \text{ and } r = \frac{1}{2}\|\mathbf{v}_{\alpha}^{\perp}(\lambda, \bar{\lambda})\| \|\mathbf{X}_g\|_2. \tag{37}$$

By Eq. (1), we have

$$h(\mathbf{w}) := \frac{1}{2}\|\mathcal{S}_1(\mathbf{w})\|^2 = \frac{1}{2}\sum_i \left(|[\mathbf{w}]_i| - 1\right)_+^2. \tag{38}$$

It is easy to see that $h(\cdot)$ is continuously differentiable. Indeed, we have

$$\nabla h(\mathbf{w}) = \mathcal{S}_1(\mathbf{w}). \tag{39}$$

Then, problem (35) can be written as

$$\frac{1}{2}(s_g^*(\lambda,\bar{\lambda};\alpha))^2 = \sup_{\xi_g}\left\{h(\xi_g) = \frac{1}{2}\sum_i ([\xi_g]_i - 1)_+^2 \,:\, \xi_g \in \Xi_g\right\}, \tag{40}$$

where $\Xi_g = \{\xi_g : \|\xi_g - \mathbf{c}\| \le r\}$. Then, Proposition 16 results in

$$\mathcal{S}_1(\xi_g^*) = \nabla h(\xi_g^*) \subseteq \partial h(\xi_g^*) \subseteq N_{\Xi_g}(\xi_g^*). \tag{41}$$

(i) Suppose that ξ_g^* is an interior point of Ξ_g. Then, we have $N_{\Xi_g}(\xi_g^*) = 0$. By Eq. (41), we can see that

$$0 = \mathcal{S}_1(\xi_g^*) \Rightarrow 0 = \frac{1}{2}\|\mathcal{S}_1(\xi_g^*)\|^2 = \frac{1}{2}(s_g^*(\lambda,\bar{\lambda};\alpha))^2 = \sup_{\xi_g}\left\{\frac{1}{2}\|\mathcal{S}_1(\xi_g)\|^2 : \xi_g \in \Xi_g\right\}.$$

Therefore, we have

$$\|\mathcal{S}_1(\xi_g)\| = 0, \forall\, \xi_g \in \Xi_g. \tag{42}$$

Because $\mathcal{S}_1(\xi_g) = \xi_g - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g)$ (see Remark 6), Eq. (42) implies that

$$\xi_g = \mathbf{P}_{\mathcal{B}_\infty}(\xi_g), \forall\, \xi_g \in \Xi_g \Rightarrow \xi_g \in \mathcal{B}_\infty, \forall\, \xi_g \in \Xi_g.$$

This completes the proof.

(ii) Suppose that ξ_g^* is a boundary point of Ξ_g. We can see that

$$N_{\Xi_g}(\xi_g^*) = \{\mu(\xi_g^* - \mathbf{c}),\, \mu \ge 0\}. \tag{43}$$

Then, Eq. (36) follows by combining Eq. (43) and the optimality condition in (41).

(iii) Suppose that there exists $\xi_g^0 \in \Xi_g$ and $\xi_g^0 \notin \mathcal{B}_\infty$.

(iiia) The definition of ξ_g^0 leads to

$$0 < \|\mathcal{S}_1(\xi_g^0)\| \le \|\mathcal{S}_1(\xi_g^*)\| \Rightarrow \xi_g^* \notin \mathcal{B}_\infty.$$

Moreover, we can see that ξ_g^* is a boundary point of Ξ_g. Because if ξ_g^* is an interior point of Ξ_g, the first part implies that $\Xi_g \subset \mathcal{B}_\infty$. This contradicts with the existence of ξ_g^0. Thus, ξ_g^* must be a boundary point of Ξ_g, i.e. $\|\xi_g^* - \mathbf{c}\| = r$.

(iiib) Because ξ_g^* is a boundary point of Ξ_g, the second part implies that Eq. (36) holds. Moreover, from (iiia), we know that $\xi_g^* \notin \mathcal{B}_\infty$. Therefore, both sides of Eq. (36) are nonzero and thus $\mu^* > 0$. This completes the proof.

■

Based on the necessary optimality conditions in Lemma 15, we derive the closed form solutions of (33) and (35) in the following Theorem. The notations are the same as the ones in the proof of Lemma 15 [see Eq. (37) and Eq. (38)].

Theorem 17 *For problems (33) and (35), let* $\mathbf{c} = \mathbf{X}_g^T\mathbf{o}_\alpha(\lambda,\bar{\lambda})$, $r = \frac{1}{2}\|\mathbf{v}_\alpha^\perp(\lambda,\bar{\lambda})\|\|\mathbf{X}_g\|_2$ *and* Ξ_g^* *be the set of the optimal solutions.*

(i) *Suppose that* $\mathbf{c} \notin \mathcal{B}_\infty$, *i.e.,* $\|\mathbf{c}\|_\infty > 1$. *Let* $\mathbf{u} = r\mathcal{S}_1(\mathbf{c})/\|\mathcal{S}_1(\mathbf{c})\|$. *Then,*

$$s_g^*(\lambda,\bar{\lambda};\alpha) = \|\mathcal{S}_1(\mathbf{c})\| + r \;\text{ and }\; \Xi_g^* = \{\mathbf{c}+\mathbf{u}\}. \tag{44}$$

(ii) *Suppose that* $\mathbf{c}$ *is a boundary point of* $\mathcal{B}_\infty$, *i.e.,* $\|\mathbf{c}\|_\infty = 1$. *Then,*

$$s_g^*(\lambda,\bar{\lambda};\alpha) = r \;\text{ and }\; \Xi_g^* = \{\mathbf{c}+\mathbf{u} : \mathbf{u} \in N_{\mathcal{B}_\infty}(\mathbf{c}), \|\mathbf{u}\| = r\}. \tag{45}$$

(iii) *Suppose that* $\mathbf{c} \in \operatorname{int}\mathcal{B}_\infty$, *i.e.,* $\|\mathbf{c}\|_\infty < 1$. *Let* $i^* \in \mathcal{I}^* = \{i : |[\mathbf{c}]_i| = \|\mathbf{c}\|_\infty\}$. *Then,*

$$s_g^*(\lambda,\bar{\lambda};\alpha) = (\|\mathbf{c}\|_\infty + r - 1)_+, \tag{46}$$

$$\Xi_g^* = \begin{cases} \Xi_g, & \text{if } \Xi_g \subset \mathcal{B}_\infty, \\ \{\mathbf{c} + r\cdot\operatorname{sgn}([\mathbf{c}]_{i^*})\mathbf{e}_{i^*} : i^* \in \mathcal{I}^*\}, & \text{if } \Xi_g \not\subset \mathcal{B}_\infty \text{ and } \mathbf{c} \ne 0, \\ \{r\cdot\mathbf{e}_{i^*}, -r\cdot\mathbf{e}_{i^*} : i^* \in \mathcal{I}^*\}, & \text{if } \Xi_g \not\subset \mathcal{B}_\infty \text{ and } \mathbf{c} = 0, \end{cases}$$

where $\mathbf{e}_i$ *is the* i^{th} *standard basis vector.*

Proof

(i) Suppose that $\mathbf{c} \notin \mathcal{B}_\infty$. By the third part of Lemma 15, we have

$$\xi_g^* \notin \mathcal{B}_\infty,\;\; \|\xi_g^* - \mathbf{c}\| = r, \tag{47}$$

$$\xi_g^* - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) = \mathcal{S}_1(\xi_g^*) = \mu^*(\xi_g^* - \mathbf{c}),\, \mu^* > 0. \tag{48}$$

By Eq. (48), we can see that $\mu^* \ne 1$ because otherwise we would have $\mathbf{c} = \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) \in \mathcal{B}_\infty$. Moreover, we can only consider the cases with $\mu^* > 1$ because $\|\mathcal{S}_1(\xi_g^*)\| = \mu^* r$ and we aim to maximize $\|\mathcal{S}_1(\xi_g^*)\|$. Therefore, if we can find a solution with $\mu^* > 1$, there is no need to consider the cases with $\mu^* \in (0,1)$.

Suppose that $\mu^* > 1$. Then, Eq. (48) leads to

$$\mathbf{c} = \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) + \left(1 - \frac{1}{\mu^*}\right)\left(\xi_g^* - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)\right), \tag{49}$$

$$\xi_g^* = \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) + \frac{\mu^*}{\mu^* - 1}\left(\mathbf{c} - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)\right). \tag{50}$$

In view of part (iv) of Proposition 13 and Eq. (49), we have

$$\mathbf{P}_{\mathcal{B}_\infty}(\mathbf{c}) = \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*). \tag{51}$$

Therefore, Eq. (50) can be rewritten as

$$\mathcal{S}_1(\xi_g^*) = \xi_g^* - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) = \frac{\mu^*}{\mu^*-1}\left(\mathbf{c} - \mathbf{P}_{\mathcal{B}_\infty}(\mathbf{c})\right) = \frac{\mu^*}{\mu^*-1}\mathcal{S}_1(\mathbf{c}). \tag{52}$$

Combining Eq. (48) and Eq. (52), we have

$$\frac{\mu^*}{\mu^*-1}\|\mathcal{S}_1(\mathbf{c})\| = \mu^*\|\xi_g^* - \mathbf{c}\| = \mu^* r \Rightarrow \mu^* = 1 + \frac{\|\mathcal{S}_1(\mathbf{c})\|}{r} > 1. \tag{53}$$

The statement holds by plugging Eq. (53) and Eq. (51) into Eq. (50) and Eq. (52). Moreover, the above discussion implies that Ξ_g^* only contains one element as shown in Eq. (44).

(ii) Suppose that $\mathbf{c}$ is a boundary point of $\mathcal{B}_\infty$. Then, we can find a point $\xi_g^0 \in \Xi_g$ and $\xi_g^0 \notin \mathcal{B}_\infty$. By the third part of Lemma 15, we also have Eq. (47) and Eq. (48) hold. We claim that $\mu^* \in (0,1]$. The argument is as follows.

Suppose that $\mu^* > 1$. By the same argument as in the proof of the first part, we can see that Eq. (52) holds. Because $\mathcal{S}_1(\xi_g^*) \neq 0$ by Eq. (47), we have $\mathcal{S}_1(\mathbf{c}) \neq 0$. This implies that $\mathbf{c} \notin \mathcal{B}_\infty$. Thus, we have a contradiction, which implies that $\mu^* \in (0,1]$.

Let us consider the cases with $\mu^* = 1$. Because $\|\mathcal{S}_1(\xi_g^*)\| = \mu^* r$ [see Eq. (48)] and we want to maximize $\|\mathcal{S}_1(\xi_g^*)\|$, there is no need to consider the cases with $\mu^* \in (0,1)$ if we can find solutions of problem (33) with $\mu^* = 1$. Therefore, Eq. (48) leads to

$$\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) = \mathbf{c}.$$

By part (iii) of Proposition 13, we can see that

$$\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) = \mathbf{c} \Leftrightarrow \xi_g^* - \mathbf{c} \in N_{\mathcal{B}_\infty}(\mathbf{c}). \tag{54}$$

Combining Eq. (54) and Eq. (47), the statement holds immediately, which confirms that $\mu^* = 1$.

(iii) Suppose that $\mathbf{c}$ is an interior point of $\mathcal{B}_\infty$.

(a) We first consider the cases with $\Xi_g \subset \mathcal{B}_\infty$. Then, we can see that

$$\mathcal{S}_1(\xi) = 0,\ \forall \xi \in \Xi_g \Rightarrow \Xi_g^* = \Xi_g.$$

In other words, an arbitrary point of Ξ_g is an optimal solution of problem (33). Thus, we have

$$\begin{aligned}&\mathbf{c} + r\cdot \mathrm{sgn}(\mathbf{e}_{i^*})\mathbf{e}_{i^*} \in \Xi_g^*,\\ &s_g^*(\lambda,\bar{\lambda};\alpha) = 0.\end{aligned}$$

On the other hand, we can see that

$$\mathbf{c} - r\mathbf{e}_i \in \Xi_g \subset \mathcal{B}_\infty,\ \mathbf{c} + r\mathbf{e}_i \in \Xi_g \subset \mathcal{B}_\infty,\ i = 1,\ldots,n_g \Rightarrow \|\mathbf{c}\|_\infty + r \leq 1.$$

Therefore, we have

$$(\|\mathbf{c}\|_\infty + r - 1)_+ = 0,$$

and thus

$$s_g^*(\lambda,\bar{\lambda};\alpha) = (\|\mathbf{c}\|_\infty + r - 1)_+.$$

(b) Suppose that $\Xi_g \not\subset \mathcal{B}_\infty$, i.e., there exists $\xi^0 \in \Xi_g$ such that $\xi^0 \notin \mathcal{B}_\infty$. By the third part of Lemma 15, we have Eq. (47) and Eq. (48) hold. Moreover, in view of the proof of the first and second part, we can see that $\mu^* \in (0,1)$. Therefore, Eq. (48) leads to

$$(1-\mu^*)\xi_g^* + \mu^*\mathbf{c} = \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*). \tag{55}$$

By rearranging the terms of Eq. (55), we have

$$\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) - \mathbf{c} = (1-\mu^*)(\xi_g^* - \mathbf{c}). \tag{56}$$

Because $\mu^* \in (0,1)$, Eq. (55) implies that $\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)$ lies on the line segment connecting ξ_g^* and $\mathbf{c}$. Thus, we have

$$\|\xi_g^* - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)\| + \|\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) - \mathbf{c}\| = \|\xi_g^* - \mathbf{c}\| = r. \tag{57}$$

Therefore, to maximize $\|\mathcal{S}_1(\xi_g^*)\| = \|\xi_g^* - \mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)\|$, we need to minimize $\|\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*) - \mathbf{c}\|$. Because $\xi_g^* \notin \mathcal{B}_\infty$, we can see that $\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)$ is a boundary point of $\mathcal{B}_\infty$. Therefore, we need to solve the following minimization problem:

$$\min_{\phi_g}\{\|\phi_g - \mathbf{c}\| : \|\phi_g\|_\infty = 1\}. \tag{58}$$

Suppose that $\mathbf{c} = 0$. We can see that the set of optimal solutions of problem (58) is

$$\Phi_g^* = \{\mathbf{e}_i\}_{i=1}^{n_g} \cup \{-\mathbf{e}_i\}_{i=1}^{n_g}.$$

For each $\phi_g^* \in \Phi_g^*$, we set it as $\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)$. In view of Eq. (56) and Eq. (47), the statement follows immediately.

Suppose that $\mathbf{c} \neq 0$. Recall that $\mathcal{I}^* = \{i^* : |[\mathbf{c}]_{i^*}| = \|\mathbf{c}\|_\infty\}$. It is easy to see that

$$\Phi_g^* = \left\{\phi_{i^*} : [\phi_{i^*}]_k = \begin{cases}\mathrm{sgn}([\mathbf{c}]_{i^*}), & \text{if } k = i^*,\\ [\mathbf{c}]_k, & \text{otherwise},\end{cases}\quad i^* \in \mathcal{I}^*\right\}.$$

We can see that

$$\phi_{i^*} - \mathbf{c} = (1 - |[\mathbf{c}]_{i^*}|)\mathrm{sgn}([\mathbf{c}]_{i^*})\mathbf{e}_{i^*},\ i^* \in \mathcal{I}^*.$$

For each ϕ_{i^*}, we set it to $\mathbf{P}_{\mathcal{B}_\infty}(\xi_g^*)$. Then, we can see that the statement holds by Eq. (56) and Eq. (47). This completes the proof.

■

4.2.2. The Solution of Problem (34)

Problem (34) can be solved directly via the Cauchy-Schwarz inequality.

Theorem 18 *For problem (34), we have*

$$t^*_{g_k}(\lambda,\bar{\lambda};\alpha) = |\mathbf{x}_{g_k}^T \mathbf{o}_\alpha(\lambda,\bar{\lambda})| + \frac{1}{2}\|\mathbf{v}_\alpha^\perp(\lambda,\bar{\lambda})\|\|\mathbf{x}_{g_k}\|.$$

We are now ready to present the proposed screening rule TLFre.

4.3. The Proposed Two-Layer Screening Rules

To develop the two-layer screening rules for SGL, we only need to plug the supreme values $s^*_g(\lambda,\bar{\lambda};\alpha)$ and $t^*_{g_k}(\lambda,\bar{\lambda};\alpha)$ in (R1*) and (R2*). We present the TLFre rule as follows.

Theorem 19 *For the SGL problem in (3), suppose that we are given a grid of parameter values $\{\alpha_i\}_{i=1}^{\mathcal{I}}$ and $\lambda^{\alpha_i}_{\max} = \lambda_{i,0} > \lambda_{i,1} > \ldots > \lambda_{i,\mathcal{J}_i}$ for each α_i. Moreover, assume that $\beta^*(\lambda_{i,j-1},\alpha_i)$ is known for an integer $0 < j < \mathcal{J}_i$. Let $\theta^*(\lambda_{i,j-1},\alpha_i)$, $\mathbf{v}^\perp_{\alpha_i}(\lambda_{i,j},\lambda_{i,j-1})$ and $s^*_g(\lambda_{i,j},\lambda_{i,j-1};\alpha_i)$ be given by Eq. (18), Theorems 14 and 17, respectively. Then, for $g = 1,\ldots,G$, the following holds*

$$s^*_g(\lambda_{i,j},\lambda_{i,j-1};\alpha_i) < \alpha_i\sqrt{n_g} \Rightarrow \beta^*_g(\lambda_{i,j},\alpha_i) = 0. \qquad (\mathcal{L}_1)$$

For the gth group that does not pass the rule in $(\mathcal{L}_1)$, we have $[\beta^*_g(\lambda_{i,j},\alpha_i)]_k = 0$ if

$$t^*_{g_k}(\lambda_{i,j},\lambda_{i,j-1};\alpha_i) \le 1, \qquad (\mathcal{L}_2)$$

where

$$\begin{aligned} & t^*_{g_k}(\lambda_{i,j},\lambda_{i,j-1};\alpha_i) \\ &= \left|\mathbf{x}_{g_k}^T\left(\frac{\mathbf{y}-\mathbf{X}\beta^*(\lambda_{i,j-1},\alpha_i)}{\lambda_{i,j-1}} + \frac{1}{2}\mathbf{v}^\perp_{\alpha_i}(\lambda_{i,j},\lambda_{i,j-1})\right)\right| + \frac{1}{2}\|\mathbf{v}^\perp_{\alpha_i}(\lambda_{i,j},\lambda_{i,j-1})\|\|\mathbf{x}_{g_k}\|. \end{aligned}$$

$(\mathcal{L}_1)$ and $(\mathcal{L}_2)$ are the first and second layer screening rules of TLFre, respectively.

We also write Theorem 19 in an algorithmic manner in Algorithm 2. For each pair of parameter values $(\lambda_{i,j},\alpha_i)$, we first apply TLFre to identify the inactive groups and inactive features, namely, the zero components of $\beta^*(\lambda_{i,j},\alpha_i)$. Then, we remove the inactive groups and inactive features from the data matrix and apply an arbitrary solver to solve the SGL problem on the remaining features.

Specifically, lines 7 to 9 in Algorithm 2 compute the ball that contains $\theta^*(\lambda_{i,j},\alpha_i)$ in terms of $\theta^*(\lambda_{i,j-1},\alpha_i)$ (see remark 20). Lines 13 till 15 apply the first layer of TLFre, i.e., the group-level screening, to identify the inactive groups. Take the gth group for an example. If the first layer identifies the gth group as an inactive group, we can set $\beta^*_g(\lambda_{i,j},\alpha_i) = 0$. Otherwise, lines 20 till 23 apply the second layer of TLFre, i.e., the feature-level screening, to identify the inactive features in the gth group. The index set $\mathcal{G}_{i,j}$ stores the indices of inactive features, i.e., if $k \in \mathcal{G}_{i,j}$, then $[\beta^*(\lambda_{i,j},\alpha_i)]_k = 0$. After we scan the entire matrix, the index set $\mathcal{G}_{i,j}$ contains all the indices of inactive features identified by TLFre, i.e., $[\beta^*(\lambda_{i,j},\alpha_i)]_{\mathcal{G}_{i,j}} = 0$. Thus, the remaining unknowns are indeed $[\beta^*(\lambda_{i,j},\alpha_i)]_{\overline{\mathcal{G}}_{i,j}}$ (see line 27). Then, we apply an arbitrary solver to solve for $[\beta^*(\lambda_{i,j},\alpha_i)]_{\overline{\mathcal{G}}_{i,j}}$ on the reduced data matrix $\mathbf{X}_{\overline{\mathcal{G}}_{i,j}}$, where $\mathbf{X}_{\overline{\mathcal{G}}_{i,j}} = (\mathbf{x}_{k_1},\ldots,\mathbf{x}_{k_q})$, $q = |\overline{\mathcal{G}}_{i,j}|$, and $k_\ell \in \overline{\mathcal{G}}_{i,j}$ for $\ell = 1,\ldots,q$. Once we have solved for $[\beta^*(\lambda_{i,j},\alpha_i)]_{\overline{\mathcal{G}}_{i,j}}$, we indeed have solved for $\beta^*(\lambda_{i,j},\alpha_i)$ as $[\beta^*(\lambda_{i,j},\alpha_i)]_{\mathcal{G}_{i,j}} = 0$. Then, line 29 computes $\theta^*(\lambda_{i,j},\alpha_i)$ by Eq. (18), based on which we can estimate $\theta^*(\lambda_{i,j+1},\alpha_i)$ and apply TLFre to identify the zero components of $\beta^*(\lambda_{i,j+1},\alpha_i)$.

Remark 20 *As shown by Theorem 19 and Algorithm 2, TLFre estimates the dual optimum at $(\lambda_{i,j},\alpha_i)$, i.e., $\theta^*(\lambda_{i,j},\alpha_i)$, in terms of a known dual optimum at a different pair of parameter values $(\lambda_{i,j-1},\alpha_i)$, i.e., $\theta^*(\lambda_{i,j-1},\alpha_i)$. Then, we can apply TLFre to identify the inactive groups and inactive features and solve the TLFre problems on a reduced data matrix. Thus, to initialize TLFre, we may need to solve the SGL problem on the entire data matrix once to compute $\theta^*(\lambda_{i,0},\alpha_i)$, which can be time consuming. However, we note that, Theorem 9 not only gives the effective interval of λ for a fixed value of α, but also a closed form solution of the dual optimum for any $\lambda \ge \lambda^\alpha_{\max}$. Thus, as we have done in Algorithm 2 (see line 3), we can always set $\lambda_{i,0} = \lambda^{\alpha_i}_{\max}$ and $\theta^*(\lambda_{i,0},\alpha_i) = \mathbf{y}/\lambda_{i,0}$ to initialize the computation of TLFre. This implies that, if we combine TLFre and an arbitrary solver, we do not need to solve the SGL problem on the entire data matrix even once.*

5. Extension to Nonnegative Lasso

The framework of TLFre is applicable to a large class of sparse models with multiple regularizers. As an example, we extend TLFre to nonnegative Lasso:

$$\min_{\beta\in\mathbb{R}^p}\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|^2 + \lambda\|\beta\|_1 : \beta \in \mathbb{R}^p_+\right\}, \qquad (59)$$

where $\lambda > 0$ is the regularization parameter and $\mathbb{R}^p_+$ is the nonnegative orthant of $\mathbb{R}^p$. In Section 5.1, we transform the constraint $\beta \in \mathbb{R}^p_+$ to a regularizer and derive the Fenchel's dual of the nonnegative Lasso problem. We then motivate the screening method—called DPC since the key step is to **d**ecom**p**ose a **c**onvex set via Fenchel's Duality Theorem—via the KKT conditions in Section 5.2. In Section 5.3, we analyze the geometric properties of the dual problem and derive the set of parameter values leading to zero solutions. We then develop the screening method for nonnegative Lasso in Section 5.4.

5.1. The Fenchel's Dual of Nonnegative Lasso

Let $\mathbf{I}_{\mathbb{R}^p_+}$ be the indicator function of $\mathbb{R}^p_+$. By noting that $\mathbf{I}_{\mathbb{R}^p_+} = \lambda\mathbf{I}_{\mathbb{R}^p_+}$ for any $\lambda > 0$, we can rewrite the nonnegative Lasso problem in (59) as

$$\min_{\beta\in\mathbb{R}^p}\frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|^2 + \lambda\|\beta\|_1 + \lambda\mathbf{I}_{\mathbb{R}^p_+}(\beta). \qquad (60)$$

In other words, we incorporate the constraint $\beta \in \mathbb{R}^p_+$ to the objective function as an additional regularizer. As a result, the nonnegative lasso problem in (60) has two regularizers. Thus, similar to SGL, we can derive the Fenchel's dual of nonnegative Lasso via Theorem 1.

We now proceed by following a similar procedure as the one in Section 3.1. We note that the nonnegative Lasso problem in (60) can also be formulated as the one in (10) with $f(\cdot) = \frac{1}{2}\|\cdot\|^2$ and $\Omega(\beta) = \|\beta\|_1 + \mathbf{I}_{\mathbb{R}^p_+}(\beta)$. To derive the Fenchel's dual of nonnegative Lasso, we need to find f^* and Ω^* by Theorem 1. Since we have already seen that $f^*(\cdot) = \frac{1}{2}\|\cdot\|^2$ in Section 3.1, we only need to find $\Omega^*(\cdot)$. The following result is indeed a counterpart of Lemma 3.

Lemma 21 *Let $\Omega_2(\beta) = \|\beta\|_1$, $\Omega_3 = \mathbf{I}_{\mathbb{R}^p_+}(\beta)$, and $\Omega(\beta) = \Omega_2(\beta) + \Omega_3(\beta)$. Then,*

(i) $(\Omega_2)^*(\xi) = \mathbf{I}_{\mathcal{B}_\infty}(\xi)$ *and* $(\Omega_3)^*(\xi) = \mathbf{I}_{\mathbb{R}^p_-}(\xi)$, *where $\mathbb{R}^p_-$ is the nonpositive orthant of $\mathbb{R}^p$.*

(ii) $\Omega^*(\xi) = ((\Omega_2)^* \boxdot (\Omega_3)^*)(\xi) = \mathbf{I}_{\mathbb{R}^p_-}(\xi - \mathbf{1})$, *where $\mathbb{R}^p \ni \mathbf{1} = (1, 1, \ldots, 1)^T$.*

We omit the proof of Lemma 21 since it is very similar to that of Lemma 3.

Remark 22 *Consider the second part of Lemma 21. Let $\mathcal{C}_\mathbf{1} = \{\xi : \xi \leq \mathbf{1}\}$, where "$\leq$" is defined component-wisely. We can see that*

$$\mathbf{I}_{\mathbb{R}^p_-}(\xi - \mathbf{1}) = \mathbf{I}_{\mathcal{C}_\mathbf{1}}(\xi).$$

On the other hand, Lemma 7 implies that

$$\Omega^*(\xi) = ((\Omega_2)^* \boxdot (\Omega_3)^*)(\xi) = \mathbf{I}_{\mathcal{B}_\infty + \mathbb{R}^p_-}(\xi).$$

Thus, we have $\mathcal{B}_\infty + \mathbb{R}^p_- = \mathcal{C}_\mathbf{1}$. The second part of Lemma 21 decomposes each $\xi \in \mathcal{B}_\infty + \mathbb{R}^p_-$ into two components: $\mathbf{1}$ and $\xi - \mathbf{1}$ that belong to $\mathcal{B}_\infty$ and $\mathbb{R}^p_-$, respectively.

By Theorem 1 and Lemma 21, we can derive the Fenchel's dual of nonnegative Lasso in the following theorem (which is indeed the counterpart of Theorem 5).

Theorem 23 *For the nonnegative Lasso problem, the following hold:*

(i) *The Fenchel's dual of nonnegative Lasso is given by:*

$$\inf_\theta \left\{ \frac{1}{2}\left\|\frac{\mathbf{y}}{\lambda} - \theta\right\|^2 - \frac{1}{2}\|\mathbf{y}\|^2 : \langle \mathbf{x}_i, \theta\rangle \leq 1,\ i = 1, \ldots, p \right\}. \tag{61}$$

(ii) *Let $\beta^*(\lambda)$ and $\theta^*(\lambda)$ be the optimal solutions of problems (60) and (61), respectively. Then,*

$$\lambda\theta^*(\lambda) = \mathbf{y} - \mathbf{X}\beta^*(\lambda), \tag{62}$$

$$\mathbf{X}^T\theta^*(\lambda) \in \partial\|\beta^*(\lambda)\|_1 + \partial\mathbf{I}_{\mathbb{R}^p_+}(\beta^*(\lambda)). \tag{63}$$

We omit the proof of Theorem 23 since it is very similar to that of Theorem 5.

5.2. Motivation of the Screening Method via KKT Conditions

The key to develop the DPC rule for nonnegative lasso is the KKT condition in (63). We can see that $\partial\|\mathbf{w}\|_1 = \mathrm{SGN}(\mathbf{w})$ and

$$\partial\mathbf{I}_{\mathbb{R}^p_+}(\mathbf{w}) = \left\{ \xi \in \mathbb{R}^p : [\xi]_i = \begin{cases} 0, & \text{if } [\mathbf{w}]_i > 0, \\ \rho,\ \rho \leq 0, & \text{if } [\mathbf{w}]_i = 0, \end{cases} \right\}.$$

Therefore, the KKT condition in (63) implies that

$$\langle \mathbf{x}_i, \theta^*(\lambda)\rangle \in \begin{cases} 1, & \text{if } [\beta^*(\lambda)]_i > 0, \\ \varrho,\ \varrho \leq 1, & \text{if } [\beta^*(\lambda)]_i = 0. \end{cases} \tag{64}$$

By Eq. (64), we have the following rule:

$$\langle \mathbf{x}_i, \theta^*(\lambda)\rangle < 1 \Rightarrow [\beta^*(\lambda)]_i = 0. \tag{R3}$$

Because $\theta^*(\lambda)$ is unknown, we can apply (R3) to identify the inactive features—which have 0 coefficients in $\beta^*(\lambda)$. Similar to TLFre, we can first find a region Θ that contains $\theta^*(\lambda)$. Then, we can relax (R3) as follows:

$$\sup_{\theta\in\Theta} \langle \mathbf{x}_i, \theta\rangle < 1 \Rightarrow [\beta^*(\lambda)]_i = 0. \tag{R3*}$$

Inspired by (R3*), we develop DPC via the following three steps:

Step 1. Given λ, we estimate a region Θ that contains $\theta^*(\lambda)$.

Step 2. We solve the optimization problem $\omega_i = \sup_{\theta\in\Theta} \langle \mathbf{x}_i, \theta\rangle$.

Step 3. By plugging in ω_i computed from **Step 2**, (R3*) leads to the desired screening method DPC for nonnegative Lasso.

5.3. Geometric Properties of the Fenchel's Dual of Nonnegative Lasso

In view of the Fenchel's dual of nonnegative Lasso in (61), we can see that the optimal solution is indeed the projection of $\mathbf{y}/\lambda$ onto the feasible set $\mathcal{F} = \{\theta : \langle \mathbf{x}_i, \theta\rangle \leq 1,\ i = 1, \ldots, p\}$, i.e.,

$$\theta^*(\lambda) = \mathbf{P}_\mathcal{F}\left(\frac{\mathbf{y}}{\lambda}\right). \tag{65}$$

Therefore, if $\mathbf{y}/\lambda \in \mathcal{F}$, Eq. (65) implies that $\theta^*(\lambda) = \mathbf{y}/\lambda$. If further $\mathbf{y}/\lambda$ is an interior point of $\mathcal{F}$, R3* implies that $\beta^*(\lambda) = 0$. The next theorem gives the set of parameter values leading to 0 solutions of nonnegative Lasso.

Theorem 24 *For the nonnegative Lasso problem (60), Let $\lambda_{\max} = \max_i \langle \mathbf{x}_i, \mathbf{y}\rangle$. Then, the following statements are equivalent:*

(i) $\frac{\mathbf{y}}{\lambda} \in \mathcal{F}$, (ii) $\theta^*(\lambda) = \frac{\mathbf{y}}{\lambda}$, (iii) $\beta^*(\lambda) = 0$, (iv) $\lambda \geq \lambda_{\max}$.

We omit the proof of Theorem 24 since it is very similar to that of Theorem 9.

5.4. The Proposed Screening Rule for Nonnegative Lasso

We follow the three steps in Section 5.2 to develop the screening rule for nonnegative Lasso. We first estimate a region that contains $\theta^*(\lambda)$. Because $\theta^*(\lambda)$ admits a closed form solution with $\lambda \geq \lambda_{\max}$ by Theorem 24, we focus on the cases with $\lambda < \lambda_{\max}$.

Theorem 25 *For the nonnegative Lasso problem, suppose that $\theta^*(\bar{\lambda})$ is known with $\bar{\lambda} \leq \lambda_{\max}$. For any $\lambda \in (0, \bar{\lambda})$, we define*

$$\mathbf{n}(\bar{\lambda}) = \begin{cases} \frac{\mathbf{y}}{\bar{\lambda}} - \theta^*(\bar{\lambda}), & \text{if } \bar{\lambda} < \lambda_{\max}^{\alpha}, \\ \mathbf{x}_*, & \text{if } \bar{\lambda} = \lambda_{\max}, \end{cases} \quad \text{where } \mathbf{x}_* = \operatorname{argmax}_{\mathbf{x}_i} \langle \mathbf{x}_i, \mathbf{y} \rangle,$$

$$\mathbf{v}(\lambda, \bar{\lambda}) = \frac{\mathbf{y}}{\lambda} - \theta^*(\bar{\lambda}),$$

$$\mathbf{v}(\lambda, \bar{\lambda})^{\perp} = \mathbf{v}(\lambda, \bar{\lambda}) - \frac{\langle \mathbf{v}(\lambda, \bar{\lambda}), \mathbf{n}(\bar{\lambda}) \rangle}{\|\mathbf{n}(\bar{\lambda})\|^2} \mathbf{n}(\bar{\lambda}).$$

Then, the following hold:

(i) $\mathbf{n}(\bar{\lambda}) \in N_{\mathcal{F}}(\theta^*(\bar{\lambda}))$,

(ii) $\left\| \theta^*(\lambda) - \left(\theta^*(\bar{\lambda}) + \frac{1}{2} \mathbf{v}^{\perp}(\lambda, \bar{\lambda}) \right) \right\| \leq \frac{1}{2} \|\mathbf{v}^{\perp}(\lambda, \bar{\lambda})\|$.

Proof We only show that $\mathbf{n}(\lambda_{\max}) \in N_{\mathcal{F}}(\theta^*(\lambda_{\max}))$ since the proof of the other statement is very similar to that of Theorem 14.

By Proposition 13 and Theorem 24, it suffices to show that

$$\langle \mathbf{x}_*, \theta - \mathbf{y}/\lambda_{\max} \rangle \leq 0, \, \forall \, \theta \in \mathcal{F}. \tag{66}$$

Because $\theta \in \mathcal{F}$, we have $\langle \mathbf{x}_*, \theta \rangle \leq 1$. The definition of $\mathbf{x}_*$ implies that $\langle \mathbf{x}_*, \mathbf{y}/\lambda_{\max} \rangle = 1$. Thus, the inequality in (66) holds, which completes the proof. ■

Theorem 25 implies that $\theta^*(\lambda)$ is in a ball—denoted by $\mathcal{B}(\lambda, \bar{\lambda})$—of radius $\frac{1}{2}\|\mathbf{v}^{\perp}(\lambda, \bar{\lambda})\|$ centered at $\theta^*(\bar{\lambda}) + \frac{1}{2}\mathbf{v}^{\perp}(\lambda, \bar{\lambda})$. Simple calculations lead to

$$\omega_i = \sup_{\theta \in \mathcal{B}(\lambda, \bar{\lambda})} \langle \mathbf{x}_i, \theta \rangle = \left\langle \mathbf{x}_i, \theta^*(\bar{\lambda}) + \frac{1}{2} \mathbf{v}^{\perp}(\lambda, \bar{\lambda}) \right\rangle + \frac{1}{2} \|\mathbf{v}^{\perp}(\lambda, \bar{\lambda})\| \|\mathbf{x}_i\|. \tag{67}$$

By plugging ω_i into (R3*), we have the DPC screening rule for nonnegative Lasso as follows.

Theorem 26 *For the nonnegative Lasso problem, suppose that we are given a sequence of parameter values $\lambda_{\max} = \lambda^{(0)} > \lambda^{(1)} > \ldots > \lambda^{(\mathcal{J})}$. Then, $[\beta^*(\lambda^{(j+1)})]_i = 0$ if $\beta^*(\lambda^{(j)})$ is known and the following holds:*

$$\left\langle \mathbf{x}_i, \frac{\mathbf{y} - \mathbf{X}\beta^*(\lambda^{(j)})}{\lambda^{(j)}} + \frac{1}{2} \mathbf{v}^{\perp}(\lambda^{(j+1)}, \lambda^{(j)}) \right\rangle + \frac{1}{2} \|\mathbf{v}^{\perp}(\lambda^{(j+1)}, \lambda^{(j)})\| \|\mathbf{x}_i\| < 1. \tag{68}$$

6. Experiments

We evaluate TLFre for SGL and DPC for nonnegative Lasso in Sections 6.1 and 6.2, respectively, on both synthetic and real data sets. To the best of knowledge, the TLFre and DPC are the first screening methods for SGL and nonnegative Lasso, respectively. The code is available at `http://dpc-screening.github.io/`.

6.1. TLFre for SGL

We perform experiments to evaluate TLFre on synthetic and real data sets in Sections 6.1.1 and 6.1.2, respectively. To measure the performance of TLFre, we compute the *rejection ratios* of $(\mathcal{L}_1)$ and $(\mathcal{L}_2)$, respectively. Specifically, let m be the number of features that have 0 coefficients in the solution, $\overline{\mathcal{G}}$ be the index set of groups that are discarded by $(\mathcal{L}_1)$ and $\overline{p}$ be the number of inactive features that are detected by $(\mathcal{L}_2)$. The rejection ratios of $(\mathcal{L}_1)$ and $(\mathcal{L}_2)$ are defined by $r_1 = \frac{\sum_{g \in \overline{\mathcal{G}}} n_g}{m}$ and $r_2 = \frac{|\overline{p}|}{m}$, respectively. Moreover, we report the *speedup* gained by TLFre, i.e., the ratio of the running time of solver without screening to the running time of solver with TLFre.

To determine appropriate values of α and λ by cross validation or stability selection, we can run TLFre with as many parameter values as we need. Given a data set, for illustrative purposes only, we select seven values of α from $\{\tan(\psi) : \psi = 5°, 15°, 30°, 45°, 60°, 75°, 85°\}$. Then, for each value of α, we run TLFre along a sequence of 100 values of λ equally spaced on the logarithmic scale of $\lambda/\lambda_{\max}^{\alpha}$ from 1 to 0.01. Thus, 700 pairs of parameter values of (λ, α) are sampled in total.

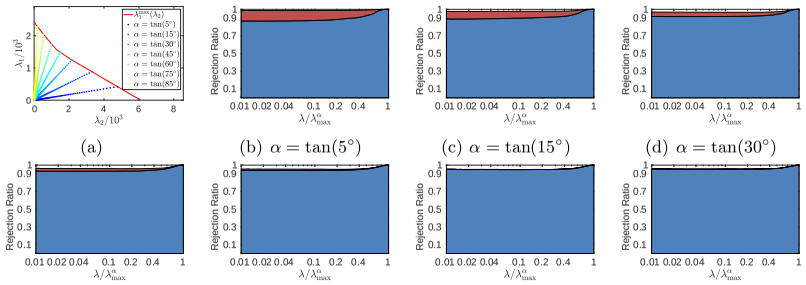
We use *sgLeastR* from the SLEP package (Liu et al., 2009) as the solver for SGL, which is one of the state-of-the-arts (Zhang et al., 2018b) [see Section G for a comparison between sgLeastR and another popular solver (Lin et al., 2014)].

For the non-screening case, we apply sgLeastR directly to solve SGL with different parameter values. We use zero as the initial point.

6.1.1. Simulation Studies

We perform experiments on two synthetic data sets that are commonly used in the literature (Tibshirani et al., 2012; Zou and Hastie, 2005). The true model is $\mathbf{y} = \mathbf{X}\beta^* + 0.01\epsilon$, $\epsilon \sim N(0, 1)$. We generate two data sets with 1000×160000 entries: Synthetic 1 and Synthetic 2. We randomly divide the 160000 features into 16000 groups. For Synthetic 1, the entries of the data matrix $\mathbf{X}$ are i.i.d. standard Gaussian with pairwise correlation zero, i.e., $\operatorname{corr}(\mathbf{x}_i, \mathbf{x}_i) = 0$. For Synthetic 2, the entries of the data matrix $\mathbf{X}$ are drawn from i.i.d. standard Gaussian with pairwise correlation $0.5^{|i-j|}$, i.e., $\operatorname{corr}(\mathbf{x}_i, \mathbf{x}_j) = 0.5^{|i-j|}$. To construct β^*, we first randomly select γ_1 percent of groups. Then, for each selected group, we randomly select γ_2 percent of features. The selected components of β^* are populated from a standard Gaussian and the remaining ones are set to 0. We set $\gamma_1 = \gamma_2 = 10$ for Synthetic 1 and $\gamma_1 = \gamma_2 = 20$ for Synthetic 2.

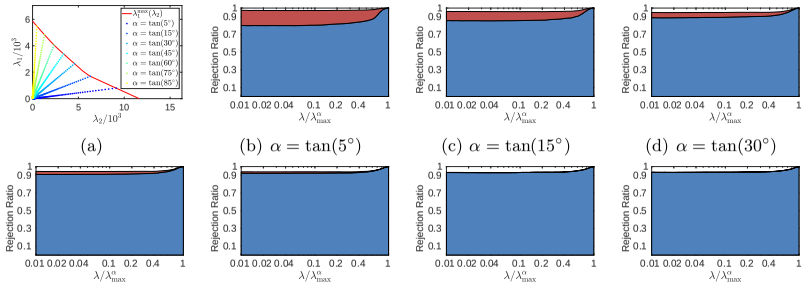
Fig. 1(a) and Fig. 2(a) show the plots of $\lambda_1^{\max}(\lambda_2)$ (see Corollary 12) and the sampled parameter values of λ and α (recall that $\lambda_1 = \alpha\lambda$ and $\lambda_2 = \lambda$). For the other figures, the blue and red regions represent the rejection ratios of $(\mathcal{L}_1)$ and $(\mathcal{L}_2)$, respectively. We can see that TLFre is very effective in discarding inactive groups/features; that is, more than



(a) (b) $\alpha = \tan(5°)$ (c) $\alpha = \tan(15°)$ (d) $\alpha = \tan(30°)$

(e) $\alpha = \tan(45°)$ (f) $\alpha = \tan(60°)$ (g) $\alpha = \tan(75°)$ (h) $\alpha = \tan(85°)$

Figure 1: Rejection ratios of TLFre on the Synthetic 1 data set.



(a) (b) $\alpha = \tan(5°)$ (c) $\alpha = \tan(15°)$ (d) $\alpha = \tan(30°)$

(e) $\alpha = \tan(45°)$ (f) $\alpha = \tan(60°)$ (g) $\alpha = \tan(75°)$ (h) $\alpha = \tan(85°)$

Figure 2: Rejection ratios of TLFre on the Synthetic 2 data set.

90% of inactive features can be detected. Moreover, we can observe that the first layer screening ($\mathcal{L}_1$) becomes more effective with a larger α. Intuitively, this is because the group Lasso penalty plays a more important role in enforcing the sparsity with a larger value of α (recall that $\lambda_1 = \alpha\lambda$). The top and middle parts of Table 1 indicate that the speedup gained by TLFre is very significant (up to 80 times) and TLFre is very efficient. Compared to the running time of the solver without screening, the running time of TLFre is negligible. The running time of TLFre includes that of computing $\|\mathbf{X}_g\|_2$, $g = 1, \ldots, G$, which can be efficiently computed by the power method (Halko et al., 2011). Indeed, this can be shared for TLFre with different parameter values.

6.1.2. Experiments on Real Data Sets

We perform experiments on two commonly used real data sets – the Alzheimer's Disease Neuroimaging Initiative (ADNI) data set (`http://adni.loni.usc.edu/`) and the news20.binary (Chang and Lin, 2011) data set. Details of these data sets are as follows.

Table 1: Running time (in seconds) for solving SGL along a sequence of 100 tuning parameter values of λ equally spaced on the logarithmic scale of $\lambda/\lambda^{\alpha}_{\max}$ from 1.0 to 0.01 by (a): the solver (Liu et al., 2009) without screening; (b): the solver combined with TLFre. The data sets are Synthetic 1 and Synthetic 2.

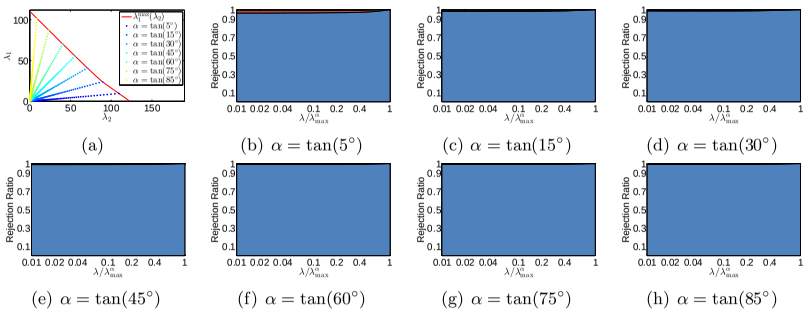
	α	tan(5°)	tan(15°)	tan(30°)	tan(45°)	tan(60°)	tan(75°)	tan(85°)
Synthetic 1	solver	15555.28	16124.08	16106.24	16293.04	16426.44	16836.16	16862.36
	TLFre	37.84	43.08	46.92	51.16	54.24	53.4	53.24
	TLFre+solver	184.16	275.36	680.08	1196.04	1465.16	1629.96	1657.00
	speedup	**84.46**	**58.55**	**23.68**	**13.62**	**11.21**	**10.33**	**10.18**
Synthetic 2	solver	15709.72	16615.16	16286.04	16826.48	16919.41	17178.08	17350.36
	TLFre	41.72	47.28	54.08	54.72	59.08	58.56	60.12
	TLFre+solver	328.52	906.72	1452.28	1702.61	1912.76	2181.23	2180.24
	speedup	**47.82**	**18.32**	**11.21**	**9.88**	**8.85**	**7.88**	**7.96**

Table 2: Running time (in seconds) for solving SGL along a sequence of 100 tuning parameter values of λ equally spaced on the logarithmic scale of $\lambda/\lambda^{\alpha}_{\max}$ from 1.0 to 0.01 by (a): the solver (Liu et al., 2009) without screening; (b): the solver combined with TLFre. We perform experiments on the ADNI data sets. The response vectors are GMV and WMV, respectively.

	α	tan(5°)	tan(15°)	tan(30°)	tan(45°)	tan(60°)	tan(75°)	tan(85°)
ADNI+GMV	solver	30652.56	30755.63	30838.29	31096.10	30850.78	30728.27	30572.35
	TLFre	64.08	64.56	64.96	65.00	64.89	65.17	65.05
	TLFre+solver	372.04	383.17	386.80	402.72	391.63	385.98	382.62
	speedup	**82.39**	**80.27**	**79.73**	**77.22**	**78.78**	**79.61**	**79.90**
ADNI+WMV	solver	29751.27	29823.15	29927.52	30078.62	30115.89	29927.58	29896.77
	TLFre	62.91	63.33	63.39	63.99	64.13	64.31	64.36
	TLFre+solver	363.43	364.78	386.15	393.03	395.87	400.11	399.48
	speedup	**81.86**	**81.76**	**77.50**	**76.53**	**76.08**	**74.80**	**74.84**

Table 3: Running time (in seconds) for solving SGL along a sequence of 100 tuning parameter values of λ equally spaced on the logarithmic scale of $\lambda/\lambda^{\alpha}_{\max}$ from 1.0 to 0.01 by (a): the solver (Liu et al., 2009) without screening; (b): the solver combined with TLFre. We perform experiments on the news20.binary data sets.

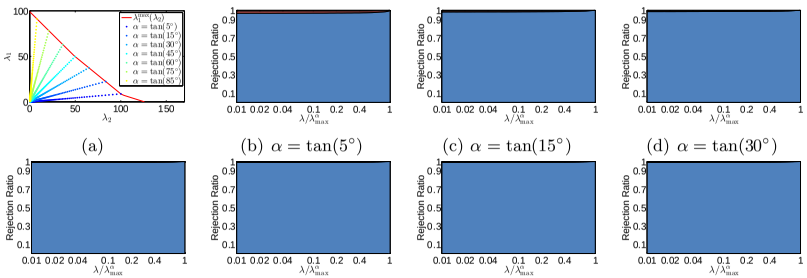
	α	tan(5°)	tan(15°)	tan(30°)	tan(45°)	tan(60°)	tan(75°)	tan(85°)
news20.binary	solver	1233401.05	1231570.22	1277630.92	1299353.68	1292879.86	1216554.09	1347890.85
	TLFre	350.51	337.75	332.01	346.98	352.78	353.52	362.43
	TLFre+solver	1434.49	1465.37	1539.37	1598.87	1608.90	1659.78	1709.35
	speedup	**859.82**	**840.45**	**829.97**	**812.67**	**803.58**	**793.21**	**788.54**



(a) (b) $\alpha = \tan(5^\circ)$ (c) $\alpha = \tan(15^\circ)$ (d) $\alpha = \tan(30^\circ)$

(e) $\alpha = \tan(45^\circ)$ (f) $\alpha = \tan(60^\circ)$ (g) $\alpha = \tan(75^\circ)$ (h) $\alpha = \tan(85^\circ)$

Figure 3: Rejection ratios of TLFre on the ADNI data set with grey matter volume as response.



(a) (b) $\alpha = \tan(5^\circ)$ (c) $\alpha = \tan(15^\circ)$ (d) $\alpha = \tan(30^\circ)$

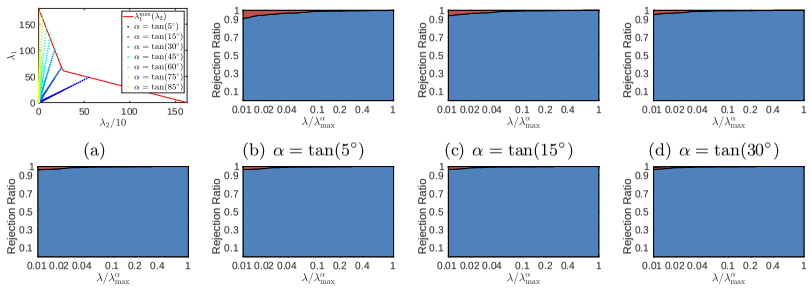
(e) $\alpha = \tan(45^\circ)$ (f) $\alpha = \tan(60^\circ)$ (g) $\alpha = \tan(75^\circ)$ (h) $\alpha = \tan(85^\circ)$

Figure 4: Rejection ratios of TLFre on the ADNI data set with white matter volume as response.

ADNI The data matrix of the ADNI data set consists of 747 samples with 426040 single nucleotide polymorphisms (SNPs), which are divided into 94765 groups. The response vectors are the grey matter volume (GMV) and white matter volume (WMV), respectively.

news20.binary The news20.binary data set consists of 19996 samples with 1355191 features, which are divided into 67760 groups. The entries of the response vectors are the labels of the corresponding samples, which are 1 or -1.

Fig. 3(a), Fig. 4(a), and Fig. 5(a) show the plots of $\lambda_1^{\max}(\lambda_2)$ (see Corollary 12) and the sampled parameter values of α and λ. The other figures present the rejection ratios of $(\mathcal{L}_1)$ and $(\mathcal{L}_2)$ by blue and red regions, respectively. We can see that almost all of the inactive groups/features are discarded by TLFre. The rejection ratios of $r_1 + r_2$ are very close to 1 in all cases. Tables 2 and 3 show that TLFre leads to a very significant speedup (about 80 times on the ADNI data set and 800 times on the news20.binary data set). In other words, the solver without screening needs about 8.5 and 360 hours to solve the 100 SGL problems for



(a) (b) $\alpha = \tan(5^\circ)$ (c) $\alpha = \tan(15^\circ)$ (d) $\alpha = \tan(30^\circ)$

(e) $\alpha = \tan(45^\circ)$ (f) $\alpha = \tan(60^\circ)$ (g) $\alpha = \tan(75^\circ)$ (h) $\alpha = \tan(85^\circ)$

Figure 5: Rejection ratios of TLFre on the news20.binary data set.

each value of α on the ADNI and news20.binary data set, respectively. However, combined with TLFre, the solver needs only 6∼8 minutes and 24∼29 minutes, respectively. Moreover, we can observe that the computational cost of TLFre is negligible compared to that of the solver without screening. This demonstrates the efficiency of TLFre.

6.2. DPC for Nonnegative Lasso

In this experiment, we evaluate the performance of DPC on two synthetic data sets and six real data sets. We integrate DPC with the solver, nnLeastR, (Liu et al., 2009) to solve the nonnegative Lasso problem along a sequence of 100 parameter values of λ equally spaced on the logarithmic scale of $\lambda/\lambda_{\max}$ from 1.0 to 0.01. The two synthetic data sets are the same as the ones we used in Section 6.1.1. To construct β^*, we first randomly select 10 percent of features. The corresponding components of β^* are populated from a standard Gaussian and the remaining ones are set to 0.

We use *nnLeastR* from the SLEP package (Liu et al., 2009) as the solver for nonnegative Lasso, which is one of the state-of-the-arts [see Section G for a comparison between nnLeastR and another popular solver (Lin et al., 2014)].

For the non-screening case, we apply nnLeastR directly to solve SGL with different parameter values. We use zero as the initial point.

We list the six real data sets and the corresponding experimental settings as follows.

Breast Cancer data set (West et al., 2001; Shevade and Keerthi, 2003): this data set contains 7129 gene expression values of 44 tumor samples (thus the data matrix $\mathbf{X}$ is of 44×7129). The response vector $\mathbf{y} \in \{1, -1\}^{44}$ contains the binary label of each sample.

Leukemia data set (Armstrong et al., 2002): this data set contains 11225 gene expression values of 52 samples ($\mathbf{X} \in \mathbb{R}^{52\times 11225}$). The response vector $\mathbf{y}$ contains the binary label of each sample.

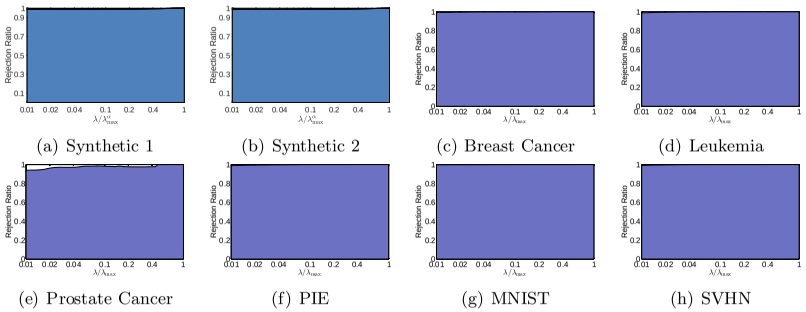
Prostate Cancer data set (Petricoin et al., 2002): this data set contains 15154 measurements of 132 patients ($\mathbf{X} \in \mathbb{R}^{132\times 15154}$). By protein mass spectrometry, the features are indexed by time-of-flight values, which are related to the mass over charge ratios of the

constituent proteins in the blood. The response vector $\mathbf{y}$ contains the binary label of each sample.

PIE face image data set (Sim et al., 2003; Cai et al., 2007): this data set contains 11554 gray face images (each has 32×32 pixels) of 68 people, taken under different poses, illumination conditions and expressions. In each trial, we first randomly pick an image as the response $\mathbf{y} \in \mathbb{R}^{1024}$, and then use the remaining images to form the data matrix $\mathbf{X} \in \mathbb{R}^{1024 \times 11553}$. We run 100 trials and report the average performance of DPC.

MNIST handwritten digit data set (Lecun et al., 1998): this data set contains grey images of scanned handwritten digits (each has 28×28 pixels). The training and test sets contain $60,000$ and $10,000$ images, respectively. We first randomly select 5000 images for each digit from the training set and get a data matrix $\mathbf{X} \in \mathbb{R}^{784 \times 50000}$. Then, in each trial, we randomly select an image from the testing set as the response $\mathbf{y} \in \mathbb{R}^{784}$. We run 100 trials and report the average performance of the screening rules.

Street View House Number (SVHN) data set (Netzer et al., 2001): this data set contains color images of street view house numbers (each has 32×32 pixels), including 73257 images for training and 26032 for testing. In each trial, we first randomly select an image as the response $\mathbf{y} \in \mathbb{R}^{3072}$, and then use the remaining ones to form the data matrix $\mathbf{X} \in \mathbb{R}^{3072 \times 99288}$. We run 20 trials and report the average performance.



(a) Synthetic 1 (b) Synthetic 2 (c) Breast Cancer (d) Leukemia

(e) Prostate Cancer (f) PIE (g) MNIST (h) SVHN

Figure 6: Rejection ratios of DPC on eight data sets.

Table 4: Running time (in seconds) for solving nonnegative Lasso along a sequence of 100 tuning parameter values of λ equally spaced on the logarithmic scale of $\lambda/\lambda_{\max}$ from 1.0 to 0.01 by (a): the solver (Liu et al., 2009) without screening; (b): the solver combined with DPC.

	Synthetic 1	Synthetic 2	Breast Cancer	Leukemia	Prostate Cancer	PIE	MNIST	SVHN
solver	13140.84	13853.84	23.40	34.04	187.82	674.04	3000.69	24761.07
DPC	3.08	3.59	0.03	0.06	0.23	1.16	3.53	30.59
DPC+solver	61.56	69.52	2.18	3.37	6.37	5.01	9.31	104.93
speedup	**213.52**	**199.30**	**10.73**	**10.10**	**29.49**	**134.54**	**322.31**	**235.98**

We present the *rejection ratios*—the ratio of the number of inactive features identified by DPC to the actual number of inactive features—in Fig. 6. We also report the running time of the solver with and without DPC, the time for running DPC, and the corresponding *speedup* in Table 4.

Fig. 6 shows that DPC is very effective in identifying the inactive features even for small parameter values: the rejection ratios are very close to 100% for the entire sequence of parameter values on the eight data sets. Table 4 shows that DPC leads to a very significant speedup on all the data sets. Take MNIST as an example. The solver without DPC takes 50 minutes to solve the 100 nonnegative Lasso problems. However, combined with DPC, the solver only needs 10 seconds. The speedup gained by DPC on the MNIST data set is thus more than 300 times. Similarly, on the SVHN data set, the running time for solving the 100 nonnegative Lasso problems by the solver without DPC is close to seven hours. However, combined with DPC, the solver takes less than two minutes to solve all the 100 nonnegative Lasso problems, leading to a speedup about 230 times. Moreover, we observe that the computational cost of DPC is very low—which is negligible compared to that of the solver without DPC.

7. Conclusion

In this paper, we propose a novel feature reduction method for SGL via decomposition of convex sets. We also derive the set of parameter values that lead to zero solutions of SGL. To the best of our knowledge, TLFre is the first method which is applicable to sparse models with multiple sparsity-inducing regularizers. More importantly, the proposed approach provides novel framework for developing screening methods for complex sparse models with multiple sparsity-inducing regularizers, e.g., ℓ_1 SVM that performs both sample and feature selection, fused Lasso and tree Lasso with more than two regularizers. To demonstrate the flexibility of the proposed framework, we develop the DPC screening rule for the nonnegative Lasso problem. Experiments on both synthetic and real data sets demonstrate the effectiveness and efficiency of TLFre and DPC. We plan to generalize the idea of TLFre to ℓ_1 SVM, fused Lasso and tree Lasso, which are expected to consist of multiple layers of screening.

Acknowledgments

This work was supported in part by NSFC (61822604, 61836006).

Appendix A. The Lagrangian Dual Problem of SGL

We derive the dual problem of SGL in (4) via the Lagrangian multiplier method.

By introducing an auxiliary variable

$$\mathbf{z} = \mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g \beta_g, \tag{69}$$

the SGL problem in (3) becomes:

$$\min_{\beta}\left\{\frac{1}{2}\|\mathbf{z}\|^2+\alpha\lambda\sum_{g=1}^{G}\sqrt{n_g}\|\beta_g\|+\lambda\|\beta\|_1:\mathbf{z}=\mathbf{y}-\sum_{g=1}^{G}\mathbf{X}_g\beta_g\right\}.$$

Let $\lambda\theta$ be the Lagrangian multiplier, the Lagrangian function is

$$L(\beta,\mathbf{z};\theta)=\frac{1}{2}\|\mathbf{z}\|^2+\alpha\lambda\sum_{g=1}^{G}\sqrt{n_g}\|\beta_g\|+\lambda\|\beta\|_1+\langle\lambda\theta,\mathbf{y}-\sum_{g=1}^{G}\mathbf{X}_g\beta_g-\mathbf{z}\rangle \tag{70}$$

$$=\alpha\lambda\sum_{g=1}^{G}\sqrt{n_g}\|\beta_g\|+\lambda\|\beta\|_1-\lambda\langle\theta,\sum_{g=1}^{G}\mathbf{X}_g\beta_g\rangle+\frac{1}{2}\|\mathbf{z}\|^2-\lambda\langle\theta,\mathbf{z}\rangle+\lambda\langle\theta,\mathbf{y}\rangle. \tag{71}$$

Let

$$f_1(\beta)=\sum_{g=1}^{G}f_1^g(\beta_g)=\sum_{g=1}^{G}\left(\alpha\lambda\sqrt{n_g}\|\beta_g\|+\lambda\|\beta_g\|_1-\lambda\langle\theta,\mathbf{X}_g\beta_g\rangle\right),$$

$$f_2(\mathbf{z})=\frac{1}{2}\|\mathbf{z}\|^2-\lambda\langle\theta,\mathbf{z}\rangle.$$

To derive the dual problem, we need to minimize the Lagrangian function with respect to β and $\mathbf{z}$. In other words, we need to minimize f_1 and f_2, respectively. We first consider

$$\min_{\beta_g}f_1^g(\beta_g)=\alpha\lambda\sqrt{n_g}\|\beta_g\|+\lambda\|\beta_g\|_1-\lambda\langle\theta,\mathbf{X}_g\beta_g\rangle.$$

By the Fermat's rule, we have

$$0\in\partial f_1^g(\beta_g)=\alpha\lambda\sqrt{n_g}\partial\|\beta_g\|+\lambda\partial\|\beta_g\|_1-\lambda\mathbf{X}_g^T\theta, \tag{72}$$

which leads to

$$\mathbf{X}_g^T\theta=\alpha\sqrt{n_g}\zeta_1+\zeta_2,\ \ \zeta_1\in\partial\|\beta_g\|,\ \zeta_2\in\partial\|\beta_g\|_1. \tag{73}$$

By noting that

$$\langle\zeta_1,\beta_g\rangle=\|\beta_g\|,\ \ \langle\zeta_2,\beta_g\rangle=\|\beta_g\|_1,$$

we have

$$\langle\mathbf{X}_g^T\theta,\beta_g\rangle=\alpha\sqrt{n_g}\|\beta_g\|+\|\beta_g\|_1.$$

Thus, we can see that

$$0=\min_{\beta_g}f_1^g(\beta_g). \tag{74}$$

Moreover, because $\zeta_1\in\partial\|\beta_g\|$, $\zeta_2\in\partial\|\beta_g\|_1$, Eq. (73) implies that

$$\mathbf{X}_g^T\theta\in\alpha\sqrt{n_g}\mathcal{B}+\mathcal{B}_\infty. \tag{75}$$

To minimize f_2, the Fermat's rule results in

$$\mathbf{z}=\lambda\theta, \tag{76}$$

and thus

$$-\frac{\lambda^2}{2}\|\theta\|^2=\min_{\mathbf{z}}f_2(z). \tag{77}$$

In view of Eq. (70), Eq. (74), Eq. (77) and Eq. (75), the dual problem of SGL can be written as

$$\sup_{\theta}\left\{\frac{1}{2}\|\mathbf{y}\|^2-\frac{1}{2}\left\|\theta-\frac{\mathbf{y}}{\lambda}\right\|^2:\mathbf{X}_g^T\theta\in\alpha\sqrt{n_g}\mathcal{B}+\mathcal{B}_\infty,\ g=1,\ldots,G\right\},$$

which is equivalent to (4).

Recall that $\beta^*(\lambda,\alpha)$ and $\theta^*(\lambda,\alpha)$ are the primal and dual optimal solutions of SGL, respectively. By Eq. (69), Eq. (72) and Eq. (76), we can see that the KKT conditions are

$$\lambda\theta^*(\lambda,\alpha)=\mathbf{y}-\mathbf{X}\beta^*(\lambda,\alpha),$$
$$\mathbf{X}_g^T\theta^*(\lambda,\alpha)\in\alpha\sqrt{n_g}\partial\|\beta_g^*(\lambda,\alpha)\|+\partial\|\beta_g^*(\lambda,\alpha)\|_1,\ g=1,\ldots,G.$$

Appendix B. Proof of Theorem 5

To show Theorem 5, we need the Fenchel-Young inequality as follows:

Lemma 27 [Fenchel-Young inequality] (Borwein and Lewis, 2006) *Any point* $\mathbf{z}\in\mathbb{R}^n$ *and* $\mathbf{w}$ *in the domain of a function* $h:\mathbb{R}^n\to(-\infty,\infty]$ *satisfy the inequality*

$$h(\mathbf{w})+h^*(\mathbf{z})\geq\langle\mathbf{w},\mathbf{z}\rangle.$$

Equality holds if and only if $\mathbf{z}\in\partial h(\mathbf{w})$.

We now give the proof of Theorem 5.

Proof We first show the first part. Combining Theorem 1 and Lemma 3, the Fenchel's dual of SGL can be written as:

$$\sup_{\theta}-\frac{\lambda^2}{2}\|\theta\|^2-\sum\nolimits_{g=1}^{G}\lambda\mathbf{I}_{\mathcal{B}}\left(\frac{\mathbf{X}_g^T\theta-\mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_g^T\theta)}{\alpha\sqrt{n_g}}\right)+\lambda\langle\mathbf{y},\theta\rangle,$$

which is equivalent to problem (17).

To show the second half, we have the following inequalities by Fenchel-Young inequality:

$$f(\mathbf{y}-\mathbf{X}\beta)+f^*(\lambda\theta)\geq\langle\mathbf{y}-\mathbf{X}\beta,\lambda\theta\rangle, \tag{78}$$
$$\lambda\Omega(\beta)+\lambda\Omega^*(\mathbf{X}^T\theta)\geq\lambda\langle\beta,\mathbf{X}^T\theta\rangle. \tag{79}$$

We sum the inequalities in (78) and (79) together and get

$$f(\mathbf{y}-\mathbf{X}\beta)+\lambda\Omega(\beta)\geq-f^*(\lambda\theta)-\lambda\Omega^*(\mathbf{X}^T\theta)+\lambda\langle\mathbf{y},\theta\rangle. \tag{80}$$

Clearly, the left and right hand sides of inequality (80) are the objective functions of the pair of Fenchel's problems. Because $\mathrm{dom}\, f = \mathbb{R}^N$ and $\mathrm{dom}\,\Omega = \mathbb{R}^p$, we have

$$0 \in \mathrm{int}\,(\mathrm{dom}\, f - \mathbf{y} + \mathbf{X}\mathrm{dom}\,\Omega).$$

Thus, the equality in (80) holds at $\beta^*(\lambda,\alpha)$ and $\theta^*(\lambda,\alpha)$, i.e.,

$$f(\mathbf{y} - \mathbf{X}\beta^*(\lambda,\alpha)) + \lambda\Omega(\beta^*(\lambda,\alpha)) = -f^*(\lambda\theta^*(\lambda,\alpha)) - \lambda\Omega^*(\mathbf{X}^T\theta^*(\lambda,\alpha)) + \lambda\langle \mathbf{y}, \theta^*(\lambda,\alpha)\rangle.$$

Therefore, the equality holds in both (78) and (79) at $\beta^*(\lambda,\alpha)$ and $\theta^*(\lambda,\alpha)$. By applying Lemma 27 again, we have

$$\begin{aligned}\lambda\theta^*(\lambda,\alpha) &\in \partial f(\mathbf{y} - \mathbf{X}\beta^*(\lambda,\alpha)) = \mathbf{y} - \mathbf{X}\beta^*(\lambda,\alpha),\\ \mathbf{X}^T\theta^*(\lambda,\alpha) &\in \partial\Omega(\beta^*(\lambda,\alpha)) = \partial\Omega_1^\alpha(\beta^*(\lambda,\alpha)) + \partial\Omega_2(\beta^*(\lambda,\alpha)),\end{aligned}$$

which completes the proof. ■

Appendix C. Proof of Theorem 9

Proof The equivalence between (i) and (ii) can be see from the fact that

$$\theta^*(\lambda,\alpha) = \mathbf{P}_{\mathcal{F}^\alpha}(\mathbf{y}/\lambda).$$

Next, we show (ii)⇔(iii). Let us first show (ii)⇒(iii). We assume that $\theta^*(\lambda,\alpha) = \mathbf{y}/\lambda$. By the KKT condition in (18), we have $\mathbf{X}\beta^*(\lambda,\alpha) = 0$. We claim that $\beta^*(\lambda,\alpha) = 0$. To see this, let $\beta' \neq 0$ with $\mathbf{X}\beta' = 0$ be another optimal solution of SGL. We denote by h the objective function of SGL in (3). Then, we have

$$h(0) = \frac{1}{2}\|\mathbf{y}\|^2 < h(\beta') = \frac{1}{2}\|\mathbf{y}\|^2 + \lambda_1\sum_g \sqrt{n_g}\|\beta'_g\| + \lambda_2\|\beta'\|_1,$$

which contradicts with the assumption $\beta' \neq 0$ is also an optimal solution. This contradiction indicates that $\beta^*(\lambda,\alpha)$ must be 0. The converse direction, i.e., (ii)⇐(iii), can be derived directly from the KKT condition in Eq. (18).

Finally, we show the equivalence (i)⇔(iv). Indeed, in view of the dual problem in (21), we can see that $\mathbf{y}/\lambda \in \mathcal{F}^\alpha$ if and only if

$$\|\mathcal{S}_1(\mathbf{X}_g^T\mathbf{y}/\lambda)\| \leq \alpha\sqrt{n_g},\ g = 1,\ldots,G. \tag{81}$$

We note that $\|\mathcal{S}_1(\mathbf{X}_g^T\mathbf{y}/\lambda)\|$ is monotonically decreasing with respect to λ. Thus, the inequality in (81) is equivalent to (iv), which completes the proof. ■

Appendix D. Proof of Corollary 12

Before we prove Corollary 12, we first derive the Fenchel's dual of (2). By letting $f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$ and $\Omega(\beta) = \lambda_1\sum_{g=1}^G\sqrt{n_g}\|\beta_g\| + \lambda_2\|\beta\|_1$, the SGL problem in (2) can be written as:

$$\min_\beta\ f(\mathbf{y} - \mathbf{X}\beta) + \Omega(\beta).$$

Then, by Fenchel's Duality Theorem, the Fenchel's dual problem of (2) is

$$\inf_\theta\left\{\frac{1}{2}\|\mathbf{y}-\theta\|^2 - \frac{1}{2}\|\mathbf{y}\|^2 : \left\|\mathcal{S}_{\lambda_2}(\mathbf{X}_g^T\theta)\right\| \leq \lambda_1\sqrt{n_g},\ g = 1,\ldots,G\right\}. \tag{82}$$

Let $\bar\beta^*(\lambda_1,\lambda_2)$ and $\bar\theta^*(\lambda_1,\lambda_2)$ be the optimal solutions of problem (2) and (82). The optimality conditions can be written as

$$\bar\theta^*(\lambda_1,\lambda_2) = \mathbf{y} - \mathbf{X}\bar\beta^*(\lambda_1,\lambda_2), \tag{83}$$

$$\mathbf{X}_g^T\bar\theta^*(\lambda_1,\lambda_2) \in \lambda_1\sqrt{n_g}\partial\|\bar\beta_g^*(\lambda_1,\lambda_2)\| + \lambda_2\partial\|\bar\beta_g^*(\lambda_1,\lambda_2)\|_1,\ g = 1,\ldots,G. \tag{84}$$

We denote by $\mathcal{F}(\lambda_1,\lambda_2)$ the feasible set of problem (82). It is easy to see that

$$\bar\theta^*(\lambda_1,\lambda_2) = \mathbf{P}_{\mathcal{F}(\lambda_1,\lambda_2)}(\mathbf{y}).$$

We now present the proof of Corollary 12.

Proof For notational convenience, let

(i). $\mathbf{y} \in \mathcal{F}(\lambda_1,\lambda_2)$,

(ii). $\bar\theta^*(\lambda_1,\lambda_2) = \mathbf{y}$,

(iii). $\bar\beta^*(\lambda_1,\lambda_2) = 0$,

(iv). $\lambda_1 \geq \lambda_1^{\max}(\lambda_2) = \max_g \frac{1}{\sqrt{n_g}}\|\mathcal{S}_{\lambda_2}(\mathbf{X}_g^T\mathbf{y})\|$.

The first half of the statement is (iii)⇔(iv). Indeed, by a similar argument as in the proof of Theorem 9, we can see that the above statements are all equivalent to each other.

We now show the second half. We first show that

$$\lambda_1 \geq \lambda_1^{\max} \Rightarrow \bar\beta^*(\lambda_1,\lambda_2) = 0. \tag{85}$$

By the first half, we only need to show

$$\lambda_1 \geq \lambda_1^{\max} \Rightarrow \mathbf{y} \in \mathcal{F}(\lambda_1,\lambda_2).$$

Indeed, the definition of λ_1 implies that

$$\|\mathbf{X}_g^T\mathbf{y}\| \leq \lambda_1\sqrt{n_g},\ g = 1,\ldots,G.$$

We note that for any $\lambda_2 \geq 0$, we have

$$\|\mathcal{S}_{\lambda_2}(\mathbf{X}_g^T\mathbf{y})\| \leq \|\mathbf{X}_g^T\mathbf{y}\|.$$

Therefore, we can see that

$$\|\mathcal{S}_{\lambda_2}(\mathbf{X}_g^T\mathbf{y})\| \leq \|\mathbf{X}_g^T\mathbf{y}\| \leq \lambda_1\sqrt{n_g},\ g=1,\ldots,G \Rightarrow \mathbf{y}\in\mathcal{F}(\lambda_1,\lambda_2).$$

The proof of (85) is complete.

Similarly, to show that $\lambda_2 \geq \lambda_2^{\max} \Rightarrow \bar{\beta}^*(\lambda_1,\lambda_2)$, we only need to show

$$\lambda_2 \geq \lambda_2^{\max} \Rightarrow \mathbf{y}\in\mathcal{F}(\lambda_1,\lambda_2).$$

By the definition of λ_2, we can see that

$$\|\mathbf{X}_g^T\mathbf{y}\|_\infty \leq \lambda_2,\ g=1,\ldots,G \Rightarrow \|\mathcal{S}_{\lambda_2}(\mathbf{X}_g^T\mathbf{y})\| = 0 \leq \lambda_1\sqrt{n_g},\ g=1,\ldots,G.$$

Thus, we have $\mathbf{y}\in\mathcal{F}(\lambda_1,\lambda_2)$, which completes the proof. ■

Appendix E. Proof of Theorem 14

Proof

(i) Suppose that $\bar{\lambda} < \lambda_{\max}^\alpha$. Theorem 9 implies that $\mathbf{y}/\bar{\lambda} \notin \mathcal{F}^\alpha$ and thus

$$\mathbf{y}/\bar{\lambda} - \mathbf{P}_{\mathcal{F}^\alpha}\left(\mathbf{y}/\bar{\lambda}\right) = \mathbf{y}/\bar{\lambda} - \theta^*(\bar{\lambda},\alpha) \neq 0.$$

By the third part of Proposition 13, we can see that

$$\mathbf{y}/\bar{\lambda} - \theta^*(\bar{\lambda},\alpha) \in N_{\mathcal{F}^\alpha}(\theta^*(\bar{\lambda},\alpha)). \tag{86}$$

Thus, the statement holds for all $\bar{\lambda} < \lambda_{\max}^\alpha$.

Suppose that $\bar{\lambda} = \lambda_{\max}^\alpha$. By Theorem 9, we have

$$\theta^*(\bar{\lambda},\alpha) = \mathbf{y}/\bar{\lambda} \in \mathcal{F}^\alpha.$$

In view of the definition of $\mathbf{X}_*$, we have

$$\left\|\mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\| = \alpha\sqrt{n_*},$$

where n_* is the number of feature contained in $\mathbf{X}_*$. Moreover, it is easy to see that

$$\|\mathcal{S}_1(\mathbf{X}_*^T\theta)\| \leq \alpha\sqrt{n_*},\ \forall\theta\in\mathcal{F}^\alpha.$$

Therefore, to prove the statement, we need to show that

$$\left\langle \mathbf{X}_*\mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \theta - \frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right\rangle \leq 0,\ \forall\theta\in\mathcal{F}^\alpha. \tag{87}$$

Recall Remark 6, we have the following identity [see Eq. (20)]

$$\mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right) = \mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha} - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right).$$

Thus, we have

$$\begin{aligned}
&\left\langle \mathbf{X}_*\mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \theta - \frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right\rangle \qquad (88)\\
=&\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{X}_*^T\left(\theta - \frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right) + \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right) - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle\\
=&\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{X}_*^T\theta - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle - \left\|\mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\|^2\\
=&\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{X}_*^T\theta - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle - \alpha^2 n_*.
\end{aligned}$$

Consider the first term on the right hand side of Eq. (88), we have

$$\begin{aligned}
&\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{X}_*^T\theta - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle \qquad (89)\\
=&\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{X}_*^T\theta - \mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_*^T\theta) + \mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_*^T\theta) - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle\\
=&\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathcal{S}_1(\mathbf{X}_*^T\theta)\right\rangle + \left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_*^T\theta) - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle.
\end{aligned}$$

Let $\mathcal{P} = \{i : [\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}]_i > 1\}$ and $\mathcal{N} = \{i : [\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}]_i < -1\}$. We note that the second term on the right hand side of Eq. (89) can be written as

$$\begin{aligned}
&\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_*^T\theta) - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle \qquad (90)\\
=&\sum_{i\in\mathcal{P}}\left([\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}]_i - 1\right)\left([\mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_*^T\theta)]_i - 1\right) + \sum_{j\in\mathcal{N}}\left([\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}]_j + 1\right)\left([\mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_*^T\theta)]_j + 1\right).
\end{aligned}$$

Because $\|\mathbf{P}_{\mathcal{B}_\infty}(\mathbf{X}_*^T\theta)\|_\infty \leq 1$, we can see that Eq. (90) is non-positive. Therefore, by Eq. (89), we have

$$\begin{aligned}
\left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathbf{X}_*^T\theta - \mathbf{P}_{\mathcal{B}_\infty}\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\rangle \leq& \left\langle \mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right), \mathcal{S}_1(\mathbf{X}_*^T\theta)\right\rangle \qquad (91)\\
\leq& \left\|\mathcal{S}_1\left(\mathbf{X}_*^T\frac{\mathbf{y}}{\lambda_{\max}^\alpha}\right)\right\| \|\mathcal{S}_1(\mathbf{X}_*^T\theta)\|\\
\leq& \alpha^2 n_*.
\end{aligned}$$

Combining Eq. (88) and the inequality in (91), we can see that the inequality in (87) holds. Thus, the statement holds for $\bar{\lambda} = \lambda_{\max}^\alpha$. This completes the proof.

(ii) We now show the second half. It is easy to see that the statement is equivalent to

$$\|\theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha)\|^2 \le \langle \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha),\, \mathbf{v}_\alpha^\perp(\lambda,\bar\lambda)\rangle. \tag{92}$$

Thus, we will show that the inequality in (92) holds.

Because of the first half, we have

$$\langle \mathbf{n}_\alpha(\bar\lambda),\, \theta-\theta^*(\bar\lambda,\alpha)\rangle \le 0,\ \ \forall\, \theta\in\mathcal{F}^\alpha. \tag{93}$$

By letting $\theta=\theta^*(\lambda,\alpha)$, the inequality in (93) leads to

$$\langle \mathbf{n}_\alpha(\bar\lambda),\, \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha)\rangle \le 0. \tag{94}$$

In view of the first half and by letting $\theta=0$, the inequality in (93) leads to

$$\langle \mathbf{n}_\alpha(\bar\lambda),\, 0-\theta^*(\bar\lambda,\alpha)\rangle \le 0 \Rightarrow \begin{cases} \langle \mathbf{n}_\alpha(\bar\lambda),\, \mathbf{y}\rangle \ge 0, & \text{if } \bar\lambda=\lambda_{\max}^\alpha,\\ \|\mathbf{y}\|/\bar\lambda \ge \|\theta^*(\bar\lambda,\alpha)\|, & \text{if } \bar\lambda<\lambda_{\max}^\alpha. \end{cases} \tag{95}$$

Moreover, the first half also leads to $\frac{\mathbf{y}}{\lambda}-\theta^*(\lambda,\alpha)\in N_{\mathcal{F}^\alpha}(\theta^*(\lambda,\alpha))$. Thus, we have

$$\langle \frac{\mathbf{y}}{\lambda}-\theta^*(\lambda,\alpha), \theta-\theta^*(\lambda,\alpha)\rangle \le 0,\ \ \forall\, \theta\in\mathcal{F}^\alpha. \tag{96}$$

By letting $\theta=\theta^*(\bar\lambda,\alpha)$, the inequality in (96) results in

$$\langle \frac{\mathbf{y}}{\lambda}-\theta^*(\lambda,\alpha), \theta^*(\bar\lambda,\alpha)-\theta^*(\lambda,\alpha)\rangle \le 0,\ \ \forall\, \theta\in\mathcal{F}^\alpha. \tag{97}$$

We can see that the inequality in (97) is equivalent to

$$\|\theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha)\|^2 \le \langle \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha),\, \mathbf{v}_\alpha(\lambda,\bar\lambda)\rangle. \tag{98}$$

On the other hand, the right hand side of (92) can be rewritten as

$$\begin{aligned} &\langle \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha),\, \mathbf{v}_\alpha^\perp(\lambda,\bar\lambda)\rangle \qquad (99)\\ =&\langle \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha),\, \mathbf{v}_\alpha(\lambda,\bar\lambda)\rangle - \langle \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha),\, \mathbf{v}_\alpha(\lambda,\bar\lambda)-\mathbf{v}_\alpha^\perp(\lambda,\bar\lambda)\rangle\\ =&\langle \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha),\, \mathbf{v}_\alpha(\lambda,\bar\lambda)\rangle - \left\langle \theta^*(\lambda,\alpha)-\theta^*(\bar\lambda,\alpha),\, \frac{\langle \mathbf{v}_\alpha(\lambda,\bar\lambda),\mathbf{n}_\alpha(\bar\lambda)\rangle}{\|\mathbf{n}_\alpha(\bar\lambda)\|^2}\mathbf{n}_\alpha(\bar\lambda)\right\rangle. \end{aligned}$$

In view of (94), (98) and (99), we can see that (92) holds if $\langle \mathbf{v}_\alpha(\lambda,\bar\lambda),\mathbf{n}_\alpha(\bar\lambda)\rangle \ge 0$. Indeed,

$$\begin{aligned} \langle \mathbf{v}_\alpha(\lambda,\bar\lambda),\mathbf{n}_\alpha(\bar\lambda)\rangle &= \left\langle \mathbf{y}/\lambda-\theta^*(\bar\lambda,\alpha),\mathbf{n}_\alpha(\bar\lambda)\right\rangle \qquad (100)\\ &= \left(1/\lambda-1/\bar\lambda\right)\langle \mathbf{y},\mathbf{n}_\alpha(\bar\lambda)\rangle + \langle \mathbf{y}/\bar\lambda-\theta^*(\bar\lambda,\alpha),\mathbf{n}_\alpha(\bar\lambda)\rangle \end{aligned}$$

Consider the first term on the right hand side of Eq. (100). By the first half of (95), we have

$$\langle \mathbf{y},\mathbf{n}_\alpha(\bar\lambda)\rangle \ge 0,\ \ \text{if } \bar\lambda=\lambda_{\max}^\alpha. \tag{101}$$

Suppose that $\bar\lambda<\lambda_{\max}^\alpha$. By the second half of (95), we can see that

$$\langle \mathbf{y},\mathbf{n}_\alpha(\bar\lambda)\rangle = \langle \mathbf{y}, \mathbf{y}/\bar\lambda-\theta^*(\bar\lambda,\alpha)\rangle \ge 1/\bar\lambda\|\mathbf{y}\|^2-\|\mathbf{y}\|\|\theta^*(\bar\lambda,\alpha)\| \ge 0. \tag{102}$$

Consider the second term on the right hand side of Eq. (100). It is easy to see that

$$\langle \mathbf{y}/\bar\lambda-\theta^*(\bar\lambda,\alpha),\mathbf{n}_\alpha(\bar\lambda)\rangle = \begin{cases} 0, & \text{if } \bar\lambda=\lambda_{\max}^\alpha,\\ \|\mathbf{n}_\alpha(\bar\lambda)\|^2, & \text{if } \bar\lambda<\lambda_{\max}^\alpha. \end{cases} \tag{103}$$

Combining (101), (102) and Eq. (103), we have $\langle \mathbf{v}_\alpha(\lambda,\bar\lambda),\mathbf{n}_\alpha(\bar\lambda)\rangle \ge 0$, which completes the proof.

■

Appendix F. Proof of Theorem 18

Proof To simplify notations, let $\mathbf{o}=\mathbf{o}_\alpha(\lambda,\bar\lambda)$, $r=\frac{1}{2}\|\mathbf{v}_\alpha^\perp(\lambda,\bar\lambda)\|$ and $t_{g_k}^*=t_{g_k}^*(\lambda,\bar\lambda;\alpha)$. Therefore, the set Θ in Eq. (31) can be written as

$$\Theta=\{\mathbf{o}+\mathbf{v}:\|\mathbf{v}\|\le r\}.$$

Then, problem (34) becomes

$$t_{g_k}^*=\sup_{\mathbf{v}}\{|\mathbf{x}_{g_k}^T(\mathbf{o}+\mathbf{v})|:\|\mathbf{v}\|\le r\}.$$

We can see that

$$|\mathbf{x}_{g_k}^T(\mathbf{o}+\mathbf{v})| \le |\mathbf{x}_{g_k}^T\mathbf{o}|+|\mathbf{x}_{g_k}^T\mathbf{v}| \le |\mathbf{x}_{g_k}^T\mathbf{o}|+\|\mathbf{x}_{g_k}\|\|\mathbf{v}\| \le |\mathbf{x}_{g_k}^T\mathbf{o}|+\|\mathbf{x}_{g_k}\|r.$$

Thus, we have

$$t_{g_k}^* \le |\mathbf{x}_{g_k}^T\mathbf{o}|+\|\mathbf{x}_{g_k}\|r.$$

Consider $\mathbf{v}_1^*=r\mathbf{x}_{g_k}/\|\mathbf{x}_{g_k}\|$ and $\mathbf{v}_2^*=-r\mathbf{x}_{g_k}/\|\mathbf{x}_{g_k}\|$. It is easy to see that $\mathbf{o}+\mathbf{v}_1^*\in\Theta$ and $\mathbf{o}+\mathbf{v}_2^*\in\Theta$. Then,

$$|\mathbf{x}_{g_k}^T(\mathbf{o}+\mathbf{v}_i^*)| = |\mathbf{x}_{g_k}^T\mathbf{o}|+\|\mathbf{x}_{g_k}\|r, \quad \text{for } i=1,2,$$

which leads to

$$t_{g_k}^*=|\mathbf{x}_{g_k}^T\mathbf{o}|+\|\mathbf{x}_{g_k}\|r.$$

This completes the proof. ■

Appendix G. Efficiency of sgLeastR, nnLeastR, and APCG

We show that sgLeastR and nnLeastR are more appropriate solvers than APCG (Lin et al., 2014)—which is an accelerated coordinate descent method—in terms of the efficiency in solving SGL and nonnegative Lasso problems, respectively. Indeed, sgLeastR and nnLeastR are among the state-of-the-arts (Zhang et al., 2018a).

From the theoretical perspective, sgLeastR has a convergence rate of $\mathcal{O}(1/k^2)$ (k is the number of iterations) for the SGL problem, and so does nnLeastR for nonnegative Lasso (Liu et al., 2009). However, as pointed out by (Lin et al., 2014), without strong convexity, APCG recovers a special case of APPROX (Fercoq and Richtrik, 2013) and has a sublinear convergence rate of $\mathcal{O}(\frac{m}{m+k})$, where m is the number of groups. Notice that, sparse learning techniques usually deal with problems with $p \gg N$ (Tibshirani et al., 2015) (p is the number of features and N is the number of samples), in which strong convexity does not hold. Thus, sgLeastR and nnLeastR converge faster than APCG in solving SGL and nonnegative Lasso, respectively.

Synthetic 1 This data set is the same as that in Section 6.1.1, which consists of 1000 samples with 160000 features. The entries of the data matrix are i.i.d. standard Gaussian with pairwise correlation zero. For SGL, we randomly divide the features into 16000 groups.

E2006-tfidf (Chang and Lin, 2011) The E2006-tfidf data set consists of 3308 samples with 150360 features. The features include the volitility in twelve months and tf-idf of unigrams. For SGL, we randomly divide the 150360 features into 15036 groups.

Tables 5 and 6 show that sgLeastR and nnLeastR significantly outperform APCG in terms of the running time. Thus, we use sgLeastR and nnLeastR to solve SGL and nonnegative Lasso, respectively, in this paper.

Table 5: Running time (in seconds) for solving SGL along a sequence of 100 tuning parameter values of λ equally spaced on the logarithmic scale of $\lambda/\lambda^{\alpha}_{\max}$ from 1.0 to 0.01 by (a): APCG (Lin et al., 2014); (b): sgLeastR Liu et al. (2009).

	α	tan(5°)	tan(15°)	tan(30°)	tan(45°)	tan(60°)	tan(75°)	tan(85°)
Synthetic 1	APCG	123088.60	126410.46	126301.35	126536.18	127348.69	127533.82	127419.43
	sgLeastR	15555.28	16124.08	16106.24	16293.04	16426.44	16836.16	16862.36
E2006-tfidf	APCG	116522.32	116546.41	115324.36	115982.47	119432.35	115433.98	115453.62
	sgLeastR	1303.25	1438.62	1458.32	1498.63	1506.35	1462.98	1408.38

Table 6: Running time (in seconds) for solving nonnegative Lasso along a sequence of 100 tuning parameter values of λ equally spaced on the logarithmic scale of $\lambda/\lambda_{\max}$ from 1.0 to 0.01 by (a): APCG (Lin et al., 2014); (b): nnLeastR Liu et al. (2009).

	Synthetic 1	E2006-tfidf
APCG	92405 .15	106446.16
nnLeastR	13140.84	634.66

References

S. Armstrong, J. Staunton, L. Silverman, R. Pieters, M. den Boer, M. Minden, S. Sallan, E. Lander, T. Golub, and S. Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30:41–47, 2002.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization, Second Edition*. Canadian Mathematical Society, 2006.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. In *ICDM*, 2007.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *Pacific Journal of Optimization*, 8:667–698, 2012.

O Fercoq and P Richtrik. Accelerated, parallel and proximal coordinate descent. *Journal on Optimization, 2013*, 25(4), 2013.

J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. arXiv:1001.0736.

N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53:217–288, 2011.

J.-B. Hiriart-Urruty. From convex optimization to nonconvex optimization. necessary and sufficient conditions for global optimality. In *Nonsmooth optimization and related topics*. Springer, 1988.

J.-B. Hiriart-Urruty. A note on the Legendre-Fenchel transform of convex composite functions. In *Nonsmooth Mechanics and Analysis*. Springer, 2006.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):52–58, 2014.

J. Liu and J. Ye. Moreau-Yosida regularization for grouped tree structure learning. In *Advances in neural information processing systems 23*, 2010.

J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

J. Liu, Z. Zhao, J. Wang, and J. Ye. Safe screening with variational inequalities and its application to lasso. In *International Conference on Machine Learning*, 2014.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in nature images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2001.

K. Ogawa, Y. Suzuki, and I. Takeuchi. Safe screening of non-support vectors in pathwise SVM computation. In *International Conference on Machine Learning*, 2013.

K. Ogawa, Y. Suzuki, S. Suzumura, and I. Takeuchi. Safe sample screening for Support Vector Machine. *arXiv:1401.6740*, 2014.

J. Peng, J. Zhu, A. Bergamaschi, W. Han, D. Noh, J. Pollack, and P. Wang. Regularized multivariate regression for indentifying master predictors with application to integrative genomics study of breast cancer. *The Annals of Appliced Statistics*, 4:53–77, 2010.

E. Petricoin, D. Ornstein, C. Paweletz, A. Ardekani, P. Hackett, B. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C. Simone, P. Levine, W. Linehan, M. Emmert-Buck, S. Steinberg, E. Kohn, and L. Liotta. Serum proteomic patterns for detection of prostate cancer. *Journal of National Cancer Institute*, 94:1576–1578, 2002.

A. Ruszczyński. *Nonlinear Optimization*. Princeton University Press, 2006.

S. Shevade and S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19:2246–2253, 2003.

T. Sim, B. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1615–1618, 2003.

N. Simon, J. Friedman., T. Hastie., and R. Tibshirani. A Sparse-Group Lasso. *Journal of Computational and Graphical Statistics*, 22:231–245, 2013.

P. Sprechmann, I. Ramírez, G. Sapiro., and Y. Eldar. C-HiLasso: a collaborative hierarchical sparse modeling framework. *IEEE Transactions on Signal Processing*, 59:4183–4198, 2011.

R. Tibshirani. Regression shringkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288, 1996.

R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society Series B*, 74:245–266, 2012.

Robert Tibshirani, Martin Wainwright, and Trevor Hastie. *Statistical learning with sparsity: the Lasso and generalizations*. Chapman and Hall/CRC, 2015.

M. Vidyasagar. Machine learning methods in the cocomputation biology of cancer. In *Proceedings of the Royal Society A*, 2014.

M. Vincent and N. Hansen. Sparse group lasso and high dimensional multinomial classification. *Computational Statistics and Data Analysis*, 71:771–786, 2014.

J. Wang, J. Jun, and J. Ye. Efficient mixed-norm regularization: Algorithms and safe screening methods. *arXiv:1307.4156v1*.

J. Wang, J. Zhou, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. In *Advances in neural information processing systems 26*, 2013.

J. Wang, P. Wonka, and J. Ye. Scaling svm and least absolute deviations via exact data reduction. In *International Conference on Machine Learning*, 2014.

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. Marks, and J. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences*, 98:11462–11467, 2001.

Z. J. Xiang and P. J. Ramadge. Fast lasso screening tests based on correlations. In *IEEE ICASSP*, 2012.

D. Yogatama and N. Smith. Linguistic structured sparsity in text categorization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 2006.

Y.J. Zhang, N. Zhang, D.F. Sun, and K.C. Toh. An efficient hessian based algorithm for solving large-scale sparse group lasso problems. *Mathematical Programming*, 2018a.

Y.J. Zhang, N. Zhang, D.F. Sun, and K.C. Toh. An efficient hessian based algorithm for solving large-scale sparse group lasso problems. *Mathematical Programming*, 2018b.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67:301–320, 2005.

Journal of Machine Learning Research 20 (2019) 1-31 Submitted 10/17; Revised 9/19; Published 12/19

The Reduced PC-Algorithm: Improved Causal Structure Learning in Large Random Networks

Arjun Sondhi[*] ASONDHI@UW.EDU

Ali Shojaie ASHOJAIE@UW.EDU

Department of Biostatistics
University of Washington
Seattle, WA 98195-4322, USA

Editor: Edo Airoldi

Abstract

We consider the task of estimating a high-dimensional directed acyclic graph, given observations from a linear structural equation model with arbitrary noise distribution. By exploiting properties of common random graphs, we develop a new algorithm that requires conditioning only on small sets of variables. The proposed algorithm, which is essentially a modified version of the PC-Algorithm, offers significant gains in both computational complexity and estimation accuracy. In particular, it results in more efficient and accurate estimation in large networks containing hub nodes, which are common in biological systems. We prove the consistency of the proposed algorithm, and show that it also requires a less stringent faithfulness assumption than the PC-Algorithm. Simulations in low and high-dimensional settings are used to illustrate these findings. An application to gene expression data suggests that the proposed algorithm can identify a greater number of clinically relevant genes than current methods.

Keywords: causal discovery, directed acyclic graphs, faithfulness, high dimensions, random graphs

1. Introduction

Directed acyclic graphs, or DAGs, are commonly used to represent causal relationships in complex biological systems. For example, in gene regulatory networks, directed edges represent regulatory interactions among genes, which are represented as nodes of the graph. While causal effects in biological networks can be accurately inferred from perturbation experiments (Shojaie et al., 2014)—including single or double gene knockouts (Schadt et al., 2005; Yoo et al., 2002)—these are costly to run. Estimating DAGs from observational data is thus an important exploratory task for generating causal hypotheses (Friedman et al., 2000; Jansen et al., 2003), and designing more efficient experiments.

Since the number of possible directed graphs grows super-exponentially in the number of nodes, estimation of DAGs is an NP-hard problem (Chickering, 1996). Methods of estimating DAGs from observational data can be broadly categorized into three classes. The first class, score-based methods, search over the space of all possible graphs, and attempt to maximize a goodness-of-fit score, generally using a greedy algorithm. Examples include the hill climbing and tabu search algorithms (Scutari, 2010), as well as Bayesian approaches (Eaton and Murphy, 2012). The second class, constraint-based methods, first estimate the graph skeleton by performing conditional independence tests; the skeleton of a directed acyclic graph is the undirected graph obtained by removing the direction of edges. Information from conditional independence relations is then used to partially orient the edges of the graph. The resulting completed partially directed acyclic graph (CPDAG) represents the class of all directed acyclic graphs that are Markov equivalent, and therefore not distinguishable from observational data. The most well-known constraint-based method is the PC-Algorithm (Spirtes et al., 2000), which was popularized by Kalisch and Bühlmann (2007) for high-dimensional estimation. Finally, hybrid methods combine score and constraint-based approaches. For example, the Max-Min Hill Climbing algorithm (Tsamardinos et al., 2006) estimates the skeleton using a constraint-based method, and then orients the edges by using a greedy search algorithm. Sparsity-inducing regularization approaches have also been used to develop efficient hybrid methods (Schmidt et al., 2007).

Estimating DAGs in high dimensions introduces new computational and statistical challenges. Until recently, graph recovery in high dimensions was only established for the PC-Algorithm (Kalisch and Bühlmann, 2007) and hybrid constraint-based methods (Ha et al., 2016). While the recent work of Nandy et al. (2015) extends these results to score-based algorithms and their hybrid extensions, the PC-Algorithm is still considered a gold standard in high-dimensional sparse settings, due to its polynomial time complexity (Kalisch and Bühlmann, 2007). Moreover, constraint-based methods are indeed the building blocks of various hybrid approaches. Therefore, we primarily focus on constraint-based methods in this paper.

Despite its appealing features, the PC-Algorithm entails several properties that do not scale well to common high-dimensional settings. Specifically, large real-world biological systems are known to commonly be sparse graphs containing a small number of highly connected *hub* nodes (Chen and Sharp, 2004; Jeong et al., 2000). In such graphs, the average node degree will be small, while the maximum tends to be much larger, and increases with the number of nodes. This is particularly problematic for the PC-Algorithm, whose computational and sample complexities scale with the *maximum node degree* in the graph. Moreover, the recent work by Uhler et al. (2013) shows that the distributional assumptions required for high-dimensional consistency of the PC-Algorithm are overly restrictive in practice, and that the class of graphs which do not satisfy these assumptions is large. Although work has been done by Peters et al. (2014) on estimating DAGs defined over a larger class of probability models, the resulting methods also do not scale to high dimensions.

A common limitation of existing methods for estimating DAGs is that they do not account for structural properties of large networks. For instance, the PC-Algorithm only incorporates the sparsity of the network, by assuming that the maximum node degree in the graph skeleton is small relative to the sample size. However, real-world networks, particularly those observed in biology, are known to posses a number of other important properties. Of particular interest in estimating DAGs is the so-called *local separation property* of large networks (Anandkumar et al., 2012), which implies that the number of short paths between any two nodes is bounded. This property is observed in many large sparse networks, including polytrees, Erdős-Rényi, power law, and small world graphs (Durrett, 2007). Power

∗. Arjun Sondhi is currently a Quantitative Scientist at Flatiron Health.

©2019 Arjun Sondhi and Ali Shojaie.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/17-601.html.

law graphs are of particular interest in many biological applications, as they allow for the presence of hub nodes.

In this paper, we propose a low-complexity constraint-based method for estimating high-dimensional sparse DAGs. The new method, termed *reduced PC* (rPC), exploits the local separation property of large random networks, which was used by Anandkumar et al. (2012) in estimation of undirected graphical models. We show that rPC can consistently estimate the skeleton of high-dimensional DAGs by conditioning only on sets of small cardinality. This is in contrast to previous heuristic DAG learning approaches that set an upper bound on the number of parents for each node (Acid and de Campos, 1994; Huete and Campos, 1993), which is an assumption that cannot be justified in many real-world networks. We also show that computational and sample complexities of rPC only depend on average sparsity of the graph—a notion that is made more precise in Sections 3 and 4. This leads to considerable advantages over the PC-Algorithm, whose computational and sample complexities scale with the maximal node degree. Moreover, these properties hold for linear structural equation models (Shojaie and Michailidis, 2010) with arbitrary noise distributions, and require a weaker faithfulness conditions on the underlying probability distributions than the PC-Algorithm. We present two version of the rPC algorithm: a "full" version for which we provide theoretical guarantees, and an approximate version which is much faster and performs almost identically in practice.

The rest of the paper is organized as follows. In Section 2 we review basic properties of graphical models over DAGs, and give a short overview of the PC-Algorithm. Our new algorithm is presented in Section 3 and its properties, including consistency in high dimensions are discussed in Section 4. Results of simulation studies and a real data example concerning the estimation of gene regulatory networks are presented in Sections 5 and 6, respectively. We end with a brief discussion in Section 7. Technical proofs and additional simulations are presented in the appendices.

2. Preliminaries

In this section, we review relevant properties of graphical models defined over DAGs, and briefly describe the theory and implementation of the PC-Algorithm.

2.1. Background

For p random variables $X_1, \ldots, X_p$, we define a graph $G = (V, E)$ with vertices, or nodes, $V = \{1, \ldots, p\}$ such that variable X_j corresponds to node j. The edge set $E \subset V \times V$ contains directed edges; that is, $(j, k) \in E$ implies $(k, j) \notin E$. Furthermore, there are no directed cycles in G. We denote an edge from j to k as $j \to k$ and call j a parent of k, and k a child of j. The set of parents of node k is denoted $pa(k)$, while the set of nodes adjacent to it, or all of k's parents and children, is denoted $adj(k)$. These notations are also used for the corresponding random variable X_k. We assume there are no hidden common parents of node pairs (that is, no unmeasured confounders). The degree of node k is defined as the number of nodes which are adjacent to it, $|adj(k)|$; we denote the maximal degree in the graph as d_{max}. A triplet of nodes (i, j, k) is called an *unshielded triple* if i and j are adjacent to k but i and j are not adjacent. An unshielded triple is called a *v-structure* if $i \to j \leftarrow k$.

We assume random variables follow a linear structural equation model (SEM),

$$X_k = \sum_{j \in pa(k)} \rho_{jk} X_j + \epsilon_k, \tag{1}$$

where for $k = 1, \ldots, p$, ϵ_k are independent random variables with finite variance, and ρ_{jk} are fixed unknown constants. The directed Markov property, stated below, is usually assumed in order to connect the joint probability distribution of $X_1, \ldots, X_p$ to the structure of the graph G.

Definition 1 *A probability distribution is Markov on a DAG $G = (V, E)$ if, conditional on its parents, every random variable X_k is independent of its non-descendants; that is, $X_k \perp\!\!\!\perp X_j \mid pa(X_k)$ for all $j \in V$ which are non-descendants of k.*

Although this assumption allows us to connect conditional independence relationships to the DAG structure, there are generally multiple graphs that generate the same distribution under the Markov property. More concretely, DAGs are Markov equivalent if they have the same skeleton and the same set of v-structures. Therefore, constraint-based methods focus primarily on estimating the skeleton of the DAGs from observational data. Conditional independence relations identified when learning the skeleton are then used to orient some of the edges to obtain the CPDAG, which represents the Markov equivalence class of directed graphs (Kalisch and Bühlmann, 2007).

We next define d-separation, a graphical property which is used to read conditional independence relationships from the DAG structure.

Definition 2 *In a DAG G, two nodes k_1 and k_2 are d-separated by a set S if and only if, for all paths π between k_1 and k_2:*
(i) π contains a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle node m is in S, or
(ii) π contains an inverted fork (or collider) $i \to m \leftarrow j$ such that the middle node m is not in S and no descendant of m is in S.

A path π which does not satisfy the requirements of this definition is known as a *d-connecting path*. Using observed data, d-separations in a graph G can be identified based on conditional independence relationships. To this end, we require the following assumption, known as *faithfulness*, on the probability distribution of random variable on G.

Definition 3 *A probability distribution is faithful to a DAG G if $X_i \perp\!\!\!\perp X_j \mid X_S$ whenever i and j are d-separated by S.*

2.2. The PC-Algorithm

Together, d-separation and faithfulness suggest a simple algorithm for recovering the DAG skeleton. If we discover that $X_i \perp\!\!\!\perp X_j \mid S$ for some set S, then there cannot be an edge $(i, j) \in E$. Conversely, if we discover $X_i \not\perp\!\!\!\perp X_j \mid S$ for all possible sets S, then there must be an edge $(i, j) \in E$. Therefore, under faithfulness, an obvious strategy for skeleton estimation would be to test all possible conditional independence relations for each pair of variables; that is, test whether $X_i \perp\!\!\!\perp X_j \mid S$ for any $S \subset V \setminus \{i, j\}$. While this strategy is

computationally infeasible for large p, and statistically problematic when $p > n$, it forms the basis of the PC-Algorithm. The PC-Algorithm starts with a complete undirected graph and deletes edges (i, j) if a set S can be found such that $X_i \perp\!\!\!\perp X_j \mid S$. The algorithm also uses the fact that if such an S exists, then there exists a set S' such that all nodes in S' are adjacent to i or j and $X_i \perp\!\!\!\perp X_j \mid S'$. Thus, at each step of the algorithm, only local neighbourhoods need to be examined in order to find the separating sets.

Although consistent for sufficiently sparse high-dimensional DAGs, the PC-Algorithm's computational and sample complexity scale with the maximal degree of the graph, d_{max}. Specifically, the algorithm's computational complexity is $O(p^{d_{max}})$ and its sample complexity is $\Omega\{\max(\log p, d_{max}^{1/b})\}$ for some $b \in (0, 1]$. This is problematic for graphs with highly connected hub nodes, which are common in real-world networks (Chen and Sharp, 2004; Jeong et al., 2000). In such graphs, d_{max} typically grows with the number of nodes p, leading to poor accuracy and runtime for the PC-Algorithm.

Another limitation of the PC-Algorithm is that it requires partial correlations between adjacent nodes to be bounded away from 0; this requirement, which needs to hold for all conditioning sets S such that $|S| \leq d_{max}$, is known as *restricted strong faithfulness* (Uhler et al., 2013), and is defined next.

Definition 4 *Given $\lambda \in (0, 1)$, a distribution P is said to be restricted λ-strong-faithful to a DAG $G = (V, E)$ if the following conditions are satisfied:*
(i) $\min\{|\rho(X_i, X_j \mid X_S)| : (i, j) \in E, S \subset V \setminus \{i, j\}, |S| \leq d_{max}\} > \lambda$, *and*
(ii) $\min\{|\rho(X_i, X_j \mid X_S)| : (i, j, S) \in N_G\} > \lambda$, *where N_G is the set of triples (i, j, S) such that i, j are not adjacent, but there exists $k \in V$ making (i, j, k) an unshielded triple, and i, j are not d-separated given S.*

Kalisch and Bühlmann (2007) assume that $\lambda = \Omega(n^{-w})$ for $w \in (0, b/2)$ where $b \in (0, 1]$ relates to the scaling of d_{max}. In the low-dimensional setting, it has been shown that the PC-Algorithm achieves uniform consistency with λ converging to zero at rate $n^{1/2}$, which then gives the same condition as ordinary faithfulness (Zhang and Spirtes, 2002). However, in the high-dimensional setting, the set of distributions which are not restricted strong faithful has nonzero measure. In fact, Uhler et al. (2013) showed that this assumption is overly restrictive and that the measure of unfaithful distributions converges to 1 exponentially in p. We will revisit the faithfulness assumption in Section 4.2.

To address the limitations of the PC-Algorithm, we next propose a new algorithm that takes advantage of the structure of large networks from common random graph families. By doing so, we obtain improved computational and sample complexity; as we will show, these complexities are unaffected by the increase in the maximal degree as the graph becomes larger. We also prove consistency under a weaker faithfulness assumption than that needed for the PC-Algorithm.

3. The Reduced PC-Algorithm

As with the PC-Algorithm, our strategy for estimating the graph skeleton is to start with a complete graph, and then delete edges by discovering separating sets. Under a faithfulness assumption on a linear structural equation model, we do so by computing partial correlations

and declaring X_i and X_j d-separated by S if $\rho(X_i, X_j \mid S)$ is smaller than some threshold α. Aside from thresholding, the key difference between our proposal and the PC-Algorithm is that we only consider partial correlations conditional on sets S with small cardinality.

We justify our method using two key observations. Our first key observation is based on the decomposition of covariances over *treks*, which are special types of paths in directed graphs.

Definition 5 *A trek between two nodes i and j in a DAG G is either a path from i to j, a path from j to i, or a pair of paths from a third node k to i and j such that the two paths only have k in common.*

In a linear SEM (1), the covariance between two random variables is characterized by the treks between them. Denoting a trek between nodes i and j as $\pi : i \leftrightarrow j$ with common node or source k, the covariance is given by

$$\mathrm{cov}(X_i, X_j) = \sum_{\pi : i \leftrightarrow j} \sigma_k \prod_{e \in \pi} \rho_e, \tag{2}$$

where σ_k is the variance of ϵ_k from Equation (1) and ρ_e denotes the weight of an edge along the trek, which is the corresponding coefficient in the SEM. This is shown in full detail by Sullivant et al. (2010), who also show that the covariance conditional on a d-separating set S leaves out treks which include any nodes in S. This conditioning effect is illustrated in Figure 1, where conditioning on an appropriate set S blocks the treks between non-adjacent nodes, without resulting in any additional d-connecting paths. If all edge weights are bounded by 1 in absolute value, then the contribution of each trek to the covariance decays exponentially in trek length. In practice, we scale the data matrix X so that each column has unit standard deviation. Under some conditions, most of the edge weights in the linear SEM (1) then satisfy $|\rho_{ij}| < 1$ (this is discussed and shown empirically in Appendix B). Hence, the contribution of long treks to the conditional covariance among non-adjacent nodes is negligible. This decay motivates the thresholding of partial correlations in rPC, and is further discussed in Section 4.

The above observation suggests a new strategy for learning DAG structures by only considering short treks. Suppose S is the set that blocks all short treks between two nodes j and k. If the correlation over all remaining d-connecting paths (after conditioning on S) between j and k is negligible, then the partial correlation given S, $\rho(X_j, X_k \mid S)$ can be used to determine whether j and k are adjacent.

To determine the size of the conditioning set, S, we need to determine the number of short treks between any two nodes j and k. Our second key observation addresses this question, by utilizing properties of large random graphs. More specifically, motivated by Anandkumar et al.'s proposal for estimating undirected graphs, we consider a key feature of large random networks, known as the the *local separation property*.

Definition 6 *Given a graph G, a γ-local separator $S_\gamma(i, j) \subset V$ between non-neighbours i and j minimally separates i and j with respect to paths of length at most γ.*

Definition 7 *A family of graphs $\mathcal{G}$ satisfies the (η, γ)-local separation property if, as $p \to \infty$, $Pr(\exists G \in \mathcal{G} : \exists (i, j) \notin E_G, |S_\gamma(i, j)| > \eta) \to 0$.*

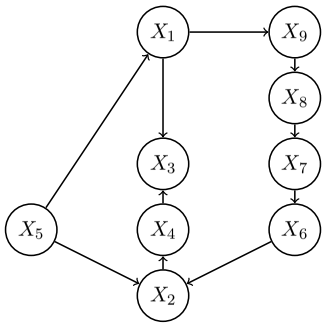


Figure 1: Illustration of treks between nodes X_1 and X_2 within a DAG. The middle path involves a collider, X_3, so does not contribute to $\text{cov}(X_1, X_2)$, which is $\rho_{51}\rho_{52} + \rho_{19}\rho_{98}\rho_{87}\rho_{76}\rho_{62}$ The conditional covariance $\text{cov}(X_1, X_2 \mid X_5)$ excludes treks that involve X_5, and is thus $\rho_{19}\rho_{98}\rho_{87}\rho_{76}\rho_{62}$. Here, $S = \{X_5, X_7\}$ is a d-separating set, as it blocks both treks, giving $\text{cov}(X_1, X_2 \mid S) = 0$.

Intuitively, under (η, γ)-local separation, with high probability, the number of short treks—of length at most γ—between any two non-neighbouring nodes is bounded above by η. In fact, as the local separation property refers to any type of path, there are likely even fewer than η short treks between any two neighbouring nodes. Therefore, we only need to consider conditioning on sets S of size at most η in order to remove the correlation induced by short treks. Combining this with our first insight, we ignore treks (and other possible d-connecting paths) of length longer than γ, which, for appropriate probability distributions, have a negligible impact on partial correlations. The resulting procedure, called the full reduced PC-Algorithm (rPC-full), is presented in Algorithm 1.

We note a key implementation difference between rPC-full and the ordinary PC-Algorithm: when searching for a separating set, rPC-full considers all $S \subset V \setminus \{i, j\}$, while PC-Algorithm only searches over the local neighbourhoods of i and j. Recall that if a set S d-separates nodes i and j, then there exists a d-separating set S' such that all nodes in S' are adjacent to i or j. However, $|S'|$ may be larger than $|S|$; because rPC only considers sets of size up to η, a full search is needed to ensure discovery of a d-separating set. Motivated by the PC-Algorithm, we also propose an approximate reduced PC-Algorithm (rPC-approx), which uses the same local neighbourhood search, i.e. $S \subset adj(i) \cup adj(j) \setminus \{i, j\}$. In practice, we show that rPC-approx performs almost identically to rPC-full.

To recap, our proposal in Algorithm 1 hinges on two important properties of probability models on large DAGs: (P1) boundedness of the number of short treks between any two nodes, and (P2) negligibility of correlation over the remaining (long) d-connecting paths. The first property, which is characterized by local separation, concerns solely the DAG structure. Anandkumar et al. (2012) show that many common graph families satisfy the local separation property with small η. Specifically, sparse, large binary trees, Erdős-Rényi

graphs, and graphs with power law degree distributions all satisfy this property with $\eta \leq 2$. Moreover, the sparsity requirement for these graph families is in terms of average node degree, and not the maximum node degree. For these graph families, our algorithm only needs to consider separating sets of size 0, 1, and 2, irrespective of the maximum node degree. Small-world graphs, as generated by the Watts-Strogatz algorithm (Kleinberg, 2000), also satisfy this property, but with $\eta > 2$. In addition, the γ parameter increases with p for these families; thus, as graphs get larger, the local separation property applies to a larger set of paths.

By only considering a bounded number of short paths, our algorithm has computational complexity $O(p^{\eta+2})$, and thus avoids the exponential scaling in d_{max} that the PC-Algorithm suffers from. This is particularly significant in the case of power-law graphs, where $d_{max} = O(p^a)$ for $a > 0$ (Molloy and Reed, 1995); in this case, PC-Algorithm has a computational complexity of $O(p^{p^a})$, which is significantly worse than rPC's complexity of $O(p^4)$. While rPC-full might not be faster in practice due to the larger search space, rPC-approx would result in significant speedup.

Unlike the first property (P1), the second property needed for our algorithm, namely the negligibility of correlation over d-connecting paths, concerns both the structure of the DAG G, and the probability distribution P of variables on the graph. In the next section, we discuss two alternative sufficient conditions that guarantee this property, and allow us to consistently estimate the DAG skeleton.

INPUT: Observations from random variables $X_1, X_2, \ldots, X_p$; threshold level α;
maximum separating set size η.
OUTPUT: Estimated skeleton C.

Set $V = \{1, \ldots, p\}$.
Form the complete undirected graph $\tilde{C}$ on the vertex set V.
Set $l = -1$; $C = \tilde{C}$.
repeat
  $l = l + 1$
  repeat
    Select a (new) ordered pair of nodes i, j that are adjacent in C
    repeat
      Choose (new) $S \subset V \setminus \{i, j\}$ with $|S| = l$
      if $\rho(X_i, X_j \mid S) \leq \alpha$
        Delete edge (i, j)
        Denote this new graph by C
      end if
    until edge (i, j) deleted or all $S \subset V \setminus \{i, j\}$ with $|S| = l$ have been chosen
  until all ordered pairs of adjacent nodes i and j have been examined for $\rho(X_i, X_j \mid S) \leq \alpha$
until $l > \eta$

Algorithm 1: The full reduced PC-Algorithm (rPC-full)

4. Algorithm Analysis and Asymptotics

In this section, we describe in detail the assumptions required for the rPC-full algorithm to consistently recover the DAG skeleton. We also discuss its computational and statistical properties, particularly in comparison with the PC-Algorithm.

4.1. Consistency

As discussed in the previous section, to consistently recover the DAG skeleton, rPC requires that properties (P1) and (P2) hold; namely, that the graph under consideration has a bounded number of short paths and the correlation over d-connecting paths of length greater than γ decays sufficiently quickly. In fact, the trek decomposition (2) indicates that the correlation for each long trek is small when most of the edge weights are bounded by 1 in absolute value. However, condition (P2) requires the total correlation over all long treks (and other d-connecting paths) to decay sufficiently quickly. To this end, we consider two alternative sufficient conditions. The first condition is a direct assumption on the boundedness of the conditional correlation. The second is inspired by Anandkumar et al. (2012), and assumes the underlying probability model satisfies what we term *directed walk-summability*. This condition mirrors the walk-summability condition for undirected graphical models, which has been well-studied and shown to hold in a large class of models (Malioutov et al., 2006).

Definition 8 *A probability model is directed β-walk-summable on a DAG with weighted adjacency matrix A, if $\|A\| \leq \beta < 1$ where $\|\cdot\|$ denotes the spectral norm.*

As an alternative to this condition, we also present a direct assumption, Assumption 4. This condition is less restrictive than directed walk-summability, but has not been characterized in the literature. However, given that γ increases with p in the graph families we are considering, it is intuitive that the sum of edge weight products over treks longer than length γ will be decreasing and asymptotically small. This also holds for non-trek d-connecting paths. These two assumptions lead to two parallel proofs of the consistency of our algorithm, presented in Theorem 1. Before stating the theorem, we discuss our assumptions.

Similar to the PC-Algorithm, our method requires a faithfulness condition. As stated previously, our condition, which we term *path faithfulness* and is defined next, is weaker than PC-Algorithm's λ-strong faithfulness stated in Definition 4 (see Section 4.2 for additional details).

Definition 9 *Given $\lambda \in (0,1)$, a distribution P is said to be λ-path-faithful to a DAG $G = (V, E)$ if both of the following conditions hold:*
(i) $\min\{|\rho(X_i, X_j \mid X_S)| : (i,j) \in E, S \subset V \setminus \{i,j\}, |S| \leq \eta\} > \lambda$, *for some η, and*
(ii) $\min\{|\rho(X_i, X_j \mid X_S)| : (i,j,S) \in N_G\} > \lambda$, *where N_G is the set of triples (i,j,S) such that i,j are not adjacent, but there exists $k \in V$ making (i,j,k) an unshielded triple, and i,j are not d-separated given S.*

Part (i) of the assumption only requires partial correlations between true edges conditioned on sets of size up to η to be bounded away from zero, while the PC-Algorithm requires this

for conditioning sets of size up to d_{max}. In Section 4.2, we discuss how this affects bounds on the true partial correlations, and also empirically show that the above path faithfulness assumption is less restrictive than corresponding assumption for the PC-Algorithm.

Assumption 1 (Path faithfulness and Markov property) *The probability distribution P of random variables corresponds to a linear SEM (1) with sub-Gaussian errors, and is λ-path-faithful to the DAG G, with $\lambda = \Omega(n^{-c})$ for $c \in (0, 1/2)$.*

Our second assumption ensures that the covariance matrix of the structural equation model and its inverse remain bounded as p grows.

Assumption 2 (Covariance and precision matrix boundedness) *The covariance matrix of the model Σ_G and its inverse Σ_G^{-1} are bounded in spectral norm, that is, $\max(\|\Sigma_G\|, \|\Sigma_G^{-1}\|) \leq M < \infty$ for all p.*

The last three assumptions characterize applicable graph families and probability distributions.

Assumption 3 ((η,γ)-local separation) *The DAG G belongs to a family of random graphs $\mathcal{G}$ that satisfies the (η,γ)-local separation property with $\eta = O(1)$ and $\gamma = O(\log p)$.*

Assumption 4 (Bounded long path weight) *Let π denote a d-connecting path of length $l(\pi)$ between two non-adjacent vertices i and j in G. Then, there exists a conditioning set S such that the total edge weight over d-connecting paths longer than γ satisfies:*

$$\sum_{l=\gamma+1}^{p-1} \sum_{l(\pi)=l} |\rho_{\pi,1} \ldots \rho_{\pi,l}| = O(\beta^{\gamma}),$$

for some $\beta \in (0,1)$.

Assumption 4 guarantees that the sum of weights over long treks between any two nodes i and j is bounded. For a single trek, a sufficient condition is that all edge weights are bounded by 1 in magnitude. In Appendix B, we provide further discussion on Assumption 4, and empirically investigate its plausibility. In particular, we show that if the data matrix X is scaled so that each column has unit standard deviation, then with high probability all edge weights are bounded by 1 in absolute value. To account for residual correlation induced through conditioning on common descendants of i and j, Assumption 4 also includes non-trek d-connecting paths.

Assumption 4′ (Directed β-walk-summability) *The probability distribution P is directed β-walk-summable.*

We are now ready to state our main result. The result is proved in Appendix A, where the error probabilities are also analyzed.

Theorem 1 *Under Assumptions 1-3 and either Assumption 4 or 4′, there exists a parameter α for thresholding partial correlations such that, as $n, p \longrightarrow \infty$ with $n = \Omega\{(\log p)^{1/(1-2c)}\}$, the full reduced PC (rPC-full) procedure, as described in Algorithm 1, consistently learns the skeleton of the DAG G.*

Several theoretical features of our algorithm are attractive. As stated previously, our faithfulness condition is weaker than the corresponding assumption for the PC-Algorithm and related methods. Similar to its computational complexity, the sample complexity of our algorithm also does not scale with the maximal node degree, and is only dependent on the parameter η as p increases. For example, in a power law graph, the sample complexity of the PC-Algorithm is $\Omega\{\max(\log p, p^{ab})\}$ for $0 < a, b < 1$, compared to $\Omega\{(\log p)^{1/1-2c}\}$ with $c \in (0, 1/2)$ for our method. This gain in efficiency is due to fact that the maximum separating set size, η, remains constant in rPC. Finally, our algorithm does not require the data to be jointly Gaussian. The proof of the algorithm's consistency only requires that the population covariance matrix can be well-approximated from the data; for simplicity, we assume a sub-Gaussian distribution.

4.2. On Faithfulness Assumption

As stated in Section 2.2, for large biological networks of interest, the maximum node degree, d_{max}, often grows with p. Therefore, the λ-restricted strong faithfulness condition of the PC-Algorithm—Definition 4—becomes exponentially harder to satisfy with increasing network size. A full discussion of this phenomenon can be found in Uhler et al. (2013), where it is shown that the measure of strong unfaithful distributions converges to 1 for various graph structures. Although this would also occur with path faithfulness (Definition 9), our condition allows a rate for λ that is independent of d_{max} and p.

The rate for λ in the PC-Algorithm is $\lambda = \Omega(n^{-w})$ for $w \in (0, b/2)$, where $d_{max} = O(n^{1-b})$ for $b \in (0, 1]$. For $b = 1$, or constant d_{max}, the PC-Algorithm's required scaling for λ is identical to that for our method in Assumption 1. This makes sense intuitively, since our method is not affected by the increase in d_{max}. For other values of b, the scaling of λ becomes more restricted for the PC-Algorithm; for example, if $b = 1/2$, then $\lambda = \Omega(n^{-w})$ for $w \in (0, 1/4)$. However, under path faithfulness (Definition 9), we can still achieve a rate of $\lambda = \Omega(n^{-1/2})$; that is, the partial correlations are allowed to be smaller and the condition is weaker.

We report the findings of a simulation study, similar to that in Uhler et al. (2013), which examines how often randomly generated DAGs satisfy part (i) of the path faithfulness assumption compared to restricted strong faithfulness. We are primarily interested in part (i), as this part is needed for consistent skeleton estimation; part (ii), on the other hand, is needed to obtain correct separating sets in order to obtain partial orientation of edges. In this simulation, 1000 random DAGs were generated from Erdős-Rényi and power law families, with edge weights drawn independently from a Uniform$(-1, 1)$ distribution. Each DAG had $p = 20$ nodes, with varying expected degrees per node. For each simulation setting, we computed the proportion of DAGs that satisfied part (i) of the λ-restricted-strong-faithfulness and λ-path-faithfulness conditions with $\lambda = 0.001$ and $\eta = 2$. The results are shown in Table 1. We see that path faithfulness is much more likely to be satisfied than restricted strong faithfulness, especially for power law graphs. This is to be expected, as the number of constraints required for restricted strong faithfulness grows with d_{max}, but remains constant for path faithfulness. It is, however, difficult for dense graphs to satisfy either condition, although there is a mild advantage for path faithfulness. In Appendix B, we provide the results of further simulation studies with $p = 10$ and $p = 30$;

Graph family	Expected degree	Pr(RSF)	Pr(PF)
Erdős-Rényi	2	0.77	0.92
Erdős-Rényi	5	0	0.08
Power law	2	0.54	0.85
Power law	6	0.003	0.08

Table 1: Empirical probabilities of random DAGs of size $p = 20$ satisfying faithfulness conditions; RSF refers to restricted strong faithfulness of the PC-Algorithm, and PF refers to path faithfulness of reduced PC (rPC).

both of these give similar conclusions and indicate that path faithfulness remains easier to satisfy with increasing network size.

4.3. Tuning Parameter Selection

Our algorithm requires two tuning parameters: the maximum separating set size η, and the threshold level for partial correlations α. The parameter η varies based on the underlying graph family. Thus, given knowledge of a plausible graph structure, η can be pre-specified. Alternatively, η can be selected by maximizing a goodness-of-fit score over a parameter grid, along with α. This may be preferable as the local separation results consider all short paths, not just treks, so better performance may be obtained by specifying a smaller η. Likewise, when using the rPC-approx algorithm, a larger η could be needed to discover appropriate separating sets.

For jointly Gaussian data, we can obtain a modified version of the Bayesian information criterion by fitting the likelihood to the CPDAG obtained based on the estimated DAG skeleton (Foygel and Drton, 2010). Following Anandkumar et al. (2012), and denoting by X_{obs} the observed data, we use:

$$\text{BIC}(X_{obs}; \hat{G}) = \log f(X_{obs}; \hat{\theta}) - 0.5|E| \log(n) - 2|E| \log(p), \tag{3}$$

where $\hat{G}$ denotes one of the DAGs obtained from the estimated CPDAG containing $|E|$ edges. The CPDAG represents the Markov equivalence class of DAGs, so all possible graphs will result in the same fitted Gaussian model with parameters $\hat{\theta}$. We use this BIC for tuning parameter selection; higher scores imply a better fit. For linear SEMs with non-Gaussian noise distributions, the Gaussian likelihood serves as a surrogate goodness-of-fit measure, and BIC can still be used to select the tuning parameters.

5. Simulation Studies

In this section, we compare the performance of rPC-approx, rPC-full, and the standard PC-Algorithm in multiple simulation settings. We consider both setting a constant value for η in rPC, and tuning it to maximize the BIC.

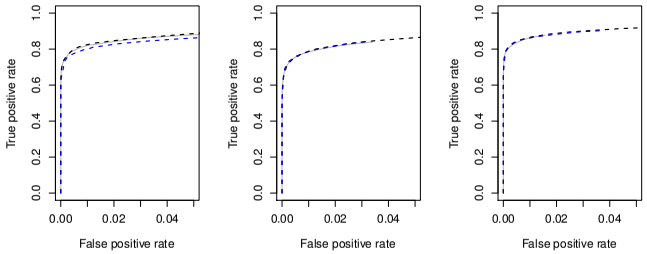


Figure 2: Average true vs. false positive rates for PC-Algorithm (grey solid line), rPC-approx (black dashed line), and rPC-full (blue dashed line) estimating Erdős-Rényi DAGs. Left: $p = 100, n = 200$; centre: $p = 200, n = 100$; right: $p = 500, n = 200$.

5.1. Pre-specified η Parameter

To facilitate the comparison with the PC-Algorithm, we generate data from Gaussian linear SEMs as in Equation 1, with the dependency structure specified by a DAG from Erdős-Rényi and power law families. We implement our algorithm with maximum separating set size $\eta = 2$ since these families are known to satisfy (η, γ)-local-separation with $\eta \leq 2$ (Anandkumar et al., 2012).

We generate a random graph with p nodes using the `igraph` library in R, assigning every edge a weight from a Uniform$(0.1, 1)$ distribution. We then use the `rmvDAG` function from the `pcalg` library to simulate n observations from the DAG. This is repeated 20 times for each thresholding level α; average true and false positive rates for both algorithms are reported over the grid of α values. Our grid of α values produces partial receiver operating characteristic (pROC) curves for varying sample sizes and graph structures, which are used to assess the estimation accuracy of the methods.

For both Erdős-Rényi and power law DAGs, we consider a low-dimensional setting with $p = 100$ nodes and $n = 200$ observations, and two high-dimensional settings with $(p, n) = (200, 100)$ and $(p, n) = (500, 200)$. In all settings, the DAGs are set to have an average degree of 2. The maximum degrees of the Erdős-Rényi graphs range from 5 to 7. The maximum degrees for power law graphs increase with p and are 42, 69, and 71 for the three simulation settings.

Results for Erdős-Rényi DAGs are shown in Figure 2. In this setting, all algorithms perform almost identically in both low and high-dimensional cases. This is because for the relatively small maximum degree in Erdős-Rényi graphs, the sizes of conditioning sets for PC and rPC are not very different.

Results for power law DAGs are shown in Figure 3. In this case, both versions of rPC offer improved accuracy compared to the PC-Algorithm, in both low and high-dimensional

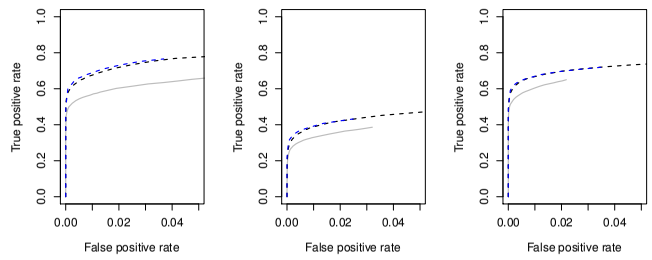


Figure 3: Average true vs. false positive rates for PC-Algorithm (grey solid line), rPC-approx (black dashed line), and rPC-full (blue dashed line) estimating power law DAGs. Left: $p = 100, n = 200$; centre: $p = 200, n = 100$; right: $p = 500, n = 200$.

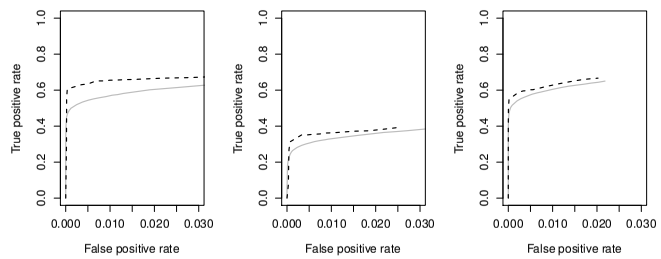


Figure 4: Average true vs. false positive rates for PC-Algorithm (grey solid line) and rPC-approx with BIC-tuned η (black dashed line) estimating power law DAGs. Left: $p = 100, n = 200$; centre: $p = 200, n = 100$; right: $p = 500, n = 200$.

settings. These results confirm our theoretical findings, and show that our algorithm performs better at estimating DAGs with hub nodes than the PC-Algorithm.

Additional simulations in Appendix B show similar results in more dense DAGs. As the underlying DAG becomes more dense, all methods perform worse; however, our algorithms maintain an advantage over the PC-Algorithm in the power law setting. All of these simulations also confirm that rPC-full and rPC-approx perform very similarly. This suggests that the situation where conditioning on non-local sets is required occurs with very low probability, indicating that rPC-approx provides suitably good estimates.

We also compare the runtimes of the algorithms for these settings. Over 100 iterations, we generate a random data set, and apply all the algorithms with a range of tuning parameters. Specifically, we set $\eta = 2$ for rPC, and $\alpha = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. We then take the total runtime over all parameters, and compare by considering the mean value of $100\left(\frac{time_{rPC}}{time_{PC}}\right)$ for both rPC-approx and rPC-full. The results are shown in Table 2. We observe that rPC-approx is significantly faster than the PC-Algorithm for power law graphs. As expected, our implementation of rPC-full is slower than the PC-Algorithm in all settings, given its exhaustive search over all possible separating sets. Given the results indicating that rPC-approx performs almost identically to rPC-full, we suggest that practitioners use rPC-approx, possibly combined with BIC tuning if a suitable η bound is not known. We present relevant simulation results in the next section.

Graph family	p	n	rPC-approx	rPC-full
Erdős-Rényi	100	200	0.98	3.08
Erdős-Rényi	200	100	1.00	6.16
Erdős-Rényi	500	200	1.00	23.25
Power law	100	200	0.60	1.13
Power law	200	100	0.92	3.81
Power law	500	200	0.36	5.23

Table 2: Empirical ratio of rPC-approx and rPC-full runtimes to PC-Algorithm runtime.

5.2. BIC-tuned η Parameter

In this section, we consider simulations where the maximum size of the conditioning set for rPC-approx, η, is selected to maximize the BIC score defined in (3). To this end, we consider power law DAGs with the same low and high-dimensional settings as before. We select the value of $\eta \in \{1, 2, 3, 4\}$ which maximizes the BIC at each α value. While rPC-full could also be tuned in this way, it would be computationally expensive to consider η beyond 3 for $p > 200$. The results in Figure 4 show that our algorithm maintains an advantage over the PC-Algorithm. Interestingly, for values of α which yielded the best estimation accuracy, the optimal η selected was most frequently 1, which confirms our intuition from Section 3 that the η parameter for a graph family should be seen as an upper bound for estimating DAGs using our algorithm.

6. Application: Estimation of Gene Regulatory Networks

We apply our algorithms and the PC-Algorithm to a gene expression data set of $n = 487$ patients with prostate cancer from The Cancer Genome Atlas (Cancer Genome Atlas Research Network, 2012). We select $p = 272$ genes with known network structure from BioGRID (Stark et al., 2006), and attempt to recover this network from the data. We choose the tuning parameters for both rPC algorithms and the p-value threshold for the PC-Algorithm by searching over a grid of values and selecting those which yielded the largest BIC (3). We found that the best rPC algorithm in terms of BIC was rPC-approx with $\eta = 3$ and $\alpha = 0.09$, while the best p-value threshold for the PC-Algorithm was 0.07.

The BioGRID database provides valuable information about known gene regulatory interactions. However, this databse mainly capture genetic interactions in normal cells. Thus, the information from BioGRID may not correctly capture interactions in cancerous cells, which are of interest in our application (Ideker and Krogan, 2012). Despite this limitation, highly connected hub genes in the BioGRID network, which usually correspond to transcription factors, are expected to stay highly connected in cancer cells. Therefore, to evaluate the performance of the two methods, we focus here on the identification of hub genes, which are often most clinically relevant (Horvath et al., 2006; Jeong et al., 2001; Carlson et al., 2006).

The two estimated networks and their hub genes are visualized in Figure 5. Here, we define hub genes as nodes with degree at least 8, which corresponds to the 75th percentile in the degree distribution of both estimates. rPC-approx identifies 19 of 57 true hubs, while the PC-Algorithm only identifies 6. Interestingly, several of the hub genes uniquely identified by rPC are known to be associated with prostate cancer, including ACP1, ARHGEF12, CDH1, EGFR, and PLXNB1 (Ruela-de Sousa et al., 2016; Robbins et al., 2011; Ratz et al., 2017; Sun et al., 2017; Liu et al., 2017). These results suggest that rPC may be a promising alternative for estimating biological networks, where highly-connected nodes are of clinical importance.

Examining the two networks also indicates that for nodes with small degrees, the estimated neighborhoods from rPC are very similar to those from the PC-Algorithm. To assess this observation, we consider the induced subgraph of nodes with degree at most 5 in the PC-Algorithm estimate. The F_1 score—which is a weighted average of precision and recall—between the two estimates of this sparse subnetwork is 0·86. This value indicates that the two algorithms perform very similarly over sparse nodes.

7. Discussion

Our new algorithm for learning directed acyclic graphs (DAGs) by conditioning on small sets leads to more efficient computation and estimation under a less restrictive faithfulness assumption than the PC-Algorithm. However, our weaker faithfulness condition may still not be satisfied for dense DAGs or in structural equation models with edge weights distributed over a larger parameter space. This is shown both geometrically and empirically in Uhler et al. (2013), and remains a direction for future research. Generalizing the idea of restricted conditioning to more complex probability models over DAGs, such as nonlinear SEMs (Voorman et al., 2014) would also be of interest. Finally, the idea of conditioning

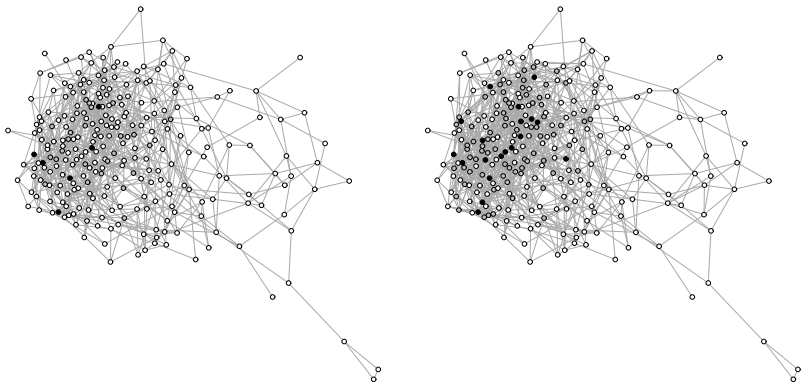


Figure 5: Estimated skeletons of gene regulatory networks in prostate cancer subjects. Black nodes are classified as hubs, having estimated degree of at least 8. Grey nodes are identified hubs that are also considered hubs in the BioGRID data. Left: PC-Algorithm; right: rPC-approx.

on small sets of variables can also be used to develop more efficient hybrid methods for learning DAGs in high dimensions.

Acknowledgments

This work was partially supported by grants from the National Science Foundation (NSF grant DMS/NIGMS-1561814 and DMS-1722246) and the National Institute of Health (NIH K01-HL124050 and R01-GM114029).

Appendix A. Proof of Theorem 1

In this section, we prove the consistency of rPC-full for estimating the skeleton of a directed acyclic graph under the assumptions stated in the main paper.

We begin by establishing that correlations decay over long paths in the graph, and use this to show that the partial correlation between two non-adjacent nodes, conditional on a suitable set S with small cardinality, is bounded above. We establish this through two possible sufficient conditions: Lemma 1 is based on Assumption 4, which directly assumes that the the total weight over long paths is sufficiently small; on the other hand, Lemma 2 uses Assumption $4'$, which assumes the underlying model is directed walk-summable. Combining this result with Assumption 1, which says that the relevant partial correlations between two adjacent nodes is bounded below, we have oracle consistency of rPC-full. In Lemma 3, we invoke a concentration inequality for sample partial correlations to bound their deviations from population quantities. Using these results, we then prove that there exists a threshold level that consistently recovers the true skeleton in the finite sample setting.

In our first two lemmas, we make use of the local separation property in Assumption 3. While Assumption 3 does not directly concern *local separating sets*, since a d-separating set is a subset of a general separating set, Assumption 3 asymptotically guarantees the existence of a *γ-local d-separator* of size at most η for any two non-neighbouring nodes.

Definition 10 *Given a graph G, a γ-local d-separator $S_\gamma(i,j) \subset V \setminus \{i,j\}$ between non-neighbours i and j minimally d-separates i and j over paths of length at most γ.*

Lemma 11 *Under Assumptions 1-4, the partial correlation between non-neighbours i and j satisfies*

$$\min_{S \in S_{\eta,\gamma}} |\rho(i,j \mid S)| = O(\beta^\gamma).$$

where $S_{\eta,\gamma}$ is the set of γ-local d-separators of size at most η.

Proof

Recall the form of the linear structural equation model:

$$X_k = \sum_{j \in pa(k)} \rho_{jk} X_j + \epsilon_k,$$

where ϵ_k are independent and $Var(\epsilon_k) = \sigma_k^2 < \infty$ for all k.

Let A_G denote the lower-triangular weighted adjacency matrix for the graph G, obtained by ordering the nodes according to a causal order (Shojaie and Michailidis, 2010), so that $j \in pa(k)$ implies $j < k$. Then, as shown by Shojaie and Michailidis (2010) for the Gaussian case and by Loh and Bühlmann (2014) in general linear structural equation models,

$$\Sigma_G = (I - A_G)^{-1} D (I - A_G)^{-T},$$

where $D = \text{diag}(\sigma_1^2, ..., \sigma_p^2)$.

First, suppose that $\sigma_i^2 = 1$ for all i. We consider the conditional covariance $\Sigma_G(i,j \mid S)$ where i and j are non-neighbours, and S is the γ-local d-separator, as defined above. Let

π denote a d-connecting path between i and j of length $l(\pi)$, and $\rho_1, \ldots, \rho_l$ denote the edge weights along the path. By conditioning on S, we have that covariance is only induced through d-connecting paths of length greater than γ, as referenced in the main paper. Then,

$$\Sigma_G(i,j \mid S) = \sum_{\substack{\pi: i \leftrightarrow j \\ \pi \cap S = \emptyset}} \sum_{\pi: l(\pi)=l} \prod_{k=1}^{l} \rho_k = \sum_{\substack{\pi: i \leftrightarrow j \\ l(\pi)=\gamma+1}} \sum_{\pi: l(\pi)=l} \prod_{k=1}^{l} \rho_k.$$

Therefore:

$$\begin{aligned}
|\Sigma_G(i,j \mid S)| &= \left| \sum_{\substack{\pi: i \leftrightarrow j \\ l(\pi)=\gamma+1}}^{p-1} \sum_{\pi: l(\pi)=l} \prod_{k=1}^{l} \rho_k \right| \\
&\leq \sum_{\substack{\pi: i \leftrightarrow j \\ l(\pi)=\gamma+1}}^{p-1} \sum_{\pi: l(\pi)=l} \prod_{k=1}^{l} |\rho_k| && \text{(by triangle inequality)} \\
&= O(\beta^\gamma). && \text{(by Assumption 4)}
\end{aligned}$$

Now, suppose that not all $\sigma_i^2 = 1$. Then, let $\sigma_{max}^2 = \max_i \sigma_i^2$. We have:

$$\begin{aligned}
|\Sigma_G(i,j \mid S)| &\leq \sum_{\substack{\pi: i \leftrightarrow j \\ l(\pi)=\gamma+1}}^{p-1} \sum_{\pi: l(\pi)=l} \sigma_{max}^2 \prod_{k=1}^{l} |\rho_k| \\
&= \sigma_{max}^2 \sum_{\substack{\pi: i \leftrightarrow j \\ l(\pi)=\gamma+1}}^{p-1} \sum_{\pi: l(\pi)=l} \prod_{k=1}^{l} |\rho_k| \\
&= \sigma_{max}^2 O(\beta^\gamma) \\
&= O(\beta^\gamma). && \text{(by Assumption 2)}
\end{aligned}$$

Finally, we have $|\rho(i,j \mid S)| = \dfrac{|\Sigma_G(i,j \mid S)|}{\sqrt{\Sigma_G(i,i \mid S)\Sigma_G(j,j \mid S)}} = O(\beta^\gamma)$ by Assumption 2, since the conditional variances are functions of the marginal variances, which are bounded. ■

Next, we show the same result by assuming directed walk-summability of the model.

Lemma 12 *Under Assumptions 1-3 and Assumption 4′, the partial correlation between non-neighbours i and j satisfies*

$$\min_{S \in S_{\eta,\gamma}} |\rho(i,j \mid S)| = O(\beta^\gamma).$$

where $S_{\eta,\gamma}$ is the set of γ-local d-separators of size at most η.

Proof

Recall from the proof of Lemma 1,

$$\Sigma_G = (I - A_G)^{-1} D (I - A_G)^{-T},$$

where $D = \mathrm{diag}(\sigma_1^2, ..., \sigma_p^2)$. First, suppose that $\sigma_i^2 = 1$ for all i. Then, we can write:

$$\begin{aligned}
\Sigma_G &= \left(\sum_{r=0}^{\infty} A_G^r \right) \left(\sum_{r=0}^{\infty} A_G^r \right)^T \\
&= \left(\sum_{r=0}^{\gamma} A_G^r + \sum_{r=\gamma+1}^{\infty} A_G^r \right) \left(\sum_{r=0}^{\gamma} A_G^r + \sum_{r=\gamma+1}^{\infty} A_G^r \right)^T.
\end{aligned}$$

Now, let Σ_H denote the covariance matrix induced by only considering d-connecting paths of length at most γ. For convenience, let $\Lambda_H := \sum_{r=0}^{\gamma} A_G^r$ and $R_\gamma := \sum_{r=\gamma+1}^{\infty} A_G^r$. Considering their spectral norms, denoted by $\|\cdot\|$, we have by walk-summability that $\|\Lambda_H\| \leq \dfrac{1-\beta^{\gamma+1}}{1-\beta}$ and $\|R_\gamma\| \leq \dfrac{\beta^{\gamma+1}}{1-\beta}$. Then,

$$\begin{aligned}
\Sigma_G &= (\Lambda_H + R_\gamma)(\Lambda_H + R_\gamma)^T \\
&= \Lambda_H \Lambda_H^T + \Lambda_H R_\gamma^T + R_\gamma \Lambda_H^T + R_\gamma R_\gamma^T \\
&= \Sigma_H + \Lambda_H R_\gamma^T + R_\gamma \Lambda_H^T + R_\gamma R_\gamma^T
\end{aligned}$$

Now, defining $E_\gamma := \Sigma_G - \Sigma_H$ and taking spectral norms, we get:

$$\begin{aligned}
\|E_\gamma\| = \|\Sigma_G - \Sigma_H\| &= \|\Lambda_H R_\gamma^T + R_\gamma \Lambda_H^T + R_\gamma R_\gamma^T\| \\
&\leq \|\Lambda_H R_\gamma^T\| + \|R_\gamma \Lambda_H^T\| + \|R_\gamma R_\gamma^T\| \\
&\leq 2\|\Lambda_H\|\|R_\gamma\| + \|R_\gamma\|^2 \\
&\leq 2\left(\frac{1-\beta^{\gamma+1}}{1-\beta}\right)\left(\frac{\beta^{\gamma+1}}{1-\beta}\right) + \left(\frac{\beta^{\gamma+1}}{1-\beta}\right)^2 \\
&= \frac{2\beta^{\gamma+1} - 2\beta^{2\gamma+2}}{(1-\beta)^2} + \frac{\beta^{2\gamma+2}}{(1-\beta)^2} \\
&= \frac{2\beta^{\gamma+1} - \beta^{2\gamma+2}}{(1-\beta)^2} \\
&= \frac{\beta^{\gamma+1}(2-\beta^{\gamma+1})}{(1-\beta)^2} \\
&= O(\beta^\gamma).
\end{aligned} \tag{S1}$$

Now, suppose that not all $\sigma_i^2 = 1$. Then, following the same expansion of Σ_G as above, we have:

$$\|E_\gamma\| = \|D\| \frac{\beta^{\gamma+1}(2-\beta^{\gamma+1})}{(1-\beta)^2} = \sigma_{max}^2 \frac{\beta^{\gamma+1}(2-\beta^{\gamma+1})}{(1-\beta)^2} = O(\beta^\gamma), \tag{S2}$$

by Assumption 2, where $\sigma^2_{max} = \max_i \sigma^2_i$.

We now show that $|\rho(i,j \mid S)| = O(\|E_\gamma\|) = O(\beta^\gamma)$ where S is a γ-local d-separator between i and j. Let $A = \{i,j\} \cup S$ and $B = V \setminus A$. Consider the marginal precision matrix, $P := \{\Sigma_G(A,A)\}^{-1}$. Then, using the Schur complement, we can write this as

$$P = \Sigma_G^{-1}(A,A) - \Sigma_G^{-1}(A,B)\{\Sigma_G^{-1}(B,B)\}^{-1}\Sigma_G^{-1}(B,A).$$

Specifically, the partial correlation of X_i and X_j conditional on S is given by $\frac{P_{1,2}}{(P_{1,1}P_{2,2})^{1/2}} = O(P_{1,2})$, by Assumption 2.

Recall from (S1) that $\Sigma_G = \Sigma_H + E_\gamma$. Let F_γ be the matrix such that $\Sigma_G^{-1} = \Sigma_H^{-1} + F_\gamma$. Because Σ_H only considers covariance induced by paths of length at most γ, we have that $\Sigma_H^{-1}(A,A)_{1,2} = 0$.

Thus,

$$\begin{aligned}
|\{\Sigma_G(A,A)\}^{-1}_{1,2}| &= |\Sigma_G^{-1}(A,A)_{1,2} - \Sigma_G^{-1}(A,B)\{\Sigma_G^{-1}(B,B)\}^{-1}\Sigma_G^{-1}(B,A)_{1,2}| \\
&= |\Sigma_H^{-1}(A,A)_{1,2} + F_\gamma(A,A)_{1,2} - \Sigma_G^{-1}(A,B)\{\Sigma_G^{-1}(B,B)\}^{-1}\Sigma_G^{-1}(B,A)_{1,2}| \\
&= |F_\gamma(A,A)_{1,2} - \Sigma_G^{-1}(A,B)\{\Sigma_G^{-1}(B,B)\}^{-1}\Sigma_G^{-1}(B,A)_{1,2}| \\
&\le \|F_\gamma(A,A) - \Sigma_G^{-1}(A,B)\{\Sigma_G^{-1}(B,B)\}^{-1}\Sigma_G^{-1}(B,A)\|_\infty \\
&\le \|F_\gamma(A,A) - \Sigma_G^{-1}(A,B)\{\Sigma_G^{-1}(B,B)\}^{-1}\Sigma_G^{-1}(B,A)\|.
\end{aligned}$$

However, since $\Sigma_G^{-1}(A,B)\{\Sigma_G^{-1}(B,B)\}^{-1}\Sigma_G^{-1}(B,A)$ is positive semi-definite, $|\{\Sigma_G(A,A)\}^{-1}_{1,2}| \le \|F_\gamma(A,A)\|$. We next show that $\|F_\gamma\| = O(\|E_\gamma\|) = O(\beta^\gamma)$. First, note that:

$$\begin{aligned}
F_\gamma &= \Sigma_G^{-1} - \Sigma_H^{-1} \\
&= (\Sigma_H + E_\gamma)^{-1} - \Sigma_H^{-1} \\
&= \Sigma_H^{-1} - \Sigma_H^{-1}(E_\gamma^{-1} + \Sigma_H^{-1})^{-1}\Sigma_H^{-1} - \Sigma_H^{-1} && \text{(by Woodbury)} \\
&= -\Sigma_H^{-1}(E_\gamma^{-1} + \Sigma_H^{-1})^{-1}\Sigma_H^{-1}
\end{aligned}$$

Then, taking spectral norms, and noting that $\Sigma_G = \Sigma_H + E_\gamma$:

$$\begin{aligned}
\|F_\gamma\| &\le \|\Sigma_H^{-1}\|\|(E_\gamma^{-1} + \Sigma_H^{-1})^{-1}\|\|\Sigma_H^{-1}\| && \text{(by sub-multiplicity)} \\
&= \|\Sigma_H^{-1}\|^2\|E_\gamma - E_\gamma(\Sigma_H + E_\gamma)^{-1}E_\gamma\| && \text{(by Woodbury)} \\
&= \|\Sigma_H^{-1}\|^2\|E_\gamma(I - \Sigma_G^{-1}E_\gamma)\| \\
&\le \|\Sigma_H^{-1}\|^2\|E_\gamma\|\|I - \Sigma_G^{-1}E_\gamma\| && \text{(by sub-multiplicity)} \\
&\le \|\Sigma_H^{-1}\|^2\|E_\gamma\|(1 + \|\Sigma_G^{-1}E_\gamma\|) && \text{(by triangle inequality)} \\
&\le \|\Sigma_H^{-1}\|^2\|E_\gamma\|(1 + \|\Sigma_G^{-1}\|\|E_\gamma\|) && \text{(by sub-multiplicity)} \\
&\le J\|E_\gamma\|^2, && \text{(by boundedness of } \|\Sigma_G^{-1}\| \ge |\Sigma_H^{-1}\|)
\end{aligned}$$

for some constant J. Then, by walk-summability, $\|F_\gamma\| = O(|E_\gamma\|)$. Hence, $|\rho(i,j \mid S)| = O(\beta^\gamma)$ by (S1).

■

By combining the result from either Lemma 1 or 2 with the λ-path-faithfulness assumption, we achieve oracle consistency for our algorithm given a threshold level α such that $\alpha = O(\lambda)$, $\alpha = \Omega(\beta^\gamma)$.

Next, we consider the finite sample setting, and establish a concentration inequality for sample partial correlations, under sub-Gaussian distributions, using a result from Ravikumar et al. (2011).

Lemma 13 *Assume $X = (X_1, ..., X_p)$ is a zero-mean random vector with covariance matrix Σ such that each $X_i/\Sigma_{ii}^{1/2}$ is sub-Gaussian with parameter σ. Assume $\|\Sigma\|_\infty$ and σ are bounded. Then, the empirical partial correlation obtained from n samples satisfies, for some bounded $M > 0$:*

$$P\Big(\max_{i \ne j, |S| \le \eta} |\hat{\rho}(i,j \mid S) - \rho(i,j \mid S)| > \epsilon\Big) \le 4\Big(3 + \frac{3}{2}\eta + \frac{1}{2}\eta^2\Big)p^{\eta+2}\exp\Big(-\frac{n\epsilon^2}{M}\Big)$$

for all $\epsilon \le \max_i(\Sigma_{ii})8(1 + 4\sigma^2)$.

Proof

Using the recursive formula for partial correlation, for any $k \in S$

$$\rho(i,j \mid S) = \frac{\rho(i,j \mid S \setminus k) - \rho(i,k \mid S \setminus k)\rho(k,j \mid S \setminus k)}{(1 - \rho^2(i,k \mid S \setminus k))^{1/2}(1 - \rho^2(k,j \mid S \setminus k))^{1/2}}.$$

For example, with $S = \{k\}$, we can simplify this to:

$$\rho(i,j \mid S) = \frac{\rho(i,j) - \rho(i,k)\rho(k,j)}{(1 - \rho^2(i,k))^{1/2}(1 - \rho^2(k,j))^{1/2}},$$

where $\rho(i,j) = \Sigma_{ij}/(\Sigma_{ii}\Sigma_{jj})^{1/2}$.

Rewriting in terms of elements of Σ, we then decompose the empirical partial correlation deviance from the true partial correlation into the deviances of covariance terms. Here, the event of the empirical partial correlation being within ϵ distance of the true partial correlation is contained in the union of the empirical covariance terms being within $C\epsilon$ distance of the true covariance terms for a sufficiently large $C > 0$:

$$\begin{aligned}
\Big[|\hat{\rho}(i,j \mid S) - \rho(i,j \mid S)| > \epsilon\Big] \subset &\Big[|\hat{\Sigma}_{ij} - \Sigma_{ij}| > C\epsilon\Big] \bigcup \Big[|\hat{\Sigma}_{ii} - \Sigma_{ii}| > C\epsilon\Big] \bigcup \Big[|\hat{\Sigma}_{jj} - \Sigma_{jj}| > C\epsilon\Big] \\
&\bigcup_{k \in S}\Big[|\hat{\Sigma}_{ik} - \Sigma_{ik}| > C\epsilon\Big] \\
&\bigcup_{k \in S}\Big[|\hat{\Sigma}_{jk} - \Sigma_{jk}| > C\epsilon\Big] \\
&\bigcup_{k \le k' \in S}\Big[|\hat{\Sigma}_{kk'} - \Sigma_{kk'}| > C\epsilon\Big],
\end{aligned}$$

The number of events on the right-hand side is $3 + |S| + |S| + |S|^2 - \binom{|S|}{2}$. For $|S| \leq \eta$, the number of events is then bounded by $3 + \frac{3}{2}\eta + \frac{1}{2}\eta^2$. Then, by applying Lemma 1 in Ravikumar et al. (2011), we have that, for any i, j:

$$Pr\Big(|\hat{\rho}(i,j \mid S) - \rho(i,j \mid S)| > \epsilon\Big) \leq 4\Big(3 + \frac{3}{2}\eta + \frac{1}{2}\eta^2\Big) \exp\Big(-\frac{n\epsilon^2}{K}\Big),$$

for some $K > 0$, bounded when $\|\Sigma\|_\infty$ and σ are bounded. From here, the result follows. ■

Combining the results established in Lemmas 1-3, we now prove the consistency of our algorithm in the finite sample setting.

Let α denote the threshold where if $\hat{\rho}_G(i,j \mid S) < \alpha$, the edge (i,j) is deleted. Let G_S denote the true undirected skeleton of G, and let $S_{\eta,\gamma}$ denote the set of γ-local d-separators or size at most η.

For any $(i,j) \notin G_S$, define the false positive event as

$$F_1(i,j) = \Big[\min_{S \in S_{\eta,\gamma}} |\hat{\rho}_G(i,j \mid S)| > \alpha\Big].$$

Define

$$\theta_{max} = \max_{(i,j)\notin G_S} \min_{S \in S_{\eta,\gamma}} |\rho_G(i,j \mid S)|$$

and

$$\hat{\theta}_{max} = \max_{(i,j)\notin G_S} \min_{S \in S_{\eta,\gamma}} |\hat{\rho}_G(i,j \mid S)|.$$

Consider

$$\begin{aligned} Pr\Big\{\bigcup_{(i,j)\notin G_S} F_1(i,j)\Big\} &= Pr(\hat{\theta}_{max} > \alpha) \\ &= Pr(|\hat{\theta}_{max} - \theta_{max}| > |\alpha - \theta_{max}|) \\ &= O\Big[p^{\eta+2} \exp\Big\{-\frac{n(\alpha - \theta_{max})^2}{M}\Big\}\Big] \qquad \text{(by Lemma 3)} \end{aligned}$$

where $\theta_{max} = O(\beta^\gamma)$ by Lemma 1 and 2.

For any true edge $(i,j) \in G_S$, define the false negative event as

$$F_2(i,j) = \Big[\min_{S \subset V \setminus \{i,j\}, |S| \leq \eta} |\hat{\rho}_G(i,j \mid S)| < \alpha\Big].$$

Define

$$\theta_{min} = \min_{(i,j)\in G_S} \min_{S \subset V \setminus \{i,j\}, |S| \leq \eta} |\rho_G(i,j \mid S)|$$

and

$$\hat{\theta}_{min} = \min_{(i,j)\in G_S} \min_{S \subset V \setminus \{i,j\}, |S| \leq \eta} |\hat{\rho}_G(i,j \mid S)|.$$

Graph family	Expected degree	Pr(RSF)	Pr(PF)
Erdős-Rényi	2	0.94	0.98
Erdős-Rényi	5	0.10	0.61
Power law	2	0.95	0.97
Power law	6	0.08	0.60

Table 3: Empirical probabilities of random DAGs of size $p = 10$ satisfying faithfulness conditions; RSF refers to restricted strong faithfulness of the PC-Algorithm, and PF refers to path faithfulness of reduced PC (rPC).

Consider

$$\begin{aligned} Pr\Big\{\bigcup_{(i,j)\in G_S} F_2(i,j)\Big\} &= Pr(\hat{\theta}_{min} < \alpha) \\ &= Pr(|\theta_{min} - \hat{\theta}_{min}| > |\theta_{min} - \alpha|) \\ &= O\Big[p^{\eta+2} \exp\Big\{-\frac{n(\alpha - \theta_{min})^2}{K}\Big\}\Big] \qquad \text{(by Lemma 3)} \end{aligned}$$

where $\theta_{min} = \Omega(\lambda)$ by restricted path-faithfulness assumption.

Under our assumptions, we have that $n = \Omega\{(\log p)^{1/1-2c}\}$, and $\lambda = \Omega(n^{-c})$ with $c \in (0, 1/2)$. Rewriting in terms of λ, we have $n = \Omega(\frac{\log p}{\lambda^2})$. Then, by selecting α such that $\alpha = O(\lambda)$, $\alpha = \Omega(\beta^\gamma)$, we have $Pr\{\bigcup_{(i,j)\notin G_S} F_1(i,j)\} = o(1)$ and $Pr\{\bigcup_{(i,j)\in G_S} F_2(i,j)\} = o(1)$. This completes the proof of Theorem 1.

Appendix B. Additional Simulation Results

In this section, we present some simulation results comparing our algorithms to the PC-Algorithm in estimating dense graphs. The simulation setup is otherwise identical to that described in the main paper. We consider a low-dimensional setting, with $p = 100$ and $n = 200$, as well as a high-dimensional setting, with $p = 200$ and $n = 100$. For Erdős-Rényi graphs, we use a constant edge probability of 0.05, corresponding to an expected degree of 5 for the low-dimensional graph and 10 for the high-dimensional graph. For power law graphs, we use an expected degree of 6 in both graphs.

While estimation quality is worse in the dense setting for all methods, we observe in Figure 6 and Figure 7 similar trends as in the main paper. Our methods perform as well as the PC-Algorithm in estimating Erdős-Rényi graphs, and shows some improvement for power law graphs. Both versions of rPC give similar results.

We also include more simulation results comparing path faithfulness to the restricted strong faithfulness assumption. As in Section 4.2, 1000 random DAGs were generated from Erdős-Rényi and power law families, with edge weights drawn independently from a Uniform$(-1, 1)$ distribution. Here, we consider graphs of size $p = 10$ and $p = 20$, with faithfulness condition parameters $\lambda = 0.001$ and $\eta = 2$. The results are shown in Table 3 and Table 4.

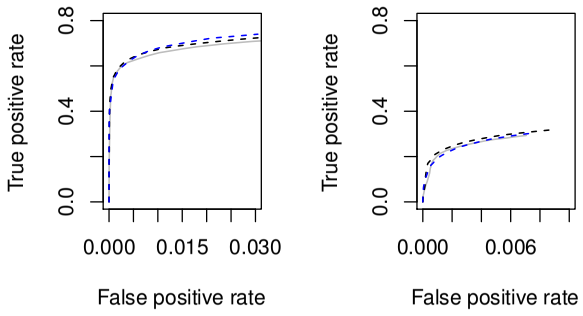


Figure 6: Average true vs. false positive rates for PC-Algorithm (grey solid line), rPC-approx (black dashed line), and rPC-full (blue dashed line) estimating Erdős-Rényi graphs. Left: $p = 100, n = 200$, average degree 5; right: $p = 200, n = 100$, average degree 10.

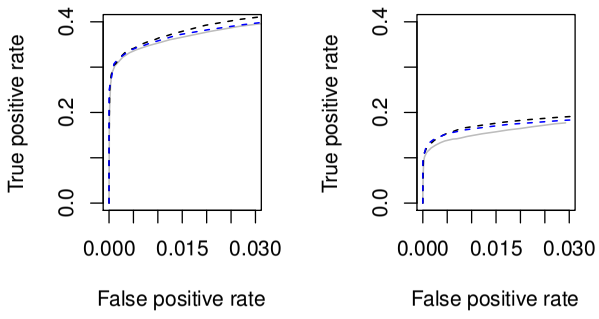


Figure 7: Average true vs. false positive rates for PC-Algorithm (grey solid line), rPC-approx (black dashed line), and rPC-full (blue dashed line) estimating power law graphs with average degree 6. Left: $p = 100, n = 200$; right: $p = 200, n = 100$.

Although it is harder for both conditions to be met as graph size and density increase, we see that path faithfulness is still more likely to be satisfied than restricted strong faithfulness, particularly with power law graphs.

Graph family	Expected degree	Pr(RSF)	Pr(PF)
Erdős-Rényi	2	0.66	0.86
Erdős-Rényi	5	0	0.01
Power law	2	0.04	0.51
Power law	6	0	0.01

Table 4: Empirical probabilities of random DAGs of size $p = 30$ satisfying faithfulness conditions; RSF refers to restricted strong faithfulness of the PC-Algorithm, and PF refers to path faithfulness of reduced PC (rPC).

Finally, we consider the plausibility of Assumption 4 when the SEM coefficients are not bounded by 1 in absolute value. Suppose that the data matrix X has been normalized by column-wise standard deviations. We show that this transformation preserves the original network structure, and leads to most edge weights being bounded by 1 in absolute value. Consider an edge $j \to k$ and let $W_j := \{X_i : i \in pa(k) \setminus \{j\}\}$. Then, taking the conditional covariance:

$$\begin{aligned} Cov(X_j, X_k|W_j) &= Cov\left(X_j, \rho_{jk}X_j + \sum_{i \in pa(k)\setminus j} \rho_{ik}X_i + \epsilon_k \middle| W_j\right) \\ &= \rho_{jk} Var(X_j|W_j). \end{aligned}$$

We can therefore write:

$$\rho_{jk} = \frac{Cov(X_j, X_k|W_j)}{Var(X_j|W_j)}.$$

Now, let $\tilde{X}_k = X_k/sd(X_k)$ for all k. Consider the edge weights $\tilde{\rho}_{jk}$ corresponding to the SEM for this transformed data. We have:

$$\tilde{\rho}_{jk} = \frac{sd(X_j)}{sd(X_k)} \frac{Cov(X_j, X_k|W_j)}{Var(X_j|W_j)} = \frac{sd(X_j)}{sd(X_k)} \rho_{jk}$$

Clearly, $\tilde{\rho}_{jk} = 0$ if and only if $\rho_{jk} = 0$. Therefore, we recover the same network by applying our algorithm to the transformed data. Furthermore,

$$\begin{aligned} |\tilde{\rho}_{jk}| &= \sqrt{\frac{Var(X_j)}{Var(X_k)} \rho_{jk}^2} \\ &= \sqrt{\frac{Var(X_j)\rho_{jk}^2}{Var(X_j)\rho_{jk}^2 + \sigma_k^2 + \sum_{i \in pa(k)\setminus\{j\}} \rho_{ik}^2 Var(X_i) + 2\sum_{i,i' \in pa(k)} \rho_{ik}\rho_{i'k} Cov(X_i, X_i')}} \end{aligned}$$

Thus, $|\tilde{\rho}_{jk}| > 1$ only if

$$\sigma_k^2 + \sum_{i \in pa(k)\setminus\{j\}} \rho_{ik}^2 Var(X_i) + 2 \sum_{i,i' \in pa(k)} \rho_{ik}\rho_{i'k} Cov(X_i, X_i') < 0$$

Graph family	p	Distribution	$Pr(\|\rho\| > 1)$
Erdős-Rényi	200	Uniform(-10, 10)	0.0036
	200	Normal(0, 3^2)	0.0017
	500	Uniform(-10, 10)	0.0036
	500	Normal(0, 3^2)	0.0012
Power law	200	Uniform(-10, 10)	0.0045
	200	Normal(0, 3^2)	0.0045
	500	Uniform(-10, 10)	0.0034
	500	Normal(0, 3^2)	0.0037

Table 5: Empirical average probabilities of standardized coefficients exceeding 1 in absolute value.

Rewriting the variance and covariance terms as trek sums, we obtain:

$$\sigma_k^2 + \sum_{i \in pa(k) \setminus \{j\}} \rho_{ik}^2 \sum_{\substack{\pi: u \leftrightarrow i \\ u \in V \setminus \{i\}}} \sigma_w^2 \prod_{\rho_e \in \pi} \rho_e^2 + 2 \sum_{i, i' \in pa(k)} \rho_{ik} \rho_{i'k} \sum_{\pi: i \leftrightarrow i'} \sigma_w \prod_{\rho_e \in \pi} \rho_e < 0$$

where w denotes the source or common node in the trek.

Since the first two terms in this expression will always be positive, $|\tilde{\rho}_{jk}| > 1$ only if the covariance sum is both negative and greater in magnitude than the sum of the first two terms. A term in the covariance sum will be negative if there is an odd number of negative terms in the product, and the entire sum will be negative only if there are a sufficient number of these to make the whole sum negative. Even then, the sum over squared trek products would need to be smaller than the covariance term sum in absolute value. Moreover, even if some $|\tilde{\rho}_{jk}| > 1$, the product over the entire trek may still be less than 1 if most of the other coefficients along a trek are small. Thus, we may still expect the product over the trek to be small.

Next, we provide simulation results showing that standardized SEM coefficients in graph configurations we consider rarely exceed 1 in absolute value. Specifically, we consider Erdős-Rényi and power law graphs, with $p \in \{200, 500\}$ having average degree 2. We consider edge weights from both a $\text{Unif}(-10, 10)$ and $N(0, 3^2)$ distribution. For each graph, we estimate the standardized coefficients using the `fitDag` function in the `ggm` library, and compute the proportion of edge weights that are greater than 1 in absolute value. We repeat this process over 1,000 iterations. The results are shown in Table 5. This experiment shows that, with high probability, the covariance induced over long treks between any two nodes is indeed small.

References

Silvia Acid and Luis M de Campos. Approximations of causal networks by polytrees: an empirical study. In *Advances in Intelligent Computing*, pages 149–158. Springer, 1994.

Animashree Anandkumar, Vincent YF Tan, Furong Huang, and Alan S Willsky. High-dimensional Gaussian graphical model selection: walk summability and local separation criterion. *Journal of Machine Learning Research*, 13(1):2293–2337, 2012.

Cancer Genome Atlas Research Network. Comprehensive genomic characterization of squamous cell lung cancers. *Nature*, 489(7417):519–525, 2012.

Marc RJ Carlson, Bin Zhang, Zixing Fang, Paul S Mischel, Steve Horvath, and Stanley F Nelson. Gene connectivity, function, and sequence conservation: predictions from modular yeast co-expression networks. *BMC Genomics*, 7(1):40, 2006.

Hao Chen and Burt M Sharp. Content-rich biological network constructed by mining pubmed abstracts. *BMC Bioinformatics*, 5(1):147, 2004.

David Maxwell Chickering. Learning Bayesian networks is NP-complete. In *Learning From Data*, pages 121–130. Springer, 1996.

Richard Durrett. *Random Graph Dynamics*, volume 200. Cambridge University Press, 2007.

Daniel Eaton and Kevin Murphy. Bayesian structure learning using dynamic programming and MCMC. *arXiv preprint arXiv:1206.5247*, 2012.

Rina Foygel and Mathias Drton. Extended Bayesian information criteria for Gaussian graphical models. In *Neural Information Processing Systems (NeurIPS)*, pages 604–612, 2010.

Nir Friedman, Michal Linial, and Iftach Nachman. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

Min Jin Ha, Wei Sun, and Jichun Xie. PenPC: a two-step approach to estimate the skeletons of high-dimensional directed acyclic graphs. *Biometrics*, 72(1):146–155, 2016.

S Horvath, B Zhang, M Carlson, KV Lu, S Zhu, RM Felciano, MF Laurance, W Zhao, S Qi, Z Chen, et al. Analysis of oncogenic signaling networks in glioblastoma identifies aspm as a molecular target. *Proceedings of the National Academy of Sciences*, 103(46): 17402–17407, 2006.

Juan F. Huete and Luis M. Campos. Learning causal polytrees. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 180–185. Springer, 1993. doi: 10.1007/BFb0028199.

Trey Ideker and Nevan J Krogan. Differential network biology. *Molecular Systems Biology*, 8(1):565, 2012.

Ronald Jansen, Haiyuan Yu, Dov Greenbaum, Yuval Kluger, Nevan J. Krogan, Sambath Chung, Andrew Emili, Michael Snyder, Jack F. Greenblatt, and Mark Gerstein. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003. ISSN 0036-8075. doi: 10.1126/science.1087361.

Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-Algorithm. *Journal of Machine Learning Research*, 8:613–636, May 2007. ISSN 1532-4435.

Jon Kleinberg. The small-world phenomenon: an algorithmic perspective. In *ACM Symposium on Theory of Computing*, pages 163–170. ACM, 2000.

Bide Liu, Xiao Gu, Tianbao Huang, Yang Luan, and Xuefei Ding. Identification of tmprss2-erg mechanisms in prostate cancer invasiveness: involvement of mmp-9 and plexin b1. *Oncology Reports*, 37(1):201–208, 2017.

Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15(1):3065–3105, 2014.

Dmitry M Malioutov, Jason K Johnson, and Alan S Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7(Oct): 2031–2064, 2006.

Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):161–180, 1995.

Preetam Nandy, Alain Hauser, and Marloes H Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *arXiv preprint arXiv:1507.02608*, 2015.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15:2009–2053, 2014.

Leonie Ratz, Mark Laible, Lukasz A Kacprzyk, Stephanie M Wittig-Blaich, Yanis Tolstov, Stefan Duensing, Peter Altevogt, Sabine M Klauck, and Holger Sültmann. Tmprss2: Erg gene fusion variants induce tgf-β signaling and epithelial to mesenchymal transition in human prostate cancer cells. *Oncotarget*, 8(15):25115, 2017.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Christiane M Robbins, Waibov A Tembe, Angela Baker, Shripad Sinari, Tracy Y Moses, Stephen Beckstrom-Sternberg, James Beckstrom-Sternberg, Michael Barrett, James Long, Arul Chinnaiyan, et al. Copy number and targeted mutational analysis reveals novel somatic events in metastatic prostate tumors. *Genome Research*, 21(1):47–55, 2011.

Roberta R Ruela-de Sousa, Elmer Hoekstra, A Marije Hoogland, Karla C Souza Queiroz, Maikel P Peppelenbosch, Andrew P Stubbs, Karin Pelizzaro-Rocha, Geert JLH van Leenders, Guido Jenster, Hiroshi Aoyama, et al. Low-molecular-weight protein tyrosine phosphatase predicts prostate cancer outcome by increasing the metastatic potential. *European Urology*, 69(4):710–719, 2016.

Eric E Schadt, John Lamb, Xia Yang, Jun Zhu, Steve Edwards, Debraj GuhaThakurta, Solveig K Sieberts, Stephanie Monks, Marc Reitman, Chunsheng Zhang, et al. An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, 37(7):710–717, 2005.

Mark Schmidt, Alexandru Niculescu-Mizil, and Kevin Murphy. Learning graphical model structure using l1-regularization paths. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 7, pages 1278–1283, 2007.

Marco Scutari. Learning Bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35(1):1–22, 2010. ISSN 1548-7660. doi: 10.18637/jss.v035.i03.

A. Shojaie, A. Jauhiainen, M. Kallitsis, and G. Michailidis. Inferring Regulatory Networks by Combining Perturbation Screens and Steady State Gene Expression Profiles. *PLoS One*, 9(2):e82393, 2014.

Ali Shojaie and George Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT press, 2nd edition, 2000.

Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34(suppl 1):D535–D539, 2006.

Seth Sullivant, Kelli Talaska, and Jan Draisma. Trek separation for Gaussian graphical models. *The Annals of Statistics*, pages 1665–1685, 2010.

Liang Sun, Jiaju Lü, Sentai Ding, Dongbin Bi, Kejia Ding, Zhihong Niu, and Ping Liu. Hcrp1 regulates proliferation, invasion, and drug resistance via egfr signaling in prostate cancer. *Biomedicine & Pharmacotherapy*, 91:202–207, 2017.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, 41(2):436–463, 2013.

Arend Voorman, Ali Shojaie, and Daniela Witten. Graph estimation with joint additive models. *Biometrika*, 101(1):85, 2014.

Changwon Yoo, Vesteinn Thorsson, and Gregory F Cooper. Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. In *Pacific Symposium on Biocomputing*, volume 7, pages 498–509, 2002.

Jiji Zhang and Peter Spirtes. Strong faithfulness and uniform consistency in causal inference. In *Uncertainty in Artificial Intelligence (UAI)*, pages 632–639. Morgan Kaufmann Publishers Inc., 2002.

Journal of Machine Learning Research 20 (2019) 1-37 Submitted 1/18; Revised 12/19; Published 12/19

On the Convergence of Gaussian Belief Propagation with Nodes of Arbitrary Size

Francois Kamper FRANCOISK@SUN.AC.ZA
Sarel J. Steel SJST@SUN.AC.ZA
Department of Statistics and Actuarial Science
Stellenbosch University
Stellenbosch, South Africa

Johan A. du Preez DUPREEZ@SUN.AC.ZA
Department of Electrical and Electronic Engineering
Stellenbosch University
Stellenbosch, South Africa

Editor: David Barber

Abstract

This paper is concerned with a multivariate extension of Gaussian message passing applied to pairwise Markov graphs (MGs). Gaussian message passing applied to pairwise MGs is often labeled Gaussian belief propagation (GaBP) and can be used to approximate the marginal of each variable contained in the pairwise MG. We propose a multivariate extension of GaBP (we label this GaBP-m) that can be used to estimate higher-dimensional marginals. Beyond the ability to estimate higher-dimensional marginals, GaBP-m exhibits better convergence behavior than GaBP, and can also provide more accurate univariate marginals. The theoretical results of this paper are based on an extension of the computation tree analysis conducted on univariate nodes to the multivariate case. The main contribution of this paper is the development of a convergence condition for GaBP-m that moves beyond the walk-summability of the precision matrix. Based on this convergence condition, we derived an upper bound for the number of iterations required for convergence of the GaBP-m algorithm. An upper bound on the dissimilarity between the approximate and exact marginal covariance matrices was established. We argue that GaBP-m is robust towards a certain change in variables, a property not shared by iterative solvers of linear systems, such as the conjugate gradient (CG) and preconditioned conjugate gradient (PCG) methods. The advantages of using GaBP-m over GaBP are also illustrated empirically.

Keywords: belief propagation, Gaussian distributions, higher-dimensional marginals, preconditioning, inference

1. Introduction

Let $\mathbf{X}_i : d_i \times 1$ for $i = 1, 2, \ldots, p$ be mutually exclusive and exhaustive subvectors of the random vector $\mathbf{X} : k \times 1$. We are concerned with the approximation of the marginal distributions of $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p$ using belief propagation (BP), where it is assumed that $\mathbf{X}$ follows a Gaussian distribution with precision matrix $\mathbf{S}$ and potential vector $\boldsymbol{b}$.

BP is an approximate marginalization algorithm that operates on a specified graph structure. The origin of BP can be traced back to the sum-product algorithm as a decoding algorithm for LDPC codes (Gallager, 1963). When the underlying graph structure forms a tree, BP will provide the correct marginals at convergence. This is in contrast to loopy graphs, where convergence is not guaranteed and, even if convergence occurs, the marginals supplied are not necessarily correct (Pearl, 1988; Weiss and Freeman, 2001). Although this seems to be a damning property, BP has become a popular tool in inference problems in situations in which inference does not necessarily require exact marginals (see, for instance, the near Shannon-limit performance of BP in turbo codes) and because of ease of implementation in distributive settings.

GaBP is BP applied to a pairwise MG constructed from a multivariate Gaussian distribution in canonical parameterization. GaBP can be used to approximate the univariate marginals of the variables represented by the MG. GaBP is guaranteed to converge if the pairwise MG forms a tree structure, but does not necessarily converge for loopy pairwise MGs. The convergence behavior of GaBP has been the subject of research in the literature (Weiss and Freeman, 2001; Malioutov et al., 2006; Su and Wu, 2015; Sui et al., 2015). Weiss and Freeman (2001) interpret the computations done by GaBP as inference on a tree-structured graph. This interpretation is used to show that univariate GaBP will converge if the precision matrix is diagonally dominant and that the marginal means supplied by GaBP, assuming convergence, are exact. We now define two important concepts related to the convergence of GaBP:

Definition 1 (Walk-summability) *Consider a precision matrix* $\mathbf{S} : k \times k = [s_{ij}]$, *and suppose* $\mathbf{D} = diag(\frac{1}{\sqrt{s_{11}}}, \frac{1}{\sqrt{s_{22}}}, \ldots, \frac{1}{\sqrt{s_{kk}}})$. *The matrix* $\mathbf{S}$ *is considered to be walk-summable if the spectral radius of* $|\mathbf{I}_k - \mathbf{DSD}|$ *is less than one.*

Definition 2 (Scaled diagonal dominance) *A precision matrix* $\mathbf{S}$ *is called scaled diagonally dominant if there exists a diagonal matrix* $\mathbf{D}$, *with only positive diagonal entries, such that* $\mathbf{DSD}$ *is strictly diagonally dominant.*

If $\mathbf{A} = [a_{ij}]$, then by $|\mathbf{A}|$ we mean $[|a_{ij}|]$. Malioutov et al. (2006) show that GaBP will converge if the precision matrix is walk-summable, while Moallemi and Van Roy (2009, 2010) show the same for scaled diagonal dominance. In fact, these two conditions are known to be equivalent (Malioutov, 2008; Ruozzi et al., 2009). Su and Wu (2015) derive necessary and sufficient conditions for the convergence of univariate GaBP under a specified initialization set. They also derive necessary and sufficient conditions for damped univariate GaBP, with an allowable interval for the damping factor.

Although technically a marginalization algorithm, GaBP has been applied to solve large and sparse systems of linear equations (Bickson, 2008; Shental et al., 2008; El-Kurdi et al., 2012b). GaBP has also been applied in areas such as channel estimation in communication systems (Montanari et al., 2006; Guo and Ping, 2008; Guo and Huang, 2011), sparse Bayesian learning in large-scale compressed sensing problems (Seeger and Wipf, 2010), estimation on Gaussian graphical models (Chandrasekaran et al., 2008; Liu et al., 2012) and the detection of F-formations in free-standing conversational groups (Kamper, 2017).

©2019 Francois Kamper, Sarel J. Steel, and Johan A. du Preez.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-040.html.

Several variants of GaBP have been proposed in the literature to combat the convergence issues of the basic algorithm in loopy graphs (Johnson et al., 2009; Liu, 2010; El-Kurdi et al., 2012a; Ruozzi and Tatikonda, 2013; Liu et al., 2012; Kamper et al., 2018). Johnson et al. (2009) employ diagonal loadings on the precision matrix and iteratively apply GaBP to compute the correct marginal means. A feedback node refers to a node which, when removed, results in a cycle-free graph. The role of (pseudo) feedback nodes in improving the convergence of basic GaBP has been studied by Liu (2010) and Liu et al. (2012). El-Kurdi et al. (2012a) use a relaxation factor to accelerate GaBP in cases where the precision components of the basic algorithm converge, while the use of a reweighted min-sum algorithm has been studied by Ruozzi and Tatikonda (2013). Kamper et al. (2018) specify a regularization scheme on GaBP that guarantees convergence to the exact marginal means, given sufficient regularization. They also show empirically that this regularization scheme can improve the accuracy of the precisions supplied by GaBP as approximations for the exact marginal precisions.

In this paper we consider a multivariate extension of GaBP that can be used to approximate the marginals of $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p$. For this purpose, we consider a higher-dimensional generalization of a pairwise MG, where each node is allowed to receive more than one random variable. We are not aware of any research that explicitly studies the use of a higher-dimensional MG to approximate higher-dimensional marginals, although related work, on distributed state estimation using Gaussian message-passing, can be found in the literature (Sui et al., 2015). The main contribution of this paper is the development of a sufficient condition for the convergence of GaBP-m. This condition, labeled preconditioned walk-summability, is a multivariate extension of the walk-summability condition for the convergence of GaBP. Under the assumption of preconditioned walk-summability, we derive a bound for the number of iterations required for convergence by GaBP-m. We also establish a bound on the dissimilarity between the approximate and exact marginal covariance matrices. Another important advantage of GaBP-m is its robustness with regard to a certain scaling of the precision matrix and potential vector. This robustness is special, since it can have a substantial effect on the performance of iterative solvers of linear systems such as the CG solver.

The main advantage of GaBP-m over GaBP is in a marginalization context, since GaBP-m can be used to approximate the precision matrix of higher-dimensional marginals. Other advantages include:

1. GaBP-m can converge in cases where GaBP does not.

2. GaBP-m can converge in fewer iterations than GaBP.

3. GaBP-m can improve the accuracy of the approximate univariate marginals. This can be done by directly computing the univariate marginals from the approximated higher-dimensional marginals.

The advantage of using GaBP over GaBP-m is that the message updates associated with each iteration are computationally less expensive.

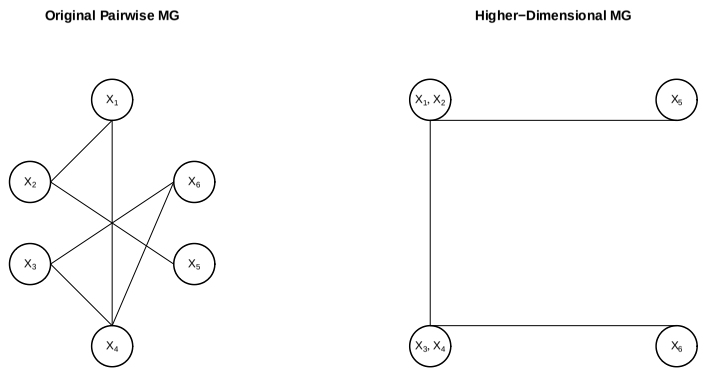


Figure 1: Example of the conversion of a pairwise MG to a higher-dimensional MG. We have six variables $(X_1, X_2, \ldots, X_6)$ clustered into $\mathcal{C}_1 = \{1, 2\}$, $\mathcal{C}_2 = \{3, 4\}$, $\mathcal{C}_3 = \{5\}$ and $\mathcal{C}_4 = \{6\}$. The original pairwise MG is given to the left, while the higher-dimensional extension is given to the right.

2. Preliminaries

In this section we discuss the mathematical machinery that is used in this paper.

2.1. Higher-dimensional MG

We assume, without loss of generality, that $\mathbf{X} = (\mathbf{X}_1', \mathbf{X}_2', \ldots, \mathbf{X}_p')'$. Let $\mathcal{C}_i$ be the set containing all the variables in $\mathbf{X}_i$. We sometimes refer to $\mathcal{C}_i$ as cluster i. Define $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ to be the pairwise MG for $\mathbf{X}$, where $\tilde{\mathcal{V}}$ and $\tilde{\mathcal{E}}$ denote the set of nodes and the set of edges respectively. Note that, when $(s, t) \in \tilde{\mathcal{E}}$, we place the restriction that $s < t$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the higher-dimensional MG obtained from the clusters $\mathcal{C}_i : i = 1, 2, \ldots, p$. Note that $\mathcal{V}$ will contain a node for each $\mathcal{C}_i : i = 1, 2, \ldots, p$. The pair $(i, j) \in \mathcal{E}$, $i < j$, if and only if there is an $s \in \mathcal{C}_i$ and a $t \in \mathcal{C}_j$ such that either (s, t) or (t, s) is in $\tilde{\mathcal{E}}$.

We illustrate this procedure in Figure 1. We consider a pairwise MG drawn for $\mathbf{X} = (X_1, X_2, X_3, X_4, X_5, X_6)'$ where we perform the clustering $\mathcal{C}_1 = \{1, 2\}$, $\mathcal{C}_2 = \{3, 4\}$, $\mathcal{C}_3 = \{5\}$ and $\mathcal{C}_4 = \{6\}$. From this point onwards, we refer to the higher-dimensional extension of the pairwise MG just as MG.

Algorithm 1 Synchronous GaBP-m

1. Provide a precision matrix $\mathbf{S} : k \times k$, a potential vector $\boldsymbol{b} : k \times 1$, and clusters $\mathcal{C}_i : i = 1, 2, \ldots, p$ as inputs.
2. Specify a tolerance ϵ and a maximum number of iterations m.
3. Initialize $\mathbf{Q}_{ij}^{(0)} = \mathbf{0} : d_j \times d_j$ and $\mathbf{v}_{ij}^{(0)} = \mathbf{0} : d_j \times 1$ for all i and all $j \in \mathcal{N}_i$.
4. Set Err = Inf and $n = 0$.
5. While Err $> \epsilon$
   (a) Compute $\mathbf{P}_i^{(n)} = \mathbf{S}_{ii} + \sum_{j \in \mathcal{N}_i} \mathbf{Q}_{ji}^{(n)}$ and $\mathbf{z}_i^{(n)} = \boldsymbol{b}_i + \sum_{j \in \mathcal{N}_i} \mathbf{v}_{ji}^{(n)}$ for $i = 1, 2, \ldots, p$.
   (b) Set $\boldsymbol{\mu}_i^{(n)} = [\mathbf{P}_i^{(n)}]^{-1} \mathbf{z}_i^{(n)}$, $\boldsymbol{e}_i^{(n)} = \sum_j \mathbf{S}_{ij} \boldsymbol{\mu}_j^{(n)} - \boldsymbol{b}_i$ and Err $= \max_i \{ ||\boldsymbol{e}_i^{(n)}||_\infty \}$.
   (c) If Err $> \epsilon$, do for all $i \in \{1, 2, \ldots, p\}$ and all $j \in \mathcal{N}_i$:
   $\mathbf{Q}_{ij}^{(n+1)} = -\mathbf{S}_{ji} [\mathbf{P}_i^{(n)} - \mathbf{Q}_{ji}^{(n)}]^{-1} \mathbf{S}_{ij}$ and
   $\mathbf{v}_{ij}^{(n+1)} = -\mathbf{S}_{ji} [\mathbf{P}_i^{(n)} - \mathbf{Q}_{ji}^{(n)}]^{-1} [\mathbf{z}_i^{(n)} - \mathbf{v}_{ji}^{(n)}]$.
   (d) Increment n.
   (e) If $n = m$, break.
6. End.

2.2. Factorization of the Gaussian Density

We assume that $\mathbf{X}$ follows a multivariate Gaussian distribution with precision matrix $\mathbf{S}$ and potential vector $\boldsymbol{b}$. By $\mathbf{S}_{ij}$ we refer to the submatrix of $\mathbf{S}$ corresponding to the variables in $\mathcal{C}_i$ and $\mathcal{C}_j$ for the rows and columns respectively. The vector $\boldsymbol{b}_i$ refers to the subvector of $\boldsymbol{b}$ corresponding to the variables in $\mathcal{C}_i$.

Note that we can factor the Gaussian density according to the MG:

$$f(\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{p} \phi_i(\mathbf{x}_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j), \tag{1}$$

where $\phi_i(\mathbf{x}_i) = \exp\Big[-\frac{1}{2} \mathbf{x}_i' \mathbf{S}_{ii} \mathbf{x}_i + \boldsymbol{x}_i' \boldsymbol{b}_i \Big]$, $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \exp\Big[-\mathbf{x}_i' \mathbf{S}_{ij} \mathbf{x}_j \Big]$, Z is a normalization constant, $\mathbf{x} = (\mathbf{x}_1', \mathbf{x}_2', \ldots, \mathbf{x}_p')'$ follows the decomposition $\mathbf{X} = (\mathbf{X}_1', \mathbf{X}_2', \ldots, \mathbf{X}_p')'$, and $\mathcal{E} = \{(i,j) : i < j \text{ and } \mathbf{S}_{ij} \neq \mathbf{0} : d_i \times d_j\}$.

2.3. Derivation of GaBP-m

We restrict our focus to a synchronous sum-product derivation of GaBP-m. Define $\mathcal{N}_i = \{j : (i,j) \text{ or } (j,i) \in \mathcal{E}\}$ to denote the neighborhood of cluster i. In the Gaussian case, we have $\mathcal{N}_i = \{j \neq i : \mathbf{S}_{ij} \neq \mathbf{0} : d_i \times d_j\}$. Suppose that $j \in \mathcal{N}_i$, then by $\mathcal{N}_i \setminus j$ we mean the set $\mathcal{N}_i$ with cluster j removed. Based on Equation (1), we formulate the message updates

of GaBP-m analogous to the message updates of BP applied to a pairwise MG:

$$m_{ij}^{(n+1)}(\mathbf{x}_j) = \int_{\mathbf{x}_i} \phi_i(\mathbf{x}_i) \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_{t \in \mathcal{N}_i \setminus j} m_{ti}^{(n)}(\mathbf{x}_i) d\mathbf{x}_i, \tag{2}$$

for all clusters i and all $j \in \mathcal{N}_i$. Note that when $i > j$ we use $\psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \psi_{ji}(\mathbf{x}_j, \mathbf{x}_i)$. For the Gaussian case, Equation (2) reduces to:

$$m_{ij}^{(n+1)}(\mathbf{x}_j) = \int_{\mathbf{x}_i} \exp\Big[-\frac{1}{2} \mathbf{x}_i' \mathbf{S}_{ii} \mathbf{x}_i + \mathbf{x}_i' \boldsymbol{b}_i \Big] \exp\Big[-\mathbf{x}_i' \mathbf{S}_{ij} \mathbf{x}_j \Big] \prod_{t \in \mathcal{N}_i \setminus j} m_{ti}^{(n)}(\mathbf{x}_i) d\mathbf{x}_i. \tag{3}$$

We now make the assumption that $m_{ti}^{(n)}(\mathbf{x}_i) \propto \exp\Big[-\frac{1}{2} \mathbf{x}_i' \mathbf{Q}_{ti}^{(n)} \mathbf{x}_i + \mathbf{x}_i' \mathbf{v}_{ti}^{(n)} \Big]$ for all $t \in \mathcal{N}_i$. Substitution into Equation (3) yields:

$$m_{ij}^{(n+1)}(\mathbf{x}_j) \propto \exp\Big[-\frac{1}{2} \mathbf{x}_j' \mathbf{Q}_{ij}^{(n+1)} \mathbf{x}_j + \mathbf{x}_j' \mathbf{v}_{ij}^{(n+1)} \Big], \tag{4}$$

where

$$\mathbf{Q}_{ij}^{(n+1)} = -\mathbf{S}_{ji} [\mathbf{P}_{ij}^{(n)}]^{-1} \mathbf{S}_{ij} \tag{5}$$

$$\mathbf{v}_{ij}^{(n+1)} = -\mathbf{S}_{ji} [\mathbf{P}_{ij}^{(n)}]^{-1} [\boldsymbol{b}_i + \sum_{t \in \mathcal{N}_i \setminus j} \mathbf{v}_{ti}^{(n)}] \tag{6}$$

$$\mathbf{P}_{ij}^{(n)} = \mathbf{S}_{ii} + \sum_{t \in \mathcal{N}_i \setminus j} \mathbf{Q}_{ti}^{(n)}. \tag{7}$$

Note that we used the following integral in the derivation of Equation (4):

$$\int_{\mathbf{y}} \exp\Big(-\frac{1}{2} \mathbf{y}' \mathbf{A} \mathbf{y} + \mathbf{y}' \boldsymbol{a} \Big) d\mathbf{y} = (2\pi)^{\frac{m}{2}} (\det(\mathbf{A}))^{-\frac{1}{2}} \exp\Big(\frac{1}{2} \boldsymbol{a}' \mathbf{A}^{-1} \boldsymbol{a} \Big),$$

where $\mathbf{A}$ is positive definite. The message updates in Equations (5) and (6) are performed efficiently in Algorithm 1. At each iteration of GaBP-m, we can construct an approximate marginal for cluster i by collecting all incoming messages from its neighborhood:

$$\hat{f}_i^{(n)}(\mathbf{x}_i) = \phi_i(\mathbf{x}_i) \prod_{t \in \mathcal{N}_i} m_{ti}^{(n)}(\mathbf{x}_i) \propto \exp\Big[-\frac{1}{2} \mathbf{x}_i' \mathbf{P}_i^{(n)} \mathbf{x}_i + \mathbf{x}_i' \mathbf{z}_i^{(n)} \Big], \tag{8}$$

where $\mathbf{P}_i^{(n)} = \mathbf{S}_{ii} + \sum_{j \in \mathcal{N}_i} \mathbf{Q}_{ji}^{(n)}$ and $\mathbf{z}_i^{(n)} = \boldsymbol{b}_i + \sum_{j \in \mathcal{N}_i} \mathbf{v}_{ji}^{(n)}$. Clearly, the density given in Equation (8) is a Gaussian density with precision $\mathbf{P}_i^{(n)}$ and mean $\boldsymbol{\mu}_i^{(n)} = [\mathbf{P}_i^{(n)}]^{-1} \mathbf{z}_i^{(n)}$. We refer to $\hat{f}_i^{(n)}(\mathbf{x}_i)$ as the posterior distribution of cluster i at iteration n. The quantities $\mathbf{P}_i^{(n)}$ and $\boldsymbol{\mu}_i^{(n)}$ are labeled the posterior precision and the posterior mean associated with cluster i at iteration n respectively.

2.4. Notes on the Perron-Frobenius Theorem

In this section we discuss results related to the Perron-Frobenius theorem. This theorem plays an important role in the theoretical considerations of this paper.

Theorem 1 (Perron-Frobenius theorem) *Consider a non-negative and irreducible real matrix* $\mathbf{A} : k \times k$. *There exists a* $\mathbf{v} > \mathbf{0}$ *such that* $\mathbf{A}\mathbf{v} = \rho(\mathbf{A})\mathbf{v}$, *where* $\rho(\mathbf{A})$ *denotes the spectral radius of* $\mathbf{A}$.

We use the notation $\mathbf{x} > \mathbf{y}$ to indicate that each component of $\mathbf{x}$ is greater than the corresponding component of $\mathbf{y}$. The following important lemma follows from the Collatz-Wielandt formula.

Lemma 1 *Let* $\rho(.)$ *be the spectral radius of a matrix. For any real matrix* $\mathbf{A} : k \times k$ *we have* $\rho(\mathbf{A}) \leq \rho(|\mathbf{A}|)$.

For the remainder of this paper we assume, without loss of generality, that GaBP-m is applied to an irreducible MG. If this assumption does not hold, then GaBP-m can be applied separately to irreducible subgraphs of the MG.

2.5. Preconditioned Walk-summability

The main theoretical result of this paper is the development of a sufficient condition for the convergence of GaBP-m. We label this condition preconditioned walk-summability, since it involves a certain preconditioning of the precision matrix based on how the clusters are chosen. We give the following definitions:

Definition 3 (Valid Preconditioner) *We call a matrix* $\mathbf{\Lambda} : k \times k$ *a valid preconditioner with respect to the clusters* $\mathcal{C}_i : i = 1, 2, \ldots, p$ *if* $\mathbf{\Lambda}_{ii}$ *is positive definite and* $\mathbf{\Lambda}_{ij} = \mathbf{0} : d_i \times d_j$.

Definition 4 (Preconditioned Walk-summability) *Consider a precision matrix* $\mathbf{S} : k \times k$ *and clusters* $\mathcal{C}_i : i = 1, 2, \ldots, p$. *The precision matrix* $\mathbf{S}$ *is preconditioned walk-summable if there exists a valid preconditioner* $\mathbf{\Lambda}$ *such that* $\mathbf{\Lambda}\mathbf{S}\mathbf{\Lambda}$ *is walk-summable.*

We note that preconditioned walk-summability is novel with respect to walk-summability and scaled diagonal dominance in the sense that the preconditioner does not need to be diagonal. In fact, the larger the chosen clusters, the wider the range of valid preconditioners. In the empirical section we give an explicit example where GaBP fails to converge on a non- (but preconditioned) walk-summable precision matrix.

2.6. Computation Trees

A computation tree is a representation of loopy belief propagation (LBP) in terms of inference on a tree-structured graph. Introduced by Weiss and Freeman (2001), it has been widely used in the study of the convergence behavior of LBP (Malioutov et al., 2006; Moallemi and Van Roy, 2009, 2010; Ruozzi and Tatikonda, 2013; Sui et al., 2015).

To generate a computation tree for GaBP-m, we start by clustering the k variables into p clusters. We then construct the higher-dimensional MG, which is used to construct the

topology of the computation tree exactly as we would have in the univariate case.

Our discussion of the process to generate a computation tree from an MG closely follows the discussion of Malioutov et al. (2006). We restrict our focus to computation trees for synchronous message-passing. Each message $m_{ij}^{(n-1)}(.)$ receives a computation tree $\mathcal{T}_{ij}^{(n)}$ (the superscript of the tree represents the depth of the tree rather than the iteration number). To construct $\mathcal{T}_{ij}^{(n)}$, we join all the trees $\mathcal{T}_{ki}^{(n-1)}$, for $k \in \mathcal{N}_i \setminus j$, at their common root node (this node refers to cluster i). We then add a new root node (with a reference to cluster j) and an edge to the previous root node. The computation tree for cluster i, $\mathcal{T}_i^{(n)}$, is constructed by joining all trees $\mathcal{T}_{ki}^{(n-1)}$, for $k \in \mathcal{N}_i$, at their common root node. The trees $\mathcal{T}_{ij}^{(1)}$ consist of a single node with reference to cluster j. An example of this process is given in Figure 2. Note that each node in the computation tree has a reference to a specific cluster.

The above process shows that a (synchronous) computation tree has a natural division into different layers. This allows a further conversion of the graph structure. All nodes in a given layer of the computation tree are collected into a single node. This leads to a graph in the form of a line topology (right-hand side of Figure 2). The line topology of $\mathcal{T}_i^{(n)}$ is labeled $\mathcal{L}_i^{(n)}$. This conversion will be used in the derivation of preconditioned walk-summability as a sufficient condition for convergence.

Each computation tree has an associated precision matrix and potential vector. The sparsity structure of the precision matrix follows the topology of the computation tree. A node in the computation tree with a reference to cluster i receives $\mathbf{S}_{ii}$ and $\boldsymbol{b}_i$ as a precision matrix and potential vector respectively. An edge between two nodes in the computation tree receives $\mathbf{S}_{ij}$ as its precision matrix, assuming the references of the nodes are i and j respectively.

The next section offers a proof that the posterior precision and posterior mean for node i of synchronous GaBP-m at iteration $n-1$ can be obtained by performing inference on the tree $\mathcal{T}_i^{(n)}$.

3. GaBP-m as Inference on a Computation Tree

In this section we discuss the interpretation of GaBP-m as inference on a computation tree and the effect of a certain change in the input variables on this algorithm. The main results of this section are:

1. A proof that the posterior mean and posterior precision of a cluster can be obtained by marginalizing a computation tree.
2. That there is a simple way of moving between the output of GaBP-m before and after a certain change in variables is introduced. This suggests that the performance of GaBP-m is robust towards this change in variables.

We focus, without loss of generality, on the computation tree associated with node 1. We denote the computation tree and line topology for node 1 by $\mathcal{T}_n$ and $\mathcal{L}_n$ respectively. Associate

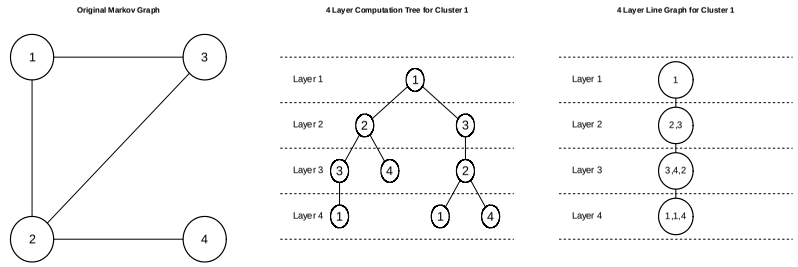


Figure 2: Different graph representations of an MG. The original MG is displayed to the left, the computation tree for cluster 1 is displayed in the middle, while the line graph representation of the same cluster is displayed to the right. Each node (in all three graphs) contains references to certain clusters.

with $\mathcal{T}_n$ the precision matrix $\mathbf{T}_n : m_n \times m_n$ and potential vector $\mathbf{t}_n : m_n \times 1$. Marginalizing $\mathcal{T}_n$ involves computing $\mathbf{T}_n^{-1}$ and $\mathbf{T}_n^{-1}\mathbf{t}_n$. We want to show that $\mathbf{P}_1^{(n-1)}$ and $\boldsymbol{\mu}_1^{(n-1)}$ can be obtained by extracting the diagonal block and subvector corresponding to cluster 1 of $\mathbf{T}_n^{-1}$ and $\mathbf{T}_n^{-1}\mathbf{t}_n$ respectively. We start by defining a few matrices for convenience. Refer to Appendix C for a running example of some of the considerations in this section.

3.1. Matrix Definitions

A movement along a computation tree is defined to be first by layer, starting at layer 1, and within each layer from left to right. Unless explicitly stated otherwise, we assume that the order of the nodes in the precision matrix and potential vector associated with $\mathcal{T}_n$ is according to the movement along $\mathcal{T}_n$.

Definition 5 (Row-extractor matrix of cluster t) *Define*

$$\mathbf{F}_t : d_t \times k = \begin{bmatrix} \mathbf{0} : d_t \times d_1 & \dots & \mathbf{0} : d_t \times d_{t-1} & \mathbf{I}_{d_t} & \mathbf{0} : d_t \times d_{t+1} & \dots & \mathbf{0} : d_t \times d_p \end{bmatrix}.$$

Recalling the ordering of our variables, we see that $\mathbf{F}_t\mathbf{S}$ extracts the rows of $\mathbf{S}$ corresponding to the variables in $\mathcal{C}_t$. Also note that:

$$\mathbf{S}_{jt}\mathbf{F}_t = \begin{bmatrix} \mathbf{0} : d_j \times d_1 & \dots & \mathbf{0} : d_j \times d_{t-1} & \mathbf{S}_{jt} & \mathbf{0} : d_j \times d_{t+1} & \dots & \mathbf{0} : d_j \times d_p \end{bmatrix}.$$

Definition 6 (Row-extractor matrix) *The row-extractor matrix $\mathbf{E}_n : m_n \times p$ is obtained by moving along the computation tree, noting the cluster reference of each node and stacking the row-extractor matrix of this cluster.*

Since $\mathbf{F}_t\boldsymbol{b} = \boldsymbol{b}_t$, we see that:

$$\mathbf{t}_n = \mathbf{E}_n\boldsymbol{b}.$$

Definition 7 (Node-extractor matrix) *Consider node j of the computation tree (read from left to right along the layers). Let $\mathbf{G}_n^{(j)}$ be such that $[\mathbf{G}_n^{(j)}]'\mathbf{T}_n$ extracts the rows of $\mathbf{T}_n$ corresponding to node j. Define the root extractor matrix as $\mathbf{G}_n = \mathbf{G}_n^{(1)}$.*

Definition 8 (Sparse replicate of cluster t) *A sparse replicate of cluster t is obtained by taking $\mathbf{F}_t\mathbf{S} = \begin{bmatrix} \mathbf{S}_{t1} & \dots & \mathbf{S}_{tt} & \dots & \mathbf{S}_{tp} \end{bmatrix}$ and setting $\mathbf{S}_{tj} \leftarrow \mathbf{0} : d_t \times d_j$ for all j in a non-empty subset of $\mathcal{N}_t$.*

Note that the matrix $\mathbf{T}_n\mathbf{E}_n$ has a row-block decomposition according to the computation tree. This means that the jth row block of $\mathbf{T}_n\mathbf{E}_n$ corresponds to the jth node of the computation tree. We now consider the following lemma.

Lemma 2 *We consider the following two scenarios for a node in the computation tree with reference to cluster t:*

1. *If this node is not in the final layer, then the corresponding row block of $\mathbf{T}_n\mathbf{E}_n$ is $\mathbf{F}_t\mathbf{S}$.*
2. *If this node is in the final layer, then the corresponding row block of $\mathbf{T}_n\mathbf{E}_n$ is a sparse replicate of cluster t.*

Proof Note that $\mathbf{T}_n\mathbf{E}_n$ has a row-block decomposition according to its computation tree, i.e. the jth row-block corresponds to the jth node in the computation tree. Consider node j of the computation tree. Note that we can extract the jth row block of $\mathbf{T}_n\mathbf{E}_n$ by computing $[\mathbf{G}_n^{(j)}]'\mathbf{T}_n\mathbf{E}_n$. We use the notation $I(j)$ to denote the cluster to which node j has a reference. Suppose that $I(j) = t$. Let O_j denote the neighborhood of node j in the computation tree, and let $J(O_j) = \{I(s) : s \in O_j\}$. We have that:

$$[\mathbf{G}_n^{(j)}]'\mathbf{T}_n\mathbf{E}_n = \mathbf{S}_{tt}\mathbf{F}_t + \sum_{k \in J(O_j)} \mathbf{S}_{tk}\mathbf{F}_k.$$

The proof follows directly from the fact that $J(O_j) = \mathcal{N}_t$, if j is not in the terminal layer, and $J(O_j) \subset \mathcal{N}_t$ otherwise. ■

A consequence of Lemma 2 is given in the following corollary.

Corollary 1 *We have:*

$$\mathbf{T}_n\mathbf{E}_n = \mathbf{E}_n\mathbf{S} + \mathbf{L}_n, \tag{9}$$

where the first m_{n-1} rows of $\mathbf{L}_n$ contain only zero entries.

We note that Lemma 2 and Corollary 1 do not only apply for computation trees of symmetric matrices, but also to computation trees of general non-symmetric matrices. We use this fact in some of the proofs contained in this paper.

The remainder of this section is dedicated to showing the validity of the following formulas:

$$\mathbf{P}_1^{(n-1)} = [\mathbf{G}_n' \mathbf{T}_n^{-1} \mathbf{G}_n]^{-1} \tag{10}$$

$$\boldsymbol{\mu}_1^{(n-1)} = \mathbf{G}_n' \mathbf{T}_n^{-1} \mathbf{E}_n \boldsymbol{b}. \tag{11}$$

We first discuss a tree-pruning procedure that is aimed at removing terminal nodes of the computation tree in such a way that inference at the root node remains unchanged. This is followed by a discussion of how the tree-pruning procedure can be used to prove formulas (10) and (11).

3.2. Tree-pruning Procedure

Consider a general tree $\mathcal{T}$ with precision matrix $\mathbf{T}$ and potential vector $\mathbf{t}$. Suppose that $r_\mu(\mathcal{T})$ and $r_P(\mathcal{T})$ give the marginal mean and marginal precision respectively of the root node of $\mathcal{T}$. We can obtain $r_P(\mathcal{T})$ by inverting the diagonal block of $\mathbf{T}^{-1}$ corresponding to the root node. In a similar way, $r_\mu(\mathcal{T})$ is obtained by extracting the first subvector (corresponding to the root node) of $\mathbf{T}^{-1}\mathbf{t}$.

The tree-pruning procedure iteratively prunes terminal nodes such that the marginal precision matrix and marginal mean vector of the root node of the remaining (trimmed) tree remain unchanged. Let $\tilde{\mathcal{T}}$ be the trimmed tree obtained after eliminating certain terminal nodes from $\mathcal{T}$, then we must have that $r_\mu(\mathcal{T}) = r_\mu(\tilde{\mathcal{T}})$ and $r_P(\mathcal{T}) = r_P(\tilde{\mathcal{T}})$.

At each step of the pruning process, we select one of the nodes in the penultimate layer (we call this node the bereaved parent), the objective being to prune its children in the terminal layer. This must be done in such a way that inference at the root node remains unchanged. In order to achieve this, we need to change certain elements of the precision and potential associated with the nodes in the trimmed tree from their corresponding values in the untrimmed tree. In the remainder of this section, we show that it is only necessary to change the precision matrix and potential vector associated with the bereaved parent. Moreover, these adjustments can be obtained by marginalizing the tree with the bereaved parent as the root node.

Let $\mathbf{T}$ and $\mathbf{t}$ be the precision matrix and potential vector respectively of $\mathcal{T}$. We can write, without loss of generality:

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{0} \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23} \\ \mathbf{0} & \mathbf{T}_{32} & \mathbf{T}_{33} \end{bmatrix}$$

$$\mathbf{t} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{bmatrix}.$$

In general, we associate $\mathbf{t}_3$ with the nodes to be pruned, $\mathbf{t}_2$ with the bereaved parent and $\mathbf{t}_1$ with the remaining nodes. A similar interpretation holds for the precision matrix. Consider

$$\mathbf{T} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

$$\mathbf{t} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix},$$

where $\mathbf{A}_{11} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix}$, $\mathbf{A}_{12} = \begin{bmatrix} \mathbf{0} \\ \mathbf{T}_{23} \end{bmatrix}$, $\mathbf{A}_{21} = \begin{bmatrix} \mathbf{0} & \mathbf{T}_{32} \end{bmatrix}$, $\mathbf{A}_{22} = \mathbf{T}_{33}$, $\mathbf{a}_1 = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix}$ and $\mathbf{a}_2 = \mathbf{t}_3$. Block-wise matrix inversion yields

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{A}_{11.2}^{-1} & -\mathbf{A}_{11.2}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ \dots & \dots \end{bmatrix},$$

where $\mathbf{A}_{11.2} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$ and the $\dots$ represents an irrelevant part of the matrix (nodes to be pruned). Let $\tilde{\mathcal{T}}$ denote the trimmed tree. In order to preserve root inference, the precision matrix and potential vector associated with $\tilde{\mathcal{T}}$ must be $\tilde{\mathbf{T}} = \mathbf{A}_{11.2}$ and $\tilde{\mathbf{t}} = \mathbf{a}_1 - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{a}_2$ respectively. It can be shown that:

$$\mathbf{A}_{11.2} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} - \mathbf{T}_{23}\mathbf{T}_{33}^{-1}\mathbf{T}_{32} \end{bmatrix} \tag{12}$$

$$\mathbf{a}_1 - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{a}_2 = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 - \mathbf{T}_{23}\mathbf{T}_{33}^{-1}\mathbf{t}_3 \end{bmatrix}. \tag{13}$$

Equations (12) and (13) show that it is only necessary to adjust the precision and potential associated with the bereaved parent. Thereafter, we can prune the terminal nodes.

Let $\mathcal{M}$ be the tree containing the nodes to be pruned, with the bereaved parent as the root node. The precision and potential associated with $\mathcal{M}$ are $\mathbf{M} = \begin{bmatrix} \mathbf{T}_{22} & \mathbf{T}_{23} \\ \mathbf{T}_{32} & \mathbf{T}_{33} \end{bmatrix}$ and $\mathbf{m} = \begin{bmatrix} \mathbf{t}_2 \\ \mathbf{t}_3 \end{bmatrix}$ respectively. Setting $\mathbf{T}_{22.3} = \mathbf{T}_{22} - \mathbf{T}_{23}\mathbf{T}_{33}^{-1}\mathbf{T}_{32}$, we see, using block-wise matrix inversion, that:

$$r_P(\mathcal{M}) = \mathbf{T}_{22.3}$$

$$r_\mu(\mathcal{M}) = \mathbf{T}_{22.3}^{-1}[\mathbf{t}_2 - \mathbf{T}_{23}\mathbf{T}_{33}^{-1}\mathbf{t}_3].$$

In summary, to prune terminal nodes while preserving the marginal quantities of the root node, we need to do the following:

1. Compute $r_\mu(\mathcal{M})$ and $r_P(\mathcal{M})$.
2. Obtain the trimmed tree by eliminating the terminal nodes from the graph and only adjusting the potential and the precision of the bereaved parent to $r_P(\mathcal{M})r_\mu(\mathcal{M})$ and $r_P(\mathcal{M})$ respectively.

We now prove computation tree correctness by showing that the computations done by GaBP-m can be represented by the pruning procedure described in this section.

3.3. Computation Tree Correctness

By computation tree correctness we mean the validity of the formulas given in (10) and (11). In order to establish this, recall that $\mathbf{P}_{ij}^{(l)} = \mathbf{S}_{ii} + \sum_{t\in\mathcal{N}_i\setminus j} \mathbf{Q}_{ti}^{(l)}$ and define

$$\mathbf{z}_{ij}^{(l)} = \boldsymbol{b}_i + \sum_{t\in\mathcal{N}_i\setminus j} \mathbf{v}_{ti}^{(l)}$$
$$\boldsymbol{\mu}_{ij}^{(l)} = [\mathbf{P}_{ij}^{(l)}]^{-1}\mathbf{z}_{ij}^{(l)}.$$

Consider the following recursive expansions (recall Equations (5) to (7)):

$$\mathbf{P}_{ij}^{(l)} = \mathbf{S}_{ii} - \sum_{t\in\mathcal{N}_i\setminus j} \mathbf{S}_{it}[\mathbf{P}_{ti}^{(l-1)}]^{-1}\mathbf{S}_{it}$$
$$\mathbf{z}_{ij}^{(l)} = \boldsymbol{b}_i - \sum_{t\in\mathcal{N}_i\setminus j} \mathbf{S}_{it}[\mathbf{P}_{ti}^{(l-1)}]^{-1}\mathbf{z}_{ti}^{(l-1)}.$$

Suppose we have pruned $\mathcal{T}_n$ to have only l layers, where $3 \le l \le n$. Consider a bereaved parent, in layer $l-1$. We assume that the bereaved parent has a reference to cluster i while its parent (in layer $l-2$) has a reference to cluster j. Due to the construction of $\mathcal{T}_n$, we know that the bereaved parent will have mutually unconnected terminal nodes as children, each with a reference to a different cluster in $\mathcal{N}_i\setminus j$. Let $|\mathcal{N}_i\setminus j| = s_{ij}$ and $\mathcal{N}_i\setminus j = \{i_1, i_2, \ldots, i_{s_{ij}}\}$, and suppose that $\mathcal{M}$ is the tree with the selected bereaved parent as root, connected to its children in the terminal layer. We assume that the precision and potential associated with one of these terminal nodes, with a reference to cluster i_t, are $\mathbf{P}_{i_t i}^{(n-l)}$ and $\mathbf{z}_{i_t i}^{(n-l)}$ respectively. If $\mathbf{M}$ and $\mathbf{m}$ denote the precision and potential of $\mathcal{M}$ respectively, we have:

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}_{ii} & \mathbf{S}_{ii_1} & \mathbf{S}_{ii_2} & \ldots & \mathbf{S}_{ii_{s_{ij}}} \\ \mathbf{S}_{i_1 i} & \mathbf{P}_{i_1 i}^{(n-l)} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{S}_{i_2 i} & \mathbf{0} & \mathbf{P}_{i_2 i}^{(n-l)} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}_{i_{s_{ij}} i} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{P}_{i_{s_{ij}} i}^{(n-l)} \end{bmatrix} \tag{14}$$

$$\mathbf{m} = \begin{bmatrix} \boldsymbol{b}_i \\ \mathbf{z}_{i_1 i}^{(n-l)} \\ \mathbf{z}_{i_2 i}^{(n-l)} \\ \vdots \\ \mathbf{z}_{i_{s_{ij}} i}^{(n-l)} \end{bmatrix}. \tag{15}$$

Using block-wise matrix inversion, it is easy to see that:

$$r_P(\mathcal{M}) = \mathbf{S}_{ii} - \sum_{t=1}^{s_{ij}} \mathbf{S}_{ii_t}[\mathbf{P}_{i_t i}^{(n-l)}]^{-1}\mathbf{S}_{i_t i} = \mathbf{P}_{ij}^{(n-l+1)} \tag{16}$$

$$r_\mu(\mathcal{M}) = [\mathbf{P}_{ij}^{(n-l+1)}]^{-1}[\boldsymbol{b}_i - \sum_{t=1}^{s_{ij}} \mathbf{S}_{ii_t}[\mathbf{P}_{i_t i}^{(n-l)}]^{-1}\mathbf{z}_{i_t i}^{(n-l)}] = [\mathbf{P}_{ij}^{(n-l+1)}]^{-1}\mathbf{z}_{ij}^{(n-l+1)}. \tag{17}$$

To prune these terminal nodes, the tree-pruning procedure requires adjusting the precision and potential of the bereaved parent to $\mathbf{P}_{ij}^{(n-l+1)}$ and $\mathbf{z}_{ij}^{(n-l+1)}$ respectively. Once we have pruned all the terminal nodes, we see that we can apply a similar process to the new terminal layer, $l-1$.

Let us consider the case $l = n$, where a terminal node with reference to cluster i_1 and its parent with reference to cluster i are considered. At this stage, the precision and potential associated with this terminal node are $\mathbf{S}_{i_1 i_1}$ and $\boldsymbol{b}_{i_1}$ respectively. Since,

$$\mathbf{Q}_{i_1 i}^{(1)} = -\mathbf{S}_{i_1 i}\mathbf{S}_{i_1 i_1}^{-1}\mathbf{S}_{i_1 i}$$
$$\mathbf{v}_{i_1 i}^{(1)} = -\mathbf{S}_{i_1 i}\mathbf{S}_{i_1 i_1}^{-1}\boldsymbol{b}_{i_1},$$

we see that $\mathbf{P}_{i_1 i}^{(0)} = \mathbf{S}_{i_1 i_1}$ and $\mathbf{z}_{i_1 i}^{(0)} = \boldsymbol{b}_{i_1}$. Hence, the type of pruning described in Equations (14) and (15) apply at all stages of pruning by induction.

Suppose we have completed the pruning such that only two layers remain. The trimmed tree will have a root node corresponding to cluster 1, and this node will be connected to terminal nodes, each of which has a reference to a different cluster in $\mathcal{N}_1$. Let $\mathcal{N}_1 = \{i_1, i_2, \ldots, i_{s_1}\}$. Since we have followed the pruning procedure, the precision matrix and potential vector associated with a terminal node, with reference to cluster i_t, are $\mathbf{z}_{i_t 1}^{(n-2)}$ and $\mathbf{P}_{i_t 1}^{(n-2)}$ respectively. The final step is to find the root marginal of the following precision matrix and potential vector:

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{1i_1} & \mathbf{S}_{1i_2} & \ldots & \mathbf{S}_{1i_{n_1}} \\ \mathbf{S}_{i_1 1} & \mathbf{P}_{i_1 1}^{(n-2)} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{S}_{i_2 1} & \mathbf{0} & \mathbf{P}_{i_2 1}^{(n-2)} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}_{i_{s_1} 1} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{P}_{i_{n_1} 1}^{(n-2)} \end{bmatrix}$$

$$\mathbf{m} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{z}_{i_1 1}^{(n-2)} \\ \mathbf{z}_{i_2 1}^{(n-2)} \\ \vdots \\ \mathbf{z}_{i_{s_1} 1}^{(n-2)} \end{bmatrix}.$$

One more application of block-wise matrix inversion reveals that:

$$r_P(\mathcal{T}_n) = \mathbf{S}_{11} - \sum_{t=1}^{s_1} \mathbf{S}_{1i_t}[\mathbf{P}_{i_t 1}^{(n-2)}]^{-1}\mathbf{S}_{i_t 1} = \mathbf{S}_{11} + \sum_{t=1}^{s_1} \mathbf{Q}_{i_t 1}^{(n-1)} = \mathbf{P}_1^{(n-1)} \tag{18}$$

$$\begin{aligned} r_\mu(\mathcal{T}_n) &= [\mathbf{P}_1^{(n-1)}]^{-1}[\boldsymbol{b}_1 - \sum_{t=1}^{s_1} \mathbf{S}_{1i_t}[\mathbf{P}_{i_t 1}^{(n-2)}]^{-1}\mathbf{z}_{i_t 1}^{(n-2)}] \\ &= [\mathbf{P}_1^{(n-1)}]^{-1}[\boldsymbol{b}_1 + \sum_{t=1}^{s_1} \mathbf{v}_{i_t 1}^{(n-1)}] = \boldsymbol{\mu}_1^{(n-1)}. \end{aligned} \tag{19}$$

Equations (18) and (19) validate the formulas given in (10) and (11). Note that similar formulas can be derived for all clusters i.

We note that GaBP-m applied to a tree-structured MG converges and yields the exact marginals at convergence. The key property is that the computation tree for a cluster in a tree-structured MG eventually replicates the tree itself. Since GaBP-m marginalizes the computation tree exactly, it will also marginalize the tree-structured MG exactly.

3.4. Recovery after Change in Variables

We discuss the relationship between GaBP-m applied to the precision matrix $\mathbf{S}$ and potential vector $\boldsymbol{b}$ before and after a certain change in variables is introduced. We show that the output of GaBP-m applied to the the original inputs can be recovered from the application of GaBP-m to the changed inputs through a simple transformation. This is a key relationship for proving that preconditioned walk-summability is a sufficient condition for convergence.

Consider a valid preconditioner $\boldsymbol{\Lambda}$ relative to the clusters $\mathcal{C}_i : i = 1, 2, \ldots, p$. We consider the relationship between GaBP-m applied to the inputs $\{\mathbf{S}, \boldsymbol{b}\}$ and $\{\tilde{\mathbf{S}}, \tilde{\boldsymbol{b}}\}$ where $\tilde{\mathbf{S}} = \boldsymbol{\Lambda}\mathbf{S}\boldsymbol{\Lambda}$ and $\tilde{\boldsymbol{b}} = \boldsymbol{\Lambda}\boldsymbol{b}$. Consider applying GaBP-m to the inputs $\{\tilde{\mathbf{S}}, \tilde{\boldsymbol{b}}\}$. For cluster i, we denote the posterior mean vector and posterior precision matrix (at iteration $n-1$) as $\tilde{\boldsymbol{\mu}}_i^{(n-1)}$ and $\tilde{\mathbf{P}}_i^{(n-1)}$ respectively. Consider the following lemma:

Lemma 3 *We have the following relationship for all clusters i:*

$$\begin{aligned}\boldsymbol{\mu}_i^{(n-1)} &= \boldsymbol{\Lambda}_{ii}\tilde{\boldsymbol{\mu}}_i^{(n-1)} \\ [\mathbf{P}_i^{(n-1)}]^{-1} &= \boldsymbol{\Lambda}_{ii}[\tilde{\mathbf{P}}_i^{(n-1)}]^{-1}\boldsymbol{\Lambda}_{ii}.\end{aligned}$$

Proof We consider, without loss of generality, the proof for cluster 1. Consider the block-diagonal matrix $\mathbf{B}_n : m_n \times m_n$ with a decomposition according to the computation tree. This matrix is constructed by moving along the computation tree, noting the reference of each node (say i) and adding a symmetric and positive definite matrix $\boldsymbol{\Lambda}_{ii}$ as diagonal blocks of $\mathbf{B}_n$ (all other entries are zero). It can be shown that $\tilde{\mathbf{T}}_n = \mathbf{B}_n\mathbf{T}_n\mathbf{B}_n$ is the precision matrix corresponding to the computation tree (for cluster 1) constructed from $\tilde{\mathbf{S}}$. The following holds:

$$\begin{aligned}\mathbf{P}_1^{(n-1)} &= [\mathbf{G}_n'\mathbf{T}_n^{-1}\mathbf{G}_n]^{-1} \\ &= [\mathbf{G}_n'\mathbf{B}_n\tilde{\mathbf{T}}_n^{-1}\mathbf{B}_n\mathbf{G}_n]^{-1}.\end{aligned} \tag{20}$$

Note that $\mathbf{G}_n'\mathbf{B}_n$ equals the first d_1 rows of $\mathbf{B}_n$; therefore, since $\mathbf{B}_n$ is block-diagonal, $\mathbf{G}_n'\mathbf{B}_n = \boldsymbol{\Lambda}_{11}\mathbf{G}_n'$. Equation (20) becomes

$$\mathbf{P}_1^{(n-1)} = \boldsymbol{\Lambda}_{11}^{-1}\tilde{\mathbf{P}}_1^{(n-1)}\boldsymbol{\Lambda}_{11}^{-1}, \tag{21}$$

where $\tilde{\mathbf{P}}_1^{(n-1)} = [\mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\mathbf{G}_n]^{-1}$. Consider

$$\begin{aligned}\boldsymbol{\mu}_1^{(n-1)} &= \mathbf{G}_n'\mathbf{T}_n^{-1}\mathbf{E}_n\boldsymbol{b} \\ &= \mathbf{G}_n'\mathbf{B}_n\tilde{\mathbf{T}}_n^{-1}\mathbf{B}_n\mathbf{E}_n\boldsymbol{b} \\ &= \boldsymbol{\Lambda}_{11}\mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\mathbf{B}_n\mathbf{E}_n\boldsymbol{b}.\end{aligned}$$

Now, since $\mathbf{B}_n$ is block-diagonal, we must have $\mathbf{B}_n\mathbf{E}_n = \mathbf{E}_n\boldsymbol{\Lambda}$, and therefore

$$\boldsymbol{\mu}_1^{(n-1)} = \boldsymbol{\Lambda}_{11}\mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\mathbf{E}_n\boldsymbol{\Lambda}\boldsymbol{b} = \boldsymbol{\Lambda}_{11}\tilde{\boldsymbol{\mu}}_1^{(n-1)}, \tag{22}$$

where $\tilde{\boldsymbol{\mu}}_1^{(n-1)} = \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\mathbf{E}_n\boldsymbol{\Lambda}\boldsymbol{b}$. ■

Equations (21) and (22) show that there is an easy way of moving between the computations done by GaBP-m on $\{\mathbf{S}, \boldsymbol{b}\}$ and $\{\tilde{\mathbf{S}}, \tilde{\boldsymbol{b}}\}$ that does not depend on the iteration number. These considerations suggest that GaBP-m should be robust towards the change in variables as discussed above.

The way in which $\tilde{\mathbf{S}}$ and $\tilde{\boldsymbol{b}}$ are obtained from $\mathbf{S}$ and $\boldsymbol{b}$ is typically used to precondition inputs to iterative solvers of linear systems, such as the conjugate gradient (CG) method. In the case of the CG method, the difference in the convergence behavior of application to the different sets of inputs can be substantial, in the sense that the preconditioned variant converges much faster (Shewchuk, 1994). Equation (22) shows that the rate of convergence of $\boldsymbol{\mu}_1^{(n)}$ and $\tilde{\boldsymbol{\mu}}_1^{(n)}$ is identical, and that GaBP-m automatically benefits from the preconditioning without having to do this explicitly (convergence is guaranteed under preconditioned walk-summability). For other solvers of linear systems, we need to incorporate the preconditioning explicitly in order to obtain the benefit of a faster convergence rate. These considerations provide theoretical backing for some of the observations made in the empirical section.

4. Convergence of GaBP-m

In this section we prove that preconditioned walk-summability is a sufficient condition for the convergence of GaBP-m. We start by assuming that $\mathbf{S}$ is preconditioned walk-summable, and hence assume the existence of a valid preconditioner $\boldsymbol{\Lambda}$ such that $\tilde{\mathbf{S}} = \boldsymbol{\Lambda}\mathbf{S}\boldsymbol{\Lambda}$ is walk-summable. We then show that GaBP-m applied to $\tilde{\mathbf{S}}$ and $\tilde{\boldsymbol{b}} = \boldsymbol{\Lambda}\boldsymbol{b}$ is guaranteed to converge. The convergence of GaBP-m applied to the original inputs is proved using the results of Section 3.4.

4.1. Convergence of GaBP-m on Preconditioned Inputs

We can assume, without loss of generality, that the diagonal entries of $\tilde{\mathbf{S}}$ are all equal to one (if not, we can incorporate it in the preconditioning). Setting $\tilde{\mathbf{S}} = \mathbf{I}_k - \tilde{\mathbf{R}}$, we see that $\rho(|\tilde{\mathbf{R}}|) < 1$ by assumption. The precision matrix of the computation tree can be expressed as:

$$\tilde{\mathbf{T}}_n = \mathbf{I}_{m_n} - \tilde{\mathbf{R}}_n,$$

where $\tilde{\mathbf{R}}_n$ is a block tri-diagonal matrix corresponding to the line topology of the computation tree. Hence, we can express,

$$\tilde{\mathbf{R}}_n = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & & & & & & & \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{C}_{23} & & & & & & \\ & \mathbf{C}_{32} & \mathbf{C}_{33} & \mathbf{C}_{34} & & & & & \\ & & & \cdots & \cdots & \cdots & & & \\ & & & & \cdots & \cdots & \cdots & & \\ & & & & & \cdots & \cdots & \cdots & \\ & & & & & & \mathbf{C}_{n-1;n-2} & \mathbf{C}_{n-1;n-1} & \mathbf{C}_{n-1;n} \\ & & & & & & & \mathbf{C}_{n;n-1} & \mathbf{C}_{nn} \end{bmatrix},$$

with the understanding that the blank spaces are all zero. The $\mathbf{C}_{ii}$ blocks are zero-diagonal square matrices with numbers of rows equal to the sums of the sizes of all nodes in the ith layer of the computation tree. We now give the following lemma (the proof is given in Appendix A.1).

Lemma 4 *Let $\tilde{\mathbf{S}} = \mathbf{I}_k - \tilde{\mathbf{R}}$ with $\rho(|\tilde{\mathbf{R}}|) < 1$, then:*

1. *$\rho(|\tilde{\mathbf{R}}_n|) \leq \rho(|\tilde{\mathbf{R}}|)$ for all n.*
2. *The quantity $||\tilde{\mathbf{R}}_n^k||_\infty \leq \kappa_1[\rho(|\tilde{\mathbf{R}}|)]^k$ for a constant κ_1 that does not depend on n.*
3. *$||\tilde{\mathbf{T}}_n^{-1}||_\infty \leq \kappa_2$ for a constant κ_2 that does not depend on n.*

The constants referenced are:

$$\kappa_1 = \frac{max_i\{v_i\}}{min_j\{v_j\}} \tag{23}$$

$$\kappa_2 = \frac{\kappa_1}{1 - \rho(|\tilde{\mathbf{R}}|)}, \tag{24}$$

where $\mathbf{v} = (v_1, v_2, \ldots, v_k)' > \mathbf{0}$ such that $|\tilde{\mathbf{R}}|\mathbf{v} = \rho(|\tilde{\mathbf{R}}|)\mathbf{v}$.

Although Lemma 4 uses the computation tree for cluster 1 as an example, we see that this cluster plays no role in the computation of κ_1 and κ_2. Lemma 4 also applies to computation trees constructed for other clusters, and the computation of κ_1 and κ_2 remains as in (23) and (24) respectively (they do not depend on the cluster chosen to represent the root node).

We restrict our focus to proving the convergence of the posterior mean and posterior variances of GaBP-m applied to the preconditioned inputs. Let $\tilde{\boldsymbol{\mu}}$ be such that $\tilde{\mathbf{S}}\tilde{\boldsymbol{\mu}} = \boldsymbol{b}$ and suppose $\lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{ii}$ represents the submatrix of $\tilde{\mathbf{S}}^{-1}$ corresponding to the variables in cluster i. The following theorem summarizes our convergence results (for a proof see Appendix B).

Theorem 2 *Consider GaBP-m applied to the walk-summable preconditioned inputs $\tilde{\mathbf{S}}$ and $\tilde{\boldsymbol{b}}$ with clusters $\mathcal{C}_i : i = 1, 2, \ldots p$. The following results apply for all clusters i:*

1. $||\tilde{\boldsymbol{\mu}}_i^{(n-1)} - \tilde{\boldsymbol{\mu}}_i||_\infty \leq \kappa_4[\rho(|\tilde{\mathbf{R}}|)]^{n-1}$.
2. $[\tilde{\mathbf{P}}_i^{(n-1)}]^{-1} \sim \lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{ii} - \tilde{\mathbf{Z}}_n^{(i)}$.
3. $||\tilde{\mathbf{Z}}_n^{(i)}||_\infty \leq \kappa_1\kappa_2[\rho(|\tilde{\mathbf{R}}|)]^{t-1}$ *for* $n > t - 1$.

Here, $\kappa_4 = \kappa_1\kappa_2\kappa_3$, with $\kappa_3 = ||\tilde{\mathbf{S}}||_\infty||\tilde{\boldsymbol{\mu}}||_\infty$, $\tilde{\mathbf{Z}}_n^{(i)}$ forms a Cauchy sequence and t is the first layer after the root node where cluster i is referenced in its computation tree.

Since $\rho(|\tilde{\mathbf{R}}|) < 1$, Theorem 2 clearly implies convergence of the posterior means and covariance matrices for all clusters. Moreover, the converged posterior means represent the exact marginal means, while the converged posterior covariance matrices can be regarded as approximations to the true marginal covariance matrices. The approximation error of the converged posterior covariance matrices depends on $\lim_{n\to\infty}\tilde{\mathbf{Z}}_n^{(i)} = \tilde{\mathbf{Z}}_\infty^{(i)}$. In Theorem 2, we always have that $t \geq 3$, but t may be larger depending on the topology of the MG. Note that, if the computation tree never references cluster i again (as in the case of a tree-structured $\mathbf{S}$), then the converged posterior covariance matrix will be equal to the exact marginal covariance matrix. We see that Theorem 2 not only proves convergence, but also gives the rate of convergence of the posterior means and a framework for analyzing the accuracy of the approximate marginal covariance matrices.

4.2. Convergence of GaBP-m on Original Inputs

Define $\boldsymbol{\mu}$, $\boldsymbol{\mu}_i$ and $\lfloor\mathbf{S}^{-1}\rfloor_{ii}$ analogous to $\tilde{\boldsymbol{\mu}}$, $\tilde{\boldsymbol{\mu}}_i$ and $\lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{ii}$ respectively. We give the following corollary to Theorem 2.

Corollary 2 *Consider GaBP-m applied to the original (preconditioned walk-summable) inputs $\mathbf{S}$ and $\boldsymbol{b}$ with clusters $\mathcal{C}_i : i = 1, 2, \ldots p$. The following results apply for all clusters i:*

1. $||\boldsymbol{\mu}_i^{(n-1)} - \boldsymbol{\mu}_i||_\infty \leq \kappa_4||\boldsymbol{\Lambda}_{ii}||_\infty[\rho(|\tilde{\mathbf{R}}|)]^{n-1}$.
2. $[\mathbf{P}_i^{(n-1)}]^{-1} \sim \lfloor\mathbf{S}^{-1}\rfloor_{ii} - \mathbf{Z}_n^{(i)}$.
3. *For $n > t - 1$ we have* $||[\mathbf{P}_i^{(n)}]^{-1} - \lfloor\mathbf{S}^{-1}\rfloor_{ii}||_\infty \leq \kappa_1\kappa_2||\boldsymbol{\Lambda}_{ii}||_\infty^2[\rho(|\tilde{\mathbf{R}}|)]^{t-1}$.

Here, $\mathbf{Z}_n^{(i)} = \boldsymbol{\Lambda}_{ii}\tilde{\mathbf{Z}}_n^{(i)}\boldsymbol{\Lambda}_{ii}$ forms a Cauchy sequence, $\kappa_i : i = 1, 2, 3, 4$ are defined as in Theorem 2, and t is the first layer after the root node where cluster i is referenced in its computation tree.

Proof The proof follows directly from Theorem 2, Lemma 3 and:

$$\boldsymbol{\mu}_i = \boldsymbol{\Lambda}_{ii}\tilde{\boldsymbol{\mu}}_i$$
$$\lfloor\mathbf{S}^{-1}\rfloor_{ii} = \boldsymbol{\Lambda}_{ii}\lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{ii}\boldsymbol{\Lambda}_{ii},$$

which is true for all clusters i. ■

Clearly, Corollary 2 implies that preconditioned walk-summability is a sufficient condition for convergence of GaBP-m applied to the original inputs.

5. Empirical Results

In this next section we provide three empirical studies of the GaBP-m algorithm. The first study shows the novelty of preconditioned walk-summability with respect to walk-summability and scaled diagonal dominance using a specific set of inputs. The second study illustrates the advantages of using GaBP-m over the CG method within the context of preconditioning. We conclude by promoting GaBP-m over GaBP in an inference context.

5.1. Preconditioned Walk-summability

Consider applying GaBP-m to a precision matrix $\mathbf{S} : k \times k$ and potential vector $\boldsymbol{b} : k \times 1$, where the variables are assigned to nodes based on the clusters $\mathcal{C}_i : i = 1, 2, \ldots, p$. We have not discussed the selection of $\boldsymbol{\Lambda}$ in practice. One type of preconditioner we found to be effective in the prediction of the convergence of GaBP is the selection:

$$\boldsymbol{\Lambda}_{ii} = \mathbf{S}_{ii}^{-0.5} \text{ for } i = 1, 2, \ldots, p. \tag{25}$$

We now provide a specific example illustrating the novelty of preconditioned walk-summability. Consider the following precision matrix and potential vector: [1]

$$\mathbf{S} = \begin{bmatrix} 1.0000000 & 0.17373710 & 0.1850847 & 0.3354267 & 0.26006082 & 0.2192431 \\ 0.1737371 & 1.00000000 & 0.0881614 & 0.2410132 & 0.03527682 & 0.1426373 \\ 0.1850847 & 0.08816140 & 1.0000000 & 0.3954153 & 0.24977742 & 0.2611699 \\ 0.3354267 & 0.24101317 & 0.3954153 & 1.0000000 & 0.24246971 & 0.1855578 \\ 0.2600608 & 0.03527682 & 0.2497774 & 0.2424697 & 1.00000000 & 0.8966630 \\ 0.2192431 & 0.14263726 & 0.2611699 & 0.1855578 & 0.89666296 & 1.0000000 \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} 0.1878888 \\ 0.0430620 \\ 0.5864501 \\ 0.4414838 \\ 0.2120225 \\ 0.1740536 \end{bmatrix}.$$

The spectral radius of $|\mathbf{I}_k - \mathbf{S}|$ is $1.4069 > 1$, which does not give a decisive answer on whether GaBP will converge; in fact, we see that the application of GaBP to these inputs diverges. We now consider GaBP-m applied to these inputs, with the clusters $\mathcal{C}_1 = \{1, 2\}$, $\mathcal{C}_2 = \{3, 4\}$ and $\mathcal{C}_3 = \{5, 6\}$. Using the preconditioning defined in (25), we see that the spectral radius of $|\mathbf{I}_k - \tilde{\mathbf{S}}|$ is $0.6689 < 1$, and hence $\mathbf{S}$ is preconditioned walk-summable. The application of GaBP-m yields convergence after 19 iterations (using $\epsilon = 10^{-10}$). This convergence is explained by preconditioned walk-summability. We also see that GaBP-m can converge in cases where GaBP does not.

We conclude this example by investigating the bounds given in Corollary 2. Suppose we want n to be sufficiently large such that $\max_i\{||\boldsymbol{\mu}_i^{(n)} - \boldsymbol{\mu}_i||_\infty\} \leq \epsilon$, for some specified ϵ. Under the assumption of preconditioned walk-summability, we have:

$$n \geq \frac{\log(\frac{\epsilon}{\kappa_4 \max_i\{||\boldsymbol{\Lambda}_{ii}||_\infty\}})}{\log(\rho(|\tilde{\mathbf{R}}|))}. \tag{26}$$

Evaluating (26), we see that convergence will occur after at most 66 iterations (almost 3.5 times the actual number). The bound for the dissimilarity between the posterior and marginal variances in Corollary 2, where $t = 3$, can be evaluated as 5.191402 for cluster 1, which is poor compared to the actual value of 0.0929461. This bound will likely be better for sparse graphs (our example represents a dense graph).

1. These inputs represent an extract from the correlation matrix and correlation vector of the diabetes data used by Efron et al. (2004) to illustrate the least angle regression algorithm.

5.2. Robustness Towards Change in Variables

In this empirical study, we compare GaBP and GaBP-m to the CG and PCG algorithms. A description of the CG algorithm can be found in Shewchuk (1994). Our goal with this empirical comparison is not to promote GaBP and GaBP-m as solvers of linear systems, but rather to illustrate the stability of these algorithms with regard to the change in variables discussed in Section 3.4. The idea is to compare the performance of these algorithms by considering the application of the basic methods to an ill-conditioned $\mathbf{S}$. The performances on a well-conditioned matrix, $\tilde{\mathbf{S}} = \boldsymbol{\Lambda}\mathbf{S}\boldsymbol{\Lambda}$, are also compared.

In Section 3.4 we found that the rate of convergence of the means in GaBP-m applied to $\mathbf{S}$ will be identical to that of GaBP-m applied to $\tilde{\mathbf{S}}$, assuming that the clusters are chosen appropriately. In this empirical study, we will argue that the CG algorithm cannot benefit from the improved conditioning of $\tilde{\mathbf{S}}$ without explicitly incorporating the preconditioning. The main advantage of GaBP-m over the CG and PCG methods is that we do not need to know the preconditioner in advance to obtain the improved convergence rate. In this sense, GaBP-m automatically benefits from the convergence rate of the best valid preconditioned inputs.

Kamper et al. (2018) specify a method for generating precision matrices with arbitrary zero-diagonal spectral radius. The zero-diagonal spectral radius of $\mathbf{S}$ is obtained as the spectral radius of $\mathbf{R}$ in $\mathbf{S} = \mathbf{I}_k - \mathbf{R}$, assuming that $\mathbf{S}$ has been scaled to have only ones along its diagonal. We use this method to generate our inputs to the different algorithms.

This empirical study involves two illustrations. In the first illustration we compare GaBP to the CG and PCG algorithms. We note that the considerations of Section 3.4 also apply to GaBP in the sense that we can choose k clusters each of size 1. This constrains $\boldsymbol{\Lambda}$ to be a diagonal matrix, and we will show that GaBP is robust towards changes in variables induced by a diagonal preconditioner. To generate a set of inputs, we do the following:

1. Select a zero-diagonal spectral radius ($\tilde{\rho}$) uniformly from the interval $[0.5; 0.9]$.
2. Use the method of Kamper et al. (2018) to generate a $1\,000 \times 1\,000$ precision matrix $\tilde{\mathbf{S}}$ (with zero-diagonal spectral radius equal to $\tilde{\rho}$) and potential vector $\tilde{\boldsymbol{b}} : 1\,000 \times 1$. These will act as our preconditioned inputs.

3. Generate $\mathbf{u} : 1\,000 \times 1$, where each element is selected independently from a uniform random distribution over the interval $[0.01; 1]$. Set $\mathbf{D_u} = \text{diag}(\mathbf{u})$.

4. Our original (ill-conditioned) inputs are $\mathbf{S} = \mathbf{D_u}\tilde{\mathbf{S}}\mathbf{D_u}$ and $\boldsymbol{b} = \mathbf{D_u}\tilde{\boldsymbol{b}}$.

We generate 1 000 of these inputs randomly and compare four methods. The methods are GaBP on $\{\mathbf{S}, \boldsymbol{b}\}$, GaBP on $\{\tilde{\mathbf{S}}, \tilde{\boldsymbol{b}}\}$, CG on $\{\mathbf{S}, \boldsymbol{b}\}$ and CG on $\{\tilde{\mathbf{S}}, \tilde{\boldsymbol{b}}\}$. The labels for these methods are gabp.org, gabp.pre, cg.org and cg.pre respectively. The "org" denotes application to the original ill-conditioned inputs, while the "pre" denotes the preconditioned inputs. We compare these methods by recording the number of iterations required by each method for convergence. Convergence is defined to occur when $||\mathbf{S}\boldsymbol{\mu}^{(n)} - \boldsymbol{b}||_\infty \leq 10^{-6}$. The results of our simulations are given in the top graph of Figure 3. We see that the performance of GaBP on the different sets of inputs is nearly identical. This is in contrast to the CG method, where the difference in performance is substantial and the robustness of GaBP to the preconditioning is well illustrated. The CG method on the original inputs performs the worst by far out of all the methods; however, the preconditioned version outperforms the GaBP variants. It is important to note that the diagonal PCG method requires the specification of a preconditioner and does not keep the computations on the scale of the original inputs.

In our second illustration, we consider the automatic preconditioning done by GaBP-m, with the understanding that the nodes selected are higher-dimensional. This allows our preconditioner to be block-diagonal (in contrast to diagonal as for GaBP). We consider $1\,000 \times 1\,000$ precision matrices where we have 31 clusters each of size 31, and one additional cluster of size 39. Consider the following simulation procedure:

1. Select a zero-diagonal spectral radius ($\tilde{\rho}$) uniformly from the interval $[0.5; 0.9]$.

2. Generate a precision matrix $\tilde{\mathbf{S}} : 1\,000 \times 1\,000$ (with zero-diagonal spectral radius equal to $\tilde{\rho}$) and potential vector $\tilde{\boldsymbol{b}} : 1\,000 \times 1$. These will act as our preconditioned inputs.

3. We define a $1\,000 \times 1\,000$ matrix $\tilde{\mathbf{\Lambda}}$. For each cluster i of size d_i, we generate a $d_i \times d_i$ precision matrix with zero-diagonal spectral radius equal to 1.5, and set $\tilde{\mathbf{\Lambda}}_{ii}$ equal to this matrix. When $i \neq j$, $\tilde{\mathbf{\Lambda}}_{ij}$ is a matrix of zeros.

4. Our original (ill-conditioned) inputs are $\mathbf{S} = \tilde{\mathbf{\Lambda}}\tilde{\mathbf{S}}\tilde{\mathbf{\Lambda}}$ and $\boldsymbol{b} = \tilde{\mathbf{\Lambda}}\tilde{\boldsymbol{b}}$.

Again, we generate 1 000 of these structures and compare four different methods. We apply GaBP-m to $\{\mathbf{S}, \boldsymbol{b}\}$ with clusters as used in the simulation, and GaBP-m to $\{\tilde{\mathbf{S}}, \tilde{\boldsymbol{b}}\}$ with clusters as used in the simulation. These methods are labeled gabp.org and gabp.pre respectively. Let $\mathbf{D}_{\tilde{\mathbf{\Lambda}}}$ be the diagonal matrix, with the diagonal obtained from the diagonal of $\tilde{\mathbf{\Lambda}}$. We define a diagonal preconditioner $\mathbf{D}_{\tilde{\mathbf{\Lambda}}}^{-1}$ (this was used in the previous simulations) and apply CG to $\{\mathbf{D}_{\tilde{\mathbf{\Lambda}}}^{-1}\mathbf{S}\mathbf{D}_{\tilde{\mathbf{\Lambda}}}^{-1}, \mathbf{D}_{\tilde{\mathbf{\Lambda}}}^{-1}\boldsymbol{b}\}$. This method is labeled cg.diag. The last method is CG applied to $\{\tilde{\mathbf{S}}, \tilde{\boldsymbol{b}}\}$, which we label cg.pre. The results are illustrated in the bottom graph in Figure 3. We see a similar pattern as in the diagonal preconditioning case. The CG-based variants show the most volatility towards the preconditioning. In contrast to

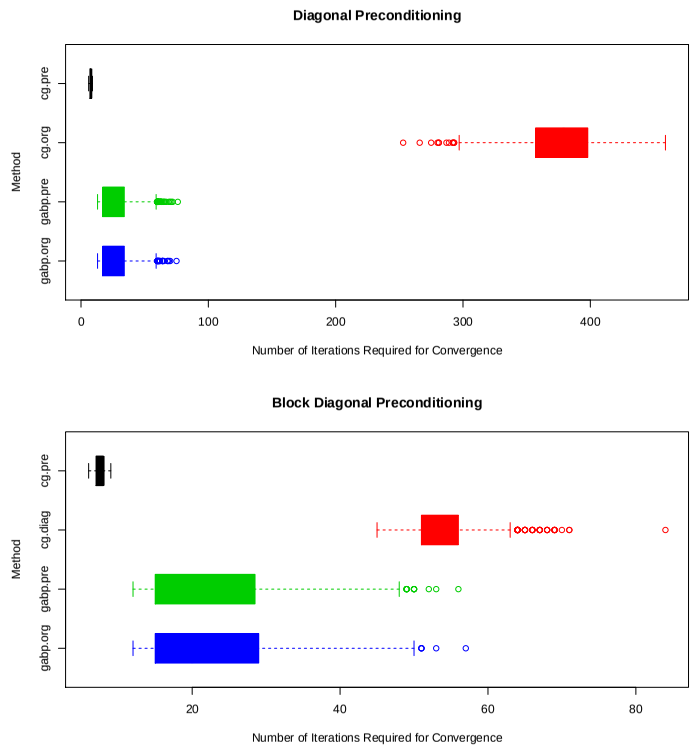


Figure 3: Visualization of the results of our simulations regarding preconditioning. The top and bottom graphs represent diagonal and block-diagonal preconditioning respectively. In both cases, we see that the performance of GaBP(-m) on the preconditioned and original inputs is nearly identical. This is in contrast to the performance of the CG method on both sets of inputs, where the difference is substantial.

the diagonal case, we see that the GaBP-m variants now tend to converge with a smaller number of iterations when compared to the diagonally preconditioned CG variant. The full preconditioned CG variant performs the best of all the methods.

5.3. GaBP-m vs GaBP

In this section, we discuss the utility of GaBP-m compared to GaBP. To facilitate this comparison, we used the following simulation procedure:

1. Select a zero-diagonal spectral radius uniformly from $[0.8; 1]$.
2. Generate a 100×100 precision matrix and a corresponding 100×1 potential vector using the simulation scheme from Kamper et al. (2018). The precision matrix is scaled to have a zero-diagonal spectral radius, as determined in (1).
3. Randomly generate 10 clusters each of size 10.

The above procedure was repeated 1 000 times to generate 1 000 different inputs. GaBP and GaBP-m (using the randomly selected clusters) were then applied to each of these data structures. The goal is to compare GaBP with GaBP-m based on the following considerations:

1. The number of iterations required for convergence.
2. The univariate inference quality of each method. We note that both GaBP and GaBP-m supply the exact marginal means at convergence. GaBP-m can be used for univariate marginal approximation by first approximating the higher-dimensional marginal, and then computing the univariate marginals from this approximation using a direct method. For instance, using GaBP-m, we can approximate the marginal precision associated with a specific cluster and then apply direct matrix inversion to approximate the univariate marginal precisions.

For the univariate inference quality we consider the KL divergence of the exact marginal distribution to the approximate marginal distribution. If f_i and $\hat{f}_i$ are the exact and approximate marginal associated with variable i, we are computing:

$$D_{KL}(f_i||\hat{f}_i) = \int_{-\infty}^{\infty} f_i(y) \log\left(\frac{f_i(y)}{\hat{f}_i(y)}\right) dy.$$

Note that, for a specific method, we get a KL divergence for each variable. The inference quality of a method for a specific input is measured by the mean of these divergences.

The results from our simulation study are summarized in Figure 4. Boxplots are drawn for the number of iterations required for convergence by GaBP and GaBP-m for the different simulated inputs. To illustrate inference quality, we draw boxplots of the mean KL divergences arising from the different inputs. The number of iterations required for convergence by GaBP-m tends to be less than that required by GaBP. We see that the univariate inference quality of GaBP-m tends to be better than that supplied by GaBP. GaBP-m also

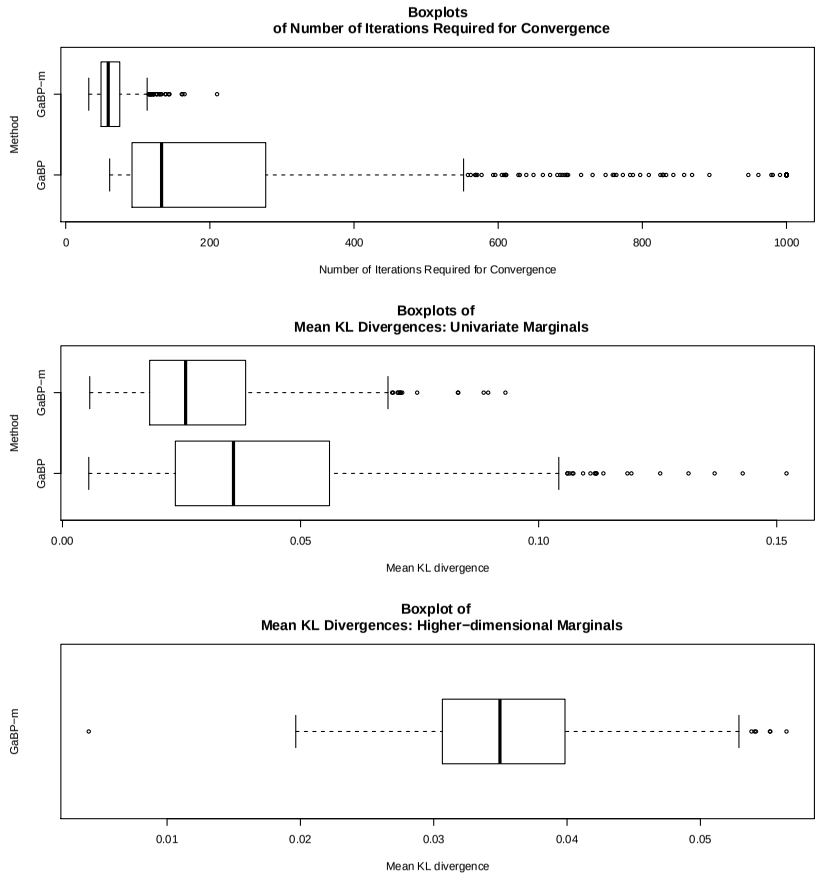


Figure 4: Visualization of the results of our simulations regarding the utility of GaBP-m. The top graph represents the number of iterations required for convergence by GaBP-m and GaBP. The middle graph illustrates the univariate inference quality of both methods, while the bottom graph analyses the higher-dimensional inference quality of GaBP-m. We see that GaBP-m tends to require a smaller number of iterations to converge and tends to provide more accurate univariate marginals. GaBP-m also provides useful approximate higher-dimensional marginals, something that GaBP cannot do.

has the advantage of offering approximations of the marginal precisions of the selected clusters. The inference quality of GaBP-m, with respect to the clusters, is also given in Figure 4. Inference quality was defined analogous to the univariate case. We see that GaBP-m can provide useful approximations for the higher-dimensional marginal distributions.

6. Conclusion

In this paper, we have considered a multivariate extension of belief propagation applied to pairwise Gaussian MGs (we labeled this GaBP-m). The main advantage of GaBP-m over GaBP is that it can be used to approximate the precision matrix of higher-dimensional marginals. In addition, GaBP-m may require fewer iterations than GaBP to converge and can be used to find better approximations for the univariate marginals. A multivariate version of the computation tree analysis, used to analyze univariate GaBP, was given. These computation trees were used to derive a relationship between GaBP-m applied to a set of inputs, before and after a certain change in variables was introduced. It was argued that GaBP-m is robust towards this change in variables, a property not shared by other solvers of linear systems such as the conjugate gradient solver. This was also illustrated empirically. The concept of preconditioned walk-summability was defined and shown to be a sufficient condition for convergence of the GaBP-m algorithm. Bounds were provided for the convergence speed and inference quality of GaBP-m under preconditioned walk-summability. The novelty of preconditioned walk-summability with respect to walk-summability and scaled diagonal dominance was discussed and illustrated empirically.

Unfortunately, GaBP-m is not guaranteed to converge for arbitrary precision matrices. Methods of augmenting this algorithm to converge, without compromising inference quality, should be considered in further research. A drawback of GaBP-m over GaBP is that the computational load per iteration scheme is higher. Methods to reduce this computational load, such as rank-1 updates, should be considered. Asynchronous GaBP-m and the effect of different initializations on the performance of the algorithm also require attention.

Acknowledgments

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF.

Appendix A. Proofs Leading to Lemma 4

We consider two additional lemmas before proving Lemma 4. Note that we focus, without loss of generality, on cluster 1. Consider clustering the real matrix $\mathbf{A} : k \times k$ according to $\mathcal{C}_i : i = 1, 2, \ldots, p$. We can construct a computation tree for $\mathbf{A}$ regardless of whether or not it is symmetrical. We denote the precision matrix of the computation tree of $\mathbf{A}$, for cluster 1, by $\mathbf{T}_n(\mathbf{A})$. In the appendices we still adopt the convention that $\mathbf{T}_n = \mathbf{T}_n(\mathbf{S})$ and $\tilde{\mathbf{T}}_n = \mathbf{T}_n(\tilde{\mathbf{S}})$. We now present and prove the two additional lemmas.

Lemma 5 *For all n, we have that*

$$||\mathbf{T}_n(\mathbf{A})||_\infty \leq ||\mathbf{A}||_\infty$$

with equality if the computation tree is of sufficient depth, such that all clusters are referenced before the terminal layer.

Proof Consider a node of the computation tree and suppose that this node has a reference to cluster s:

1. If this node is before the terminal layer, then its neighborhood in the computation tree will reference each cluster in $\mathcal{N}_s$ exactly once. Hence, the infinity norm of the row block of $\mathbf{T}_n(\mathbf{A})$ corresponding to this node will be equal to the infinity norm of the row block of $\mathbf{A}$ corresponding to the clusters in $\mathcal{N}_s$.

2. If this node is in the terminal layer, then its neighborhood in the computation tree will reference each cluster in $\mathcal{N}_s$ at most once (certain clusters will not be referenced). Hence, the infinity norm of the row block of $\mathbf{T}_n(\mathbf{A})$ corresponding to this node will be at most equal to the infinity norm of the row block of $\mathbf{A}$ corresponding to the clusters in $\mathcal{N}_s$.

The result follows directly from these points. ■

Lemma 6 *Consider an irreducible matrix $\mathbf{A} : t \times t$ consisting of non-negative elements. There exists an eigenvector $\mathbf{v}$, containing only positive elements, such that $\mathbf{A}\mathbf{v} = \rho(\mathbf{A})\mathbf{v}$ (the spectral radius of $\mathbf{A}$ is also an eigenvalue of $\mathbf{A}$). Furthermore, if $\mathbf{D} = diag(\mathbf{v})$, then:*

$$\rho(\mathbf{A}) = ||\mathbf{D}^{-1}\mathbf{A}\mathbf{D}||_\infty.$$

Proof First we note that the Perron-Frobenius theorem guarantees the existence of a vector $\mathbf{v}$, consisting only of positive elements, such that $\mathbf{A}\mathbf{v} = \rho(\mathbf{A})\mathbf{v}$. Let $\mathbf{A} = [a_{st}]$ and $\mathbf{v} = (v_1, v_2, \ldots, v_t)'$. Note that

$$\mathbf{D}^{-1}\mathbf{A}\mathbf{D} = \left[\frac{v_s a_{ts}}{v_t}\right],$$

and all elements of $\mathbf{D}^{-1}\mathbf{A}\mathbf{D}$ are non-negative. The consequence is that the sum of the absolute elements of row t of $\mathbf{D}^{-1}\mathbf{A}\mathbf{D}$ is

$$\sum_{s=1}^{t} \frac{v_s a_{ts}}{v_t} = \frac{1}{v_t}\sum_{s=1}^{t} v_s a_{ts} = \frac{v_t \rho(\mathbf{A})}{v_t} = \rho(\mathbf{A}),$$

since $\mathbf{v}$ is the eigenvector of $\mathbf{A}$ corresponding to the eigenvalue $\rho(\mathbf{A})$. The row sums of the absolute elements of $\mathbf{D}^{-1}\mathbf{A}\mathbf{D}$ are all equal to $\rho(\mathbf{A})$, and hence $||\mathbf{D}^{-1}\mathbf{A}\mathbf{D}||_\infty = \rho(\mathbf{A})$. ■

A.1. Proof of Lemma 4

Proof The proofs for the components of Lemma 4 are:

1. We assume that the matrix $\tilde{\mathbf{S}}$ is irreducible; if it is not, then GaBP-m can be applied to separate irreducible matrices. Note that the diagonal entries of $\tilde{\mathbf{S}}$ are all equal to one by assumption. Set $\tilde{\mathbf{S}} = \mathbf{I}_k - \tilde{\mathbf{R}}$, and therefore $|\tilde{\mathbf{R}}|$ is irreducible. Let $\mathbf{v}$ be the eigenvector of $|\tilde{\mathbf{R}}|$ corresponding to its spectral radius. Since $|\tilde{\mathbf{R}}|$ is non-negative and irreducible, the Perron-Frobenius theorem guarantees that the elements of $\mathbf{v}$ will be positive. Define $\bar{\mathbf{v}}_n = \mathbf{E}_n\mathbf{v}$, $\mathbf{D}_\mathbf{v} = \text{diag}(\mathbf{v})$ and $\mathbf{D}_{\bar{\mathbf{v}}_n} = \text{diag}(\bar{\mathbf{v}}_n)$. By Lemmas 5 and 6, we see that:

$$\begin{aligned}\rho(|\tilde{\mathbf{R}}_n|) &= \rho(\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|\mathbf{D}_{\bar{\mathbf{v}}_n}) \\ &\leq ||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty \\ &= ||\mathbf{T}_n(\mathbf{D}_\mathbf{v}^{-1}|\tilde{\mathbf{R}}|\mathbf{D}_\mathbf{v})||_\infty \\ &\leq ||\mathbf{D}_\mathbf{v}^{-1}|\tilde{\mathbf{R}}|\mathbf{D}_\mathbf{v}||_\infty \\ &= \rho(|\tilde{\mathbf{R}}|).\end{aligned}$$

2. Consider:

$$\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^k\mathbf{D}_{\bar{\mathbf{v}}_n} = \mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^{k-1}\mathbf{D}_{\bar{\mathbf{v}}_n}\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\mathbf{R}_n|\mathbf{D}_{\bar{\mathbf{v}}_n}. \tag{27}$$

From Equation (27),

$$\begin{aligned}||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^k\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty &\leq ||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^{k-1}\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty \\ &\leq \rho(|\tilde{\mathbf{R}}|)||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^{k-1}\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty.\end{aligned} \tag{28}$$

If we apply Inequality (28) recursively, we see that:

$$||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^k\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty \leq [\rho(|\tilde{\mathbf{R}}|)]^k,$$

and

$$\begin{aligned}|||\tilde{\mathbf{R}}_n|^k||_\infty &= ||\mathbf{D}_{\bar{\mathbf{v}}_n}\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^k\mathbf{D}_{\bar{\mathbf{v}}_n}\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}||_\infty \\ &\leq ||\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}||_\infty||\mathbf{D}_{\bar{\mathbf{v}}_n}^{-1}|\tilde{\mathbf{R}}_n|^k\mathbf{D}_{\bar{\mathbf{v}}_n}||_\infty \\ &\leq \frac{\max_i\{v_i\}}{\min_j\{v_j\}}[\rho(|\tilde{\mathbf{R}}|)]^k.\end{aligned}$$

Setting $\kappa_1 = \frac{\max_i\{v_i\}}{\min_j\{v_j\}}$,

$$||\tilde{\mathbf{R}}_n^k||_\infty \leq |||\tilde{\mathbf{R}}_n|^k||_\infty \leq \kappa_1[\rho(|\tilde{\mathbf{R}}|)]^k.$$

3. By assumption, $\rho(|\tilde{\mathbf{R}}|) < 1$, and hence we can write $\tilde{\mathbf{T}}_n^{-1}$ as a Neumann power series:

$$\tilde{\mathbf{T}}_n^{-1} = \sum_{i=0}^{\infty}\tilde{\mathbf{R}}_n^i. \tag{29}$$

The series in Equation (29) implies:

$$\begin{aligned}||\tilde{\mathbf{T}}_n^{-1}||_\infty &\leq \sum_{i=0}^{\infty}||\tilde{\mathbf{R}}_n^i||_\infty \\ &\leq \kappa_1\sum_{i=0}^{\infty}[\rho(|\tilde{\mathbf{R}}|)]^i \\ &= \frac{\kappa_1}{1-\rho(|\tilde{\mathbf{R}}|)} = \kappa_2.\end{aligned}$$

■

Appendix B. Proof of Theorem 2

Proof We focus, without loss of generality, on the computation tree for cluster 1. The proofs for the two components are:

1. Consider $\tilde{\boldsymbol{\mu}}_1^{(n-1)} = \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\mathbf{E}_n\tilde{\boldsymbol{b}}$. Since $\tilde{\mathbf{S}}$ is positive definite, there is one vector $\tilde{\boldsymbol{\mu}}$ in $\Re^k$ with the property $\tilde{\mathbf{S}}\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{b}}$. Let us consider solving $\mathbf{z}_n$ in

$$\tilde{\mathbf{T}}_n\mathbf{z}_n = \mathbf{E}_n\tilde{\boldsymbol{b}}.$$

Since $\tilde{\mathbf{S}}\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{b}}$, we have $\tilde{\mathbf{T}}_n\mathbf{z}_n = \mathbf{E}_n\tilde{\mathbf{S}}\tilde{\boldsymbol{\mu}}$. Substitution of Equation (9) into this expression yields $\tilde{\mathbf{T}}_n\mathbf{z}_n = (\tilde{\mathbf{T}}_n\mathbf{E}_n - \tilde{\mathbf{L}}_n)\tilde{\boldsymbol{\mu}}$, or

$$\mathbf{z}_n = \mathbf{E}_n\tilde{\boldsymbol{\mu}} - \tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}}.$$

Using the fact that $\tilde{\boldsymbol{\mu}}_1^{(n-1)} = \mathbf{G}_n'\mathbf{z}_n$, we have

$$\tilde{\boldsymbol{\mu}}_1^{(n-1)} = \mathbf{G}_n'\mathbf{E}_n\tilde{\boldsymbol{\mu}} - \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{\mu}}_1 - \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}},$$

where $\tilde{\boldsymbol{\mu}}_1$ is the subvector of $\tilde{\boldsymbol{\mu}}$ corresponding to the variables in cluster 1. Since $\rho(\tilde{\mathbf{R}}_n) \leq \rho(|\tilde{\mathbf{R}}_n|) \leq \rho(|\tilde{\mathbf{R}}|) < 1$, by Lemma 4, we can express $\tilde{\mathbf{T}}_n^{-1}$ as a Neumann power series,

$$\tilde{\mathbf{T}}_n^{-1} = \sum_{i=0}^{\infty}\tilde{\mathbf{R}}_n^i.$$

Consider the vector $\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}}$. Since the rows of $\tilde{\mathbf{L}}_n$ corresponding to the first $n-1$ blocks are all equal to zero (see Lemma 2), we can write $\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} = (\mathbf{0}_1', \mathbf{0}_2', \ldots, \mathbf{0}_{n-1}', \mathbf{h}_n')'$. The notation $\mathbf{0}_l$ stands for a vector containing a number of zeros equal to the sum of the dimensionalities of the nodes in layer l. The vector $\mathbf{h}_n$ is a specified vector (entries

can be non-zero) equal in size to the sum of the dimensionalities of the nodes in the terminal layer. With this notation, it is easy to see that

$$\mathbf{G}_n'\tilde{\mathbf{R}}_n^i\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} = \mathbf{0}_1$$

for all $i < n-1$ (note that the first layer contains only one node corresponding to cluster 1, hence $\mathbf{0}_1$ will contain d_1 zeros). We have:

$$\begin{aligned}
\mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} &= \mathbf{G}_n'\sum_{i=0}^{\infty}\tilde{\mathbf{R}}_n^i\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} \\
&= \mathbf{G}_n'\sum_{i=n-1}^{\infty}\tilde{\mathbf{R}}_n^i\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} \\
&= \mathbf{G}_n'\tilde{\mathbf{R}}_n^{n-1}\sum_{i=0}^{\infty}\tilde{\mathbf{R}}_n^i\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} \\
&= \mathbf{G}_n'\tilde{\mathbf{R}}_n^{n-1}\sum_{i=0}^{\infty}\tilde{\mathbf{R}}_n^i\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}} \\
&= \mathbf{G}_n'\tilde{\mathbf{R}}_n^{n-1}\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}}. \qquad (30)
\end{aligned}$$

The rows with non-zero entries of $\tilde{\mathbf{L}}_n$ are (sparse) replicates of a row of $\tilde{\mathbf{S}}$ by Lemma 2, and we therefore can say that $||\tilde{\mathbf{L}}_n\tilde{\boldsymbol{\mu}}||_\infty \leq ||\tilde{\mathbf{S}}||_\infty||\tilde{\boldsymbol{\mu}}||_\infty = \kappa_3$ (does not depend on n). Since $||\mathbf{G}_n'||_\infty = 1$, we have

$$||\tilde{\boldsymbol{\mu}}_1^{(n-1)} - \tilde{\boldsymbol{\mu}}_1||_\infty \leq \kappa_4[\rho(|\tilde{\mathbf{R}}|)]^{n-1}, \qquad (31)$$

where $\kappa_4 = \kappa_1\kappa_2\kappa_3$. Hence $\tilde{\boldsymbol{\mu}}_1^{(n)}$ will converge to $\tilde{\boldsymbol{\mu}}_1$ as $n \to \infty$. We can repeat the above analysis for all clusters i.

2. Note that Inequality (31) also applies when $\tilde{\boldsymbol{b}}$ is a matrix. Due to the order of the variables (that is $\mathbf{X} = (\mathbf{X}_1', \mathbf{X}_2', \ldots, \mathbf{X}_p')'$), we have that cluster 1 will contain the variables $1, 2, \ldots, d_1$ of the original pairwise MG. Let $\mathbf{e}_l : k \times 1$ be a vector of zeros except for entry l, which contains 1. Set $\tilde{\boldsymbol{b}} = \begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \ldots & \mathbf{e}_{d_1}\end{bmatrix}$. Using the convergence implied by (31), we see that

$$\mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\mathbf{E}_n\tilde{\boldsymbol{b}} \to \lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{11}$$

as $n \to \infty$, where $\lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{11}$ is the submatrix of $\tilde{\mathbf{S}}^{-1}$ corresponding to the variables in cluster 1. The matrix $\mathbf{E}_n\tilde{\boldsymbol{b}}$ has a row-block decomposition according to the nodes in the computation tree. If a node in the computation tree does not reference cluster 1, then the corresponding block of $\mathbf{E}_n\tilde{\boldsymbol{b}}$ will be zero, otherwise the block is $\mathbf{I}_{d_1}$ (this is because $\mathbf{F}_t\mathbf{e}_l$, for $t \neq 1$ and $l \leq d_1$, extracts a subvector of $\mathbf{e}_l$ containing only zeros). As a consequence, we have $\mathbf{E}_n\tilde{\boldsymbol{b}} = \mathbf{G}_n + \bar{\mathbf{E}}_n$, where $\bar{\mathbf{E}}_n$ is equal to $\mathbf{E}_n\tilde{\boldsymbol{b}}$, except for the first block, which contains only zeros. We can write

$$[\tilde{\mathbf{P}}_1^{(n-1)}]^{-1} = \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\mathbf{G}_n \sim \lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{11} - \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n$$

for large n. We now present the following lemma (proof follows in Section B.1).

Lemma 7 *If $\rho(|\tilde{\mathbf{R}}|) < 1$, then the sequence $\tilde{\mathbf{Z}}_n^{(1)} = \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n$ is a Cauchy sequence.*

Hence we have:

$$[\tilde{\mathbf{P}}_1^{(n-1)}]^{-1} \sim \lfloor\tilde{\mathbf{S}}^{-1}\rfloor_{11} - \tilde{\mathbf{Z}}_n^{(1)},$$

with $\tilde{\mathbf{Z}}_n^{(1)}$ forming a Cauchy sequence. Note that we can repeat this analysis for any cluster i.

3. Let t be the first layer, excluding layer 1, where cluster 1 is referenced in its computation tree. Note that the first $t-1$ row-blocks of $\bar{\mathbf{E}}_n$, corresponding to the line topology of the computation tree, will all be zero matrices. Hence we can write $\tilde{\mathbf{Z}}_n^{(1)} = \mathbf{G}_n'\mathbf{R}_n^{t-1}\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n$. Since $||\mathbf{G}_n'||_\infty = 1$ and $||\bar{\mathbf{E}}_n||_\infty \leq 1$, then by Lemma 4,

$$\begin{aligned}
||\tilde{\mathbf{Z}}_n^{(1)}||_\infty &= ||\mathbf{G}_n'\mathbf{R}_n^{t-1}\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n||_\infty \\
&\leq ||\mathbf{R}_n^{t-1}||_\infty||\tilde{\mathbf{T}}_n^{-1}||_\infty \\
&\leq \kappa_1\kappa_2[\rho(|\tilde{\mathbf{R}}|)]^{t-1}.
\end{aligned}$$

■

B.1. Proof of Lemma 7

Proof For convenience we write $\tilde{\mathbf{Z}}_n = \tilde{\mathbf{Z}}_n^{(1)}$. Consider the following recursive expansions:

$$\tilde{\mathbf{T}}_{n+1} = \begin{bmatrix}\tilde{\mathbf{T}}_n & \tilde{\mathbf{T}}_{1n} \\ \tilde{\mathbf{T}}_{n1} & \mathbf{U}_{nn}\end{bmatrix}$$

$$\bar{\mathbf{E}}_{n+1} = \begin{bmatrix}\bar{\mathbf{E}}_n \\ \mathbf{W}_n\end{bmatrix}$$

for certain matrices $\tilde{\mathbf{T}}_{1n}$, $\mathbf{U}_{nn}$ and $\mathbf{W}_n$. We have

$$\tilde{\mathbf{T}}_{n+1}^{-1} = \begin{bmatrix}\bar{\mathbf{T}}_n^{-1} & -\bar{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1} \\ \ldots & \ldots\end{bmatrix},$$

where the $\ldots$ represents an irrelevant part of the matrix, and $\bar{\mathbf{T}}_n = \tilde{\mathbf{T}}_n - \tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1}$. Consider

$$\begin{aligned}
\tilde{\mathbf{Z}}_{n+1} &= \mathbf{G}_{n+1}'\tilde{\mathbf{T}}_{n+1}^{-1}\bar{\mathbf{E}}_{n+1} \\
&= \begin{bmatrix}\mathbf{G}_n' & \mathbf{0}\end{bmatrix}\begin{bmatrix}\bar{\mathbf{T}}_n^{-1} & -\bar{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1} \\ \ldots & \ldots\end{bmatrix}\begin{bmatrix}\bar{\mathbf{E}}_n \\ \mathbf{W}_n\end{bmatrix} \\
&= \mathbf{G}_n'\bar{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n - \mathbf{G}_n'\bar{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n. \qquad (32)
\end{aligned}$$

It can be shown that

$$\begin{aligned}
\bar{\mathbf{T}}_n^{-1} &= \tilde{\mathbf{T}}_n^{-1} + \tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}[\mathbf{U}_{nn} - \tilde{\mathbf{T}}_{n1}\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}]^{-1}\tilde{\mathbf{T}}_{n1}\tilde{\mathbf{T}}_n^{-1} \\
&= \tilde{\mathbf{T}}_n^{-1} + \tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1},
\end{aligned}$$

where $\mathbf{V}_n = [\mathbf{U}_{nn} - \tilde{\mathbf{T}}_{n1}\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}]^{-1}\tilde{\mathbf{T}}_{n1}$. Furthermore,

$$\begin{aligned}\mathbf{G}_n'\bar{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n &= \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n + \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n \\ &= \tilde{\mathbf{Z}}_n + \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n.\end{aligned}$$

Considering again Equation (32),

$$\begin{aligned}\tilde{\mathbf{Z}}_{n+1} &= \tilde{\mathbf{Z}}_n + \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n - \mathbf{G}_n'\bar{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n \\ &= \tilde{\mathbf{Z}}_n + \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n - \mathbf{G}_n'[\tilde{\mathbf{T}}_n^{-1} + \tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}]\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n \\ &= \tilde{\mathbf{Z}}_n + \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n - \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n - \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n \\ &= \tilde{\mathbf{Z}}_n + \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}[\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\bar{\mathbf{E}}_n - \mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n - \mathbf{U}_{nn}^{-1}\mathbf{W}_n] \\ &= \tilde{\mathbf{Z}}_n + \mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}[\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}(\bar{\mathbf{E}}_n - \tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n) - \mathbf{U}_{nn}^{-1}\mathbf{W}_n]. \qquad (33)\end{aligned}$$

Note that the sparsity pattern of $\tilde{\mathbf{T}}_{1n}$ is similar to that of $\tilde{\mathbf{L}}_n$ and contains zeros in all the rows, except for the rows corresponding to the terminal layer of the computation tree (of depth n). Similar to Equation (30), we can show that

$$\mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n} = \mathbf{G}_n'\tilde{\mathbf{R}}_n^{n-1}\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n},$$

and hence:

$$||\mathbf{G}_n'\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}||_\infty \leq \kappa_1\kappa_2||\tilde{\mathbf{T}}_{1n}||_\infty[\rho(|\tilde{\mathbf{R}}|)]^{n-1}. \qquad (34)$$

By Equations (33) and (34), we see that:

$$||\tilde{\mathbf{Z}}_{n+1} - \tilde{\mathbf{Z}}_n||_\infty \leq \kappa_1\kappa_2||\tilde{\mathbf{T}}_{1n}||_\infty||\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}(\bar{\mathbf{E}}_n - \tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n) - \mathbf{U}_{nn}^{-1}\mathbf{W}_n||_\infty \times [\rho(|\tilde{\mathbf{R}}|)]^{n-1}. \qquad (35)$$

We now make some comments on the matrices present in Equation (35).

1. The matrix $\tilde{\mathbf{T}}_{1n}$ represents the links between parents in layer n and their children in layer $n+1$. Note that a parent in layer n has links to nodes in layer $n+1$ corresponding to its neighbors only, hence $||\tilde{\mathbf{T}}_{1n}||_\infty \leq \max_{i\neq j}\{||\tilde{\mathbf{S}}_{ij}||_\infty\}$.
2. The matrix $\mathbf{U}_{nn}$ is block diagonal, with diagonal blocks corresponding to the cluster diagonal blocks of $\tilde{\mathbf{S}}$ referenced in the final layer of the computation tree (of depth $n+1$). Therefore, $||\mathbf{U}_{nn}^{-1}||_\infty \leq \max_i\{||\tilde{\mathbf{S}}_{ii}^{-1}||_\infty\}$ and $||\mathbf{U}_{nn}||_\infty \leq \max_i\{||\tilde{\mathbf{S}}_{ii}||_\infty\}$.
3. Because $||\tilde{\mathbf{T}}_s^{-1}||_\infty \leq \kappa_2$ for all s, we have $||\bar{\mathbf{T}}_n^{-1}||_\infty \leq \kappa_2$ since $\bar{\mathbf{T}}_n^{-1}$ is a block of $\tilde{\mathbf{T}}_{n+1}^{-1}$.
4. Consider $\mathbf{V}_n = \mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1} + \mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1}[\tilde{\mathbf{T}}_n - \tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1}]^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1} = \mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1} + \mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1}\tilde{\mathbf{T}}_n^{-1}\tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\tilde{\mathbf{T}}_{n1}$. Since all the matrices in this expression have an infinity norm bounded by some constant which does not depend on n, there will also be a constant (independent of n) that bounds the infinity norm of $\mathbf{V}_n$.
5. Note that $||\bar{\mathbf{E}}_n||_\infty \leq 1$ and $||\mathbf{W}_n||_\infty \leq 1$ (recall the definition of $\bar{\mathbf{E}}_n$ in the proof of Theorem 2).

The above considerations imply the existence of a constant K (independent of n) such that $\kappa_1\kappa_2||\tilde{\mathbf{T}}_{1n}||_\infty||\mathbf{V}_n\tilde{\mathbf{T}}_n^{-1}(\bar{\mathbf{E}}_n - \tilde{\mathbf{T}}_{1n}\mathbf{U}_{nn}^{-1}\mathbf{W}_n) - \mathbf{U}_{nn}^{-1}\mathbf{W}_n||_\infty \leq K$. Therefore, $||\tilde{\mathbf{Z}}_{n+1} - \tilde{\mathbf{Z}}_n||_\infty \leq K[\rho(|\tilde{\mathbf{R}}|)]^{n-1}$, hence $\tilde{\mathbf{Z}}_n$ is a Cauchy sequence. ■

Appendix C. Examples for Section 3

In this appendix we give some examples for the considerations of Section 3. We use the following precision matrix and potential vector for our illustrations:

$$\mathbf{S} = \begin{bmatrix}\mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} & \mathbf{S}_{24} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{S}_{44}\end{bmatrix}$$

and

$$\boldsymbol{b} = \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4\end{bmatrix}.$$

We consider a computation tree, of depth 4, constructed for cluster 1. The different topologies corresponding to the MG, the computation tree and the line topology are illustrated in Figure 2. The precision matrix and potential vector for the computation tree are given by:

$$\mathbf{T}_4 = \begin{bmatrix}\mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{S}_{24} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{31} & \mathbf{0} & \mathbf{S}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{32} & \mathbf{0} & \mathbf{S}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{31} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{44} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{22} & \mathbf{0} & \mathbf{S}_{21} & \mathbf{S}_{24} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{12} & \mathbf{0} & \mathbf{S}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{44}\end{bmatrix}$$

$$\mathbf{t}_4 = \begin{bmatrix}\boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_4\end{bmatrix}.$$

C.1. Special Matrices

Some special matrices were introduced in Section 3.1. The row extractor matrix for cluster 1 is $\mathbf{F}_1 = \begin{bmatrix}\mathbf{I}_{d_1} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}$. Note that $\mathbf{F}_1\mathbf{S} = \begin{bmatrix}\mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{S}_{14}\end{bmatrix}$, and hence we are extracting the rows corresponding to cluster 1. The row extractor matrix for our running

example is:

$$\mathbf{E}_4 = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \mathbf{F}_3 \\ \mathbf{F}_4 \\ \mathbf{F}_2 \\ \mathbf{F}_1 \\ \mathbf{F}_1 \\ \mathbf{F}_4 \end{bmatrix}.$$

The sixth node of the computation tree is the rightmost node in layer 3 in the middle graph of Figure 2. The node extractor matrix for this node is $[\mathbf{G}_4^{(6)}]' = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$. We see that $[\mathbf{G}_4^{(6)}]'\mathbf{T}_4$ extracts the rows of $\mathbf{T}_4$ corresponding to node 6 in the computation tree. Lemma 2 can be validated using our running example:

$$\mathbf{T}_4\mathbf{E}_4 = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} & \mathbf{S}_{24} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} & \mathbf{0} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{S}_{44} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} & \mathbf{S}_{24} \\ \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} \\ \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{S}_{44} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{S}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{S}_{13} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} = \mathbf{E}_4\mathbf{S} + \mathbf{L}_4.$$

C.2. Tree-pruning Procedure

We now illustrate the tree-pruning procedure discussed in Sections 3.2 and 3.3. A visual illustration is given in Figure 5.

In Step 1 we have the original computation tree. We select node 6 of the computation tree as our bereaved parent. To move to Step 2, we prune the children of node 6 and adjust the precision matrix and potential vector associated with node 6, as discussed in Section 3.2. If we perform inference on the tree in Step 2, then the marginal at the root node is equivalent to the corresponding marginal in Step 1. We repeat this process until all nodes in the final layer have been removed. We then proceed to the next layer and continue until we are left only with the root node, as in Step 6.

We consider how the computation tree preserves the message-passing structure for our running example. Consider moving from Step 1 to Step 2 in Figure 5. Noting that $\mathbf{P}_{12}^{(0)} = \mathbf{S}_{11}$

and $\mathbf{P}_{42}^{(0)} = \mathbf{S}_{44}$, we see that the tree-pruning procedure requires us to perform inference on:

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}_{22} & \mathbf{S}_{21} & \mathbf{S}_{24} \\ \mathbf{S}_{12} & \mathbf{S}_{11} & \mathbf{0} \\ \mathbf{S}_{42} & \mathbf{0} & \mathbf{S}_{44} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{22} & \mathbf{S}_{21} & \mathbf{S}_{24} \\ \mathbf{S}_{12} & \mathbf{P}_{12}^{(0)} & \mathbf{0} \\ \mathbf{S}_{42} & \mathbf{0} & \mathbf{P}_{42}^{(0)} \end{bmatrix}$$

$$\mathbf{m} = \begin{bmatrix} \boldsymbol{b}_2 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_2 \\ \mathbf{z}_{12}^{(0)} \\ \mathbf{z}_{42}^{(0)} \end{bmatrix}.$$

Equations (16) and (17) imply that $r_P(\mathcal{M}) = \mathbf{P}_{23}^{(1)}$ and $r_\mu(\mathcal{M}) = [\mathbf{P}_{23}^{(1)}]^{-1}\mathbf{z}_{23}^{(1)}$. We remove nodes 8 and 9 of the computation tree (1 and 4 in the red rectangle) and adjust the potential and precision of node 6 (2 in red rectangle) to $r_P(\mathcal{M})r_\mu(\mathcal{M}) = \mathbf{z}_{23}^{(1)}$ and $r_P(\mathcal{M}) = \mathbf{P}_{23}^{(1)}$ respectively. The potential vector and precision matrix of the trimmed tree are given in Equations (36) and (37) (the vector and matrix to the right of the arrows indicate the potential vector and precision matrix of the tree obtained after the next pruning step):

$$\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \mathbf{z}_{23}^{(1)} \\ \boldsymbol{b}_1 \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \mathbf{z}_{32}^{(1)} \\ \mathbf{z}_{42}^{(1)} \\ \mathbf{z}_{23}^{(1)} \end{bmatrix} \tag{36}$$

$$\begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{S}_{24} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{31} & \mathbf{0} & \mathbf{S}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{32} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{32} & \mathbf{0} & \mathbf{S}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{31} \\ \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{44} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{23}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{11} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{S}_{24} & \mathbf{0} \\ \mathbf{S}_{31} & \mathbf{0} & \mathbf{S}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{32} \\ \mathbf{0} & \mathbf{S}_{32} & \mathbf{0} & \mathbf{P}_{32}^{(1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{42} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{42}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{23}^{(1)} \end{bmatrix}. \tag{37}$$

Note that, due to the specific sparsity pattern, we can write $\mathbf{z}_{42}^{(1)} = \mathbf{z}_{42}^{(0)} = \boldsymbol{b}_2$ and $\mathbf{P}_{42}^{(1)} = \mathbf{P}_{42}^{(0)} = \mathbf{S}_{44}$. We can repeat this process until we are left with the following potential vector and precision matrix:

$$\begin{bmatrix} \boldsymbol{b}_1 \\ \mathbf{z}_{21}^{(2)} \\ \mathbf{z}_{31}^{(2)} \end{bmatrix}; \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} \\ \mathbf{S}_{21} & \mathbf{P}_{21}^{(2)} & \mathbf{0} \\ \mathbf{S}_{31} & \mathbf{0} & \mathbf{P}_{31}^{(2)} \end{bmatrix}.$$

We then apply one more pruning step in order to obtain:

$$\begin{aligned} r_P(\mathcal{T}_4) &= \mathbf{S}_{11} - \mathbf{S}_{12}\mathbf{P}_{21}^{(2)}\mathbf{S}_{21} - \mathbf{S}_{13}\mathbf{P}_{31}^{(2)}\mathbf{S}_{31} = \mathbf{S}_{11} + \mathbf{Q}_{21}^{(3)} + \mathbf{Q}_{31}^{(3)} = \mathbf{P}_1^{(3)} \\ r_\mu(\mathcal{T}_4) &= [\mathbf{P}_1^{(3)}]^{-1}[\boldsymbol{b}_1 - \mathbf{S}_{12}[\mathbf{P}_{21}^{(2)}]^{-1}\mathbf{z}_{21}^{(2)} - \mathbf{S}_{13}[\mathbf{P}_{31}^{(2)}]^{-1}\mathbf{z}_{31}^{(2)}] \\ &= [\mathbf{P}_1^{(3)}]^{-1}[\boldsymbol{b}_1 + \mathbf{v}_{21}^{(2)} + \mathbf{v}_{31}^{(2)}] = \boldsymbol{\mu}_1^{(3)}. \end{aligned}$$

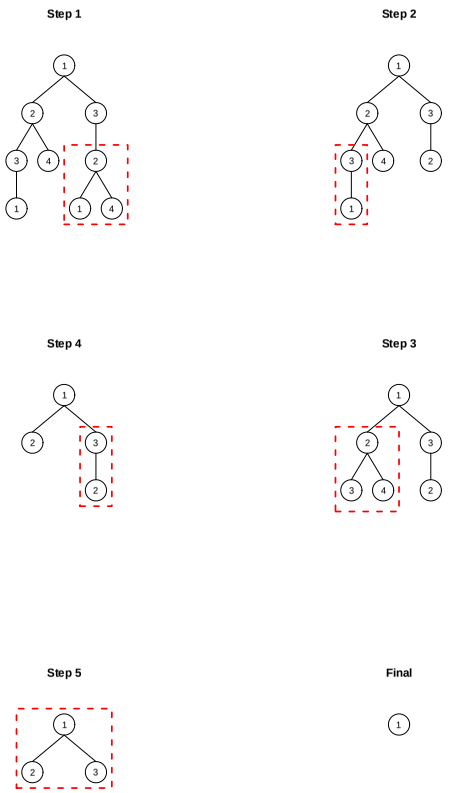


Figure 5: Visualization of the process used to invert a tree-structured precision matrix/solve a linear system through node pruning, where the block of the inverse/solution of the root node is of interest.

References

Danny Bickson. *Gaussian Belief Propagation: Theory and Application*. PhD thesis, The Hebrew University of Jerusalem, October 2008.

Venkat Chandrasekaran, Jason K. Johnson, and Alan S. Willsky. Estimation in Gaussian graphical models using tractable subgraphs: a walk-sum analysis. *IEEE Transactions on Signal Processing*, 56:1916–1930, 2008.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Yousef El-Kurdi, Dennis Giannacopoulos, and Warren J. Gross. Relaxed Gaussian belief propagation. In *Proceedings IEEE International Symposium on Information Theory*, September 2012a.

Yousef El-Kurdi, Warren J. Gross, and Dennis Giannacopoulos. Efficient implementation of Gaussian belief propagation solver for large sparse diagonally dominant linear systems. *IEEE Transactions on Magnetics*, 48:471–474, February 2012b.

Robert G. Gallager. *Low-Density Parity-Check Codes*. MA: MIT Press, Cambridge, 1963.

Qinghua Guo and Defeng Huang. EM-based joint channel estimation and detection for frequency selective channels using Gaussian message passing. *IEEE Transactions on Signal Processing*, 59:4030–4035, 2011.

Qinghua Guo and Li Ping. LMMSE turbo equalization based on factor graphs. *IEEE Journal on Selected Areas in Communications*, 26:311–319, 2008.

Jason K. Johnson, Danny Bickson, and Danny Dolev. Fixing convergence of Gaussian belief propagation. In *Proceedings IEEE International Symposium on Information Theory*, Seoul, South Korea, 2009.

Francois Kamper. An empirical study of Gaussian belief propagation and application in the detection of F-formations. In *Proceedings of the ACM Multimedia 2017 Workshop on South African Academic Participation*, October 2017.

Francois Kamper, Johan A. du Preez, Sarel J. Steel, and Stephan Wagner. Regularized Gaussian belief propagation. *Statistics and Computing*, 28(3):653–672, May 2018.

Ying Liu. Feedback message passing for inference in Gaussian graphical models. Master's thesis, Massachusetts Institute of Technology, June 2010.

Ying Liu, Venkat Chandrasekaran, Animashree Anandkumar, and Alan S. Willsky. Feedback message passing for inference in Gaussian graphical models. *IEEE Transactions on Signal Processing*, 60(8):4135–4150, 2012.

Dmitry M. Malioutov. *Approximate inference in Gaussian graphical models*. PhD thesis, MIT, 2008.

Dmitry M. Malioutov, Jason K. Johnson, and Alan S. Willsky. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research*, 7: 2031–2064, October 2006.

Ciamac C. Moallemi and Benjamin Van Roy. Convergence of min-sum message passing for quadratic optimization. *IEEE Transactions on Information Theory*, 55(5):2413–2423, May 2009.

Ciamac C. Moallemi and Benjamin Van Roy. Convergence of min-sum message passing for convex optimization. *IEEE Transactions on Information Theory*, 56(4):2041–2050, April 2010.

Andrea Montanari, Balaji Prabhakar, and David Tse. Belief propagation based multi-user detection. In *Proceedings IEEE Information Theory Workshop*, Punta del Este, Uruguay, March 2006.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, San Francisco, CA, USA, 1988.

Nicholas Ruozzi and Sekhar Tatikonda. Message-passing algorithms for quadratic minimization. *Journal of Machine Learning Research*, 14:2287–2314, 2013.

Nicholas Ruozzi, Justin Thaler, and Sekhar Tatikonda. Graph covers and quadratic minimization. In *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, September 2009.

Matthias W. Seeger and David P. Wipf. Variational Bayesian inference techniques. *IEEE Signal Processing Magazine*, 27:81–91, November 2010.

Ori Shental, Paul H. Siegel, Jack K. Wolf, Danny Bickson, and Danny Dolev. Gaussian belief propagation solver for systems of linear equations. In *Proceedings IEEE International Symposium on Information Theory*, pages 1863–1867, 2008.

Jonathan R. Shewchuk. *An introduction to the conjugate gradient method without the agonizing pain.* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, August 4 1994.

Qinliang Su and Yik-Chung Wu. On convergence conditions of Gaussian belief propagation. *IEEE International Transactions on Signal Processing*, 63:1144–1155, March 2015.

Tianju Sui, Damian E. Marelli, and Minyue Fu. Convergence analysis of Gaussian belief propagation for distributed state estimation. In *Proceedings IEEE Annual Conference on Decision and Control*, Osaka, Japan, Dec. 2015.

Yair Weiss and William T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

Journal of Machine Learning Research 20 (2019) 1-40 Submitted 2/18; Revised 9/19; Published 11/19

Unsupervised Evaluation and Weighted Aggregation of Ranked Classification Predictions

Mehmet Eren Ahsen[1,2] AHSEN@ILLINOIS.EDU
Robert M Vogel[2,3] R.VOGEL@IBM.COM
Gustavo A Stolovitzky[2,3] GUSTAVO@US.IBM.COM
[1]*University of Illinois at Urbana-Champaign*
Department of Business Administration,
1206 S 6th St, Champaign, IL 61820, USA
[2]*Icahn School of Medicine at Mount Sinai*
Department of Genetics and Genomic Sciences
One Gustave Levy Place, Box 1498
New York, NY , USA
[3]*IBM T.J. Watson Research Center*
1101 Kitchawan Road, Route 134,
Yorktown Heights, N.Y., 10598, USA.

Editor: Samy Bengio

Abstract

Ensemble methods that aggregate predictions from a set of diverse base learners consistently outperform individual classifiers. Many such popular strategies have been developed in a supervised setting, where the sample labels have been provided to the ensemble algorithm. However, with the rising interest in unsupervised algorithms for machine learning and growing amounts of uncurated data, the reliance on labeled data precludes the application of ensemble algorithms to many real world problems. To this end we develop a new theoretical framework for ensemble learning, the Strategy for Unsupervised Multiple Method Aggregation (SUMMA), that estimates the performances of base classifiers and uses these estimates to form an ensemble classifier. SUMMA also generates an ensemble ranking of samples based on the confidence score it assigns to each sample. We illustrate the performance of SUMMA using a synthetic example as well as two real world problems.

Keywords: Ensemble learning, Ensemble classifier, Unsupervised Learning, AUC, Spectral Decomposition

1. Introduction

Algorithmic solutions to typical machine learning problems are quickly increasing in number and complexity. However, no algorithm performs best under every circumstance: there is no one-size-fits-all solution in machine learning. A mathematical formulation of this phenomena is known as the "no free lunch" theorem (Wolpert, 1996). On the other hand, it has long been appreciated that the combination of different algorithms, both in classification and regression tasks, results in more robust solutions (Dietterich, 2002).

©2019 Mehmet Eren Ahsen, Robert Vogel, and Gustavo Stolovitzky.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-094.html.

Indeed, the endeavor of combining predictions or preferences from multiple sources has been institutionalized and incorporated in our everyday decision making. For example, in democratic elections the candidate who gets the most votes wins, or in online purchases the products with better customer reviews are more likely to be chosen. In social sciences this collaborative decision-making process is known as the Wisdom of Crowds (WOC) (Surowiecki, 2005). In the machine learning literature, the process of combining multiple base learners is known as ensemble learning.

Ensemble algorithms can be divided into two main categories, namely supervised and unsupervised ensemble algorithms. Supervised ensemble algorithms can be further subdivided into two main subcategories. The first represents homogeneous ensemble algorithms that consist of training several instances of a single type of algorithm on various splits of labeled data. This subcategory of methods generates diversity by subsampling training examples (bagging) (Breiman, 1996) or assigning weights to training examples (boosting) (Freund and Schapire, 1997; Schapire, 1990) utilizes a single type of base classifier to build the ensemble classifier. Homogeneous ensembles may suffer in applications where the data is very complex, and consequently the most appropriate base classifier model is not known *a priori*. The second subcategory is known as heterogeneous ensemble algorithms, where each base classifier represents a distinct algorithm. Two popular heterogeneous ensemble methods are a form of meta-learning called stacking (Wolpert, 1992) as well as the ensemble selection procedure proposed in (Caruana et al., 2004). Among them, stacking trains a higher-level classifier over the predictions of base classifiers, while ensemble selection uses an iterative strategy to select an optimal set of base classifiers that balances diversity and performance.

Although supervised heterogeneous ensembles often out-perform homogeneous ensembles (Niculescu-Mizil et al., 2009; Gashler et al., 2008), there are several limitations regarding their use in real life scenarios. Along with the risk of overfitting associated with the meta-training of ensemble classifiers, another very important limitation of heterogeneous supervised ensemble algorithms is that in many domains labeled data for training ensemble classifiers are scarce and, in some cases, there may not be any available at all. For such cases, unsupervised learning methods might be more appropriate. Unsupervised ensemble learning has been used successfully in diverse applications including computational biology (Marbach et al., 2012), crowdsourcing in natural language tasks (Snow et al., 2008), and business (Tsai and Hsiao, 2010). The simplest unsupervised ensemble algorithm creates an ensemble score by an unweighted average of the ranks assigned by the base classifiers to each item being classified (Marbach et al., 2012; Emerson, 2013). We will call this simple unsupervised ensemble method as the "Wisdom of Crowds" (WOC) approach (Surowiecki, 2005). The WOC approach suffers when the majority of base classifiers perform poorly. A strategy to mitigate this shortcoming would be to assign each base classifier a weight commensurate (e.g., proportional) to its performance. However, this approach may not be advisable or applicable when (1) the biases of the training sets are different from the test set, or (2) there is scarce or no prior labeled data sets to calculate performance.

An early example of ensemble classification is given in Dawid and Skene (1979), where the authors propose to infer the performance of each base classifier and the true class labels associated with each sample together. Specifically, they recast the problem so that a maximum likelihood solution could be estimated by the Expectation Maximization (EM) al-

gorithm (Dempster et al., 1977). While elegant, this solution to the unsupervised ensemble construction suffers from the known limitations of the EM algorithm for non-convex optimization problems. Another important limitation is the strong assumption of conditional independence which might be violated in practical settings.

In this work, we propose a new theoretical framework for unsupervised ensemble learning, the Strategy for Unsupervised Multiple Method Aggregation (SUMMA). Our work is a generalization of the Spectral Meta-Learner (SML) theory developed in Parisi et al. (2014). In SML, the authors starting point is the covariance matrix of predicted binary labels of a set of base classifiers. They show that under the assumption of conditional independence of the class predictions of base classifiers given the true class labels, the off-diagonal elements of the covariance matrix are related to the balanced accuracy of each base classifier. Many popular classification algorithms such as logistic regression (Harrell, 2001), SVM (Support Vector Machines) (Cortes and Vapnik, 1995), and deep learning algorithms (Chollet, 2017) produce continuous scores that can be interpreted as a measure of the confidence assigned by the base classifier that a given item is of one of the two binary classes. By forcing base classifiers to produce binary predictions SML is not only ignoring the readily available continuous scores outputted by typical base classifiers, but it also runs into the risk of overfitting by requiring each algorithm to define a threshold to binarize its output. In the case of SUMMA, our starting point is the covariance matrix of ranked predictions which can be created by rank ordering the continuous scores outputted by the base classifiers. We show that the off-diagonal entries of this covariance matrix are related to the Area Under the Receiver Operating Characteristics Curve (AUC) (Marzban, 2004) of the base classifiers. Under the assumption of conditional independence of the ranks assigned by base classifiers to an item given the class of the item, the SUMMA algorithm:

1. Infers the empirical performance, as measured by the AUC, of each base classifier without labeled data,
2. Uses the inferred performances of the base classifiers to generate an aggregate continuous score for each sample, and
3. Uses these scores to predict the binary class labels for each sample.

To exemplify our theoretical results we first apply SUMMA to a simulated dataset where our theoretical assumption of conditional independence strictly holds. We also apply SUMMA to two datasets taken from real applications where we show that SUMMA performs robustly even if the conditional independence assumption is slightly violated.

2. Problem Setup

Given a classification problem, we assume that an instance pair $(X, Y) \in \mathcal{X} \times \{0, 1\}$ is a random vector with probability density function $p(x, y)$ and marginals $P_X(x)$ and $P_Y(y)$, where the set $\mathcal{X}$ denotes the feature space, and without loss of generality, $y = 0$ denotes the negative class and $y = 1$ denotes the positive class. Let $\{g_i\}_{i=1}^M$ represent an ensemble of classifiers with unknown reliability, where each classifier, $g_i : \mathcal{X} \to \mathbb{R}$, is a mapping from the feature space $\mathcal{X}$ to real numbers. We assume that each classifier is trained on its own possibly labeled dataset which is unknown to us and produces confidence scores

$g_i(x_k)$ associated with a set of N instances $\mathcal{D} = \{(x_k)\}_{k=1}^N \subset \mathcal{X}$, whose true but unknown class labels are denoted by the vector $\boldsymbol{y} = [y_1, \cdots, y_N]^T$. We let $N_1 := \sum_{i=1}^N y_i$ denote the number of positive instances, $N_0 = N - N_1$ denote the number of negative instances and $\rho = N_1/N$ denote the *prevalence* of the positive class, which we simply denote as prevalence in the text. We can interpret the output of each classifier as a measure of the relative confidence that the sample belongs to one of the classes, which, without loss of generality, will be assumed to be class 1. For example, in a Bayesian framework the output of a classifier is the posterior probability that a sample belongs to the positive class. In the case of SVM, the output is the distance to the separating hyperplane.

Recent research suggests that appropriate calibration of the classifier outputs will boost the performance of the ensemble classifier (Whalen and Pandey, 2013; Bella et al., 2013). In the current manuscript, we will use the rank transformation as a calibration tool due to its theoretical implications presented in the subsequent sections. Accordingly, let $\Omega = \{(x_1, y_1), \cdots, (x_N, y_N) \in (\mathcal{X} \times \{0, 1\})^N : \sum_{i=1}^N y_i = N_1\}$ denote the space of all N i.i.d realizations of $\mathcal{X} \times \{0, 1\}$ with exactly N_1 positive instances. For each $T \in \Omega$ and classifier i, consider the following vector $g_i(T) = [g_i(x_1), \cdots, g_i(x_N)]^T \in \mathbb{R}^N$. By using the well-ordering property of the real numbers, we can map $g_i(T)$ to a rank vector $\mathbf{r} = [r_1, ..., r_N]^T \in \mathcal{S}_N$, where $\mathcal{S}_N$ denotes the set of all permutations of the integers $\{1, \cdots, N\}$. As in the recent literature (Agarwal et al., 2005), we assume that ties, i.e. $g_i(x_j) = g_i(x_k)$ for $j \neq k$, are broken uniformly at random. In this case, w.l.o.g., we assume that g_i assigns ranks to tied samples according to their position in the vector $g_i(T)$. Throughout the rest of the manuscript, unless otherwise noted when we say we are given a realization $T \in \Omega$, we assume that we do not observe the labels associated with each sample and only observe the feature vector x_k associated with each sample. With the above notation, we let $P(R_i = r|y_k)$ denote the probability, over all realizations from Ω, that a sample of class $y_k \in \{0, 1\}$ has been assigned rank $r \in \{1, \cdots, N\}$ by the classifier i. For simplicity, we will refer to this quantity as $P_i(r|y_k)$ and also let $P(R_1 = r_{1k}, R_2 = r_{2k}, \cdots, R_M = r_{Mk}|y_k)$ denote the joint probability of each classifier i assigning rank r_{ik} to a sample k of class y_k.

3. Theory

3.1. Performance Metric

We start by defining the main performance metric that is primarily used throughout the paper.

Definition 1 *The performance of the ith classifier is measured by,*

$$\Delta_i = \mathbb{E}[R_i|y = 0] - \mathbb{E}[R_i|y = 1],$$

where $\mathbb{E}[R_i|y = j]$, for $j = 0, 1$, represents the average rank given the respective class for the ith classifier.

Δ_i is defined over the set Ω and as such it may depend on N and ρ. Note that a random method has Δ_i=0. Intuitively, this is because random methods are those unable to rank samples according to the latent class, a consequence of the fact that for a random method $P_i(r|y = 1) = P_i(r|y = 0) = \mathcal{U}(1, N)$, where $\mathcal{U}(1, N)$ denotes the uniform distribution on the

set of integers $\{1, 2, ..., N\}$. On the other hand if $|\Delta_i| > 0$, then method i is an informative method and can discriminate rank assignments by the sample class.

We will assume that all methods $i = 1, ..., M$ use the same convention consisting of assigning ranks to the examples predicted to be in the positive class in the lower range values of the interval $1, ..., N$, and the examples predicted to be in the negative class to the upper range of the same interval. In this way, a perfect classifier g_P would assign the ranks $[1, 2, ..., N_1]$ to the N_1 positive examples, and the ranks $[N_1 + 1, ..., N]$ to the negative examples. In this case $\mathbb{E}[R_P|y = 1] = (N_1 + 1)/2$, $\mathbb{E}[R_P|y = 0] = (N + N_1 + 1)/2$ and $\Delta_P = N/2$.

3.2. Conditionally Independent Classifiers

In this section we first define the conditional independence assumption which is central to our theoretical results. We call the classifiers, $\{g_1, \cdots, g_M\}$, l^{th} order conditionally independent, if for any $L = \{g_{i_1}, \cdots, g_{i_l}\} \subset \{g_1, \cdots, g_M\}$ their conditional distribution factorizes,

$$P\left(R_{i_1} = r_{i_1 k}, R_{i_2} = r_{i_2 k}, \cdots, R_{i_l} = r_{i_l k}|y_k\right) = P_{i_1}(r_{i_1 k}|y_k)P_{i_2}(r_{i_2 k}|y_k)\cdots P_{i_l}(r_{i_l k}|y_k). \quad (1)$$

Next we prove that under the assumption of conditionally independent classifiers, their higher-order covariance tensors have a specific form.

Theorem 2 *Suppose the classifiers $\{g_1, \cdots, g_M\}$ are l^{th} order conditionally independent so that equation (1) holds. For any $n \leq l \leq M$, let the nth order covariance tensor, Σ_n , be defined as*

$$\Sigma_n(1, \cdots, n) := \mathbb{E}\big[\big(R_1 - \mathbb{E}[R_1]\big)\cdots\big(R_n - \mathbb{E}[R_n]\big)\big], \quad (2)$$

where w.l.o.g. we denoted a given subset of classifiers of size n by the set $\{1, \cdots, n\}$. Then the following equality holds,

$$\Sigma_n(1, \cdots, n) = C(\rho)(\rho^{n-1} - (\rho - 1)^{n-1})\prod_{j=1}^{n}\Delta_j, \quad (3)$$

where $C(\rho) := \rho(1 - \rho)$ and $\rho = N_1/N$ denotes the prevalence of the positive class.

Proof See Appendix B. ■

Theorem 2 shows that under the assumption of conditionally independent ensemble members, the nth central moment contains information of each method performance, Δ_i. In addition, note the symmetry $\rho \to 1 - \rho$, and $\Delta \to -\Delta$ of the n-th central moment. This symmetry is expected because the classes 1 and 0 are assigned by convention. As mentioned above, we assume that all the base classifiers have adopted the same convention of assigning lower (higher) ranks to items predicted to be in the positive (negative) class. Therefore if the positive class were now called negative and vice versa, then ρ would change to $1 - \rho$, Δ would change to $-\Delta$, and the formula for the n-th central moment would remain invariant to the change of convention.

3.3. Connections between Classifier Performance and Covariance Matrix

The theory of the current section depends on the assumption that the base classifiers are mutually or second order conditionally independent. Classifiers that are nearly mutually conditionally independent may arise, for example, from experts with different technical backgrounds, or from algorithms that are based on different design principles or independent sources of information. Note that similar independence assumptions appear also in other works considering a setting similar to ours (Dawid and Skene, 1979; Parisi et al., 2014). For completeness, we next state the second order conditional independence assumption.

Assumption 1 *The ensemble of classifiers $\{g_i\}_{i=1}^M$ are second order conditionally independent. Therefore, for any $i \neq j \in \{1, \cdots, M\}$ and any sample k,*

$$P\left(R_i = r_{ik}, R_j = r_{jk}|y_k\right) = P_i(r_{ik}|y_k)P_j(r_{jk}|y_k). \quad (4)$$

We let Σ_2 represent the covariance matrix between base classifier, i.e.

$$\Sigma_2(i, j) = \mathbb{E}[(R_i - \mathbb{E}[R_i])(R_j - \mathbb{E}[R_j])], \quad (5)$$

for given classifiers i and j. We next demonstrate that the covariance matrix has a special decomposition under assumption 1.

Corollary 3 *The covariance matrix Σ_2 as defined in Equation (5) can be expressed as,*

$$\Sigma_2(i, j) = \begin{cases} \frac{N^2-1}{12} & \text{if } i = j \\ C(\rho)\Delta_i\Delta_j & \text{if } i \neq j \end{cases}$$

if Assumption 1 holds.

Proof Under Assumption 1 all the classifiers are mutually conditionally independent. Let $i \neq j$, then the result follows from Theorem 2 by letting $n = l = 2$. Next, let $i = j$ and recall that $\mathbf{r}$ is a vector in which each element represents the unique rank of each sample $k \in \{1, \ldots, N\}$. Therefore, the variance, $\Sigma_2(i, i)$, of $\mathbf{r}$ is that a uniform discrete distribution, $(N^2 - 1)/12$, which completes the proof of the theorem. ■

Note that the off-diagonal entry $\Sigma_2(i, j)$ contains information on the performances Δ_i and Δ_j of base classifiers i and j. An intuitive understanding of this result can be grasped as follows. For an arbitrary pair of base classifiers, not necessarily conditionally independent, their covariance can be decomposed into intraclass and interclass covariance. The intraclass covariance is often a manifestation of the similarities between the methodologies used in the classifiers, or the fact that base classifiers were trained on similarly distributed data. The interclass covariance, on the other hand, represents the agreement of ranking samples based on each latent class; hence, if two algorithms perform well, their interclass covariance will reflect that performance. It follows that if two base classifiers are conditionally independent, the intraclass covariance will vanish, making the measured covariance exclusively attributable to interclass covariance which in turn reflects the performance of the base classifiers.

Inspection of Corollary 3 motivates a strategy for estimating each base predictor performance metric, Δ_i, from the covariance matrix. Here we see that Σ_2 can be decomposed as

$$\Sigma_2 = R - diag(R) + \frac{N^2-1}{12} I, \tag{6}$$

where I is the identity matrix and $R := \lambda_c v v^T$ is a rank-one matrix, with

$$\lambda_c := C(\rho) \left\| \Delta \right\|_2^2, \tag{7}$$

where $\|\Delta\|_2$ denotes the euclidean norm of the vector $\Delta := [\Delta_1, \cdots, \Delta_M]^T \in \mathbb{R}^M$, and v is the unit norm vector with entries defined as

$$v_i := \frac{\Delta_i}{\sqrt{\sum_{j=1}^M \Delta_j^2}}. \tag{8}$$

Note that λ_c is dependent on ρ which is unknown to us in our unsupervised setup. However, as we will see in the next section, SUMMA only needs an estimate of v_i, which is a quantity proportional to the estimated performance Δ_i of each base classifier i, to calculate the ensemble classifier. However, estimation of ρ is necessary to estimate Δ_i and from it the AUC of base classifier i as will be shown in Theorem 4. In Section 6, we present a way to estimate ρ.

It follows from Corollary 3 and the assumption that at least 3 base classifiers are better than random that $3 \leq M$ is a sufficient condition for estimating the quantities v_i's. To see why we need the requirement of $3 \leq M$, observe that we have $(M(M-1)/2)$ known off-diagonal elements of Σ_2 but M unknowns (v_i for $i = 1, \cdots, M$), so if $M < 3$ the system is underdetermined. Moreover, note that if only a single classifier i performs better than random, and the rest perform randomly, i.e.

$$v_i > 0, \quad v_j = 0 \quad \text{for } j \neq i,$$

then the off-diagonal entries of Σ_2 are exactly equal to zero in which case it is impossible to infer v_i. Similarly, if only two classifiers $v_i > 0$ and $v_j > 0$ are performing better than random, then we have again two unknowns but only one known non-zero off-diagonal element. Therefore, throughout the paper, we assume that at least three classifiers perform better than random. Under this assumption, the off-diagonal elements of the covariance matrix are equal to that of the rank-one matrix R whose unique eigenvector is proportional to the vector of individual classifier performances.

Although Δ is both an intuitive and statistically principled measure of classifier performance, it is the expected performance metric over all the realizations in our probability space Ω and it can not be calculated explicitly as the probability distribution $P_i(r|y)$ is not readily available. For our results to be applicable in practice, we will use the unbiased empirical estimator of Δ_i defined as

$$\widehat{\Delta_i} = \frac{1}{N_0} \sum_{\{k|y_k=0\}} r_{ik} - \frac{1}{N_1} \sum_{\{k|y_k=1\}} r_{ik} = \widehat{R}_{i|y=0} - \widehat{R}_{i|y=1}, \tag{9}$$

where $T \in \Omega$ is a single realization from the probability space Ω, r_{ik} is the empirical rank assigned to sample k by classifier i by ordering the confidence scores $\{g_i(x_i)\}_{i=1}^N$ in descending order, and $\widehat{R_{i|y}}$ denotes the empirical conditional sample mean of rank assignment. We finish this section by showing the relation of $\widehat{\Delta_i}$ to another canonical empirical measure of performance for base classifier i, namely its AUC (Marzban, 2004).

Theorem 4 *Given a ranked list of predictions and the corresponding sample class, $\{(r_{ik}, y_k)\}_{k=1}^N$, where r_{ik} is the rank assigned to the sample k by classifier i and y_k is the true class of sample k, we have the following equivalence*

$$AUC_i = \frac{\widehat{\Delta_i}}{N} + \frac{1}{2}, \tag{10}$$

where AUC_i is estimated using the rectangle rule.

Proof For the definition of AUC and the proof see Appendix C. ■

It is easy to compute the empirical performance $\widehat{\Delta}$ using (9) if the labels associated with each sample are available. However, in our unsupervised setting, we only observe r_{ik} which is the rank assigned to sample k by classifier i. Indeed, one of the main contributions of the current manuscript is to infer $\widehat{\Delta}$ without access to class labels. For that purpose, we will use the relation between the covariance matrix Σ_2 and Δ derived in Section 3.2. The next section proposes two algorithms for this inference task.

4. Methods for Estimating Performances from the Covariance Matrix

As we stated in the previous section, due to the non-availability of Σ_2, we use $\widehat{\Sigma_2}$, the empirical covariance matrix, which is an unbiased estimator of the covariance matrix Σ_2, to infer the performance of base classifiers. In order to calculate $\widehat{\Sigma_2}$, we only require the rank predictions by each classifier. We then calculate an estimate $\widehat{v_i}$ of v_i from $\widehat{\Sigma_2}$. For the sake of completeness, we will next define the empirical covariance matrix, $\widehat{\Sigma_2}$. As before, let $T \in \Omega$ be given, and an ensemble of classifiers $\{g_i\}_{i=1}^M$ ranks samples $\{x_k\}_{k=1}^N$ using $g_i(x_k)$ and assigns rank r_{ik} to sample k. Then, the empirical covariance matrix $\widehat{\Sigma_2}$ is given by

$$\widehat{\Sigma_2}(i,j) = \frac{1}{N} \sum_{k=1}^N \left(r_{ik} - \frac{N+1}{2} \right) \left(r_{jk} - \frac{N+1}{2} \right).$$

In the next two subsections, we briefly describe two methods to calculate an estimator $\widehat{v_i}$ of v_i from $\widehat{\Sigma_2}$. Our first approach formulates this task as a semi-definite program (SDP) which then can be solved using any SDP solver. Our second strategy is coming from the recent literature on the famous matrix completion problem, see e.g. (Candès and Recht, 2009; Barber and Ha, 2018; Cai et al., 2010; Jain et al., 2010). It is an iterative method that only requires calculating the largest singular value and associated singular vector at each iteration.

4.1. Semi-Definite Programming

In this section we present an SDP approach to estimate $\widehat{v}_i$ of v_i from $\widehat{\Sigma_2}$. Here we will only describe the main idea and leave the details to the Appendix A. Let's assume that the covariance matrix Σ_2 is known and using the notation of equation (6), we will write it as $\Sigma_2 = R + D_{\Sigma_2}$, where $D_{\Sigma_2} := -diag(R) + \frac{N^2-1}{12}I$ is a diagonal matrix. Therefore, for $D = D_{\Sigma_2}$ the matrix R is a feasible point for the following optimization problem,

$$\min_D rank(\Sigma_2 - D) \quad \text{s.t.} \quad D(i,j) = 0 \quad \text{for} \quad i \neq j, \quad \text{and} \quad \Sigma_2 - D \succeq 0, \tag{11}$$

where the positive semi-definite requirement on $\Sigma_2 - D$ is due to the non-negativity of the eigenvalues of the intended target $R = \lambda_c vv^T$ with λ_c specified in Equation (7). We show in Lemma 10 (see Appendix A) that R is the unique solution to the problem (11).

The optimization problem described in (11) is intuitive, but it is difficult to solve in practice. This is because the rank function is not convex (Candès and Recht, 2009). Indeed, the rank minimization problem for an arbitrary matrix is NP-hard (Jain et al., 2010). To circumvent this shortcoming, we chose to optimize the convex relaxation of the rank function, namely the nuclear norm, as is done in the matrix completion literature (Candès and Tao, 2010). For a given matrix A the nuclear norm is defined as,

$$\|A\|_* = \sum_{i=1}^{M} \sigma_i(A), \tag{12}$$

where $\sigma_i(A)$ represents the ith singular value of A. Accordingly, the optimization problem in (11) can be relaxed to the following SDP,

$$\min_D \quad \|\Sigma_2 - D\|_* \quad \text{s.t.} \quad D \text{ is diagonal, and} \quad \Sigma_2 - D \succeq 0. \tag{13}$$

SDP is an active area of research, with applications in control theory and signal processing (Vandenberghe and Boyd, 1999), and many efficient solvers are readily available. Next we characterize the solutions of the semidefinite program in Equation (13).

Theorem 5 *Let Σ_2 be the covariance matrix, then the optimization problem in Equation (13) has the unique solution D_{Σ_2}, provided that*

$$v_i^2 < \sum_{j \neq i} v_j^2, \quad \forall i \in \{1, 2, \ldots, M\}. \tag{14}$$

Proof Let $Q = \Sigma_2$ in Theorem 11. ■

The inference of the vector v from the covariance matrix Σ_2 using Theorem 5 is unique up to its sign. Without further assumptions it is impossible to determine the sign of each coordinate of v, and hence the performance of each base classifier. In this manuscript, we assume that all base classifiers have adopted the right convention that lower (respect. higher) ranks correspond to samples predicted to be in the positive (respect. negative) class and that the majority of them perform better than random. As a result v should have more positive entries than negative entries. Consequently, we solve the ambiguity between v and $-v$ based upon which of the two choices has the greatest number of positive entries.

Up to this point we have assumed that the covariance matrix Σ_2 was known. However, in practice we only observe the sample covariance matrix $\widehat{\Sigma_2}$. To obtain the estimator $\widehat{R}$ we solve problem (13) using $\widehat{\Sigma_2}$, and interpret the vector associated with the leading singular value as the estimator $\widehat{v}$ such that $\widehat{R} = \widehat{\sigma_1}\widehat{v}\widehat{v}^T$.

4.2. An Iterative Approach

The iterative algorithm we present in this section and its convergence are inspired from the matrix completion literature and more details can be found in the recent papers (Barber and Ha, 2018; Jain et al., 2010). Similar to the previous section, we first assume that Σ_2 is known to us. Before presenting the iterative algorithm, let us define the set $\omega = \{(i,j) : i \neq j \in \{1, \cdots, M\}\}$. Then, using the notation of Barber and Ha (2018), we can formulate the following matrix completion problem:

$$\min_X \frac{||P_\omega(X) - P_\omega(\Sigma_2)||_F^2}{2} \quad s.t. \quad X \in C = \{X \in R^{M \times M} : rank(X) = 1, X = X^T, X \succeq 0\}, \tag{15}$$

where $||A||_F = \sqrt{\text{Tr}(AA^T)}$ is the Frobenius norm of matrix A and the linear operator $P_\omega(X)$ is defined as

$$P_\omega(X) = \begin{cases} X_{ij} & \text{if } (i,j) \in \omega \\ 0 & \text{if } (i,j) \notin \omega. \end{cases} \tag{16}$$

Again, it is obvious that $R = \Sigma_2 - D_{\Sigma_2} \in C$ and $||P_\omega(R) - P_\omega(\Sigma_2)||_F^2 = 0$, as such R is a minimizer of the optimization problem in (15). Moreover, under the assumptions of Lemma 10, R is the unique minimizer. Therefore, by solving the optimization problem in (15), we can recover the unknown vector v. The optimization problem in (15) is exactly the matrix completion problem presented in Barber and Ha (2018), with the additional fact that the matrix R is of rank-one. From Theorem 3 of Barber and Ha (2018), under the assumption that R has more than one element different from zero, which is the same as the condition $v_1 > 0, v_i = 0 \; \forall i > 1$, the following gradient descent algorithm converges to R:

$$\begin{aligned} Y_t &= X_t - \nabla\left(\frac{1}{2}||P_\omega(X_t) - P_\omega(\Sigma_2)||_F^2\right) \\ X_{t+1} &= P_C(Y_t),, \end{aligned} \tag{17}$$

where $P_C(Y_t)$ is the projection of Y_t into the set C. First note that

$$\nabla\left(\frac{1}{2}||P_\omega(X_t) - P_\omega(\Sigma_2)||_F^2\right) = P_\omega(X_t) - P_\omega(\Sigma_2),$$

and from Eckart - Young - Mirsky Theorem (Eckart and Young, 1936), we have

$$P_C(Y_t) = y_1(Y_t)u_1u_1^T,$$

where $y_1(Y_t)$ is the largest singular value of Y_t and u_1 is the corresponding singular vector.

Since we do not observe Σ_2 in practice, we will use the empirical covariance matrix $\widehat{\Sigma_2}$ with the gradient descent algorithm given in (17) to calculate $\widehat{v}$ which is an estimator of v. The pseudo-code of the iterative algorithm is given in Algorithm 1.

Algorithm 1 Find rank 1 matrix from off-diagonal observations of covariance matrix

Given $\widehat{\Sigma_2}$, fix $\epsilon > 0$, $t = 1$ and let $\lambda(0) = u(0) = 0$, $\lambda(1), u(1) \leftarrow SVD(\widehat{\Sigma_2})$, where $\lambda(1)$, $u(1)$ represent the largest singular value and corresponding singular vector of $\widehat{\Sigma_2}$.
while $|\lambda(t) - \lambda(t-1))| > \epsilon$ **do**
  $Y(t) \leftarrow \widehat{\Sigma_2} - diag(\widehat{\Sigma_2}) + diag(\lambda(t)u(t)u(t)^T)$
  $t = t + 1$
  $\lambda(t), u(t) \leftarrow SVD(Y(t))$
return $[\ \widehat{\lambda}_c = \lambda(t), \widehat{v} = u(t)]$

5. Strategy for Unsupervised Multiple Method Aggregation: SUMMA

In the previous section, we showed how to estimate v_i, a quantity proportional to the performance of each base classifier, without knowing the labels associated with each sample. In this section, we develop a meta-learner that infers each sample's latent class and produces an aggregate ranking of samples using a weighted sum of base classifier rank predictions. As in the seminal work of Dawid and Skene (1979) and subsequent work by others (Nitzan and Paroush, 1982; Parisi et al., 2014), we cast the class inference task as a maximum likelihood estimation (MLE) problem.

As in the previous section, assume that a realization $T = \{(x_1, y_1), \cdots, (x_N, y_N)\} \in \Omega$ is given where the labels associated with samples are not available to us. The task at hand is to find the most likely class label, y_k, associated with each sample k, when the only available data are the rank predictions by conditionally independent base classifiers. Formally, the maximum likelihood estimate of y_k is:

$$\widehat{y_k}^{\text{MLE}} = \underset{y}{\text{argmax}} \left\{ \sum_{i=1}^{M} \log \left(P_i \left(r_{ik} | y \right) \right) \right\}.$$

By application of Bayes' Theorem, we can write $P_i\left(r_{ik}|y\right) = P_i\left(y|r_{ik}\right) P(r_{ik})/P(y)$. Using that the prior probability for the ranks is $P(r_{ik}) = 1/N$ and the prior probability for being in the positive class is the class prevalence (ρ for class $y = 1$ and $1 - \rho$ for class $y = 0$), the MLE can be equivalently written as,

$$\widehat{y_k}^{\text{MLE}} = \Theta \left\{ M \log \left(\frac{1-\rho}{\rho} \right) + \sum_{i=1}^{M} \log \left(\frac{P_i\left(y_k = 1 | r_{ik}\right)}{1 - P_i\left(y_k = 1 | r_{ik}\right)} \right) \right\}. \tag{18}$$

The central challenge in applying the MLE is that the functional form of the probability distribution $P(y = 1|r)$ is *a priori* unknown. We are then faced with two choices: 1) *a priori* assume the functional form of the probability distribution $P(y = 1|r)$, or 2) infer a distribution. If we were to *a priori* assume a distribution, we would in effect be biasing the MLE and the dependence of the ensemble algorithm on the base classifier performance Δ. This is problematic, because we have no reason to pick one distribution over another. A more principled approach would be to infer a distribution using information that is available to us and no more. The maximum entropy methodology for inferring distributions performs such a task (Jaynes, 1957). In short, by choosing the maximum entropy distribution we

are selecting the distribution which reproduces known statistical quantities and is otherwise maximally noncommittal to the remaining unknown moments. In our case we will assume that we have inferred the value of $\rho = N_1/N$ and the difference of conditional means Δ, which will serve as constraints for the maximum entropy calculation. In the proceeding Lemma, we derive the maximum entropy distribution for $P(y = 1|r)$ given those constraints.

Lemma 6 *The functional form of the maximum entropy probability distribution of the latent class label given the rank for a sufficiently weakly predictive classifier is approximated as*

$$P_i(y_k = 1|r_{ik}) = \left(1 + e^{\frac{12\Delta_i}{N^2-1}\left(r_{ik} - \frac{N+1}{2}\right) + \log\left(\frac{N-N_1}{N_1}\right)} \right)^{-1}.$$

Proof For the derivation see Appendix D. ■

Although we proved the above lemma for weakly predictive classifiers, we observed empirically through extensive simulations that the above formula is still a good approximation beyond that limit. Next we apply the maximum entropy probability distribution of Lemma 6 to derive a maximum likelihood estimator of each sample's latent class label.

Theorem 7 *The maximum likelihood estimator (MLE) of sample k is given as,*

$$\widehat{y_k}^{MLE} := \Theta \left\{ \sum_{i=1}^{M} v_i \left(\frac{N+1}{2} - r_{ik} \right) \right\}, \tag{19}$$

with Θ representing the unit step function.

Proof Applying Lemma 6 to our maximum likelihood estimator in Equation (18),

$$\begin{aligned} \widehat{y_k}^{\text{MLE}} &= \Theta \left\{ M \log \left(\frac{1-\rho}{\rho} \right) - \sum_{i=1}^{M} \left[\frac{12\Delta_i}{N^2-1} \left(r_{ik} - \frac{N+1}{2} \right) + \log \left(\frac{N-N_1}{N_1} \right) \right] \right\} \\ &= \Theta \left\{ \frac{12}{N^2-1} \sum_{i=1}^{M} \Delta_i \left(\frac{N+1}{2} - r_{ik} \right) \right\} \end{aligned}$$

and recall from Equations (7, 8) that $\Delta_i = \sqrt{\frac{\lambda_c}{\rho(1-\rho)}} v_i$. Then by substitution,

$$y_k^{\text{MLE}} = \Theta \left\{ \frac{12}{N^2-1} \sqrt{\frac{\lambda_c}{\rho(1-\rho)}} \sum_{i=1}^{M} v_i \left(\frac{N+1}{2} - r_{ik} \right) \right\} \tag{20}$$

where the terms preceding the sum have no influence to the image of the argument under the unit step function, and consequently may be ignored, which completes the proof. ■

Note that by our setup $\mathbb{E}[R_i] = (N+1)/2$, and also the rank assignment r_{ik} are known. Therefore, in order to calculate the maximum likelihood estimate associated with each sample k, we only need to estimate v_i. For that we will use the empirical estimate $\hat{v}_i$

calculated by one of the two methods we proposed in Section 4. Then, the SUMMA estimate of y_k is given as

$$\widehat{y_k}^{SUMMA} := \Theta\left\{\sum_{i=1}^{M} \hat{v}_i \left(\frac{N+1}{2} - r_{ik}\right)\right\}.$$

As the quantity $\sum_{i=1}^{M} \hat{v}_i \frac{N+1}{2}$ is the same for each sample k, the samples can be ranked using the SUMMA score, defined as

$$c_k^{SUMMA} = -\sum_{i=1}^{M} \hat{v}_i r_{ik}, \tag{21}$$

which can be interpreted as the score assigned to sample k by SUMMA. The SUMMA score defined above is closely related to a more popular quantity in rank aggregation literature, namely the Borda count (Nitzan and Rubinstein, 1981). Borda count is a very popular aggregation technique in social choice theory (Sen, 1986) as well as computer science especially in the field of information retrieval (Subbian and Melville, 2011). Borda aggregation is equivalent up to the sign to ranking samples by assigning each ranker equal weight, i.e. letting $\widehat{v}_i = 1/M$ in (21). We would like to caution that in the pure rank aggregation problem the task in hand is to rank samples based on the preference of the ranker and there is usually a true but unknown ranking of samples that we want to estimate. However, in the information retrieval literature as well as the SUMMA, we rank the samples according to their probability belonging to the positive class and the aim is to find the true but unknown class label associated with each sample. Although in social endeavors such as democratic elections it is preferred to assign equal weight to each ranker, in machine learning it is desirable to assign each ranker a weight proportional to its performance. In the information retrieval literature, this weighted ranking scheme is called the weighted Borda-count (Aslam and Montague, 2001), where it is shown superior performance over Borda-count as well as other popular rank aggregation methods. However, in the literature, available algorithms for assigning weights in ranking problems require labeled data (supervision) (Aslam and Montague, 2001; Liu et al., 2010), or at least some semi-supervision (Balsubramani and Freund, 2015) or some prior information about each classifier's performance such as the public leaderboard scores in Kaggle (Sun et al.). The SUMMA rank aggregation is a weighted Borda-counting where the weights of each ranker, which are proportional to their classification performance, are estimated using unlabeled data. Therefore, the SUMMA score is closely related to rank aggregation literature and produces a robust ranking of samples in an unsupervised setting.

Apart from re-ranking a given set of samples and ranking classifiers based on their performance using a given matrix of ranked predictions, SUMMA can also use this matrix to learn an ensemble classifier to further classify an unseen sample. For this assume that we are given a dataset $T \in \Omega$ and an ensemble of classifiers $\{g_i\}_{i=1}^{M}$. Actually, in this situation the dataset T will serve as an unlabeled training set to train an ensemble classifier. Suppose now we are given a new instance $(\psi, y_\psi) \in X \times \{0,1\}$, where we want to get an estimate of y_ψ. We can achieve this by running SUMMA on the combined dataset of $N+1$ samples. As in the supervised setting, the quality and quantity of the samples in the training set would greatly affect the performance of the ensemble classifier.

6. Estimation of AUC of the Base Classifiers using the Third Order Covariance Tensor

The results of Section 4 help us estimate the vector v whose entries are proportional to the performance of individual classifiers. Moreover, from (21) we see that this estimate, $\widehat{v}$ was sufficient to form the SUMMA ensemble classifier. However, it is not possible to estimate the actual performance, i.e. the AUC for each classifier, from the covariance matrix without *a priori* knowledge of the sample class prevalences. To address this shortcoming, we develop a strategy for estimating the prevalence from the third order covariance tensor of unlabeled rank data. Our approach is similar to that of Jaffe et al. (2015); however, we have extended the iterative algorithm presented in Section 4 by using the generalization of singular value decomposition to tensor decomposition, (Karami et al., 2012). Given three classifiers i, j and k, the third order covariance tensor is defined as

$$\Sigma_3(i,j,k) = \mathbb{E}\left[\Big(R_i - \mathbb{E}[R_i]\Big)\Big(R_j - \mathbb{E}[R_j]\Big)\Big(R_k - \mathbb{E}[R_k]\Big)\right]. \tag{22}$$

Previously, we assumed that the rank predictions by base classifiers were mutually conditionally independent. For the theory in this section, we will extend this to triplets and will make the following assumption.

Assumption 2 *The ensemble of classifiers $\{g_i\}_{i=1}^{M}$ are third order conditionally independent. Therefore, for any $i \neq j \neq l \in \{1, \cdots, M\}$ and sample k,*

$$P\left(R_i = r_{ik}, R_j = r_{jk}, R_l(x_k) = r_{lk} | y_k\right) = P_i(r_{ik}|y_k) P_j(r_{jk}|y_k) P_l(r_{lk}|y_k), \tag{23}$$

for any $y_k \in \{0,1\}$.

Using this observation we present the following corollary of Theorem 2.

Corollary 8 *Under Assumption 2, for any $i \neq j \neq l$ the covariance tensor is given as*

$$\Sigma_3(i,j,l) = C(\rho)(2\rho - 1)\Delta_i \Delta_j \Delta_l. \tag{24}$$

Proof Let i, j and l be integers in the set $\{1, 2, \ldots, M\}$ such that $i \neq j \neq l$. Then the result trivially follows from Theorem 2 with $n = 3$. ■

Corollary 8 shows that the covariance tensor Σ_3 is off-diagonal rank-one given by $\Sigma_3 \approx \lambda_t v \otimes v \otimes v$, where $\lambda_t := \Big((C(\rho)(2\rho-1)\Big)||\Delta||_2^3$. Recall from Equation (7), $\lambda_c = C(\rho)\,\|\Delta\|_2^2$. Next, let $\beta := \lambda_t^2/\lambda_c^3$ and observe that

$$\begin{aligned} \beta &= (2\rho-1)^2/(C(\rho)) \\ &\implies \rho^2(\beta+4) - (\beta+4)\rho + 1 = 0. \end{aligned} \tag{25}$$

Therefore we can explicitly solve for ρ

$$\rho = \frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{\beta}{\beta+4}} \tag{26}$$

where we use the + sign if λ_t is negative and the − sign otherwise. By rearranging the terms in Equation (25) we find that

$$C(\rho) = \frac{1}{\beta+4} \implies \|\Delta\| = \sqrt{\lambda_c(\beta+4)}, \tag{27}$$

which gives us the norm of Δ.

Again since Σ_3 is not available in practical situations, we will use the empirical covariance tensor $\widehat{\Sigma}_3$ defined as

$$\widehat{\Sigma}_3(i,j,l) = \frac{1}{N}\sum_{k=1}^{N}\left(r_{ik} - \frac{N+1}{2}\right)\left(r_{jk} - \frac{N+1}{2}\right)\left(r_{lk} - \frac{N+1}{2}\right).$$

In order to find an estimator of λ_t from the third order statistics, we extend our iterative algorithm for decomposing the covariance matrix by using tensor SVD (tSVD) ((Kolda and Bader, 2009)). Pseudo-code for the generalized iterative algorithm is given in Algorithm 2.

Algorithm 2 Find rank 1 matrix from off-diagonal observations of covariance tensor

Given $\widehat{\Sigma_3}$, fix $\epsilon > 0$, $t = 1$ and let $\lambda(0) = u(0) = 0$, $\lambda(1), u(1) \leftarrow \text{tSVD}(\widehat{\Sigma_3})$,
where $\lambda(1)$, $u(1)$ represent the largest singular value and corresponding singular tensor of $\widehat{\Sigma_3}$.
while $|\lambda(t) - \lambda(t-1)| > \epsilon$ **do**
  $Y(t) = \widehat{\Sigma}_3 - diag(\widehat{\Sigma}_3) + diag(\lambda(t)u(t)u(t)^T)$
  t=t+1
  $\lambda(t), u(t) \leftarrow \text{tSVD}(Y(t))$
return $[\widehat{\lambda}_t = y_1(X_t), \widehat{v} = u_1]$

Algorithm 2 gives us an estimate of the singular value of the covariance tensor λ_t which is denoted as $\widehat{\lambda}_t$. Note that we can estimate both λ_c and λ_t using the iterative algorithms Algorithm 1 and Algorithm 2, respectively. This allows us to calculate an estimate of β which in turn gives us an estimate of $\|\Delta\|_2$ from (27). Knowledge of $\|\Delta\|$ allows us to estimate the prevalence of each class label and for each i base classifiers to compute an estimate of Δ_i using equation (7). Furthermore, we can use this estimate of Δ_i to compute the AUC for each i classifier using the closed form formula given in Theorem 4. Apart from the iterative approach we proposed, other methods for estimating ρ based on a restricted likelihood approach (Jaffe et al., 2015) or on spectral decomposition (Jain and Oh, 2014) exists albeit with additional assumption such as higher order conditional independence.

7. Examples of Applications of SUMMA

In this section we apply the SUMMA methodology and assess its performance in different example settings, including i) synthetic data, ii) predictions submitted to a crowd-sourced challenge and iii) several classification problems in different domains using datasets available from the UCI Machine Learning Repository (Lichman 2013). For the computations in this section we have used the R language on a personal laptop which has 4 computational cores and 16GB of RAM. Running the SUMMA algorithm is very fast (less than 5 minutes in

the worst case); and apart from the calculation of the covariance matrix and tensor, it is unaffected by the number of samples.

In the instances in which classifiers return the same score for different samples, their corresponding ranks will be tied. In those cases we will break the ties by assigning those samples random but unique ranks ranging between the immediately higher and lower untied ranks.

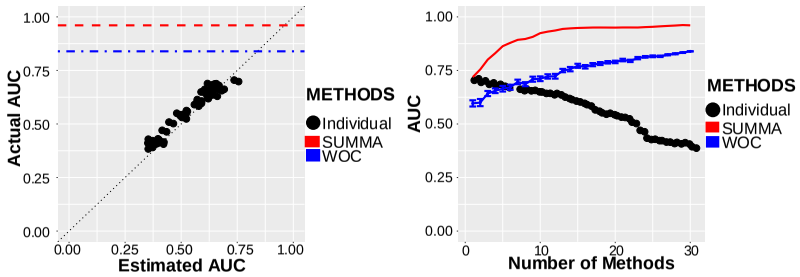
7.1. Synthetic Data Examples

In this section we use synthetic data to illustrate the ability of SUMMA to infer the AUC of individual methods as well as the performance improvements obtained using the SUMMA ensemble. We investigate the influence of the number of methods, the number of samples, and class prevalence on SUMMA performance. Each data set represents N sample rank predictions from M conditionally independent base classifiers. We generated synthetic predictions by producing random scores from two Gaussian distributions, each of which is used to simulate scores from the negative and positive class. The AUC of each base classifier was controlled by adjusting the parameters (mean and variance) of the respective class specific Gaussian distributions using the closed form formula of AUC for normally distributed class specific scores (Marzban, 2004):

$$AUC = \Phi\left(\frac{\mu_+ - \mu_-}{\sqrt{\sigma_+^2 + \sigma_-^2}}\right),$$

where μ_i and σ_i^2 denote the mean and variance for the positive ($i = +$) and negative ($i = -$) classes, respectively, and $\Phi(\cdot)$ is the standard normal cumulative distribution function. The conditional independence assumption between classifiers was satisfied by independently sampling from the Gaussian distributions associated with each base classifier. Once the samples were generated, we converted the scores to sample ranks using the convention that lower (higher) ranks correspond to samples predicted to be in the positive (negative) class.

For the first part of the analysis, we generated synthetic predictions for $M = 30$ base classifiers, with a total of $N = 1000$ samples of which 500 belongs to the positive class. We adjusted the parameters of the Gaussian distributions such that the distribution of base classifier AUC values was uniformly distributed between $(0.4, 0.74)$. Under this setting, Figure 1a shows that SUMMA reliably estimated the AUC of each base classifier with a correlation coefficient of 0.95 between the estimated and actual AUC. In addition, Figure 1a shows that the SUMMA ensemble (red, AUC=0.95) out-performs the best individual classifier (AUC=0.74), and the WOC ensemble (blue, AUC=0.89), which aggregates predictions by averaging the sample ranks of the base classifiers. Next we investigate the effect of changing the number of base classifiers. Figure 1b shows how the performances of the SUMMA and WOC ensembles change with the addition of extra classifiers. Here, we estimated the performance of base classifiers using the covariance of all 30 classifiers but constructed the nth SUMMA ensemble by aggregating the top n performing classifiers where we rank classifiers based on their performance as estimated by SUMMA. Unlike SUMMA, in this unsupervised setting the WOC ensemble has no knowledge of the performances of each base classifier. Hence, for the WOC ensemble we randomly sub-selected n methods

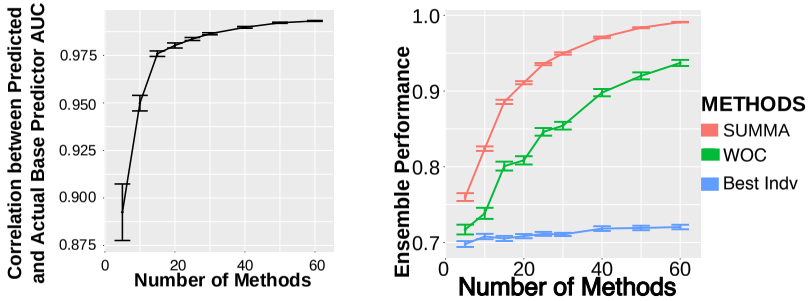


(a) Comparison between the SUMMA inferred AUC of the base classifiers, and their actual AUC

(b) AUC of the SUMMA ensemble as a function of the number of base classifiers

Figure 1: Validation and analysis of the SUMMA results with simulated data for $N = 1000$ samples and positive class prevalence $\rho = 0.5$.

out of 30 base classifiers thirty times and calculated the mean and standard error of the mean AUC associated with the WOC ensemble. Figure 1b shows that the SUMMA ensemble increases in performance more readily as classifiers are added in the ensemble and saturates at a higher AUC than the WOC ensemble (Figure 1b). Moreover, we see that the SUMMA performance saturates at $n = 15$ classifiers suggesting that we can achieve a performance close to that of the full ensemble with only 15 classifiers (or 50% of all the base classifiers). We leave the investigation of finding an optimal number of classifiers for the ensemble construction for future research.



(a) Correlation between the SUMMA inferred AUC of the base classifiers and their actual AUC, as a function of the number of base classifiers

(b) AUC of the SUMMA ensemble as a function of the number of base classifiers

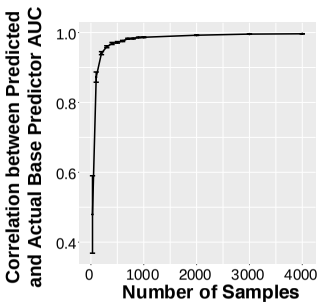
Figure 2: The dependence of the SUMMA results with the number of base classifiers for $N = 1000$ samples and $\rho = 0.5$.

In the second part of the analysis, we empirically tested the dependence of SUMMA on varying number of classifiers, samples, and class prevalence. In each case we change only one of the simulation parameter from their default values of, $M = 30$ base classifiers, $N = 1000$ samples, and the prevalence of class 1, $\rho = 1/2$. The error bars in the figures represents the standard error of the mean for 30 repeated experiments.

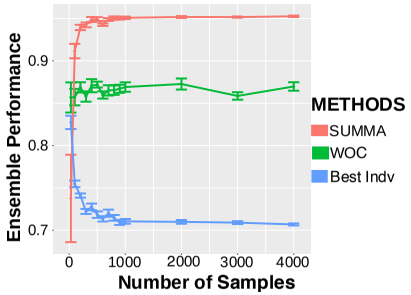
First, we tested the influence of the number of classifiers by simulating predictions and applying SUMMA to ensembles composed of $M = \{5, 6, 7, \ldots, 60\}$ base classifiers. This experiment is different from the analysis of Fig 1.b, because here we run SUMMA using $M = \{5, 6, 7, \ldots, 60\}$ (Fig. 2a) base classifiers as opposed to using the same covariance matrix of all 30 base classifiers as done in the experiments leading to Fig 1.b. In other words, the inference of weights is done for each set of M classifiers as opposed to the earlier analysis. Intuitively, inferring Δ for larger covariance matrices should become more accurate because the number of equations grows faster, $M(M-1)/2$, than the number of parameters Δ_i, M. Indeed, this intuition is confirmed in Figure 2. We see that the correlation between the predicted versus actual AUC of base classifiers inferred by SUMMA for $M = 5$ is $\approx$0.87, and increases readily to $\approx$0.97 for $M \geq 15$. We then tested how the number of classifiers affected the performance of the corresponding SUMMA ensemble. In Figure 2b we find that the SUMMA ensemble outperforms the WOC and best individual performing classifier in the ensemble. Moreover, as the number of classifiers increases, SUMMA's performance increases towards the perfect $AUC = 1$ even if the best individual classifier AUC never exceeds 0.75.

Next, we tested how the number of samples affect the performance of SUMMA. From the law of large numbers, one would expect that the performance of SUMMA would improve with the number of samples. This is simply because the error in estimating the covariance matrix elements decreases with N which in turn results in a more reliably estimate of the AUCs of base classifiers. Indeed, in Figure 3a we see that the correlation between the SUMMA inferred AUC and the actual AUC of base classifiers monotonically increases from $\approx$0.57 to $\approx$1 when increasing N from 30 to 4000 samples. Furthermore, the AUC of the SUMMA ensemble is also increasing with the number of samples mainly due to the fact that we can estimate the performance of base classifiers more reliably with increasing sample size (Figure 3a). Note that the WOC performance is not much affected by the sample size. When we have a very small number of samples ($N <= 50$) the uncertainty in the estimated performances of base classifiers is large enough to negatively influence the SUMMA ensemble performance as shown in Figure 3b. In such cases the WOC ensemble can be preferred to SUMMA. However, when ($N > 50$) the SUMMA ensemble performs significantly better than the WOC ensemble.

Lastly, we tested the influence of class prevalence on the performance of SUMMA. The accuracy of the SUMMA inferred AUC decreases for $\rho \leq$0.2 or $\rho \geq$0.8 (see Figure 4a). This is an intuitive result if we think that as the class prevalence moves to the extremes, the minority class has less samples, which results in larger errors in the estimation of minority class averages. The result is that the performance of the SUMMA ensemble in data with highly imbalanced classes is worse than in the case of balanced classes, albeit not by much as shown in Figure 4b. In this context it is relevant to note that high class imbalance also has a negative effect in the performance of supervised ensemble methods such as stacking (Padmaja et al., 2007).

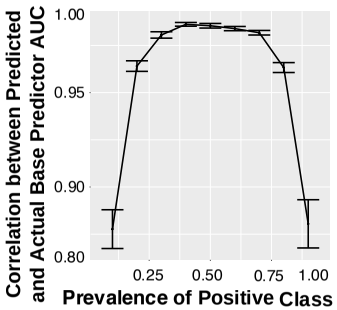


(a) Correlation between the SUMMA inferred AUC of the base classifiers and their actual AUC, as a function of the number of samples

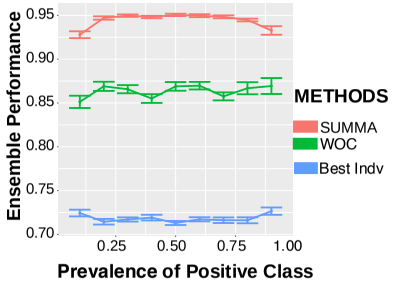


(b) AUC of the SUMMA ensemble as a function of the number of samples

Figure 3: The dependence of the SUMMA results with the number of samples for $M = 30$ base classifiers and positive class prevalence $\rho = 0.5$.



(a) Correlation between the SUMMA inferred AUC of the base classifiers and their actual AUC, as a function of the prevalence of the positive class



(b) AUC of the SUMMA ensemble as a function of the prevalence of the positive class

Figure 4: The dependence of the SUMMA results with the prevalence of the positive class ρ for $M = 30$ base classifiers and $N = 1000$ samples

7.2. Inference of genes targeted by a transcriptional regulator: The BCL6 DREAM Challange

Crowd-sourcing data competitions, such as Kaggle (www.kaggle.com) and the DREAM Challenges (www.dreamchallenges.org) (Prill et al., 2010; Saez-Rodriguez et al., 2016), have become effective at benchmarking a diverse array of machine learning methods while finding efficient solutions to challenging real life problems. The diversity of machine learning strategies applied by challenge participants in these competitions results in a large number of independent predictions for the same test set which can be aggregated into an ensemble prediction with methods such as SUMMA. In this section we will use one such competition, the DREAM BCL6 challenge (Stolovitzky et al., 2009) to test the performance of SUMMA. DREAM Challenges lend themselves for this task because after the finalization of the competitions, DREAM organizers share the gold standard labels associated with the test set as well as the collection of participants' algorithms and predictions. This allowed us to objectively evaluate the performance of SUMMA in a problem of biological interest.

In the BCL6 challenge, participants were asked to infer whether the activity of a given gene is regulated, or in biological jargon: is targeted, by the transcription factor BCL6. The participants were provided an unlabeled feature matrix consisting of micro array measurements, with each element representing the relative abundance of RNA transcripts corresponding to a specific gene and given a specific perturbation. Additionally, they were allowed to incorporate any additional data that could be of use. They were then asked to report the confidence scores, in rank order for 200 genes, indicating whether a gene was a target of BCL6. Hidden from the participants were the experimentally determined class labels for each gene, in which 53 of the 200 ($\rho = 0.265$) genes were determined to be targets of BCL6. This way of creating gold standard labels is typical in biology and is very expensive both in monetary value as well as human resources. Therefore, the BCL6 challenge is a perfect application of SUMMA where there is no labeled data readily available for training.

Eleven teams participated and submitted predictions to this challenge. Therefore, the input to SUMMA was a matrix of size 200 genes by 11 methods, the elements of which are confidence scores between 0 and 1. We used the gold standard labels created by the organizers to compare the performance of SUMMA to that of individual methods, as well as to other methods including the Spectral Meta Learner (SML) algorithm (Parisi et al., 2014), the WOC and an unsupervised classifier based on k-means clustering.

We first tested the ability of SUMMA to estimate the AUCs of individual classifiers. As seen in Figure 5, SUMMA could reliably estimate the performance of individual classifiers with a correlation of 0.96 between the inferred and actual AUCs. Figure 5 also shows the AUC of the SUMMA ensemble, obtained by ranking samples according to the SUMMA score c_k^{SUMMA} defined in (21). The SUMMA AUC (0.93) was significantly better than the WOC AUC (0.82) and the best individual method AUC (0.85). The success of the SUMMA ensemble in predicting the targets of BCL6 can be attributed in part to the fact that, due to the lack of labeled training data, challenge participants created a very diverse set of predictions based on biological interpretations of the data rather than, as is customary in other contexts, applying similar supervised models to the same training data for classifier generation. As a result the conditional independence assumption is not significantly violated, which allowed us to create relatively accurate estimates of the performances of base classifiers. Next we checked the effect of using the ranks rather than the confidence scores provided by the participants in the ensemble performance. For that we have used the AUCs of base classifiers inferred by SUMMA but formed the aggregate scores with the confidence scores generated by individual classifiers as opposed to converting them to ranks. This procedure resulted in an AUC of 0.83 as opposed to 0.93 obtained using SUMMA, which suggests a benefit in using rank transformation. The reason for this could be that, even

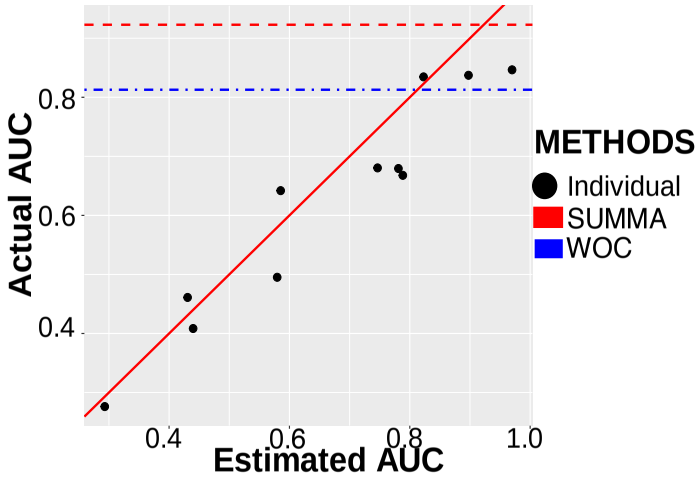


Figure 5: Comparison between the SUMMA inferred AUC on the BCL6 data and the actual AUC of the base 11 base classifiers

though confidence values are provided in the interval $[0, 1]$, it is likely that their distribution and interpretation is different in each prediction, which may have a negative impact in confidence score aggregation. Our results show that in situations in which we don't have control on the generation of confidence scores by classifiers or do not have labeled data to re-calibrate outputs of base classifiers, rank transformation allows for a robust way of normalizing confidence scores.

Next we investigate the classification performance of SUMMA. The results of this endeavor are summarized in Table 1. We calculated the balanced accuracy of SUMMA using the binary labels from the MLE estimate in (19) which resulted in a balanced accuracy (BA) of 0.82. For comparison we ran the SML algorithm in the same dataset by binarizing outputs of base classifiers. Since the challenge participants were not asked to provide a threshold, we binarized outputs of individual classifiers by using the natural threshold of 0.5 for the confidence. The SML had a BA of 0.80 whereas the best individual classifier had a BA of 0.79. We would also note that the second best method had a BA of 0.66 which shows that putting the right threshold is a challenging task that might lead to overfitting. As a further comparison, we also run k-means algorithm with $k = 2$ on the same dataset. In particular, we provided the matrix of confidence scores produced by base classifiers as input to the k-means algorithm. The $k-$means algorithm clustered the samples in to two clusters which we used to assign each sample a class label. The resulting classifier had a BA of 0.52 significantly lower than the other methods mentioned above.

Method	Balanced Accuracy
SUMMA	0.82
SML	0.8
Best Individual	0.79
Second Best Individual	0.66
k-means (k=2)	0.52

Table 1: The performance (balance accuracy) of ensemble methods and the two best individual methods used to solve the BCL6 DREAM Challenge

Name	# Features	# Samples	Reference	Prevalence of Minority Class
Bank Marketing	17	45211	(Moro et al., 2014)	0.11
Ionosphere	35	351	(Sigillito et al., 1989)	0.35
Mammographic Mass	6	830	(Elter et al., 2007)	0.48
Parkinsons	23	195	(Little et al., 2007)	0.24
Yeast	9	892	(Nakai and Kanehisa, 1991)	0.48

Table 2: Summary of Real World Data Sets from UCI Machine Learning Repository

7.3. Applying SUMMA to Real World Data in Diverse Domains

The main purpose of the analysis in this example is to study the robustness of SUMMA in typical real-life cases where our assumption of conditional independence is likely violated. For this purpose we trained a variety of base classifiers on the same training set and ran SUMMA on an independent test set. We applied the same procedure on five data sets coming from different fields (Table 2) taken from the UCI Machine Learning Repository (Lichman, 2013). More information about these five datasets can be found in Appendix E. With the exception of the Bank Marketing data, we divided each dataset into half, and used the first half to train the base classifiers and the second half to evaluate SUMMA. For the Bank Marketing data, which had 45,211 samples, we randomly selected 1000 samples for training and another 1000 samples for the evaluation. We restricted the sample size mainly to reduce computational burden of training base classifiers. We trained base classifiers using the R package caret (Kuhn et al., 2008b), which we chose for its ease of use, its inclusion of large diversity of popular classifiers, and its automatic layout for doing cross-validation (Kuhn et al., 2008a). However, we would like to emphasize that any base classifier such as the AUC maximizing algorithms OPAUC (Gao et al., 2013) can be used to form the SUMMA ensemble. We chose $M = 22$ base classifiers as shown in (Table 3), and used ten-fold cross validation for their training. We first tested whether SUMMA can reliably infer the AUC of each base classifier in the test set. In all the data sets considered, the correlation between the true AUC and the SUMMA inferred AUC of classifiers was above 0.75, as shown in Figure 6. This remarkable correlation in AUC values in these data sets is, however, lower than that obtained in the synthetic data (0.98) and the BCL6 challenge data

Name	Main Method	RLibrary
adaboost	Adaboost	fastAdaboost
avNNet	Model Averaged Neural Network	nnet
bayesglm	Bayesian Generalized Linear Model	arm
ctree	Conditional Inference Tree	party
earth	Multivariate Adaptive Regression Spline	earth
gbm	Stochastic Gradient Boosting	gbm
glm	Generalized Linear Model	stats
glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models	glmnet
J48	C4.5-like Trees	RWeka
Jrip	Rule-Based Classifier	RWeka
C5.0	Decision Trees and Rule-Based Models	C50
knn	k-Nearest Neighbors	kknn
LMT	Logistic Model Trees	RWeka
mlp	Multi-Layer Perceptron	RSNNS
nb	Naive Bayes	klaR
nnet	Neural network	nnet
rf	Random Forest	randomForest
rpart	Recursive Partitioning and Regression Trees	rpart
simpls	Partial Least Squares	pls
svmLinear2	Support Vector Machine with Linear Kernel	e1071
svmRadial	Support Vector Machine with Radial Kernel	kernlab
xgbLinear	eXtreme Gradient Boosting	xgboost
xgbTree	eXtreme Gradient Boosting	xgboost

Table 3: Machine learning methods used in this Section

(0.96). This is likely due to the conditional dependence between base classifiers. Table 4 shows the ranking of classifiers including SUMMA ensemble in terms of AUC. We can easily observe that classifiers that perform best in one data set do not necessarily perform well in other data sets. In fact, they can be one of the worst in other data sets. This is most likely due to the distributions of the data being different in different data sets and classifiers with different theoretical backgrounds are more suitable to be applied in one type of data than other. In comparison, SUMMA performs better than the best individual classifier in the Bank Marketing, Parkinsons and Yeast datasets, and second best in the mammographic masses and fifth in ionosphere data sets. Next, for two of the datasets, Bank Marketing and Yeast, we analyzed how the number of integrated classifiers affects the performance of SUMMA prediction by examining randomly sampled combinations of individual classifiers (Figures 7a and b respectively). SUMMA performs better than individual classifiers even when integrating small sets of individual predictions. Performance increases further with the number of integrated classifiers. For instance, for 15 randomly selected individual classifiers, the SUMMA ensemble performs better than the best amongst the 15 classifiers in 98% of the cases in the Bank Marketing data set and it ranks best in 80% of the cases and best or second best in 97% of the cases in the Yeast data set demonstrating the robustness of SUMMA. Table 5 shows the frequency with which SUMMA outperforms the WOC ensemble prediction in the Bank Marketing and Yeast data. For example, for the Bank Marketing data set if we combine random 10 teams, 99% of the times SUMMA performs better than WOC. For the Yeast data, SUMMA gives a better prediction than WOC in about 65% of the times.

We next compared the classification accuracy of SUMMA to individual classifiers. For that we used the binary label predictions output by caret and computed the balanced accuracy of each individual classifier. Table 4 shows the ranking of classifiers including SUMMA ensemble in terms of their balanced accuracies. Similar to the rank based outputs, SUMMA is better than the best in 3 of the 5 datasets and is the second best in the rest. Moreover, as before classifiers that perform best in one data set do not necessarily perform well in other data sets.

These results highlight the robustness of SUMMA even in real applications where the conditional independence assumption is not guaranteed. As noticed above, the best team in any one example could perform relatively poorly or even amongst the worst in other examples. However, SUMMA is amongst the top in all examples, suggesting that using SUMMA is a sound strategy for aggregation when many predictions to the same problem are available and one do not have any strong prior belief on the top performing algorithm.

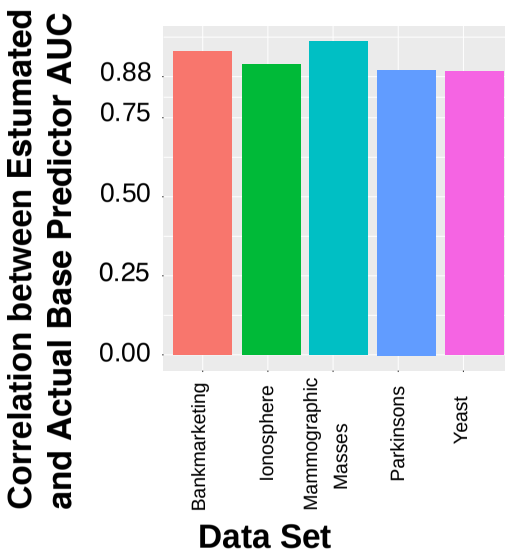
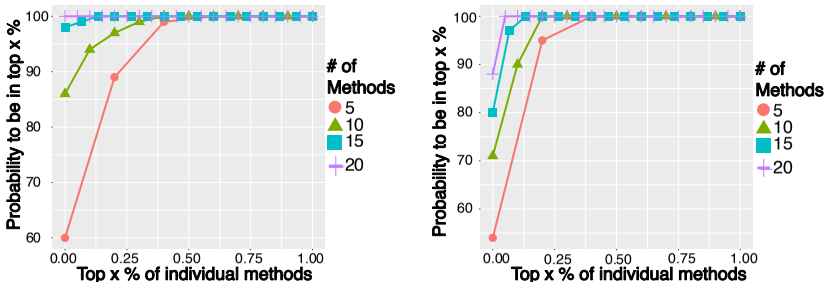


Figure 6: Correlation between the actual and estimated AUC of the base classifiers tested on the six UCI datasets considered in this paper.



(a) Bank Marketing Data (b) Yeast Data

Figure 7: Robustness of SUMMA with increasing number base classifiers (Y axis represents how often SUMMA is in the top x% of methods)

8. Conclusions

In this paper, we introduced a novel methodology for unsupervised ensemble learning: the Strategy for Unsupervised Multiple Method Aggregation (SUMMA). Under the assumption

Method	Bank Marketing	Ionosphere	Mammographic Mass	Parkinsons	Yeast
Earth	2	22	5	6	6
svmRadial	12	1	18	5	5
gbm	5	7	1	4	4
C5.0	8	8	7	2	21
rf	7	3	6	5	2
SUMMA	1	5	2	1	1

Table 4: Ranking of the different methods in each application domain. Only the classifiers that ranked first in at least one application are listed.

Number Classifiers	% SUMMA is better than WOC	
	Bank Marketing Data	Yeast Data
5	90	69
10	99	68
15	100	63
20	100	68

Table 5: Percentage of times SUMMA outperforms WOC.

Method	Bank Marketing	Ionosphere	Mammographic Mass	Parkinsons	Yeast
glm	2	14	12	9	12
svmRadial	15	1	11	7	10
simpls	12	10	1	2	11
rf	10	3	4	6	2
SUMMA	1	2	2	1	1

Table 6: Ranking of Methods using classification performance (Balanced Accuracy) in each application domain. Only the classifiers that ranked first in at least one application are listed.

that base classifiers assign ranks independently of each other to samples within each class (conditional independence assumption) we showed that the SUMMA algorithm can infer the AUC of base classifiers from the covariance of their rank predictions in the absence of labeled data. We then used the inferred AUCs as constraints in the context of a maximum entropy inference of the probability of the class of a sample given its rank. This probability was then used in a maximum likelihood estimation to derive a score that integrated the ranking assigned by each base classifier to each sample, which allowed us to predict the

class labels most likely associated with each sample, as well as a new integrated ranking for the samples from which we can compute the ensemble AUC.

We evaluated the performance of SUMMA on synthetic data, on the predictions submitted to a crowd-sourced competition (the BCL6 DREAM Challenge), and on multiple predictions to classification problems arising in real life contexts. The application of SUMMA in different settings shows that SUMMA can reliably estimate the AUCs of base classifiers in idealized problems where the assumption of conditional independence holds, as well as in real life applications where the assumption of conditional independence is likely violated. SUMMA performed better than the best base classifiers in the synthetic data as well as the BCL6 DREAM Challenge. In a third application, we used five datasets available from UCI machine learning repository which we split into a training set to train base classifiers and a test set to run ensemble methods including SUMMA. Our results show that SUMMA performs better than the best performer in three of the five datasets and among the top performers in the other cases.

In this paper we have introduced SUMMA for a binary classification problem. SUMMA can be extended to multi-class classification problems as follows. Assume that each classifier in an ensemble provides a score (such as confidence level or the distance to a surface) associated to the assignment of each sample to each of $k > 2$ classes. For each class i we run SUMMA as a binary classification problem, with class i being the positive class and all the other $k-1$ classes constituting the negative class. In this way, for each class i and each sample j we have a SUMMA score S_{ij}. A multi-class classification extension of SUMMA would consist of assigning sample j the class i that makes S_{ij} maximum. This approach requires that the classifiers assign class scores independently given the positive and negative class in each of the k binary problems.

SUMMA could be extended to cases where the conditional independence assumption between the classifiers does not hold and there is some structured correlation in the predictions. An example could be a block structure where a group of classifiers are correlated with each other but uncorrelated with predictors in other groups. A first step in that direction would be developing a method to identify the block structure and then average the predictions in each block before running SUMMA.

Another extension of our work would be to develop a modified version of SUMMA for the so-called rank-aggregation problem which is a very active area of research (Bhowmik and Ghosh, 2017). In the rank-aggregation problem, there is an unknown but true ordering of samples and each method is trying to predict this unknown ordering. The performance of a method is measured using various quantities such as *Kendall-tau* distance (Klementiev et al., 2008). In fact, one of the well established baseline methods used is Borda-counting which is strongly connected to the SUMMA score as we showed in the manuscript (Bhowmik and Ghosh, 2017). The first step towards this direction is determining the right performance metric to be used and setting up the right assumptions under which it can be estimated.

References

Shivani Agarwal, Thore Graepel, Ralf Herbrich, Sariel Har-Peled, and Dan Roth. Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

Javed A Aslam and Mark Montague. Models for metasearch. In *Proceedings of the ACM SIGIR Conference*, 2001.

Akshay Balsubramani and Yoav Freund. Scalable semi-supervised aggregation of classifiers. In *Advances in Neural Information Processing Systems*, 2015.

Rina Foygel Barber and Wooseok Ha. Gradient descent with non-convex constraints: local concavity determines convergence. *Information and Inference*, 7(4):755–806, 2018.

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. On the effect of calibration in classifier combination. *Applied Intelligence*, 38(4):566–585, 2013.

Avradeep Bhowmik and Joydeep Ghosh. Letor methods for unsupervised rank aggregation. In *Proceedings of the International Conference on World Wide Web*, 2017.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, 2009.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the International Conference on Machine learning*, 2004.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C*, 28 (1):20–28, 1979.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39 (1):1–22, 1977.

Thomas G Dietterich. Ensemble learning. *The Handbook of Brain Theory and Neural Networks*, 2:110–125, 2002.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

M Elter, R Schulz-Wendtland, and T Wittenberg. The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical Physics*, 34(11):4164–4172, 2007.

Peter Emerson. The original borda count and partial voting. *Social Choice and Welfare*, 40(2):353–358, 2013.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass auc optimization. In *International Conference on Machine Learning*, 2013.

Mike Gashler, Christophe Giraud-Carrier, and Tony Martinez. Decision tree ensemble: Small heterogeneous is better than large homogeneous. In *Proceedings of International Conference on Machine Learning and Applications*, 2008.

Frank E Harrell. Ordinal logistic regression. In *Regression Modeling Strategies*, pages 331–343. Springer, 2001.

Ariel Jaffe, Boaz Nadler, and Yuval Kluger. Estimating the accuracies of multiple classifiers without labeled data. In *Artificial Intelligence and Statistics*, 2015.

Prateek Jain and Sewoong Oh. Learning mixtures of discrete product distributions using spectral decompositions. In *Conference on Learning Theory*, 2014.

Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4): 620, 1957.

Azam Karami, Mehran Yazdi, and Grégoire Mercier. Compression of hyperspectral images using discerete wavelet transform and tucker decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):444–450, 2012.

Alexandre Klementiev, Dan Roth, and Kevin Small. Unsupervised rank aggregation with distance-based models. In *Proceedings of the International Conference on Machine Learning*, 2008.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Max Kuhn et al. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5):1–26, 2008a.

Max Kuhn et al. Caret package. *Journal of Statistical Software*, 28(5):1–26, 2008b.

M. Lichman. Uci machine learning repository, 2013.

Max A Little, Patrick E McSharry, Stephen J Roberts, Declan AE Costello, and Irene M Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering Online*, 6(1):23, 2007.

Tie-Yan Liu, Hang Li, and Yu-Ting Liu. Supervised rank aggregation based on rankings, 2010. US Patent 7,840,522.

Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Andrej Aderhold, Richard Bonneau, Yukun Chen, et al. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796, 2012.

Caren Marzban. The roc curve and the area under it as performance measures. *Weather and Forecasting*, 19(6):1106–1114, 2004.

Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Kenta Nakai and Minoru Kanehisa. Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins: Structure, Function, and Bioinformatics*, 11(2): 95–110, 1991.

Alexandru Niculescu-Mizil, Claudia Perlich, Grzegorz Swirszcz, Vikas Sindhwani, Yan Liu, Prem Melville, Dong Wang, Jing Xiao, Jianying Hu, Moninder Singh, et al. Winning the kdd cup orange challenge with ensemble selection. In *Proceedings of the International Conference on KDD-Cup*, 2009.

Shmuel Nitzan and Jacob Paroush. Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, pages 289–297, 1982.

Shmuel Nitzan and Ariel Rubinstein. A further characterization of borda ranking method. *Public Choice*, 36(1):153–158, 1981.

T Maruthi Padmaja, Narendra Dhulipalla, Raju S Bapi, and P Radha Krishna. Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection. In *Proceedings of the International Conference on Advanced Computing and Communications*, 2007.

Fabio Parisi, Francesco Strino, Boaz Nadler, and Yuval Kluger. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, 111(4):1253–1258, 2014.

Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: The dream3 challenges. *PloS One*, 5(2):e9202, 2010.

Julio Saez-Rodriguez, James C Costello, Stephen H Friend, Michael R Kellen, Lara Mangravite, Pablo Meyer, Thea Norman, and Gustavo Stolovitzky. Crowdsourcing biomedical research: leveraging communities as innovation engines. *Nature Reviews Genetics*, 17(8): 470, 2016.

Robert E Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

Amartya Sen. Social choice theory. *Handbook of Mathematical Economics*, 3:1073–1181, 1986.

Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

Gustavo Stolovitzky, Robert J Prill, and Andrea Califano. Lessons from the dream2 challenges: a community effort to assess biological network inference. *Annals of the New York Academy of Sciences*, 1158(1):159–195, 2009.

Karthik Subbian and Prem Melville. Supervised rank aggregation for predicting influencers in twitter. In *Proceedings of the International Conference on Privacy, Security, Risk and Trust*, 2011.

Liang Sun, Tomonori Honda, Vesselin Diev, Gregory Gancarz, Jeong-Yoon Lee, Ying Liu, Mona Mahmoudi, Raghav Mathur, Shahinur Rahman, Steve Wickert, et al. Maximize auc in default prediction: Modeling and blending.

James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

Chih-Fong Tsai and Yu-Chieh Hsiao. Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, 50(1):258–269, 2010.

Lieven Vandenberghe and Stephen Boyd. Applications of semidefinite programming. *Applied Numerical Mathematics*, 29(3):283–299, 1999.

Sean Whalen and Gaurav Pandey. A comparative analysis of ensemble classifiers: case studies in genomics. In *Proceedings of the International Conference on Data Mining*, 2013.

David H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

Appendix A. Details of SDP Approach

A.1. Semi-Definite Programming

We start this section by the following definition.

Definition 9 *We say that a matrix Q is off-diagonal rank-one if there exists a vector q and a diagonal matrix D s.t.*

$$Q = qq^T + D. \tag{28}$$

In order to show that R is the unique solution to the problem (11), we prove a lemma stating that a unique solution of the problem (11) exists for an arbitrary matrix Q whose off-diagonal elements coincide with a rank one matrix. The application of this lemma with $Q = \Sigma_2$ will then allow us to show the desired result.

Lemma 10 *Let Q be an off-diagonal rank-one square matrix of size M such that*

$$Q = qq^T + D_0, \tag{29}$$

for some $q \in \mathbb{R}^M$ with $M \geq 3$, and diagonal matrix D_0 and assume that $q_1, q_2, q_3 \neq 0$. Then the optimization problem

$$\min_D rank(Q - D) \quad s.t. \quad D(i,j) = 0 \quad for \quad i \neq j, \quad and \quad Q - D \succeq 0, \tag{30}$$

has the unique solution D_0, with $Q - D_0 = qq^T = (-q)(-q)^T$ so that we can recover q up to its sign.

Proof Let Q be defined as in Equation (29) with $q_1, q_2, q_3 \neq 0$ and D_0. Since $Q(i,j) = q_i q_j \neq 0$, it is obvious that for any diagonal matrix D, $Q - D \neq 0$. Hence, $rank(Q - D) > 0$ for any diagonal matrix D. Moreover, since $Q - D_0 = qq^T$ one solution of the optimization problem in Equation (11) is D_0. Next, we show that D_0 is the only diagonal matrix that is a feasible point of the optimization problem.

Suppose there exists a diagonal matrix D_s such that $D_s \neq D_0$ and $rank(Q - D_s) = 1$. Then since $Q - D_s$ is symmetric, there exists $s \in \mathbb{R}^M$ such that $Q - D_s = ss^T$ and $q \neq s$. Since both D_0 and D_s are diagonal and $D_s + ss^T = Q = D_0 + qq^T$, then the equality

$$q_i q_j = s_i s_j. \tag{31}$$

must be true for all $i \neq j$ where $i, j \in \{1, 2, \ldots, M\}$.

Without loss of generality lets assume that $q_1 \neq 0 > s_1 \neq 0$. Equation (31) for $(i,j) = (1,2)$ implies that $s_2 > q_2$, and for $(i,j) = (1,3)$ implies that $s_3 > q_3$. Under these inequalities, when $(i,j) = (2,3)$ we see that $s_2 s_3 > q_2 q_3$, which contradicts equation (31). Therefore, $q_1 = s_1$ which from Equation (31) implies that for all $i, q_i = s_i$. Therefore, $D_0 = D_s$ and consequently D_0 is the unique solution to the optimization problem given in equation (11). ■

As we discussed in the main text the optimization problem in (30) can be relaxed to the following SDP:

$$\min_D \quad \|Q - D\|_* \quad \text{s.t.} \quad D \text{ is diagonal, and} \quad Q - D \succeq 0. \tag{32}$$

Theorem 11 *Suppose that Q is an $M \times M$ matrix of the form given in Equation (29) for some q and diagonal matrix D_0, then the optimization problem in Equation (32) has the unique solution D_0, provided that*

$$q_i^2 < \sum_{j \neq i} q_j^2, \quad \forall i \in \{1, 2, \ldots, M\}. \tag{33}$$

Moreover, $Q - D_0 = qq^T$ so that we can recover q up to its sign.

Proof Let D be an arbitrary diagonal matrix such that $Q - D$ is a PSD (Positive Semidefinite) matrix. Then the eigenvalues of $Q - D$ are non-negative and equal to the singular values of $Q - D$. Combined with the fact that the trace of a matrix is equal to the sum of its eigenvalues, we know the following:

$$\|Q - D\|_* = Tr(Q - D).$$

Without loss of generality lets assume that the diagonal entries of Q are equal to 0, that is $Q = qq^T - diag(qq^T)$. In this case, note that $D_0 = -diag(qq^T)$ is a feasible solution for the optimization problem in Equation (13). Next let D be an arbitrary solution and since $Q - D$ is PSD, $D_{ii} \leq 0$ for all i. Suppose for some i we have $D_{ii} > -q_i^2$ and for all $j \neq i$ we have $D_{jj} \geq -q_j^2$. For $j \neq i$, consider the following sub-matrix of $Q - D$,

$$(Q - D)^{ij} = \begin{bmatrix} -D_{ii} & q_i q_j \\ q_i q_j & -D_{jj} \end{bmatrix}.$$

Since $det((Q - D)^{ij}) = D_{ii} D_{jj} - q_i^2 q_j^2 < 0$, the submatrix $(Q - D)^{ij}$ of $Q - D$ is negative definite which contradicts the fact that $Q - D$ is PSD. Therefore, combined with the fact that $D_{ii} \leq 0$ for each i, this implies that in order for a diagonal matrix D to be a feasible point for the optimization problem (13)

1. Either for all i, we have $D_{ii} \leq -q_i^2$
2. Or there exist i such that $D_{ii} > -q_i^2$, and $\forall j \neq i$ $D_{jj} \leq -q_j^2$.

Suppose D is a feasible point that satisfies condition 1, then $Tr(Q - D) \geq Tr(Q + diag(qq^t))$ for every feasible D. Hence, $D_0 = -diag(qq^t)$ remains to be the unique solution of the optimization problem. Now suppose there exists a feasible point D such that condition 2 of above is satisfied. Then WLO assume that $D_{11} > -q_1^2$ and $D_{jj} \leq -q_i^2$ for $j \geq 2$. Next, for each $j \geq 2$ consider the following submatrix:

$$(Q - D)^{1j} = \begin{bmatrix} -D_{11} & q_1 q_j \\ q_1 q_j & -D_{jj} \end{bmatrix}.$$

In order for $Q - D$ to be PSD, we should have $det((Q - D)^{1j}) = D_{11} D_{jj} - q_1^2 q_j^2 > 0$, which in turn implies that

$$D_{jj} < \frac{q_1^2 q_j^2}{D_{11}}. \tag{34}$$

Using Equation (34), we observe that

$$\begin{aligned} \|Q - D\|_* &= Tr(Q - D) = -D_{11} - \sum_{j>1} D_{jj} \\ &> -D_{11} + \sum_{j>1} -\frac{q_1^2 q_j^2}{D_{11}}, \end{aligned} \tag{35}$$

where Equation (35) comes from Equation (34). Now from the assumption that $q_1^2 < \sum_{j>1} q_j^2$, we have $D_{11} > -q_1^2 > -q_1 \sqrt{\sum_{j>1} q_j^2}$, and for any $D_{11} \in [-q_1^2, 0]$

$$\|Q - D\|_* < -D_{11} - \sum_{j>1} \frac{q_1^2 q_j^2}{D_{11}} > -q_1^2 - \sum_{j>1} q_j^2 = \left\|Q + diag(qq^t)\right\|_*. \tag{36}$$

A generalization of the above argument implies that if for each i, $q_i^2 < \sqrt{\sum_{j \neq i} q_j^2}$, then $\|Q - D\|_* > \left\|Q + diag(qq^t)\right\|_*$ so that $D_0 = -diag(qq^t)$ the unique solution to the optimization problem (13). ■

Our next result shows that the solutions of (30) and (32) coincide.

Appendix B. Proof of Theorem 2

Proof For simplicity, we denote the set of M classifiers $\{g_1, \cdots, g_M\}$ as $\{1, ..., M\}$ and proceed by induction on n. Note that due to our problem setup for each i, we have $\mathbb{E}[R_i] = N/2$. Suppose the classifiers $\{g_i\}_{i=1}^M$ are l^{th} order conditionally independent. The conditional independence assumption implies that any given subset of classifiers of size $n \leq l$ of $\{1, ..., M\}$ are n^{th} order conditionally independent. Next, let $n = 2$, and suppose that the two methods 1 and 2 are conditionally independent as such equation (1) is satisfied with $n = 2$. Then by using law of total expectation, we obtain the following:

$$\begin{aligned} \Sigma_2(1,2) &= \mathbb{E}[(R_1 - \mathbb{E}[R_1])(R_2 - \mathbb{E}[R_2])] = \mathbb{E}[R_1 R_2] - \mathbb{E}[R_1]\,\mathbb{E}[R_2] \\ &= \mathbb{E}[R_1 R_2 | y = 1]\rho + \mathbb{E}[R_1 R_2 | y = 0](1 - \rho) - \mathbb{E}[R_1]\,\mathbb{E}[R_2] \\ &= \mathbb{E}[R_1 | y = 1]\,\mathbb{E}[R_2 | y = 1](\rho) + \mathbb{E}[R_1 | y = 0]\,\mathbb{E}[R_2 | y = 0](1 - \rho) \\ &- \mathbb{E}[R_1]\,\mathbb{E}[R_2] \end{aligned} \tag{37}$$

where as defined before ρ is the prevalence of class y=1. Similarly, from the law of total expectation we have

$$\begin{aligned} \mathbb{E}[R_1] &= \mathbb{E}[R_1 | y = 1]\rho + \mathbb{E}[R_1 | y = 0](1 - \rho). \\ \mathbb{E}[R_2] &= \mathbb{E}[R_2 | y = 1]\rho + \mathbb{E}[R_2 | y = 0](1 - \rho). \end{aligned} \tag{38}$$

Substituting Equation (38) into Equation (37), we obtain

$$\Sigma_2(1,2) = C(\rho)\Delta_1\Delta_2.$$

where $C(\rho) = \rho(1 - \rho)$. To ease the notation for the inductive step, for any $n \geq 1$, let the random variable S_n be defined as follows

$$S_n = (R_1 - \mathbb{E}[R_1]) \cdots (R_n - \mathbb{E}[R_n]) = (R_n - \mathbb{E}[R_n])S_{n-1},$$

where $S_0 = 1$. Also note that $\Sigma_n(1, \cdots, n) = \mathbb{E}[S_n]$ Next, we inductively show two formulas which are required for the proof of the theorem.
Claim 1: For any $1 \le n \le l \le M$, $\mathbb{E}[S_n|y=0] = \rho^n \prod_{i=1}^n \Delta_i$.
Proof: For $n = 1$,

$$\begin{aligned}\mathbb{E}[S_1|y=0] &= \mathbb{E}[R_1 - \mathbb{E}[R_1]|y=0] = \mathbb{E}[R_1|y=0] - \mathbb{E}[R_1] \\ &= \mathbb{E}[R_1|y=0] - \rho\,\mathbb{E}[R_1|y=1] - (1-\rho)\,\mathbb{E}[R_1|y=0] \\ &= \rho(\mathbb{E}[R_1|y=0] - \mathbb{E}[R_1|y=1]) = \rho\Delta_1. \end{aligned} \tag{39}$$

Next assume the claim is true for $n-1$, then from the inductive hypothesis we have

$$\mathbb{E}[S_{n-1}|y=0] = \rho^{n-1}\prod_{j=1}^{n-1}\Delta_j. \tag{40}$$

Let us now prove that if it is true for $n-1$, then it is true for n. From conditional independence and total law of expectation we get

$$\begin{aligned}\mathbb{E}[S_n|y=0] &= \mathbb{E}[(R_n - \mathbb{E}[R_n])S_{n-1}|y=0] = (\mathbb{E}[R_n|y=0] - \mathbb{E}[R_n])\,\mathbb{E}[S_{n-1}|y=0] \\ &= \mathbb{E}[R_n|y=0]\,\mathbb{E}[S_{n-1}|y=0] - \mathbb{E}[R_n]\,\mathbb{E}[S_{n-1}|y=0] \\ &= \mathbb{E}[R_n|y=0]\,\mathbb{E}[S_{n-1}|y=0] - (1-\rho)\,\mathbb{E}[R_n|y=0]\,\mathbb{E}[S_{n-1}|y=0] - \rho\,\mathbb{E}[R_n|y=1]\,\mathbb{E}[S_{n-1}|y=0] \\ &= \rho\,\mathbb{E}[S_{n-1}|y=0](\mathbb{E}[R_n|y=0] - \mathbb{E}[R_n|y=1]) \\ &= \rho\,\mathbb{E}[S_{n-1}|y=0]\Delta_n = \rho\Delta_n\rho^{n-1}\prod_{j=1}^{n-1}\Delta_j = \rho^n\prod_{j=1}^{n}\Delta_j,\end{aligned}$$

where the previous to last equality follows from inductive assumption and completes the proof.

Claim 2: For any $1 \le n \le l \le M$, $\mathbb{E}[S_n|y=1] = (\rho-1)^n \prod_{i=1}^n \Delta_i$.
The proof of this claims follows from Claim 1 by replacing $\rho \to 1-\rho$ and $\Delta_i \to -\Delta_i$.
Now we are ready to prove the main theorem. Using law of total expectation and claim 1 and 2, we obtain the following set of equations

$$\begin{aligned}\mathbb{E}[S_n] &= \mathbb{E}[S_n|y=1]\rho + \mathbb{E}[S_n|y=0](1-\rho) = (\rho(\rho-1)^l + (1-\rho)\rho^n)\prod_{j=1}^{n}\Delta_j \\ &= \rho(1-\rho)(\rho^{n-1} - (\rho-1)^{n-1})\prod_{j=1}^{n}\Delta_j,\end{aligned} \tag{41}$$

which completes the proof of the theorem. ■

Appendix C. Proof of Theorem 4

Proof Let N_1 denote the positive samples and $N_0 = N - N_1$ denote the negative samples. Given a ranking of samples, the receiver operating characteristic (ROC) curve consists of points representing the False Positive Rates (FPR_i) and the True Positive Rates (TPR_i) empirically evaluated for each threshold $i \in \{1, ..., N\}$. The area under the ROC (AUC) can be calculated according to the rectangle rule using the following formula:

$$AUC_i = \sum_{i=0}^{N-1} \mathrm{TPR}_i\,(\mathrm{FPR}_{i+1} - \mathrm{FPR}_i)\,, \tag{42}$$

where $FPR_0 := 0$ and $TPR_0 := 0$. The elements of the sum behave as follows. If the ith ranked sample has a positive label then the $\mathrm{FPR}_{i+1} = \mathrm{FPR}_i$, and if it has a negative label then $FPR_{i+1} = FPR_i + 1/N_0$. Moreover, if we let $\{r_{i_1}, \cdots, r_{i_{N_0}}\}$ denote the ranks of negative samples, then for any threshold i_l the number of true positives is given as $i_l - l$ and consequently $TPR_l = (i_l - l)/N_1$. Using these observations, Equation (42) becomes

$$\begin{aligned}AUC_i &= \sum_{i=1}^{N}\mathrm{TPR}_i(\mathrm{FPR}_{i+1} - \mathrm{FPR}_i) \\ &= \sum_{l:y_l=0}\frac{\mathrm{TPR}_l}{N_1}\frac{1}{N_0} = \sum_{l=1}^{N_0}\frac{(r_{i_l} - l)}{N_1}\frac{1}{N_0} = \frac{\mathbb{E}[\widehat{R}_i|y=0]}{N_1} - \frac{N_0+1}{2N_1}.\end{aligned} \tag{43}$$

Next we express $\widehat{\Delta}_i$ in terms of $\mathbb{E}[\widehat{R}_i|y=0]$:

$$\begin{aligned}\frac{\widehat{\Delta}_i}{N} &= \frac{\frac{\sum_{i:y_k=0} r_{ik}}{N_0} - \frac{\sum_{k:y_k=1} r_{ik}}{N_1}}{N} \\ &= \frac{\frac{\sum_{k:y_k=0} r_{ik}}{N_0} + \frac{\sum_{k:y_k=0} r_{ik}}{N_1} - \frac{\sum_{i=1}^{N} r_{ik}}{N_1}}{N} = \frac{\mathbb{E}[\widehat{R}_i|y=0]_i}{N_1} - \frac{N+1}{2N_1} \\ \implies \frac{\mathbb{E}[\widehat{R}_i|y=0]_i}{N_1} &= \frac{\widehat{\Delta}_i}{N} + \frac{N+1}{2N_1}.\end{aligned} \tag{44}$$

If we substitute Equation (44) into Equation (43), we obtain

$$\begin{aligned}AUC_i &= \frac{\mathbb{E}[\widehat{R}_i|y=0]}{N_1} - \frac{N_1+1}{2N_1} = \frac{\widehat{\Delta}_i}{N} + \frac{N+1}{2N_1} - \frac{N_0+1}{2N_1} \\ &= \frac{\widehat{\Delta}_i}{N} + \frac{1}{2},\end{aligned} \tag{45}$$

■

which completes the proof of the theorem.

Appendix D. Proof of Lemma 6

Proof Consider the kth sample assigned rank r_{ik} by the ith base classifier. Let $P_i(y_k|r_{ik})$ be the probability distribution that is *a priori* unknown and that is an extremum of the maximum entropy functional

$$J = -\sum_{k=1}^{N}\sum_{y_k\in\{0,1\}} P(r_{ik})P_i(y_k|r_{ik})\log(P_i(y_k|r_{ik})) + \sum_{j=0}^{2}\lambda_{ij}\Lambda_j,$$

subject to the constraints Λ_j for $j = (0, 1, 2)$, with

$$\Lambda_0 = N - \sum_{k=1}^{N} \sum_{y_k \in \{0,1\}} P_i(y_k|r_{ik}), \tag{46a}$$

$$\Lambda_1 = N\rho - \sum_{k=1}^{N} \sum_{y_k \in \{0,1\}} y_k P_i(y_k|r_{ik}), \tag{46b}$$

$$\Lambda_2 = N\rho\,\mathbb{E}[R_i|y_k = 1] - \sum_{k=1}^{N} r_{ik} \sum_{y_k \in \{0,1\}} y_k P_i(y_k|r_{ik}), \qquad \text{or equivalently}$$

$$= N\rho\big(\mathbb{E}[R_i] - (1-\rho)\Delta_i\big) - \sum_{k=1}^{N} r_{ik} \sum_{y_k \in \{0,1\}} y_k P_i(y_k|r_{ik}). \tag{46c}$$

Here we constrain the inferred function such that the sum over labels normalizes to 1 (Equation 46a), the average occurrence of class one samples is reflective of its occurrence in the population (N_1) (Equation 46b), and lastly that the average rank condition by the class label recovers our empirical estimate of Δ_i (Equation 46c).

We consider the maximum entropy criterion satisfied when the variation of the functional δJ with respect to $P_i(y_k|r_{ik})$ is stationary,

$$\delta J = \sum_{k=1}^{N} \sum_{y_k \in \{0,1\}} \left[-\frac{1}{N} \log\left(P_i(y_k|r_{ik})\right) - \frac{1}{N} - \lambda_{i0} - \lambda_{i1} y_k - \lambda_{i2} y_k r_{ik} \right] \delta P_i(y_k|r_{ik}) = 0$$

and consequently,

$$P_i(y_k|r_{ik}) = \frac{1}{Z} e^{-\lambda_{i1} y_k - \lambda_{i2} y_k r_{ik}}. \tag{47}$$

In Equation (47), λ_{i0} is incorporated in the constant Z, which satisfies the normalization constraint in Equation (46a). Therefore, $Z = 1 + e^{-\lambda_{i1} - \lambda_{i2} r_{ik}}$.

Next, we solve for the remaining Lagrange multipliers by substituting Equation (47) into Equations (46b, 46c). Each of these equations, after summing over class labels, amount to calculating sums over $P_i(y_k = 1|r_{ik}) = (1 + e^{\lambda_{i1} + \lambda_{i2} r_{ik}})^{-1}$. Here, the functional form of $P_i(y_k = 1|r_{ik})$ precludes clear analytical solutions to these summations and consequently precludes solutions for λ_{i1} and λ_{i2}. To circumvent this challenge we approximate the distribution $P_i(y_k = 1|r_{ik})$ by its Taylor series in λ_{i1} and λ_{i2} about their respective values, λ_1^* and λ_2^*, representative of an uninformative classifier.

Consider an uninformative classifier as one that assigns each sample a rank without regard to its true class label. That is, it assigns the N_1 positive samples and the N_0 negative samples to be classified, random ranks between 1 and N. As the probability of such classifier to locate a positive class samples at rank r is independent of rank, then $\lambda_2^* = 0$ and given that there are $N_1 = \rho N$ positive class items in the sample, $\lambda_1^* = \log((1-\rho)/\rho)$, where the asterisk explicitly indicates that the parameters are those of the uninformative classifier. For clarity in exposition, we will explicitly include the Lagrange multipliers in the notation of the probability of the class of an item at a given rank as $P_i(y_k|r_{ik}, \lambda_{i1}, \lambda_{i2})$;

then by substituting these parameters into the maximum entropy distribution we see that the uninformative classifier,

$$P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*) = \left(1 + e^{\lambda_1^* + \lambda_2^* r_{ik}}\right)^{-1} = \rho.$$

The Taylor series of $P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})$ about the uninformative classifier amounts to expanding $P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})$ in λ_{i1} and λ_{i2} about λ_1^* and λ_2^*. We assume that all considered classifiers are weakly predictive, so that the terms of order two and higher negligibly contribute the approximated function. Then the Taylor series to first order is,

$$\begin{aligned} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2}) \approx\; & P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*) \\ & + \partial_{\lambda_{i1}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} \delta\lambda_{i1} \\ & + \partial_{\lambda_{i2}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} \lambda_{i2} \end{aligned} \tag{48}$$

with $\partial_{\lambda_{i1}}$ and $\partial_{\lambda_{i2}}$ being the partial derivatives $\frac{\partial}{\partial\lambda_{i1}}$ and $\frac{\partial}{\partial\lambda_{i2}}$, and $\delta\lambda_{i1} = \lambda_{i1} - \log((1-\rho)/\rho)$. The partial derivatives are as follows,

$$\partial_{\lambda_{i1}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} = -P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\left(1 - P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\right),$$

$$\partial_{\lambda_{i2}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} = -P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\left(1 - P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*)\right) r_{ik},$$

and recalling that $P_*(y_k = 1|r_{ik}, \lambda_1^*, \lambda_2^*) = \rho$, then

$$\partial_{\lambda_{i1}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} = -\rho(1-\rho), \tag{49}$$

$$\partial_{\lambda_{i2}} P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2})\Big|_{\lambda_1^*, \lambda_2^*} = -\rho(1-\rho) r_{ik}. \tag{50}$$

By substituting Equation (49) and (50) into the truncated Taylor series of Equation (48),

$$P_i(y_k = 1|r_{ik}, \lambda_{i1}, \lambda_{i2}) \approx \rho - \rho(1-\rho)\delta\lambda_{i1} - \rho(1-\rho) r_{ik} \lambda_{i2}. \tag{51}$$

The Lagrange multipliers (λ_{i1}, λ_{i2}) are then solved for by substituting Equation (51) into the equations of constraint written in Equations (46b) and (46c).

Application of approximate distribution to Λ_1

Application of Equation (51) to the equation of constraint of Equation (46b),

$$\begin{aligned} \Lambda_1 &= N\rho - \sum_{k=1}^{N} \rho - \rho(1-\rho)\delta\lambda_{i1} - \rho(1-\rho) r_{ik}\lambda_{i2} \\ &= N\rho - N\rho + N\rho(1-\rho)\delta\lambda_{i1} + N\rho(1-\rho)\,\mathbb{E}[R_i]\lambda_{i2}, \end{aligned}$$

where $\mathbb{E}[R_i] = \frac{1}{N}\sum_{k=1}^{N} r_{ik} = (N+1)/2$. By definition $\Lambda_1 = 0$, resulting in

$$\delta\lambda_{i1} = -\,\mathbb{E}[R_i]\lambda_{i2}. \tag{52}$$

Application of approximate distribution to Λ_2

Similarly, for the constraint in Equation (46c),

$$\begin{aligned}\Lambda_2 &= N\rho\big(\,\mathbb{E}[R_i] - (1-\rho)\Delta_i\big) - \sum_{k=1}^{N}\rho r_{ik} - \rho(1-\rho)r_{ik}\delta\lambda_{i1} - \rho(1-\rho)r_{ik}^2\lambda_{i2}\\ &= N\rho\big(\,\mathbb{E}[R_i] - (1-\rho)\Delta_i\big) - N\rho\,\mathbb{E}[R_i] + N\rho(1-\rho)\,\mathbb{E}[R_i]\delta\lambda_{i1} + N\rho(1-\rho)\,\mathbb{E}[R_i^2]\lambda_{i2},\\ &= -N\rho(1-\rho)\left[\Delta_i - \mathbb{E}[R_i]\delta\lambda_{i1} - \mathbb{E}[R_i^2]\lambda_{i2}\right],\end{aligned}$$

where $\mathbb{E}[R_i^2] = \frac{1}{N}\sum_{k=1}^{N} r_{ik}^2$. By definition $\Lambda_2 = 0$, resulting in

$$\Delta_i = \mathbb{E}[R_i]\delta\lambda_{i1} + \mathbb{E}[R_i^2]\lambda_{i2}. \tag{53}$$

Determining λ_{i1}, λ_{i2}, and the approximate maximum entropy distribution

We first determine λ_{i2} by substituting $\delta\lambda_{i1}$ from Equation (52) into Equation (53),

$$\begin{aligned}\Delta_i &= \left(-\,\mathbb{E}[R_i]^2 + \mathbb{E}[R_i^2]\right)\lambda_{i2}\\ \implies \lambda_{i2} &= \frac{\Delta_i}{\sigma^2}\end{aligned} \tag{54}$$

where $\sigma^2 = \mathbb{E}[R_i^2] - \mathbb{E}[R_i]^2$ is simply the variance of a uniform discrete random variable between 1 and N. Then, by substituting Equation (54) into Equation (52),

$$\lambda_{i1} = \log\left(\frac{1-\rho}{\rho}\right) - \frac{\Delta_i}{\sigma^2}\,\mathbb{E}[R_i] \tag{55}$$

With these expressions for λ_{i1} and λ_{i2}, setting $\sigma^2 = (N^2-1)/12$, and setting $E[R_i] = (N+1)/2$ the first order approximation of the maximum entropy distribution around an uninformative classifier is,

$$P_i(y_k = 1|r_{ik}) = \left(1 + e^{\log\left(\frac{1-\rho}{\rho}\right) + \frac{12\Delta_i}{N^2-1}\left(r_{ik} - \frac{N+1}{2}\right)}\right)^{-1}.$$

The proof is completed by substituting N_1/N for ρ, in which $(1-\rho)/\rho = (N-N_1)/N_1$. ■

Appendix E. Description of UCI Datasets

We used the following five UCI Datasets in this manuscript: Bank Marketing, Ionosphere, Mammographic Mass, Parkinsons and Yeast datasets. We next briefly describe these datasets.

Bank Marketing (Moro et al., 2014):
This data aims to predict whether a client would subscribe to a product (bank term deposit) or not ('no') based on a marketing survey conducted using phone calls. The data has 20 features including age,job,marital status, and education.

Ionosphere (Sigillito et al., 1989):
This is a radar dataset that was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere and "Bad" returns are those that do not; their signals pass through the ionosphere.

Mammographic Masses (Elter et al., 2007):
This data set is used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains a BI-RADS assessment, the patient's age and three BI-RADS attributes together with the ground truth (the severity field) for 516 benign and 445 malignant masses that have been identified on full field digital mammograms collected at the Institute of Radiology of the University Erlangen-Nuremberg between 2003 and 2006.

Parkinsons (Little et al., 2007):
This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each feature corresponds to a particular voice measure, and each sample corresponds to one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

Yeast (Nakai and Kanehisa, 1991):
The objective of this data is to use 9 descriptors to predict the localizations (called cellular components) of proteins in a yeast' s cell.

Journal of Machine Learning Research 20 (2019) 1-46 Submitted 2/18; Published 10/19

Stochastic Canonical Correlation Analysis

Chao Gao CHAOGAO@GALTON.UCHICAGO.EDU
University of Chicago
Chicago, IL 60637, USA

Dan Garber DANGAR@TECHNION.AC.IL
Technion – Israel Institute of Technology
Haifa, 3200003, Israel

Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago
Chicago, IL 60637, USA

Jialei Wang JIALEI@UCHICAGO.EDU
University of Chicago
Chicago, IL 60637, USA

Weiran Wang WEIRANWANG@TTIC.EDU
Toyota Technological Institute at Chicago
Chicago, IL 60637, USA

Editor: John Shawe-Taylor

Abstract

We study the sample complexity of canonical correlation analysis (CCA), i.e., the number of samples needed to estimate the population canonical correlation and directions up to arbitrarily small error. With mild assumptions on the data distribution, we show that in order to achieve ϵ-suboptimality in a properly defined measure of alignment between the estimated canonical directions and the population solution, we can solve the empirical objective exactly with $N(\epsilon, \Delta, \gamma)$ samples, where Δ is the singular value gap of the whitened cross-covariance matrix and $1/\gamma$ is an upper bound of the condition number of auto-covariance matrices. Moreover, we can achieve the same learning accuracy by drawing the same level of samples and solving the empirical objective approximately with a stochastic optimization algorithm; this algorithm is based on the shift-and-invert power iterations and only needs to process the dataset for $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ passes. Finally, we show that, given an estimate of the canonical correlation, the streaming version of the shift-and-invert power iterations achieves the same learning accuracy with the same level of sample complexity, by processing the data only once.

Keywords: Canonical correlation analysis, sample complexity, shift-and-invert preconditioning, streaming CCA

1. Introduction

Let $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{y} \in \mathbb{R}^{d_y}$ be two random vectors with a joint probability distribution $P(\mathbf{x}, \mathbf{y})$. The objective of CCA (Hotelling, 1936) in the population setting is to find $\mathbf{u} \in \mathbb{R}^{d_x}$ and $\mathbf{v} \in \mathbb{R}^{d_y}$ such that projections of the random variables onto these directions are

©2019 Chao Gao, Dan Garber, Nathan Srebro, Jialei Wang, and Weiran Wang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-095.html.

maximally correlated:[1]

$$\max_{\mathbf{u},\mathbf{v}} \frac{\mathbb{E}[(\mathbf{u}^\top \mathbf{x})(\mathbf{v}^\top \mathbf{y})]}{\sqrt{\mathbb{E}[(\mathbf{u}^\top \mathbf{x})^2]}\sqrt{\mathbb{E}[(\mathbf{v}^\top \mathbf{y})^2]}}. \tag{1}$$

This objective can be written in the equivalent constrained form

$$\max_{\mathbf{u},\mathbf{v}} \mathbf{u}^\top \mathbf{E}_{xy} \mathbf{v} \quad \text{s.t.} \quad \mathbf{u}^\top \mathbf{E}_{xx} \mathbf{u} = \mathbf{v}^\top \mathbf{E}_{yy} \mathbf{v} = 1 \tag{2}$$

where the cross- and auto-covariance matrices are defined as

$$\mathbf{E}_{xy} = \mathbb{E}[\mathbf{x}\mathbf{y}^\top], \qquad \mathbf{E}_{xx} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top], \qquad \mathbf{E}_{yy} = \mathbb{E}[\mathbf{y}\mathbf{y}^\top]. \tag{3}$$

The global optimum of (2), denoted by $(\mathbf{u}^*, \mathbf{v}^*)$, can be computed in closed-form. Define

$$\mathbf{T} := \mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{E}_{xy} \mathbf{E}_{yy}^{-\frac{1}{2}} \in \mathbb{R}^{d_x \times d_y}, \tag{4}$$

and let $(\mathbf{a}_1, \mathbf{b}_1)$ be the (unit-length) top left and right singular vector pair associated with $\mathbf{T}$'s largest singular value $\rho_1 = \sigma_1(\mathbf{T})$. Then the optimal objective value, i.e., the canonical correlation between $\mathbf{x}$ and $\mathbf{y}$, is $\rho_1 \leq 1$ (see Lemma 20), achieved by $(\mathbf{u}^*, \mathbf{v}^*) = (\mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{a}_1, \mathbf{E}_{yy}^{-\frac{1}{2}} \mathbf{b}_1)$.

In practice, we do not have access to the population covariance matrices, but observe samples pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ drawn from $P(\mathbf{x}, \mathbf{y})$. In this paper, we are concerned with both the number of samples $N(\epsilon)$ needed to approximately solve (2), and the time complexity for obtaining the approximate solution. Note that the CCA objective is not a stochastic convex program due to the ratio form (1), and standard stochastic approximation methods do not apply (Arora et al., 2012). Globally convergent stochastic optimization of CCA has long been a challenge even for the empirical objective, and attracted continuous effort (Lu and Foster, 2014; Ma et al., 2015; Wang and Livescu, 2016), until the recent breakthrough by Ge et al. (2016); Wang et al. (2016). And our understanding of the stochastic objective, e.g., the existence of an efficient algorithm and the sample complexity, has been very limited.

Our contributions The contributions of our paper are summarized as follows.

- First, we provide the ERM sample complexity of CCA. We show that in order to achieve ϵ-suboptimality in the alignment between the estimated canonical directions and the population solution (relative to the population covariances, see Section 2), we can solve the empirical objective exactly with $N(\epsilon, \Delta, \gamma)$ samples where Δ is the singular value gap of the whitened cross-covariance and $1/\gamma$ is a upper bound of the condition number of the auto-covariance, for several general classes of distributions widely used in statistics and machine learning.

1. For simplicity (especially for the streaming setting), we assume that $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ and $\mathbb{E}[\mathbf{y}] = \mathbf{0}$. Nonzero means can be easily handled in the ERM approach (see Remark 4).

- Second, to alleviate the high computational complexity of exactly solving the empirical objective, we show that we can achieve the same learning accuracy by drawing the same level of samples and solving the empirical objective approximately with the stochastic optimization algorithm of Wang et al. (2016). This algorithm is based on the shift-and-invert power iterations (Saad, 1992; Garber and Hazan, 2015; Garber et al., 2016). We provide tightened analysis of the algorithm's time complexity, removing an extra $\log \frac{1}{\epsilon}$ factor from the complexity given by Wang et al. (2016). Our analysis shows that asymptotically it suffices to process the sample set for $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ passes. While near-linear runtime in the required number of samples is known and achieved for convex learning problems using SGD, no such result was estabilished for the nonconvex CCA objective previously.
- Third, we show that the streaming version of shift-and-invert power iterations achieves the same learning accuracy with the same level of sample complexity, given a good estimate of the canonical correlation. This approach requires only $\mathcal{O}(d)$ memory where $d := d_x + d_y$ is the input dimensionality, and thus further alleviates the memory cost of solving the empirical objective. This addresses the challenge of the existence of a stochastic algorithm for CCA proposed by Arora et al. (2012).

Notation We use $\sigma_i(\mathbf{A})$ to denote the i-th largest singular value of a matrix $\mathbf{A}$, and use $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ to denote the largest and smallest singular values of $\mathbf{A}$ respectively. We use $\|\cdot\|$ to denote the spectral norm of a matrix or the ℓ_2-norm of a vector. For a positive definite matrix $\mathbf{M}$, the vector norm $\|\cdot\|_{\mathbf{M}}$ is defined as $\|\mathbf{w}\|_{\mathbf{M}} = \sqrt{\mathbf{w}^\top \mathbf{M} \mathbf{w}}$ for any $\mathbf{w}$. We use C and C' to denote universal constants that are independent of problem parameters, and their specific values may vary among appearances. We hide poly-logarithmic dependencies in the notation $\tilde{\mathcal{O}}(\cdot)$.

2. Problem setup

Assumptions We assume the following properties of the input random variables.

1. **Bounded covariances**: Eigenvalues of population auto-covariance matrices are bounded:[2]

$$\max\left(\|\mathbf{E}_{xx}\|, \|\mathbf{E}_{yy}\|\right) \le 1,$$
$$\gamma := \min\left(\sigma_{\min}(\mathbf{E}_{xx}), \sigma_{\min}(\mathbf{E}_{yy})\right) > 0.$$

   Hence $\mathbf{E}_{xx}$ and $\mathbf{E}_{yy}$ are invertible with condition numbers bounded by $1/\gamma$.

2. **Singular value gap**: For the purpose of learning the canonical directions $(\mathbf{u}^*, \mathbf{v}^*)$, we assume that there exists a positive singular value gap $\Delta := \sigma_1(\mathbf{T}) - \sigma_2(\mathbf{T}) \in (0, 1)$, such that the top left- and right-singular vector pair of $\mathbf{T}$ is uniquely defined.

Distribution classes In this paper, we analyze three input distribution classes commonly used in the statistics and machine learning literature. Let

$$\mathbf{z} = \begin{bmatrix} \mathbf{E}_{xx} & \mathbf{E}_{xy} \\ \mathbf{E}_{xy}^\top & \mathbf{E}_{yy} \end{bmatrix}^{-\frac{1}{2}} \cdot \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \in \mathbb{R}^d, \tag{5}$$

2. CCA is invariant to linear transformations of the inputs, so we could always rescale the data.

the distribution classes are defined with $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ as follows.

- **(Sub-Gaussian)** Let $\mathbf{z}$ be isotropic and sub-Gaussian, that is, $\mathbb{E}\left[\mathbf{z}\mathbf{z}^\top\right] = \mathbf{I}$ and there exists constant $C > 0$ such that $\mathbb{P}\left(\left|\mathbf{q}^\top \mathbf{z}\right| > t\right) \le \exp(-Ct^2)$ for any unit vector $\mathbf{q}$.
- **(Regular polynomial-tail, Srivastava and Vershynin, 2013)** Let $\mathbf{z}$ be isotropic and regular polynomial-tail, that is, $\mathbb{E}\left[\mathbf{z}\mathbf{z}^\top\right] = \mathbf{I}$ and there exist constants $r > 1, C > 0$ such that $\mathbb{P}\left(\|\mathbf{V}\mathbf{z}\|^2 > t\right) \le Ct^{-1-r}$ for any orthogonal projection $\mathbf{V}$ in $\mathbb{R}^d$ and any $t > C \cdot \operatorname{rank}(\mathbf{V})$. Note that this class is general and only implies the existence of a $(4+\delta)$-moment condition for some $\delta > 0$.
- **(Bounded)** Let $\mathbf{x}$ and $\mathbf{y}$ be bounded and in particular $\sup\left(\|\mathbf{x}\|^2, \|\mathbf{y}\|^2\right) \le 1$ (which implies $\max\left(\|\mathbf{E}_{xx}\|, \|\mathbf{E}_{yy}\|\right) \le 1$ as in Assumption 1).

As shown later, these classes satisfy the same concentration property, allowing us to study them (and potentially other distributions) in a unified framework.

Measure of error For an estimate $(\mathbf{u}, \mathbf{v})$ of the optimal solution to (2), which need not be correctly normalized (i.e., they may not satisfy the constraints of (2)), we can always define $(\overline{\mathbf{u}}, \overline{\mathbf{v}}) := \left(\frac{\mathbf{u}}{\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\|}, \frac{\mathbf{v}}{\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\|}\right)$ as the correctly normalized version. And we can measure the quality of these directions by the alignment (cosine of the angle) between $\left(\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}} \\ \mathbf{E}_{yy}^{\frac{1}{2}}\overline{\mathbf{v}} \end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^* \\ \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^* \end{bmatrix}\right)$, or the sum of alignment between $\left(\mathbf{E}_{xx}^{\frac{1}{2}}\overline{\mathbf{u}}, \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*\right)$ and alignment between $\left(\mathbf{E}_{yy}^{\frac{1}{2}}\overline{\mathbf{v}}, \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^*\right)$ (all vectors have unit length):

$$\operatorname{align}\left((\mathbf{u}, \mathbf{v}); (\mathbf{u}^*, \mathbf{v}^*)\right) := \frac{1}{2}\left(\frac{\mathbf{u}^\top \mathbf{E}_{xx}\mathbf{u}^*}{\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\|} + \frac{\mathbf{v}^\top \mathbf{E}_{yy}\mathbf{v}^*}{\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\|}\right).$$

This measure of alignment is invariant to the lengths of $\mathbf{u}$ and $\mathbf{v}$, and achieves the maximum of 1 if $(\mathbf{u}, \mathbf{v})$ lie in the same direction as $(\mathbf{u}^*, \mathbf{v}^*)$. Intuitively, this measure respects the geometry imposed by the CCA constraints that the projections of each view have unit length. As we will show later, this measure is also closely related to the learning guarantee we can achieve with power iterations. Moreover, high alignment implies accurate estimate of the canonical correlation.

Lemma 1 *Let $\eta \in (0, 1)$. If $\operatorname{align}\left((\mathbf{u}, \mathbf{v}); (\mathbf{u}^*, \mathbf{v}^*)\right) \ge 1 - \frac{\eta}{8}$, then*

$$\frac{\mathbf{u}^\top \mathbf{E}_{xy}\mathbf{v}}{\sqrt{\mathbf{u}^\top \mathbf{E}_{xx}\mathbf{u}}\sqrt{\mathbf{v}^\top \mathbf{E}_{yy}\mathbf{v}}} \ge \rho_1(1 - \eta).$$

All proofs are deferred to the appendix.

3. The sample complexity of ERM

One approach to address this problem is empirical risk minization (ERM): We draw N samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ from $P(\mathbf{x}, \mathbf{y})$ and solve the empirical version of (2):

$$\max_{\mathbf{u},\mathbf{v}} \quad \mathbf{u}^\top \mathbf{\Sigma}_{xy} \mathbf{v} \quad \text{s.t.} \quad \mathbf{u}^\top \mathbf{\Sigma}_{xx} \mathbf{u} = \mathbf{v}^\top \mathbf{\Sigma}_{yy} \mathbf{v} = 1 \tag{6}$$

where the empirical covariance matrices are defined as

$$\mathbf{\Sigma}_{xy} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i \mathbf{y}_i^\top, \quad \mathbf{\Sigma}_{xx} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top, \quad \mathbf{\Sigma}_{yy} = \frac{1}{N}\sum_{i=1}^N \mathbf{y}_i \mathbf{y}_i^\top. \tag{7}$$

Similarly, define the empirical version of $\mathbf{T}$ as

$$\widehat{\mathbf{T}} := \mathbf{\Sigma}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xy} \mathbf{\Sigma}_{yy}^{-\frac{1}{2}} \in \mathbb{R}^{d_x \times d_y}. \tag{8}$$

We will approximate the population canonical correation and directions based on solution to the above empirical objective.

Before going to the detailed analysis, we highlight the key property that enable us to study different input distributions in a unified manner. In fact this property is the only place we handle the stochasticity of data in studying ERM.

Proposition 2 (Concentration property) *For any $\nu > 0$, with sufficiently large sample sizes $N_0(\nu)$, the following inequality is satisfied with high probability by sub-Gaussian, regular polynomial-tail, and bounded random variables:* [3]

$$\max\left(\|\mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xx} \mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{I}\|,\ \|\mathbf{E}_{yy}^{-\frac{1}{2}} \mathbf{\Sigma}_{yy} \mathbf{E}_{yy}^{-\frac{1}{2}} - \mathbf{I}\|,\ \|\mathbf{E}_{xx}^{-\frac{1}{2}} (\mathbf{\Sigma}_{xy} - \mathbf{E}_{xy}) \mathbf{E}_{yy}^{-\frac{1}{2}}\| \right) \le \nu. \tag{9}$$

We provide detailed bounds on $N_0(\nu)$ for different distributions in Lemma 3.

Roadmap for this section We proceed to analyze the sample complexities, eventually obtained in Theorem 9. We first analyze the concentration property of different classes in Lemma 3, and provide the number of samples needed to guarantee small perturbation between $\widehat{\mathbf{T}}$ and $\mathbf{T}$ in Lemma 6, which by the Weyl's inequality (Horn and Johnson, 1986) provides the sample complexity for learning canonical correlations (regardless of the existence of a singular value gap for $\mathbf{T}$). Then by the perturbation of singular vectors and after fixing the issue of normalization, we obtain guarantees for the alignment between the estimated and the optimal canonical directions.

3.1. Approximating the canonical correlation

We first discuss the error of approximating ρ_1 by $\widehat{\rho}_1 = \sigma_1(\widehat{\mathbf{T}})$. Observe that, although the empirical covariance matrices are unbiased estimates of their population counterparts, we do *not* have $\mathbb{E}[\widehat{\mathbf{T}}] = \mathbf{T}$ due to the nonlinear operations (matrix multiplication, inverse, and square root) involved in computing $\mathbf{T}$. Nonetheless, we can provide approximation

3. We refrain from specifying the failure probability as it only adds additional mild dependences to our results.

guarantee based on concentrations. We will separate the probabilistic property of data—the concentration property in Proposition 2—from the deterministic error analysis, and we show below that it is satisfied by distributions considered here.

Lemma 3 *Let Assumption 1 hold for the random variables. Then the concentration property* (9) *is satisfied with high probability, if*

$$N_0(\nu) \ge C' \frac{d}{\nu^2} \qquad \textit{for the sub-Gaussian class,}$$

$$N_0(\nu) \ge C' \frac{d}{\nu^{2(1+r^{-1})}} \qquad \textit{for the polynomial-tail class,}$$

$$N_0(\nu) \ge C \frac{1}{\nu^2 \gamma^2} \qquad \textit{for the bounded class.}$$

Remark 4 *When $(\mathbf{x}, \mathbf{y})$ have nonzero means, we use the unbiased estimate of covariance matrices $\mathbf{\Sigma}_{xy} = \frac{\sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{y}_i - \bar{\mathbf{y}})^\top}{N-1}$, $\mathbf{\Sigma}_{xx} = \frac{\sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top}{N-1}$, and $\mathbf{\Sigma}_{yy} = \frac{\sum_{i=1}^N (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^\top}{N-1}$ instead of those in* (7)*, where $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i$ and $\bar{\mathbf{y}} = \frac{1}{N}\sum_{i=1}^N \mathbf{y}_i$. We have similar concentration results, and all results in Sections 3 and 4 still apply.*

We will decompose the difference $\mathbf{T} - \widehat{\mathbf{T}}$ and apply the above concentration results. In the decomposition, we need to bound terms of the form $\mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xx}^{\frac{1}{2}} - \mathbf{I}$. Such bounds can be derived from our assumption on $\left\|\mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xx} \mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{I}\right\|$ using Lemma 5 below. This lemma is derived from the main result of Mathias (1997), with extra effort taken to understand the size of perturbation for which higher order error terms can be safely ignored.

Lemma 5 (Perturbation of matrix square root) *Let $\mathbf{H} \in \mathbb{R}^{d \times d}$ be positive definite, with eigenvalues in the range $[\sigma_{\min}, \sigma_{\max}]$ for some $\sigma_{\min} > 0$. Let $\mathbf{\Theta} \in \mathbb{R}^{d \times d}$ be Hermitian, satisfying $\left\|\mathbf{H}^{-\frac{1}{2}} \mathbf{\Theta} \mathbf{H}^{-\frac{1}{2}}\right\| = 1$. Then for $\zeta \le \frac{3}{4}\sigma_{\max}^{-2}\sigma_{\min}^2$, we have*

$$\left\|(\mathbf{H} + \zeta \cdot \mathbf{\Theta})^{\frac{1}{2}} \mathbf{H}^{-\frac{1}{2}} - \mathbf{I}\right\| \le C_d \cdot \zeta$$

where $C_d = \mathcal{O}(\log d)$ is independent of ζ.

Lemma 6 *Assume that we draw N samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ independently from the underlying joint distribution $P(\mathbf{x}, \mathbf{y})$ for computing the sample covariance matrices in* (7)*, and $P(\mathbf{x}, \mathbf{y})$ satisfies Assumption 1 and the concentration property* (9)*. Then for $\nu \le \frac{1}{4}\gamma^2$, we have*

$$|\widehat{\rho}_1 - \rho_1| \le \left\|\mathbf{T} - \widehat{\mathbf{T}}\right\| \le 4C_d \cdot \nu$$

where C_d is the same constant in Lemma 5.

We note that the requirement of $\nu = \mathcal{O}(\gamma^2)$ is not too constraining, since the size of the perturbation ν is closely related to the statistical error, and we are mainly interested in the regime of the statistical error going to zero. It is then straightforward to combine Lemma 3 and Lemma 6 to obtain the following sample complexities for the three distribution classes.

Corollary 7 (Sample complexity for learning canonical correlation by ERM) *Let $\epsilon' \in (0,1)$ and $\epsilon' \le C_d\gamma^2$. Then for $N \ge N_0\left(\frac{\epsilon'}{4C_d}\right)$, i.e,*

$$N \ge C\frac{d\log^2 d}{\epsilon'^2} \qquad \textit{for the sub-Gaussian class,}$$
$$N \ge C\frac{d\log^{2(1+r^{-1})} d}{\epsilon'^{2(1+r^{-1})}} \qquad \textit{for the polynomial-tail class,}$$
$$N \ge C\frac{\log^2 d}{\epsilon'^2\gamma^2} \qquad \textit{for the bounded class,}$$

we have with high probability that $|\widehat{\rho}_1 - \rho_1| \le \epsilon'$.

Remark 8 *Due to better concentration properties, the sample complexity for the sub-Gaussian and regular polynomial-tail classes are independent of the condition number $\frac{1}{\gamma}$ of the auto-covariances.*

Comparison to Arora et al. (2017) In a parallel work by Arora et al. (2017), the authors studied the top-k stochastic CCA for bounded inputs, and proposed stochastic approximation-type algorithms with $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon'^2\gamma^2}\right)$ sample-complexity upper bound for approximating the top canonical correlation. We note, however, their stochastic algorithms are derived from the convex relaxation of stochastic CCA, which lifts the original problem into the space of matrices in $\mathbb{R}^{d_x\times d_y}$ and requires a whitening operation (multiplying each fresh sample by $\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}$ or $\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}$) and a projection operation (onto the set of low 2-norm and nuclear-norm matrices) in each iteration, which are inefficient in high dimensions. Our work studies three different classes of input distributions in a uniform manner[4], with the goal of matching the statistical limits for the Gaussian inputs (see Section 5.2). The algorithms we provide in the next sections require only elementary vector operations and thus more practical for high dimensional data.

3.2. Approximating the canonical directions

We now discuss the error in learning $(\mathbf{u}^*, \mathbf{v}^*)$ by ERM, when $\mathbf{T}$ has a singular value gap $\Delta > 0$. Let the nonzero singular values of $\mathbf{T}$ be $1 \ge \rho_1 \ge \rho_2 \ge \cdots \ge \rho_r$, where $r = \mathrm{rank}(\mathbf{T}) \le \min(d_x, d_y)$, and the corresponding (unit-length) singular vector pairs be $(\mathbf{a}_1, \mathbf{b}_1), \ldots, (\mathbf{a}_r, \mathbf{b}_r)$. Define

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{T} \\ \mathbf{T}^\top & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{d\times d}. \tag{10}$$

The eigenvalues of $\mathbf{C}$ are

$$\rho_1 \ge \cdots \ge \rho_r > 0 = \cdots = 0 > -\rho_r \ge \cdots \ge -\rho_1,$$

4. If we only study the case of bounded inputs, we can bypass Lemma 5 and the $\log d$ dependence in our bound can be reduced.

with corresponding unit eigenvectors

$$\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \end{bmatrix}, \ldots, \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{a}_r \\ \mathbf{b}_r \end{bmatrix}, \ldots, \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{a}_r \\ -\mathbf{b}_r \end{bmatrix}, \ldots, \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{a}_1 \\ -\mathbf{b}_1 \end{bmatrix}.$$

Thus, learning canonical directions $(\mathbf{u}^*, \mathbf{v}^*)$ reduces to learning the top eigenvector of $\mathbf{C}$.

We denote the empirical version of $\mathbf{C}$ by $\widehat{\mathbf{C}}$, and the singular vector pairs of $\widehat{\mathbf{T}}$ by $\{(\widehat{\mathbf{a}}_i, \widehat{\mathbf{b}}_i)\}$. Due to the block structure of $\mathbf{C}$ and $\widehat{\mathbf{C}}$, we have $\left\|\mathbf{C} - \widehat{\mathbf{C}}\right\| = \left\|\mathbf{T} - \widehat{\mathbf{T}}\right\|$. Let the ERM solution be $(\widehat{\mathbf{u}}, \widehat{\mathbf{v}}) = \left(\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\widehat{\mathbf{a}}_1, \mathbf{\Sigma}_{yy}^{-\frac{1}{2}}\widehat{\mathbf{b}}_1\right)$, which satisfy $\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\| = \left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}\right\| = 1$. We now state the sample complexity for learning the canonical directions by ERM.

Theorem 9 *Let $\epsilon \in (0,1)$ and $\epsilon \le \frac{16C_d^2\gamma^4}{\Delta^2}$. Then for $N \ge N_0\left(\frac{\sqrt{\epsilon}\Delta}{16C_d}\right)$, i.e.,*

$$N \ge C\frac{d\log^2 d}{\epsilon\Delta^2} \qquad \textit{for the sub-Gaussian class,}$$
$$N \ge C\frac{d\log^{2(1+r^{-1})} d}{\epsilon^{(1+r^{-1})}\Delta^2} \qquad \textit{for the regular polynomial-tail class,}$$
$$N \ge C\frac{\log^2 d}{\epsilon\Delta^2\gamma^2} \qquad \textit{for the bounded class,}$$

we have with high probability that align$((\widehat{\mathbf{u}}, \widehat{\mathbf{v}}); (\mathbf{u}^*, \mathbf{v}^*)) \ge 1 - \epsilon$.

Proof sketch The proof of Theorem 9 consists of two steps. We first bound the error between $\widehat{\mathbf{C}}$'s top eigenvector $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}} \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}} \end{bmatrix}$ and $\mathbf{C}$'s top eigenvector $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^* \\ \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^* \end{bmatrix}$ using a standard result on perturbation of eigenvectors, namely the Davis-Kahan $\sin\theta$ theorem (Davis and Kahan, 1970) which states $\sin^2\theta \le \frac{\left\|\mathbf{C}-\widehat{\mathbf{C}}\right\|^2}{\Delta^2} \le \frac{\epsilon'^2}{\Delta^2}$ where θ is the angle between top eigenvectors of $\mathbf{C}$ and $\widehat{\mathbf{C}}$. We then show that $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}} \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}} \end{bmatrix}$ is very close to the "correctly normalized" $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}/\|\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\| \\ \mathbf{E}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}/\|\mathbf{E}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}\| \end{bmatrix}$, so the later still aligns well with the population solution.

Comparison to prior analysis For the sub-Gaussian class, the tightest analysis of the sample complexity upper bound we are aware of was by Gao et al. (2017). However, their proof relies on the assumption that $\rho_2 = o(\rho_1)$, i.e., they require that $\rho_2 \ll \rho_1$. In contrast, we do not require this assumption, and our bound is sharp in terms of the gap $\Delta = \rho_1 - \rho_2$. Up to the $\log^2 d$ factor, our ERM sample complexity for the same loss matches the minimax lower bound $\frac{d}{\epsilon\Delta^2}$ given by Gao et al. (2017) (see also Section 5.2).

4. Stochastic optimization for ERM

A disadvantage of the empirical risk minimization approach is that it can be time and memory consuming. To obtain the exact solution to (6), we need to explicitly form and

store the covariance matrices and to compute their singular value decompositions (SVDs); these steps have a time complexity of $\mathcal{O}(Nd^2 + d^3)$ and a memory complexity of $\mathcal{O}(d^2)$.

In this section, we study the stochastic optimization of the empirical objective, and show that the computational complexity is low: We just need to process a large enough dataset (with the same level of samples as ERM requires) nearly constant times in order to achieve small error with respect to the population objective. The basic algorithm we use here is the shift-and-invert meta-algorithm proposed by Wang et al. (2016). However, in this section we provide refined analysis of the algorithm's time complexity than that provided by Wang et al. (2016). We show that, using a better measure of progress and careful initializations for each least squares problem, the algorithm enjoys linear convergence (see Theorem 12), i.e., the time complexity for achieving η-suboptimalilty in the empirical objective depends on $\log \frac{1}{\eta}$, whereas the result of Wang et al. (2016) has a dependence of $\log^2 \frac{1}{\eta}$.

We also note that the recent work of Allen-Zhu and Li (2016) and Allen-Zhu and Li (2017) have extended the ERM problem to extracting the top $k \geq 1$ pairs of canonical directions, and applied the technique of peeling/deflation together with shift-and-invert. However, their convergence rate for the fist pair of canonical directions does not improve that of Wang et al. (2016).[5] As mentioned above, our result strictly improves that of Wang et al. (2016), and in particular replaces the $\tilde{\mathcal{O}}(\cdot)$ notation with the $\mathcal{O}(\cdot)$ notation in total runtime, achieving true linear convergence.

Roadmap for this section We first introduce the shift-and-invert power iterations and provide its iteration complexity, assuming that each matrix-vector multiplication or equivalently a convex least squares problem is solved to sufficient accuracy (Lemma 10). We then show each least squares can be warm-started using rescaled estimates from the previous iteration (Lemma 11). Finally, we plug in the time complexity of SVRG for each subproblem, and give runtime complexities for each distribution class which have different "condition numbers" (Corollary 13).

The condition numbers depend on, among other things, the smallest eigenvalues of the covariance matrices, which are bounded away from zero as discussed below.

Eigenvalues of empirical covariance According to the analysis of ERM from previous section, we have been working in the regime that the concentration parameter in (9) satisfies $\nu \leq \frac{\gamma^2}{4} \leq \frac{\gamma}{2}$. Thus in view of Assumption 1, we have with high probability that

$$\|\mathbf{\Sigma}_{xx} - \mathbf{E}_{xx}\| = \left\|\mathbf{E}_{xx}^{\frac{1}{2}}(\mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{\Sigma}_{xx}\mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{I})\mathbf{E}_{xx}^{\frac{1}{2}}\right\| \leq \|\mathbf{E}_{xx}\| \cdot \left\|\mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{\Sigma}_{xx}\mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{I}\right\| \leq \frac{\gamma}{2}$$

and similarly $\|\mathbf{\Sigma}_{yy} - \mathbf{E}_{yy}\| \leq \frac{\gamma}{2}$. According to Weyl's inequality, these inequalities make sure eigenvalues of $\mathbf{\Sigma}_{xx}$ and $\mathbf{\Sigma}_{yy}$ lie in $[\frac{\gamma}{2}, 1+\frac{\gamma}{2}]$, and consequently the involved subproblems are strongly-convex and can be solved efficiently.

5. See the second and third last lines of Table 1, and the last paragraph of Section 1.2 in Allen-Zhu and Li (2017): "Our running time matches that of [29] when $k = 1$".

4.1. Shift-and-invert power iterations

Our algorithm runs the shift-and-invert power iterations on the following matrix

$$\widehat{\mathbf{M}}_\lambda = \left(\lambda\mathbf{I} - \widehat{\mathbf{C}}\right)^{-1} = \begin{bmatrix} \lambda\mathbf{I} & -\widehat{\mathbf{T}} \\ -\widehat{\mathbf{T}}^\top & \lambda\mathbf{I} \end{bmatrix}^{-1} \tag{11}$$

where $\lambda > \widehat{\rho}_1$. It is straightforward to see that $\widehat{\mathbf{M}}_\lambda$ is positive definite with eigenvalues

$$\frac{1}{\lambda - \widehat{\rho}_1} \geq \cdots \geq \frac{1}{\lambda - \widehat{\rho}_r} \geq \cdots \geq \frac{1}{\lambda + \widehat{\rho}_r} \geq \cdots \geq \frac{1}{\lambda + \widehat{\rho}_1},$$

and has the same set of eigenvectors as $\widehat{\mathbf{C}}$.

Assume that there exists a singular value gap for $\widehat{\mathbf{T}}$ (this can be guaranteed by drawing sufficiently many samples so that the singular values of $\widehat{\mathbf{T}}$ are within a fraction of the gap Δ of $\mathbf{T}$), denoted as $\widehat{\Delta} = \widehat{\rho}_1 - \widehat{\rho}_2$. The key observation is that, as opposed to running power iterations on $\widehat{\mathbf{C}}$ (which is essentially done by Ge et al. 2016), $\widehat{\mathbf{M}}_\lambda$ has a large eigenvalue gap when $\lambda = \widehat{\rho}_1 + c(\widehat{\rho}_1 - \widehat{\rho}_2)$ with $c = \mathcal{O}(1)$, and thus power iterations on $\widehat{\mathbf{M}}_\lambda$ converge more quickly. In particular, we assume for now the availability of an estimated eigenvalue λ such that $\lambda - \widehat{\rho}_1 \in [l\widehat{\Delta}, u\widehat{\Delta}]$ where $0 < l < u < 1$; locating such a λ is discussed later in Remark 14.

Define

$$\widehat{\mathbf{A}}_\lambda := \begin{bmatrix} \lambda\mathbf{\Sigma}_{xx} & -\mathbf{\Sigma}_{xy} \\ -\mathbf{\Sigma}_{xy}^\top & \lambda\mathbf{\Sigma}_{yy} \end{bmatrix}, \qquad \widehat{\mathbf{B}} := \begin{bmatrix} \mathbf{\Sigma}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{yy} \end{bmatrix},$$

and we have $\widehat{\mathbf{M}}_\lambda = \widehat{\mathbf{B}}^{\frac{1}{2}}\widehat{\mathbf{A}}_\lambda^{-1}\widehat{\mathbf{B}}^{\frac{1}{2}}$. And by the relationship $\widehat{\mathbf{A}}_\lambda = \widehat{\mathbf{B}}^{\frac{1}{2}}\widehat{\mathbf{M}}_\lambda^{-1}\widehat{\mathbf{B}}^{\frac{1}{2}}$, eigenvalues of $\widehat{\mathbf{A}}_\lambda$ are bounded:

$$\sigma_{\max}\left(\widehat{\mathbf{A}}_\lambda\right) \leq \sigma_{\max}\left(\widehat{\mathbf{M}}_\lambda^{-1}\right) \cdot \sigma_{\max}\left(\widehat{\mathbf{B}}\right) \leq (\lambda + \widehat{\rho}_1)(1 + \frac{\gamma}{2}),$$
$$\sigma_{\min}\left(\widehat{\mathbf{A}}_\lambda\right) \geq \sigma_{\min}\left(\widehat{\mathbf{M}}_\lambda^{-1}\right) \cdot \sigma_{\min}\left(\widehat{\mathbf{B}}\right) \geq (\lambda - \widehat{\rho}_1)\gamma/2.$$

It is convenient to study the convergence in the concatenated variables

$$\mathbf{w}_t := \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix}, \quad \mathbf{r}_t := \widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{w}_t = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_t \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_t \end{bmatrix}.$$

Define the following quantities using the ERM solution

$$\widehat{\mathbf{w}} := \frac{1}{\sqrt{2}}\begin{bmatrix} \widehat{\mathbf{u}} \\ \widehat{\mathbf{v}} \end{bmatrix}, \qquad \widehat{\mathbf{r}} := \widehat{\mathbf{B}}^{\frac{1}{2}}\widehat{\mathbf{w}} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}} \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}} \end{bmatrix},$$

which satisfy $\widehat{\mathbf{w}}^\top\widehat{\mathbf{B}}\widehat{\mathbf{w}} = 1$ and $\widehat{\mathbf{r}}^\top\widehat{\mathbf{r}} = 1$ respectively.

4.2. Convergence of inexact shift-and-invert

Our algorithm iteratively applies the approximate matrix-vector multiplications: for $t = 0, 1, \ldots$

$$\mathbf{r}_{t+1} \approx \widehat{\mathbf{M}}_\lambda \mathbf{r}_t, \qquad \Longleftrightarrow \qquad \mathbf{w}_{t+1} \approx \widehat{\mathbf{A}}_\lambda^{-1} \widehat{\mathbf{B}} \mathbf{w}_t. \tag{12}$$

This equivalence allows us to directly work with $(\mathbf{u}_t, \mathbf{v}_t)$ and avoids computing $\mathbf{\Sigma}_{xx}^{\frac{1}{2}}$ or $\mathbf{\Sigma}_{yy}^{\frac{1}{2}}$ explicitly. Note that we do not perform normalizations of the form $\mathbf{w}_t \leftarrow \mathbf{w}_t / \left\| \widehat{\mathbf{B}}^{\frac{1}{2}} \mathbf{w}_t \right\|$ at each iteration as done by Wang et al. (2016) (Phase-I of their SI meta-algorithm); the length of each iterate is irrelevant for the purpose of optimizing the alignment between vectors and we could always perform the normalization in the end to satisfy the length constants. Exact power iterations is known to converge linearly when there exist an eigenvalue gap (Golub and van Loan, 1996).

The matrix-vector multiplication $\widehat{\mathbf{A}}_\lambda^{-1} \widehat{\mathbf{B}} \mathbf{w}_t$ is equivalent to solving the least squares problem

$$\min_{\mathbf{w}} \; f_{t+1}(\mathbf{w}) := \frac{1}{2} \mathbf{w}^\top \widehat{\mathbf{A}}_\lambda \mathbf{w} - \mathbf{w}^\top \widehat{\mathbf{B}} \mathbf{w}_t \tag{13}$$

whose unique solution is $\mathbf{w}_{t+1}^* = \widehat{\mathbf{A}}_\lambda^{-1} \widehat{\mathbf{B}} \mathbf{w}_t$ with the optimal objective $f_{t+1}^* = -\frac{1}{2} \mathbf{w}_t^\top \widehat{\mathbf{B}} \widehat{\mathbf{A}}_\lambda^{-1} \widehat{\mathbf{B}} \mathbf{w}_t$. Of course, solving the problem exactly is costly and we will apply stochastic gradient methods to it. We will show that, when the least squares problems are solved accurately enough, the iterates are of the same quality as those of the exact solutions and enjoys linear convergence.

We begin by introducing the measure of progress for the iterates. Denote the eigenvalues of $\widehat{\mathbf{M}}_\lambda$ by $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_d$, with corresponding eigenvectors $\mathbf{p}_1, \ldots, \mathbf{p}_d$ forming an orthonormal basis of $\mathbb{R}^d$. Recall that $\mathbf{p}_1 = \widehat{\mathbf{r}}$, $\mathbf{p}_i^\top \widehat{\mathbf{M}}_\lambda \mathbf{p}_i = \beta_i$ for $i = 1, \ldots, d$, and $\mathbf{p}_i^\top \widehat{\mathbf{M}}_\lambda \mathbf{p}_j = 0$ for $i \neq j$.

We therefore can write each iterate as a linear combination of the eigenvectors: $\frac{\mathbf{r}_t}{\|\mathbf{r}_t\|} = \sum_{i=1}^d \xi_{ti} \mathbf{p}_i$, where $\xi_{ti} = \frac{\mathbf{r}_t^\top \mathbf{p}_i}{\|\mathbf{r}_t\|}$ for $i = 1, \ldots, d$, and $\sum_{i=1}^d \xi_{ti}^2 = 1$. The potential function we use to evaluate the progress of each iteration is

$$G(\mathbf{r}_t) = \frac{\left\| \mathbf{P}_\perp \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|} \right\|_{\widehat{\mathbf{M}}_\lambda^{-1}}}{\left\| \mathbf{P}_\parallel \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|} \right\|_{\widehat{\mathbf{M}}_\lambda^{-1}}} = \frac{\sqrt{\sum_{i=2}^d \xi_{ti}^2 / \beta_i}}{\sqrt{\xi_{t1}^2 / \beta_1}},$$

where $\mathbf{P}_\perp$ and $\mathbf{P}_\parallel$ denote projections onto the subspaces perpendicular and parallel to $\widehat{\mathbf{r}}$ respectively.

The same potential function was used by Garber et al. (2016) for analyzing the convergence of shift-and-invert for PCA. The potential function is invariant to the length of $\mathbf{r}_t$, and is equivalent to the criterion $|\tan \theta_t| := \frac{\sqrt{\sum_{i=2}^d \xi_{ti}^2}}{\sqrt{\xi_{t1}^2}}$ where θ_t is the angle between $\mathbf{r}_t$ and $\widehat{\mathbf{r}}$: in the following sense:

$$|\sin \theta_t| = \sqrt{\sum_{i=2}^d \xi_{ti}^2} \leq \sqrt{\frac{\beta_1}{\beta_2}} |\tan \theta_t| \leq G(\mathbf{r}_t) \leq \sqrt{\frac{\beta_1}{\beta_d}} |\tan \theta_t| .$$

The lemma below shows that under the iterative scheme (12), $\{G(\mathbf{r}_t)\}_{t=1,\ldots}$ converges linearly to 0.

Lemma 10 *Let $\eta \in (0, 1)$. Assume that for each approximate matrix-vector multiplication, we solve the least squares problem so accurately that the approximate solution $\mathbf{w}_{t+1}$ satisfies*

$$\epsilon_t := \frac{f_{t+1}(\mathbf{w}_{t+1}) - f_{t+1}^*}{\mathbf{w}_t^\top \widehat{\mathbf{B}} \mathbf{w}_t} \leq \min \left(\sum_{i=2}^d \xi_{ti}^2 / \beta_i, \; \xi_{t1}^2 / \beta_1 \right) \cdot \frac{(\beta_1 - \beta_2)^2}{32}. \tag{14}$$

Let $T = \lceil \log_{\frac{7}{5}} \left(\frac{G(\mathbf{r}_0)}{\eta} \right) \rceil$. Then we have $|\sin \theta_t| \leq G(\mathbf{r}_t) \leq \eta$ for all $t \geq T$.

4.3. Bounding initial error for least squares

It is natural to use an initialization of the form $\alpha \mathbf{w}_t$ for minimizing $f_{t+1}(\mathbf{w})$. The following lemma provides the optimal α and the resulting initial suboptimality, see detailed analysis in Appendix D.2.

Lemma 11 (Warm start for least squares) *Initializing $\min_{\mathbf{w}} f_{t+1}(\mathbf{w})$ with $\alpha_t^* \mathbf{w}_t$ where $\alpha_t^* = \frac{\mathbf{w}_t^\top \widehat{\mathbf{B}} \mathbf{w}_t}{\mathbf{w}_t^\top \widehat{\mathbf{A}}_\lambda \mathbf{w}_t}$, it suffices to set the ratio between the initial and the final error to be $64 \cdot \max(1, G(\mathbf{r}_t))$ so that (14) is satisfied.*

This result indicates that in the converging stage ($G(\mathbf{r}_t) \leq 1$), we just need to set the ratio between the initial and the final error to the constant 64 (and set it to be the constant $64 G(\mathbf{r}_0)$ before that). This will ensure that the time complexity of least squares has no dependence on the final error ϵ.

4.4. Solving the least squares by SGD

The least squares objective (13) is the sum of N functions: $f_{t+1}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N f_{t+1}^i(\mathbf{w})$ where

$$f_{t+1}^i(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\top \begin{bmatrix} \lambda \mathbf{x}_i \mathbf{x}_i^\top & -\mathbf{x}_i \mathbf{y}_i^\top \\ -\mathbf{y}_i \mathbf{x}_i^\top & \lambda \mathbf{y}_i \mathbf{y}_i^\top \end{bmatrix} \mathbf{w} - \mathbf{w}^\top \begin{bmatrix} \mathbf{\Sigma}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{yy} \end{bmatrix} \mathbf{w}_t. \tag{15}$$

There has been much recent progress on developping linearly convergent stochastic algorithms for solving finite-sum problems. We use SVRG (Johnson and Zhang, 2013) here due to its algorithmic simplicity and memory efficiency; in the next section, we will be using the "online" version of SVRG for stochastic CCA in the streaming setting. Note that although $f_{t+1}(\mathbf{w})$ is convex, each component f_{t+1}^i may not be convex.

We provide the time complexity of SVRG for this case (based on Garber and Hazan, 2015, Appendix B), as well as the "condition number" for the three classes of distributions in Appendix D.3 and D.4 respectively.

4.5. Total time complexity

We first provide the runtime for solving the empirical objective using the (offline) shift-and-invert CCA algorithm.

Theorem 12 *Let $\eta \in (0,1)$. Draw N samples for ERM such that $\sigma_{\min}(\mathbf{\Sigma}_{xx}) \geq \frac{\gamma}{2}$ and $\sigma_{\min}(\mathbf{\Sigma}_{yy})) \geq \frac{\gamma}{2}$. Initialize $\mathbf{w}_0 = \frac{\tilde{\mathbf{w}}_0}{\sqrt{\tilde{\mathbf{w}}_0^\top \widehat{\mathbf{B}} \tilde{\mathbf{w}}_0}}$ where entries of $\tilde{\mathbf{w}}_0 \in \mathbb{R}^d$ are randomly sampled from the standard Gaussian distribution. Then with high probability, offline shift-and-invert outputs an $(\mathbf{u}_T, \mathbf{v}_T)$ satisfying $\min\left(\frac{\mathbf{u}_T{}^\top \mathbf{\Sigma}_{xx}\widehat{\mathbf{u}}}{\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\|}, \frac{\mathbf{v}_T{}^\top \mathbf{\Sigma}_{yy}\widehat{\mathbf{v}}}{\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\|}\right) \geq 1-\eta$ in total time*

$$\mathcal{O}\left(d\left(N + \frac{d^2}{\widehat{\Delta}^2\gamma^2}\right)\log\frac{d}{\widehat{\Delta}\gamma}\log\frac{d}{\widehat{\Delta}\gamma\eta}\right) \qquad \textit{for sub-Gaussian/polynomial-tail,}$$

$$\mathcal{O}\left(d\left(N + \frac{1}{\widehat{\Delta}^2\gamma^2}\right)\log\frac{d}{\widehat{\Delta}\gamma}\log\frac{d}{\widehat{\Delta}\gamma\eta}\right) \qquad \textit{for the bounded class.}$$

We have already shown in Theorem 9 that the ERM solution aligns well with the population solution. By drawing slighly more samples and requiring our algorithm to find an approximate solution that aligns well with the ERM solution, we can guarantee high alignment for the approximate solution as shown in the following corollary.

Corollary 13 *Let $\epsilon \in (0,1)$ and $\epsilon \leq \frac{64C_d^2\gamma^4}{\Delta^2}$. Draw $N = N_0\left(\frac{\sqrt{\epsilon}\Delta}{32C_d}\right)$ samples for the ERM objective, and use the initialization strategy in Theorem 12. Then with high probability, the total time for offline shift-and-invert to output $(\mathbf{u}_T, \mathbf{v}_T)$ with $\mathit{align}((\mathbf{u}_T, \mathbf{v}_T); (\mathbf{u}^*, \mathbf{v}^*)) \geq 1-\epsilon$ is*

$$\mathcal{O}\left(d\left(\frac{d\log^2 d}{\epsilon\Delta^2} + \frac{d^2}{\Delta^2\gamma^2}\right)\log\frac{d}{\Delta\gamma}\log\frac{d}{\Delta\gamma\epsilon}\right) \qquad \textit{for sub-Gaussian,}$$

$$\mathcal{O}\left(d\left(\frac{d\log^{2(1+r^{-1})} d}{\epsilon^{(1+r^{-1})}\Delta^2} + \frac{d^2}{\Delta^2\gamma^2}\right)\log\frac{d}{\Delta\gamma}\log\frac{d}{\Delta\gamma\epsilon}\right) \qquad \textit{for polynomial-tail,}$$

$$\mathcal{O}\left(d\left(\frac{\log^2 d}{\epsilon\Delta^2\gamma^2} + \frac{1}{\Delta^2\gamma^2}\right)\log\frac{d}{\Delta\gamma}\log\frac{d}{\Delta\gamma\epsilon}\right) \qquad \textit{for the bounded class.}$$

The ϵ-dependent term is near-linear in the ERM sample complexity $N(\epsilon, \Delta, \gamma)$ and is also the dominant term in the total runtime (when $\epsilon = o(\gamma^2)$ for the first two classes). For sub-Gaussian/regular polynomial-tail classes, we incur an undesirable d^2 dependence for the least squares problem's condition number (see more details in Appendix D.3), mainly due to weak concentration regarding the data norm (we have stronger concentration for the streaming setting discussed next). One can alleviate the issue of large condition number using accelerated SVRG (Lin et al., 2015).

Remark 14 *We have assumed so far the availability of $\lambda = \widehat{\rho}_1 + c(\widehat{\rho}_1 - \widehat{\rho}_2)$ with $c = \mathcal{O}(1)$ for shift-and-invert to work. There exists an efficient algorithm for locating such an λ, see the* ***repeat-until*** *loop of Algorithm 3 in Wang et al. (2016). This procedure computes $\mathcal{O}\left(\log\frac{1}{\Delta}\right)$ approximate matrix-vector multiplications, and its time complexity does not depend on ϵ as we only want to achieve good estimate of the top eigenvalue (and not the top eigenvector). So the cost of locating λ is not dominant in the total runtime.*

5. Streaming shift-and-invert CCA

A disadvantage of the ERM approach is that we need to store all the samples in order to go through the dataset multiple times. We now study the shift-and-invert algorithms in the streaming setting in which we draw samples from the underlying distribution $P(\mathbf{x}, \mathbf{y})$ and process them once. Clearly, the streaming approach requires only $\mathcal{O}(d)$ memory.

In this section, we assume the availability of a $\lambda = \rho_1 + c\Delta$, where $0 < c < 1$.[6] Our algorithm is the same as in the ERM case, except that we now directly work with the population covariances through fresh samples instead of their empirical estimates. With slight abuse of notation, we use $(\mathbf{A}_\lambda, \mathbf{B}, \mathbf{M}_\lambda)$ to denote the population version of $(\widehat{\mathbf{A}}_\lambda, \widehat{\mathbf{B}}, \widehat{\mathbf{M}}_\lambda)$:

$$\mathbf{A}_\lambda := \begin{bmatrix} \lambda\mathbf{E}_{xx} & -\mathbf{E}_{xy} \\ -\mathbf{E}_{xy}^\top & \lambda\mathbf{E}_{yy} \end{bmatrix}, \qquad \mathbf{B} := \begin{bmatrix} \mathbf{E}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_{yy} \end{bmatrix}, \qquad \mathbf{M}_\lambda = \mathbf{B}^{\frac{1}{2}}\mathbf{A}_\lambda^{-1}\mathbf{B}^{\frac{1}{2}},$$

use $\{(\beta_i, \mathbf{p}_i)\}_{i=1}^d$ to denote the eigensystem of $\mathbf{M}_\lambda$, and use $(\mathbf{u}_t, \mathbf{v}_t)$ as well as

$$\mathbf{w}_t = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix}, \quad \mathbf{r}_t = \mathbf{B}^{\frac{1}{2}}\mathbf{w}_t = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}_t \\ \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}_t \end{bmatrix},$$

$t = 0, \ldots$ to denote the iterates of our algorithm. Also, define ξ_{ti}, θ_t and $G(\mathbf{r}_t)$ similarly as in Section 4.

Handling normalizations It is sufficient to achieve high alignment between

$$\frac{\mathbf{r}_T}{\|\mathbf{r}_T\|} = \begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}_T \\ \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}_T \end{bmatrix} \Big/ \sqrt{\mathbf{u}_T^\top\mathbf{E}_{xx}\mathbf{u}_T + \mathbf{v}_T^\top\mathbf{E}_{yy}\mathbf{v}_T}$$

where $(\mathbf{u}, \mathbf{v})$ are normalized jointly, and $\mathbf{r}^* = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^* \\ \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^* \end{bmatrix}$ where $(\mathbf{u}, \mathbf{v})$ are normalized separately. According to the lemma below, this would imply high alignment between $\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}_T/\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}_T\| \\ \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}_T/\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}_T\| \end{bmatrix}$ and $\mathbf{r}^*$ which is our final goal.

Lemma 15 (Conversion from joint alignment to separate alignment) *Let $\eta \in (0,1)$. If the output $(\mathbf{u}_T, \mathbf{v}_T)$ of our online shift-and-invert algorithm satisfy that*

$$\frac{1}{\sqrt{2}} \cdot \frac{\mathbf{u}^{*\top}\mathbf{E}_{xx}\mathbf{u}_T + \mathbf{v}^{*\top}\mathbf{E}_{yy}\mathbf{v}_T}{\sqrt{\mathbf{u}_T^\top\mathbf{E}_{xx}\mathbf{u}_T + \mathbf{v}_T^\top\mathbf{E}_{yy}\mathbf{v}_T}} \geq 1 - \frac{\eta}{4},$$

we also have

$$\mathit{align}((\mathbf{u}_T, \mathbf{v}_T); (\mathbf{u}^*, \mathbf{v}^*)) = \frac{1}{2}\left(\frac{\mathbf{u}^{*\top}\mathbf{E}_{xx}\mathbf{u}_T}{\sqrt{\mathbf{u}_T^\top\mathbf{E}_{xx}\mathbf{u}_T}} + \frac{\mathbf{v}^{*\top}\mathbf{E}_{yy}\mathbf{v}_T}{\sqrt{\mathbf{v}_T^\top\mathbf{E}_{yy}\mathbf{v}_T}}\right) \geq 1 - \eta.$$

6. Based on the same intuition given in Remark 14, we believe that a procedure similar to that of Wang et al. (2016) also works in the streaming setting and the cost in locating λ is not dominant, although we do not have a formal analysis.

Algorithm 1 Streaming SVRG for $\min_{\mathbf{w}} \ f(\mathbf{w})$.

Input: Initialization $\mathbf{w}^0 = \mathbf{0}$, stepsize scaling factor $s = \frac{1}{352}$, (μ, S, σ^2) are respectively the strong convexity, streaming smoothness, and streaming variance given in Lemma 16.
for $\tau = 1, \ldots, \Gamma$ **do**
  $\bar{\mathbf{z}} \leftarrow \mathbf{w}^{\tau-1}$
  $m_\tau \leftarrow \lceil \frac{44^2 S}{\mu} \rceil, \quad k_\tau \leftarrow \max\left(\lceil \frac{44S}{\mu} \rceil, \lceil \frac{20\sigma^2 \cdot 2^{\tau-1}}{\beta_1 \|\mathbf{r}_t\|^2} \rceil \right)$
  Draw k_τ samples $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{k_\tau}, \mathbf{y}_{k_\tau})$ and estimate the batch gradient

$$\mathbf{g} \leftarrow \frac{1}{k_\tau} \sum\nolimits_{i=1}^{k_\tau} \nabla \phi(\bar{\mathbf{z}}; \mathbf{x}_i, \mathbf{y}_i)$$

  Sample $\widetilde{m}_\tau$ uniformly at random from $\{1, \ldots, m_\tau\}$
  $\mathbf{z} \leftarrow \bar{\mathbf{z}}$
  for $i = 1, \ldots, \widetilde{m}_\tau$ **do**
    Draw sample $(\mathbf{x}_i, \mathbf{y}_i)$
    $\mathbf{z} \leftarrow \mathbf{z} - \frac{s}{S} \left(\nabla \phi(\mathbf{z}; \mathbf{x}_i, \mathbf{y}_i) - \nabla \phi(\bar{\mathbf{z}}; \mathbf{x}_i, \mathbf{y}_i) + \mathbf{g} \right)$
  end for
  $\mathbf{w}^\tau \leftarrow \mathbf{z}$
end for
Output: Return $\mathbf{w}^\Gamma$ as the approximate solution.

Note that Lemma 15 improves over a similar result by Wang et al. (2016, Theorem 5), which requires the joint alignment to be $\mathcal{O}(\eta^2)$-suboptimal for the separate alignment to be $\mathcal{O}(\eta)$-suboptimal.

5.1. Solving least squares by streaming SVRG

Turning to the streaming algorithm, the least squares problem at iteration $t+1$, is now a stochastic program:

$$\min_{\mathbf{w}} \ f_{t+1}(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\top \mathbf{A}_\lambda \mathbf{w} - \mathbf{w}^\top \mathbf{B} \mathbf{w}_t = \mathbb{E}\left[\phi_{t+1}(\mathbf{w}; \mathbf{x}, \mathbf{y}) \right]$$

where $\phi_{t+1}(\mathbf{w}; \mathbf{x}, \mathbf{y}) := \frac{1}{2} \mathbf{w}^\top \begin{bmatrix} \lambda \mathbf{x}\mathbf{x}^\top & -\mathbf{x}\mathbf{y}^\top \\ -\mathbf{y}\mathbf{x}^\top & \lambda \mathbf{y}\mathbf{y}^\top \end{bmatrix} \mathbf{w} - \mathbf{w}^\top \begin{bmatrix} \mathbf{x}\mathbf{x}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{y}\mathbf{y}^\top \end{bmatrix} \mathbf{w}_t$, and the expectation is computed over $P(\mathbf{x}, \mathbf{y})$. The optimal solution to this stochastic program is $\mathbf{w}_{t+1}^* = \mathbf{A}_\lambda^{-1} \mathbf{B} \mathbf{w}_t$.

Due to the high sample complexity of accurately estimating $\alpha_t^* = \frac{\mathbf{w}_t^\top \mathbf{B} \mathbf{w}_t}{\mathbf{w}_t^\top \mathbf{A}_\lambda \mathbf{w}_t}$ in the streaming setting, we instead initialize each linear systems with the zero vector. With this initialization, we have

$$f_{t+1}(\mathbf{0}) - f_{t+1}^* = 0 - \left(-\frac{1}{2} \mathbf{w}_t^\top \mathbf{B} \mathbf{A}_\lambda^{-1} \mathbf{B} \mathbf{w}_t \right) = \frac{\mathbf{r}_t^\top \mathbf{M}_\lambda \mathbf{r}_t}{2} \leq \frac{\beta_1 \|\mathbf{r}_t\|^2}{2}. \tag{16}$$

We then solve the linear system with the streaming SVRG algorithm proposed by Frostig et al. (2015), as detailed in Algorithm 1. This is the same approach taken by Garber et al.

(2016) for streaming PCA, and our analysis follows the same structure. Streaming SVRG is a natural choice here since it is the "online" version of the SVRG algorithm for optimizing empirical objectives and enjoys the same algorithmic simplicity and low computational complexity. Moreover, for stochastic least squares problems, streaming SVRG is shown to have the same sample complexity as solving the ERM problem (Frostig et al., 2015), which aligns well with our goal of an overall sample efficient algorithm. With this choice, the final algorithm is very similar to the stochastic optimization algorithm in Section 4, except that fresh samples are used for each update.

To analyze the sample complexity of streaming SVRG, we first calculate the streaming smoothness and streaming variance parameters for the three classes of distributions.

Lemma 16 (Parameters of streaming SVRG) *For any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$, we have*

- *Strong convexity:*

$$f_{t+1}(\mathbf{w}) \geq f_{t+1}(\mathbf{w}') + \left\langle \nabla f_{t+1}(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \right\rangle + \frac{\mu}{2} \left\| \mathbf{w} - \mathbf{w}' \right\|^2,$$

- *Streaming smoothness:*

$$\mathbb{E}\left[\left\| \nabla \phi_{t+1}(\mathbf{w}) - \nabla \phi_{t+1}(\mathbf{w}_{t+1}^*) \right\|^2 \right] \leq 2S \left(f_{t+1}(\mathbf{w}) - f_{t+1}^* \right),$$

- *Streaming variance:*

$$\mathbb{E}\left[\frac{1}{2} \left\| \nabla \phi(\mathbf{w}_{t+1}^*) \right\|^2_{\left(\nabla^2 f(\mathbf{w}_{t+1}^*)\right)^{-1}} \right] \leq \sigma^2.$$

Here $\mu := \frac{\gamma}{\beta_1} \geq C \Delta \gamma$ for some $C > 0$, and

$$S = \mathcal{O}\left(\frac{d\beta_1}{\gamma} \right), \quad \sigma^2 = \mathcal{O}\left(d\beta_1^3 \|\mathbf{r}_t\|^2 \right)$$

for sub-Gaussian/regular polynomial-tail classes, and

$$S = \mathcal{O}\left(\frac{\beta_1}{\gamma} \right), \quad \sigma^2 = \mathcal{O}\left(\frac{\beta_1^3 \|\mathbf{r}_t\|^2}{\gamma^2} \right)$$

for the bounded class.

The proof of Lemma 16 is somewhat technical for the sub-Gaussian/regular polynomial-tail classes, which repeatedly applies the concentration properties of these two classes. But this lemma is the key for the sample complexity of our streaming algorithm to match the lower bound in the case of sub-Gaussian inputs: since we always draw fresh samples in the streaming setting, the "condition number" S/μ for these two classes depend on d only linearly (as opposed to quadratically in approximate ERM). These quantities determine the number of samples to be used in each round τ of Algorithm 1: m_τ is on the order of the condition number, and with m_τ stochastic updates, one can reduce the suboptimality by a constant factor in each round; κ_τ has to eventually increase geometrically to make sure the variance is reduced at the same pace.

Based on these quantities, we can apply the structural result of Frostig et al. (2015) and give the sampling complexity for driving the final suboptimality to η_t times the initial suboptimality in (16).

Lemma 17 (Sample complexity of streaming SVRG for least squares) *Let $\eta_t \in (0,1)$. Applying streaming-SVRG in Algorithm 1 to $\min_{\mathbf{w}} \; f_{t+1}(\mathbf{w})$ with initialization $\mathbf{0}$, we have*

$$\mathbb{E}\left[f_{t+1}(\mathbf{w}^\tau) - f_{t+1}^*\right] \le \eta_t \left(\frac{\beta_1 \left\|\mathbf{r}_t\right\|^2}{2}\right)$$

for $\tau \ge \Gamma = \mathcal{O}\left(\log \frac{1}{\eta_t}\right)$. The sample complexity of the first Γ iterations is $\mathcal{O}\left(\frac{d}{\Delta^2 \eta_t} + \frac{d}{\Delta^2 \gamma^2} \log \frac{1}{\eta_t}\right)$ for the sub-Gaussian/regular polynomial-tail classes, and $\mathcal{O}\left(\frac{1}{\Delta^2 \gamma^2 \eta_t}\right)$ for the bounded class.

Based on the linear convergence of shift-and-invert, we need only solve $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ linear systems, and we can bound $\frac{1}{\eta}$ by a geometrically increasing series where the last term is $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ (so the sum of this truncated series is still $\mathcal{O}\left(\frac{1}{\epsilon}\right)$). This results in the following total sample complexity.

Theorem 18 (Total sample complexity of streaming shift-and-invert CCA) *Let $\epsilon \in (0,1)$. After solving $T = \mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ linear systems to sufficient accuracy, streaming shift-and-invert CCA algorithm outputs $(\mathbf{u}_T, \mathbf{v}_T)$ with $align\left((\mathbf{u}_T, \mathbf{v}_T); (\mathbf{u}^*, \mathbf{v}^*)\right) \ge 1 - \epsilon$. Our algorithm processes each sample in $\mathcal{O}(d)$ time, and has a total sample complexity of*

$$\mathcal{O}\left(\frac{d}{\epsilon \Delta^2} + \frac{d}{\Delta^2 \gamma^2} \log^2 \frac{1}{\epsilon}\right) \qquad \textit{for the sub-Gaussian/regular polynomial-tail classes,}$$

$$\mathcal{O}\left(\frac{1}{\epsilon \Delta^2 \gamma^2}\right) \qquad \textit{for the bounded class.}$$

Interestingly, the sample complexity of our streaming CCA algorithm (assuming the parameter λ) improves over that of ERM we showed in Theorem 9: it removes small $\log d$ factors for all classes, and most remarkably achieves polynomial improvement in ϵ for the regular polynomial-tail class. This is due to the fact that the sample complexity of streaming SVRG basically only uses the moments, and does not require concentration of the whole covariance in Lemma 3. As a result, it is not clear if our analysis of ERM is the tightest possible.

5.2. Lower bound for Gaussian inputs

Consider the following Gaussian distribution named *single canonical pair model* (Chen et al., 2013):

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{I} & \Delta \boldsymbol{\phi} \boldsymbol{\psi}^\top \\ \Delta \boldsymbol{\psi} \boldsymbol{\phi}^\top & \mathbf{I} \end{bmatrix}\right), \tag{17}$$

where $\|\boldsymbol{\phi}\| = \|\boldsymbol{\psi}\| = 1$. It is straightforward to check that $\mathbf{T} = \mathbf{E}_{xy} = \Delta \boldsymbol{\phi} \boldsymbol{\psi}^\top$ for such a distribution. Observe that $\mathbf{T}$ is of rank one and has a singular value gap Δ, and the single pair of canonical directions are $(\mathbf{u}^*, \mathbf{v}^*) = (\boldsymbol{\phi}, \boldsymbol{\psi})$. Denote this class of model by $\mathcal{F}(d_x, d_y, \Delta)$. We have the following minimax lower bound for CCA under this model, which is an application of the result of Gao et al. (2017) for sparse CCA (by using rank $r = 1$ and hard sparsity, i.e., $q = 0$ and sparsity level d in their Theorem 3.2).

Table 1: Summary of sample, time (measured in floating point operations), and memory complexities of different approaches, in terms of (d, Δ, ϵ), for stochastic CCA with Gaussian inputs. We give the dominant term in complexities as $\epsilon \to 0$. Note that the time complexity of exact ERM is dominated by forming the eigen-system, while the memory complexity of ERM is dominated by saving the dataset.

Method	Sample	Time	Memory
Exact ERM	$\tilde{\mathcal{O}}\left(\frac{d}{\epsilon\Delta^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{d^3}{\epsilon\Delta^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{d^2}{\epsilon\Delta^2}\right)$
Approximate ERM by shift-and-invert	$\tilde{\mathcal{O}}\left(\frac{d}{\epsilon\Delta^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{d^2}{\epsilon\Delta^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{d^2}{\epsilon\Delta^2}\right)$
Streaming shift-and-invert (assuming close init. for λ)	$\mathcal{O}\left(\frac{d}{\epsilon\Delta^2}\right)$	$\mathcal{O}\left(\frac{d^2}{\epsilon\Delta^2}\right)$	$\mathcal{O}(d)$

Lemma 19 (Lower bound for single canonical pair model) *Suppose the data is generated by the single canonical pair model. Let $(\mathbf{u}, \mathbf{v})$ be some estimate of the canonical directions $(\mathbf{u}^*, \mathbf{v}^*)$ based on N samples. Then, there is a universal constant C, so that for N sufficiently large, we have:*

$$\inf_{\mathbf{u},\mathbf{v}} \sup_{\substack{\mathbf{u}^*, \mathbf{v}^* \in \\ \mathcal{F}(d_x, d_y, \Delta)}} \mathbb{E}\left[1 - align\left((\mathbf{u}_T, \mathbf{v}_T); (\mathbf{u}^*, \mathbf{v}^*)\right)\right] \ge C \frac{d}{\Delta^2 N}.$$

This lemma implies that, to estimate the canonical directions up to ϵ-suboptimality in our measure of alignment, we expect to use at least $\mathcal{O}\left(\frac{d}{\epsilon\Delta^2}\right)$ samples. We therefore observe that, for Gaussian inputs, the sample complexity of the our streaming algorithm matches that of the minimax rate of CCA, up to small factors.

In Table 1, we collect the complexities of different approaches, namely exact optimization of ERM (Section 3), stochastic optimization of ERM with shift-and-invert (Section 4), and streaming shift-and-invert (Section 5). We observe that while all three approaches are sample efficient (up to small factors), stochastic and streaming algorithms are more efficient in time and memory.

6. Conclusion

In this paper, we have studied the sample complexity of population CCA for several classes of input distributions, and proposed sample-efficient algorithms for learning the first pair of canonical directions. While the original problem is nonconvex, we exploit its structure as an eigenvalue problem, and analyze the statistical performance of the shift-and-invert power iterations.

Based on the deflation/peeling scheme (Allen-Zhu and Li, 2016, 2017) for eigenvalue problems, our results shall be extended to extracting the top-k canonical direction pairs. Our algorithms also apply to related eigenvalue problems in machine learning, such as partial least squares (Chen et al., 2017) and linear discriminant analysis (Bach and Jordan, 2005), which are special versions of CCA with the population covariances being identity

(i.e., $\mathbf{E}_{xx} = \mathbf{E}_{yy} = \mathbf{I}$) and $\mathbf{y}$ being one-hot representations for class labels respectively. It is an interesting question if our general approach can be adapted to study the statistical performance of the kernel extension of CCA (Fukumizu et al., 2007).

Acknowledgement

Research partially supported by NSF BIGDATA award 1546462.

References

Zeyuan Allen-Zhu and Yuanzhi Li. LazySVD: Even faster SVD decomposition yet without agonizing pain. In *Advances in Neural Information Processing Systems*, 2016.

Zeyuan Allen-Zhu and Yuanzhi Li. Doubly accelerated methods for faster CCA and generalized eigendecomposition. In *Proc. of the International Conference on Machine Learning*, 2017.

Raman Arora, Andy Cotter, Karen Livescu, and Nati Srebro. Stochastic optimization for PCA and PLS. In *50th Annual Allerton Conference on Communication, Control, and Computing*, 2012.

Raman Arora, Teodor V. Marinov, Poorya Mianjy, and Nathan Srebro. Stochastic approximation for canonical correlation analysis. In *Advances in Neural Information Processing Systems*, 2017.

Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM*, (2), 2009.

Francis R. Bach and Michael I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, April 21 2005.

Marcus Carlsson. Perturbation theory for the matrix square root and matrix modulus. arXiv:1810.01464 [math.FA], October 2 2018.

Mengjie Chen, Chao Gao, Zhao Ren, and Harrison H. Zhou. Sparse CCA via precision adjusted iterative thresholding. arXiv:1311.6186 [math.ST], November 24 2013.

Zhehui Chen, Lin F. Yang, Chris J. Li, and Tuo Zhao. Dropping convexity for more efficient and scalable online multiview learning. In *Proc. of the International Conference on Machine Learning*, 2017.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation III. *SIAM Journal of Numerical Analysis*, 7(1):1–46, 1970.

Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on Learning Theory (COLT)*, pages 728–763, 2015.

Kenji Fukumizu, Francis R. Bach, and Arthur Gretton. Statistical consistency of kernel canonical correlation analysis. *Journal of Machine Learning Research*, 8:361–383, February 2007.

Chao Gao, Zongming Ma, and Harrison H. Zhou. Sparse CCA: Adaptive estimation and computational barriers. *Annals of Statistics*, 45(5):2074–2101, 2017.

Dan Garber and Elad Hazan. Fast and simple PCA via convex optimization. arXiv:1509.05647 [math.OC], November25 2015.

Dan Garber, Elad Hazan, Chi Jin, Sham M. Kakade, Cameron Musco, Praneeth Netrapalli, and Aaron Sidford. Faster eigenvector computation via shift-and-invert preconditioning. In *Proc. of the International Conference on Machine Learning*, 2016.

Rong Ge, Chi Jin, Sham M. Kakade, Praneeth Netrapalli, and Aaron Sidford. Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. In *Proc. of the International Conference on Machine Learning*, 2016.

Eckart Gekeler. On the pointwise matrix product and the mean value theorem. *Linear Algebra and its Applications*, 35:183–191, 1981.

Gene H. Golub and Charles F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1986.

Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Daniel Hsu, Sham Kakade, and Tong Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electron. Commun. Probab.*, 17(52):1–6, 2012.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, 2015.

Yichao Lu and Dean P. Foster. Large scale canonical correlation analysis with iterative least squares. In *Advances in Neural Information Processing Systems*, 2014.

Zhuang Ma, Yichao Lu, and Dean Foster. Finding linear structure in large datasets with scalable canonical correlation analysis. In *Proc. of the International Conference on Machine Learning*, 2015.

Roy Mathias. A bound for the matrix square root with application to eigenvector perturbation. *SIAM J. Matrix Anal. and Apps.*, 18(4):861–867, 1997.

Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, 1992.

Nikhil Srivastava and Roman Vershynin. Covariance estimation for distributions with $2+\varepsilon$-moments. *Annals of Probability*, 41(5):3081–3111, 2013.

Roman Vershynin. *Compressed Sensing: Theory and Applications*, chapter Introduction to the Non-asymptotic Analysis of Random Matrices. Cambridge University Press, 2012.

Weiran Wang and Karen Livescu. Large-scale approximate kernel canonical correlation analysis. In *Proc. of the International Conference on Learning Representations*, 2016.

Weiran Wang, Jialei Wang, Dan Garber, and Nathan Srebro. Globally convergent stochastic optimization for canonical correlation analysis. In *Advances in Neural Information Processing Systems*, 2016.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Appendix A. Auxiliary Lemmas

Lemma 20 *The population canonical correlation is bounded by* 1*, i.e.,*

$$\rho_1 = \sigma_1(\mathbf{T}) \le 1.$$

Proof By the Cauchy-Schwarz inequality of random variables, we have

$$\rho_1 = \mathbb{E}[(\mathbf{u}^{*\top}\mathbf{x})(\mathbf{v}^{*\top}\mathbf{y})] \le \sqrt{\mathbb{E}[(\mathbf{u}^{*\top}\mathbf{x})^2]} \cdot \sqrt{\mathbb{E}[(\mathbf{v}^{*\top}\mathbf{y})^2]} = \sqrt{\mathbf{u}^{*\top}\mathbf{E}_{xx}\mathbf{u}} \cdot \sqrt{\mathbf{v}^{*\top}\mathbf{E}_{yy}\mathbf{v}} = 1.$$

■

Lemma 21 (Distance between normalized vectors) *For two nonzero vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$*, we have*

$$\left\| \frac{\mathbf{a}}{\|\mathbf{a}\|} - \frac{\mathbf{b}}{\|\mathbf{b}\|} \right\| \le 2\frac{\|\mathbf{a}-\mathbf{b}\|}{\|\mathbf{a}\|}.$$

Proof By direct calculation, we have

$$\begin{aligned}
\left\| \frac{\mathbf{a}}{\|\mathbf{a}\|} - \frac{\mathbf{b}}{\|\mathbf{b}\|} \right\| &\le \left\| \frac{\mathbf{a}}{\|\mathbf{a}\|} - \frac{\mathbf{b}}{\|\mathbf{a}\|} \right\| + \left\| \frac{\mathbf{b}}{\|\mathbf{a}\|} - \frac{\mathbf{b}}{\|\mathbf{b}\|} \right\| \\
&= \frac{\|\mathbf{a}-\mathbf{b}\|}{\|\mathbf{a}\|} + \|\mathbf{b}\| \cdot \frac{|\|\mathbf{a}\| - \|\mathbf{b}\||}{\|\mathbf{a}\|\,\|\mathbf{b}\|} \\
&\le \frac{\|\mathbf{a}-\mathbf{b}\|}{\|\mathbf{a}\|} + \frac{\|\mathbf{a}-\mathbf{b}\|}{\|\mathbf{a}\|} \\
&= 2\frac{\|\mathbf{a}-\mathbf{b}\|}{\|\mathbf{a}\|}
\end{aligned}$$

where we have used the triangle inequality in the two inequalities. ■

Lemma 22 (Conversion from joint alignment to separate alignment) *Let* $\eta \in \left(0, \frac{1}{4}\right)$*. Consider the four nonzero vectors* $\mathbf{a}, \mathbf{x} \in \mathbb{R}^{d_x}$ *and* $\mathbf{b}, \mathbf{y} \in \mathbb{R}^{d_y}$ *such that* $\|\mathbf{a}\| = \|\mathbf{b}\| = 1$*. If*

$$\frac{1}{\sqrt{2}} \cdot \frac{\mathbf{a}^\top\mathbf{x} + \mathbf{b}^\top\mathbf{y}}{\sqrt{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2}} \ge 1 - \eta, \tag{18}$$

we also have

$$\frac{1}{2}\left(\left| \frac{\mathbf{a}^\top\mathbf{x}}{\|\mathbf{x}\|} \right| + \left| \frac{\mathbf{b}^\top\mathbf{y}}{\|\mathbf{y}\|} \right| \right) \ge 1 - 4\eta.$$

Proof By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\frac{\mathbf{a}^\top\mathbf{x} + \mathbf{b}^\top\mathbf{y}}{\sqrt{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2}} &= \frac{\mathbf{a}^\top\mathbf{x}}{\|\mathbf{x}\|} \cdot \frac{\|\mathbf{x}\|}{\sqrt{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2}} + \frac{\mathbf{b}^\top\mathbf{y}}{\|\mathbf{y}\|} \cdot \frac{\|\mathbf{y}\|}{\sqrt{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2}} \\
&\le \sqrt{\left(\frac{\mathbf{a}^\top\mathbf{x}}{\|\mathbf{x}\|}\right)^2 + \left(\frac{\mathbf{b}^\top\mathbf{y}}{\|\mathbf{y}\|}\right)^2}.
\end{aligned}$$

Thus according to (18), we obtain

$$\left(\frac{\mathbf{a}^\top\mathbf{x}}{\|\mathbf{x}\|}\right)^2 + \left(\frac{\mathbf{b}^\top\mathbf{y}}{\|\mathbf{y}\|}\right)^2 \ge 2(1-\eta)^2 \ge 2 - 4\eta.$$

Since $\left(\frac{\mathbf{b}^\top\mathbf{y}}{\|\mathbf{y}\|}\right)^2 \le 1$, this implies

$$\left| \frac{\mathbf{a}^\top\mathbf{x}}{\|\mathbf{x}\|} \right| \ge \sqrt{1-4\eta} \ge 1 - 4\eta$$

where the last step is due to the fact that $\sqrt{x} \ge x$ for $x \in (0,1)$. Similarly we have $\left|\frac{\mathbf{b}^\top\mathbf{y}}{\|\mathbf{y}\|}\right| \ge 1 - 4\eta$. Then the theorem follows. ■

Lemma 23 (Moment inequalities of sub-Gaussian and regular polynomial-tail random vectors) *Let* $\mathbf{z} \in \mathbb{R}^d$ *be isotropic and sub-Gaussian or regular polynomial-tail (see their definitions in Lemma 3). Then for some constant* $C' > 0$*, we have*

$$\mathbb{E}\|\mathbf{z}\|^2 \le d, \qquad \mathbb{E}\|\mathbf{z}\|^4 \le C'd^2, \qquad \mathbb{E}\left|\mathbf{q}^\top\mathbf{z}\right|^4 \le C'$$

where $\mathbf{q}$ *is any unit vector.*

Proof Sub-Gaussian case The first bound is by $\mathbb{E}\|\mathbf{z}\|^2 = \mathbb{E}\operatorname{tr}\left(\mathbf{z}\mathbf{z}^\top\right) = \operatorname{tr}(I) = d$. To prove the second one, note that according to Theorem 2.1 in Hsu et al. (2012), we have

$$\mathbb{P}\left(\|\mathbf{z}\|^2 > C_1(d+t)\right) < e^{-t}$$

for all $t > 0$. Therefore

$$\begin{aligned}
\mathbb{E}\|\mathbf{z}\|^4 &= \int_0^\infty \mathbb{P}\left(\|\mathbf{z}\|^4 > s\right) ds \\
&= \int_0^{C_1^2 d^2} \mathbb{P}\left(\|\mathbf{z}\|^4 > s\right) ds + \int_{C_1^2 d^2}^\infty \mathbb{P}\left(\|\mathbf{z}\|^4 > s\right) ds \\
&\le C_1^2 d^2 + \int_{C_1^2 d^2}^\infty \exp\left(-\left(\frac{\sqrt{s}}{C_1} - d\right)\right) ds \\
&\le C' d^2.
\end{aligned}$$

Lastly,

$$\begin{aligned}
\mathbb{E}\left|\mathbf{q}^\top \mathbf{z}\right|^4 &= \int_0^\infty \mathbb{P}\left(\left|\mathbf{q}^\top \mathbf{z}\right|^4 > s\right) ds \\
&\le \int_0^\infty e^{-C\sqrt{s}} ds \\
&\le C'.
\end{aligned}$$

Regular polynomial-tail case The first bound is still by $\mathbb{E}\|\mathbf{z}\|^2 = \mathbb{E}\operatorname{tr}\left(\mathbf{z}\mathbf{z}^\top\right) = \operatorname{tr}(I) = d$. When $r > 1$, we have

$$\begin{aligned}
\mathbb{E}\|\mathbf{z}\|^4 &= \int_0^\infty \mathbb{P}\left(\|\mathbf{z}\|^4 > s\right) ds \\
&\le \int_0^{C^2 d^2} \mathbb{P}\left(\|\mathbf{z}\|^4 > s\right) ds + \int_{C^2 d^2}^\infty \mathbb{P}\left(\|\mathbf{z}\|^4 > s\right) ds \\
&\le C^2 d^2 + \int_{C^2 d^2}^\infty C s^{-\frac{1+r}{2}} ds \\
&\le C' d^2.
\end{aligned}$$

To prove the last bound, take $\mathbf{V} = \mathbf{q}\mathbf{q}^\top$ in the definition of regular polynomial-tail random vectors, and then

$$\mathbb{P}\left(\left|\mathbf{q}^\top \mathbf{z}\right|^2 > t\right) \le C t^{-1-r},$$

for any $t > C$. We have

$$\begin{aligned}
\mathbb{E}\left|\mathbf{q}^\top \mathbf{z}\right|^4 &= \int_0^\infty \mathbb{P}\left(\left|\mathbf{q}^\top \mathbf{z}\right|^4 > s\right) ds \\
&\le \int_0^{C^2} \mathbb{P}\left(\left|\mathbf{q}^\top \mathbf{z}\right|^4 > s\right) ds + \int_{C^2}^\infty \mathbb{P}\left(\left|\mathbf{q}^\top \mathbf{z}\right|^4 > s\right) ds \\
&\le C^2 + \int_{C^2}^\infty C s^{-\frac{1+r}{2}} ds \\
&\le C'.
\end{aligned}$$

■

Appendix B. Proofs for Section 1

B.1. Proof of Lemma 1

Proof Using the fact that $\frac{\mathbf{u}^\top \mathbf{E}_{xx}\mathbf{u}^*}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|}$ and $\frac{\mathbf{v}^\top \mathbf{E}_{yy}\mathbf{v}^*}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|}$ are at most 1, the condition on alignment implies

$$\frac{\mathbf{u}^\top \mathbf{E}_{xx}\mathbf{u}^*}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|} = \mathbf{a}_1^\top \frac{\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|} \ge 1 - \frac{\eta}{4}, \qquad \frac{\mathbf{v}^\top \mathbf{E}_{yy}\mathbf{v}^*}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|} = \mathbf{b}_1^\top \frac{\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|} \ge 1 - \frac{\eta}{4}.$$

Since $\{\mathbf{a}_i\}_{i=1}^r$ and $\{\mathbf{b}_i\}_{i=1}^r$ are orthonormal, we have

$$\sum_{i=2}^r \left(\mathbf{a}_i^\top \frac{\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|}\right)^2 \le 1 - \left(1 - \frac{\eta}{4}\right)^2 \le \frac{\eta}{2},$$

$$\sum_{i=2}^r \left(\mathbf{b}_i^\top \frac{\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|}\right)^2 \le 1 - \left(1 - \frac{\eta}{4}\right)^2 \le \frac{\eta}{2}.$$

Observe that

$$\begin{aligned}
\frac{\mathbf{u}^\top \mathbf{E}_{xy}\mathbf{v}}{\sqrt{\mathbf{u}^\top \mathbf{E}_{xx}\mathbf{u}}\sqrt{\mathbf{v}^\top \mathbf{E}_{yy}\mathbf{v}}} &= \frac{(\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u})^\top \mathbf{T}(\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v})}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|} = \sum_{i=1}^d \rho_i \left(\mathbf{a}_i^\top \frac{\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|}\right)\left(\mathbf{b}_i^\top \frac{\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|}\right) \\
&\ge \rho_1 \left(\mathbf{a}_i^\top \frac{\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|}\right)\left(\mathbf{b}_1^\top \frac{\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|}\right) - \rho_2 \sqrt{\sum_{i=2}^r \left(\mathbf{a}_i^\top \frac{\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}\right\|}\right)^2}\sqrt{\sum_{i=2}^r \left(\mathbf{b}_i^\top \frac{\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}\right\|}\right)^2} \\
&\ge \rho_1 \left(1 - \frac{\eta}{4}\right)^2 - \rho_1 \cdot \frac{\eta}{2} \ge \rho_1 (1 - \eta)
\end{aligned}$$

where we have used the Cauchy-Schwarz inequality in the first inequality. ■

Appendix C. Proofs for Section 3

C.1. Proof of Lemma 3

Proof Sub-Gaussian/regular polynomial-tail cases Consider the random variable $\mathbf{z}$ defined in (5), and draw i.i.d. samples $\mathbf{z}_1, \ldots, \mathbf{z}_n$ of $\mathbf{z}$. It is known that when the sample

size n is large enough (as specified in the lemma), we have

$$\left\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^\top - \mathbf{I} \right\| \le \frac{\nu}{2}$$

with high probability for the sub-Gaussian class (Vershynin, 2012) and for the regular polynomial-tail class (Srivastava and Vershynin, 2013), given $N > C' \frac{d}{\nu^2}$ and $N \ge C' \frac{d}{\nu^{2(1+r^{-1})}}$ respectively.

We then turn to bounding the error in each covariance matrix. We note that the covariance of $\mathbf{f} := \begin{bmatrix} \mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{x} \\ \mathbf{E}_{yy}^{-\frac{1}{2}} \mathbf{y} \end{bmatrix}$ is $\mathbf{\Xi} = \begin{bmatrix} \mathbf{I} & \mathbf{T} \\ \mathbf{T}^\top & \mathbf{I} \end{bmatrix}$ with $\|\mathbf{\Xi}\| = 1 + \rho_1 \le 2$ (since the eigenvalues of $\mathbf{\Xi}$ are of the form $1 \pm \sigma_i(\mathbf{T})$). On the other hand, define

$$\mathbf{\Phi} := \begin{bmatrix} \mathbf{E}_{xx} & \\ & \mathbf{E}_{yy} \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} \mathbf{E}_{xx} & \mathbf{E}_{xy} \\ \mathbf{E}_{xy}^\top & \mathbf{E}_{yy} \end{bmatrix}^{\frac{1}{2}} \qquad \text{satisfying} \quad \mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{\Xi}$$

and we have $\mathbf{f} = \mathbf{\Phi}\mathbf{z}$ by the definition of $\mathbf{z}$. Furthermore, $\mathbf{f}_i = \mathbf{\Phi}\mathbf{z}_i$, $i = 1, \dots, N$ are i.i.d. samples of $\mathbf{f}$. Therefore, it holds that

$$\left\| \begin{bmatrix} \frac{1}{N}\sum_{i=1}^N \mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{x}_i \mathbf{x}_i^\top \mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{I} & \frac{1}{N}\sum_{i=1}^N \mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{x}_i \mathbf{y}_i^\top \mathbf{E}_{yy}^{-\frac{1}{2}} - \mathbf{T} \\ \frac{1}{N}\sum_{i=1}^N \mathbf{E}_{yy}^{-\frac{1}{2}} \mathbf{y}_i \mathbf{x}_i^\top \mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{T}^\top & \frac{1}{N}\sum_{i=1}^N \mathbf{E}_{yy}^{-\frac{1}{2}} \mathbf{y}_i \mathbf{y}_i^\top \mathbf{E}_{yy}^{-\frac{1}{2}} - \mathbf{I} \end{bmatrix} \right\| = \left\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}_i \mathbf{f}_i^\top - \mathbf{\Xi} \right\|$$

$$= \left\| \mathbf{\Phi} \left(\frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^\top - \mathbf{I} \right) \mathbf{\Phi}^\top \right\| \le \left\| \mathbf{\Phi}\mathbf{\Phi}^\top \right\| \cdot \left\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^\top - \mathbf{I} \right\| = \|\mathbf{\Xi}\| \cdot \left\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i \mathbf{z}_i^\top - \mathbf{I} \right\| \le \nu.$$

Since the norm of each block is bounded by the norm of the entire matrix, we conclude that the error in estimating each covariance matrix is bounded by ν, as required by Proposition 2.

Remark 24 *In view of Lemma 23 and the proof technique here, for the sub-Gaussian/regular polynomial-tail cases, the bound of $\|\mathbf{z}\|^2$ leads to a bound for $\|\mathbf{x}\|^2$ and $\|\mathbf{y}\|^2$:*

$$\mathbb{E}(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2) \le \|\mathbf{E}_{xx}\| \cdot \mathbb{E} \left\| \mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{x} \right\|^2 + \|\mathbf{E}_{yy}\| \cdot \mathbb{E} \left\| \mathbf{E}_{yy}^{-\frac{1}{2}} \mathbf{y} \right\|^2 \le \mathbb{E} \|\mathbf{f}\|^2 \le 2\mathbb{E} \|\mathbf{z}\|^2 \le Cd$$

for some constant $C > 0$, where we have used Assumption 1 in the second inequality. And similarly, we have

$$\mathbb{E}(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2)^2 \le \mathbb{E} \|\mathbf{f}\|^4 \le 4\mathbb{E} \|\mathbf{z}\|^4 \le C' d^2$$

for some constant $C' > 0$.

Bounded case Consider the joint covariance matrix

$$\begin{bmatrix} \mathbf{E}_{xx} & \mathbf{E}_{xy} \\ \mathbf{E}_{xy}^\top & \mathbf{E}_{yy} \end{bmatrix} \in \mathbb{R}^{d \times d}$$

which has eigenvalue bounded by 2 due to the assumption that $\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 \le 2$. Applying Vershynin (2012, Corollary 5.52), we obtain that

$$\left\| \begin{bmatrix} \mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{xy} \\ \mathbf{\Sigma}_{xy}^\top & \mathbf{\Sigma}_{yy} \end{bmatrix} - \begin{bmatrix} \mathbf{E}_{xx} & \mathbf{E}_{xy} \\ \mathbf{E}_{xy}^\top & \mathbf{E}_{yy} \end{bmatrix} \right\| \le \nu' \tag{19}$$

with probability at least $1 - d^{-t^2}$ when $N \ge C(t/\nu')^2 \log d$ for some constant $C > 0$. Setting the failure probability $\delta = d^{-t^2}$ gives $t^2 = \frac{\log \frac{1}{\delta}}{\log d}$, and thus we require $N \ge C \frac{1}{\nu'^2} \log \frac{1}{\delta}$ for $1 - \delta$ success probability.

Due to the block structure of the joint covariance matrix, (19) implies

$$\|\mathbf{\Sigma}_{xy} - \mathbf{E}_{xy}\| \le \nu', \qquad \|\mathbf{\Sigma}_{xx} - \mathbf{E}_{xx}\| \le \nu', \qquad \|\mathbf{\Sigma}_{yy} - \mathbf{E}_{yy}\| \le \nu'$$

hold simultaneously.

Now, to satisfy the first inequality of (9), observe that

$$\begin{aligned} \left\| \mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xx} \mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{I} \right\| &= \left\| \mathbf{E}_{xx}^{-\frac{1}{2}} (\mathbf{\Sigma}_{xx} - \mathbf{E}_{xx}) \mathbf{E}_{xx}^{-\frac{1}{2}} \right\| \\ &\le \left\| \mathbf{E}_{xx}^{-\frac{1}{2}} \right\| \cdot \|\mathbf{\Sigma}_{xx} - \mathbf{E}_{xx}\| \cdot \left\| \mathbf{E}_{xx}^{-\frac{1}{2}} \right\| \\ &\le \|\mathbf{\Sigma}_{xx} - \mathbf{E}_{xx}\| / \gamma \end{aligned}$$

where we have used the assumption that $\sigma_{\min}(\mathbf{E}_{xx}) \ge \gamma$ in the last inequality. Therefore, we obtain $\left\| \mathbf{E}_{xx}^{-\frac{1}{2}} \mathbf{\Sigma}_{xx} \mathbf{E}_{xx}^{-\frac{1}{2}} - \mathbf{I} \right\| \le \nu$ by setting $\nu' = \gamma\nu$ in (19), and this yields the $N_0(\nu)$ chosen in the lemma. The other two inequalities of (9) can be obtained analogously. ■

C.2. Proof of Lemma 5

Proof This result can be derived from the main result of Mathias (1997) with a bit of detective work, which is needed to understand the higher order error term. As in Mathias (1997), assume without loss of generality that $\mathbf{H} = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$ is diagonal. In the proof of Theorem 2, Mathias (1997) applied the Daleckii-Krein formula in his equation (4) with the $\mathcal{O}$-notation, which can be rephrased as (see also Carlsson (2018)[Theorem 2.1]): for ζ in a neighborhood of 0, it holds

$$\varepsilon := \left\| (\mathbf{H} + \zeta\mathbf{\Theta})^{\frac{1}{2}} - \mathbf{H}^{\frac{1}{2}} - \zeta \left[\frac{\lambda_i^{\frac{1}{2}} \lambda_j^{\frac{1}{2}}}{\lambda_i^{\frac{1}{2}} + \lambda_j^{\frac{1}{2}}} \right]_{i,j=1}^{d} \circ (\mathbf{H}^{-\frac{1}{2}} \mathbf{\Theta} \mathbf{H}^{-\frac{1}{2}}) \right\| = \mathcal{O}(\zeta^2)$$

where $\circ$ denotes elementwise (Hadamard) multiplication.

To locate the neighborhood of ζ for which the above is true, we apply the matrix mean value theorem (Gekeler, 1981) with the second order derivative of matrix square root (Horn

and Johnson, 1991, Theorem 6.6.30 and page 549) to obtain:

$$\varepsilon \leq \frac{\zeta^2}{2} \max_{\tau\in[0,\zeta]} \left\| \mathbf{U}(\tau) \left(\sum_{k=1}^{d} \left[\frac{1}{\sqrt{\lambda_i(\tau)}+\sqrt{\lambda_j(\tau)}} \cdot \frac{1}{\sqrt{\lambda_i(\tau)}+\sqrt{\lambda_k(\tau)}} \cdot \frac{1}{\sqrt{\lambda_j(\tau)}+\sqrt{\lambda_k(\tau)}} \right]_{i,j=1}^{d} \circ \left[\mathbf{c}_k(\tau)\mathbf{c}_k^\top(\tau) \right] \right) \mathbf{U}(\tau)^\top \right\| \tag{20}$$

where $\mathbf{H}+\tau\mathbf{\Theta} = \mathbf{U}(\tau)\cdot \operatorname{diag}(\lambda_1(\tau),\ldots,\lambda_d(\tau))\cdot \mathbf{U}(\tau)^\top$ is the eigenvalue decomposition of the perturbation of $\mathbf{H}$, and $\mathbf{c}_k(\tau)$ is the k-th column of $\mathbf{X}(\tau) = \mathbf{U}(\tau)^\top\mathbf{\Theta}\mathbf{U}(\tau)$.

Define $\mathbf{Z}(\tau) = \left[\frac{1}{\sqrt{\lambda_i(\tau)}+\sqrt{\lambda_j(\tau)}}\right]_{i,j=1}^{d}$. The summation enclosed in () of the right hand side of (20) can be written as $\mathbf{Z}(\tau)\circ(\mathbf{Z}(\tau)\circ\mathbf{X}(\tau))^2$. Thus continuing from (20) yields

$$\varepsilon \leq \frac{\zeta^2}{2}\cdot \max_{\tau\in[0,\zeta]} \left\| \mathbf{Z}(\tau)\circ(\mathbf{Z}(\tau)\circ\mathbf{X}(\tau))^2 \right\|.$$

On the one hand, by the assumption that $\|\mathbf{H}\|\leq\sigma_{\max}$, we have

$$\|\mathbf{X}(\tau)\| = \|\mathbf{\Theta}\| = \left\| \mathbf{H}^{\frac12}(\mathbf{H}^{-\frac12}\mathbf{\Theta}\mathbf{H}^{-\frac12})\mathbf{H}^{\frac12} \right\| \leq \|\mathbf{H}\|\cdot\left\|\mathbf{H}^{-\frac12}\mathbf{\Theta}\mathbf{H}^{-\frac12}\right\| \leq \sigma_{\max}. \tag{21}$$

On the other hand, the matrix $\mathbf{Z}(\tau)$ is positive semidefinite (see Horn and Johnson, 1991, Problem 9, page 348). Using the fact that $\|\mathbf{A}\circ\mathbf{B}\|\leq(\max_i \mathbf{A}_{ii})\cdot\|\mathbf{B}\|$ for positive semidefinite $\mathbf{A}$ and Hermitian $\mathbf{B}$ (Horn and Johnson, 1991, Theorem 5.5.18), we conclude

$$\varepsilon \leq \frac{\zeta^2\sigma_{\max}^2}{2}\left[\max_{\tau\in[0,\zeta]}\max_i \frac{1}{2\sqrt{\lambda_i(\tau)}}\right]^3.$$

Let $\zeta\leq\frac34\sigma_{\max}^{-1}\sigma_{\min}$. Then in view of (21) and the Weyl's inequality, $\lambda_i(\tau)\geq\frac{\sigma_{\min}}{4}$ for all $\tau\in[0,\zeta]$, and we have $\varepsilon\leq\frac12\zeta^2\sigma_{\max}^2\sigma_{\min}^{-\frac32}$.

To sum up, we have shown so far the following first order approximation: for certain error matrix $\mathcal{E}\in\mathbb{R}^{d\times d}$, it holds

$$(\mathbf{H}+\zeta\mathbf{\Theta})^{\frac12} = \mathbf{H}^{\frac12} + \zeta\left[\frac{\lambda_i^{\frac12}\lambda_j^{\frac12}}{\lambda_i^{\frac12}+\lambda_j^{\frac12}}\right]_{i,j=1}^{d}\circ(\mathbf{H}^{-\frac12}\mathbf{\Theta}\mathbf{H}^{-\frac12}) + \mathcal{E} \qquad \text{where} \qquad \|\mathcal{E}\|\leq\frac12\zeta^2\sigma_{\max}^2\sigma_{\min}^{-\frac32}.$$

Consequently, we have

$$(\mathbf{H}+\zeta\mathbf{\Theta})^{\frac12}\mathbf{H}^{-\frac12}-\mathbf{I} = \zeta\left[\frac{\lambda_i^{\frac12}}{\lambda_i^{\frac12}+\lambda_j^{\frac12}}\right]_{i,j=1}^{d}\circ(\mathbf{H}^{-\frac12}\mathbf{\Theta}\mathbf{H}^{-\frac12}) + \mathcal{E}\mathbf{H}^{-\frac12}.$$

Mathias (1997) showed that the norm of the first term on the right hand size is of the order $\mathcal{O}(\log d\cdot\zeta)$. Combining this with the fact that $\left\|\mathcal{E}\mathbf{H}^{-\frac12}\right\|\leq\frac12\zeta^2\sigma_{\max}^2\sigma_{\min}^{-2}$, we conclude that $\left\|\mathcal{E}\mathbf{H}^{-\frac12}\right\| = \mathcal{O}(\zeta)$ for $\zeta=\mathcal{O}(\sigma_{\max}^{-2}\sigma_{\min}^2)$ and the lemma follows. ■

C.3. Proof of Lemma 6

Proof In view of the Weyl's inequality, we have

$$|\widehat{\rho}_1-\rho_1| \leq \left\|\widehat{\mathbf{T}}-\mathbf{T}\right\| = \left\|\mathbf{\Sigma}_{xx}^{-\frac12}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac12} - \mathbf{E}_{xx}^{-\frac12}\mathbf{E}_{xy}\mathbf{E}_{yy}^{-\frac12}\right\|. \tag{22}$$

For the right hand side of (22), we have the following decomposition

$$\begin{aligned}&\mathbf{\Sigma}_{xx}^{-\frac12}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac12} - \mathbf{E}_{xx}^{-\frac12}\mathbf{E}_{xy}\mathbf{E}_{yy}^{-\frac12}\\ &= \left(\mathbf{\Sigma}_{xx}^{-\frac12}-\mathbf{E}_{xx}^{-\frac12}\right)\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac12} + \mathbf{E}_{xx}^{-\frac12}\left(\mathbf{\Sigma}_{xy}-\mathbf{E}_{xy}\right)\mathbf{\Sigma}_{yy}^{-\frac12} + \mathbf{E}_{xx}^{-\frac12}\mathbf{E}_{xy}\left(\mathbf{\Sigma}_{yy}^{-\frac12}-\mathbf{E}_{yy}^{-\frac12}\right).\end{aligned} \tag{23}$$

By the equality

$$\mathbf{A}^{-\frac12}-\mathbf{B}^{-\frac12} = \mathbf{B}^{-\frac12}\left(\mathbf{B}^{\frac12}-\mathbf{A}^{\frac12}\right)\mathbf{A}^{-\frac12},$$

the first term of the RHS of (23) becomes

$$\left(\mathbf{\Sigma}_{xx}^{-\frac12}-\mathbf{E}_{xx}^{-\frac12}\right)\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac12} = \mathbf{E}_{xx}^{-\frac12}\left(\mathbf{E}_{xx}^{\frac12}-\mathbf{\Sigma}_{xx}^{\frac12}\right)\mathbf{\Sigma}_{xx}^{-\frac12}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac12}.$$

When $\left\|\mathbf{E}_{xx}^{-\frac12}\mathbf{\Sigma}_{xx}\mathbf{E}_{xx}^{-\frac12}-\mathbf{I}\right\|\leq\nu$, according to Lemma 5, we have (by making the identification that $\mathbf{H}=\mathbf{E}_{xx}$, $\zeta=\left\|\mathbf{E}_{xx}^{-\frac12}(\mathbf{\Sigma}_{xx}-\mathbf{E}_{xx})\mathbf{E}_{xx}^{-\frac12}\right\|$, and $\mathbf{\Theta}=(\mathbf{\Sigma}_{xx}-\mathbf{E}_{xx})/\left\|\mathbf{E}_{xx}^{-\frac12}(\mathbf{\Sigma}_{xx}-\mathbf{E}_{xx})\mathbf{E}_{xx}^{-\frac12}\right\|$)

$$\left\|\mathbf{E}_{xx}^{-\frac12}\left(\mathbf{E}_{xx}^{\frac12}-\mathbf{\Sigma}_{xx}^{\frac12}\right)\right\|\leq C_d\cdot\nu$$

for $\nu\leq\frac34\gamma^2$. Combining with the fact that $\left\|\mathbf{\Sigma}_{xx}^{-\frac12}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac12}\right\|\leq 1$, we have

$$\left\|\left(\mathbf{\Sigma}_{xx}^{-\frac12}-\mathbf{E}_{xx}^{-\frac12}\right)\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac12}\right\|\leq C_d\cdot\nu.$$

A similar bound can be obtained for the third term of (23). Observe that when $\left\|\mathbf{E}_{yy}^{-\frac12}\mathbf{\Sigma}_{yy}\mathbf{E}_{yy}^{-\frac12}-\mathbf{I}\right\|\leq\nu<1$, we have $\|\mathbf{\Sigma}_{yy}-\mathbf{E}_{yy}\|\leq\nu$ and all eigenvalues of $\mathbf{\Sigma}_{yy}$ lie in $[\gamma-\nu,1+\nu]$. Additionally, $\mathbf{E}_{yy}^{-\frac12}\mathbf{\Sigma}_{yy}\mathbf{E}_{yy}^{-\frac12}$ is invertible, and all eigenvalues of $\mathbf{\Sigma}_{yy}^{-\frac12}\mathbf{E}_{yy}\mathbf{\Sigma}_{yy}^{-\frac12}$ lie in $[\frac{1}{1+\nu},\frac{1}{1-\nu}]$, impling that $\left\|\mathbf{\Sigma}_{yy}^{-\frac12}\mathbf{E}_{yy}\mathbf{\Sigma}_{yy}^{-\frac12}-\mathbf{I}\right\|\leq\frac{\nu}{1-\nu}$. According to Lemma 5, we have (by making the identification that $\mathbf{H}=\mathbf{\Sigma}_{yy}$, $\zeta=\left\|\mathbf{\Sigma}_{yy}^{-\frac12}(\mathbf{\Sigma}_{yy}-\mathbf{E}_{yy})\mathbf{\Sigma}_{yy}^{-\frac12}\right\|$, and $\mathbf{\Theta}=(\mathbf{\Sigma}_{yy}-\mathbf{E}_{yy})/\left\|\mathbf{\Sigma}_{yy}^{-\frac12}(\mathbf{\Sigma}_{yy}-\mathbf{E}_{yy})\mathbf{\Sigma}_{yy}^{-\frac12}\right\|$)

$$\left\|\left(\mathbf{E}_{yy}^{\frac12}-\mathbf{\Sigma}_{yy}^{\frac12}\right)\mathbf{\Sigma}_{yy}^{-\frac12}\right\|\leq\frac{C_d\cdot\nu}{1-\nu} \tag{24}$$

for $\nu \le \frac{3}{4}\left(\frac{\gamma-\nu}{1+\nu}\right)^2$, which is satisfied for $\nu \le \frac{1}{4}\gamma^2$. Therefore, we can bound the third term of (23) as

$$\begin{aligned}\left\|\mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{E}_{xy}\left(\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}-\mathbf{E}_{yy}^{-\frac{1}{2}}\right)\right\| &= \left\|\mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{E}_{xy}\mathbf{E}_{yy}^{-\frac{1}{2}}\left(\mathbf{E}_{yy}^{\frac{1}{2}}-\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\right)\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}\right\| \\ &\le \left\|\mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{E}_{xy}\mathbf{E}_{yy}^{-\frac{1}{2}}\right\|\cdot\left\|\left(\mathbf{E}_{yy}^{\frac{1}{2}}-\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\right)\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}\right\| \\ &\le \frac{C_d\cdot\nu}{1-\nu} \le 2C_d\cdot\nu\end{aligned}$$

where we have used the fact that $\left\|\mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{E}_{xy}\mathbf{E}_{yy}^{-\frac{1}{2}}\right\| \le 1$ by Lemma 20.

For the second term of (23), we have by assumption that

$$\left\|\mathbf{E}_{xx}^{-\frac{1}{2}}\left(\mathbf{\Sigma}_{xy}-\mathbf{E}_{xy}\right)\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}\right\| \le \nu.$$

Applying the triangle inequality, we obtain from (23) that

$$\left\|\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-\frac{1}{2}}-\mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{E}_{xy}\mathbf{E}_{yy}^{-\frac{1}{2}}\right\| \le 4C_d\cdot\nu. \tag{25}$$

To sum up, it suffices to set $\nu = \frac{\epsilon'}{4C_d}$ to ensure $\left\|\widehat{\mathbf{T}}-\mathbf{T}\right\| \le \epsilon'$, and this yields the desired sample complexity. ■

C.4. Proof of Theorem 9

Proof Apply Lemma 6 with $\epsilon' = \frac{\sqrt{\epsilon}\Delta}{4}$ where $\epsilon \in (0,1)$ is the desired accuracy in Theorem 9. Since $\left\|\mathbf{T}-\hat{\mathbf{T}}\right\| \le \epsilon' < \frac{\Delta}{4}$, the eigenvalues of $\hat{\mathbf{T}}$ are within $\frac{\Delta}{4}$ of those of $\mathbf{T}$ due to Weyl's inequality, so there exists a positive singular gap of $\frac{\Delta}{2}$ for the empirical estimate $\hat{\mathbf{T}}$, whose first pair of singular vectors is unique. In view of the off-diagonal structure of $\widehat{\mathbf{C}}$, we observe that $\left\|\mathbf{C}-\hat{\mathbf{C}}\right\| = \left\|\mathbf{T}-\hat{\mathbf{T}}\right\| \le \epsilon'$ and that the top eigenvector of $\hat{\mathbf{C}}$ is unique. Then with the number of samples given in the theorem ensuring the ϵ' perturbation, according to the Davis-Kahan $\sin\theta$ theorem (Davis and Kahan, 1970), the top eigenvectors of $\mathbf{C}$ and $\widehat{\mathbf{C}}$ are well aligned:

$$\sin^2\theta \le \frac{\left\|\mathbf{C}-\widehat{\mathbf{C}}\right\|^2}{\Delta^2} \le \frac{\epsilon'^2}{\Delta^2} = \frac{\epsilon}{16} \tag{26}$$

where θ is the angle between the top eigenvector of $\mathbf{C}$ and that of $\widehat{\mathbf{C}}$. This is equivalent to

$$\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|^2 + \left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^*-\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}\right\|^2 \le \frac{\epsilon}{8} \tag{27}$$

and so $\max\left(\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|^2, \left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^*-\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}\right\|^2\right) \le \frac{\epsilon}{8}$.

In the rest of the proof, we fix the issue of incorrect normalization of $(\widehat{\mathbf{u}},\widehat{\mathbf{v}})$. Recall we have shown in the proof of Lemma 6 that (see e.g., (24))

$$\left\|\mathbf{I}-\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\right\| \le \epsilon' \le \frac{\sqrt{\epsilon}}{4}.$$

Consequently, we have

$$\begin{aligned}\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|^2 &= \left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\left(\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\right)\left(\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right)\right\|^2 \\ &\le \left(\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\| + \left\|\left(\mathbf{I}-\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\right)\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|\right)^2 \\ &\le 2\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|^2 + 2\left\|\mathbf{I}-\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{\Sigma}_{xx}^{-\frac{1}{2}}\right\|^2 \\ &\le \frac{\epsilon}{4}+\frac{\epsilon}{8} \\ &\le \frac{\epsilon}{2}\end{aligned}$$

where we have used the facts that $(x+y)^2 \le 2x^2+2y^2$ and $\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\| = 1$ in the second inequality.

According to Lemma 21, we then have

$$\left\|\frac{\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*\right\|}-\frac{\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|}\right\|^2 \le \frac{4\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|^2}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*\right\|^2} = 4\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*-\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|^2 \le 2\epsilon$$

and thus the alignment between these two vectors is

$$\frac{\widehat{\mathbf{u}}^\top\mathbf{E}_{xx}\mathbf{u}^*}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|} = 1-\frac{1}{2}\left\|\frac{\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*\right\|}-\frac{\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}}{\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}\right\|}\right\|^2 \ge 1-\epsilon.$$

A similar bound is obtained for $\widehat{\mathbf{v}}$:

$$\frac{\widehat{\mathbf{v}}^\top\mathbf{E}_{yy}\mathbf{v}^*}{\left\|\mathbf{E}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}\right\|} \ge 1-\epsilon.$$

Averaging the above two inequalities yields the desired result. Requiring that $\epsilon' = \frac{\sqrt{\epsilon}\Delta}{4} \le C_d\gamma^2$ as in Corollary 7 leads to the extra condition that $\epsilon \le \frac{16C_d^2\gamma^4}{\Delta^2}$. ■

Appendix D. Proofs for Section 4

D.1. Proof of Lemma 10

Proof If we obtain an approximate solution $\mathbf{w}_{t+1}$ to (13), such that $f_{t+1}(\mathbf{w}_{t+1}) - f_{t+1}(\mathbf{w}^*_{t+1}) = \epsilon_t(\mathbf{w}_t^\top \widehat{\mathbf{B}}\mathbf{w}_t)$, it holds that

$$\begin{aligned}
\epsilon_t \left\|\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{w}_t\right\|^2 &= \frac{1}{2}\left(\mathbf{w}_{t+1} - \mathbf{w}^*_{t+1}\right)^\top \widehat{\mathbf{A}}_\lambda \left(\mathbf{w}_{t+1} - \mathbf{w}^*_{t+1}\right) \\
&= \frac{1}{2}\left(\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{w}_{t+1} - \widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{w}^*_{t+1}\right)^\top \widehat{\mathbf{B}}^{-\frac{1}{2}}\widehat{\mathbf{A}}_\lambda\widehat{\mathbf{B}}^{-\frac{1}{2}}\left(\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{w}_{t+1} - \widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{w}^*_{t+1}\right) \\
&= \frac{1}{2}\left(\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right)^\top \widehat{\mathbf{M}}_\lambda^{-1}\left(\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right) = \frac{1}{2}\left\|\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right\|^2_{\widehat{\mathbf{M}}_\lambda^{-1}},
\end{aligned}$$

or equivalently

$$\left\|\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}} = \sqrt{2\epsilon_t}\cdot\left\|\mathbf{r}_t\right\|.$$

Note that our choice of ϵ_t is also invariant to the length of $\mathbf{r}_t$ (or whether normalization is performed).

For the exact solution to the linear system, we have

$$\mathbf{r}^*_{t+1} = \widehat{\mathbf{M}}_\lambda\mathbf{r}_t = \|\mathbf{r}_t\|\sum_{i=1}^d \beta_i\xi_{ti}\mathbf{p}_i.$$

As a result, we can bound the numerator and denominator of $G(\mathbf{r}_{t+1})$ respectively:

$$\begin{aligned}
\left\|\mathbf{P}_\perp\frac{\mathbf{r}_{t+1}}{\|\mathbf{r}_{t+1}\|}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}} &\le \frac{1}{\|\mathbf{r}_{t+1}\|}\left(\left\|\mathbf{P}_\perp\mathbf{r}^*_{t+1}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}} + \left\|\mathbf{P}_\perp\left(\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right)\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}}\right) \\
&\le \frac{1}{\|\mathbf{r}_{t+1}\|}\left(\left\|\mathbf{P}_\perp\mathbf{r}^*_{t+1}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}} + \left\|\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}}\right) \\
&= \frac{\|\mathbf{r}_t\|}{\|\mathbf{r}_{t+1}\|}\left(\sqrt{\sum_{i=2}^d \beta_i\xi_{ti}^2} + \sqrt{2\epsilon_t}\right),
\end{aligned}$$

and

$$\begin{aligned}
\left\|\mathbf{P}_\parallel\frac{\mathbf{r}_{t+1}}{\|\mathbf{r}_{t+1}\|}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}} &\ge \frac{1}{\|\mathbf{r}_{t+1}\|}\left(\left\|\mathbf{P}_\parallel\mathbf{r}^*_{t+1}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}} - \left\|\mathbf{P}_\parallel\left(\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right)\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}}\right) \\
&\ge \frac{1}{\|\mathbf{r}_{t+1}\|}\left(\left\|\mathbf{P}_\parallel\mathbf{r}^*_{t+1}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}} - \left\|\mathbf{r}_{t+1} - \mathbf{r}^*_{t+1}\right\|_{\widehat{\mathbf{M}}_\lambda^{-1}}\right) \\
&= \frac{\|\mathbf{r}_t\|}{\|\mathbf{r}_{t+1}\|}\left(\sqrt{\beta_1\xi_{t1}^2} - \sqrt{2\epsilon_t}\right).
\end{aligned}$$

Consequently, we have

$$\begin{aligned}
G(\mathbf{r}_{t+1}) &\le \frac{\sqrt{\sum_{i=2}^d \beta_i\xi_{ti}^2} + \sqrt{2\epsilon_t}}{\sqrt{\beta_1\xi_{t1}^2} - \sqrt{2\epsilon_t}} \le \frac{\beta_2\sqrt{\sum_{i=2}^d \xi_{ti}^2/\beta_i} + \sqrt{2\epsilon_t}}{\beta_1\sqrt{\xi_{t1}^2/\beta_1} - \sqrt{2\epsilon_t}} \\
&= G(\mathbf{r}_t)\cdot\frac{\beta_2 + \frac{\sqrt{2\epsilon_t}}{\sqrt{\sum_{i=2}^d \xi_{ti}^2/\beta_i}}}{\beta_1 - \frac{\sqrt{2\epsilon_t}}{\sqrt{\xi_{t1}^2/\beta_1}}}.
\end{aligned}$$

As long as $\sqrt{2\epsilon_t} \le \min\left(\sqrt{\sum_{i=2}^d \xi_{ti}^2/\beta_i},\ \sqrt{\xi_{t1}^2/\beta_1}\right)\cdot\frac{\beta_1-\beta_2}{4}$, i.e.,

$$\epsilon_t \le \min\left(\sum_{i=2}^d \xi_{ti}^2/\beta_i,\ \xi_{t1}^2/\beta_1\right)\cdot\frac{(\beta_1-\beta_2)^2}{32},$$

we are guaranteed that

$$G(\mathbf{r}_{t+1}) \le G(\mathbf{r}_t)\cdot\frac{\beta_1 + 3\beta_2}{3\beta_1 + \beta_2}.$$

Substituting in $\beta_i = \frac{1}{\lambda - \widehat{\rho}_i}$ with $\lambda - \widehat{\rho}_1 \le \widehat{\Delta}$, we obtain that

$$\frac{\beta_1 + 3\beta_2}{3\beta_1 + \beta_2} \le \frac{5}{7} < 1.$$

This means that if (14) holds for each least squares problem, the sequence $\{G(\mathbf{r}_t)\}_{t=0,\ldots}$ decreases (at least) at a constant geometric rate of $\frac{5}{7}$. Therefore, the number of inexact matrix-vector multiplications T needed to achieve $|\sin\theta_T| \le \eta$ is $\log_{\frac{7}{5}}\left(\frac{G(\mathbf{r}_0)}{\eta}\right)$. ■

D.2. Bounding the initial error for each least squares

We can minimize the initial suboptimality for the least squares problem f_{t+1} for reducing the time complexity of its solver. It is natural to use an initialization of the form $\alpha\mathbf{w}_t$, a scaled version of the previous iterate, which gives the following objective

$$f_{t+1}(\alpha\mathbf{w}_t) = \frac{(\mathbf{w}_t^\top\widehat{\mathbf{A}}_\lambda\mathbf{w}_t)}{2}\alpha^2 - (\mathbf{w}_t^\top\widehat{\mathbf{B}}\mathbf{w}_t)\alpha.$$

This is a quadratic function of α, and minimizing $f_{t+1}(\alpha\mathbf{w}_t)$ over α gives the optimal scaling $\alpha_t^* = \frac{\mathbf{w}_t^\top\widehat{\mathbf{B}}\mathbf{w}_t}{\mathbf{w}_t^\top\widehat{\mathbf{A}}_\lambda\mathbf{w}_t}$ (and this quantity is also invariant to the length of $\mathbf{w}_t$). Observe that α_t^* naturally measures the quality of $\mathbf{w}_t$: As $\mathbf{w}_t$ converges to $\widehat{\mathbf{w}}$, α_t^* converges to β_1. This initialization technique plays an important role in showing the linear convergence of our algorithm, and was used by Ge et al. (2016) for their standard power iterations (alternating least squares) scheme for CCA.

Proof [Proof of Lemma 11] With the given initialization, we have

$$\begin{aligned} f_{t+1}(\alpha_t^* \mathbf{w}_t) - f_{t+1}^* &\le f_{t+1}(\beta_1 \mathbf{w}_t) - f_{t+1}^* \\ &= \frac{\beta_1^2 \mathbf{r}_t^\top \widehat{\mathbf{M}}_\lambda^{-1} \mathbf{r}_t}{2} - \beta_1 \mathbf{r}_t^\top \mathbf{r}_t + \frac{\mathbf{r}_t \widehat{\mathbf{M}}_\lambda \mathbf{r}_t}{2} \\ &= \frac{\|\mathbf{r}_t\|^2}{2} \sum_{i=1}^d \xi_{ti}^2 \left(\frac{\beta_1^2}{\beta_i} - 2\beta_1 + \beta_i \right) \\ &= \frac{\|\mathbf{r}_t\|^2}{2} \sum_{i=1}^d \frac{\xi_{ti}^2}{\beta_i} (\beta_1 - \beta_i)^2 \\ &\le \frac{(\mathbf{w}_t^\top \widehat{\mathbf{B}} \mathbf{w}_t)}{2} \cdot \beta_1^2 \sum_{i=2}^d \frac{\xi_{ti}^2}{\beta_i}. \end{aligned}$$

Therefore, in view of (14), it suffices to set the ratio between the initial and the final error of f_{t+1} to

$$\max(1, G(\mathbf{r}_t)) \cdot \frac{16\beta_1^2}{(\beta_1 - \beta_2)^2}.$$

In the initial phase, $G(\mathbf{r}_t)$ is large, we can set the ratio to be $G(\mathbf{r}_0) \cdot \frac{16\beta_1^2}{(\beta_1-\beta_2)^2}$, until it is reduced to 1 after $\mathcal{O}(\log G(\mathbf{r}_0))$ iterations. Afterwards, we can set the ratio to be the constant of $\frac{16\beta_1^2}{(\beta_1-\beta_2)^2}$, until we reach the desired accuracy. Observe that

$$\frac{\beta_1^2}{(\beta_1 - \beta_2)^2} = \left(\frac{\frac{1}{\lambda - \widehat{\rho}_1}}{\frac{1}{\lambda - \widehat{\rho}_1} - \frac{1}{\lambda - \widehat{\rho}_2}} \right)^2 = \left(\frac{\lambda - \widehat{\rho}_2}{\widehat{\rho}_1 - \widehat{\rho}_2} \right)^2 \le (u+1)^2 \le 4.$$

■

D.3. Time complexity of SVRG for finite sum with nonconvex component

Lemma 25 (Time complexity of SVRG for (15)) *With the initialization $\alpha_t^* \mathbf{w}_t$, SVRG outputs an $\mathbf{w}_{t+1}$ such that $f_{t+1}(\mathbf{w}_{t+1}) - f_{t+1}^* \le \epsilon_t (\mathbf{w}_t^\top \widehat{\mathbf{B}} \mathbf{w}_t)$ in time*

$$\mathcal{O}\left(d(N + \kappa^2) \log(64 \max(G(\mathbf{r}_t), 1)) \cdot \kappa \right),$$

where $\kappa = \frac{\max_i L_i}{\Lambda}$ with L_i being the gradient Lipschitz constant of f_{t+1}^i, and Λ is the strongly-convex constant of f_{t+1}. Futhermore, if we sample each component f_{t+1}^i non-uniformly with probability proportional to L_i^2 for the SVRG stochastic updates, we have instead $\kappa = \sqrt{\frac{\frac{1}{N}\sum_{i=1}^N L_i^2}{\Lambda^2}}$.

Although not explicitly stated by Garber and Hazan (2015), the result for non-uniform sampling is straightforward by a careful investigation of their analysis; we provide detailed proof of this result in Appendix F. The effect of improved dependence on L_i's through non-uniform sampling agrees with related work (Xiao and Zhang, 2014). The purpose of the non-uniform sampling variant is to bound κ^2 with high probability for sub-Gaussian/regular polynomial-tail inputs.

D.4. Bounding the condition number for SVRG

The next lemma upper-bounds the "condition number" κ^2.

Lemma 26 *Solving $\min_{\mathbf{w}} f_{t+1}(\mathbf{w})$ using SVRG with non-uniform sampling, we have $\kappa^2 = \mathcal{O}\left(\frac{d^2}{\widehat{\Delta}^2 \gamma^2}\right)$ for the sub-Gaussian/regular polynomial-tail classes with high probability over the sample set, and $\kappa^2 = \mathcal{O}\left(\frac{1}{\widehat{\Delta}^2 \gamma^2}\right)$ for the bounded class.*

Proof The gradient Lipschitz constant L_i is bounded by the largest eigenvalue (in absolute value) of its Hessian

$$\mathbf{Q}_\lambda^i = \begin{bmatrix} \lambda \mathbf{x}_i \mathbf{x}_i^\top & -\mathbf{x}_i \mathbf{y}_i^\top \\ -\mathbf{y}_i \mathbf{x}_i^\top & \lambda \mathbf{y}_i \mathbf{y}_i^\top \end{bmatrix},$$

and the largest eigenvalue is defined as

$$\max_{\mathbf{g}_x \in \mathbb{R}^{d_x}, \mathbf{g}_y \in \mathbb{R}^{d_y}} \beta := \left| [\mathbf{g}_x^\top, \mathbf{g}_y^\top] \mathbf{Q}_\lambda^i \begin{bmatrix} \mathbf{g}_x \\ \mathbf{g}_y \end{bmatrix} \right| \quad \text{s.t.} \quad \|\mathbf{g}_x\|^2 + \|\mathbf{g}_y\|^2 = 1.$$

We have

$$\begin{aligned} \beta &= \left| \lambda (\mathbf{g}_x^\top \mathbf{x}_i)^2 + \lambda (\mathbf{g}_y^\top \mathbf{y}_i)^2 - 2(\mathbf{g}_x^\top \mathbf{x}_i)(\mathbf{g}_y^\top \mathbf{y}_i) \right| \\ &\le \lambda (\mathbf{g}_x^\top \mathbf{x}_i)^2 + \lambda (\mathbf{g}_y^\top \mathbf{y}_i)^2 + 2 \left| \mathbf{g}_x^\top \mathbf{x}_i \right| \left| \mathbf{g}_y^\top \mathbf{y}_i \right| \\ &\le \lambda (\mathbf{g}_x^\top \mathbf{x}_i)^2 + \lambda (\mathbf{g}_y^\top \mathbf{y}_i)^2 + (\mathbf{g}_x^\top \mathbf{x}_i)^2 + (\mathbf{g}_y^\top \mathbf{y}_i)^2 \\ &= (\lambda + 1) \left((\mathbf{g}_x^\top \mathbf{x}_i)^2 + (\mathbf{g}_y^\top \mathbf{y}_i) \right) \\ &\le (\lambda + 1) \left(\|\mathbf{g}_x\|^2 \|\mathbf{x}_i\|^2 + \|\mathbf{g}_y\|^2 \|\mathbf{y}_i\|^2 \right) \\ &\le (\lambda + 1) \cdot \max\left(\|\mathbf{x}_i\|^2, \|\mathbf{y}_i\|^2 \right) \\ &\le (\lambda + 1) \cdot \left(\|\mathbf{x}_i\|^2 + \|\mathbf{y}_i\|^2 \right) \end{aligned}$$

where we have used the Cauchy-Schwarz inequality in the third inequality.

Note that, for bounded inputs, we have $\|\mathbf{x}_i\|^2 + \|\mathbf{y}_i\|^2 \le 2$ and so $L_i^2 \le 4(\lambda+1)^2$ for all $i = 1, \ldots, N$. For sub-Gaussian/regular polynomial-tail inputs, we have

$$\frac{1}{N} \sum_{i=1}^N L_i^2 \le (\lambda + 1)^2 \cdot \frac{1}{N} \sum_{i=1}^N \left(\|\mathbf{x}_i\|^2 + \|\mathbf{y}_i\|^2 \right)^2 = \mathcal{O}((\lambda + 1)^2 d^2)$$

with high probability in view of Remark 24.

On the other hand, we have shown that $\Lambda = \sigma_{\min}(\mathbf{A}_\lambda) \ge (\lambda - \widehat{\rho}_1)\gamma/2$. Recalling $\lambda = \widehat{\rho}_1 + c\widehat{\Delta}$ with $c \in (0, 1)$, we have $\lambda \le 2$ and $\Lambda \ge c\widehat{\Delta}\gamma/2$. Combining this with the data norm bound above yields the desired result. ■

D.5. Proof of Theorem 12

Proof Since $\frac{\mathbf{u}_T{}^\top \mathbf{\Sigma}_{xx}\widehat{\mathbf{u}}}{\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\right\|} \le 1$ and $\frac{\mathbf{v}_T{}^\top \mathbf{\Sigma}_{yy}\widehat{\mathbf{v}}}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\right\|} \le 1$, it suffices to require

$$\frac{\mathbf{u}_T{}^\top \mathbf{\Sigma}_{xx}\widehat{\mathbf{u}}}{\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\right\|} + \frac{\mathbf{v}_T{}^\top \mathbf{\Sigma}_{yy}\widehat{\mathbf{v}}}{\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\right\|} \ge 2-\eta.$$

According to Lemma 22 (making the identification that $\mathbf{a} = \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}}$, $\mathbf{x} = \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T$, $\mathbf{b} = \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}$, and $\mathbf{y} = \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T$), it then suffices to have

$$\cos\theta_T = \frac{1}{\sqrt{2}}\frac{\widehat{\mathbf{u}}^\top \mathbf{\Sigma}_{xx}\mathbf{u}_T + \widehat{\mathbf{v}}^\top \mathbf{\Sigma}_{yy}\mathbf{v}_T}{\sqrt{\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}_T + \mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}_T}} \ge 1-\frac{\eta}{8}. \tag{28}$$

Since $\cos\theta_T = \sqrt{1-\sin^2\theta_T} \ge 1-\sin^2\theta_T$, we just need $|\sin\theta_T| \le \frac{\sqrt{\eta}}{\sqrt{8}}$, and we ensure it by requiring $G(\mathbf{r}_T) \le \frac{\sqrt{\eta}}{\sqrt{8}}$. Applying results from the previous sections, we need to solve $\mathcal{O}\left(\log\frac{G(\mathbf{r}_0)}{\eta}\right)$ linear systems, and the time complexity for solving each is at most $\mathcal{O}\left((N+\kappa^2)\log\left(G(\mathbf{r}_0)\cdot\kappa\right)\right)$ for SVRG.

It remains to bound $G(\mathbf{r}_0)$. By the definition of $G(\cdot)$, we have

$$G(\mathbf{r}_0) = \frac{\sqrt{\sum_{i=2}^d \xi_{0i}^2/\beta_i}}{\sqrt{\xi_{01}^2/\beta_1}} \le \sqrt{\frac{\beta_1}{\beta_d}}\cdot\frac{1}{|\xi_{01}|}$$

where $|\xi_{01}|$ is the alignment between $\mathbf{r}_0$ and $\mathbf{p}_1$ which, by the relationship between $\mathbf{r}_0$ and $\mathbf{w}_0$, satisfy

$$\begin{aligned}\left|\mathbf{r}_0^\top\mathbf{p}_1\right| &= \frac{\left|\tilde{\mathbf{w}}_0^\top\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1\right|}{\left\|\widehat{\mathbf{B}}^{\frac{1}{2}}\tilde{\mathbf{w}}_0\right\|} \ge \frac{\left|\tilde{\mathbf{w}}_0^\top(\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1)\right|}{\sigma_{\max}(\widehat{\mathbf{B}}^{\frac{1}{2}})\cdot\|\tilde{\mathbf{w}}_0\|} = \frac{\left\|\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1\right\|\cdot\left|(\tilde{\mathbf{w}}_0/\|\tilde{\mathbf{w}}_0\|)^\top(\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1/\left\|\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1\right\|)\right|}{\sigma_{\max}(\widehat{\mathbf{B}}^{\frac{1}{2}})}\\ &\ge \frac{\sigma_{\min}(\widehat{\mathbf{B}}^{\frac{1}{2}})}{\sigma_{\max}(\widehat{\mathbf{B}}^{\frac{1}{2}})}\cdot\left|\left(\frac{\tilde{\mathbf{w}}_0}{\|\tilde{\mathbf{w}}_0\|}\right)^\top\left(\frac{\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1}{\left\|\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1\right\|}\right)\right|.\end{aligned}$$

According to the way $\tilde{\mathbf{w}}_0$ is initialized and Arora et al. (2009, Lemma 5), we have with probability at least $1-C$ that $\left|\left(\frac{\tilde{\mathbf{w}}_0}{\|\tilde{\mathbf{w}}_0\|}\right)^\top\left(\frac{\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1}{\left\|\widehat{\mathbf{B}}^{\frac{1}{2}}\mathbf{p}_1\right\|}\right)\right| \ge \frac{C}{\sqrt{d}}$. On the other hand, we have $\frac{\beta_1}{\beta_d} = \mathcal{O}(\frac{1}{\widehat{\Delta}})$ and $\frac{\sigma_{\max}(\widehat{\mathbf{B}}^{\frac{1}{2}})}{\sigma_{\min}(\widehat{\mathbf{B}}^{\frac{1}{2}})} = \mathcal{O}\left(\frac{1}{\sqrt{\gamma}}\right)$. Combining these results yields that $G(\mathbf{r}_0) = \mathcal{O}\left(\sqrt{\frac{d}{\widehat{\Delta}\gamma}}\right)$ with high probability. Then the theorem follows. ■

D.6. Proof of Corollary 13

Proof Denote $\widetilde{\mathbf{r}} := \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T/\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\right\| \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T/\left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\right\|\end{bmatrix}$, with $\|\widetilde{\mathbf{r}}\| = 1$. Assume without loss of generality that $\left\|\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\right\| = \left\|\mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\right\| = 1$; this does not affect our measure of alignment, and can be ensured by a final (separate) normalization step with cost $\mathcal{O}(Nd)$ (Wang et al., 2016).

Apply Lemma 6 with $\epsilon' = \frac{\sqrt{\epsilon}\Delta}{8}$; requiring that $\epsilon' = \frac{\sqrt{\epsilon}\Delta}{8} \le C_d\gamma^2$ as in Corollary 7 leads to the extra condition that $\epsilon \le \frac{64C_d^2\gamma^4}{\Delta^2}$.

With the specified sample complexity, we have that with high probability

$$\left\|\mathbf{T}-\widehat{\mathbf{T}}\right\| \le \frac{\sqrt{\epsilon}\Delta}{8} \le \frac{\Delta}{8}. \tag{29}$$

In view of the Weyl's inequality, (29) implies that $\widehat{\Delta} \ge \frac{3\Delta}{4}$.

Let $\mathbf{r}^* = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^* \\ \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^*\end{bmatrix}$ be the top eigenvector of $\mathbf{C}$. And recall $\widehat{\mathbf{r}} := \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{\Sigma}_{xx}^{\frac{1}{2}}\widehat{\mathbf{u}} \\ \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\widehat{\mathbf{v}}\end{bmatrix}$ is the top eigenvector of $\widehat{\mathbf{C}}$. According to the Davis-Kahan $\sin\theta$ theorem (Davis and Kahan, 1970), with the number of samples given in the theorem, the top eigenvectors of $\mathbf{C}$ and $\widehat{\mathbf{C}}$ are well aligned:

$$\sin^2\theta \le \frac{\left\|\mathbf{C}-\widehat{\mathbf{C}}\right\|^2}{\Delta^2} \le \frac{\epsilon}{64}$$

where θ is the angle between $\mathbf{r}^*$ and $\widehat{\mathbf{r}}$. This implies that

$$\widehat{\mathbf{r}}^\top\mathbf{r}^* = \cos\theta = \sqrt{1-\sin^2\theta} \ge 1-\sin^2\theta \ge 1-\frac{\epsilon}{64}.$$

We now show that the theorem follows if we manage to solve the ERM objective so accurately that

$$\widetilde{\mathbf{r}}^\top\widehat{\mathbf{r}} = \frac{1}{2}\left(\frac{\widehat{\mathbf{u}}^\top\mathbf{\Sigma}_{xx}\mathbf{u}_T}{\sqrt{\mathbf{u}_T^\top\mathbf{\Sigma}_{xx}\mathbf{u}_T}} + \frac{\widehat{\mathbf{v}}^\top\mathbf{\Sigma}_{yy}\mathbf{v}_T}{\sqrt{\mathbf{v}_T^\top\mathbf{\Sigma}_{yy}\mathbf{v}_T}}\right) \ge 1-\frac{\epsilon^2}{8192}. \tag{30}$$

To see this, first observe that (30) implies

$$\|\widetilde{\mathbf{r}}-\widehat{\mathbf{r}}\| = \sqrt{2-2(\widetilde{\mathbf{r}}^\top\widehat{\mathbf{r}})} \le \frac{\epsilon}{64},$$

and as a result

$$\widetilde{\mathbf{r}}^\top\mathbf{r}^* \ge \widehat{\mathbf{r}}^\top\mathbf{r}^* - \left|(\widetilde{\mathbf{r}}-\widehat{\mathbf{r}})^\top\mathbf{r}^*\right| \ge \widehat{\mathbf{r}}^\top\mathbf{r}^* - \|\widetilde{\mathbf{r}}-\widehat{\mathbf{r}}\| \ge 1-\frac{\epsilon}{32}.$$

Consequently, we have

$$\frac{1}{2}\left(\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^* - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\right\|^2 + \left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^* - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\right\|^2\right) = \|\widetilde{\mathbf{r}} - \mathbf{r}^*\|^2 = 2\left(1 - \widetilde{\mathbf{r}}^\top \mathbf{r}^*\right) \le \frac{\epsilon}{16}$$

and so $\max\left(\left\|\mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^* - \mathbf{\Sigma}_{xx}^{\frac{1}{2}}\mathbf{u}_T\right\|^2, \left\|\mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^* - \mathbf{\Sigma}_{yy}^{\frac{1}{2}}\mathbf{v}_T\right\|^2\right) \le \frac{\epsilon}{8}$. We are now in the same situation as (27); we can fix the incorrect normalization of $\widetilde{\mathbf{r}}$ analogously and then our lemma follows.

It remains to show the time complexity to achieve (30). According to Lemma 22, it suffices to have

$$\cos\theta_T = \frac{1}{\sqrt{2}}\frac{\widehat{\mathbf{u}}^\top \mathbf{\Sigma}_{xx}\mathbf{u}_T + \widehat{\mathbf{v}}^\top \mathbf{\Sigma}_{yy}\mathbf{v}_T}{\sqrt{\mathbf{u}_T^\top \mathbf{\Sigma}_{xx}\mathbf{u}_T + \mathbf{v}_T^\top \mathbf{\Sigma}_{yy}\mathbf{v}_T}} \ge 1 - \frac{\epsilon^2}{2^{15}}. \tag{31}$$

In turn, it suffices to have $|\sin\theta_T| \le \frac{\epsilon}{256}$ and we ensure it by requiring $G(\mathbf{r}_T) \le \frac{\epsilon}{256}$. We obtain the stated time complexity by applying Theorem 12 with $\eta = \frac{\epsilon}{256}$. ■

Appendix E. Proofs for Section 5

E.1. Proof of Lemma 15

Proof The desired result is a direct consequence of Lemma 22, by making the identification that

$$\mathbf{a} = \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}^*, \qquad \mathbf{x} = \mathbf{E}_{xx}^{\frac{1}{2}}\mathbf{u}_T, \qquad \mathbf{b} = \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}^*, \qquad \mathbf{y} = \mathbf{E}_{yy}^{\frac{1}{2}}\mathbf{v}_T.$$

■

E.2. Parameters of Streaming SVRG for stochastic least squares

We divide Lemma 16 into the following three propositions.

Proposition 27 (Strong convexity) *For any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$, we have*

$$f_{t+1}(\mathbf{w}) \ge f_{t+1}(\mathbf{w}') + \left\langle \nabla f_{t+1}(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \right\rangle + \frac{\mu}{2}\left\|\mathbf{w} - \mathbf{w}'\right\|^2$$

where $\mu := \frac{\gamma}{\beta_1} \ge C\Delta\gamma$ for some $C > 0$.

Proof Just observe that the Hessian of $f_{t+1}(\mathbf{w})$ is $\mathbf{A}_\lambda = \mathbf{B}^{\frac{1}{2}}\mathbf{M}_\lambda^{-1}\mathbf{B}^{\frac{1}{2}}$, whose eigenvalues are bounded from below: $\sigma_{\min}(\mathbf{A}_\lambda) \ge (\lambda - \rho_1)\cdot\sigma_{\min}(\mathbf{B}) = \gamma/\beta_1$. The lemma follows from the assumption that $\lambda = \rho_1 + c\Delta$ for $c \in (0,1)$. ■

Proposition 28 (Streaming smoothness) *For any $\mathbf{w} \in \mathbb{R}^d$, we have*

$$\mathbb{E}\left[\left\|\nabla\phi_{t+1}(\mathbf{w}) - \nabla\phi_{t+1}(\mathbf{w}_{t+1}^*)\right\|^2\right] \le 2S\left(f_{t+1}(\mathbf{w}) - f_{t+1}^*\right)$$

where $S = \mathcal{O}\left(\frac{d\beta_1}{\gamma}\right) = \mathcal{O}\left(\frac{d}{\Delta\gamma}\right)$ for the sub-Gaussian/regular polynomial-tail classes, and $S = \mathcal{O}\left(\frac{\beta_1}{\gamma}\right) = \mathcal{O}\left(\frac{1}{\Delta\gamma}\right)$ for the bounded class.

Proof Observe that

$$\nabla\phi_{t+1}(\mathbf{w}) = \begin{bmatrix} \lambda\mathbf{x}\mathbf{x}^\top & -\mathbf{x}\mathbf{y}^\top \\ -\mathbf{y}\mathbf{x}^\top & \lambda\mathbf{y}\mathbf{y}^\top \end{bmatrix}\mathbf{w} - \begin{bmatrix} \mathbf{x}\mathbf{x}^\top & \\ & \mathbf{y}\mathbf{y}^\top \end{bmatrix}\mathbf{w}_t.$$

As shown in Lemma 26, this gradient function is Lipschitz continuous:

$$\left\|\nabla\phi_{t+1}(\mathbf{w}) - \nabla\phi_{t+1}(\mathbf{w}_{t+1}^*)\right\| \le (\lambda+1)\cdot\sup\left(\|\mathbf{x}\|, \|\mathbf{y}\|\right)\cdot\left\|\mathbf{w} - \mathbf{w}_{t+1}^*\right\|.$$

Note that $\lambda \le \rho_1 + u\Delta$ where $\rho_1 \le 1$, $\Delta \le 1$, and $u < 1$ by assumption, and thus $\lambda \le 2$. As a result, we obtain

$$\mathbb{E}\left\|\nabla\phi_{t+1}(\mathbf{w}) - \nabla\phi_{t+1}(\mathbf{w}_{t+1}^*)\right\|^2 \le 9\mathbb{E}\left[\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2\right]\cdot\left\|\mathbf{w} - \mathbf{w}_{t+1}^*\right\|^2.$$

For the distributions of $P(\mathbf{x},\mathbf{y})$ considered here, $\mathbb{E}\|\mathbf{x}\|^2$ and $\mathbb{E}\|\mathbf{y}\|^2$ are both $\mathcal{O}(d)$ for the sub-Gaussian/regular polynomial-tail inputs (see Remark 24), and bounded by 1 for the bounded inputs.

On the other hand, according to Lemma 27, we have

$$f(\mathbf{w}) - f(\mathbf{w}_{t+1}^*) \ge C\Delta\gamma\left\|\mathbf{w} - \mathbf{w}_{t+1}^*\right\|^2$$

for some $C > 0$.

Combining the above two inequalities gives the desired result. ■

Proposition 29 (Streaming variance) *We have*

$$\mathbb{E}\left[\frac{1}{2}\left\|\nabla\phi(\mathbf{w}_{t+1}^*)\right\|^2_{\left(\nabla^2 f(\mathbf{w}_{t+1}^*)\right)^{-1}}\right] \le \sigma^2.$$

where $\sigma^2 = \mathcal{O}\left(d\beta_1^3\|\mathbf{r}_t\|^2\right)$ for the sub-Gaussian/regular polynomial-tail classes, and $\sigma^2 = \mathcal{O}\left(\frac{\beta_1^3\|\mathbf{r}_t\|^2}{\gamma^2}\right)$ for the bounded class.

Proof Observe that $\mathbf{w}_{t+1}^* = \mathbf{A}_\lambda^{-1}\mathbf{B}\mathbf{w}_t$ and

$$\nabla\phi(\mathbf{w}_{t+1}^*) = \left(\begin{bmatrix} \lambda\mathbf{x}\mathbf{x}^\top & -\mathbf{x}\mathbf{y}^\top \\ -\mathbf{y}\mathbf{x}^\top & \lambda\mathbf{y}\mathbf{y}^\top \end{bmatrix}\mathbf{A}_\lambda^{-1}\mathbf{B} - \begin{bmatrix} \mathbf{x}\mathbf{x}^\top & \\ & \mathbf{y}\mathbf{y}^\top \end{bmatrix}\right)\mathbf{w}_t.$$

Define the shorthands $\mathbf{D} = \mathbf{B}^{-\frac{1}{2}} \begin{bmatrix} \lambda \mathbf{x}\mathbf{x}^\top & -\mathbf{x}\mathbf{y}^\top \\ -\mathbf{y}\mathbf{x}^\top & \lambda \mathbf{y}\mathbf{y}^\top \end{bmatrix} \mathbf{B}^{-\frac{1}{2}}$ and $\mathbf{E} = \mathbf{B}^{-\frac{1}{2}} \begin{bmatrix} \mathbf{x}\mathbf{x}^\top & \\ & \mathbf{y}\mathbf{y}^\top \end{bmatrix} \mathbf{B}^{-\frac{1}{2}}$.

Then we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{2}\left\|\nabla\phi(\mathbf{w}_{t+1}^*)\right\|^2_{(\nabla^2 f(\mathbf{w}_{t+1}^*))^{-1}}\right] &= \mathbb{E}\left[\frac{1}{2}\left\|\nabla\phi(\mathbf{w}_{t+1}^*)\right\|^2_{\mathbf{A}_\lambda^{-1}}\right] \\
&= \mathbb{E}\left[\frac{1}{2}\left\|\mathbf{B}^{-\frac{1}{2}}\nabla\phi(\mathbf{w}_{t+1}^*)\right\|^2_{\mathbf{B}^{\frac{1}{2}}\mathbf{A}_\lambda^{-1}\mathbf{B}^{\frac{1}{2}}}\right] \\
&= \frac{1}{2}\mathbb{E}\left[\left(\mathbf{B}^{\frac{1}{2}}\mathbf{w}_t\right)^\top (\mathbf{M}_\lambda\mathbf{D} - \mathbf{E})\cdot\mathbf{M}_\lambda\cdot(\mathbf{D}\mathbf{M}_\lambda - \mathbf{E})\left(\mathbf{B}^{\frac{1}{2}}\mathbf{w}_t\right)\right] \\
&= \frac{1}{2}\mathbb{E}\left[\mathbf{r}_t^\top(\mathbf{M}_\lambda\mathbf{D} - \mathbf{E})\cdot\mathbf{M}_\lambda\cdot(\mathbf{D}\mathbf{M}_\lambda - \mathbf{E})\mathbf{r}_t\right]. \qquad (32)
\end{aligned}$$

Bounded case For the bounded case where $\sup\left(\|\mathbf{x}\|^2, \|\mathbf{y}\|^2\right) \le 1$, the derivation is relatively simple. We can bound $\|\mathbf{D}\| \le \frac{3}{\gamma}$ and $\|\mathbf{E}\| \le \frac{1}{\gamma}$, and thus

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{2}\left\|\nabla\phi(\mathbf{w}_{t+1}^*)\right\|^2_{(\nabla^2 f(\mathbf{w}_{t+1}^*))^{-1}}\right] &\le \frac{\|\mathbf{M}_\lambda\|\,\|\mathbf{r}_t\|^2}{2}\mathbb{E}\left\|\mathbf{D}\mathbf{M}_\lambda - \mathbf{E}\right\|^2 \\
&\le \beta_1\|\mathbf{r}_t\|^2\left(\mathbb{E}\|\mathbf{D}\mathbf{M}_\lambda\|^2 + \mathbb{E}\|\mathbf{E}\|^2\right) \\
&= \mathcal{O}\left(\frac{\beta_1^3\|\mathbf{r}_t\|^2}{\gamma^2}\right)
\end{aligned}$$

where we have used the triangle inequality and the fact that $(x+y)^2 \le 2x^2 + 2y^2$ in the second inequality.

Sub-Gaussian/regular polynomial tail cases We now omit the subscript λ in $\mathbf{M}_\lambda$ and t from iterates for convenience. Using the fact that $\|\mathbf{x}+\mathbf{y}\|^2 \le 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2$ with $\mathbf{x} = \mathbf{M}^{\frac{1}{2}}\mathbf{D}\mathbf{M}\mathbf{r}$ and $\mathbf{y} = \mathbf{M}^{\frac{1}{2}}\mathbf{E}\mathbf{r}$, we continue from (32) and obtain

$$\mathbb{E}\left[\frac{1}{2}\left\|\nabla\phi(\mathbf{w}_{t+1}^*)\right\|^2_{(\nabla^2 f(\mathbf{w}_{t+1}^*))^{-1}}\right] \le \mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}\mathbf{M}\mathbf{D}\mathbf{M}\mathbf{r}\right] + \mathbb{E}\left[\mathbf{r}^\top\mathbf{E}\mathbf{M}\mathbf{E}\mathbf{r}\right].$$

Introduce the notation $\mathbf{u} = \mathbf{E}_{xx}^{-\frac{1}{2}}\mathbf{x}$, $\mathbf{v} = \mathbf{E}_{yy}^{-\frac{1}{2}}\mathbf{y}$, and parition $\mathbf{r}$ and $\mathbf{M}$ according to $(\mathbf{x}, \mathbf{y})$:

$$\mathbf{r} = \begin{bmatrix}\mathbf{r}_x \\ \mathbf{r}_y\end{bmatrix}, \qquad \mathbf{M} = \begin{bmatrix}\mathbf{M}_{xx} & \mathbf{M}_{xy} \\ \mathbf{M}_{yx} & \mathbf{M}_{yy}\end{bmatrix}.$$

In view of Lemma 23, we can assume $\max\left(\mathbb{E}\|\mathbf{u}\|^4, \mathbb{E}\|\mathbf{v}\|^4\right) \le Cd^2$. From now on, we use C for a generic constant whose specific value may change between appearances.

We have

$$\mathbf{E}\mathbf{M}\mathbf{E} = \begin{bmatrix}\mathbf{u}\mathbf{u}^\top\mathbf{M}_{xx}\mathbf{u}\mathbf{u}^\top & \mathbf{u}\mathbf{u}^\top\mathbf{M}_{xy}\mathbf{v}\mathbf{v}^\top \\ \mathbf{v}\mathbf{v}^\top\mathbf{M}_{yx}\mathbf{u}\mathbf{u}^\top & \mathbf{v}\mathbf{v}^\top\mathbf{M}_{yy}\mathbf{v}\mathbf{v}^\top\end{bmatrix}.$$

and thus

$$\mathbf{r}^\top\mathbf{E}\mathbf{M}\mathbf{E}\mathbf{r} = \mathbf{r}_x^\top\mathbf{u}\mathbf{u}^\top\mathbf{M}_{xx}\mathbf{u}\mathbf{u}^\top\mathbf{r}_x + \mathbf{r}_y^\top\mathbf{v}\mathbf{v}^\top\mathbf{M}_{yy}\mathbf{v}\mathbf{v}^\top\mathbf{r}_y + 2\mathbf{r}_x^\top\mathbf{u}\mathbf{u}^\top\mathbf{M}_{xy}\mathbf{v}\mathbf{v}^\top\mathbf{r}_y.$$

We have for the first term that

$$\begin{aligned}
\mathbb{E}\left[\mathbf{r}_x^\top\mathbf{u}\mathbf{u}^\top\mathbf{M}_{xx}\mathbf{u}\mathbf{u}^\top\mathbf{r}_x\right] &= \mathbb{E}\left[\left|\mathbf{r}_x^\top\mathbf{u}\right|^2\mathbf{u}^\top\mathbf{M}_{xx}\mathbf{u}\right] \\
&\le \sqrt{\mathbb{E}\left|\mathbf{r}_x^\top\mathbf{u}\right|^4}\sqrt{\mathbb{E}\left|\mathbf{u}^\top\mathbf{M}_{xx}\mathbf{u}\right|^2} \\
&\le C\|\mathbf{r}_x\|^2\sqrt{\mathbb{E}\left|\mathbf{u}^\top\mathbf{M}_{xx}\mathbf{u}\right|^2} \\
&\le C\|\mathbf{M}_{xx}\|\,\|\mathbf{r}_x\|^2\sqrt{\mathbb{E}\|\mathbf{u}\|^4} \\
&\le C\|\mathbf{M}\|\,\|\mathbf{r}_x\|^2 d.
\end{aligned}$$

Similar arguments also lead to

$$\mathbb{E}\left[\mathbf{r}_y^\top\mathbf{v}\mathbf{v}^\top\mathbf{M}_{yy}\mathbf{v}\mathbf{v}^\top\mathbf{r}_y\right] \le C\|\mathbf{M}\|\,\|\mathbf{r}_y\|^2 d.$$

For the third term, we have

$$\begin{aligned}
\mathbb{E}\left[\mathbf{r}_x^\top\mathbf{u}\mathbf{u}^\top\mathbf{M}_{xy}\mathbf{v}\mathbf{v}^\top\mathbf{r}_y\right] &\le \sqrt{\mathbb{E}\left|\mathbf{r}_x^\top\mathbf{u}\right|^2\left|\mathbf{r}_y^\top\mathbf{v}\right|^2}\sqrt{\mathbb{E}\left|\mathbf{u}^\top\mathbf{M}_{xy}\mathbf{v}\right|^2} \\
&\le \|\mathbf{M}\|\left(\mathbb{E}\left|\mathbf{r}_x^\top\mathbf{u}\right|^4\right)^{\frac{1}{4}}\left(\mathbb{E}\left|\mathbf{r}_y^\top\mathbf{v}\right|^4\right)^{\frac{1}{4}}\left(\mathbb{E}\|\mathbf{u}\|^4\right)^{\frac{1}{4}}\left(\mathbb{E}\|\mathbf{v}\|^4\right)^{\frac{1}{4}} \\
&\le C\|\mathbf{M}\|\,\|\mathbf{r}_x\|\,\|\mathbf{r}_y\| d.
\end{aligned}$$

Therefore,

$$\mathbb{E}\left[\mathbf{r}^\top\mathbf{E}\mathbf{M}\mathbf{E}\mathbf{r}\right] \le C\|\mathbf{M}\|\,\|\mathbf{r}\|^2 d.$$

Now we need to bound $\mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}\mathbf{M}\mathbf{D}\mathbf{M}\mathbf{r}\right]$. Using the fact that $\|\mathbf{x}+\mathbf{y}\|^2 \le 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2$ with $\mathbf{x} = \mathbf{M}^{\frac{1}{2}}\mathbf{D}_1\mathbf{M}\mathbf{r}$ and $\mathbf{y} = \mathbf{M}^{\frac{1}{2}}\mathbf{D}_2\mathbf{M}\mathbf{r}$, this can be bounded by two terms:

$$\mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}\mathbf{M}\mathbf{D}\mathbf{M}\mathbf{r}\right] \le 2\mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}_1\mathbf{M}\mathbf{D}_1\mathbf{M}\mathbf{r}\right] + 2\mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}_2\mathbf{M}\mathbf{D}_2\mathbf{M}\mathbf{r}\right]$$

where

$$\mathbf{D}_1 = \lambda\begin{bmatrix}\mathbf{u}\mathbf{u}^\top & 0 \\ 0 & \mathbf{v}\mathbf{v}^\top\end{bmatrix}, \qquad \mathbf{D}_2 = -\begin{bmatrix}0 & \mathbf{u}\mathbf{v}^\top \\ \mathbf{v}\mathbf{u}^\top & 0\end{bmatrix}.$$

The bound for $\mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}_1\mathbf{M}\mathbf{D}_1\mathbf{M}\mathbf{r}\right]$ can be derived using the same argument that bounds $\mathbb{E}\left[\mathbf{r}^\top\mathbf{E}\mathbf{M}\mathbf{E}\mathbf{r}\right]$ (now $\mathbf{M}\mathbf{r}$ plays the role of $\mathbf{r}$ in bounding $\mathbb{E}\left[\mathbf{r}^\top\mathbf{E}\mathbf{M}\mathbf{E}\mathbf{r}\right]$), and thus we have

$$\mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}_1\mathbf{M}\mathbf{D}_1\mathbf{M}\mathbf{r}\right] \le C\lambda^2\|\mathbf{M}\|\,\|\mathbf{M}\mathbf{r}\|^2 d \le C\|\mathbf{M}\|^3\|\mathbf{r}\|^2\lambda^2 d.$$

Finally, we bound $\mathbb{E}\left[\mathbf{r}^\top\mathbf{M}\mathbf{D}_2\mathbf{M}\mathbf{D}_2\mathbf{M}\mathbf{r}\right]$. Note that

$$-\mathbf{D}_2\mathbf{M}\mathbf{D}_2 = \begin{bmatrix}\mathbf{u}\mathbf{v}^\top\mathbf{M}_{yy}\mathbf{v}\mathbf{u}^\top & \mathbf{u}\mathbf{v}^\top\mathbf{M}_{yx}\mathbf{u}\mathbf{v} \\ \mathbf{v}\mathbf{u}^\top\mathbf{M}_{xy}\mathbf{v}\mathbf{u}^\top & \mathbf{v}\mathbf{u}^\top\mathbf{M}_{xx}\mathbf{u}\mathbf{v}^\top\end{bmatrix}.$$

Let

$$\mathbf{M}\mathbf{r} = \begin{bmatrix} \mathbf{m}_x \\ \mathbf{m}_y \end{bmatrix},$$

and then

$$-\mathbf{r}^\top \mathbf{M}\mathbf{D}_2\mathbf{M}\mathbf{D}_2\mathbf{M}\mathbf{r} = \mathbf{m}_x^\top \mathbf{u}\mathbf{v}^\top \mathbf{M}_{yy}\mathbf{v}\mathbf{u}^\top \mathbf{m}_x + \mathbf{m}_y^\top \mathbf{v}\mathbf{u}^\top \mathbf{M}_{xx}\mathbf{u}\mathbf{v}^\top \mathbf{m}_y + 2\mathbf{m}_x^\top \mathbf{u}\mathbf{v}^\top \mathbf{M}_{yx}\mathbf{u}\mathbf{v}^\top \mathbf{m}_y.$$

Similarly to what we have done above,

$$\begin{aligned}
\mathbb{E}\left|\mathbf{m}_x^\top \mathbf{u}\mathbf{v}^\top \mathbf{M}_{yy}\mathbf{v}\mathbf{u}^\top \mathbf{m}_x\right| &\le \sqrt{\mathbb{E}\left|\mathbf{m}_x^\top \mathbf{u}\right|^4}\sqrt{\mathbb{E}\left|\mathbf{v}^\top \mathbf{M}_{yy}\mathbf{v}\right|^2} \\
&\le C\left\|\mathbf{M}\right\|\left\|\mathbf{m}_x\right\|^2 d \\
&\le C\left\|\mathbf{M}\right\|^3\left\|\mathbf{r}\right\|^2 d.
\end{aligned}$$

The same bound also holds for $\mathbb{E}\left|\mathbf{m}_y^\top \mathbf{v}\mathbf{u}^\top \mathbf{M}_{xx}\mathbf{u}\mathbf{v}^\top \mathbf{m}_y\right|$ with the same argument. For the term $\mathbb{E}\left|\mathbf{m}_x^\top \mathbf{u}\mathbf{v}^\top \mathbf{M}_{yx}\mathbf{u}\mathbf{v}^\top \mathbf{m}_y\right|$, we have

$$\begin{aligned}
\mathbb{E}\left|\mathbf{m}_x^\top \mathbf{u}\mathbf{v}^\top \mathbf{M}_{yx}\mathbf{u}\mathbf{v}^\top \mathbf{m}_y\right| &\le \left\|\mathbf{M}\right\|\left(\mathbb{E}\left|\mathbf{m}_x\mathbf{u}\right|^4\right)^{\frac{1}{4}}\left(\mathbb{E}\left|\mathbf{m}_y\mathbf{v}\right|^4\right)^{\frac{1}{4}}\left(\mathbb{E}\left\|\mathbf{u}\right\|^4\right)^{\frac{1}{4}}\left(\mathbb{E}\left\|\mathbf{v}\right\|^4\right)^{\frac{1}{4}} \\
&\le C\left\|\mathbf{M}\right\|\left\|\mathbf{m}_x\right\|\left\|\mathbf{m}_y\right\| d \\
&\le C\left\|\mathbf{M}\right\|^3\left\|\mathbf{r}\right\|^2 d.
\end{aligned}$$

Combining all the terms, and noting that $\lambda \le 2$, we have shown that

$$\mathbb{E}\left[\mathbf{r}^\top \mathbf{M}\mathbf{D}\mathbf{M}\mathbf{D}\mathbf{M}\mathbf{r}\right] \le C\left\|\mathbf{M}\right\|^3\left\|\mathbf{r}\right\|^2 d.$$

And the final bound is

$$\mathbb{E}\left[\mathbf{r}^\top(\mathbf{M}\mathbf{D}-\mathbf{E})\mathbf{M}(\mathbf{D}\mathbf{M}-\mathbf{E})\mathbf{r}\right] \le C\left[\left\|\mathbf{M}\right\|^3 + \left\|\mathbf{M}\right\|\right]\left\|\mathbf{r}\right\|^2 d = \mathcal{O}\left(\beta_1^3\left\|\mathbf{r}\right\|^2 d\right).$$

■

E.3. Proof of Lemma 17

Proof For notational simplicity, we omit the subscript $t+1$ below.

According to Frostig et al. (2015, Theorem 4.1), we have that for iteration τ of Algorithm 1

$$\begin{aligned}
\mathbb{E}\left[f(\mathbf{w}^\tau) - f^*\right] \le \frac{1}{1-4s}\Bigg[&\left(\frac{S}{\mu m_\tau s} + 4s\right)\mathbb{E}\left[f(\mathbf{w}^{\tau-1}) - f^*\right] \\
&+ \frac{1+2s}{k_\tau}\left(\sqrt{\frac{S}{\mu}\mathbb{E}\left[f(\mathbf{w}^{\tau-1}) - f^*\right]} + \sigma\right)^2\Bigg]. \qquad (33)
\end{aligned}$$

Using the inequality $(x+y)^2 \le 2(x^2+y^2)$, it holds that

$$\left(\sqrt{\frac{S}{\mu}\mathbb{E}\left[f(\mathbf{w}^{\tau-1}) - f^*\right]} + \sigma\right)^2 \le \frac{2S}{\mu}\mathbb{E}\left[f(\mathbf{w}^{\tau-1}) - f^*\right] + 2\sigma^2.$$

Now, set for this iteration $s = \frac{c_2}{8}$, $m_\tau = \lceil \frac{S}{\mu c_2^2} \rceil$, and $k_\tau = \max\left(\lceil \frac{S}{\mu c_2} \rceil, \lceil \frac{\sigma^2}{\beta_1 \|\mathbf{r}_t\|^2 c_3} \rceil\right)$, for some $c_2, c_3 \in (0,1)$. We continue from (33) and have

$$\begin{aligned}
\mathbb{E}\left[f(\mathbf{w}^\tau) - f^*\right] &\le \frac{1}{1-4s}\left[\left(\frac{S}{\mu m_\tau s} + 4s + \frac{2+4s}{k_\tau}\frac{S}{\mu}\right)\mathbb{E}\left[f(\mathbf{w}^{\tau-1}) - f^*\right] + \frac{2+4s}{k_\tau}\sigma^2\right] \\
&\le \frac{1}{1-c_2/2}\left[\left(8c_2 + \frac{c_2}{2} + \frac{2+4c_2}{2}c_2\right)\mathbb{E}\left[f(\mathbf{w}^{\tau-1}) - f^*\right] + \frac{4+c_2}{2k_\tau}\sigma^2\right] \\
&\le 22c_2 \cdot \mathbb{E}\left[f(\mathbf{w}^{\tau-1}) - f^*\right] + 10c_3 \cdot \frac{\beta_1\left\|\mathbf{r}_t\right\|^2}{2}.
\end{aligned}$$

We can now calculate the number of samples used in this iteration, which is

$$k_\tau + m_\tau = \mathcal{O}\left(\frac{d\beta_1^2}{c_3} + \frac{d\beta_1^2}{\gamma^2 c_2^2}\right) = \mathcal{O}\left(\frac{d}{\Delta^2 c_3} + \frac{d}{\Delta^2\gamma^2 c_2^2}\right) \qquad (34)$$

for sub-Gaussian/regular polynomial-tail inputs, and

$$k_\tau + m_\tau = \mathcal{O}\left(\frac{\beta_1^2}{\gamma^2 c_3} + \frac{\beta_1^2}{\gamma^2 c_2^2}\right) = \mathcal{O}\left(\frac{1}{\Delta^2\gamma^2 c_3} + \frac{1}{\Delta^2\gamma^2 c_2^2}\right) \qquad (35)$$

for bounded inputs.

Let us fix $c_2 = \frac{1}{44}$ for $\tau = 1, \ldots, \Gamma$. In view of our initialization strategy (16), setting $c_3 = \frac{1}{20}$ for $\tau = 1$ gives $\mathbb{E}\left[f(\mathbf{w}^1) - f^*\right] \le \frac{\beta_1\|\mathbf{r}_t\|^2}{2}$. Afterwards, we halve c_3 at each outer loop $\tau = 2, \ldots$, and this makes sure that $\mathbb{E}\left[f(\mathbf{w}^\tau) - f^*\right] \le \frac{\beta_1\|\mathbf{r}_t\|^2}{2^\tau}$.

To achieve the desired accuracy, we need $\Gamma = \mathcal{O}\left(\log\frac{1}{\eta_t}\right)$ outer iterations. Summing (34) and (34) over $\tau = 1, \ldots, \Gamma$, and noting $\sum_{\tau=1}^{\Gamma} 2^{\tau-1} = \mathcal{O}\left(\frac{1}{\eta_t}\right)$, the total sample complexity is

$$\mathcal{O}\left(\frac{d}{\Delta^2} \cdot 20\sum_{\tau=1}^{\Gamma} 2^{\tau-1} + \frac{44^2 d}{\Delta^2\gamma^2} \cdot \log\frac{1}{\eta_t}\right) = \mathcal{O}\left(\frac{d}{\Delta^2\eta_t} + \frac{d}{\Delta^2\gamma^2}\log\frac{1}{\eta_t}\right)$$

for sub-Gaussian/regular polynomial-tail inputs, and

$$\mathcal{O}\left(\frac{1}{\Delta^2\gamma^2}\left(20\sum_{\tau=1}^{\Gamma} 2^{\tau-1} + 44^2 \cdot \log\frac{1}{\eta_t}\right)\right) = \mathcal{O}\left(\frac{1}{\Delta^2\gamma^2\eta_t}\right)$$

for bounded inputs (we have dropped the second term since $\log\frac{1}{\eta_t}$ is of lower order compared with $\frac{1}{\eta_t}$). ■

E.4. Proof of Theorem 18

Proof Recall that our streaming CCA algorithm performs shift-and-invert power iterations on the population matrices directly. Following the same argument in the ERM case in Corollary 13, as long as each least squares objective is solved to sufficient accuracy, i.e.,

$$\frac{f_{t+1}(\mathbf{w}_{t+1}) - f_{t+1}^*}{\mathbf{w}_t^\top \widehat{\mathbf{B}} \mathbf{w}_t} \leq \min\left(\sum_{i=2}^d \xi_{ti}^2/\beta_i,\ \xi_{t1}^2/\beta_1\right) \cdot \frac{(\beta_1-\beta_2)^2}{32}, \tag{36}$$

the algorithm converges linearly, and therefore we only need to solve $T = \mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ linear systems. But due to the zero initialization we use in the online setting, the ratio between initial error and final error for each f_{t+1} is different from the offline setting.

When $G(\mathbf{r}_t) > 1$, we are in the regime where $\sum_{i=2}^d \xi_{ti}^2/\beta_i \geq \xi_{t1}^2/\beta_1$, and we can ensure the sufficient accuracy in (36) by setting the ratio between the initial and the final error to be

$$\eta_t = \frac{(\beta_1-\beta_2)^2\left(\xi_{t1}^2\right)}{16\beta_1^2}$$

in Lemma 17. Since $\frac{\beta_1^2}{(\beta_1-\beta_2)^2} \leq 4$, this implies that

$$\frac{1}{\eta_t} \leq \frac{64}{\cos^2\theta_t} = 64(1+\tan^2\theta_t) \leq 64\left(1+\frac{\beta_2}{\beta_1}G^2(\mathbf{r}_t)\right) \leq 64\left(1+G^2(\mathbf{r}_t)\right) \leq 64\left(1+G^2(\mathbf{r}_0)\right).$$

Note that the sample complexity of this phase does not depend on the final accuracy in alignment.

When $G(\mathbf{r}_t) \leq 1$, indicating that we are in the converging regime where $\sum_{i=2}^d \xi_{ti}^2/\beta_i \leq \xi_{t1}^2/\beta_1$, we can ensure the sufficient accuracy in (36) by setting

$$\eta_t = \frac{(\beta_1-\beta_2)^2\left(\sum_{i=2}^d \xi_{ti}^2\right)}{16\beta_1^2}$$

in Lemma 17. This implies that

$$\frac{1}{\eta_t} \leq \frac{64}{\sin^2\theta_t}.$$

Our goal is to have $\sin^2\theta_T \leq \frac{\epsilon}{4}$, as this implies $\cos\theta_T = \sqrt{1-\sin^2\theta_T} \geq 1-\sin^2\theta_T \geq 1-\frac{\epsilon}{4}$, and by Lemma 15 this further implies $\operatorname{align}((\mathbf{u}_T,\mathbf{v}_T);(\mathbf{u}^*,\mathbf{v}^*)) \geq 1-\epsilon$ as desired. Since $\sin^2\theta_t \leq G^2(\mathbf{r}_t)$, and we have shown that $G^2(\mathbf{r}_t)$ decreases at a geometric rate, we can bound $\frac{1}{\sin^2\theta_t}$ by a geometrically increasing series where the last term is $\frac{4}{\epsilon}$, and the sum of the truncated series up to time T is of the same order of the last term, i.e., $\sum_{t=1}^T \frac{1}{\eta_t} = \mathcal{O}\left(\frac{1}{\epsilon}\right)$.

And the theorem follows from Lemma 17, by summing the sample complexity of least squares problems over the outer shift-and-invert iterations.

We remark that to achieve the result with probability $1-\delta$, we require each least squares problem to be solved to the desired accuracy with failure probability $\delta/\log(1/\epsilon)$ (using the Markov inequality) and finally apply the union bound. This would only cause additional $\log(1/\epsilon)$ factors in the total sample complexity. ■

Algorithm 2 Non-uniform sampling SVRG for optimizing finite-sum of nonconvex components $F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w})$.

Input: Stepsize s.
  Initialize $\mathbf{w}_0 \in \mathbb{R}^d$.
  for $j = 1, 2, \ldots, M$ **do**
    $\tilde{\mathbf{u}} \leftarrow \mathbf{w}_{j-1}$
    Evaluate the batch gradient $\nabla F(\tilde{\mathbf{u}}) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{\mathbf{u}})$
    $\mathbf{u}_0 \leftarrow \mathbf{w}_{j-1}$
    for $t = 1, 2, \ldots, m$ **do**
      Randomly pick i_t from $\{1, \ldots, n\}$ with probability $\{p_i\}_{i=1}^n$.
      $\mathbf{u}_t \leftarrow \mathbf{u}_{t-1} - s\left(\frac{\nabla f_{i_t}(\mathbf{u}_{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{u}})}{p_{i_t} n} + \nabla F(\tilde{\mathbf{u}})\right)$
    end for
    $\mathbf{w}_j \leftarrow \frac{1}{n}\sum_{t=1}^m \mathbf{w}_t$
  end for
Output: $\mathbf{w}_M$ is the approximate solution.

Appendix F. SVRG with non-uniform sampling for finite-sum of nonconvex components

In this section, we show that for optimizing a convex objective that is the finite-sum of nonconvex components, sampling each components with probability proportional to its smoothness parameter, as shown in Algorithm 2, leads to improved convergence rate. In particular, the final time complexity depends on average smoothness parameter rather than the maximum smoothness.

Lemma 30 *Let $F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w})$, where $F(\mathbf{w})$ is μ-strongly convex, and each component $f_i(\mathbf{w})$ is L_i-smooth. Let $\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w})$. In the inner loop of Algorithm 2, sample i_t using weighted sampling probability $\{p_i\}_{i=1}^n$ from $\{1, \ldots, n\}$ where $p_i = \frac{L_i^2}{\sum_{j=1}^n L_j^2}$, and set $s = \frac{2\mu n}{11\sum_{i=1}^n L_i^2}$, $m = \frac{121\sum_{i=1}^n L_i^2}{8n\mu^2}$. Then the iteration complexity (number of vector operations) to reach ϵ-suboptimality is*

$$\mathcal{O}\left(\left(n + \frac{\sum_{i=1}^n L_i^2}{n\mu^2}\right)\log\frac{\left(\frac{1}{n}\sum_{i=1}^n L_i\right)\cdot\left(F(\mathbf{w}_0) - F(\mathbf{w}^*)\right)}{\mu\varepsilon}\right).$$

Proof For the inner loop of Algorithm 2, we are performing updates of the following form:

$$\mathbf{u}_t \leftarrow \mathbf{u}_{t-1} - s\mathbf{v}_t,$$

where

$$\mathbf{v}_t = \frac{\nabla f_{i_t}(\mathbf{u}_{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{u}})}{p_{i_t} n} + \nabla F(\tilde{\mathbf{u}}).$$

Taking expectation over the random choice of component i_t, we have

$$\mathbb{E}_t[\mathbf{v}_t] = \nabla F(\mathbf{u}_{t-1})$$

We now upper bound the variance of $\mathbf{v}_t$:

$$\begin{aligned}
\mathbb{E}_t \left\|\mathbf{v}_t - \nabla F(\mathbf{u}_{t-1})\right\|^2 =& \mathbb{E}_t \left[\frac{\nabla f_{i_t}(\mathbf{u}_{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{u}})}{p_{i_t} n} + \nabla F(\tilde{\mathbf{u}}) - \nabla F(\mathbf{u}_{t-1})\right]^2 \\
=& \mathbb{E}_t \left[\frac{1}{(p_{i_t} n)^2} \left\|\nabla f_{i_t}(\mathbf{u}_{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{u}})\right\|^2\right] - \left\|\nabla F(\mathbf{u}_{t-1}) - \nabla F(\tilde{\mathbf{u}})\right\|^2 \\
\leq& \frac{1}{n} \sum_{i=1}^n \frac{1}{p_i n} \left\|\nabla f_i(\mathbf{u}_{t-1}) - \nabla f_i(\tilde{\mathbf{u}})\right\|^2 \\
\leq& \frac{2}{n} \sum_{i=1}^n \frac{1}{p_i n} \left(\left\|\nabla f_i(\mathbf{u}_{t-1}) - \nabla f_i(\mathbf{w}^*)\right\|^2 + \left\|\nabla f_i(\tilde{\mathbf{u}}) - \nabla f_i(\mathbf{w}^*)\right\|^2\right) \\
\leq& \frac{2}{n} \sum_{i=1}^n \frac{L_i^2}{p_i n} \left(\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + \left\|\tilde{\mathbf{u}} - \mathbf{w}^*\right\|^2\right)
\end{aligned}$$

where we have used the fact that $\mathbb{E}\left\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\right\|^2 = \mathbb{E}\left\|\mathbf{x}\right\|^2 - (\mathbb{E}[\mathbf{x}])^2$ for a random vector $\mathbf{x}$ in the second equality, and that $\left\|\mathbf{x}+\mathbf{y}\right\|^2 \leq 2\left\|\mathbf{x}\right\|^2 + 2\left\|\mathbf{y}\right\|^2$ in the second inequality.

By choosing

$$p_i = \frac{L_i^2}{\sum_{i=1}^n L_i^2},$$

the above inequality becomes

$$\mathbb{E}_t \left\|\mathbf{v}_t - \nabla F(\mathbf{u}_{t-1})\right\|^2 \leq \frac{2\sum_{i=1}^n L_i^2}{n} \left(\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + \left\|\tilde{\mathbf{u}} - \mathbf{w}^*\right\|^2\right).$$

Define $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$ which is an upper bound of the smoothness parameter of the average function $F(\mathbf{w})$ as

$$\begin{aligned}
F(\mathbf{a}) - F(\mathbf{b}) &= \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{a}) - f_i(\mathbf{b}) \leq \frac{1}{n}\sum_{i=1}^n \langle \nabla f_i(\mathbf{b}), \mathbf{a}-\mathbf{b}\rangle + \frac{L_i}{2}\left\|\mathbf{a}-\mathbf{b}\right\|^2 \\
&= \langle \nabla F(\mathbf{b}), \mathbf{a}-\mathbf{b}\rangle + \frac{\frac{1}{n}\sum_{i=1}^n L_i}{2}\left\|\mathbf{a}-\mathbf{b}\right\|^2,
\end{aligned}$$

and define $\hat{L} = \sqrt{\frac{1}{n}\sum_{i=1}^n L_i^2}$. We then bound the distance from each iterate to the optimum:

$$\begin{aligned}
&\mathbb{E}_t \left\|\mathbf{u}_t - \mathbf{w}^*\right\|^2 = \left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 - 2s\left\langle \mathbf{u}_{t-1} - \mathbf{w}^*, \mathbb{E}_t\left[\mathbf{v}_t\right]\right\rangle + s^2 \mathbb{E}_t \left\|\mathbf{v}_t\right\|^2 \\
&\leq \left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 - 2s\left\langle \mathbf{u}_{t-1} - \mathbf{w}^*, \nabla F(\mathbf{u}_{t-1})\right\rangle + s^2\left\|\nabla F(\mathbf{u}_{t-1})\right\|^2 + 2s^2\hat{L}^2\left(\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + \left\|\tilde{\mathbf{u}} - \mathbf{w}^*\right\|^2\right) \\
&\leq \left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 - 2s\mu\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + s^2\bar{L}^2\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + 2s^2\hat{L}^2\left(\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + \left\|\tilde{\mathbf{u}} - \mathbf{w}^*\right\|^2\right)
\end{aligned}$$

where we have used the fact that $\mathbb{E}\left\|\mathbf{x}\right\| = (\mathbb{E}[\mathbf{x}])^2 + \mathbb{E}\left\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\right\|^2$ in the first inequality, and the smoothness and strong convexity of $F(\mathbf{w})$ in the second inequality.

By the Jensen's inequality, we have $\bar{L} \leq \hat{L}$. Therefore, we continue from above and obtain

$$\mathbb{E}_t\left[\left\|\mathbf{u}_t - \mathbf{w}^*\right\|^2\right] - \mathbb{E}\left[\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2\right] \leq \left(3s^2\hat{L}^2 - 2s\mu\right)\left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + 2s^2\hat{L}^2\left\|\tilde{\mathbf{u}} - \mathbf{w}^*\right\|^2.$$

Summing the above inequality over the inner loop yields

$$\mathbb{E}\left\|\mathbf{u}_m - \mathbf{w}^*\right\|^2 - \mathbb{E}\left\|\mathbf{u}_0 - \mathbf{w}^*\right\|^2 \leq \left(3s^2\hat{L}^2 - 2s\mu\right)\sum_{t=1}^m \left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 + 2ms^2\hat{L}^2\left\|\tilde{\mathbf{u}} - \mathbf{w}^*\right\|^2$$

Using $\mathbf{u}_0 = \tilde{\mathbf{u}}$ and rearranging terms, we have

$$\left(2s\mu - 3s^2\hat{L}^2\right)\sum_{t=1}^m \left\|\mathbf{u}_{t-1} - \mathbf{w}^*\right\|^2 \leq \left(1 + 2ms^2\hat{L}^2\right)\left\|\tilde{\mathbf{u}} - \mathbf{w}^*\right\|^2$$

Using $\tilde{\mathbf{u}} = \mathbf{w}_{j-1}$ and $\mathbf{w}_j = \frac{1}{m}\sum_{t=0}^{m-1}\mathbf{u}_t$, we obtain

$$\mathbb{E}\left\|\mathbf{w}_j - \mathbf{w}^*\right\|^2 \leq \frac{1 + 2ms^2\hat{L}^2}{2ms\mu - 3ms^2\hat{L}^2}\mathbb{E}\left\|\mathbf{w}_{j-1} - \mathbf{w}^*\right\|^2.$$

Setting

$$s = \frac{2\mu}{11\hat{L}^2}, \qquad m = \frac{1}{2s^2\hat{L}^2} = \frac{121\hat{L}^2}{8\mu^2},$$

we obtain

$$\mathbb{E}\left\|\mathbf{w}_j - \mathbf{w}^*\right\|^2 \leq \frac{1}{2}\mathbb{E}\left\|\mathbf{w}_{j-1} - \mathbf{w}^*\right\|^2.$$

Therefore the squared distance to minimum decreases geometrically for each outer loop. After M iterations, we have

$$F(\mathbf{w}_M) - F(\mathbf{w}^*) \leq \frac{\bar{L}}{2}\left\|\mathbf{w}_M - \mathbf{w}^*\right\|^2 \leq \frac{\bar{L}}{2}\left(\frac{1}{2}\right)^M\left\|\mathbf{w}_0 - \mathbf{w}^*\right\|^2 \leq \frac{\bar{L}}{\mu}\left(\frac{1}{2}\right)^M (F(\mathbf{w}_0) - F(\mathbf{w}^*)).$$

Setting the right hand side to ε gives the number of outer iterations $M = \mathcal{O}\left(\log\frac{(\bar{L}/\mu)\cdot(F(\mathbf{w}_0) - F(\mathbf{w}^*))}{\varepsilon}\right)$. Finally, the total iteration complexity to reach ε-suboptimality is

$$\mathcal{O}\left((n+m)M\right) = \mathcal{O}\left(\left(n + \frac{\hat{L}^2}{\mu^2}\right)\log\frac{\bar{L}(F(\mathbf{w}_0) - F(\mathbf{w}^*))}{\mu\varepsilon}\right).$$

■

Journal of Machine Learning Research 20 (2019) 1-70 Submitted 3/18; Revised 10/19; Published 11/19

Determinantal Point Processes for Coresets

Nicolas Tremblay NICOLAS.TREMBLAY@GIPSA-LAB.FR
Simon Barthelmé SIMON.BARTHELME@GIPSA-LAB.FR
Pierre-Olivier Amblard PIERRE-OLIVIER.AMBLARD@GIPSA-LAB.FR
CNRS, Univ. Grenoble Alpes, Grenoble INP, GIPSA-lab, Grenoble, France

Editor: Michael Mahoney

Abstract

When faced with a data set too large to be processed all at once, an obvious solution is to retain only part of it. In practice this takes a wide variety of different forms, and among them "coresets" are especially appealing. A coreset is a (small) weighted sample of the original data that comes with the following guarantee: a cost function can be evaluated on the smaller set instead of the larger one, with low relative error. For some classes of problems, and via a careful choice of sampling distribution (based on the so-called "sensitivity" metric), iid random sampling has turned to be one of the most successful methods for building coresets efficiently. However, independent samples are sometimes overly redundant, and one could hope that enforcing diversity would lead to better performance. The difficulty lies in proving coreset properties in non-iid samples. We show that the coreset property holds for samples formed with determinantal point processes (DPP). DPPs are interesting because they are a rare example of repulsive point processes with tractable theoretical properties, enabling us to prove general coreset theorems. We apply our results to both the k-means and the linear regression problems, and give extensive empirical evidence that the small additional computational cost of DPP sampling comes with superior performance over its iid counterpart. Of independent interest, we also provide analytical formulas for the sensitivity in the linear regression and 1-means cases.

Keywords: Coresets, Determinantal Point Processes, Sensitivity

1. Introduction

Given a learning task, if an algorithm is too slow on large data sets, one can either speed up the algorithm or reduce the amount of data. The theory of "coresets" gives theoretical guarantees on the latter option. A coreset is a weighted sub-sample of the original data, with the guarantee that for any learning parameter, the task's cost function estimated on the coreset is equal to the cost computed on the entire data set up to a controlled relative error.

An elegant consequence of such a property is that one may run learning algorithms solely on the coreset, allowing for a significant decrease in the computational cost while guaranteeing almost-equal performance. There are many algorithms that produce coresets, with some tailored for a specific task (such as k-means, k-medians, logistic regression, etc.), and others more generic. Also, there exists coreset sampling strategies both for the streaming setting and the offline setting: we choose here to focus on the offline setting. We

©2019 Nicolas Tremblay, Simon Barthelmé, Pierre-Olivier Amblard.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-167.html.

follow the review of Munteanu and Schwiegelshohn (2017) and classify coreset construction techniques in four categories:

1. Geometric decompositions (*e.g.,* Har-Peled and Mazumdar, 2004; Har-Peled and Kushal, 2005; Agarwal et al., 2005; Har-Peled, 2011). These methods propose to first discretize the ambient space of the data into a set of cells, snap each data point to its nearest cell in the discretization, and then use these weighted cells to approximate the target tasks. In all these constructions, the minimum number of samples required to guarantee the coreset property depends exponentially in the dimensionality of the ambient space, making them less useful in high-dimensional problems.

2. Gradient descent (*e.g.,* Badoiu and Clarkson, 2008; de la Vega et al., 2003; Kumar et al., 2010; Clarkson, 2010). These methods have been originally designed for the smallest enclosing ball problem (*i.e.*, finding the ball of minimum radius enclosing all datapoints), and have been later generalized to other problems. One of the main drawback of these algorithms in the k-means setting for instance is that their running time grows exponentially in the number of classes k (Kumar et al., 2010). Also, these algorithms provide only so-called *weak* coresets.

3. Random sampling (*e.g.,* Chen, 2009; Langberg and Schulman, 2010; Feldman and Langberg, 2011; Braverman et al., 2016; Bachem et al., 2017). The state of the art for many different tasks such as k-means or k-median is currently via iid random non-uniform sampling. For optimal performance, the probability to sample an element should be set proportional to a quantity known as its *sensitivity* (introduced by Langberg and Schulman (2010)). See Definition 2 for the formal definition of sensitivity. In practice, it is unpractical to compute sensitivities: state of the art algorithms rely on bi-criteria approximations to find upper bounds, and set the probability distribution proportional to this upper bound. More details on these results are provided in Section 2.4.

4. Sketching and projections (*e.g.,* Phillips, 2016; Woodruff, 2014; Mahoney, 2011; Boutsidis et al., 2015; Boutsidis and Gittens, 2013; Cohen et al., 2015; Keriven et al., 2017; Clarkson and Woodruff, 2017; Becchetti et al., 2019). Another direction of research regarding data reduction that provably keeps the relevant information for a given learning task is via sketches (Woodruff, 2014): compressed mappings (obtained via projections) of the original data set that are in general easy to update with new or modified data. Sketches are not strictly speaking coresets, and the difference resides in the fact that coresets are subsets of the data, whereas sketches are projections of the data. Note finally that the frontier between the two is permeable and some data summaries may combine both.

Our work falls into the random sampling category, in which the state of the art consists in tailoring a sampling distribution for the data set at hand, and then sampling iid from that distribution (Chen, 2009; Langberg and Schulman, 2010; Feldman and Langberg, 2011; Braverman et al., 2016). Independent processes being ignorant of the past, and thus liable to sample similar points repeatedly, an avenue for improvement is to produce samples that are less redundant than what iid sampling produces. A natural idea is to consider

negatively correlated point processes, *i.e.*, point processes for which sampling jointly two similar datapoints is less probable than sampling two very different datapoints. Methods based on negatively correlated sampling have been studied in the past for specific tasks.

For instance, for the column subset selection problem (CSSP), a method called volume sampling has been investigated in the literature (see Deshpande et al., 2006; Deshpande and Rademacher, 2010). A determinant-based sampling strategy has also been studied by Belabbas and Wolfe (2009). Also, a recent work (Belhadji et al., 2018) discusses with some details the different existing sampling-based methods (iid or with negative correlations) for the CSSP, and compares them versus a determinantal sampling strategy. Another specific task for which volume sampling strategies have been used is linear regression (see for instance Derezinski et al., 2018; Derezinski and Warmuth, 2018).

We propose: i/ to concentrate on a specific type of negatively-correlated sampling: determinantal point processes (DPPs), known to provide samples that are representative of the "diversity" in the data set (Kulesza and Taskar, 2012); ii/ to study their coreset performance on generic tasks. To the best of our knowledge, we provide the first general coreset guarantee using non-iid random sampling.

DPPs are parametrized by a positive semi-definite matrix called L-ensemble and denoted by L. This matrix encodes for the inclusion probabilities of each sample as well as higher order inclusion probabilities defining the correlation between samples. Note that DPPs have in general a random number of samples which in many practical situations is not adapted. This lead Kulesza and Taskar (2012) to define m-DPPs: DPPs conditioned to output m samples (for precise definitions related to DPPs and m-DPPs, see Section 2.6). It so happens that DPPs are more tractable than m-DPPs, making some proofs easier to show in the DPP context; however, m-DPPs are more useful in practice, especially when one needs to compare with fixed-size sampling methods as we do in this work. The reader should thus be mentally prepared to juggle from one concept to the other throughout the remainder of this paper.

1.1. Contributions

Theoretical contributions. Our theorems are quite generic, and assume mostly that the cost functions under study are Lipschitz. We have two main lines of argument: the first is that DPP samples do indeed verify the coreset property, the second is that DPPs produce *better* coresets than their iid counterparts if one uses the right L-ensemble to define the DPP. More specifically, we show:

- Theorem 8 and 20. Whatever the higher-order inclusion probabilities, if the inclusion probability of each sample of a DPP (or m-DPP) is set proportional to the sensitivity, then the results are at least as good as in the iid case. Technical limitations in controlling the concentration properties of correlated samples currently keep us from deriving exactly the minimum coreset size one may hope for when using DPPs.

- Theorem 11. A DPP sample necessarily has a lower variance than its (independent) Poisson counterpart with same inclusion probabilities.

- Theorem 12 and its Corollary 14. In the fixed-size context: a sample from an m-DPP with a rank m projective L-ensemble (also called projective DPP) necessarily leads to a lower variance than its iid counterpart with same inclusion probabilities.

We also show Theorem 16, stating that samples from a particular polynomial L-ensemble based on the Vandermonde matrix of the data asymptotically have a rebalancing property, made precise in Section 3.3. For instance in the k-means setting, this rebalancing property means that, asymptotically, such a DPP produces samples in each cluster, even if some are much smaller than others (see Figure 1 for an illustration).

Finally, of independent interest, we provide for the first time analytical formulas for the sensitivity, in two specific settings: the 1-means and the linear regression cases (Lemmas 23 and 25).

Empirical contributions. In the iid setting, for optimal performance, the probability of sampling an element should be set proportional to its sensitivity. In general, the sensitivity is not computable in polynomial time, thus out of reach in practice. For the specific 1-means and linear regression tasks, now that we have provided analytical formulas, these quantities become computable in polynomial time but turn out to be heavier to compute than solving the task on the whole data set –thus useless in practice. The usual workaround in the iid setting is to set the sampling probability proportional to an upper bound (efficiently computed via, *e.g.*, bi-criteria approximations) of the sensitivity. Thankfully, one still controls the performance of the obtained coreset (as a function of the upper bound's tightness).

Sensitivity playing a central role in the DPP-based coreset theorems we provide, these theorems also suffer from the same impracticality. Unfortunately, due to the dependencies introduced by DPPs, mere upper bounds of the sensitivity are not sufficient to propose a controlled workaround. The theorems enable to discuss in some detail what is the ideal task-specific choice of L-ensemble, but in practice we for now need to resort to heuristics.

We apply our results to both the k-means and the linear regression problems where the initial data consists in n points in $\mathbb{R}^d$. As explained, the ideal choice of L-ensemble L for DPP sampling is untractable in practice, we thus provide two efficient heuristics: one based on random Fourier features of the Gaussian kernel, the other on polynomial features. We pay particular attention to the computation cost of these two heuristics, and provide implementation details. These heuristics output a coreset sample in respectively $\mathcal{O}(nm^2 + nmd)$ and $\mathcal{O}(nm^2)$ time where m is the number of samples of the coreset. In the k-means context, this is to compare to $\mathcal{O}(nkd)$ the cost of the current state of the art iid sampling algorithm via bi-criteria approximation. m being necessarily larger than d and k to obtain the coreset guarantee in this context, our proposition is computationally heavier, especially as m increases. We provide nonetheless extensive empirical evidence showing that this additional cost stays reasonable, given the enhanced performance it provides. In particular, given that we provide analytical formulas for the sensitivities in the 1-means and linear regression contexts, we are able, in these two settings, to compare the DPP-based heuristics to the ideal iid coresets (*i.e.*, the coresets sampled iid from the distribution *exactly* proportional to the sensitivity): results clearly show the superior performance of our heuristics.

Finally, a Julia toolbox called DPP4Coresets is available on the authors' website.[1]

1.2. Organization of the paper

The paper is organized as follows. Section 2 recalls the background: the types of learning problems under consideration, the formal definition of coresets, sensitivities and DPPs. The theoretical Section 3 presents our main theorems on the performance of DPPs for coreset sampling: while Section 3.1 details coreset performance in the usual formulation of coreset theorems, Section 3.2 shows general variance arguments in favor of DPPs, and finally Section 3.3 provides an original asymptotic rebalancing property of DPPs. Section 4.1 shows how these theorems are applicable to both the k-means and the linear regression problems. We provide in Section 5 a discussion on the choice of L-ensemble adapted to these problems, and detail our sampling algorithms. Finally, the empirical Section 6 presents experiments on artificial as well as real-world data sets comparing the performance of DPP sampling to iid sampling. Section 7 concludes the paper. Note that for the sake of readability, many proofs and some implementation details are pushed to the Appendix.

2. Background

Let $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ be a set of n datapoints. Let (Θ, d_Θ) be a metric space of parameters, and θ an element of Θ. We consider cost functions of the form:

$$L(\mathcal{X}, \theta) = \sum_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}, \theta), \tag{1}$$

where f is a non-negative γ-Lipschitz function ($\gamma > 0$) with respect to θ, *i.e.*, $\forall \boldsymbol{x} \in \mathcal{X}$:

$$\begin{aligned} &\forall \theta \in \Theta && f(\boldsymbol{x}, \theta) \geq 0, \\ &\forall (\theta, \theta') \in \Theta^2 && |f(\boldsymbol{x}, \theta) - f(\boldsymbol{x}, \theta')| \leq \gamma \, d_\Theta(\theta, \theta'). \end{aligned}$$

Many classical machine learning cost functions fall under this model: k-means, k-median, logistic or linear regression, support-vector machines, low-rank approximations of matrices, etc.

2.1. Problem considered

A standard learning task is to minimize the cost L over all $\theta \in \Theta$. We write:

$$\theta^{\text{opt}} = \operatorname*{argmin}_{\theta \in \Theta} L(\mathcal{X}, \theta), \qquad L^{\text{opt}} = L(\mathcal{X}, \theta^{\text{opt}}) \qquad \text{and} \qquad \langle f \rangle_{\text{opt}} = \frac{L^{\text{opt}}}{n}. \tag{2}$$

In some instances of this problem, e.g., if n is very large and/or if f is expensive to evaluate and should be computed as rarely as possible, one may rely on sampling strategies to efficiently perform this optimization task.

1. The DPP4Coresets toolbox is also available at `https://gricad-gitlab.univ-grenoble-alpes.fr/tremblan/dpp4coresets.jl`.

2.2. Coresets

Let $\mathcal{S} = \{\boldsymbol{x}_{s_1}, \dots, \boldsymbol{x}_{s_m}\}$ be a subset of $\mathcal{X}$ (possibly with repetitions). To each element $\boldsymbol{x}_s \in \mathcal{S}$, associate a weight $\omega(\boldsymbol{x}_s) \in \mathbb{R}^+$. Define the estimated cost associated to the weighted subset $\mathcal{S}$ as:

$$\hat{L}(\mathcal{S}, \theta) = \sum_{\boldsymbol{x}_s \in \mathcal{S}} \omega(\boldsymbol{x}_s) f(\boldsymbol{x}_s, \theta). \tag{3}$$

Definition 1 (Coreset) *Let $\epsilon \in (0, 1)$. The weighted subset $\mathcal{S}$ is a ϵ-coreset for L if,* for any parameter θ, *the estimated cost is equal to the exact cost up to a relative error:*

$$\forall \theta \in \Theta \qquad \left| \frac{\hat{L}}{L} - 1 \right| \leq \epsilon. \tag{4}$$

This is the so-called "strong" coreset definition, as the ϵ-approximation is required for all $\theta \in \Theta$. A weaker version of this definition exists in the literature where the ϵ-approximation is only required for θ^{opt}. In the following, we focus on theorems guaranteeing the strong coreset property.

Let us write $\hat{\theta}^{\text{opt}}$ the optimal solution computed on the weighted subset $\mathcal{S}$: $\hat{\theta}^{\text{opt}} = \operatorname{argmin}_{\theta \in \Theta} \hat{L}(\mathcal{S}, \theta)$. An important consequence of the coreset property is the following:

$$(1-\epsilon) L(\mathcal{X}, \theta^{\text{opt}}) \leq (1-\epsilon) L(\mathcal{X}, \hat{\theta}^{\text{opt}}) \leq \hat{L}(\mathcal{S}, \hat{\theta}^{\text{opt}}) \leq \hat{L}(\mathcal{S}, \theta^{\text{opt}}) \leq (1+\epsilon) L(\mathcal{X}, \theta^{\text{opt}}),$$

i.e., running an optimization algorithm on the weighted sample $\mathcal{S}$ will result in a minimal learning cost that is a controlled ϵ-approximation of the learning cost one would have obtained by running the same algorithm on the entire data set $\mathcal{X}$. Note that the guarantee is over costs only: the estimated optimal parameters $\hat{\theta}^{\text{opt}}$ and θ^{opt} may be different. Nevertheless, if the cost function is well suited to the problem: either there is one clear global minimum and the estimated parameters will almost coincide; or there are multiple solutions for which the learning cost is similar and selecting one over the other is not an issue.

In terms of computation cost, if the sampling scheme is efficient, n is very large and/or f is expensive to compute for each datapoint, coresets thus enable a significant gain in computing time.

2.3. Sensitivity

To define appropriate sampling schemes for coresets, Langberg and Schulman (2010) introduce the notion of sensitivity:

Definition 2 (Sensitivity) *The sensitivity of a datapoint $\boldsymbol{x}_i \in \mathcal{X}$ with respect to a fuction $f : \mathcal{X}, \Theta \to \mathbb{R}^+$ is:*

$$\sigma_i = \max_{\theta \in \Theta} \frac{f(\boldsymbol{x}_i, \theta)}{L(\mathcal{X}, \theta)} \qquad \in [0, 1]. \tag{5}$$

Also, the total sensitivity is defined as :

$$\mathfrak{S} = \sum_{i=1}^{n} \sigma_i.$$

Note that the fraction defining the sensitivity is not defined for $L(\mathcal{X}, \theta) = 0$ (that may happen for instance in the 1-means problem, in the degenerate case where all $\boldsymbol{x}_i$ are superimposed and equal to θ). For simplicity, we suppose that $\forall \theta \in \Theta,\ L(\mathcal{X}, \theta) > 0$.

The sensitivity is related to the concept of statistical leverage score (*e.g.*, Drineas and Mahoney, 2018; Drineas et al., 2012), which plays a crucial role in iid random sampling theorems in the randomized numerical linear algebra literature (Mahoney, 2011). Both notions are similar, but not equivalent. For instance, we show in Lemma 25 that sensitivities for the linear regression task are different from the usual definition of leverage score in this context. Thus, in general, leverage scores used in the randomized linear algebra literature are not sensitivities, *i.e.*, they do not necessarily verify Eq. (5).

In words, the sensitivity σ_i is the worse case contribution of datapoint x_i in the total cost. Informally, the larger it is, the larger its "outlierness" (Lucic et al., 2016).

2.4. iid importance sampling and state of the art results

In the iid sampling paradigm, the importance sampling estimator of L is the following. Say the sample set $\mathcal{S}$ consists in m samples drawn iid with replacement from a (discrete) probability distribution $\boldsymbol{p} \in \mathbb{R}^n$ (with p_i the probability of sampling $\boldsymbol{x}_i$ at each draw, and $\sum_i p_i = 1$). Denote by ϵ_i the random variable counting the number of occurences of $\boldsymbol{x}_i$ in $\mathcal{S}$. One may define $\hat{L}_{\text{iid}}$, the so-called importance sampling estimator of L, as :

$$\hat{L}_{\text{iid}}(\mathcal{S}, \theta) = \sum_i \frac{f(\boldsymbol{x}_i, \theta)\epsilon_i}{m p_i}. \tag{6}$$

One can show that $\mathbb{E}(\epsilon_i) = m p_i$, such that $\hat{L}_{\text{iid}}$ is an unbiased estimator of L:

$$\mathbb{E}(\hat{L}_{\text{iid}}(\mathcal{S}, \theta)) = L(\mathcal{X}, \theta).$$

The concentration of $\hat{L}_{\text{iid}}$ around its expected value is controlled by the following state of the art theorem:

Theorem 3 (Coresets with iid random sampling) *Let $\boldsymbol{p} \in [0,1]^n$ be a probability distribution over all datapoints $\mathcal{X}$ with p_i the probability of sampling $\boldsymbol{x}_i$ and $\sum_i p_i = 1$. Draw m iid samples with replacement according to $\boldsymbol{p}$. Associate to each sample $\boldsymbol{x}_s$ a weight $\omega(\boldsymbol{x}_s) = 1/m p_s$. The weighted subset obtained is a ϵ-coreset with probability $1-\delta$ provided that:*

$$m \geq m^*$$

with

$$m^* = \mathcal{O}\left(\frac{1}{\epsilon^2}\left(\max_i \frac{\sigma_i}{p_i}\right)^2 (d' + \log(1/\delta))\right),$$

where d' is the pseudo-dimension of Θ (a generalization of the Vapnik-Chervonenkis dimension). The optimal probability distribution minimizing m^* is $p_i = \sigma_i / \mathfrak{S}$. In this case, the weighted subset is a ϵ-coreset with probability $1-\delta$ provided that:

$$m \geq \mathcal{O}\left(\frac{\mathfrak{S}^2}{\epsilon^2}(d' + \log(1/\delta))\right).$$

For instance, in the k-means setting[2]*, $d' = dk \log k$ and $\mathfrak{S} = \mathcal{O}(k)$ such that the coreset property is guaranteed with probability $1-\delta$ provided that:*

$$m \geq \mathcal{O}\left(\frac{k^2}{\epsilon^2}(dk \log k + \log(1/\delta))\right).$$

This theorem is taken from the paper by Bachem et al. (2017). Its original form goes back to Langberg and Schulman (2010). Note that sensitivities cannot be computed rapidly, such that, as it is, this theorem is unpractical. Thankfully, bi-criteria approximation schemes (such as Algorithm 2 of Bachem et al. 2017, or other propositions such as in Feldman and Langberg 2011; Makarychev et al. 2016) may be used to efficiently find an upper bound of the sensitivity for all i: $b_i \geq \sigma_i$. Noting $B = \sum b_i$, and setting $p_i = b_i/B$, one shows that the coreset property may be guaranteed in the iid framework provided that $m \geq \mathcal{O}\left(\frac{B^2}{\epsilon^2}(d' + \log(1/\delta))\right)$.

Note that if one authorizes coresets with negative weights (that is, authorizes negative weights in the estimated cost of Eq. (3)), then the above theorem may be further improved (Feldman and Langberg, 2011). Nevertheless, we prefer to restrict ourselves to positive weights as optimization algorithms such as Lloyd's k-means heuristics (Lloyd, 1982) are in practice more straightforward to implement on positively weighted sets rather than on sets with possibly negative weights.

Finally, Braverman et al. (2016, Theorem 5.5) improve the previous theorem by showing that under the same non-uniform iid framework, the coreset property is guaranteed provided that $m \geq \mathcal{O}\left(\frac{\mathfrak{S}}{\epsilon^2}(d' \log \mathfrak{S} + \log(1/\delta))\right)$, thus reducing the term in $\mathfrak{S}^2$ to $\mathfrak{S} \log \mathfrak{S}$.

2.5. Correlated importance sampling

Eq. (6) is not suited to correlated sampling and, in the following, we will use a slightly different importance sampling estimator, more adapted to this case. Consider a point process defined on $\mathcal{X}$ that outputs a random sample $\mathcal{S} \subset \mathcal{X}$. For each data point $\boldsymbol{x}_i$, denote by π_i its inclusion (or marginal) probability:

$$\pi_i = \mathbb{P}(\boldsymbol{x}_i \in \mathcal{S}). \tag{7}$$

Moreover, denote by ϵ_i the random Boolean variable such that $\epsilon_i = 1$ if $\boldsymbol{x}_i \in \mathcal{S}$, and 0 otherwise. In this paper, we focus on the following definition[3] of the importance sampling cost estimator $\hat{L}$:

$$\hat{L}(\mathcal{S}, \theta) = \sum_i \frac{f(\boldsymbol{x}_i, \theta)\epsilon_i}{\pi_i}. \tag{8}$$

By construction, $\mathbb{E}(\epsilon_i) = \pi_i$, such that $\hat{L}$ is an unbiased estimator of L:

$$\mathbb{E}(\hat{L}(\mathcal{S}, \theta)) = L(\mathcal{X}, \theta).$$

2. In the literature (Feldman and Langberg, 2011; Balcan et al., 2013), d' is often taken to be equal to dk in the k-means setting. We nevertheless agree with Bachem et al. (2017) and their discussion in Section 2.6 regarding k-means' pseudo-dimension and thus write $d' = dk \log k$

3. Note that in fact $\hat{L}_{\text{iid}}$ and $\hat{L}$ are the same objects if one defines ϵ_i to be the number of times i is sampled (which will be in practice Boolean in the DPP case as the same sample can never be sampled twice in this context) and write $\hat{L}(\mathcal{S}, \theta) = \sum_i \frac{f(\boldsymbol{x}_i, \theta)\epsilon_i}{\mathbb{E}(\epsilon_i)}$. We prefer to introduce both notations to avoid confusions.

Studying the coreset property in this setting boils down to studying the concentration properties of $\hat{L}$ around its expected value.

2.6. Determinantal Point Processes

In order to induce negative correlations within the samples, we choose to focus on Determinantal Point Processes (DPP), point processes that have recently gained attention due to their ability to output "diverse" subsets within a tractable probabilistic framework (for instance with explicit formulas for marginal probabilities). In the following, $2^{[n]}$ denotes the set of all possible subsets of the n first integers.

The central object is called the L-ensemble, and is nothing else than a positive semi-definite matrix $\mathsf{L} \in \mathbb{R}^{n\times n}$. We will write its eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

Definition 4 (DPP, Kulesza and Taskar 2012) *Consider a point process, i.e., a process that randomly draws an element $\mathcal{S} \in 2^{[n]}$. It is determinantal with L-ensemble L if*

$$\mathbb{P}(\mathcal{S}) = \frac{\det(\mathsf{L}_{\mathcal{S}})}{\det(\mathsf{I}+\mathsf{L})},$$

where $\mathsf{L}_{\mathcal{S}}$ is the restriction of L to the rows and columns indexed by the elements of $\mathcal{S}$.

The following well-known properties are verified (see Kulesza and Taskar (2012) for details):

- one can indeed show that the normalization is proper: $\sum_{\mathcal{S}} \det(\mathsf{L}_{\mathcal{S}}) = \det(\mathsf{I}+\mathsf{L})$.
- all inclusion probabilities, at any order, are explicit:

$$\forall \mathcal{A} \in 2^{[n]} \qquad \mathbb{P}(\mathcal{A} \subseteq \mathcal{S}) = \det(\mathsf{K}_{\mathcal{A}})$$

  where $\mathsf{K} = \mathsf{L}(\mathsf{I}+\mathsf{L})^{-1} \in \mathbb{R}^{n\times n}$ is called the marginal kernel. In particular, the probability of inclusion of i, π_i, is equal to K_{ii}. Also, to gain insight in the repulsive nature of DPPs, one may readily see that the joint marginal probability of sampling i and j reads: $\det(\mathsf{K}_{\{i,j\}}) = \pi_i\pi_j - \mathsf{K}_{ij}^2$ and is necessarily smaller than $\pi_i\pi_j$, the joint probability in the case of Poisson uncorrelated sampling. The stronger the "interaction" between i and j (encoded by the absolute value of element K_{ij}), the smaller the probability of sampling both jointly: this determinantal nature thus favors diverse sets of samples.
- K is also positive semi-definite. The eigenvalues of K are $\{\frac{\lambda_i}{1+\lambda_i}\}_i$ and are necessarily between 0 and 1.
- it can be shown that the number of samples of a DPP is itself random and distributed as a sum of Bernoulli parametrized by the eigenvalues of K. In particular, the expected number of samples is $\mu = \mathrm{Tr}(\mathsf{K}) = \sum_i \frac{\lambda_i}{1+\lambda_i}$.

In many cases, one prefers to specify deterministically the number of samples, instead of having a random number of them. This leads to m-DPPs: DPPs conditioned to output m samples.

Definition 5 (m-DPP, Kulesza and Taskar 2012) *Consider a point process that randomly draws an element $\mathcal{S} \in 2^{[n]}$. This process is an m-DPP with L-ensemble L if:*

i) $\forall \mathcal{S}$ *s.t.* $|\mathcal{S}| \neq m, \quad \mathbb{P}(\mathcal{S}) = 0$

ii) $\forall \mathcal{S}$ *s.t.* $|\mathcal{S}| = m, \quad \mathbb{P}(\mathcal{S}) = \frac{1}{Z}\det(\mathsf{L}_{\mathcal{S}})$ *with Z the normalization constant.*

The following properties hold:

- the normalization constant Z is in fact the m-th order elementary symmetric polynomial of the eigenvalues of L:

$$Z = \sum_{\mathcal{S}' \text{ s.t. } |\mathcal{S}'|=m} \det(\mathsf{L}_{\mathcal{S}'}) \;=\; e_m(\lambda_1,\ldots,\lambda_n) = \sum_{1\leq j_1<j_2<\cdots<j_m\leq n} \lambda_{j_1}\cdots\lambda_{j_m}.$$

- in general, m-DPPs are not DPPs: for instance the probability of including element i, π_i, is no longer K_{ii} in general. In fact, one has $\pi_i = \frac{1}{Z}\sum_{\mathcal{S}' \text{ s.t } |\mathcal{S}'|=m \text{ and } i\in\mathcal{S}'} \det(\mathsf{L}_{\mathcal{S}'})$.
- by construction, $\sum_i \pi_i = m$.

Let us define the specific but important case of projective DPPs.

Definition 6 (projective-DPP) *A projective DPP is a m-DPP whose L-ensemble is a projection of rank m:*

$$\mathsf{L} = \mathsf{U}\mathsf{U}^\top$$

where $\mathsf{U} \in \mathbb{R}^{n\times m}$ has orthonormal columns (i.e., $\mathsf{U}^\top\mathsf{U} = \mathsf{I}_m$).

Lemma 7 (Lemma 1.3 of Barthelmé et al. 2019) *A projective DPP with L-ensemble L is also a DPP, with* marginal kernel L.

In fact, the set of projective DPPs is precisely the intersection between the set of DPPs and the set of m-DPPs. Projective DPPs are very practical objects: they have both the practical convenience of m-DPPs (a fixed number of samples) and the theoretical convenience of DPPs (for instance, π_i is simply L_{ii}, *i.e.*, the sum of squares of the i-th line of U).

3. Coreset theorems

We now detail our main theoretical contributions. In Section 3.1, we present a coreset theorem for m-DPPs providing sufficient conditions on the marginal probabilities $\{\pi_i\}_i$ to guarantee the coreset property. We will see that, similar to the iid case (Theorem 3), the optimal marginal probability should be set proportional to the sensitivity. A similar result is derived for DPPs in Appendix B. These theorems are valid for any choice of higher order inclusion probabilities (the conditions are only on the first-order inclusion probabilities $\{\pi_i\}$). We further discuss in Section 3.2 how one may take advantage of these additional degrees of freedom encoding the negative correlations of DPPs to improve the coreset performance over iid sampling. Finally, in Section 3.3, we show that a particular polynomial projective DPP asymptotically verifes a rebalancing property, thus making them natural candidates for the coreset problem.

3.1. m-Determinantal Point Processes for coresets

Theorem 8 (m-DPP for coresets) *Let $\mathcal{S}$ be a sample from an m-DPP with L-ensemble L, $\epsilon \in (0,1)$, and η the minimal number of balls of radius $\frac{\epsilon \langle f \rangle_{opt}}{6\gamma}$ necessary to cover Θ, with γ the Lipschitz parameter of f. $\mathcal{S}$ is a ϵ-coreset with probability larger than $1-\delta$ provided that:*

$$m \geq m^* = \max(m_1^*, m_2^*)$$

with:

$$m_1^* = \frac{32}{\epsilon^2}\left(\max_i \frac{\sigma_i}{\bar{\pi}_i}\right)^2 \log\frac{4\eta}{\delta},$$
$$m_2^* = \frac{32}{\epsilon^2}\left(\frac{1}{n\bar{\pi}_{min}}\right)^2 \log\frac{4\eta}{\delta},$$

and $\forall i, \bar{\pi}_i = \pi_i/m$.

The proof is provided in Appendix A. Note that m_1^* and m_2^* are not independent of m: they are in fact dependent via $\bar{\pi}_i = \pi_i/m$. While this formulation may be surprising at first, this is due to the fact that in non-iid settings, separating m from π_i is not as straightforward as in the iid case (in Theorem 3, m and p_i are independent) . Also, we give this particular formulation of the theorem to mimic classical concentration results obtained with iid sampling.

In order to simplify further analysis, we suppose from now on that $n\sigma_{\min} \geq 1$. As shown in the second lemma of Appendix D, this is in fact verified in the k-means case for instance. Nevertheless, the following results may be generalized to cases with unconstrained $\sigma_{\min}$ if needed, with little effects on the main results.

Corollary 9 *If $n\sigma_{min} \geq 1$, then $m_1^* \geq m_2^*$ and the coreset property of Theorem 8 is verified if:*

$$m \geq m^* = \frac{32}{\epsilon^2}\left(\max_i \frac{\sigma_i}{\bar{\pi}_i}\right)^2 \log\frac{4\eta}{\delta} \tag{9}$$

with $\forall i,\ \bar{\pi}_i = \pi_i/m$.

Proof Denote by j the index for which $\bar{\pi}_i$ is minimal and, provided that $n\sigma_{\min} \geq 1$, one has:

$$\max_i \frac{\sigma_i}{\bar{\pi}_i} n\bar{\pi}_{\min} \geq n\sigma_j \geq n\sigma_{\min} \geq 1,$$

which implies $m^* = \max(m_1^*, m_2^*) = m_1^*$. ■

One would like to have the coreset guarantee for a minimal number of samples, that is: to find the marginal probabilities π_i minimizing m^*. A quick glance at Eq. (9) tells us to set $\pi_i = m\sigma_i/\mathfrak{S}$ in order to minimize the bound m^* while satisfying the constraint $\sum_i \pi_i = m$. In practice, however, computing the sensitivities is often untractable. We thus propose to set the marginal probabilities according to the following looser condition.

Corollary 10 *If one sets the π_i's such that there exists $\alpha > 0$ and $\beta \geq 1$ verifying:*

$$\forall i \qquad \alpha\sigma_i \leq \pi_i \leq \alpha\beta\sigma_i, \tag{10}$$
$$\text{and} \qquad \frac{\alpha}{\beta} \geq \frac{32}{\epsilon^2}\mathfrak{S}\log\frac{4\eta}{\delta}, \tag{11}$$

then $\mathcal{S}$ is a ϵ-coreset with probability at least $1-\delta$. In this case, the number of samples verifies:

$$m \geq \frac{32}{\epsilon^2}\beta\mathfrak{S}^2\log\frac{4\eta}{\delta}.$$

Proof Let us suppose that the marginal probabilities π_i are set such that there exists $\alpha > 0$ and $\beta \geq 1$ verifying:

$$\forall i, \qquad \alpha\sigma_i \leq \pi_i \leq \alpha\beta\sigma_i.$$

Note that:

$$\left(\max_i \frac{\sigma_i}{\pi_i}\right)^2 m \leq \frac{m}{\alpha^2} = \frac{1}{\alpha^2}\sum_i \pi_i \leq \frac{\beta}{\alpha}\sum_i \sigma_i = \frac{\beta}{\alpha}\mathfrak{S}.$$

Thus, the inequality

$$\frac{\alpha}{\beta} \geq \frac{32}{\epsilon^2}\mathfrak{S}\log\frac{4\eta}{\delta}$$

implies:

$$1 \geq \frac{32}{\epsilon^2}\left(\max_i \frac{\sigma_i}{\pi_i}\right)^2 m\log\frac{4\eta}{\delta},$$

that we recognize as the coreset condition (9) by multiplying on both sides by m: $\mathcal{S}$ is indeed a ϵ-coreset with probability larger than $1-\delta$. Moreover, in this case:

$$m = \sum_i \pi_i \geq \alpha\sum_i \sigma_i = \alpha\mathfrak{S} \geq \frac{32}{\epsilon^2}\beta\mathfrak{S}^2\log\frac{4\eta}{\delta}.$$

■

Corollary 10 is applicable to cases where $\sigma_{\max}$ is not too large. In fact, in order for $\alpha\sigma_i$ to be smaller than π_i, and thus smaller than 1 as π_i is a probability, α should always be set inferior to $\frac{1}{\sigma_{\max}}$. Now, if $\sigma_{\max}$ is so large that $\frac{1}{\sigma_{\max}} \leq \frac{32}{\epsilon^2}\mathfrak{S}\log\frac{4\eta}{\delta}$, then, even by setting β to its minimum value 1, there is no admissible α verifying both conditions (10) and (11). We refer to Appendix E for a simple workaround if this issue arises. We will further see in the experimental section (Section 6) that elements with large sensitivities (*i.e.*, outliers, Lucic et al., 2016) are not an issue in practice.

Similar results are obtained for DPPs (instead of m-DPPs) in Appendix B.

3.2. Links with the iid case and variance arguments

Let us first compare these results with Theorem 3 obtained in the iid setting. A few remarks are in order:

1. setting β and α to 1 in Corollary 10, that is, setting each π_i exactly to σ_i, the minimum number of required samples is $\frac{32\mathfrak{S}^2}{\epsilon^2}(\log\eta+\log\frac{4}{\delta})$, to compare to $\mathcal{O}(\frac{\mathfrak{S}^2}{\epsilon^2}(d'+\log(1/\delta)))$ of Theorem 3, where d' is the pseudo-dimension of Θ. η being the number of balls of radius $\frac{\epsilon\langle f\rangle_{\text{opt}}}{6\gamma}$ necessary to cover Θ, it will typically be $\frac{\epsilon\langle f\rangle_{\text{opt}}}{6\gamma}$ to the power of the ambient dimension of Θ (analogous to d'). For instance, in the k-means case, $d'=dk\log k$ (see footnote 2), whereas, as shown later in Section 4.1, $\log\eta = dk\log\left(\frac{12\rho\gamma}{\epsilon\langle f\rangle_{\text{opt}}}+1\right)$ where ρ is the diameter of the minimum enclosing ball of the data $\mathcal{X}$. Up to the log term, d' and $\log\eta$ are the same. The difference observed in the log term is due to the fact that coreset theorems in the iid case (see for instance Bachem et al., 2017) take advantage of powerful results from the Vapnik-Chervonenkis (VC) theory, as detailed in Li et al. (2001). Unfortunately, these fundamental results are valid in the iid case only, and are not easily generalized to the correlated case. Possible improvements to reduce this small gap could take advantage of chaining arguments in correlated contexts such as in Baraud (2010), in order to improve over the repeated loose union bounds we have used in the proof.

2. Outliers are not naturally dealt with using our proof techniques, mainly due to our multiple use of the union bound that necessarily englobes the worse-case scenario. In fact, in the importance sampling estimator used in the iid case (Eq. 6), outliers are not problematic as they can be sampled several times. In our setting, outliers are constrained to be sampled only once, which in itself makes sense, but complicates the analysis. Empirically, we will see in Section 6 that outliers are not an issue.

3. The DPP coreset theorems obtained are in a sense disappointing: they do not show that the concentration is tighter in the DPP case than in the iid case. They are in fact limited by the current state-of-the-art in concentration of strongly Rayleigh measures (Pemantle and Peres, 2014). On the bright side, our results take *only* into account first-order inclusion probabilities: the $\{\pi_i\}$'s; meaning that these DPP sampling theorems are valid for any choice of higher-order inclusion probabilities (encoding the correlation between samples). We will now see how these extra degrees of freedom enable to provably decrease the variance of the cost estimator, compared to the iid case.

3.2.1. A first variance argument: improvement over the Poisson point process

Consider a DPP with marginal kernel K. Build the diagonal kernel K_d with $\mathsf{K}_d(i,i)=\mathsf{K}(i,i)$. Note that a DPP from K_d reduces to a Poisson point process. Note also that marginal probabilities π_i of both processes (and consequently their expected number of samples) are the same. We compare the variance of the estimator $\hat{L}$ obtained with a DPP with marginal kernel K versus its variance obtained with its Poisson uncorrelated counterpart: a DPP with marginal kernel K_d.

Theorem 11 *For any admissible marginal kernel* K *(i.e., positive semi-definite with eigenvalues between* 0 *and* 1*), we have:*

$$\forall\theta\in\Theta \qquad \mathrm{Var}(\hat{L}) = \mathrm{Var}_d - \sum_{i\neq j}\frac{\mathsf{K}_{ij}^2}{\pi_i\pi_j}f(\boldsymbol{x}_i,\theta)f(\boldsymbol{x}_j,\theta)$$

where Var_d *is the variance of the estimator based on the diagonal DPP. As the function* f *is positive, the variance of* $\hat{L}$ *via DPP sampling with kernel* K *is thus necessarily smaller than its Poisson counterpart with same inclusion probabilities.*

Proof We have:

$$\begin{aligned}\mathrm{Var}(\hat{L}) &= \mathbb{E}(\hat{L}^2)-\mathbb{E}(\hat{L})^2\\ &= \sum_{i,j}\frac{\mathbb{E}(\epsilon_i\epsilon_j)}{\pi_i\pi_j}f(\boldsymbol{x}_i,\theta)f(\boldsymbol{x}_j,\theta)-L^2.\end{aligned}$$

As $\mathcal{S}$ is sampled from a DPP, the following is verified. If $i\neq j$, $\mathbb{E}(\epsilon_i\epsilon_j)=\det(\mathsf{K}_{\{i,j\}})=\pi_i\pi_j-\mathsf{K}_{ij}^2$. If $i=j$, $\mathbb{E}(\epsilon_i\epsilon_j)=\mathbb{E}(\epsilon_i)=\pi_i$. One obtains:

$$\mathrm{Var}(\hat{L}) = \sum_i\left(\frac{1}{\pi_i}-1\right)f(x_i,\theta)^2-\sum_{i\neq j}\frac{\mathsf{K}_{ij}^2}{\pi_i\pi_j}f(\boldsymbol{x}_i,\theta)f(\boldsymbol{x}_j,\theta). \tag{12}$$

The first term of the right-hand side is in fact the variance in the case of a diagonal kernel: $\sum_i\left(\frac{1}{\pi_i}-1\right)f(\boldsymbol{x}_i,\theta)^2=\mathrm{Var}_d$, finishing the proof. ■

The important message here is that this variance reduction occurs *regardless* of the choice of K's off-diagonal elements: any choice –provided that $0\preceq\mathsf{K}\preceq 1$ stays true– will reduce the variance.

Proving such a variance reduction when comparing a m-DPP with L-ensemble L versus its conditional Poisson equivalent (a Poisson point process conditioned to m samples, with same $\{\pi_i\}$) is much more involved, and remains open.

3.2.2. A second variance argument: improvement over the iid estimator with replacement

We now compare the variance of the iid estimator with replacement $\hat{L}_{\text{iid}}$ of Eq. (6) and the variance of the DPP estimator $\hat{L}$ of Eq. (8). Consider a DPP with marginal kernel K, with $\forall i \quad \pi_i=\mathsf{K}_{ii}$ the marginal probability of sampling element i such that the expected number of samples $\mu=\sum_i\pi_i$ is an integer. We compare the variance of $\hat{L}$ with such a DPP and the variance of $\hat{L}_{\text{iid}}$ with μ independent draws with replacement with $p_i=\pi_i/\mu$ (in order to have a fair comparison).

Before we state the result, suppose that K is of rank r (with, necessarily, $\mu\leq r\leq n$). K being positive-semi definite and of rank r, there exists $\mathsf{V}=(\boldsymbol{v}_1|\boldsymbol{v}_2|\ldots|\boldsymbol{v}_n)\in\mathbb{R}^{r\times n}$ a set of n vectors in dimension r such that $\mathsf{K}=\mathsf{V}^\top\mathsf{V}$. By construction, $\forall i \quad \|\boldsymbol{v}_i\|^2=\mathsf{K}_{ii}=\pi_i$. For

each vector $\boldsymbol{v}$, consider its diagram vector (Copenhaver et al., 2014, Definition 2.3), denoted $\tilde{\boldsymbol{v}}$, defined as:

$$\tilde{\boldsymbol{v}} = \frac{1}{\sqrt{r-1}} \begin{bmatrix} v(1)^2 - v(2)^2 \\ \vdots \\ v(r-1)^2 - v(r)^2 \\ \sqrt{2r}\, v(1)v(2) \\ \vdots \\ \sqrt{2r}\, v(r-1)v(r) \end{bmatrix} \in \mathbb{R}^{r(r-1)}, \tag{13}$$

where the difference of squares $v(i)^2 - v(j)^2$ and the product $v(i)v(j)$ occur exactly once for $i < j, i = 1, 2, \cdots, r-1$.

Theorem 12 *One has:*

$$\mathrm{Var}(\hat{L}) = \mathrm{Var}(\hat{L}_{\mathrm{iid}}) + \left(\frac{1}{\mu} - \frac{1}{r}\right) L^2 - \frac{r-1}{r} \left\| \sum_i \frac{f(\boldsymbol{x}_i, \theta)}{\pi_i} \tilde{\boldsymbol{v}}_i \right\|^2.$$

Proof In the iid case,

$$\mathbb{E}(\hat{L}_{\mathrm{iid}}^2) = \sum_{i=1}^n \sum_{j=1}^n \frac{f(\boldsymbol{x}_i, \theta) f(\boldsymbol{x}_j, \theta) \mathbb{E}(\epsilon_i \epsilon_j)}{\mu^2 p_i p_j}$$

where ϵ_i is not Boolean but counts the number of times i is sampled. One can show that if $i \neq j$, $\mathbb{E}(\epsilon_i \epsilon_j) = p_i p_j (\mu^2 - \mu)$, and if $i = j$, $\mathbb{E}(\epsilon_i \epsilon_j) = p_i \mu + p_i^2 \mu^2 - \mu p_i^2$. Thus:

$$\begin{aligned} \mathrm{Var}(\hat{L}_{\mathrm{iid}}) &= \mathbb{E}(\hat{L}_{\mathrm{iid}}^2) - L^2 \\ &= \sum_{i=1}^n \sum_{j \neq i} f(x_i, \theta) f(x_j, \theta)(1 - 1/\mu) + \sum_{i=1}^n f(x_i, \theta)^2 \frac{1 + p_i \mu - p_i}{\mu p_i} - L^2 \\ &= \frac{1}{\mu} \sum_{i=1}^n \frac{f(\boldsymbol{x}_i, \theta)^2}{p_i} - \frac{1}{\mu} L^2 \end{aligned}$$

Moreover:

$$\mathrm{Var}(\hat{L}) = \sum_i \frac{f(x_i, \theta)^2}{\pi_i} - \sum_i \sum_j \frac{f(\boldsymbol{x}_i, \theta) f(\boldsymbol{x}_j, \theta)}{\pi_i \pi_j} \mathsf{K}_{ij}^2.$$

Thus:

$$\mathrm{Var}(\hat{L}) = \mathrm{Var}(\hat{L}_{\mathrm{iid}}) + \frac{1}{\mu} L^2 - \sum_i \sum_j \frac{f(\boldsymbol{x}_i, \theta) f(\boldsymbol{x}_j, \theta)}{\pi_i \pi_j} \mathsf{K}_{ij}^2 \tag{14}$$

Proposition 2.5 of Copenhaver et al. (2014) states:

$$\begin{aligned} \forall (i,j) \qquad \mathsf{K}_{ij}^2 &= \left(\boldsymbol{v}_i^\top \boldsymbol{v}_j\right)^2 = \frac{1}{r} \|\boldsymbol{v}_i\|^2 \|\boldsymbol{v}_j\|^2 + \frac{r-1}{r} \tilde{\boldsymbol{v}}_i^\top \tilde{\boldsymbol{v}}_j \\ &= \frac{1}{r} \pi_i \pi_j + \frac{r-1}{r} \tilde{\boldsymbol{v}}_i^\top \tilde{\boldsymbol{v}}_j. \end{aligned}$$

Replacing this in Eq. (14) yields the desired result. ■

Remark 13 *The variance of the DPP estimator is partly due to the fact that the number of samples is random, which is not the case with the iid scheme we compare it to. The following corollary compares variances when the number of samples is fixed, i.e., in the case where the DPP is projective.*

Corollary 14 *The marginal kernel of a projective DPP with a (fixed) number of samples μ is, by definition, of rank $r = \mu$. In this case:*

$$\forall \theta \in \Theta \qquad \mathrm{Var}(\hat{L}) = \mathrm{Var}(\hat{L}_{\mathrm{iid}}) - \frac{\mu - 1}{\mu} \left\| \sum_i \frac{f(\boldsymbol{x}_i, \theta)}{\pi_i} \tilde{\boldsymbol{v}}_i \right\|^2. \tag{15}$$

The variance is thus necessarily improved when using a projective DPP compared to its iid counterpart. This result is remarkable: the variance reduction is independent of the sign of f (supposed positive in the coreset context). This opens interesting generalizing perspectives to a more general class of cost functions L.

3.2.3. A link with tight frames

In order to design the ideal marginal kernel K, and according to the previous discussion, one wants K to verify:

- The previous corollary suggests to design a projective DPP, that is: $\mathsf{K} = \mathsf{V}^\top \mathsf{V}$ with $\mathsf{V}\mathsf{V}^\top = \mathsf{I}_m$.
- Theorem 8 suggests to set $\pi_i = \mathsf{K}_{ii} = \frac{m\sigma_i}{\mathfrak{S}}$.

Finding such a marginal kernel boils down to finding $\mathsf{V} = (\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_n)$ a set of n vectors $\boldsymbol{v}_i$ in dimension m with specified norms $\|\boldsymbol{v}_i\|^2 = \pi_i$, such that $\sum_i \pi_i = m$ and $\mathsf{V}\mathsf{V}^\top = \mathsf{I}_m$. This is exactly the problem of finding a tight frame of n vectors in dimension m, with specified norms (Casazza and Kutyniok, 2012).

Lemma 15 *Such a tight frame exists.*

Proof Let us denote by $\pi_{(i)}$ the non-decreasing ordered sequence of π_i: $\pi_{(1)} \leq \pi_{(2)} \leq \ldots \leq \pi_{(n)}$. The Schur-Horn theorem states that a hermitian matrix K of size $n \times n$ with diagonal entries π_i and eigenvalues $(0, \ldots, 0, 1, \ldots, 1)$ with $n - m$ zeros and m ones, exists if $\pi_{(i)}$ majorizes $(0, \ldots, 0, 1, \ldots, 1)$, that is, if all the following inequalities are simultaneously verified:

$$\pi_{(1)} \geq 0, \qquad \pi_{(1)} + \pi_{(2)} \geq 0, \qquad \cdots, \qquad \sum_{i=1}^{n-m} \pi_{(i)} \geq 0$$

$$\sum_{i=1}^{n-m+1} \pi_{(i)} \geq 1, \qquad \cdots, \qquad \sum_{i=1}^{n-1} \pi_{(i)} \geq m - 1, \qquad \sum_{i=1}^{n} \pi_{(i)} \geq m.$$

The first $n-m$ inequalities are trivially verified as all π_i are supposed positive. Now, $\sum_{i=1}^{n-m+1} \pi_{(i)} \geq 1$ is also verified. Indeed, if it was not case, *i.e.*, if $\sum_{i=1}^{n-m+1} \pi_{(i)} < 1$, then $\sum_{i=1}^{n} \pi_{(i)} < m$ as the largest $m-1$ values of π_i are by hypothesis upper bounded by 1. This would contradict $\sum_i \pi_i = m$. A similar argument can be applied to the remaining inequalities. ■

Also, a tight frame not only exists, but several solutions exist in general, and efficient algorithms have been designed to build one (see for instance Tropp et al., 2004). Out of all these possibilities, the ideal would be to find the tight frame that minimizes the variance of Eq.(15). Up to our knowledge, this is an open and difficult question, rooted in frame theory.

Let us recap the above variance results. We showed that a DPP sampling scheme has necessarily a lower variance than its Poisson counterpart, *regardless of the choice of off-diagonal elements of* K, provided that K stays PSD with eigenvalues between 0 and 1. We also showed that a projective DPP sampling scheme has necessarily a lower variance than its iid counterpart *regardless of the choice of off-diagonal elements of* K, provided that K stays projective. We finally showed that finding the projective DPP that minimizes the variance is equivalent to a difficult problem in frame theory. In other words: finding the optimal DPP for a given problem and data set may be very hard, but on the other hand *any* DPP is guaranteed to do at least as well as iid sampling, in the sense discussed above. Further, we can easily design DPPs which are not optimal, but still have valuable properties, as the next section shows.

3.3. DPPs provide balanced sampling: a new type of guarantee

An important insight of coreset theory is that the datapoints which are different from the rest should be kept in the sample. We show in this section that one can construct a DPP which asymptotically guarantees a rebalancing of the datapoints $\mathcal{X}$, meaning that points which are relatively isolated have a high chance of being retained. For instance, in the k-means setting, this property implies that, asymptotically, one can construct a DPP that provably produces a balanced sample across clusters, even in data sets where some clusters are much smaller than others. The result is illustrated in Figs. 1 and 2.

In a nutshell, the result is as follows. Suppose that the data $\mathcal{X}$ is a set of n elements drawn iid from a continuous distribution μ defined on $\Omega \subset \mathbb{R}^d$. Build a projective DPP $\mathcal{S}$ of size m based on the monomials of the x_i's (see Section 3.3.1 for a precise definition). Under mild regularity assumptions on μ, we show that the intensity measure of $\mathcal{S}$, marginalized over $\mathcal{X}$ is *independent* of μ. Our proof is based on a powerful theorem from Kroo and Lubinsky (2013).

Note that this rebalancing property also occurs for iid sampling with sensitivities (that provide a sort of density estimation: the lower the density of points around x_i, the larger σ_i, the higher the chance of sampling it). What is noteworthy here is that the rebalancing property occurs "naturally": without any sort of prior density-like estimation. We will emphasize this important point at the end of this Section.

In Section 3.3.1, we present the specific type of polynomial DPPs for which our proof holds, that are similar to those used by Bardenet and Hardy (2016). Our result is then formally stated in Section 3.3.2.

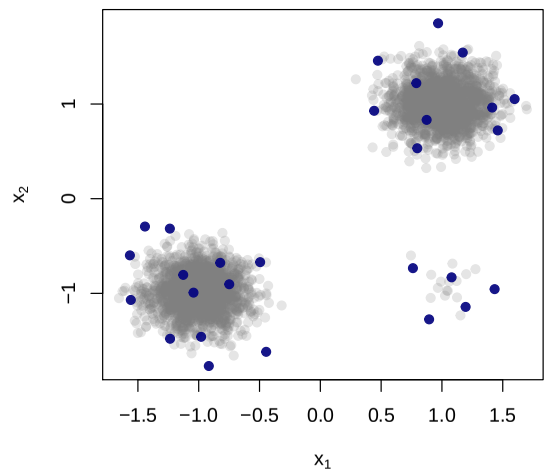


Figure 1: It is possible to construct a DPP with a (asymptotic) rebalancing property, meaning that it will sample several points from each cluster even when clusters are severely imbalanced. Here, we show three imbalanced clusters: two have size 2,000 and one has size 20. In blue, a sample from a polynomial DPP (see text for definition): it samples from each cluster despite their very different sizes. The formal rebalancing property is illustrated in Figure 2.

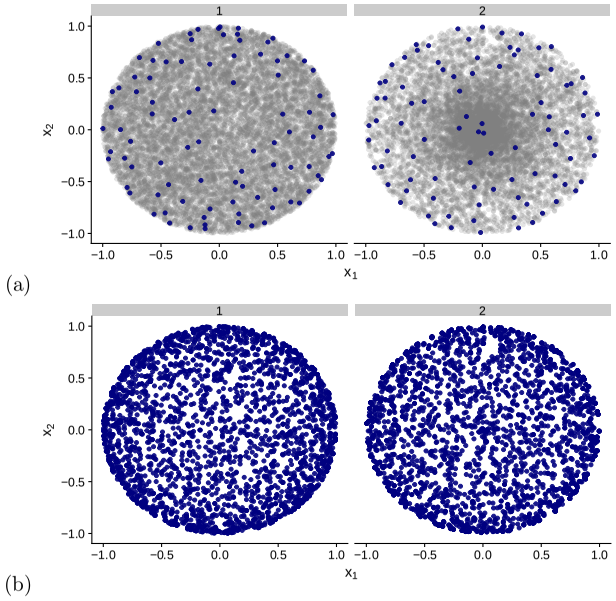


Figure 2: An illustration of the rebalancing result. (a) We sample ground sets (grey points) from two different distributions on the disc. In blue, two realisations of a polynomial DPP constructed from the ground set. Note how similar the two realisations are (density-wise), despite the very different ground sets they are drawn from. (b) We overlay 30 realisations of each DPP: the two resulting densities are very similar, again despite the different ground sets. Our result states that they should indeed converge in large n and m, and that the limiting density depends only on the shape of the domain. Here points close to the boundary are oversampled relative to points in the center, as predicted.

3.3.1. Projective polynomial DPPs

The L-ensemble we shall build is based on the first m monomials. In dimension one this is easy to define, so we start there and generalize later to dimension $d \geq 2$. For $d = 1$, we denote by $\mathcal{X} = \{x_1, \ldots, x_n\}$ the original set (supposed to be drawn iid from μ defined on Ω), and form the $n \times m$ Vandermonde matrix

$$\mathsf{V}(\mathcal{X}) = [x_i^{j-1}]_{i,j=1}^{n,m} \in \mathbb{R}^{n \times m}. \tag{16}$$

Note that this matrix has rank m a.s. (as μ is supposed regular enough) and contains all monomials up to degree $m-1$. The L-ensemble we consider equals:

$$\mathsf{L} = \mathsf{V}\mathsf{V}^\top \in \mathbb{R}^{n \times n}. \tag{17}$$

The orthogonal polynomials (defined on Ω) under the empirical measure $d\mu_n = (1/n)\sum \delta_{x_i}$ associated to $\mathcal{X}$, are defined in the usual manner, i.e. $q_0(x)$ of degree 0, $q_1(x)$ of degree 1, ... such that: $\int q_i(x)q_j(x)d\mu_n = \delta_{ij}$ and $\int x^i q_j(x) d\mu_n = 0$ if $i < j$. In other words, the sequence is constructed from Gram-Schmidt orthogonalisation under $d\mu_n$. Let us write $\boldsymbol{q}_j(\mathcal{X}) = (q_j(x_1), \ldots, q_j(x_n))^\top \in \mathbb{R}^n$ the vector consisting of the polynomial $q_j(x)$ taken at values in $\mathcal{X}$. It is well-known (and easily verified) that the QR decomposition of V verifies:

$$\mathsf{V} = \mathsf{Q}\mathsf{R} \tag{18}$$

with $\mathsf{Q} = (\boldsymbol{q}_0(\mathcal{X}) | \ldots | \boldsymbol{q}_{m-1}(\mathcal{X})) \in \mathbb{R}^{n \times m}$ and $\mathsf{R} \in \mathbb{R}^{m \times m}$ an upper triangular matrix.

Now, consider the m-DPP $\mathcal{S}$ with L-ensemble $\mathsf{L} = \mathsf{V}\mathsf{V}^\top$. Using the fact that $\det(AB) = \det(A)\det(B)$ if A and B are square, we have:

$$\begin{aligned}
\mathbb{P}(\mathcal{S}) &= Z^{-1} \det(\mathsf{L}_\mathcal{S}) \\
&= Z^{-1} \det((\mathsf{Q}\mathsf{R}\mathsf{R}^\top\mathsf{Q}^\top)_\mathcal{S}) \\
&= Z^{-1} \det((\mathsf{Q}\mathsf{Q}^\top)_\mathcal{S}) \det(\mathsf{R})^2 \\
&= Z'^{-1} \det((\mathsf{Q}\mathsf{Q}^\top)_\mathcal{S})
\end{aligned}$$

such that $\mathcal{S}$ is also a m-DPP with L-ensemble $\mathsf{Q}\mathsf{Q}^\top$. As $\mathsf{Q}^\top\mathsf{Q} = \mathsf{I}_m$, *i.e.*, $\mathsf{Q}\mathsf{Q}^\top$ is projective, $\mathcal{S}$ is in fact a projective DPP. As a result, its associated marginal kernel is also $\mathsf{Q}\mathsf{Q}^\top$ (see Lemma. 7) and, for instance:

$$\mathbb{P}(x_i \in \mathcal{S}) = \sum_{j=0}^{m-1} q_j^2(x_i).$$

The extension to $d > 1$ is mostly straightforward, but there are a few differences to keep in mind when defining the Vandermonde matrix of monomials. Monomials $\boldsymbol{x}^{\boldsymbol{\alpha}}$ are now defined as:

$$\boldsymbol{x}^{\boldsymbol{\alpha}} = \prod_{j=1}^{d} x(j)^{\alpha_j}$$

The total degree of a monomial equals the sum of the degrees in each variable, i.e. $\sum \alpha_i = ||\boldsymbol{\alpha}||_1$. The most significant difference between the one-dimensional case and the general

case is that there is more than one monomial of total degree ϕ. For example, in dimension 2, $\boldsymbol{x} = (x(1), x(2))^\top$ and the monomials of degree 2 are given by the powers $(2,0)$, $(0,2)$ and $(1,1)$: $x(1)^2, x(2)^2, x(1)x(2)$. A good way of thinking about the construction of a polynomial DPP in the multidimensional case is to pick first a maximum order (e.g. $\phi = 3$), meaning that all monomials with total degree up to 3 are included. Then the natural sample size m for the DPP equals the total number of features, giving $m = \binom{d+\phi}{\phi}$. Again, for $d = 2$ and $\phi = 3$, this gives $m = 1 + 2 + 3 + 4 = 10$. In fact, there is one monomial of order 0: 1, two monomials of order 1: $x(1)$ and $x(2)$, three monomials of order 2 (the ones stated above), and four monomials of order 3: $x(1)^3$, $x(2)^3$, $x(1)^2x(2)$ and $x(1)x(2)^2$. This implies that in dimension d, the m-DPP detailed earlier is well-defined only for specific values of m: $m = \binom{d+1}{1} = d+1$, or $m = \binom{d+2}{2} = \frac{1}{2}(d+1)(d+2)$, or $m = \binom{d+3}{3} = \frac{1}{6}(d+1)(d+2)(d+3)$, etc.

A slight technical difficulty arises in defining the orthogonal polynomials of a multivariate measure: in dimension 1, the fact that there is a single monomial of a given degree leads to a natural order in which to perform the Gram-Schmidt procedure. In higher dimensions the order is only a partial order, so that we can introduce the monomials by blocks of equal degree, but within a block the ordering is arbitrary. So we may pick any arbitrary order (e.g. lexicographic) and run Gram-Schmidt in that order (for more, see Dunkl and Xu, 2014). Given this choice the QR decomposition remains well-defined and all properties given above in the 1D case carry over to the general case. In particular, the link with the orthogonal polynomials on the discrete measure μ_n stays valid.

3.3.2. The rebalancing theorem

Formally, the result is as follows. The intensity function $\iota(\boldsymbol{x})$ of a point process quantifies the expected number of points to be found around $\boldsymbol{x}$. We characterize the asymptotics of the intensity function of a DPP $\mathcal{S}$ when both $\mathcal{S}$ and the ground set $\mathcal{X}$ are large, and show that, in that limit, the intensity is *independent* of the measure μ from which $\mathcal{X}$ is sampled from.

The result is stated formally as a double limit, letting first n go to infinity (an easy discrete-to-continuous limit), and then increasing the order ϕ of the polynomial DPP, which implies m going to infinity too. We emphasize that, empirically speaking, rebalancing occurs for reasonable values of n and m but the rate of convergence is hard to quantify.

Certain regularity assumptions are inherited from the work of Kroo and Lubinsky (2013), to which we refer for more thorough details. The formal assumptions are as follows:

1. The initial data set $\mathcal{X} = \{x_1, \ldots, x_n\}$ is drawn i.i.d. from a measure μ over a compact, convex[4] domain $\Omega \subset \mathbb{R}^d$.
2. μ and the Lebesgue measure ν are mutually absolutely continuous on Ω, so that μ', the density, is well-defined everywhere on the domain (we use the Lebesgue measure for simplicity, another measure may be substituted)
3. We are interested in convergence "in the bulk", ie. inside the domain. Formally, the results hold for $D \subset D_1 \subset \Omega$, where D is compact and D_1 is open

4. The convexity assumption can probably be relaxed.

4. μ' is bounded above and below on D_1
5. We form a m-DPP $\mathcal{S}$ on the set $\mathcal{X}$, with a polynomial kernel of degree ϕ (defined in the previous Section), such that $m = \binom{\phi+d}{\phi}$.
6. (technical) μ is regular in the sense of Stahl, Totik, and Ullman, and the Christoffel function with respect to μ verifies condition (1.7) in Kroo and Lubinsky (2013).

The intensity measure of $\mathcal{S}$, marginalizing over $\mathcal{X}$, which we denote by $I_{n,\phi}(\mathcal{A})$ equals the expected number of points of $\mathcal{S}$ in set $\mathcal{A}$, i.e.:

$$I_{n,\phi}(\mathcal{A}) = \mathbb{E}_{\mathcal{X},\mathcal{S}}\left(|\mathcal{S} \cap \mathcal{A}|\right) = \mathbb{E}_{\mathcal{X},\mathcal{S}}\left\{\sum_{s \in \mathcal{S}} \mathbb{I}\left(s \in \mathcal{A}\right)\right\} \tag{19}$$

Note that the expectation is over *both* $\mathcal{X}$ and $\mathcal{S}$. Furthermore, $I_{n,\phi}(\Omega)$ equals m, the total number of points in $\mathcal{S}$.

Our result may be stated as follows.

Theorem 16 *Under the assumptions above, for all $\mathcal{A} \subset D_1$,*

$$\lim_{\phi \to \infty} \frac{1}{\binom{\phi+d}{\phi}} \lim_{n \to \infty} I_{n,\phi}(\mathcal{A}) = \int_{\mathcal{A}} \kappa(\boldsymbol{y}) d\boldsymbol{y}$$

where κ is a density independent *of μ*

The proof is in Appendix C.

Lemma 17 *κ is mostly dependent on the distance to the boundaries of Ω. For example, if Ω is the unit ball in $\mathbb{R}^d$, $\kappa(\boldsymbol{y}) = \left(1 - ||\boldsymbol{y}||^2\right)^{-1/2}$*

See Kroo and Lubinsky (2013) for a proof.

Several important remarks are in order:

- unlike iid sampling with sensitivities or other density-related measure for which such rebalancing property will also occur, there is here no prior density estimation: the L-ensemble is defined via the Vandermonde matrix that is trivial to compute. Thus, this rebalancing is a property that "naturally" arises from the DPP.
- this is only an asymptotic result as n and m go to infinity. Finding minimal values of m for which rebalancing is highly probable, or even rates of convergence is likely a difficult endeavour. We emphasize nevertheless that, empirically speaking, rebalancing occurs for reasonable values of n and m, as visible in Figs. 1 and 2.

This ends the theoretical results of this paper. We now move on to applications. In the next Section, we apply the results to two problems: k-means and linear regression. In Section 5, implementation details are provided. Finally, experimental validation on artificial and real-world data sets is provided in Section 6.

4. Application to two problems: k-means and linear regression

We focus on two problems: k-means and linear regression. Admittedly, these are not the best problems to exhibit the usefulness of coresets: there already exists very efficient algorithms to solve them and the need for a small controlled summary is in fact rare. We nevertheless focus on these two problems as they have been well studied in the iid setting, which it is our goal to improve on. Moreover, we derived analytical formulas for the sensitivity in the 1-means and the linear regression settings: we will thus be able to compare, in those two cases, DPP sampling vs the ideal iid setting (later in the experimental Section 6).

4.1. Application to k-means

The theoretical results of Section 3 are valid for any learning problem of the form detailed in Section 2.1. We now specifically consider the k-means problem on a set $\mathcal{X}$ comprised of n datapoints in $\mathbb{R}^d$. This problem boils down to finding k centroids $\theta = (c_1, \dots, c_k)$, all in $\mathbb{R}^d$, such that the following cost is minimized:

$$L(\mathcal{X}, \theta) = \sum_{x \in \mathcal{X}} f(x, \theta) \quad \text{with} \quad f(x, \theta) = \min_{c \in \theta} \|x - c\|^2 .$$

Let ρ be the diameter of the minimum enclosing ball of $\mathcal{X}$ (the smallest ball enclosing all points in $\mathcal{X}$). Theorem 8 and its corollaries are applicable to the k-means problem, such that:

Corollary 18 (m-DPP for k-means) *Let $\mathcal{S}$ be a sample from an m-DPP with L-ensemble L. Let $\epsilon, \delta \in (0,1)^2$. With probability at least $1-\delta$, $\mathcal{S}$ is a ϵ-coreset provided that:*

$$m \geq m^* = \frac{32}{\epsilon^2} \left(\max_i \frac{\sigma_i}{\bar{\pi}_i} \right)^2 \left(kd \log \left(\frac{24\rho^2}{\epsilon \langle f \rangle_{opt}} + 1 \right) + \log \frac{4}{\delta} \right),$$

with $\forall i, \bar{\pi}_i = \pi_i / m$.

Setting the marginal probabilities to their optimal values $\pi_i = m\sigma_i / \mathfrak{S}$, $\mathcal{S}$ is a ϵ-coreset with probability larger than $1-\delta$ provided that:

$$m \geq \frac{32}{\epsilon^2} \mathfrak{S}^2 \left(kd \log \left(\frac{24\rho^2}{\epsilon \langle f \rangle_{opt}} + 1 \right) + \log \frac{4}{\delta} \right).$$

Proof Let us write $\mathcal{B}$ the minimum enclosing ball of $\mathcal{X}$, of diameter ρ. The potentially interesting centroids are necessarily included in $\mathcal{B}$ such that the space of parameters Θ in the k-means setting is the set of all possible k centroids in $\mathcal{B}$: $\Theta = \mathcal{B}^k$. The metric d_Θ we consider is the Hausdorff metric associated with the Euclidean distance:

$$\forall \theta, \theta', \qquad d_\Theta(\theta, \theta') = \max \left\{ \max_{c \in \theta} \min_{c' \in \theta'} \left\| c - c' \right\|_2 , \quad \max_{c' \in \theta'} \min_{c \in \theta} \left\| c - c' \right\|_2 \right\}.$$

An ϵ'-net of Θ. Consider $\Gamma_\mathcal{B}$ an ϵ'-net of $\mathcal{B}$ consisting of $(\frac{2\rho}{\epsilon'} + 1)^d$ small balls of radius ϵ'. Such a covering indeed exists: see, *e.g.*, Lemma 2.5 in Geer (2000). Consider $\Gamma = \Gamma_\mathcal{B}^k$ of cardinality $|\Gamma| = (\frac{2\rho}{\epsilon'} + 1)^{kd}$. Let us show that Γ is an ϵ'-net of Θ, that is:

$$\forall \theta \in \mathcal{B}^k, \quad \exists \theta^* \in \Gamma \quad \text{s.t.} \quad d_\Theta(\theta, \theta^*) \leq \epsilon'.$$

In fact, consider $\theta = (c_1, \dots, c_k) \in \mathcal{B}^k$. By construction, as $\Gamma_\mathcal{B}$ is an ϵ'-net of $\mathcal{B}$, we have:

$$\forall i = 1, \dots, k \qquad \exists c_i^* \in \Gamma_\mathcal{B} \quad \text{s.t.} \quad \|c_i - c_i^*\| \leq \epsilon'.$$

Writing $\theta^* = (c_1^*, \dots, c_k^*) \in \Gamma$, one has:

$$d_\Theta(\theta, \theta^*) \leq \epsilon',$$

which proves that Γ is an ϵ'-net of Θ. The number of balls of radius $\epsilon' = \epsilon \langle f \rangle_{\text{opt}} / 6\gamma$ sufficient to cover Θ is thus $\eta = (\frac{12\rho\gamma}{\epsilon \langle f \rangle_{\text{opt}}} + 1)^{kd}$.

$f(x, \theta)$ is γ-Lipschitz with $\gamma = 2\rho$. Consider any θ, θ' and $x \in \mathcal{X}$. We want to show that:

$$-\gamma \; d_\Theta(\theta, \theta') \leq f(x, \theta) - f(x, \theta') \leq \gamma \; d_\Theta(\theta, \theta').$$

Let us write $c = \text{argmin}_{t \in \theta} \|x - t\|^2$ the centroid in θ closest to x and $c' = \text{argmin}_{t' \in \theta'} \|x - t'\|^2$ the centroid in θ' closest to x. Moreover, let us write $\tilde{c}' = \text{argmin}_{t' \in \theta'} \|c - t'\|^2$ the centroid in θ' closest to c. Note that c' and $\tilde{c}'$ are not necessarily equal. By definition of c', one has:

$$\left\| x - c' \right\| \leq \left\| x - \tilde{c}' \right\|,$$

such that:

$$\left\| x - c' \right\|_2 - \|x - c\|_2 \leq \left\| x - \tilde{c}' \right\|_2 - \|x - c\|_2 \leq \left\| \tilde{c}' - c \right\|_2 \leq d_\Theta(\theta, \theta').$$

Thus:

$$\begin{aligned} f(x, \theta') - f(x, \theta) = \left\| x - c'(x) \right\|^2 - \|x - c\|^2 &= (\left\| x - c' \right\| - \|x - c\|)(\left\| x - c' \right\| + \|x - c\|) \\ &\leq (\left\| x - c' \right\| + \|x - c\|) \; d_\Theta(\theta, \theta') \leq 2\rho \; d_\Theta(\theta, \theta'). \end{aligned}$$

Finally, $n\sigma_{\min} \geq 1$, as shown by the second lemma of Appendix D.

Given all these elements, Theorem 8 and its subsequent corollaries are thus applicable to the k-means setting and one obtains the desired result. ■

Note that, in the case of DPPs, one could apply Theorem 20 to the k-means problem, and obtain similar results.

4.2. Application to linear regression

We now consider the linear regression problem: find $\theta \in \mathbb{R}^d$ such that a measured vector $y \in \mathbb{R}^n$ is closest to $\mathsf{X}\theta$ where $\mathsf{X}^\top = (x_1 | \dots | x_n) \in \mathbb{R}^{d \times n}$ are n data points in $\mathbb{R}^d$. Let us write $X_i = (y_i, x_i)$ and $\mathcal{X} = \{X_1, \dots, X_n\}$. The least squares estimator minimizes:

$$L(\mathcal{X}, \theta) = \|y - \mathsf{X}\theta\|_2^2 .$$

By denoting

$$f(X_i, \theta) = (y_i - x_i^\top \theta)^2,$$

one can thus write the least squares solution to the linear regression problem in the form of the problems investigated in this paper: the objective is to minimize the cost L with f a positive function:

$$L(\mathcal{X},\theta) = \|y - \mathsf{X}\theta\|_2^2 = \sum_{i=1}^{n} f(X_i,\theta).$$

We suppose that all $\boldsymbol{x}_i$ are enclosed in the unit ball in dimension d and that $y_i \in [0,1]$. Moreover, we suppose that the space Θ is bounded and enclosed in a d-dimensional ball $\mathcal{B}$ centered in 0 of diameter ρ.

Even though we derived the analytical formulation of the sensitivity for linear regression (Lemma 25), we were not able to show that $n\sigma_{\min} \geq 1$ in general. We thus have the following slightly more complicated result:

Corollary 19 (m-DPP for linear regression) *Let $\mathcal{S}$ be a sample from an m-DPP with L-ensemble L. Let $\epsilon, \delta \in (0,1)^2$. With probability at least $1-\delta$, $\mathcal{S}$ is a ϵ-coreset provided that:*

$$m \geq \max(m_1^*, m_2^*)$$

with

$$m_1^* = \frac{32}{\epsilon^2}\left(\max_i \frac{\sigma_i}{\bar{\pi}_i}\right)^2 \left(d\log\left(\frac{12\rho(4\rho+2)}{\epsilon\langle f\rangle_{opt}}+1\right)+\log\frac{4}{\delta}\right),$$
$$m_2^* = \frac{32}{\epsilon^2}\left(\frac{1}{n\bar{\pi}_{min}}\right)^2 \left(d\log\left(\frac{12\rho(4\rho+2)}{\epsilon\langle f\rangle_{opt}}+1\right)+\log\frac{4}{\delta}\right)$$

with $\forall i, \bar{\pi}_i = \pi_i/m$.

Setting the marginal probabilities to their optimal values $\pi_i = m\sigma_i/\mathfrak{S}$, $\mathcal{S}$ is a ϵ-coreset with probability larger than $1-\delta$ provided that:

$$m \geq \frac{32}{\epsilon^2}\max\left(\mathfrak{S}^2, \frac{\mathfrak{S}^2}{n^2\sigma_{min}^2}\right)\left(d\log\left(\frac{12\rho(4\rho+2)}{\epsilon\langle f\rangle_{opt}}+1\right)+\log\frac{4}{\delta}\right).$$

Proof The metric d_Θ we consider is the Euclidean distance in dimension d.
- An ϵ'-net of Θ. Consider $\Gamma_{\mathcal{B}}$ an ϵ'-net of $\mathcal{B}$ consisting of $(\frac{2\rho}{\epsilon'}+1)^d$ small balls of radius ϵ'. Such a covering indeed exists: see, *e.g.*, Lemma 2.5 in Geer (2000). The number of balls of radius $\epsilon' = \epsilon\langle f\rangle_{\text{opt}}/6\gamma$ sufficient to cover Θ is thus $\eta = (\frac{12\rho\gamma}{\epsilon\langle f\rangle_{\text{opt}}}+1)^d$.

- $f(X,\theta)$ is γ-Lipschitz with $\gamma = 4\rho+2$. Consider any θ, θ', $\boldsymbol{x}_i$ and y_i. We want to show that:

$$\left((y_i - \boldsymbol{x}_i^\top\theta)^2 - (y_i - \boldsymbol{x}_i^\top\theta')^2\right)^2 \leq \gamma^2 \ \|\theta-\theta'\|^2.$$

In fact:

$$\begin{aligned}
\left((y_i - \boldsymbol{x}_i^\top\theta)^2 - (y_i - \boldsymbol{x}_i^\top\theta')^2\right)^2 &= \left(\theta^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top\theta - \theta'^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top\theta' - 2y_i\boldsymbol{x}_i^\top(\theta-\theta')\right)^2 \\
&= \left[\left(2\theta'^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top + (\theta-\theta')^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top - 2y_i\boldsymbol{x}_i^\top\right)(\theta-\theta')\right]^2 \\
&\leq \left[\left\|2\theta'^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top + (\theta-\theta')^\top\boldsymbol{x}_i\boldsymbol{x}_i^\top - 2y_i\boldsymbol{x}_i^\top\right\| \|\theta-\theta'\|\right]^2 \\
&\leq \left[2\left\|\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\|\|\theta'\| + \left\|\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\|\|\theta-\theta'\| + 2y_i\|\boldsymbol{x}_i\|\right]^2 \|\theta-\theta'\|^2
\end{aligned}$$

by triangular inequality and writing $\left\|\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\|$ the 2-norm of the matrix $\boldsymbol{x}_i\boldsymbol{x}_i^\top$, which is equal to $\|\boldsymbol{x}_i\|^2$ and bounded by one by hypothesis. As Θ is supposed to be enclosed in a ball of radius ρ, we further have:

$$\left((y_i - \boldsymbol{x}_i^\top\theta)^2 - (y_i - \boldsymbol{x}_i^\top\theta')^2\right)^2 \leq (4\rho+2)^2\|\theta-\theta'\|^2$$

Given these elements, Theorem 8 is applicable to the linear regression setting and one obtains the desired result. ■

5. Implementation

5.1. The DPP's ideal marginal kernel

Following the theoretical results, the ideal strategy (although unrealistic) to build the marginal kernel K of the ideal DPP sampling scheme would be as follows. 1/ Deal with outliers as explained in Appendix E until $\sigma_{\max}$ is not too large. 2/ Compute all σ_i. 3/ Set all π_i to $m\sigma_i/\mathfrak{S}$ with m sufficiently large as detailed in the theorems. 4/ Find all non-diagonal elements of K in order to minimize for all θ the estimator's variance, as derived in Eq. (12):

$$\text{Var}(\hat{L}) = \sum_i \left(\frac{1}{\pi_i}-1\right) f(x_i,\theta)^2 - \sum_{i\neq j}\frac{\mathsf{K}_{ij}^2}{\pi_i\pi_j} f(\boldsymbol{x}_i,\theta)f(\boldsymbol{x}_j,\theta).$$

while constraining K to be a valid marginal kernel, *i.e.*: SDP with $0 \preceq \mathsf{K} \preceq 1$, 5/ sample a DPP with kernel K. On our way to derive a practical algorithm with a linear complexity in n, many obstacles stand before us: there is no known polynomial algorithm to compute all σ_i in the general setting, solving exactly the minimization problem of step 4 under eigenvalue constraint remains open, and sampling from this engineered ideal K costs $\mathcal{O}(n^3)$ number of operations (see Algorithm 1 of Kulesza and Taskar (2012): it necessitates a full diagonalization of K). Designing a linear-time algorithm that provably verifies under a controlled error the conditions of our previous theorems is out-of-scope of this paper. In the following, we prefer to first recall the intuitions behind the construction of a good kernel, and then discuss two choices of kernel we advocate: a Gaussian kernel and a Vandermonde-based kernel.

5.2. A first choice: the Gaussian kernel

In order for K to be a good candidate for coresets, it needs to verify the following two properties:

- As indicated by the theorems, the diagonal entries K_{ii} should increase as the associated σ_i increases.
- As indicated by the variance equation of Eq. (12), off-diagonal elements should be as large as possible (in absolute value) given the eigenvalue constraints. In fact, we cannot set all non-diagonal entries of K to large values as the matrix's 2-norm would rapidly be larger than 1. We thus need to choose the best pairs (i, j) for which it is worth setting a large value of K_{ij}. A first glance at the variance equation indicates that the larger $f(x_i, \theta)f(x_j, \theta)$ is, the larger K_{ij} should be, in order to decrease the variance as much as possible. Recall nevertheless that in the coreset setting, all sampling parameters should be independent of θ. The off-diagonal elements should thus verify the following property: the larger is the correlation between x_i and x_j (the more similar are $f(x_i, \theta)$ and $f(x_j, \theta)$ for all θ), the larger K_{ij} should be.

We show in the following in what ways the choice of marginal kernel

$$\mathsf{K} = \mathsf{L}(\mathsf{I} + \mathsf{L})^{-1}$$

with L the Gaussian kernel matrix with parameter τ:

$$\forall (i, j) \quad \mathsf{L}_{ij} = \exp^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\tau^2}},$$

is a good candidate to build coresets for k-means (the linear regression case is discussed later). Let us write $\mathsf{U} = (\boldsymbol{u}_1 | \ldots | \boldsymbol{u}_n)$ the orthonormal eigenvector basis of L and $\Lambda = \mathrm{diag}(\lambda_1 | \ldots | \lambda_n)$ its diagonal matrix of sorted eigenvalues, $0 \leq \lambda_1 \leq \ldots \leq \lambda_n$. U and Λ naturally depend on τ. One shows for instance that, with respect to τ, λ_n is a monotonically increasing function between 1 and n.

Concerning the off-diagonal elements of K, let us first note that if x_i and x_j are correlated (that is, in the k-means setting, if they are close to each other), then

$$\mathsf{K}_{ij} = \sum_k \frac{\lambda_k}{1 + \lambda_k} u_k(i) u_k(j)$$

should be large in absolute value. In fact, in the limit where $x_i = x_j$, then $\forall k, u_k(i) = u_k(j)$ and $\mathsf{K}_{ij} = \mathsf{K}_{ii} = \mathsf{K}_{jj}$. The determinant of the 2×2 submatrix of K indexed by i and j is therefore null: sampling both will never occur. Thus, the closer are x_i and x_j, the lower is the chance of sampling both jointly. Moreover, if x_i and x_j are far from each other (for instance, in different clusters), then the entries i and j of $\mathsf{L}'s$ eigenvectors will be very different. For instance, say the data set contains two well separated clusters of similar size. If the Gaussian parameter τ is set to the size of these clusters, then the kernel matrix L will be quasi-block diagonal, with each block corresponding to the entries of each cluster. Also, each eigenvector $\boldsymbol{u}_k$ will have energy either in one cluster or the other such that K_{ij}

is necessarily small if i and j belong to different clusters, and the event of sampling both jointly is probable.

Concerning the probability of inclusion of i, we have:

$$\mathsf{K}_{ii} = \sum_k \frac{\lambda_k}{1 + \lambda_k} v_i(k)^2,$$

where $\boldsymbol{v}_i$ is the vector of size n verifying $\forall k, v_i(k) = u_k(i)$. For all i, $\|\boldsymbol{v}_i\|^2 = 1$. The probability of inclusion is thus directly linked to the values of k that contain the energy of $\boldsymbol{v}_i$: the more the energy of $\boldsymbol{v}_i$ is contained on high values of k, the larger is the probability of inclusion. Say we are again in a situation where the clusters and the choice of Gaussian parameter τ are such that L is quasi block diagonal. Within each block, the eigenvector associated with the highest eigenvalue corresponds approximately to the constant vector. These eigenvectors being normalized, the associated entry of $v_i(k)$ is thus approximately equal to $1/\sqrt{\#C_i}$ where $\#C_i$ is the size of the cluster containing data x_i. Typically, if the cluster is small, that is, if $\#C_i$ tends to 1, the associated entry $v_i(k)$ tends to 1 as well, such that all the energy of v_i is drawn towards high values of k, thus increasing the probability of inclusion of i. In other words, the more isolated, the higher the chance of being sampled. This corresponds to the intuition one may obtain for the sensitivity σ_i. It has indeed been shown that the sensitivity may be interpreted as a measure of outlierness (Lucic et al., 2016).

In the linear regression case, a similar argumentation is possible, up to the fact that point i can be an outlier from the point of view of $\boldsymbol{x}_i$ and/or y_i, such that the kernel should take both into account: we suggest the Gaussian kernel in dimension $d + 1$:

$$\forall (i, j) \quad \mathsf{L}_{ij} = \exp^{-\frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2}{2\tau^2}},$$

with $\boldsymbol{z}_i = [\boldsymbol{x}_i^\top, y_i]^\top \in \mathbb{R}^{d+1}$.

In both contexts, we thus advocate to sample DPPs via a Gaussian kernel L-ensemble. We now move on to detailing an efficient sampling implementation.

5.2.1. Efficient implementation

Sampling exactly a DPP from the Gaussian L-ensemble verifying

$$\forall (i, j) \qquad \mathsf{L}_{ij} = \exp^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\tau^2}}$$

consists in the following steps:

1. Compute L.
2. Diagonalize L in its set of eigenvectors $\{\boldsymbol{u}_k\}$ and eigenvalues $\{\lambda_k\}$.
3. Sample a DPP given $\{\boldsymbol{u}_k\}$ and $\{\lambda_k\}$ via Algorithm 1 of Kulesza and Taskar (2012).

Step 1 costs $\mathcal{O}(n^2 d)$, step 2 costs $\mathcal{O}(n^3)$, step 3 costs $\mathcal{O}(n\mu^3)$, where we recall that μ is the expected number of samples of the DPP. This naive approach is thus not practical.

Algorithm 1 The Gaussian kernel coreset sampling heuristics

Input: $\mathcal{X} = \{x_i\}$ a set of n points in $\mathbb{R}^d$, a Gaussian kernel parameter τ, a number of samples m

- Draw $r \geq \mathcal{O}(m)$ random Fourier vectors associated to the Gaussian kernel with parameter τ
- Compute the associated RFF matrix $\Psi \in \mathbb{R}^{2r \times n}$ as explained in Appendix F.1
- Compute $\mathsf{C} = \Psi\Psi^\top \in \mathbb{R}^{2r \times 2r}$ the dual representation
- Compute the eigendecomposition of C: obtain eigenvectors $\{\boldsymbol{v}_k\}$ and eigenvalues $\{\nu_k\}$
- Draw a sample $\mathcal{S}$ from a m-DPP with L-ensemble $\mathsf{L} = \Psi^\top\Psi$ as explained in Appendix F.3.
- Compute the marginal probabilities π_s for all $\boldsymbol{x}_s \in \mathcal{S}$ as explained in Appendix F.3, and set weights $\omega(\boldsymbol{x}_s) = 1/\pi_s$.

Output: $\{\mathcal{S}, \omega\}$ a weighted sample of size m.

We detail in Appendix F how to reduce the overall complexity to $\mathcal{O}(n\mu^2)$, by 1/ taking advantage of Random Fourier Features (RFF) (Rahimi and Recht, 2008) to estimate a low dimensional representation $\Psi \in \mathbb{R}^{2r \times n}$ of the L-ensemble $\mathsf{L} \simeq \Psi^\top\Psi$, where r is the chosen number of features; and 2/ running a DPP sampling algorithm adapted to such a low rank representation.

In the experimental section, we will concentrate on m-DPPs as they are simpler to compare with state of the art methods that all have a fixed known-in-advance number of samples. The overall m-DPP sampling algorithm adapted to the k-means problem that we will consider is summarized in Algorithm 1: given the data $\mathcal{X}$, the number of desired samples m, and the Gaussian parameter τ, it outputs a weighted set of m samples $\mathcal{S}$ that is a good candidate to be a coreset if m is large enough. The runtime to build Ψ is $\mathcal{O}(ndr)$; to compute C and diagonalize it is $\mathcal{O}(nr^2)$; to sample a m-DPP given this dual eigendecomposition is $\mathcal{O}(nm^2)$. Given that r is set to a few times m, the overall runtime of Algorithm 1 is $\mathcal{O}(ndm + nm^2)$.

Given a number of samples m to draw, how should one set the Gaussian parameter τ? The larger is τ, the more repulsive is the m-DPP, and the smaller is the numerical rank of Ψ (the number of eigenvalues ν such that $n\nu$ is larger than the machine's precision). Now, numerical instabilities arise while sampling an m-DPP if the numerical rank of Ψ decreases below m: τ should not be set too large. Also, the smaller is τ, the closer is L to the identity matrix, such that the closer is the m-DPP to uniform sampling without replacement: τ should not be set too small. We will see in the following experimental section how the choice of τ affects results.

5.3. A second choice: a projective DPP based on the Vandermonde matrix

A second choice of DPP sampling, that derives from our analysis, is the projective DPP with m samples from the L-ensemble $\mathsf{L} = \mathsf{V}\mathsf{V}^\top$ where V is the Vandermonde matrix (discussed in Section 3.3.1). This choice has several advantages over the Gaussian kernel:

- V takes $\mathcal{O}(nm)$ operations to compute: the overall m-DPP sampling cost is thus naturally $\mathcal{O}(nm^2)$, with no need for any approximation technique.

Algorithm 2 The Vandermonde-based coreset sampling heuristics

Input: $\mathcal{X} = \{x_i\}$ a set of n points in $\mathbb{R}^d$, a number of samples m

- m should verify: $\exists \phi \in \mathbb{N}$ such that $m = \binom{\phi+d}{\phi}$.
- Compute the Vandermonde matrix $\mathsf{V} \in \mathbb{R}^{n \times m}$.
- Compute the $(\mathsf{Q} \in \mathbb{R}^{n \times m}, \mathsf{R} \in \mathbb{R}^{m \times m})$ decomposition of V: $\mathsf{V} = \mathsf{Q}\mathsf{R}$ with $\mathsf{Q}^\top\mathsf{Q} = \mathsf{I}_m$ and R an upper triangular matrix.
- Draw a sample $\mathcal{S}$ from a projective DPP with L-ensemble $\mathsf{L} = \mathsf{Q}\mathsf{Q}^\top$ as explained in Algorithm 3.
- Compute the marginal probabilities π_s for all $\boldsymbol{x}_s \in \mathcal{S}$ with $\pi_s = \|\mathsf{Q}(s,:)\|^2$ the energy of the s-th line of Q; and set weights $\omega(\boldsymbol{x}_s) = 1/\pi_s$.

Output: $\{\mathcal{S}, \omega\}$ a weighted sample of size m.

- no particular scale τ is introduced.

This choice however has the drawback that in dimension higher than 1, not all values of m are allowed (only values of m for which there exists $\phi \in \mathbb{N}$ s.t. $m = \binom{\phi+d}{\phi}$), as explained at the end of Section 3.3.1.

5.4. Alternative algorithms for sampling DPPs, and potential improvements

The algorithm we suggest scales in our experience rather well with n, and makes it practical to find coresets with n in the millions or more. Our method scales more poorly in m, the number of points retained, which in practice should be in the hundreds at most. Recall that m should scale roughly as the intrisic dimension of the parameter space: it is therefore possible that in certain difficult problems no reliable coreset can be found if[5] $m < 1,000$. With that in mind, we now review other methods for sampling DPPs.

As an alternative to direct sampling of the kind used here, MCMC methods have been suggested several times (*e.g.,* Anari et al., 2016), and the earliest reference we could find is Belabbas and Wolfe (2009). The most basic kind starts with a set of points sampled uniformly, and uses random swapping moves: at each iteration, a point from the current set may be replaced by one not in the set.[6] Acceptance probabilities are set so that the limiting distribution of the chain is the correct DPP. Each iteration has cost $\mathcal{O}(m^2)$, and approximately $\mathcal{O}(n)$ such iterations are required for mixing (Hermon and Salez, 2019). The total cost is therefore the same as in our method ($\mathcal{O}(nm^2)$), so not much gain is to be expected here. However, there is no need for a low-rank approximation of the kernel such as the RFF approximation used here. In a nutshell, MCMC techniques sample approximately from the correct kernel instead of sampling exactly from an approximate kernel: which is better is as yet unknown but an interesting problem in itself.

There are two immediate strategies for increasing m. One is to use a crude heuristic for dividing the original data set into p different subsets, and sampling a DPP independently from each subset. This is equivalent to using a block-diagonal kernel, and along these lines there is a less radical approach, which is to force the kernel to be sparse and exploit

5. One might argue that in such cases the coreset methodology is of dubious value anyway.
6. A more advanced algorithm by Gautier et al. (2017) mixes faster than the basic algorithm outlined here, but the iterations are more involved.

sparsity in the sampling. Poulson (2019) shows how to exploit sparsity for sampling DPPs when the *marginal* kernel is sparse. Unfortunately, we use L-ensembles here, and one would have to adapt the tools given by Poulson to L-ensembles. A different strategy to increase m is to sample the DPP several times rather than just once. The resulting sample has less diversity but is much cheaper to generate. One can take advantage of recent methods that use pre-processing for speeding up repeated sampling of the same DPP (Gillenwater et al., 2019; Derezinski et al., 2019). Here the challenge is to find the right trade-off between computational cost and repulsion, which is again an interesting question for future research.

6. Experiments

6.1. Different strategies to compare...

We will empirically compare results obtained with the five following approaches:

1. `m-DPP` : The strategy summarized in Algorithm 1.
2. `PolyProj-DPP` : The strategy summarized in Algorithm 2.
3. `matched iid` : An iid sampling strategy with replacement, matched to either `m-DPP` or `PolyProj-DPP` (depending on the context). More precisely, m samples are drawn iid with replacement, the probability of selecting x_i at each draw being set to $p_i = \pi_i/m$, where π_i is the marginal probability of drawing x_i in `m-DPP` (or `PolyProj-DPP`).
4. `uniform iid` : Uniform iid sampling with replacement.
5. `sensitivity iid` : The current state of the art iid sampling based on a bi-criteria approximation to upper bound the sensitivity (Algorithm 2 of Bachem et al. 2017), or, if available (for instance in the case of 1-means and linear regression), an analytical formula of the sensitivity.

For the three iid methods (methods 3, 4 and 5), we will use the importance sampling estimator adapted to iid sampling of Eq. (6). For methods 1 and 2, we will use the importance sampling estimator adapted to correlated sampling of Eq. (8).

Empirically, when the ambient dimension d is small, performance of all methods is enhanced if the weights in $\hat{\mathsf{L}}$ are set via Voronoi cells rather than set to inverse probabilities: given the sample $\mathcal{S}$ of size m, compute its Voronoi tessellation in m cells, and associate to each sample $\boldsymbol{x}_s$ a weight $\omega(\boldsymbol{x}_s)$ equal to the number of datapoints in its associated Voronoi cell. We will call the associated cost estimators $\hat{L}$ the Voronoi estimators.

For completeness, we compare all these methods with another negatively correlated sampling method called D^2-sampling (commonly used for k-means++ seeding, see Arthur and Vassilvitskii 2007):

6) $\mathtt{D}^2$: sample the first element of $\mathcal{S}$ uniformly at random. Each subsequent element of $\mathcal{S}$ is drawn according to a probability proportional to the squared distance to the closest of the already sampled elements. The marginal probabilities are not known in this algorithm, so we will only be able to build the associated Voronoi cost estimator.

To measure the performance of each method, we will empirically estimate the probability that, given the method's sampled weighted subset, it verifies the coreset property of Eq. (4) for a given randomly chosen θ (setting ϵ to 0.1). On the artificial data models we investigate, we estimate this probability via 50 randomly chosen θ on 1000 realizations of the data. On the real-world data sets, we estimate this probability via 5000 randomly chosen θ. We will in general plot this probability versus the number of samples: the closer it is to 1, the better the sampling method for coresets.

In Sections 6.2.2 and 6.2.3, we will not only compare the coreset property of the samples obtained by each method, we will also compare the result of Lloyd's classical k-means heuristics (Lloyd, 1982) performed on the entire data versus the result obtained on the weighted samples of each method. To be precise, once the k-means heuristics on the weighted subset outputs k centroids, we classify all nodes (sampled or not) according to their closest distance to the centroids: this gives us a partition that we then compare using the Adjusted Rand (AR) similarity index (Hubert and Arabie, 1985) to the ground truth associated to the data set. The AR index is a number between -1 and 1: the closer it is to 1, the closer are the partitions, the better the sampling method.

6.2. ...on different data sets

6.2.1. To start with: two well controlled cases

We start with two perfectly controlled cases (for which we derived the sensitivities analytically – see the first and third lemmas of Appendix D)):

- the 1-means case, for which we show that, supposing without loss of generality that the data is centered ($\sum_j x_j = 0$), the sensitivity verifies the following analytic form:

$$\sigma_i = \frac{1}{n}\left(1 + \frac{\|x_i\|^2}{v}\right),$$

  where $v = \frac{1}{n}\sum_{x\in\mathcal{X}} \|x\|^2$.

- the linear regression case, for which we show that:

$$\forall i \qquad \sigma_i = x_i^\top \mathsf{H}^{-1} x_i + \frac{(y_i - y_i^*)^2}{\|y - y^*\|^2}$$

  where $\mathsf{H} = \mathsf{X}^\top\mathsf{X}$ and y^* reads $y^* = \mathsf{X}\theta^* = \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y$.

We are thus able to compare our method versus the ideal iid sampling scheme for which we set p_i, the probability of drawing x_i, exactly to its ideal value given in Theorem 3: $p_i = \sigma_i/\mathfrak{S}$ ($= \sigma_i/2$ for 1-means, $= \sigma_i/(d+1)$ for linear regression).

Experiments with 1-means. We will work on a simple isotropic Gaussian data set of $n = 1000$ points in dimension $d = 2, 20$ or 100. A percentage q of the n points are drawn as outliers (uniformly in the ambient space and far from the Gaussian mean). An instance of such a data set in $d = 2$ dimensions, and with $q = 0.01$ is shown in Figure 3a.

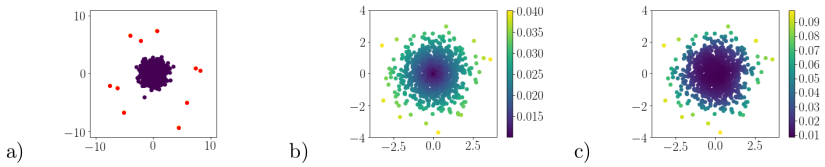


Figure 3: a) A realization of an artificial data set of $n = 1000$ data points; blue points are drawn from an isotropic Gaussian, a proportion $q = 0.01$ of the points are drawn as outliers (displayed in red). b) In a case without outliers, and for $m = 20$, we represent the inverse importance sampling weights of `sensitivity iid`, *i.e.*, $m\sigma_i/\mathfrak{S}$, using the exact analytical formulation of the sensitivity in the 1-means case (see Lemma 23). c) On the same data realization, and also setting $m = 20$, we represent the inverse importance sampling weights of `m-DPP`: the inclusion probability π_i.

We start by showing in Figure 4 the results of `m-DPP` versus the number of dimensions and the choice of parameter τ for the Gaussian kernel. All shown results are with $q = 0$ (no outlier) and with a number of random Fourier features $r = 200$. Several comments are in order. Firstly, compared to the importance sampling estimator, the Voronoi estimator produces good results in low dimensions, and fails as the dimension increases. Secondly, the performance of all methods increase and uniformize as the dimension increases. This is due to the fact that in large dimensions, interpoint distances tend to uniformize such that any pair of points tend to be representative of all interpoint distances, thus simplifying the problem of finding good coresets. This may also explain why the choice of τ is less crucial in higher dimension. In low dimensions, however, the choice of τ has a strong impact on performance. The best choice for τ depends in fact on the number of samples m one requires: as m increases, τ should be set smaller. This is in fact natural: if one desires a very short summary of the data set (small m), the repulsion of the DPP has to be strong in order to sample a diverse subset. Whereas if the length of the summary is less constrained, τ should be decreased to allow for a less coarse-grained description. This observation leads to the natural question of the optimal τ given the data and m. We currently lack of a satisfying answer to this question, both theoretically and empirically. A usual heuristics in kernel methods is to set τ to the average (or median) interdistance of the points in the data set. In the experiments of Figure 4, the average interdistance corresponds to $\tau \simeq 1.7, 6.2$ and 14.0 for $d = 2, 20$ and 100 respectively, which give in fact a good order of magnitude for the choice of τ. In the following, to simplify the discussion, we will sometimes set τ to be the average interdistance, that we will denote by $\bar{\tau}$.

We compare next the performance of several methods in Figure 5. One observes that the superior performance of the Voronoi estimator over the importance sampling estimator in low dimension d is verified for all methods. Moreover, as the dimension increases, all methods converge to the performance of the uniform iid sampling method. Finally, `m-DPP`

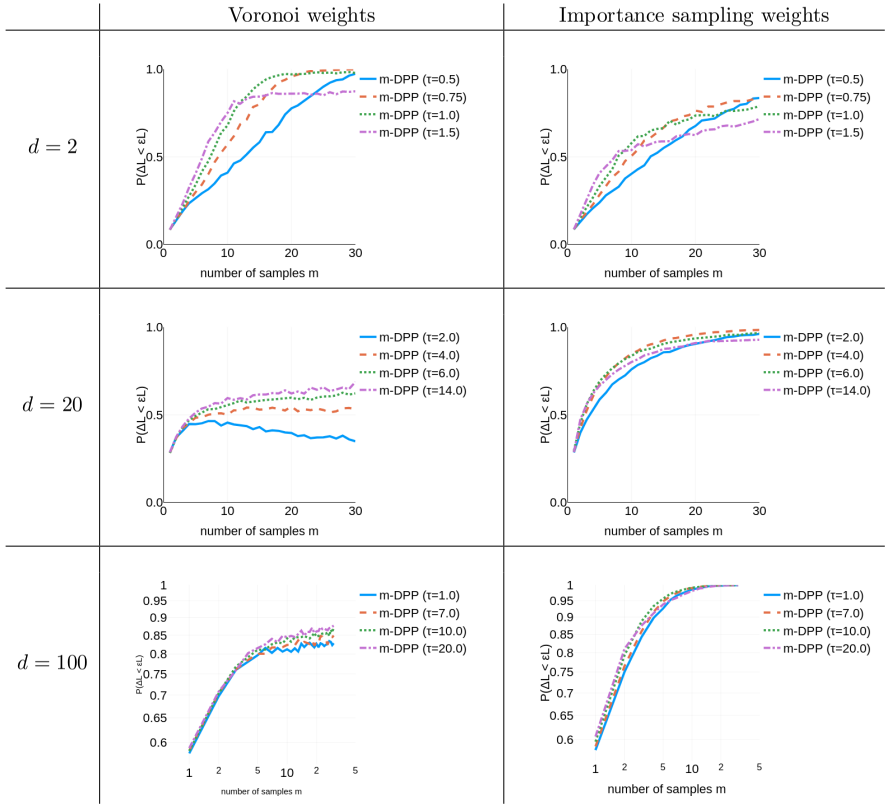


Figure 4: Performance of `m-DPP` on the 1-means problem, versus the dimension d, the parameter τ of the Gaussian kernel and the choice of weights (Voronoi or importance sampling weights) in the cost estimator. The two bottom figures are plotted in log-log scale, for readability.

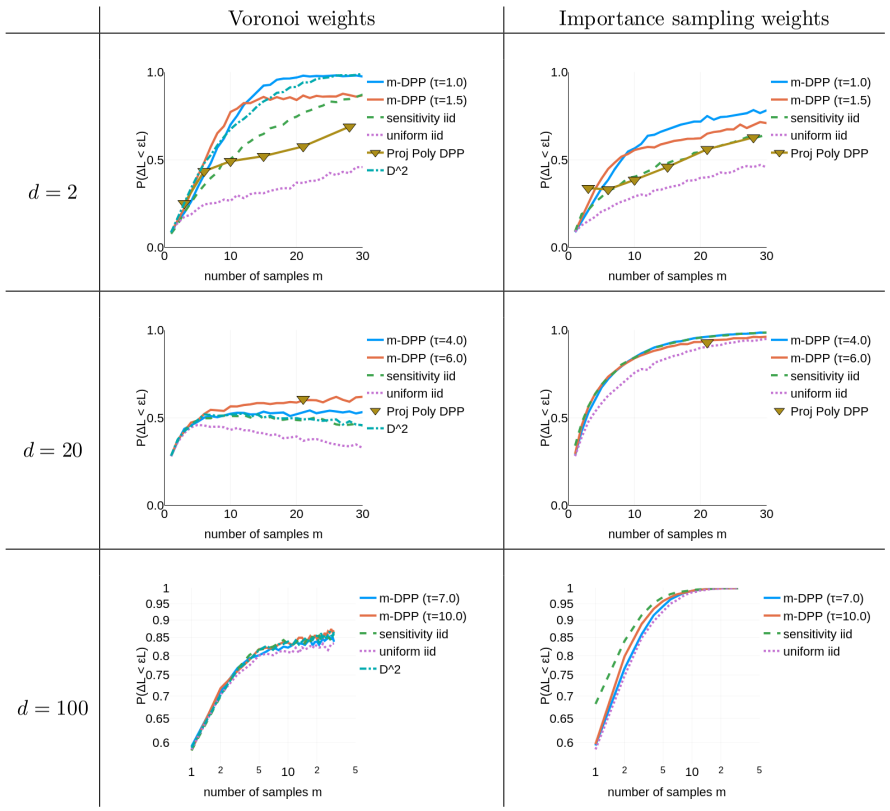


Figure 5: Performance comparison of different sampling methods on the 1-means problem, versus the dimension d and the choice of weights (Voronoi or importance sampling weights) in the cost estimator. The two bottom figures are plotted in log-log scale, for readability.

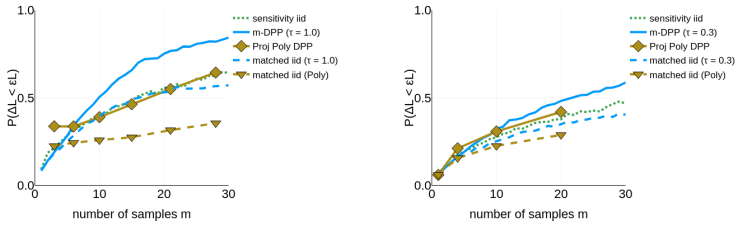


Figure 6: Comparison between `m-DPP` and `PolyProj-DPP` versus `matched iid` and `sensitivity iid` on the 1-means problem (left) and the linear regression problem (right), for $d = 2$.

associated with Voronoi weights is competitive with $\mathtt{D}^2$ in low d; and, regardless of how one chooses the weights, m-`DPP` has a clear edge over the sensitivity-based iid random sampling (the lower the dimension, the clearer the edge). Finally, `PolyProj-DPP` matches the performance of `sensitivity iid` for importance sampling weights. For Voronoi weights, the results for $d = 2$ and $d = 20$ are contradictory and we cannot conclude.

In order to clarify further discussion, we will now focus on the importance sampling estimated cost. One should keep in mind that in low dimensions, Voronoi-based estimated costs usually perform well, but fail (sometimes drastically) as the dimension increases.

A natural question arises at this point: is the observed edge of `m-DPP` over `sensitivity iid` due to a better probability of inclusion of the point process? Or is it truly due to the negative correlations induced by the determinantal nature of our method? In fact, we compare in Figure 3 the probability of inclusion for `sensitivity iid` versus `m-DPP`: they have a similar general behavior but are nevertheless quantitatively different. In Figure 6 (left), we compare `m-DPP` and `PolyProj-DPP` versus `matched iid` and `sensitivity iid`: the observed edge is clearly due to the negative correlations induced by the determinantal nature of our method. As expected from Corollary 18, the best inclusion probability is based on the sensitivity. Nevertheless, the figure shows that even if it is not set to its ideal value (as in `m-DPP` and `PolyProj-DPP`), one can still improve the performance by inducing negative correlations.

For completeness, we still need to discuss the impact of two variables: the number of random Fourier features r used in `m-DPP`, and the percentage of outliers q in the data. In the following, we set τ to $\bar{\tau}$, the average interdistance. Figure 7 shows the impact of the choice of r on performances: as expected, as r increases, performance increases, and as d increases, performances become more sensitive to the choice of r. The impact of the choice of r is nevertheless very limited: setting r to a multiple of m has been a safe choice in all our experiments. Finally, Figure 8 shows the impact of the percentage of outliers q on performances. Empirically, we see here that outliers have a smaller impact on DPP sampling

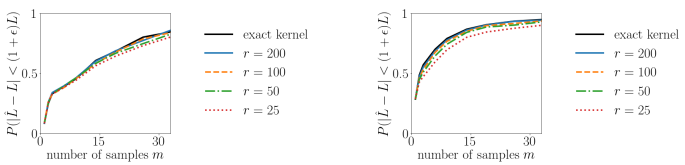


Figure 7: Performance comparison of `m-DPP` on the 1-means problem versus the number of RFF r, for $d = 2$ (left) and $d = 20$ (right).

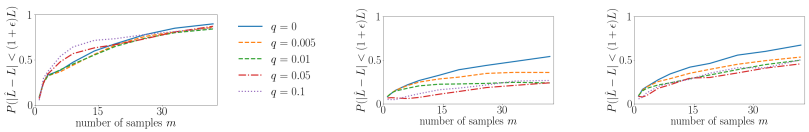


Figure 8: Performance comparison of `m-DPP` (left), `uniform iid` (middle) and `sensitivity iid` (right) on the 1-means problem versus the percentage of outliers q.

than on uniform or sensitivity-based iid sampling.

Experiments with linear regression. The data $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ are generated by sampling n points in the hypercube $[0,1]^d$. Each entry of the vector y is also sampled uniformly from $[0,1]$. The outlier percentage q is set to zero. We show the equivalent of Figs 4 and 5 in, respectively, Figs 9 and 10. Results for $d = 20$ and $d = 100$ are very similar so the case $d = 100$ is not shown.

We observe similar results: `m-DPP` matches $\mathtt{D}^2$ in the Voronoi estimator, and outperforms `sensitivity iid` in all cases; `PolyProj-DPP` at least matches `sensitivity iid` in all cases.

Finally, in Figure 6 (right), we compare `m-DPP` and `PolyProj-DPP` versus `matched iid` and `sensitivity iid` for the linear regression problem: once again, the observed edge is clearly due to the negative correlations induced by the determinantal nature of our method.

We conclude these first well-controlled experimental results by summarizing the observed behaviors:

- `m-DPP` outperforms the current state of the art `sensitivity iid`, even in the 1-means and the linear regression cases, where sensitivities do not need to be estimated but may be computed exactly.
- `PolyProj-DPP` matches and in some cases outperforms `sensitivity iid`, at least for the importance sampling estimator.

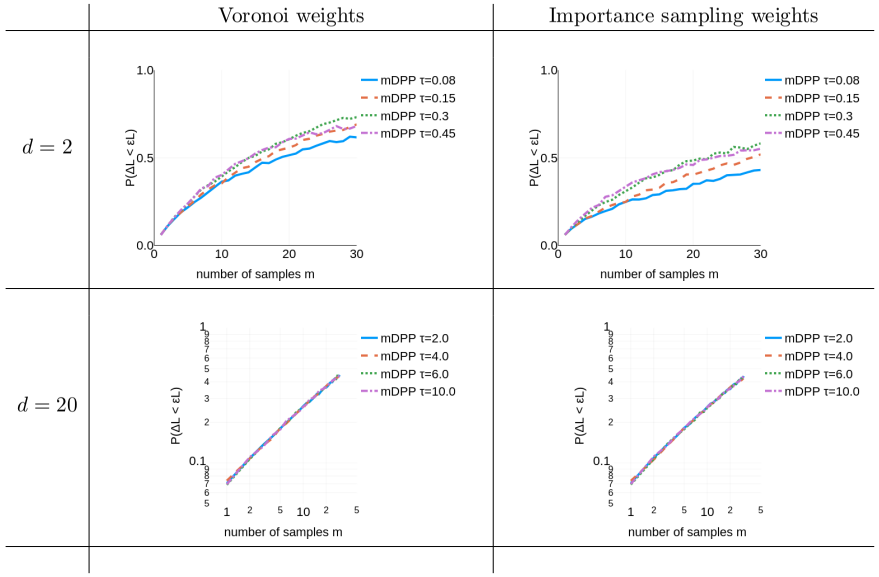


Figure 9: Performance of `m-DPP` on the linear regression problem, versus the dimension d, the parameter τ of the Gaussian kernel and the choice of weights (Voronoi or importance sampling weights) in the cost estimator. The two bottom figures (for $d = 20$) are in log-log scale. Performances are so similar that even in this scale they remain undecipherable.

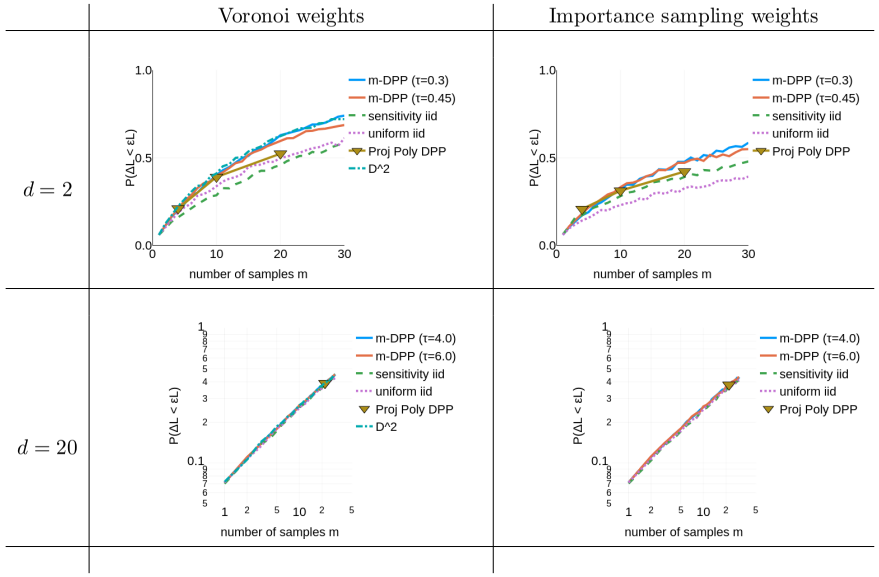


Figure 10: Performance comparison of different sampling methods on the linear regression problem, versus the dimension d and the choice of weights (Voronoi or importance sampling weights) in the cost estimator. The two bottom figures (for $d = 20$) are in log-log scale. Performances are so similar that even in this scale they remain undecipherable.

- As the dimension increases, the edge over iid sampling decreases. In fact, all performances tend to `uniform iid`.
- The best choice of parameter τ of the Gaussian kernel in `m-DPP` is still an open problem. Empirically, a good order of magnitude is the average interdistance of the datapoints. Ideally, nevertheless, τ should increase as m, the number of wanted samples, decreases.
- Regarding the number of RFFs r, setting r to $\mathcal{O}(m)$ is sufficient.
- Regarding the impact of outliers. Our theorems are not well suited to outliers (due to the proof techniques used); nevertheless, in practice, we see that outliers are not an issue in our method: they even have a smaller impact on our method's performances than on other methods.
- Replacing weights by Voronoi weights yields in general better results, but only in low dimension. As the dimension increases, the Voronoi cost estimator fails (sometimes drastically).

6.2.2. Experiments on non-Gaussian data: the case of spectral features

Spectral features. Given a graph of n nodes where $\mathsf{W} \in \mathbb{R}^{n\times n}$ is the adjacency matrix (*i.e.*, $\mathsf{W}_{ij} = 1$ if nodes i and j are connected, and 0 ortherwise), a standard problem consists in partitioning the nodes in k communities, *i.e.*, sets of nodes more connected to themselves than to other nodes of the graph (Fortunato, 2010). A classical algorithm to solve efficiently this problem is the so-called spectral clustering algorithm (Ng et al., 2002):

- Define the normalized Laplacian matrix $\mathsf{L} = \mathsf{I} - \mathsf{D}^{-\frac{1}{2}}\mathsf{W}\mathsf{D}^{-\frac{1}{2}} \in \mathbb{R}^{n\times n}$ where I is here the identity matrix in dimension n, and $\mathsf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $\mathsf{D}_{ii} = d_i = \sum_j \mathsf{W}_{ij}$ the degree of node i.
- Compute via Arnoldi iterations or a similar algorithm the k first eigenvectors of L: $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$.
- Associate to each node i a (spectral) feature vector $\boldsymbol{x}_i \in \mathbb{R}^k$: $\forall l = 1, \ldots, k \quad x_i(l) = u_l(i)$.
- Normalize all feature vectors: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i / \|\boldsymbol{x}_i\|_2$.
- Run k-means on all such normalized spectral features.

An extensive literature exists on spectral clustering and it has shown to be a very successful unsupervised classification algorithm in many situations (von Luxburg, 2007; Tremblay and Loukas, 2020).

The Stochastic Block Model (SBM). We consider random community-structured graphs drawn from the SBM, a classical class of structured random graphs (see for instance Abbe and Sandon, 2015). We first look at graphs with k communities of same size n/k. In the SBM, the probability of connection between any two nodes i and j is q_1 if they are in the same community, and q_2 otherwise. One can show that the average degree reads

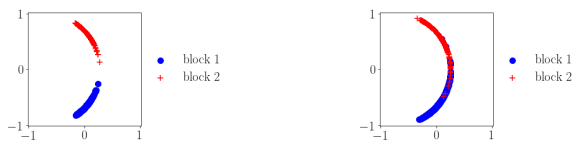


Figure 11: Examples of SBM spectral features $\boldsymbol{x}_i$, here with $k = 2$. Each colour corresponds to one block of the SBM. On the left, for an "easy" classification task ($\zeta = \zeta_c/4$), and on the right, for a harder setting ($\zeta = \zeta_c/2$).

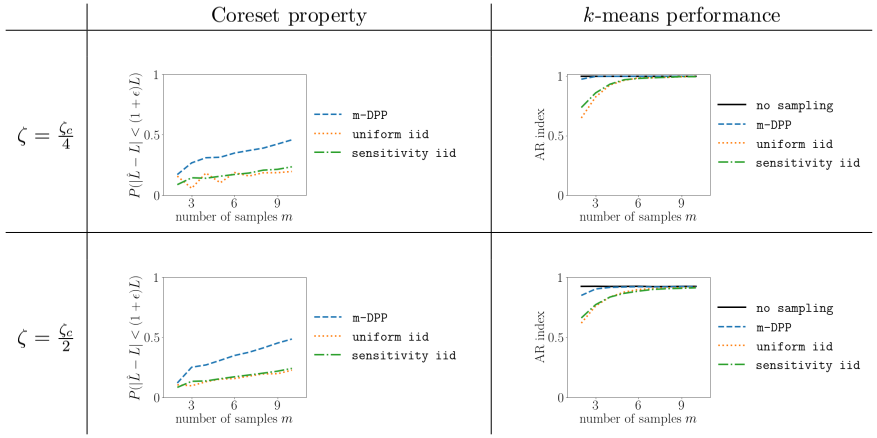


Figure 12: Performance comparison of different methods on the k-means problem for spectral features of balanced SBM graphs (here, $k = 2$). Left: testing the coreset property. Right: the Adjusted Rand index between the partition recovered by k-means on the weighted subsets and the ground truth partition of the SBM. ζ quantifies the difficulty of the classification task (see text): the lower it is, the easier the classification task.

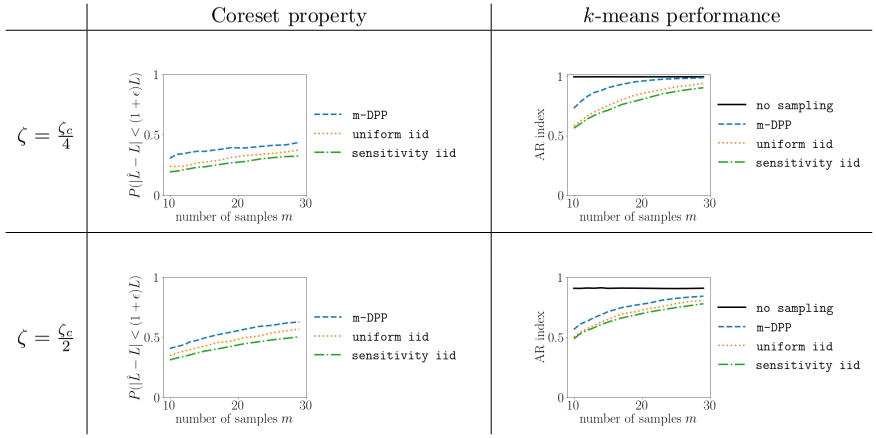


Figure 13: Same as Figure 12 but with $k = 10$.

$c = q_1\left(\frac{n}{k} - 1\right) + q_2\left(n - \frac{n}{k}\right)$. Thus, instead of providing the probabilities (q_1, q_2), one may characterize a SBM by considering $(\zeta = \frac{q_2}{q_1}, c)$. The larger ζ, the fuzzier the community structure, the harder the classification task. In fact, Decelle et al. (2011) show that above the critical value $\zeta_c = (c - \sqrt{c})/(c + \sqrt{c}(k-1))$, community structure becomes undetectable in the large n limit. In the following, we set $n = 1000$ and $c = 16$; k and ζ will vary. Note that spectral features $\boldsymbol{x}_i$ are not Gaussian and, in fact, do not fall into any classical data model (see Figure 11 to visualize instances of SBM spectral features with $k = 2$). They are thus interesting candidates to test k-means algorithms.

Results. For different values of ζ and k, we generate 1000 such SBM graphs from which we sample subsets according to different methods. We test both the coreset property (as before) and the k-means performance on the weighted subset compared to the k-means performed on all data. We plot in Figure 12 (resp. Figure 13) the results obtained for $k = 2$ (resp. $k = 10$). Note that in this case, we have no explicit formula for the sensitivity such that for `sensitivity iid`, we use the bi-criteria approximation scheme provided in Algorithm2 of Bachem et al. (2017). Here again, we see how our method outperforms iid sampling schemes, even in difficult classification contexts (for instance when $\zeta = \zeta_c/2$: even with all the data, k-means' performance saturates at an AR index of 0.9). Moreover, as the dimension increases (here $d = k$), performances of all methods tend to uniformize. Surprisingly, `uniform iid` performs as well ($k = 2$) and even outperforms ($k = 10$) `sensitivity iid`. We believe this is due to approximation errors of the bi-criteria scheme used to find upper bounds of the sensitivity. Also, in this balanced case (communities have the same number of nodes), uniform sampling is in fact a good option. We will now see how this changes in the unbalanced case.

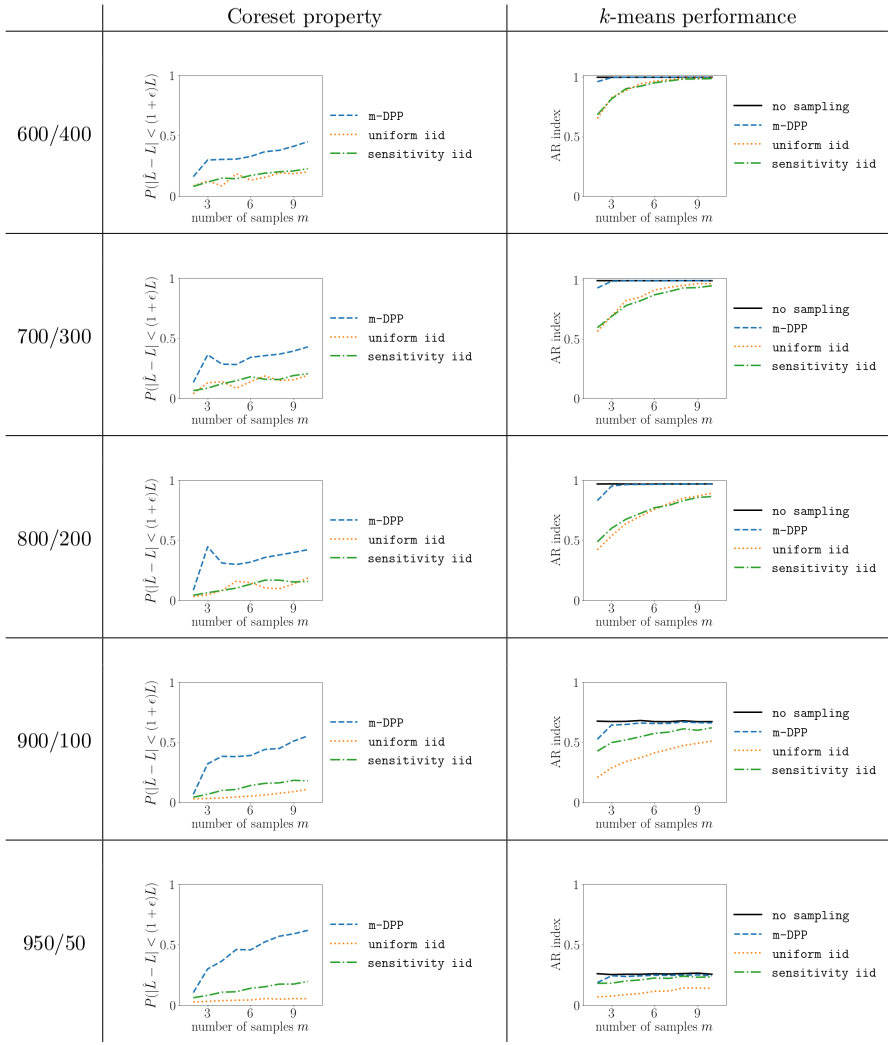


Figure 14: Same as Figure 12 but with a fixed $\zeta = \zeta_c/4$ and a varying level of balance within the sizes of the $k = 2$ communities. n_1/n_2 means one community with n_1 nodes and the other with n_2 nodes.

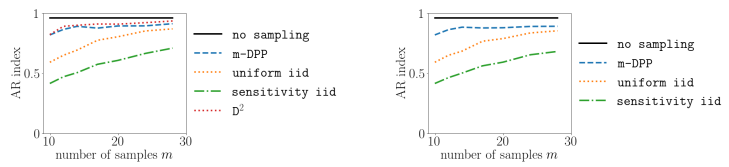


Figure 15: Classification performance on the MNIST data set obtained with different sampling methods versus the result obtained without sampling, using Voronoi weights (left), or importance sampling weights (right).

The unbalanced case. In the unbalanced case, ζ_c is no longer a recovery threshold, but we may still use ζ as a marker of difficulty of the recovery task. We set ζ to $\zeta_c/4$ and perform the same experiments as previously with $k = 2$ blocks of unbalanced size. Results are shown in Figure 14. For a fixed ζ, the more unbalanced, the more difficult the recovery task. Also, the more unbalanced, the better is `sensitivity iid` compared to `uniform iid`. Nevertheless, `m-DPP` shows an edge over all iid methods in all tested configurations.

6.2.3. Experiments on two real world data sets

The MNIST data set. We perform a first experiment on the MNIST data set (LeCun, 1998) that consists in $7 \cdot 10^4$ images of handwritten digits (from 0 to 9) for which the ground truth is known. The classical associated machine learning goal is to classify them in 10 classes (one for each digit). To do so, we pre-process the data in the following unsupervised way. We consider all images and extract SIFT descriptors (Vedaldi and Fulkerson, 2010) for each image. We then use FLANN (Muja and Lowe, 2014) to compute a κ-nearest neighbor graph (with κ=10) based on these descriptors. We finally run the spectral clustering algorithm with $k = 10$ to find the 10 classes corresponding to each digit, as explained in Section 6.2.2. The k-means step is thus the last step of the overall processing. We compare results obtained with different sampling methods versus the results obtained without sampling in Figure 15 (bottom). For `m-DPP`, several values of τ were tried, and we show here the result obtained for $\tau = 1.5$. Also, a number $r = 200$ of Fourier features were used. We see that, in the Voronoi weight setting, `m-DPP` is competitive with $\mathtt{D}^2$. Moreover, `uniform iid` outperforms `sensitivity iid` certainly due to approximation errors of the bi-criteria procedure and to the fact that the data is balanced (there are more or less $7 \cdot 10^3$ instances of each digit in the data set), thus favoring uniform sampling. Finally, `m-DPP` outperforms once again the iid random sampling techniques. Note that the methods' classification performance is remarkable. Without sampling, the overall classification performance in terms of AR index with the ground truth is 0.95. With only $\sim$ 20 samples, `m-DPP` reaches a performance of ~ 0.9!

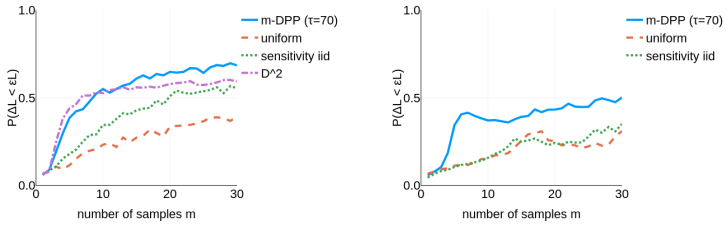


Figure 16: Performance of different sampling methods on the US Census data set. Left: performance using Voronoi weights. Right: performance using importance sampling weights.

The US Census data set. We also perform experiments on the 1990 US Census data set[7], that consists in $n = 2458285$ surveyed persons, and $d = 68$ categorical attributes such as age, income, etc. The data was pre-processed by a series of operation detailed on its download webpage. As there is no ground truth in this data set to compare to, we arbitrarily decide $k = 15$ classes, and show solely the coreset property of the samples obtained via different methods. For `m-DPP`, τ was set to 70 (the mean interdistance estimated on 1000 randomly chosen pairs of datapoints), and a number $r = 30$ of Fourier features was chosen. Experiments were done with τ ranging from $\tau = 30$ to $\tau = 140$ with no qualitative change in performance (not shown). Figure 16 shows the results of the experiments. We see that `m-DPP` outperforms all other methods, in both Voronoi and importance sampling settings. In this example, note that `sensitivity iid` outperforms `uniform iid` probably due to the fact that the 15 potential classes are unbalanced.

6.3. Computation time

Comparing computation times is always a tricky endeavour: they heavily depend on the quality of implementation, choice of language, choice of experiments, choice of parameters, hardware, etc. Giving the full picture is out-of-scope of this paper. We hope here to give some insight in the computational complexity of the proposed methods. We suggest to first recap theoretical times before looking at times observed in a few experiments.

In theory. The theoretical time for `m-DPP` is $\mathcal{O}(nrd)$ for the RFFs, $\mathcal{O}(nr^2)$ for the SVD of the RFFs, $\mathcal{O}(nm^2)$ for the sampling, and $\mathcal{O}(nm + r)$ for the computation of the inclusion probabilities π. As r is set to $\mathcal{O}(m)$, this sums up to $\mathcal{O}(nm^2 + nmd)$. The theoretical time for `PolyProj-DPP` is $\mathcal{O}(nm)$ to compute the Vandermonde matrix, $\mathcal{O}(nm^2)$ for the QR decomposition, $\mathcal{O}(nm^2)$ for the sampling, and $\mathcal{O}(nm)$ to compute the inclusion probabilities π (which are simply the sum of squares of each line of `Q`); which sums up to $\mathcal{O}(nm^2)$. The theoretical time for Algorithm 2 of Bachem et al. (2017) used here for the bi-criteria

7. downloaded from `https://archive.ics.uci.edu/ml/datasets/US+Census+Data+(1990)`.

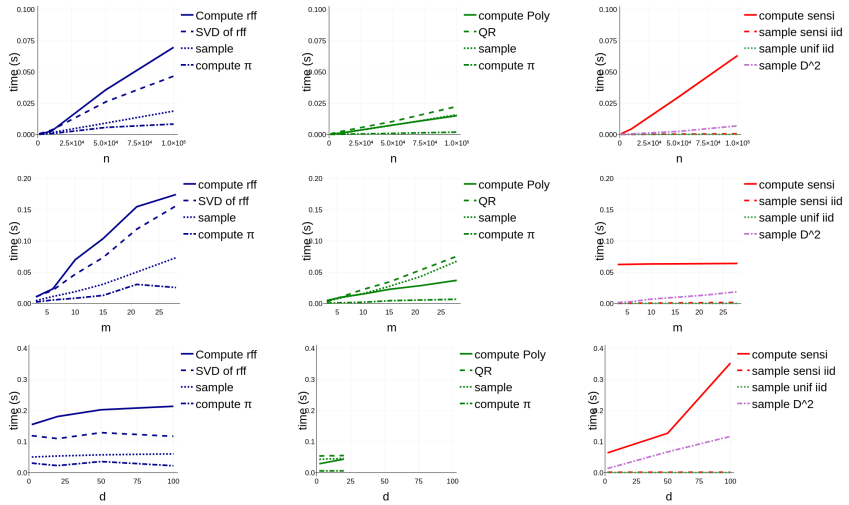


Figure 17: Computation times for all methods. Top line: $d = 2$, $m = 10$, versus n. Middle line: $n = 10^5$, $d = 2$, versus m. Bottom line: $n = 10^5$, $m = 21$, versus d. Left column: the four fundamental operations associated to `m-DPP`: i/ compute the RFFs with r set to m here, ii/ compute the SVD of the RFFs, iii/ sample from the m-DPP, iv/ compute the marginal probabilities π useful for the importance sampling estimator. Middle column: the four equivalent operations for `PolyProj-DPP`: i/ compute the Vandermonde matrix ("compute Poly"), ii/ compute the QR decomposition of that matrix, iii/ sample from the projective DPP, iv/ compute the marginal probabilities π. Right column: the two fundamental operations for `sensitivity iid`: i/ the computation of the bi-criteria approximation ("compute sensi"), ii/ the weighted iid sampling with replacement itself, along with `uniform iid` and `D`^{2}.

approximation of the sensitivities in the k-means context, is $\mathcal{O}(Indk)$ where I is the number of times Algorithm 1 of Bachem et al. (2017) is run to find the best initialization (we set it to $I = 10$ in our experiments). Once upper bounds of the sensitivities have been computed, the iid sampling time is negligeable. The theoretical time of $\mathtt{D}^2$ is $\mathcal{O}(ndm)$.

In practice. Experiments were made on a laptop with 8 cores and 16 GB of memory, with the Julia toolbox available on the authors's website.[8] Figure 17 shows some computation times versus m, d and n for the 1-means problem. We observe a linear progression in n of the determinantal methods, as well as a superlinear progression in m for some parts of the computations. Comparing with `sensitivity iid`, one observes: the lower m and the larger d, the more our methods are comparable in terms of computation time.

Note that this comparison is in fact in favor of `sensitivity iid`: the computation time of the bi-criteria approximation increases in fact with the number of expected classes k. The figures represented here are for $k = 1$. We reproduce in the following table the computation times for the US-Census data set, with $k = 15$, $m = 30$, $r = 30$:

`m-DPP`: rff	`m-DPP`: svd	`m-DPP`: sample	`m-DPP`: π	`sensi iid`: bi-criteria	`sensi iid`: sample	$\mathtt{D}^2$
2.6s	2.5s	2.7s	1.3s	18.7s	0.04s	3.1s

The total time for m-DPP sampling is thus half the time necessary for the bi-criteria approximation; for a substantial gain in performance, as seen in Figure 16.

To conclude, the determinantal methods proposed in this paper are in many situations, especially as m increases, and d decreases, slightly heavier in computation time than iid sampling. Due to the observed gain in coreset performance, however, we believe that the additional cost is worth the effort.

7. Conclusion

In this work, we introduced a new random sampling method based on DPPs to build coresets. Different from sensitivity-based iid random sampling, our method introduces negative correlations between samples due to its determinantal nature. Also, different from D^2 sampling, also known to be repulsive, the proposed method is tractable in the sense that marginal probabilities are known and importance sampling schemes can be used. Our theoretical results may be summarized in three points. Firstly, Thms 8 and 20 provide coreset guarantees in function of the point process' probabilities of inclusion. These guarantees are not stronger than the iid case and are in fact similar: they both show that the ideal marginal probabilities are proportional to the sensitivity. Nevertheless, these results do not take into account higher order inclusion probabilities coding for the repulsion within the sampled subsets and are in fact verified for any choice of such high-order marginals (provided K stays SDP with eigenvalues between 0 and 1). This leads to the second point: given that these higher-order inclusion probabilities offer extra degrees of freedom and due to simple variance arguments (theorems of Section 3.2), we show that DPP-based random sampling

8. The DPP4Coresets Julia toolbox is also available at `https://gricad-gitlab.univ-grenoble-alpes.fr/tremblan/dpp4coresets.jl`.

necessarily yields better performance than its independent counterparts. On the theoretical side, additional work is required to specify precisely the minimum number of required samples guaranteeing the coreset property. We expect that further research on concentration properties of strongly Rayleigh measures, involving not only first order marginals, but higher-order ones, should enable to move forward in this direction. The third and final point is the rebalancing property of polynomial DPPs: without any prior density-like estimation, polynomial DPPs such as the ones described in Section 3.3.1 provide samples that are asymptotically independent of the underlying distribution of the data. Even though this result is only asymptotic, it is yet another argument in favor of DPP sampling for coresets.

From an application point-of-view, the coreset theorems were applied to the ubiquitous k-means and linear regression problems. Given a data set, the ideal L-ensemble L adapted to these problems is untractable and we thus propose two heuristics, one via random Fourier features of the Gaussian kernel, and one based on the Vandermonde matrix, in order to efficiently sample a DPP that has the desirable properties to sample coresets (if not provably, at least quantitatively). To sample a subset of size m, our heuristics run resp. in $\mathcal{O}(nm^2 + nmd)$ and $\mathcal{O}(nm^2)$. This is more expensive than the sensitivity-based iid strategy, especially as the number of samples m increases; but empirically provides better coresets on different artificial and real-world data sets.

Finally, this work calls for several extensions. First of all, two likely difficult theoretical questions: how to improve the concentration inequalities for DPPs? (such improvements would directly benefit the coreset theorem's bounds). How to find the optimal DPP kernel given a data set? (which asks in fact difficult questions in frame theory). Also, these DPP sampling schemes should be extended to the streaming and/or distributed settings.

Acknowledgments

This work was partly funded by the ANR GenGP (ANR-16-CE23-0008), the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01), the CNRS PEPS I3A (Project RW4SPEC), the Grenoble Data Institute (ANR-15-IDEX-02) and the LIA CNRS/Melbourne Univ Geodesic.

Appendix A. Proof of Theorem 8

Proof The theorem consists in proving that Eq. (4) is true under the assumptions of Theorem 8. We follow a classical proof scheme from compressed sensing (Baraniuk et al., 2008), in four steps:

1. we first use concentration arguments for a given $\theta \in \Theta$.
2. we then build an ϵ-net paving the space of parameters Θ.
3. via the union bound, we obtain the result for all θ in the ϵ-net.
4. via the Lipschitz property of f, we obtain the desired result for all $\theta \in \Theta$.

Step 1 (Concentration around $\theta \in \Theta$) DPPs are instances of sampling schemes that are strongly Rayleigh. Since strongly Rayleigh distributions are closed under truncation, any m-DPP is also strongly Rayleigh. We can thus apply the concentration results of Pemantle and Peres (2014). For a given $\theta \in \Theta$, we have : $\forall \epsilon \in (0,1), \forall \delta \in (0,1)$:

$$\mathbb{P}\left(\left|\frac{\hat{L}}{L} - 1\right| \geq \epsilon\right) = \mathbb{P}\left(\left|\hat{L} - L\right| \geq \epsilon L\right) \leq \delta,$$

provided that:

$$m \geq \frac{8}{\epsilon^2} C^2 \log \frac{2}{\delta}, \tag{20}$$

with $C = \max_i \frac{f(x_i, \theta)}{L\bar{\pi}_i}$, where $\bar{\pi}_i$ is a shorthand for π_i/m, and π_i is the marginal probability of sampling element i.

Using the same concentration results, we also have:

$$\forall (\epsilon, \delta) \in (0,1)^2, \quad \mathbb{P}\left(\left|\frac{\sum_i \frac{\epsilon_i}{\pi_i}}{n} - 1\right| \geq \epsilon\right) \leq \delta, \tag{21}$$

provided that:

$$m \geq \frac{8}{\epsilon^2} \frac{1}{n^2 \bar{\pi}_{\min}^2} \log \frac{2}{\delta}, \tag{22}$$

where $\bar{\pi}_{\min} = \min_i \bar{\pi}_i$.

Step 2 (ϵ'-net of Θ) Consider $\Gamma_{\epsilon'} = (\theta_1^*, \ldots, \theta_\eta^*)$ the smallest subset of Θ such that balls of radius ϵ' centered around the elements in $\Gamma_{\epsilon'}$ cover Θ. $\Gamma_{\epsilon'}$ is called an ϵ'-net of Θ and $\eta = |\Gamma_{\epsilon'}|$ its covering number. The covering property entails that:

$$\forall \theta \in \Theta \quad \exists \theta^* \in \Gamma_{\epsilon'} \quad \text{s.t.} \quad d_\Theta(\theta, \theta^*) \leq \epsilon'.$$

Step 3. (Union bound) Write $\delta' = \delta/2\eta$. From step 1, we know that, $\forall \theta^* \in \Gamma_{\epsilon'}$:

$$\mathbb{P}\left(\left|\frac{\hat{L}}{L} - 1\right| \geq \epsilon\right) \leq \delta'$$

provided that:

$$m \geq \frac{8}{\epsilon^2} C^2 \log \frac{2}{\delta'}.$$

From the union bound, we have:

$$\mathbb{P}\left(\forall \theta^* \in \Gamma_{\epsilon'}, \quad \left|\frac{\hat{L}}{L} - 1\right| \leq \epsilon\right) \geq 1 - \sum_{\theta^* \in \Gamma} \delta' = 1 - \frac{\delta}{2},$$

provided that:

$$m \geq \frac{8}{\epsilon^2} \left(\max_{\theta^* \in \Gamma_{\epsilon'}} C\right)^2 \log \frac{4\eta}{\delta} \tag{23}$$

Given that $\bar{\pi}_i$ will *in fine* be independent of θ (as we want the coreset property to be true for all $\theta \in \Theta$),

$$\max_{\theta^* \in \Gamma_{\epsilon'}} C = \max_{\theta^* \in \Gamma_{\epsilon'}} \max_i \frac{f(x_i, \theta)}{L\bar{\pi}_i} \tag{24}$$

$$= \max_i \frac{1}{\bar{\pi}_i} \max_{\theta^* \in \Gamma_{\epsilon'}} \frac{f(x_i, \theta)}{L} \tag{25}$$

$$\leq \max_i \frac{1}{\bar{\pi}_i} \max_{\theta \in \Theta} \frac{f(x_i, \theta)}{L} = \max_i \frac{\sigma_i}{\bar{\pi}_i}, \tag{26}$$

where we see how the sensitivity σ_i naturally arises in the proof. Eq. (26) entails that Eq. (23) is verified if $m \geq m_1$ with

$$m_1 = \frac{8}{\epsilon^2} \left(\max_i \frac{\sigma_i}{\bar{\pi}_i}\right)^2 \log \frac{4\eta}{\delta}.$$

Write $\delta'' = \delta/2$. From Eq. (21), we have:

$$\mathbb{P}\left(\left|\frac{\sum_i \frac{\epsilon_i}{\pi_i}}{n} - 1\right| \geq \epsilon\right) \leq \delta'',$$

provided that $m \geq m_2$ with

$$m_2 = \frac{8}{\epsilon^2 n^2 \bar{\pi}_{\min}^2} \log \frac{4}{\delta}.$$

We have (with the union bound again):

$$\mathbb{P}\left(\left|\frac{\sum_i \frac{\epsilon_i}{\pi_i}}{n} - 1\right| \leq \epsilon \quad \text{AND} \quad \forall \theta^* \in \Gamma_{\epsilon'}, \quad \left|\frac{\hat{L}}{L} - 1\right| \leq \epsilon\right)$$
$$\geq 1 - \delta/2 - \delta'' = 1 - \delta,$$

provided that:

$$m \geq \max(m_1, m_2).$$

Step 4 (Continuity argument) Suppose that $m \geq \max(m_1^*, m_2^*)$ with m_1^*, m_2^* as defined in the theorem. The result of step 3 with $\epsilon \leftarrow \epsilon/2$ states that, with probability at least $1-\delta$, one has:

$$\left| \frac{\sum_i \frac{\epsilon_i}{\pi_i}}{n} - 1 \right| \leq \frac{\epsilon}{2} \quad \text{AND} \quad \forall \theta^* \in \Gamma_{\epsilon'}, \quad \left| \frac{\hat{L}}{L} - 1 \right| \leq \frac{\epsilon}{2}. \tag{27}$$

We now look for the maximum value of ϵ' such that Eq. (27) implies the following desired result:

$$\forall \theta \in \Theta, \qquad \left| \frac{\hat{L}}{L} - 1 \right| \leq \epsilon. \tag{28}$$

Consider $\theta \in \Theta$. By the covering property of $\Gamma_{\epsilon'}$, we have:

$$\exists \theta^* \in \Gamma_{\epsilon'} \text{ s.t. } d_\Theta(\theta, \theta^*) \leq \epsilon'.$$

Moreover, as f is γ-Lipschitz, $\forall x_i \in \mathcal{X}$:

$$|f(x_i, \theta) - f(x_i, \theta^*)| \leq \gamma \, d_\Theta(\theta, \theta^*) \leq \gamma \epsilon'. \tag{29}$$

Thus, using Eqs. (29) and then (27):

$$\begin{aligned} \hat{L}(\mathcal{X}, \theta) &\leq \hat{L}(\mathcal{X}, \theta^*) + \gamma \epsilon' \sum_i \frac{\epsilon_i}{\pi_i} \\ &\leq (1 + \frac{\epsilon}{2})(L(\mathcal{X}, \theta^*) + n\gamma\epsilon'). \end{aligned}$$

Also, using Eq. (29) again:

$$L(\mathcal{X}, \theta^*) \leq L(\mathcal{X}, \theta) + n\gamma\epsilon'.$$

Thus:

$$\hat{L}(\mathcal{X}, \theta) \leq (1 + \frac{\epsilon}{2}) L(\mathcal{X}, \theta) + 2n\gamma\epsilon'(1 + \frac{\epsilon}{2}). \tag{30}$$

Similarly, for the lower bound, one obtains:

$$(1 - \frac{\epsilon}{2})(L(\mathcal{X}, \theta) - 2n\gamma\epsilon') \leq \hat{L}(\mathcal{X}, \theta) \tag{31}$$

In order for Eqs (30) and (31) to imply Eq.(28), we need:

$$2n\gamma\epsilon'(1 + \frac{\epsilon}{2}) \leq \frac{\epsilon}{2} L(\mathcal{X}, \theta),$$

i.e.:

$$\epsilon' \leq \frac{\epsilon L(\mathcal{X}, \theta)}{4n\gamma(1 + \frac{\epsilon}{2})} \leq \frac{\epsilon L(\mathcal{X}, \theta)}{6n\gamma}.$$

In order for this condition to be true for all θ, we choose:

$$\epsilon' = \frac{\epsilon \min_{\theta \in \Theta} L(\mathcal{X}, \theta)}{6n\gamma} = \frac{\epsilon L^{\text{opt}}}{6n\gamma} = \frac{\epsilon \langle f \rangle_{\text{opt}}}{6\gamma}. \tag{32}$$

Concluding the proof. Consider $\mathcal{S}$ a sample from a DPP with L-ensemble L, with marginal probabilities π_i and normalized marginal probabilities $\bar{\pi}_i = \pi_i/m$. Consider $\epsilon \in (0,1)$ and $\delta \in (0,1)$. Define ϵ' as in Eq. (32) and Γ the set of centers of the η balls of radius ϵ' covering the parameter space. We showed that if $m \geq \max(m_1^*, m_2^*)$, then $\mathcal{S}$ is an ϵ-coreset with probability at least $1-\delta$. ■

Appendix B. Coreset results for DPPs

Theorem 20 (DPP for coresets) *Consider $\mathcal{S}$ a sample from a DPP with L-ensemble L, and an average number of sample $\mu = \sum_i \frac{\lambda_i}{1+\lambda_i}$. Let $\epsilon \in (0,1)$, $\delta \in (0,1)$. Denote by η the minimum number of balls of radius $\epsilon\langle f \rangle_{opt}/6\gamma$ necessary to cover Θ. With probability higher than $1-\delta$, $\mathcal{S}$ is a ϵ-coreset provided that*

$$\mu \geq \mu^* = \max(\mu_1^*, \mu_2^*)$$

with:

$$\begin{aligned} \mu_1^* &= \frac{32}{\epsilon^2} \left(\epsilon \max_i \frac{\sigma_i}{\bar{\pi}_i} + 4 \left(\max_i \frac{\sigma_i}{\bar{\pi}_i} \right)^2 \right) \log \frac{10\eta}{\delta}, \\ \mu_2^* &= \frac{32}{\epsilon^2} \left(\frac{\epsilon}{n\bar{\pi}_{min}} + \frac{4}{n^2 \bar{\pi}_{min}^2} \right) \log \frac{10}{\delta}, \end{aligned}$$

and $\forall i$, $\bar{\pi}_i = \pi_i/\mu$.

Proof According to Pemantle and Peres (2014), replace Eq. (20) by:

$$\mu \geq \frac{16}{\epsilon^2} \left(\epsilon C + 2C^2 \right) \log \frac{5}{\delta},$$

with $C = \max_i \frac{f(x_i, \theta)}{L\bar{\pi}_i}$, where $\bar{\pi}_i$ is a shorthand for π_i/μ; and Eq. (22) by:

$$\mu \geq \frac{16}{\epsilon^2 n \bar{\pi}_{\min}} \left(\epsilon + \frac{2}{n\bar{\pi}_{\min}} \right) \log \frac{5}{\delta},$$

and change accordingly the rest of the proof. ■

For the same reasons as the m-DPP case, we have:

Corollary 21 *If $n\sigma_{min} \geq 1$, then $\mu_1^* \geq \mu_2^*$ and the coreset property of Theorem 20 is verified if:*

$$\mu \geq \mu^* = \frac{32}{\epsilon^2}\left(\epsilon \max_i \frac{\sigma_i}{\pi_i} + 4\left(\max_i \frac{\sigma_i}{\pi_i}\right)^2\right)\log\frac{10\eta}{\delta}.$$

Corollary 22 *If there exists $\alpha > 0$ and $\beta \geq 1$ such that:*

$$\begin{aligned} \forall i \qquad & \alpha\sigma_i \leq \pi_i \leq \alpha\beta\sigma_i, \\ \textit{and} \qquad & \frac{\alpha}{\beta} \geq \frac{32}{\epsilon^2}(\epsilon + 4\mathfrak{S})\log\frac{10n}{\delta}, \end{aligned}$$

then $\mathcal{S}$ is a ϵ-coreset with probability at least $1-\delta$. In this case, the expected number of samples verifies:

$$\mu \geq \frac{32}{\epsilon^2}\beta\mathfrak{S}(\epsilon + 4\mathfrak{S})\log\frac{10n}{\delta}.$$

Appendix C. Proof of Theorem 16

We split the proof into two parts. We first show the convergence of the discrete intensity function to its continuous limit (as n goes to infinity). We then deal with the outer limit (as the degree ϕ goes to infinity) to prove the theorem.

C.1. Discrete-to-continuous limit

The discrete DPP defined in Section 3.3.1 has a natural continuous counterpart: namely, instead of sampling $\mathcal{S}$ from $\mathcal{X}$, we directly sample $\mathcal{S}$ from Ω. The corresponding orthogonal polynomials are now orthogonal w.r.t. the measure μ, and the inclusion probabilities turn into intensity functions (in the continous limit, any given point in Ω has probability 0 of being selected, which is why we need to integrate over an ϵ ball). The counterpart of the discrete marginal kernel $\mathsf{K} = \mathsf{Q}\mathsf{Q}^\top$ is now a positive-definite kernel

$$k_\mu(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^m q_{\mu,i}(\boldsymbol{x})q_{\mu,i}(\boldsymbol{y})$$

where $q_{\mu,i}$ is the i'th orthogonal polynomial under μ. As in the discrete case, the intensity function for the continuous DPP simply equals the diagonal values of the kernel, i.e:

$$\iota(\boldsymbol{y}) = k_\mu(\boldsymbol{y}, \boldsymbol{y})$$

We need to introduce the Christoffel functions of a measure. The Christoffel function of μ is defined as:

$$\lambda_{\mu,\phi}(\boldsymbol{y}) = \min_{f \in \Pi_\phi^d} \frac{\int f(\boldsymbol{x})^2 d\mu}{f(\boldsymbol{y})^2} \tag{33}$$

where Π_ϕ^d is the set of polynomials in $\mathbb{R}^d$ of degree less than or equal to ϕ. Re-expressing f in the orthonormal basis for μ, and solving for the argmin in (33), we find:

$$\lambda_{\mu,\phi}(\boldsymbol{y}) = \frac{1}{k_\mu(\boldsymbol{y}, \boldsymbol{y})}$$

so that the intensity function of the (continuous) polynomial DPP is just one over the Christoffel function. The argument is also valid for the discrete case, replacing μ with the empirical distribution $\mu_n = (1/n)\sum \delta_{\boldsymbol{x}_i}$. We may rewrite the empirical Christoffel function as:

$$\lambda_{\mu_n,\phi}(\boldsymbol{y}) = \min_{f \in \Pi_\phi^d, f(\boldsymbol{y})=1} \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)^2$$

Convergence of the empirical Christoffel function to its continuous limit is proved formally by Lasserre and Pauwels (2017, Theorem 3.11), and is easy to see from the formula above ($\lambda_{\mu_n,\phi}(\boldsymbol{y})$ is just the minimum of a quadratic empirical functional). In the large n limit, we have:

$$\lim_{n\to\infty} \lambda_{\mu_n,\phi}(\boldsymbol{y}) = \lambda_{\mu,\phi}(\boldsymbol{y})$$

a.s., uniformly in $\boldsymbol{y} \in D$. $\lambda_{\mu,\phi}(\boldsymbol{y})$ for $\boldsymbol{y} \in D$ is bounded below in convex domains (Prymak, 2017), so that convergence of the inclusion probabilities to the intensity function follows:

$$\lim_{n\to\infty} \mathbb{P}(\boldsymbol{x}_i \in \mathcal{S}|\mathcal{X}) = \iota_\phi(\boldsymbol{x}_i)$$

For the unconditional intensity measure, we need to average the left-hand side over $\mathcal{X}$:

$$\begin{aligned} \lim_{n\to\infty} I_{n,\phi}(\mathcal{A}) &= \lim_{n\to\infty} \mathbb{E}_{\mathcal{X},\mathcal{S}}\left\{\sum_{s_i \in \mathcal{S}} \mathbb{I}(s_i \in \mathcal{A})\right\} \\ &= \lim_{n\to\infty} \mathbb{E}_{\mathcal{X}}\left\{\sum_{x_i \in \mathcal{X}} \mathbb{P}(\boldsymbol{x}_i \in \mathcal{S}|\mathcal{X})\mathbb{I}(x_i \in \mathcal{A})\right\} \end{aligned}$$

The quantities in the expectation are bounded (≤ 1), and by dominated convergence we may interchange the limit and the expectation, so that:

$$\lim_{n\to\infty} I_{n,\phi}(\boldsymbol{y}) = \int_{\mathcal{A}} \iota_\phi(\boldsymbol{y}) d\mu(\boldsymbol{y}) \tag{34}$$

uniformly in $\boldsymbol{y} \in D$.

C.2. Large-m asymptotics

For the next step, we use the fact that asymptotics of Christoffel functions are well-studied. We let $\phi \to \infty$, in which case Theorem 1.5 in Kroo and Lubinsky (2013) gives us:

$$\lim_{\phi\to\infty} \frac{\iota_\phi(\boldsymbol{y})}{k_{\nu,\phi}(\boldsymbol{y}, \boldsymbol{y})} = \frac{1}{\mu'(\boldsymbol{y})} \tag{35}$$

Here $k_{\nu,\phi}$ is the projection kernel for orthogonal polynomials of degree l under the Lebesgue measure. Note that $\iota_\phi(\boldsymbol{y})$ and $k_{\nu,\phi}(\boldsymbol{y}, \boldsymbol{y})$ both integrate to m over Ω, and $\frac{1}{k_{\nu,\phi}(\boldsymbol{y},\boldsymbol{y})}$ is the Christoffel function for ν, which tends to a well-defined limit independent of μ. Injecting (35) into (34) proves our result.

Appendix D. Proof of three Lemmas

Lemma 23 *In the 1-means problem (the k-means problem with $k = 1$), and supposing without loss of generality that the data is centered (i.e.: $\sum_j x_j = 0$), we have:*

$$\sigma_i = \frac{1}{n}\left(1 + \frac{\|x_i\|^2}{v}\right),$$

where $v = \frac{1}{n}\sum_{x\in\mathcal{X}} \|x\|^2$. Thus:

$$\mathfrak{S} = \sum_i \sigma_i = 2.$$

Proof By definition:

$$\frac{1}{\sigma_i} = \min_c \frac{\sum_x \|x-c\|^2}{\|x_i - c\|^2}.$$

Consider $\mathcal{S}(x_i, R)$ the sphere centered on x_i and radius $R \geq 0$. We have that:

$$\min_c \quad = \quad \min_{R\geq 0} \quad \min_{c\in\mathcal{S}}$$

We thus have:

$$\frac{1}{\sigma_i} = \min_{R\geq 0} \frac{1}{R^2} \min_{c\in\mathcal{S}} \sum_x \|x-c\|^2 .$$

Writing $x - c = x - x_i - (c - x_i)$, we may write

$$\sum_x \|x-c\|^2 = nR^2 + \sum_x \|x-x_i\|^2 - 2R\left\|\sum_x x - x_i\right\| \cos\theta,$$

with θ the angle formed by $\sum_x x - x_i$ and $c - x_i$. As the minimum is sought for c on the sphere, the angle θ may take any value, such that the minimum is always attained with θ s.t. $\cos\theta = 1$. We finally obtain:

$$\frac{1}{\sigma_i} = n + \min_{R\geq 0} \frac{1}{R^2}\left(\sum_x \|x-x_i\|^2 - 2R\left\|\sum_x x - x_i\right\|\right).$$

Studying analytically the function $f(R) = \frac{a-2bR}{R^2}$, its minimum is attained for $R^* = \frac{a}{b}$ and $f(R^*) = -\frac{b^2}{a}$, such that:

$$\frac{1}{\sigma_i} = n - \frac{||\sum_x x - x_i||^2}{\sum_x \|x-x_i\|^2}.$$

Supposing without loss of generality that the data is centered, *i.e.*: $\sum_x x = 0$ and denoting $v = \frac{1}{n}\sum_x \|x\|^2$, we have:

$$\frac{1}{\sigma_i} = n - \frac{n^2\|x_i\|^2}{nv + n\|x_i\|^2}.$$

Inverting this equation yields:

$$\sigma_i = \frac{v + \|x_i\|^2}{nv + n\|x_i\|^2 - n\|x_i\|^2} = \frac{1}{n}\left(1 + \frac{\|x_i\|^2}{v}\right)$$

■

Lemma 24 *In the k-means problem, $n\sigma_{min} \geq 1$.*

Proof Consider $\theta^{\text{opt}} = (c_1^{\text{opt}}, \ldots, c_k^{\text{opt}})$ the optimal solution of k-means and $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k\}$ their associated Voronoi sets. Consider $x_i \in \mathcal{X}$ and suppose, without loss of generality that $x_i \in \mathcal{V}_1$. Also, for any $x \in \mathcal{X}$, we denote by $c(x) = \text{argmin}_{c\in\theta} \|x-c\|^2$. We have:

$$\begin{aligned}\frac{1}{\sigma_i} &= \min_{c_1,\ldots,c_k} \frac{\sum_{x\in\mathcal{X}} \|x - c(x)\|^2}{\|x_i - c(x_i)\|^2} \\ &= \min_{c_1,\ldots,c_k} \frac{\sum_{x\in\mathcal{V}_1} \|x - c(x)\|^2}{\|x_i - c(x_i)\|^2} + \sum_{j=2}^{k} \frac{\sum_{x\in\mathcal{V}_j} \|x - c(x)\|^2}{\|x_i - c(x_i)\|^2}\end{aligned}$$

Given that, by definition of $c(x)$, $\forall j,\ \|x - c(x)\|^2 \leq \|x - c_j\|^2$, we have:

$$\frac{1}{\sigma_i} \leq \min_{c_1,\ldots,c_k} \frac{\sum_{x\in\mathcal{V}_1} \|x - c_1\|^2}{\|x_i - c(x_i)\|^2} + \sum_{j=2}^{k} \frac{\sum_{x\in\mathcal{V}_j} \|x - c_j\|^2}{\|x_i - c(x_i)\|^2}$$

To further bound this quantity, let us constrain the domain over which the minimum is sought. Consider $\mathcal{B}(x_i, R)$ the ball centered on x_i and radius $R \geq 0$. Consider $\mathcal{S}(x_i, R)$ its surface (*i.e.*, the associated sphere). We have that:

$$\min_{c_1,\ldots,c_k} \quad \leq \quad \min_{R\geq 0} \quad \min_{c_1\in\mathcal{S},(c_2,\ldots,c_k)\notin\mathcal{B}}$$

Given this restricted search space, we have: $c(x_i) = c_1$ and $\|x_i - c_1\|^2 = R^2$, and thus:

$$\frac{1}{\sigma_i} \quad \leq \quad \min_{R\geq 0} \frac{1}{R^2} \min_{c_1\in\mathcal{S}} \left(\sum_{x\in\mathcal{V}_1} \|x - c_1\|^2 + \min_{(c_2,\ldots,c_k)\notin\mathcal{B}} \sum_{j=2}^{k} \sum_{x\in\mathcal{V}_j} \|x - c_j\|^2\right)$$

Now, one may show, for all $j = 2, \ldots, k$, that:

$$\sum_{x\in\mathcal{V}_j} \|x - c_j\|^2 = \sum_{x\in\mathcal{V}_j} \left\|x - c_j^{\text{opt}}\right\|^2 + \#\mathcal{V}_j \left\|c_j - c_j^{\text{opt}}\right\|^2,$$

due to the fact that $c_j^{\text{opt}} = \frac{1}{\#\mathcal{V}_j}\sum_{x\in\mathcal{V}_j} x$. Given that the minimum of $\left\|c_j - c_j^{\text{opt}}\right\|^2$ is necessarily smaller than R^2:

$$\min_{c_j\notin\mathcal{B}} \sum_{x\in\mathcal{V}_j} \|x - c_j\|^2 \leq \sum_{x\in\mathcal{V}_j} \left\|x - c_j^{\text{opt}}\right\|^2 + \#\mathcal{V}_j R^2,$$

such that:

$$\frac{1}{\sigma_i} \leq \min_{R\geq 0} \frac{1}{R^2} \min_{c_1\in\mathcal{S}} \left(\sum_{x\in\mathcal{V}_1} \|x-c_1\|^2 + \alpha + (n-\#\mathcal{V}_1)R^2 \right)$$
$$= n - \#\mathcal{V}_1 + \min_{R\geq 0} \frac{1}{R^2} \min_{c_1\in\mathcal{S}} \left(\sum_{x\in\mathcal{V}_1} \|x-c_1\|^2 + \alpha \right)$$

with $\alpha = L^{\text{opt}\backslash\mathcal{V}}$ the optimal $(k-1)$-means cost on $\mathcal{X}\backslash\mathcal{V}$. Writing $x-c_1 = x-x_i-(c_1-x_i)$, we may decompose $\sum_{x\in\mathcal{V}_1}\|x-c_1\|^2$ in $R^2\#\mathcal{V}_1+\sum_{x\in\mathcal{V}_1}\|x-x_i\|^2-2R\left\|\sum_{x\in\mathcal{V}_1} x-x_i\right\|\cos\theta$, with θ the angle formed by $\sum_{x\in\mathcal{V}_1} x-x_i$ and c_1-x_i. As the minimum is sought for c_1 on the sphere, the angle θ may take any value, such that the minimum is always attained with θ s.t. $\cos\theta = 1$. We finally obtain, denoting $\forall x\in\mathcal{V}_1,\ y = x-x_i$:

$$\frac{1}{\sigma_i} \leq n + \min_{R\geq 0} \frac{1}{R^2} \left(\sum_{x\in\mathcal{V}_1} \|y\|^2 - 2R \left\| \sum_{x\in\mathcal{V}_1} y \right\| + \alpha \right).$$

Studying analytically the function $f(R) = \frac{a-2bR+\alpha}{R^2}$, its minimum is attained for $R^* = \frac{a+\alpha}{b}$ and $f(R^*) = -\frac{b^2}{a+\alpha}$, such that:

$$\frac{1}{\sigma_i} \leq n - \frac{||\sum_{x\in\mathcal{V}_1} y||^2}{\sum_{x\in\mathcal{V}_1}\|y\|^2 + \alpha} \leq n.$$

This is true for all i, and in particular for $\sigma_{\min}$. ■

Lemma 25 *With the notations of Section 4.2, the sensitivities in the linear regression problem verify:*

$$\forall i \qquad \sigma_i = x_i^\top \mathsf{H}^{-1} x_i + \frac{(y_i - y_i^*)^2}{\|y-y^*\|^2}$$

where $\mathsf{H} = \mathsf{X}^\top\mathsf{X}$ *and* y^* *reads* $y^* = \mathsf{X}\theta^* = \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y$. *Also:*

$$\mathfrak{S} = \sum_i \sigma_i = d+1.$$

As a remark, note that the sensitivity is different from the usual definition of leverage score in the context of linear regression, which simply reads $l_i = x_i^\top \mathsf{H}^{-1} x_i$ *(see, e.g., Hoaglin and Welsch, 1978; Chatterjee, 1988; Chatterjee and Hadi, 1986; Chen et al., 2016).*

Proof We have:

$$\frac{1}{\sigma_i} = \min_{\theta\in\Theta} \frac{\sum_j (y_j - x_j^\top\theta)^2}{(y_i - x_i^\top\theta)^2}$$

Let us write $\theta = u+v$ where u is colinear to x_i and v is orthogonal to x_i. We obtain:

$$\begin{aligned}\frac{1}{\sigma_i} &= \min_{u,v} \frac{\sum_j (y_j - x_j^\top(u+v))^2}{(y_i - x_i^\top u)^2} \\ &= \min_u \frac{1}{(y_i - x_i^\top u)^2} \left(\|y\|^2 + \min_v \left[(u+v)^\top \mathsf{H}(u+v) - 2(u+v)^\top \mathsf{X}^\top y \right] \right) \\ &= \min_u \frac{1}{(y_i - x_i^\top u)^2} \left(\|y\|^2 + u^\top\mathsf{H}u - 2u^\top\mathsf{X}^\top y + \min_v \left[v^\top\mathsf{H}v + 2u^\top\mathsf{H}v - 2v^\top\mathsf{X}^\top y \right] \right)\end{aligned}$$

where $\mathsf{H} = \sum_j x_j x_j^\top = \mathsf{X}^\top\mathsf{X}$. Let us first concentrate on solving:

$$\min_v v^\top\mathsf{H}v + 2u^\top\mathsf{H}v - 2v^\top\mathsf{X}^\top y = \min_v v^\top\mathsf{H}v + 2z^\top v$$

with $z = \mathsf{H}^\top u - \mathsf{X}^\top y$. The minimum is to be found for v orthogonal to x_i. We write the Lagrangian:

$$L(v,\lambda) = v^\top\mathsf{H}v + 2z^\top v - \lambda x_i^\top v.$$

We solve it wrt v:

$$2\mathsf{H}v + 2z - \lambda x_i = 0$$

i.e.:

$$v = \mathsf{H}^{-1}\left(\frac{\lambda}{2}x_i - z\right).$$

We know that v should be orthogonal to x_i such that:

$$0 = x_i^\top \mathsf{H}^{-1}\left(\frac{\lambda}{2}x_i - z\right)$$

i.e.:

$$\frac{\lambda}{2} = \frac{x_i^\top\mathsf{H}^{-1}z}{x_i^\top\mathsf{H}^{-1}x_i}.$$

We finally have:

$$v^* = \mathsf{H}^{-1}\left(\frac{x_i^\top\mathsf{H}^{-1}z}{x_i^\top\mathsf{H}^{-1}x_i}x_i - z\right)$$

and thus:

$$\min_v v^\top\mathsf{H}v + 2z^\top v = v^{*\top}\mathsf{H}v^* + 2z^\top v^* = \frac{(x_i^\top\mathsf{H}^{-1}z)^2}{x_i^\top\mathsf{H}^{-1}x_i} - z^\top\mathsf{H}^{-1}z.$$

i.e.:

$$\frac{1}{\sigma_i} = \min_u \frac{1}{(y_i - x_i^\top u)^2}\left(\|y\|^2 + \frac{(x_i^\top u - x_i^\top \mathsf{H}^{-1}\mathsf{X}^\top y)^2}{x_i^\top \mathsf{H}^{-1} x_i} - y^\top \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y\right)$$
$$= \frac{1}{x_i^\top \mathsf{H}^{-1} x_i} \min_u \frac{\left(\|y\|^2 - y^\top \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y\right) x_i^\top \mathsf{H}^{-1} x_i + \left(x_i^\top u - x_i^\top \mathsf{H}^{-1}\mathsf{X}^\top y\right)^2}{\left(y_i - x_i^\top u\right)^2}$$

Let us write $u = \alpha \frac{x_i}{\|x_i\|^2}$. We have:

$$\frac{1}{\sigma_i} = \frac{1}{x_i^\top \mathsf{H}^{-1} x_i} \min_\alpha \frac{a + (\alpha - b)^2}{(\alpha - c)^2}$$

with: $a = \left(\|y\|^2 - y^\top \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y\right) x_i^\top \mathsf{H}^{-1} x_i$, $b = x_i^\top \mathsf{H}^{-1}\mathsf{X}^\top y$ and $c = y_i$. The minimum of $f(\alpha) = \frac{a+(\alpha-b)^2}{(\alpha-c)^2}$ is attained for $\alpha^* = \frac{a}{b-c} + b$ which entails:

$$f(\alpha^*) = \frac{a}{a + (b-c)^2}.$$

And thus:

$$\frac{1}{\sigma_i} = \frac{\|y\|^2 - y^\top \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y}{\left(\|y\|^2 - y^\top \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y\right) x_i^\top \mathsf{H}^{-1} x_i + \left(x_i^\top \mathsf{H}^{-1}\mathsf{X}^\top y - y_i\right)^2}$$

i.e.:

$$\sigma_i = x_i^\top \mathsf{H}^{-1} x_i + \frac{\left(x_i^\top \mathsf{H}^{-1}\mathsf{X}^\top y - y_i\right)^2}{\|y\|^2 - y^\top \mathsf{X}\mathsf{H}^{-1}\mathsf{X}^\top y}.$$

Writing $\theta^* = \mathsf{H}^{-1}\mathsf{X}^\top y$ the least-square solution to the problem, this is re-written:

$$\sigma_i = x_i^\top \mathsf{H}^{-1} x_i + \frac{\left(x_i^\top \theta^* - y_i\right)^2}{\|y\|^2 - \theta^{*\top} \mathsf{H} \theta^*}.$$

Finally, denoting $y^* = \mathsf{X}\theta^*$:

$$\sigma_i = x_i^\top \mathsf{H}^{-1} x_i + \frac{(y_i - y_i^*)^2}{\|y\|^2 - \|y^*\|^2}$$
$$= x_i^\top \mathsf{H}^{-1} x_i + \frac{(y_i - y_i^*)^2}{\|y - y^*\|^2}$$

Thus:

$$\mathfrak{S} = \sum_i \sigma_i = \mathrm{Tr}(\mathsf{X}^\top \mathsf{H}^{-1}\mathsf{X}) + \sum_i \frac{(y_i - y_i^*)^2}{\|y - y^*\|^2}$$
$$= d + 1.$$

■

Appendix E. The issue of outliers

Corollary 10 is applicable to cases where $\sigma_{\max}$ is not too large. In fact, in order for $\alpha\sigma_i$ to be smaller than π_i, and thus smaller than 1 as π_i is a probability, α should always be set inferior to $\frac{1}{\sigma_{\max}}$. Now, if $\sigma_{\max}$ is so large that $\frac{1}{\sigma_{\max}} \leq \frac{32}{\epsilon^2}\mathfrak{S}\log\frac{4\eta}{\delta}$, then, even by setting β to its minimum value 1, there is no admissible α verifying both conditions (10) and (11). Large values of σ_i means strong outliers.[9] A simple workaround in this case is to separate the data in two: $\mathcal{X}_o = \{x_i \text{ s.t. } \sigma_i > \sigma^*\}$ the set of outliers and $\bar{\mathcal{X}} = \{x_i \text{ s.t. } \sigma_i \leq \sigma^*\}$ the others, where σ^* is the threshold sensitivity over which a data point is considered as an outlier (it is discussed in the following). The initial cost L may also be separated in two: $L = L_o + \bar{L}$ where

$$L_o = \sum_{x \in \mathcal{X}_o} f(x, \theta) \qquad \text{and} \qquad \bar{L} = \sum_{x \in \bar{\mathcal{X}}} f(x, \theta).$$

Let us write $\bar{\sigma}_i$ the sensitivity of data point i in $\bar{\mathcal{X}}$ and $\bar{\mathfrak{S}} = \sum_{x \in \bar{\mathcal{X}}} \bar{\sigma}_i$. Let us choose σ^* to be the largest value in $[0, 1]$ for which $\frac{1}{\bar{\sigma}_{\max}} \geq \frac{32}{\epsilon^2}\bar{\mathfrak{S}}\log\frac{4\eta}{\delta}$ is verified. One can thus apply the corollary to $\bar{\mathcal{X}}$ to obtain $\bar{\mathcal{S}}$ such that:

$$\forall \theta \in \Theta \qquad (1-\epsilon)\bar{L}(\bar{\mathcal{X}}, \theta) \leq \hat{\bar{L}}(\bar{\mathcal{S}}, \theta) \leq (1+\epsilon)\bar{L}(\bar{\mathcal{X}}, \theta).$$

Trivially, one may add to $\bar{\mathcal{S}}$ all outliers in $\mathcal{X}_o$ and associate to each of them a weight 1 in the estimated cost. The resulting set $\mathcal{S}$ is thus necessarily a coreset for all datapoints:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L \leq (1-\epsilon)\bar{L} + L_o \leq \hat{L} = \hat{\bar{L}} + L_o \leq (1+\epsilon)\bar{L} + L_o \leq (1+\epsilon)L.$$

The number of required samples is thus the number required for $\bar{\mathcal{S}}$ to be a coreset for $\bar{\mathcal{X}}$ plus the number of outliers in $\mathcal{X}_o$: $\mathcal{O}(|\mathcal{X}_o| + \frac{\bar{\mathfrak{S}}^2}{\epsilon^2}\log\frac{\eta}{\delta})$. The exact value of σ^* is application and data dependent. In general, we expect it to be $\mathcal{O}(1)$, such that the number of outliers $|\mathcal{X}_o|$ may be considered as a constant and the number of required samples is of the order $\mathcal{O}(\frac{\mathfrak{S}^2}{\epsilon^2}\log\frac{\eta}{\delta})$.

Appendix F. Implementation

F.1. Approximating the kernel via Random Fourier Features

In order to approximate L in time linear in n, we rely on random Fourier features (RFF) (Rahimi and Recht, 2008). We briefly recall the RFF framework in the following.

Let us write κ the Gaussian kernel that we use: $\kappa(\boldsymbol{t}) = \exp(-\boldsymbol{t}^2/2\tau^2)$. Its Fourier transform is:

$$\hat{\kappa}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} \kappa(\boldsymbol{t}) \exp^{-i\boldsymbol{\omega}^\intercal \boldsymbol{t}} \, \mathrm{d}\boldsymbol{t}.$$

It has real values as κ is symmetrical. One may write:

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \kappa(\boldsymbol{x} - \boldsymbol{y}) = \frac{1}{Z}\int_{\mathbb{R}^d} \hat{\kappa}(\boldsymbol{\omega}) \exp^{i\boldsymbol{\omega}^\intercal(\boldsymbol{x}-\boldsymbol{y})} \, \mathrm{d}\boldsymbol{\omega},$$

9. Sensitivities have indeed been shown to be good outlierness indicators (Lucic et al., 2016).

where, in order to ensure that $\kappa(\boldsymbol{x}, \boldsymbol{x}) = 1$:

$$Z = \int_{\mathbb{R}^d} \hat{\kappa}(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}.$$

According to Bochner's theorem, and due to the fact that κ is positive-definite, $\hat{\kappa}/Z$ is a valid probability density function. $\kappa(\boldsymbol{x}, \boldsymbol{y})$ may thus be interpreted as the expected value of $\exp^{i\boldsymbol{\omega}^\intercal(\boldsymbol{x}-\boldsymbol{y})}$ provided that $\boldsymbol{\omega}$ is drawn from $\hat{\kappa}/Z$:

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{\boldsymbol{\omega}} \left(\exp^{i\boldsymbol{\omega}^\intercal(\boldsymbol{x}-\boldsymbol{y})} \right)$$

The distribution $\hat{\kappa}/Z$ from which ω should be drawn from may be shown to be $\mathcal{N}(\omega; 0, 1/\tau^2)$, where $\mathcal{N}(x; \mu, v)$ is the normal law:

$$\mathcal{N}(x; \mu, v) = \frac{1}{\sqrt{2v\pi}} \exp^{-\frac{(x-\mu)^2}{2v}}.$$

In practice, we draw r random Fourier vectors from $\hat{\kappa}/Z$:

$$\Omega_r = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_r).$$

For each data point $\boldsymbol{x}_j$, we define a column feature vector associated to Ω_r:

$$\boldsymbol{\psi}_j = \frac{1}{\sqrt{r}} [\cos(\boldsymbol{\omega}_1^\intercal \boldsymbol{x}_j) | \cdots | \cos(\boldsymbol{\omega}_r^\intercal \boldsymbol{x}_j) | \sin(\boldsymbol{\omega}_1^\intercal \boldsymbol{x}_j) | \cdots | \sin(\boldsymbol{\omega}_r^\intercal \boldsymbol{x}_j)]^\intercal \in \mathbb{R}^{2r},$$

and call $\Psi = (\boldsymbol{\psi}_1 | \cdots | \boldsymbol{\psi}_n) \in \mathbb{R}^{2r \times n}$ the RFF matrix. Other embeddings are possible in the RFF framework, but this one was shown to be the most appropriate to the Gaussian kernel (Sutherland and Schneider, 2015). As r increases, $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ concentrates around its expected value: $\boldsymbol{\psi}_i^\intercal \boldsymbol{\psi}_j \simeq \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The Gaussian kernel matrix is thus approximated via:

$$\mathsf{L} \simeq \Psi^\intercal \Psi.$$

Computing the RFF matrix requires $\mathcal{O}(nrd)$ operations.

Remark 26 *How many random features r should we choose? Firstly, note that the entry-wise concentration of $\Psi^\intercal\Psi$ around its expected value L is controlled by a multiplicative error ϵ provided that $r \geq \mathcal{O}(d/\epsilon^2)$ (Rahimi and Recht, 2008). Thus, r should at least be of the order of the dimension d if one wants a proper approximation of the Gaussian kernel. However, this is not our goal here. In fact, what is needed is to obtain in average μ samples from a DPP with L-ensemble $\Psi^\intercal\Psi$. The maximum number of samples of such a DPP is the rank of Ψ, such that r should necessarily be chosen larger than μ. In the following, we thus set r to simply be a few times μ.*

F.2. Fast sampling of DPPs

In order to sample a DPP from a L-ensemble given its eigenvectors $\{\boldsymbol{u}_k\}$ and eigenvalues λ_k, one may follow Algorithm 1 of Kulesza and Taskar (2012), originally from Hough et al. (2006). This algorithm runs in $\mathcal{O}(n\mu^3)$ in average. The limiting step of the overall sampling

algorithm is the $\mathcal{O}(n^3)$ cost of the diagonalisation of L. Thankfully, the RFFs not only provide us with an approximation of L in linear time, it also provides us with a dual representation, *i.e.*, a representation of L in the form

$$\mathsf{L} = \Psi^\intercal \Psi.$$

Thus, we may circumvent the prohibitive diagonalization cost of L and only diagonalize its dual form:

$$\mathsf{C} = \Psi\Psi^\intercal \in \mathbb{R}^{2r \times 2r},$$

costing only $\mathcal{O}(nr^2) = \mathcal{O}(n\mu^2)$ (time to compute C from Ψ and to compute the low-dimensional diagonalization). C's eigendecomposition yields:

$$\mathsf{C} = \mathsf{V}\mathsf{D}\mathsf{V}^\intercal,$$

with $\mathsf{V} = (\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_{2r})$ the orthonormal basis of eigenvectors and D the diagonal matrix of eigenvalues such that $0 \leq \nu_1 \leq \ldots \leq \nu_{2r}$.

Note that all eigenvectors associated to non-zero eigenvalues of L can be recovered from C's eigendecomposition (see, *e.g.*, Kulesza and Taskar, 2012, Proposition 3.1). More precisely, if $\boldsymbol{v}_k$ is an eigenvector of C associated to eigenvalue ν_k, then:

$$\boldsymbol{u}_k = \frac{1}{\sqrt{\nu_k}} \Psi^\intercal \boldsymbol{v}_k$$

is a normalized eigenvector of L associated to the same eigenvalue.

In the case of such a dual representation, two standard approaches are used in the literature: 1) either follow Algorithm 1 of Kulesza and Taskar (2012) with the reconstructed eigenvectors $\mathsf{U} = \Psi^\intercal \mathsf{V}\mathsf{D}^{-1/2}$ as inputs, running in $\mathcal{O}(n\mu^3)$; 2) or follow Algorithm 3 of Kulesza and Taskar (2012) with the dual eigendecomposition $\{\boldsymbol{v}_k\}$ and $\{\nu_k\}$ as inputs, running in $\mathcal{O}(nr\mu^2 + r^2\mu^3)$. Both approaches are nevertheless suboptimal and we show in Tremblay et al. (2018) that the first (resp. second) one has an equivalent formulation running in $\mathcal{O}(n\mu^2)$ (resp. $\mathcal{O}(n\mu r)$). In this paper, we work with the following sampling strategy, given the dual eigendecomposition $\{\boldsymbol{v}_k\}$ and $\{\nu_k\}$:

i/ Sample eigenvectors. Draw n Bernoulli variables with parameters $\nu_k/(1+\nu_k)$: for $k = 1, \ldots, 2r$, add k to the set of sampled indices $\mathcal{J}$ with probability $\nu_k/(1+\nu_k)$. We generically denote by J the number of elements in $\mathcal{J}$. Note that the expected value of J is μ.

ii/ Run Algorithm 3 to sample a J-DPP with projective L-ensemble $\mathsf{P} = \mathsf{W}\mathsf{W}^\intercal$ where $\mathsf{W} \in \mathbb{R}^{n \times J}$ concatenates all the reconstructed eigenvectors $\boldsymbol{u}_k = \frac{1}{\sqrt{\nu_k}} \Psi^\intercal \boldsymbol{v}_k$ such that $k \in \mathcal{J}$.

The runtime of this strategy given the dual eigendecomposition is $\mathcal{O}(n\mu^2)$. Also, for a proof that this strategy does sample from a DPP with L-ensemble $\mathsf{L} = \Psi^\intercal\Psi$ we refer the reader to our technical report (Tremblay et al., 2018). These algorithms are implemented in the Julia toolbox we have developed for this work (DPP.jl available at `https://gricad-gitlab.univ-grenoble-alpes.fr/barthesi/dpp.jl`). Alternatively, one could use G. Gautier's well-documented Python toolbox DPPy (available at `http://github.com/guilgautier/DPPy`).

Algorithm 3 Efficient J-DPP sampling algorithm with projective L-ensemble $\mathsf{P} = \mathsf{W}\mathsf{W}^\intercal$

Input: $\mathsf{W} \in \mathbb{R}^{n \times J}$ such that $\mathsf{W}^\intercal \mathsf{W} = \mathsf{I}_J$
  Write $\forall i, \quad \boldsymbol{y}_i = \mathsf{W}^\intercal \boldsymbol{\delta_i} \in \mathbb{R}^J$.
  $\mathcal{S} \leftarrow \emptyset$
  Define $\boldsymbol{p} \in \mathbb{R}^n : \forall i, \quad p(i) = \|\boldsymbol{y}_i\|^2$
  for $n = 1, \ldots, J$ **do**:
    · Draw s_n with proba $\mathbb{P}(s) = p(s)/\sum_i p(i)$
    · $\mathcal{S} \leftarrow \mathcal{S} \cup \{s_n\}$
    · Compute $\boldsymbol{f}_n = \boldsymbol{y}_{s_n} - \sum_{l=1}^{n-1} \boldsymbol{f}_l(\boldsymbol{f}_l^\intercal \boldsymbol{y}_{s_n}) \in \mathbb{R}^J$
    · Normalize $\boldsymbol{f}_n \leftarrow \boldsymbol{f}_n / \sqrt{\boldsymbol{f}_n^\intercal \boldsymbol{y}_{s_n}}$
    · Update $\boldsymbol{p} : \forall i \quad p(i) \leftarrow p(i) - (\boldsymbol{f}_n^\intercal \boldsymbol{y}_i)^2$
  end for
Output: $\mathcal{S}$ of size J.

F.3. Fast sampling of m-DPPs

In the experiments, we will only provide results for m-DPP sampling. In fact, results are easier to compare with classical i.i.d. coreset methods when the number of samples is fixed and not random. Given the eigendecomposition of the dual representation C, one samples a m-DPP via the following two steps (we refer once again to Tremblay et al. (2018) for a proof):

i/ Sample m eigenvectors. Draw $2r$ Bernoulli variables with parameters $\nu_k/(1+\nu_k)$ under the constraint that exactly m variables should be equal to one. Call $\mathcal{J}$ the set of indices thus drawn: $|\mathcal{J}| = J = m$.

ii/ Run Algorithm 3 to sample a J-DPP with projective L-ensemble $\mathsf{P} = \mathsf{W}\mathsf{W}^\intercal$ where $\mathsf{W} \in \mathbb{R}^{n \times J}$ concatenates all the reconstructed eigenvectors $\boldsymbol{u}_k = \frac{1}{\sqrt{\nu_k}} \Psi^\intercal \boldsymbol{v}_k$ such that $k \in \mathcal{J}$.

The only difference with a usual DPP is in the first step, where the n Bernoulli variables are not drawn independently anymore, but under constraint that exactly m of them should be equal to one. To do so, one may follow Algorithm 8 of Kulesza and Taskar (2012) which runs in $\mathcal{O}(nm)$. Step ii/ runs in $\mathcal{O}(nm^2)$, such that the overall cost of sampling a m-DPP given the dual eigendecomposition is also $\mathcal{O}(nm^2)$. Algorithm 8 of Kulesza and Taskar (2012) makes use of elementary polynomials. Given the eigenvalues of C, $\{\nu_i\}$, the p-th order associated elementary polynomial reads:

$$e_p(\nu_1, \ldots, \nu_{2r}) = \sum_{\mathcal{J} \subseteq \{1,2,\ldots,2r\} \text{ s.t. } |\mathcal{J}| = p} \prod_{j \in \mathcal{J}} \nu_j \qquad \in \mathbb{R}.$$

As r increases, these polynomials become less and less stable to compute and Algorithm 8 of Kulesza and Taskar (2012) fails in many practical situations due to numerical precision errors as m becomes too large. In order to avoid these errors, we follow the saddle-point approximation method detailed by Barthelmé et al. (2019). This method has the additional advantage of providing very accurate approximations of the probabilities of inclusion of

the m-DPP (that are exactly written as a ratio of elementary polynomials and thus also vulnerable to numerical instability). We in fact need these marginals for the importance sampling estimator.

References

Emmanuel Abbe and Colin Sandon. Community Detection in General Stochastic Block models: Fundamental Limits and Efficient Algorithms for Recovery. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 670–688. IEEE, 2015.

Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte Carlo Markov Chain Algorithms for Sampling Strongly Rayleigh Distributions and Determinantal Point Processes. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 103–115, Columbia University, New York, New York, USA, June 2016. PMLR. URL `http://proceedings.mlr.press/v49/anari16.html`.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Olivier Bachem, Mario Lucic, and Andreas Krause. Practical Coreset Constructions for Machine Learning. *arXiv:1703.06476 [stat]*, March 2017. URL `http://arxiv.org/abs/1703.06476`. arXiv: 1703.06476.

Mihai Badoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Computational Geometry*, 40(1):14–22, May 2008. ISSN 09257721. doi: 10.1016/j.comgeo.2007.04.002. URL `http://linkinghub.elsevier.com/retrieve/pii/S0925772107000454`.

Maria-Florina F Balcan, Steven Ehrlich, and Yingyu Liang. Distributed k-means and k-median Clustering on General Topologies. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1995–2003. Curran Associates, Inc., 2013. URL `http://papers.nips.cc/paper/5096-distributed-k-means-and-k-median-clustering-on-general-topologies.pdf`.

Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A Simple Proof of the Restricted Isometry Property for Random Matrices. *Constructive Approximation*, 28(3):253–263, December 2008. ISSN 0176-4276, 1432-0940. doi: 10.1007/s00365-007-9003-x. URL `http://link.springer.com/10.1007/s00365-007-9003-x`.

Yannick Baraud. A Bernstein-type inequality for suprema of random processes with applications to model selection in non-Gaussian regression. *Bernoulli*, 16(4):1064–1085, 2010. doi: 10.3150/09-BEJ245. URL `https://doi.org/10.3150/09-BEJ245`.

Rémi Bardenet and Adrien Hardy. Monte Carlo with determinantal point processes. *arXiv preprint arXiv:1605.00361*, 2016.

Simon Barthelmé, Pierre-Olivier Amblard, and Nicolas Tremblay. Asymptotic equivalence of fixed-size and varying-size determinantal point processes. *Bernoulli*, 25(4B):3555–3589, 2019. doi: 10.3150/18-BEJ1102. URL https://doi.org/10.3150/18-BEJ1102.

Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for k -means: beyond subspaces and the Johnson-Lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing - STOC 2019*, pages 1039–1050, Phoenix, AZ, USA, 2019. ACM Press. ISBN 978-1-4503-6705-9. doi: 10.1145/3313276.3316318. URL http://dl.acm.org/citation.cfm?doid=3313276.3316318.

M.-A. Belabbas and P. J. Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of the National Academy of Sciences*, 106(2):369–374, January 2009. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0810600105. URL http://www.pnas.org/cgi/doi/10.1073/pnas.0810600105.

Ayoub Belhadji, Rémi Bardenet, and Pierre Chainais. A determinantal point process for column subset selection. *arXiv:1812.09771 [cs, stat]*, December 2018. URL http://arxiv.org/abs/1812.09771. arXiv: 1812.09771.

Christos. Boutsidis and Alex. Gittens. Improved Matrix Algorithms via the Subsampled Randomized Hadamard Transform. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1301–1340, 2013. doi: 10.1137/120874540. URL https://doi.org/10.1137/120874540.

Christos Boutsidis, Anastasios Zouzias, Michael W. Mahoney, and Petros Drineas. Randomized Dimensionality Reduction for k -Means Clustering. *IEEE Transactions on Information Theory*, 61(2):1045–1062, February 2015. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2014.2375327. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6967844.

Vladimir Braverman, Dan Feldman, and Harry Lang. New Frameworks for Offline and Streaming Coreset Constructions. *arXiv preprint arXiv:1612.00889*, 2016. URL https://arxiv.org/abs/1612.00889.

Peter G Casazza and Gitta Kutyniok. *Finite frames: Theory and applications*. Springer, 2012.

Samprit Chatterjee. *Sensitivity Analysis in Linear Regression*. John Wiley and Sons, Inc., New York, NY, USA, 1988. ISBN 0-471-82216-7.

Samprit Chatterjee and Ali S. Hadi. Influential Observations, High Leverage Points, and Outliers in Linear Regression. *Statistical Science*, 1(3):379–393, 1986. doi: 10.1214/ss/1177013622. URL https://doi.org/10.1214/ss/1177013622.

Ke Chen. On Coresets for k -Median and k -Means Clustering in Metric and Euclidean Spaces and Their Applications. *SIAM Journal on Computing*, 39(3):923–947, January 2009. ISSN 0097-5397, 1095-7111. doi: 10.1137/070699007. URL http://epubs.siam.org/doi/10.1137/070699007.

Siheng Chen, Rohan Varma, Aarti Singh, and Jelena Kovacevi. A statistical perspective of sampling scores for linear regression. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1556–1560, July 2016. doi: 10.1109/ISIT.2016.7541560. ISSN: 2157-8117.

Kenneth L. Clarkson. Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm. *ACM Trans. Algorithms*, 6(4):63:1–63:30, September 2010. ISSN 1549-6325. doi: 10.1145/1824777.1824783. URL http://doi.acm.org/10.1145/1824777.1824783.

Kenneth L. Clarkson and David P. Woodruff. Low-Rank Approximation and Regression in Input Sparsity Time. *J. ACM*, 63(6):54:1–54:45, January 2017. ISSN 0004-5411. doi: 10.1145/3019134. URL http://doi.acm.org/10.1145/3019134.

Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality Reduction for k-Means Clustering and Low Rank Approximation. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 163–172, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3536-2. doi: 10.1145/2746539.2746569. URL http://doi.acm.org/10.1145/2746539.2746569. event-place: Portland, Oregon, USA.

Martin S. Copenhaver, Yeon Hyang Kim, Cortney Logan, Kyanne Mayfield, Sivaram K. Narayan, Matthew J. Petro, and Jonathan Sheperd. Diagram vectors and tight frame scaling in finite dimensions. *Operators and Matrices*, (1):73–88, 2014. ISSN 1846-3886. doi: 10.7153/oam-08-02. URL http://oam.ele-math.com/08-02.

W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation Schemes for Clustering Problems. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 50–58, New York, NY, USA, 2003. ACM. ISBN 1-58113-674-9. doi: 10.1145/780542.780550. URL http://doi.acm.org/10.1145/780542.780550.

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborov. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E*, 84(6):066106, December 2011.

Michal Derezinski and Manfred K Warmuth. Reverse Iterative Volume Sampling for Linear Regression. *Journal of Machine Learning Research*, 19(23), 2018.

Michal Derezinski, Manfred K. K Warmuth, and Daniel J Hsu. Leveraged volume sampling for linear regression. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2505–2514. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7517-leveraged-volume-sampling-for-linear-regression.pdf.

Michal Derezinski, Daniele Calandriello, and Michal Valko. Exact sampling of determinantal point processes with sublinear time preprocessing. *arXiv:1905.13476 [cs, stat]*, July 2019. URL http://arxiv.org/abs/1905.13476. arXiv: 1905.13476.

Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 329–338. IEEE, 2010. URL http://ieeexplore.ieee.org/abstract/document/5671202/.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1117–1126. Society for Industrial and Applied Mathematics, 2006. URL http://dl.acm.org/citation.cfm?id=1109681.

Petros Drineas and Michael W Mahoney. Lectures on randomized numerical linear algebra. *The Mathematics of Data*, 25:1, 2018.

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

Charles F Dunkl and Yuan Xu. *Orthogonal polynomials of several variables*. Number 155. Cambridge University Press, 2014.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578. ACM, 2011. URL http://dl.acm.org/citation.cfm?id=1993712.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

Guillaume Gautier, Rémi Bardenet, and Michal Valko. Zonotope Hit-and-run for Efficient Sampling from Projection DPPs. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1223–1232. JMLR.org, 2017. URL http://dl.acm.org/citation.cfm?id=3305381.3305508. event-place: Sydney, NSW, Australia.

Sara A Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

Jennifer Gillenwater, Alex Kulesza, Zelda Mariet, and Sergei Vassilvtiskii. A Tree-Based Method for Fast Repeated Sampling of Determinantal Point Processes. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2260–2268, Long Beach, California, USA, June 2019. PMLR. URL http://proceedings.mlr.press/v97/gillenwater19a.html.

Sariel Har-Peled. *Geometric approximation algorithms*. Number 173. American Mathematical Soc., 2011.

Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 126–134. ACM, 2005. URL http://dl.acm.org/citation.cfm?id=1064114.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300. ACM, 2004.

Jonathan Hermon and Justin Salez. Modified log-Sobolev inequalities for strong-Rayleigh measures. *arXiv:1902.02775 [math]*, February 2019. URL http://arxiv.org/abs/1902.02775. arXiv: 1902.02775.

David C. Hoaglin and Roy E. Welsch. The Hat Matrix in Regression and ANOVA. *The American Statistician*, 32(1):17–22, 1978. doi: 10.1080/00031305.1978.10479237. URL https://amstat.tandfonline.com/doi/abs/10.1080/00031305.1978.10479237.

J. Ben Hough, Manjunath Krishnapur, Yuval Peres, and Blint Virg. Determinantal Processes and Independence. *Probability Surveys*, 3(0):206–229, 2006. ISSN 1549-5787. doi: 10.1214/154957806000000078. URL http://projecteuclid.org/euclid.ps/1146832696.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

N. Keriven, N. Tremblay, Y. Traonmilin, and R. Gribonval. Compressive K-means. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6369–6373, March 2017. doi: 10.1109/ICASSP.2017.7953382.

A. Kroo and D. S. Lubinsky. Christoffel Functions and Universality in the Bulk for Multivariate Orthogonal Polynomials. *Canadian Journal of Mathematics*, 65(3):600–620, June 2013. ISSN 0008-414X, 1496-4279. doi: 10.4153/CJM-2012-016-x. URL https://www.cambridge.org/core/product/identifier/S0008414X00002236/type/journal_article.

Alex Kulesza and Ben Taskar. Determinantal Point Processes for Machine Learning. *Foundations and Trends in Machine Learning*, 5(23):123–286, 2012. ISSN 1935-8237. doi: 10.1561/2200000044. URL http://dx.doi.org/10.1561/2200000044.

Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the ACM (JACM)*, 57(2):5, 2010.

Michael Langberg and Leonard J. Schulman. Universal ϵ-approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 598–607. SIAM, 2010.

Jean B Lasserre and Edouard Pauwels. The empirical christoffel function in statistics and machine learning. *arXiv preprint arXiv:1701.02886*, 2017.

Yann LeCun. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Yi Li, Philip M. Long, and Aravind Srinivasan. Improved Bounds on the Sample Complexity of Learning. *Journal of Computer and System Sciences*, 62(3):516 – 527, 2001. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.2000.1741. URL http://www.sciencedirect.com/science/article/pii/S0022000000917410.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489.

Mario Lucic, Olivier Bachem, and Andreas Krause. Linear-time Outlier Detection via Sensitivity. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 1795–1801. AAAI Press, 2016. ISBN 978-1-57735-770-4. URL http://dl.acm.org/citation.cfm?id=3060832.3060872. event-place: New York, New York, USA.

Michael W. Mahoney. Randomized Algorithms for Matrices and Data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011. ISSN 1935-8237. doi: 10.1561/2200000035. URL http://dx.doi.org/10.1561/2200000035.

Konstantin Makarychev, Yury Makarychev, Maxim Sviridenko, and Justin Ward. A Bi-Criteria Approximation Algorithm for k-Means. In Klaus Jansen, Claire Mathieu, Jos D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, volume 60 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:20, Dagstuhl, Germany, 2016. Schloss DagstuhlLeibniz-Zentrum fuer Informatik. ISBN 978-3-95977-018-7. doi: 10.4230/LIPIcs.APPROX-RANDOM.2016.14. URL http://drops.dagstuhl.de/opus/volltexte/2016/6637.

M. Muja and D. G. Lowe. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, November 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2321376.

Alexander Munteanu and Chris Schwiegelshohn. Coresets-Methods and History: A Theoreticians Design Pattern for Approximation and Streaming Algorithms. *KI - Knstliche Intelligenz*, December 2017. ISSN 0933-1875, 1610-1987. doi: 10.1007/s13218-017-0519-3. URL http://link.springer.com/10.1007/s13218-017-0519-3.

A.Y. Ng, M.I. Jordan, Y. Weiss, and others. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

Robin Pemantle and Yuval Peres. Concentration of Lipschitz Functionals of Determinantal and Other Strong Rayleigh Measures. *Combinatorics, Probability and Computing*, 23(1):140–160, 2014. doi: 10.1017/S0963548313000345.

Jeff M. Phillips. Coresets and Sketches. *arXiv:1601.00617 [cs]*, January 2016. URL http://arxiv.org/abs/1601.00617. arXiv: 1601.00617.

Jack Poulson. High-performance sampling of generic Determinantal Point Processes. *arXiv:1905.00165 [cs, math]*, August 2019. URL http://arxiv.org/abs/1905.00165. arXiv: 1905.00165.

A. Prymak. Upper estimates of Christoffel function on convex domains. *Journal of Mathematical Analysis and Applications*, 455(2):1984 – 2000, 2017. ISSN 0022-247X. doi: https://doi.org/10.1016/j.jmaa.2017.06.079. URL http://www.sciencedirect.com/science/article/pii/S0022247X17306418.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

Dougal J. Sutherland and Jeff Schneider. On the Error of Random Fourier Features. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 862–871, Arlington, Virginia, United States, 2015. AUAI Press. ISBN 978-0-9966431-0-8. URL http://dl.acm.org/citation.cfm?id=3020847.3020936. event-place: Amsterdam, Netherlands.

Nicolas Tremblay and Andreas Loukas. Approximating Spectral Clustering via Sampling: A Review. In Frédéric Ros and Serge Guillaume, editors, *Sampling Techniques for Supervised or Unsupervised Tasks*, pages 129–183. Springer International Publishing, Cham, 2020. ISBN 978-3-030-29349-9. doi: 10.1007/978-3-030-29349-9_5. URL https://doi.org/10.1007/978-3-030-29349-9_5.

Nicolas Tremblay, Simon Barthelme, and Pierre-Olivier Amblard. Optimized Algorithms to Sample Determinantal Point Processes. *arXiv:1802.08471 [cs, stat]*, February 2018. URL http://arxiv.org/abs/1802.08471. arXiv: 1802.08471.

J.A. Tropp, I.S. Dhillon, and R.W. Heath. Finite-Step Algorithms for Constructing Optimal CDMA Signature Sequences. *IEEE Transactions on Information Theory*, 50(11):2916–2921, November 2004. ISSN 0018-9448. doi: 10.1109/TIT.2004.836698. URL http://ieeexplore.ieee.org/document/1347383/.

Andrea Vedaldi and Brian Fulkerson. Vlfeat: An Open and Portable Library of Computer Vision Algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1469–1472, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1874249. URL http://doi.acm.org/10.1145/1873951.1874249. event-place: Firenze, Italy.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. ISSN 0960-3174.

David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(12):1–157, 2014.

Journal of Machine Learning Research 20 (2019) 1-26 Submitted 6/18; Revised 4/19; Published 11/19

Embarrassingly Parallel Inference for Gaussian Processes

Michael Minyi Zhang MZ8@CS.PRINCETON.EDU
Department of Computer Science
Princeton University
Princeton, NJ 08544, USA

Sinead A. Williamson SINEAD.WILLIAMSON@MCCOMBS.UTEXAS.EDU
Department of Statistics and Data Science
Department of Information, Risk and Operations Management
The University of Texas at Austin
Austin, TX 78712, USA

Editor: Manfred Opper

Abstract

Training Gaussian process-based models typically involves an $O(N^3)$ computational bottleneck due to inverting the covariance matrix. Popular methods for overcoming this matrix inversion problem cannot adequately model all types of latent functions, and are often not parallelizable. However, judicious choice of model structure can ameliorate this problem. A mixture-of-experts model that uses a mixture of K Gaussian processes offers modeling flexibility and opportunities for scalable inference. Our embarrassingly parallel algorithm combines low-dimensional matrix inversions with importance sampling to yield a flexible, scalable mixture-of-experts model that offers comparable performance to Gaussian process regression at a much lower computational cost.

Keywords: Gaussian process, parallel inference, machine learning, Bayesian non-parametrics.

1. Introduction

Gaussian processes (GPs) provide a flexible family of distributions over functions, that have been widely adopted for problems including regression, classification and optimization due to their ease of use in modeling latent functions. Additional flexibility can be achieved using mixture-of-experts models (Gramacy and Lee, 2008; Rasmussen and Ghahramani, 2002; Meeds and Osindero, 2006), which use a mixture of Gaussian processes. Different mixture components can have different covariance patterns, allowing for non-stationarity in the resulting function without resorting to explicitly non-stationary covariance functions.

Unfortunately, the flexibility of Gaussian processes and related models comes at a cost. Inference in GP models with N observations involves repeated inversion of an $N \times N$ matrix, which typically scales $O\left(N^3\right)$. Mixture-of-experts models fare slightly better since, conditioned on the partition, we are inverting a block-diagonal matrix; however the computational savings are tempered by the cost of averaging over partitions, typically using MCMC. This computational bottleneck has thus far prevented Gaussian processes and mixtures-of-experts from being used in so-called "big data" situations.

©2019 Michael Minyi Zhang and Sinead Williamson.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-374.html.

Two main approaches have been proposed to ameliorate the computational complexity for inference in the simple Gaussian process regression setting: sparse methods, that aim to reduce the size of the matrix to be inverted, and local methods, that aim to simplify its structure. Unfortunately, both methods exhibit key failure modes as we reduce the computational cost: local methods can miss long-range correlations, and sparse methods tend to miss short-range fluctuations. Further, methods of these types are not typically parallelizable to run efficiently on a distributed architecture.

In this paper, we propose a novel inference algorithm for fitting mixtures of partitioned Gaussian processes that is flexible and easily distributed. The "Importance Sampled Mixture of Experts" (IS-MOE) uses importance sampling to average over Gaussian processes with block-diagonal covariance matrices. A location-based distribution over partitions allows us to capture non-stationarity. We learn, in parallel, multiple partitioned Gaussian processes sampled from the conditional distribution over partitions given the input locations thereby allowing us to take advantage of the lower inversion cost of a block-diagonal matrix. Minibatch-based stochastic approximations reduce the size of these blocks further, while maintaining competitive performance. Importance weights are also calculated in a distributed manner, with the only global communication occurring when the importance-weighted samples are combined. The resulting posterior predictive distribution has a more expressive expected covariance matrix than a single block-diagonal matrix, avoiding edge effects common with local methods and allowing for an expressive covariance structure that can model both long- and short-range covariance as well as non-stationary behavior in the latent function.

We start in Section 2 by reviewing the Gaussian process and describing existing methods for scaling inference. We then describe our approach in Section 3, before presenting detailed experimental evaluation in Section 4 to showcase its efficacy versus existing methods. Further extensions and applications are discussed in Section 5.

2. Background

2.1. Gaussian processes and related models

A Gaussian process is a distribution over functions $f : \mathbb{R}^D \to \mathbb{R}$, parametrized by some mean function $m(x)$, typically taken as zero, and a covariance function $\Sigma(x, x')$. For a given m and Σ, a GP is a unique distribution over functions f such that for any finite set of points, $x_1, \ldots, x_N \in \mathbb{R}^D$, the function evaluated at those points is multivariate normally distributed with mean and covariance given by m and Σ evaluated at these inputs.

This distribution over functions can be used in a variety of applications, including regression, classification and optimization (see for example Rasmussen and Williams, 2006; Snoek et al., 2012; Wang et al., 2005). For simplicity, we focus here on the regression setting, where a function $f \sim \mathrm{GP}(0, \Sigma)$ maps our inputs $X = (x_i)_{i=1}^N$ to our outputs $Y = (y_i)_{i=1}^N$, such that $y_i \sim \mathrm{N}(f(x_i), \sigma^2)$.

In this setting, the posterior distribution of f given X, Y and Σ is analytically tractable, and the inference challenge reduces to inferring the hyperparameters, Θ, that control the form of the covariance function. Optimizing or sampling these hyperparameters involves inverting the covariance matrix $\mathbf{\Sigma}$ obtained by evaluating $\Sigma(\cdot, \cdot)$ at the inputs $x_1, \ldots, x_N$. In general, the computational cost of inverting this matrix is $O(N^3)$.

Mixture-of-experts models (Jacobs et al., 1991) are a hierarchical extension of Gaussian processes, that model each output y_i using a mixture of Gaussian processes. This mixture can be specified either by a distribution over partitions of the input space (Rasmussen and Ghahramani, 2002; Gramacy and Lee, 2008; Yuan and Neubauer, 2009) or on the joint space of inputs and outputs (Meeds and Osindero, 2006). This offers two advantages over a single Gaussian process. First, each composite Gaussian process can have a separate covariance function, allowing us to capture different behaviors in different regions. Second, conditioned on the partition we have K independent Gaussian processes with average size N/K, reducing the computational cost of matrix inversion.

Unfortunately, this computational advantage is counterbalanced by the computational cost of inferring the distribution over partitions, which is done using either MCMC (Rasmussen and Ghahramani, 2002; Gramacy and Lee, 2008; Meeds and Osindero, 2006) or variational methods (Yuan and Neubauer, 2009). As a result, mixture of expert models are typically not considered "scalable". Performing MCMC-based inference over partitions can be expensive and while variational methods are generally faster, the Markovian relationship between samples precludes direct parallelization.

2.2. Scalable inference methods for Gaussian processes

Most scalable inference approaches focus on reducing the $O(N^3)$ cost of covariance matrix inversion. Two broad classes of methods have been proposed: "sparse" methods which parametrize the covariance based on $M << N$ inducing inputs, and "local" methods that replace the dense $N \times N$ covariance matrix with a block-diagonal matrix.

Sparse GP approximations parameterize the covariance matrix of the GP model with M pseudo-inputs, where $M << N$. The pseudo-input locations are chosen so that the posterior function evaluated at these points is a good approximation to the true posterior, for example by maximum likelihood optimization (Snelson and Ghahramani, 2005) or variational inference (Titsias, 2009). The computational saving comes from replacing an $N \times N$ covariance matrix with an $M \times M$ matrix. In the regression case, this reduces the training cost to $O(NM^2)$. Further computational savings can be obtained by using stochastic variational inference (SVI) to update the inducing points by calculating necessary gradients based only on size-B subsets of the N datapoints, reducing computational cost to $O\left(M^2 \min\{M, B\}\right)$ (Hensman et al., 2013). While sparse methods can yield impressive speed-ups, they tend to have a decreased ability to model high-frequency fluctuations in the function, since the number of inducing points limits the amount of variation we can capture. Additionally, as with the full-covariance GP it approximates, the sparse GP cannot naturally model non-stationary data without resorting to a non-stationary kernel.

Local Gaussian process methods make local approximations to the dense covariance matrix so that a low-rank representation of the covariance matrix is inverted instead of the full-rank matrix. Mixture-of-experts models, described above, fall under this framework, since conditioned on the partition we have a block-diagonal covariance matrix; however the cost of averaging over partitions means these are not generally seen as scalable models. Product-of-experts models (Tresp, 2000; Cao and Fleet, 2014; Ng and Deisenroth, 2014; Deisenroth and Ng, 2015) avoid this by using a single partition, and avoid edge effects by multiplying the predictions of the local Gaussian processes. Conditioned on the partitioning,

inference in the local GP scales approximately as $O(N^3/K^2)$, since we need to invert K matrices of average size $\frac{N}{K} \times \frac{N}{K}$. Park et al. (2011) also use a block-diagonal approximation, and use a boundary value function to ensure continuity between regions.

Taking a different perspective, Kaufman et al. (2008) applies a "tapering" function to the covariance matrix so that observation pairs with low correlation are set to zero and provides theorems for estimator consistency when the covariance function used is a Matérn kernel. Gramacy and Apley (2015) try to learn the local approximation by taking the n-nearest neighbors of a predictive value X^* to the data X and learns both the function hyperparameters and predictive distribution jointly by iteratively increasing the size of the nearest neighbors until a stopping criteria is satisfied for all predictive inputs.

As noted in Low et al. (2015), local methods will be good at capturing short-range correlations, where the correlation structure is well approximated. Further, if a block-diagonal covariance is used, they allow us to use different covariance hyperparameters in different blocks, capturing behavior which is locally approximately stationary, but where the lengthscale varies across the input space (Tresp, 2000; Rasmussen and Ghahramani, 2002). This is in contrast with sparse methods, where the number of inducing points limits the ability to learn very short-range correlations, and which can only capture non-stationarity if we use an explicitly non-stationary covariance function.

The disadvantage of the local methods, however, is that they risk ignoring important correlations. For example, the block-diagonal approaches assume zero correlation between different blocks in the partition. If the data points are partitioned based on location, this means that long-range correlations will be ignored; if they are partitioned randomly, the model will tend to perform poorly if the number of observations in some region of $\mathbb{R}^D$ is low. Moreover, Liu et al. (2018) show that local methods like the RBCM will systematically produce overconfident predictions which violates one of the major benefits of using Bayesian methods in the first place–that being proper uncertainty quantification.

2.3. Distributed inference for Gaussian processes

The sparse and local approximations described above aim to reduce the overall computational burden by reducing the size of matrices to be inverted. When run on a single machine, this reduction in computational cost leads directly to faster inference. However, we may also be interested in distributing computation cost across multiple threads or machines. Even if the total computational cost is the same, we can reduce total time by distributing computation onto multiple parallel threads. Alternatively, if we increase the computational budget then we may be able to improve our posterior estimate by running multiple samplers in parallel and then combining the results without increasing the time budget.

Local partition-based GP methods that do not average over partitions, such as product-of-experts models are well suited to this sort of parallelism. They split a single GP problem into K independent problems whose parameters can be inferred in parallel. We only need to communicate between the K subproblems at the end when we combine their predictions. This type of algorithm, where global communication occurs only once after all the local computation is complete, is known as "embarrassingly parallel". Ng and Deisenroth (2014) exploit these independences, in a weighted product-of-experts model, to obtain a distributable algorithm appropriate for large datasets.

2.4. Fast Bayesian inference via stochastic approximations

When performing Bayesian inference on large datasets, much of the computational cost is due to calculating functions of the data – for example, gradients or likelihoods. One way to reduce computational costs is to approximate these functions using noisy estimates based on much smaller subsets of the data. The intuition here is that much of the data at hand is "redundant" for learning the posterior so it is more efficient from a computational and memory perspective to perform Bayesian inference on a subset of the data. For example, stochastic variational inference (Hoffman et al., 2013) uses minibatches of data to approximate gradients in a variational context. Stochastic gradient MCMC methods (Ma et al., 2015; Welling and Teh, 2011) perform a similar approximation in a gradient-based MCMC setting. In a Gaussian process context, as mentioned in Section 2.2, SVI has been used to speed up inference in sparse Gaussian processes from $O(NM^2)$ to $O\left(M^2 \min\{M, B\}\right)$, where B is the minibatch size.

An alternative is to use a minibatch to approximate the full posterior. Several embarrassingly parallel MCMC methods combine noisy posterior estimates obtained using subsets of the data (Minsker et al., 2014). Srivastava et al. (2015) show that such stochastic approximation of sub-posteriors is strongly consistent. The Bayesian coresets approach aims to learn the posterior based on a reweighted posterior (Huggins et al., 2016). While not directly equivalent (since it uses a single subset), Banerjee et al. (2008) approximates a full Gaussian process model using a smaller subset of the data to form a prediction of the entire model.

3. Embarrassingly parallel inference with importance sampled mixture of experts

We now introduce our novel method of fitting mixtures of Gaussian processes. We assume our covariates $X = \{x_1, \ldots, x_N\}$ are distributed according to a Dirichlet mixture of K Gaussian components,

$$\begin{aligned} x_i \sim \text{Normal}(\mu_{z_i}, \Gamma_{z_i}), \quad (\mu_k, \Gamma_k) \sim \text{Normal-Inv. Wishart}(\mu_0, \lambda, \Psi, \nu) \\ z_i \sim \text{Categorical}(\pi), \quad \pi \sim \text{Dirichlet}(\alpha). \end{aligned} \tag{1}$$

The outputs are then assumed to be generated by K independent Gaussian processes,

$$\{y_i : z_i = k\} \,|\, (\{x_i : z_i = k\}, \Theta_k) \sim \text{GP}(0, \Sigma_{\Theta_k}), \tag{2}$$

where Σ_{Θ_k} is the covariance matrix between $\{x_i : z_i = k\}$ parametrized by Θ_k. Conditioned on the z_i, we have a simple local GP with block diagonal structure, of the sort considered in Section 2.2. Rather than invert the $N \times N$ covariance matrix, we only need invert the K blocks. These can be inverted in parallel using K threads, each costing $O(N^3/K^2)$.

Marginalizing over the z_i in Equation 1 to give a mixture-of-experts model avoids the key limitation of the local GP methods: that they ignore correlation between the fixed blocks. In a setting like ours, where the inputs are clustered based on location, this means we ignore long-range correlation. Conversely, a mixture-of-experts approach allows long-range correlations and yields a dense expected covariance matrix.

The typical mixture of experts approach of marginalizing over partitions using MCMC is expensive (due to slow mixing) and difficult to parallelize. Instead, we use a trivially parallelizable importance sampling scheme. In short, we independently sample J partitions of the input space, conditioned on the covariates X, by sampling from a Dirichlet mixture of K Gaussians, conditioned on the inputs X and ignoring the outputs Y,

$$P(z_i = k|-) \propto \pi_k P(X_i|\mu_k, \Sigma_k),$$

where the mixture parameters are drawn from the prior distribution $P(\mu, \Sigma)$, which we assume is Normal-Inverse Wishart.

We then fit independent Gaussian processes to each of the K partitions of each of the J samples. We then (independently) calculate the appropriate importance weights, and use these weights to combine the J samples. If our we assume the latent function we are modeling is generated from a mixture of GP functions, then our proposed method is an exact algorithm for fitting the model and calculating the marginal likelihood. However, if the latent function is a single GP then our algorithm is a fast approximation of the full GP and its marginal likelihood.

To further reduce memory and computational constraints, we can sample size $B << N$ minibatches of the data without replacement and approximate the full likelihood by raising it to the N/B power. We detail these steps below in Sections 3.2 and 3.3, and provide a summary of the process in Algorithm 1.

Algorithm 1: Importance Sampled Mixture of Experts (IS-MOE)

for $j = 1, \ldots, J$ *in parallel* **do**
- Draw partition with K clusters of data from $P(Z|X)$
- Fit K independent GP models on the partitioned data.
- Predict new observations on each importance sample with

$$P(f_j^*|Z_j, -) = \sum_{k=1}^{K} P(f_j^*|Z_j^*, -)P(Z_j^*|-).$$

- Obtain weights $w_j = \prod_{k=1}^{K} P(Y_{k,j}|X_{k,j}, Z_j)$.

Normalize weights, $w_j := w_j / \sum_{j=1}^{J} w_j$.

Average predictions using importance weights: $P(\bar{f}^*|-) = \sum_{j=1}^{J} w_j P(f_j^*|Z_j, -)$

3.1. Design choices

Our proposed method makes a number of design choices, each of which carries important consequences for the performance of our algorithm which we will discuss in this section. In Equation 2, we assume that each GP has its own set of hyperparameters, Θ_k. This allows us to capture a degree of non-stationarity and heteroscedasticity, for minimal additional cost. Alternatively, if we believe the model is stationary, we can share hyperparameters across partitions.

The Gaussian likelihood in Equation 1 is a design choice that is chosen to be appropriate in a wide range of settings. A mixture of Gaussians allows us to exploit correlations in the input location, and encourages preservation of short-range covariances though alternative likelihoods could also be used. In Section 4.1.2, we will show that placing structure on the input space produces better results than simple uniform partitioning of the data or, at the very least, produces results that are not different than uniform partitioning when there is no structure in the input space. We consider the general setting where each GP has its own set of hyperparameters, Θ_k.

To avoid explicitly selecting the number of mixtures, K, to use to model our input space, we may instead draw partitions from the Dirichlet process mixture model (DPMM) instead, as seen in Rasmussen and Ghahramani (2002); Meeds and Osindero (2006); Yuan and Neubauer (2009) and control the number of partitions via the concentration parameter, α. We choose a finite mixture model for two reasons. First, a Dirichlet distribution with $\alpha > 1$ avoids the rich-get-richer behavior of the Dirichlet process, encouraging similarly sized clusters rather than one very large cluster. Second, a finite mixture model allows us to explicitly investigate the effect of increasing the number of clusters on the performance of our algorithm. However, the issue of selecting K is vital in practice. To address this problem, we could fit a mixture model on the data (or a subset, in "big data" cases) beforehand and empirically estimate K from this mixture model's posterior. Or, we could adopt a more systematic method of selecting K by using a Bayesian optimization method (Snoek et al., 2012, for example) to explore the optimal number of partitions.

3.2. Importance sampling

We wish to capture posterior uncertainty about the partition and the associated covariance function $\mathbf{\Sigma}$, while ensuring our algorithm can be distributed. Importance sampling allows us to estimate the posterior expectations $E[g(f)]$ of some functional of f, such as the posterior predictive distribution, using an appropriately weighted collection of samples from some simpler distribution. Unlike MCMC, these samples can be collected independently, facilitating distributed computing.

We choose our proposal distribution over partitions to be the posterior distribution $P(Z|X)$ under the Gaussian mixture model given in Equation 1 conditioned on the input values X, ignoring the output values Y. We obtain approximate samples from this distribution by drawing mixture locations randomly from the prior, $P(\theta_k, \Gamma_k)$ and assign data to clusters from $P(z_i = k|-)$. After fitting the local GPs derived from this partition, we then weight these particles using self-normalized importance sampled weights [1]

$$w_j \propto \frac{p(Z|X,Y)}{p(Z|X)},$$

where $\sum_j w_j = 1$ and $p(Z|X,Y) \propto P(Y,X|Z)P(Z)$ is the cluster assignments using output and input data (as opposed to only input data in $P(Z|X)$). We can then obtain an asymptotically unbiased estimate $\hat{\mu}$ to $E[g(f)]$ as

$$\hat{\mu} = \sum_{j=1}^{J} w_j g(f_j).$$

As a concrete example, the posterior predictive distribution is approximated as

$$\hat{p}(f^*|x^*, X, Y) = \sum_{j=1}^{J} w_j p(f_j^*|Z_j, x^*, X, Y),$$

where Z_j is the partition associated with the jth sample.

Calculating the w_j involves integrating over the covariance parameters Θ,

$$w_j \propto \frac{p(Z|X,Y)}{p(Z|X)} \propto \frac{p(X,Y|Z)p(Z)}{p(X|Z)p(Z)} = p(Y|X,Z) = \int p(Y|X,Z,\Theta)p(\Theta)d\Theta. \tag{3}$$

where $p(\Theta)$ is the prior over the covariance parameters. If we are allowing separate hyperparameters, $\Theta_{j,k}$, for each partition, we assume that $p(\Theta) = \prod_k p(\Theta_k)$, so

$$w_j \propto \prod_{k=1}^{K} p(\{y_i : z_{j,i} = k\}|\{x_i : z_{j,i} = k\}) \tag{4}$$

$$= \prod_{k=1}^{K} \int p(\{y_i : z_{j,i} = k\}|\{x_i : z_{j,i} = k\}, \Theta_k)p(\Theta_k)d\Theta_k \tag{5}$$

We must approximate the intractable integral. Depending on our accuracy/speed trade-off, we can obtain an unbiased estimate of the w_j using a sample-based approximation; we can perform a Laplace approximation about the MAP solution $\hat{\Theta}$; or we can directly use the MAP approximation $p(Y|X,Z) \approx p(Y|X,Z,\hat{\Theta})$. In our experiments, we choose to directly use the MAP solution, obtained using gradient descent; while this is not as accurate as sampling hyperparameters it is significantly faster, and mirrors the choices made by our comparison methods.

Calculating the MAP approximation of Θ (or indeed, inferring the hyperparameters using MCMC or another method) requires calculating the marginal likelihood $p(Y|X,Z,\hat{\Theta})$. This means there is no additional cost involved in calculating the importance weights, up to a normalizing constant. Independence between each importance sample means that the samples and their normalizing constants can be obtained in parallel. The only global communication required is at the end of the procedure, when the importance weights are normalized and the samples are combined to give our predictive distribution (or other desired expectation). In the regression scenario, we can obtain the exact marginal likelihood to fit mixtures of GPs due to the tractability of the classic regression model. However, we can show that in Section 4.2.3 it is possible to use an approximation to the marginal likelihood as an importance weight to fit classification IS-MOE models that obtain good performance.

The overall computational cost of the IS-MOE, using J importance-weighted samples and K blocks, is therefore $O(JN^3/K^2)$. In Table 1, we compare this with the overall computational cost of the full GP, sparse approximations (FITC, DTC and SVI), the

1. Since we are working with self-normalized weights, the estimate has a bias of $O(1/J)$ (Kong, 1992), but will often have a lower variance than the unbiased estimate obtained with $w_j = p(Z|X,Y)/p(Z|X)$, which involves calculating an intractable normalizing constant.

Bayesian treed GP (BTGP), and the robust Bayesian committee machine (RBCM). While the $O(JN^3/K^2)$ cost is $O(J)$ higher than sparse methods and local methods based on a fixed partition such as RBCM, we note that the J samples can be performed and weighted in parallel—meaning the time taken is comparable if we are willing to sacrifice computational resources.[2] In this procedure, the only communication between processors occurs at the end of the prediction step when we normalize the weights, w_j, and obtain the importance averaged predictions, $\bar{f}^*$. This is vital in any distributed computation algorithm due to the high overhead cost of inter-processor communication. We can also make use of the independence of the K partitions to parallelize further, using JK threads each taking $O(N^3/K^3)$. As shown in Table 1, this leads to an equivalent wall-time cost comparable with the distributed RBCM. As we will see in Section 4, the extra computational cost required to ensure a full posterior predictive distribution yields improved performance over methods that are based on a fixed partition.

However, importance sampling has inherent issues that can hinder practical performance for inference and prediction. First, importance samplers have a tendency to produce weights where one proposal completely dominates the rest of the proposals and obtains an importance probability of nearly one. To this end, we could smooth out the importance weights with a Pareto distributed smoother in order to obtain more stable estimates from our importance sampler (Vehtari et al., 2015). Additionally, choosing a good proposal distribution is critical to the performance of the importance sampler but it is not obvious how to select the best distribution. Kahn and Marshall (1953) show that the optimal distribution which minimizes the estimator variance is $|g(f)|p(f)$ for the expectation $\mu = \int g(f)p(f)/q(f)\ df$ though practically speaking we may not be able to easily sample from this distribution or calculate importance weights.

3.3. Minibatched importance samples

Although we can obtain significant computational and memory saving advantages using our low-rank approximation, we still may encounter major bottlenecks from attempting to approximate the covariance matrix of the full training set. To overcome this issue, we propose a "minibatching" solution, where each importance sample is obtained and weighted based only on a subset of size $B << N$ sampled uniformly without replacement. Given a random subset B of observations, we can approximate $p(\Theta|X, Y, Z)$ with the subset posterior $p(\Theta|X^{mb}, Y^{mb}, Z)$ evaluated on a size-B minibatch (X^{mb}, Y^{mb}). Such a posterior estimate is strongly consistent, but will tend to underestimate the posterior variance (Srivastava et al., 2015). To achieve realistic credible intervals, we can assume we have seen each pair (x_i, y_i) in our minibatch N/B times; mathematically, this corresponds to raising the contribution of the likelihood to the subset posterior to the (N/B)-th power.

We use this stochastic approximation trick to estimate the posterior distribution over parameters for each importance sample, allowing us to reduce our overall complexity from $O(JN^3/K^2)$ to $O(JB^3/K^2)$. Empirical results in Section 4 will show that this stochastic approximation performs favorably on large datasets in comparison with both the non-SA IS-MOE method and other scalable GP inference methods.

2. In general, a sparse model with M inducing points obtains comparable accuracy to a local method with N/K local GPs, and has equivalent computational complexity.

Table 1: Comparison of inference complexity. N is the number of data points, K is the number of experts or local GPs, and $M = N/K$ is the number of inducing points. For the Monte Carlo based methods, J is the number of MCMC iterations or importance samples.

	Full GP	Sparse	SVI	
Complexity	N^3	NM^2	$M^2 \min(M, B)$	
	RBCM	**BTGP**	**IS-MOE**	**SA IS-MOE**
Complexity	N^3/K^2	JN^3/K^2	JN^3/K^2	JB^3/K^2
Comp/thread	N^3/K^3	$\times$	N^3/K^3	B^3/K^3

4. Experimental evaluation

To showcase the performance of our method, we compare it with a number of competing methods on both synthetic and real data sets.

4.1. Evaluation on synthetic data

4.1.1. Comparison with competing methods

We begin by evaluating our method on synthetically generated data, in order to allow us to explore and visualize a range of regimes, and to allow comparison with methods that do not scale to our real-world dataset. In our studies, we will compare our Importance Sampled Mixture of Experts approach (IS-MOE) against a full Gaussian process (GP); three sparse approximations to this model: FITC (Snelson and Ghahramani, 2005), DTC (Seeger et al., 2003), and SVI (Hensman et al., 2013); the Bayesian treed GP (Gramacy and Lee, 2008, BTGP); and the robust Bayesian committee machine (Deisenroth and Ng, 2015, RBCM). All models use a squared exponential covariance matrix. Our IS-MOE code uses the Gaussian process modules in `GPy` in Python with parallelization executed through `mpi4py` (Dalcín et al., 2005).[3] We ran the full GP, FITC, DTC and SVI implementations also through `GPy`, BTGP in `tgp`, and RBCM in `gptf`.

We first consider three data settings, the first two were generated on a linearly spaced grid of values on $[-1, 1]$ and the last one has more data generated in the center of the range $[-1, 1]$ in order to test how our method approach works when data are not uniformly generated. For these experiments we did not use minibatching in IS-MOE.

1. **Stationary, long-range correlations** generated with inverse length scale $\gamma = 15$.
2. **Stationary, short-range correlations** generated with inverse length scale $\gamma = 5000$.
3. **Non-stationary** generated piecewise with fast and slow moving periodic functions.

In examples 1 and 2, we generated data from a GP with zero mean squared exponential covariance kernel with amplitude $\nu = 1$. For all examples we added Gaussian noise $\sigma^2 = 1$

3. The code is available at `https://github.com/michaelzhang01/ISMOE`.

to the observed outputs. We generated a training data set with 1,000 observations and a test set with 100 observations.

For fitting the stationary data, we restrict the hyperparameters on our IS-MOE method to be the same on all K blocks. For (3), we allowed each mixture to have its own hyperparameters in order to model the non-stationarity of the data. In all methods except BTGP we infer hyperparameters through the MAP estimate via gradient descent optimization, and for BTGP we infer the hyperparameters through MCMC sampling. For the sparse methods, we used $M = 100$ inducing points, and for the local methods (including the IS-MOE) we used $K = 10$ partitions to have a comparable level of computational complexity. For the BTGP we ran the MCMC sampler for 10 iterations; for the IS-MOE we used $J = 10$ independent importance-weighted samples. Figures 2, 3, and 4 shows the posterior predictive results and predictive intervals obtained using the five methods, and Tables 2a and 2b show the corresponding test set log likelihoods and mean squared errors.

We first consider the one-dimensional stationary examples. Recall that, in general, sparse methods perform well when the covariance structure is dominated by longer-range correlations, and local methods perform well when we have significant local variation in our function. For these results, we deliberately set J to a small number to see how IS-MOE performs when there is "not enough" importance samples as a difficult scenario in comparison to the other methods. Looking at the results on the dataset with long-range correlations (Figure 2), we see that the IS-MOE can capture the predictive variance unlike RBCM which is over-confident in its predictions, and only performs slightly worse than the full GP and the sparse approximations, again due to a small number of importance samples.

If we look at the dataset with short-range correlation (Figure 3), we see the sparse methods struggle to learn the function–with a small number of inducing points, it is impossible to capture the high-frequency variation. Looking at the quantitative results in Tables 2a and 2b, we see that the IS-MOE outperforms the RBCM because our method is capable of learning the proper predictive variance whereas the RBCM is over-confident in its results. The BTGP likely produces poor predictive performance is because of lack of convergence of the MCMC chain: the underlying model is fairly complex and will tend to mix slowly and at a comparable level of computational complexity in this evaluation, does not have enough MCMC iterations to converge.

We can also see that the full GP struggles in this scenario between assuming the function is one that exhibits high noise in the data or that the function exhibits short-range correlation. Figure 1 shows an example of this multimodal structure in the marginal GP likelihood. Here, if we randomly initialize the hyper-parameter values from the distribution Gamma$(1, 4)$ for the full GP we can see that the optimizer can get stuck in suboptimal modes. Monte Carlo methods like importance sampling generally do not suffer from converging to suboptimal local optima as much as gradient descent methods or MCMC would because importance samplers will independently explore different parts of the marginal likelihood surface and place more weight on particles in more optimal modes than less optimal ones.

Finally, consider the non-stationary example, which combines known failure modes of local and sparse GPs. We have a combination of slowly varying behavior (which is poorly captured by local methods) and fast-varying behavior (which is poorly captured by sparse methods). The full GP, RBCM and sparse methods, fitted with stationary kernels, obviously cannot account for the non-stationary components in the data, and by assuming a stationary

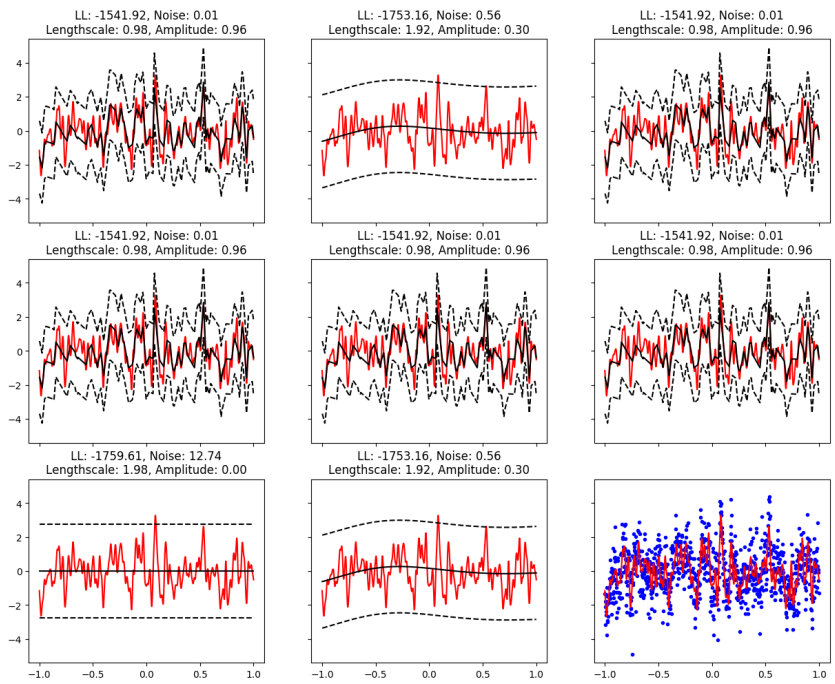


Figure 1: Multiple instantiations of full Gaussian process fits to short lengthscale data. The true latent function is plotted in red with the predictive mean in black with 95% credible intervals in black dashed lines. The observed data is plotted in blue dots.

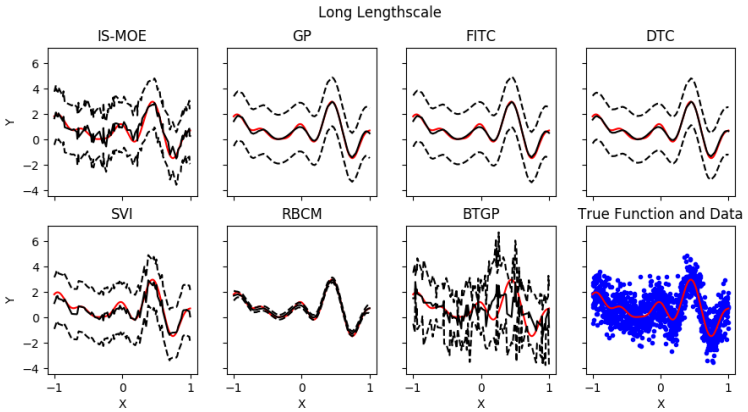


Figure 2: Posterior mean and 95% predictive intervals on Synthetic 1 (stationary, long length scale).

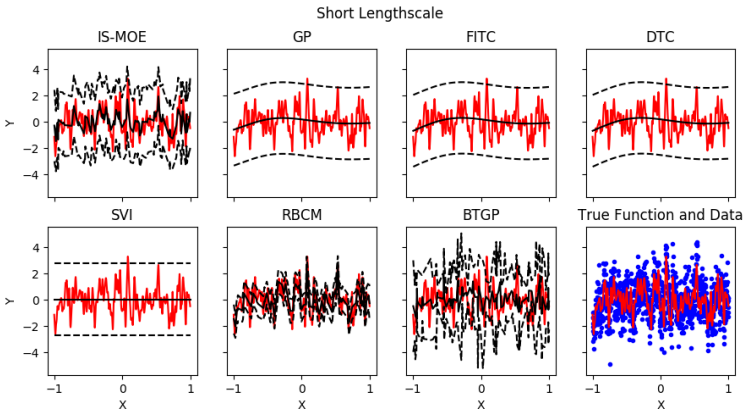


Figure 3: Posterior mean and 95% predictive intervals on Synthetic 2 (stationary, short length scale).

covariance they give poorer test-set performance. The BTGP does a reasonable job at capturing the function; again its performance is likely to be hampered by slow mixing and lack of convergence. Figure 4 shows that the IS-MOE is able to capture the function, and Tables 2a and 2b show that it can provide confident predictions at all regions of the function.

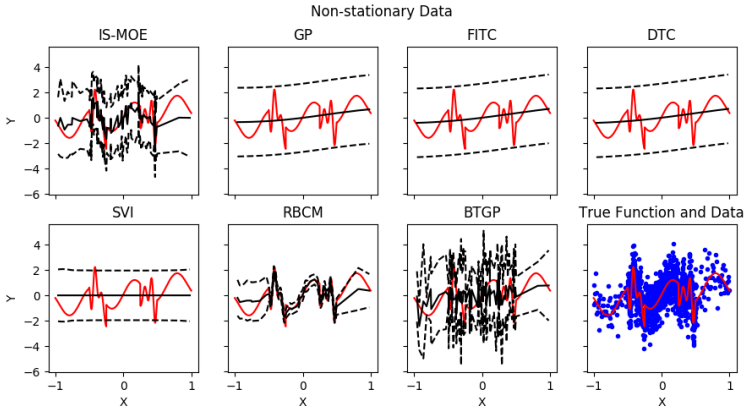


Figure 4: Posterior mean and 95% predictive intervals on Synthetic 3 (non-stationary).

Table 2: Test set performance on synthetic datasets

(a) Log likelihood

Data	**IS-MOE**	GP	FITC	DTC	SVI	RBCM	BTGP
Long Lengthscale	-152.41	-143.52	-143.39	-143.81	-156.51	-5207.89	-231.84
Short Lengthscale	-157.16	-172.32	-172.26	-172.26	-173.38	-251.50	-212.61
Non-stationary	-158.21	-181.40	-181.30	-181.30	-198.00	-910.54	-256.73

(b) MSE

Data	**IS-MOE**	GP	FITC	DTC	SVI	RBCM	BTGP
Long Lengthscale	1.20	1.03	1.02	1.02	1.30	1.03	2.07
Short Lengthscale	1.35	1.83	1.83	1.83	1.87	1.06	2.46
Non-stationary	1.39	2.18	2.17	2.17	2.12	1.00	2.64

Table 3: Predictive log likelihood, MSE and total run time for the non-stationary example, $J = 6000$

	Log Likelihood	MSE	Wall time (s.)
IS-MOE	-154.79	1.28	61.16
BTGP	-145.71	1.04	1116.75
RBCM	-910.55	1.01	13.08

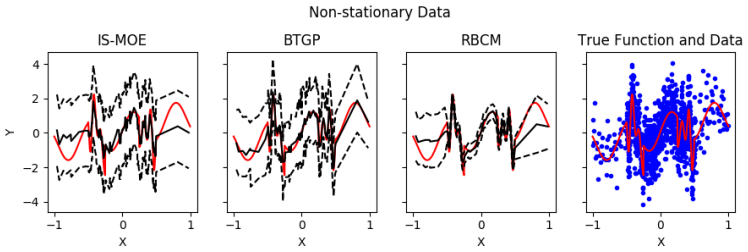


Figure 5: Posterior mean and 95% predictive intervals on Synthetic 3 (non-stationary), $J = 6000$.

To provide a deeper comparison with the treed Gaussian process, we then run the non-stationary example with the local methods (IS-MOE, BTGP and RBCM, as these methods are the only ones considered in the experiments that can model non-stationary data) when there is sufficient computational resources for the MCMC chain in the BTGP to converge (or in other words, when we set $J = 6000$). In Figure 5, we see that BTGP learns a much smoother latent function that the one in Figure 4 while the results for IS-MOE and RBCM still largely remain the same. Table 3 shows the predictive and timing comparisons between the three methods. We see that BTGP obtains the best predictive log likelihood and RBCM obtains the best MSE and fastest run time. However, our method, IS-MOE, produces the best compromise between BTGP and RBCM as we obtain only slightly worse predictive log likelihood results than BTGP while being more than eighteen times as fast. Thus, we can understand our method as being a fast, parallelizable, approximate variant of the BTGP. Though IS-MOE is slower and less accurate (with respect to MSE) than RBCM in this experiment, we obtain far better uncertainty quantification of our predictions as reflected in the poor log likelihood results of the RBCM.

4.1.2. The importance of importance sampling

The IS-MOE falls under the "local" framework, much like the RBCM and the BTGP; however it out-performs both methods. This can be attributed to importance sampling a distribution over partitions. To demonstrate this, we consider our performance on a synthetic dataset of 10,000 training observations, 2,000 test observations and 100 covariates. The input data are sampled from a GMM on $\mathbb{R}^{100}$ with $K = 50$ mixture components. The output, Y, is drawn from a GP model with zero mean and an RBF kernel with inverse length scale .001, observation noise variance of .25 and amplitude of 2.

The RBCM uses a single, fixed partition. Conversely, the IS-MOE uses a distribution over partitions, combined using importance sampling weights. Figures 6 and 7 show how varying the number of importance samples, for a range of values of K and B (remember, $K = 1, B = N = 10,000$ corresponds to the full Gaussian process, and as K increases or B decreases, we expect a drop in quality). In most cases, we see a similar pattern: there is a clear improvement in performance between $J = 1$ to around $J = 50$, but beyond that the

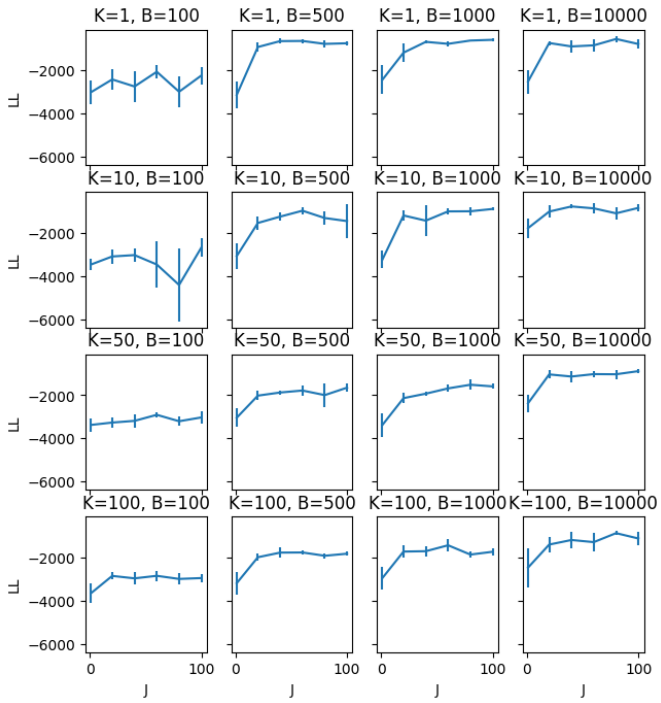


Figure 6: Evaluation on synthetic data of the effect of the number of samples J on the test set log likelihood of IS-MOE (with 95% confidence intervals), for various values of B and K.

improvements level off. This confirms that averaging over partitions improves performance, but suggests that in this setting, we need relatively few samples to approximate the posterior. In Figure 8, we can visualize why this is the case if we compare the resulting expected covariance matrices in a product of expert type approach like the RBCM with a mixture of expert approach like ours. We note that BTGP also averages over partitions and can achieve high quality predictions as a result; however the slow mixing of the MCMC algorithm and the inability to distribute inference means we get worse performance for the same computational effort, and precludes the use of BTGP on large datasets.

The IS-MOE uses importance sampled weights to average over partitions. In the minibatch setting, these weights and the samples themselves are obtained using a stochastic approximation. It is reasonable to question whether either the calculation of importance weights, or the up-weighting of the likelihood to obtain a stochastic approximation, affect the performance. In other words – would we do as well using uniform weights or avoiding the stochastic approximation? As we see in Table 4 (which uses the same synthetic dataset

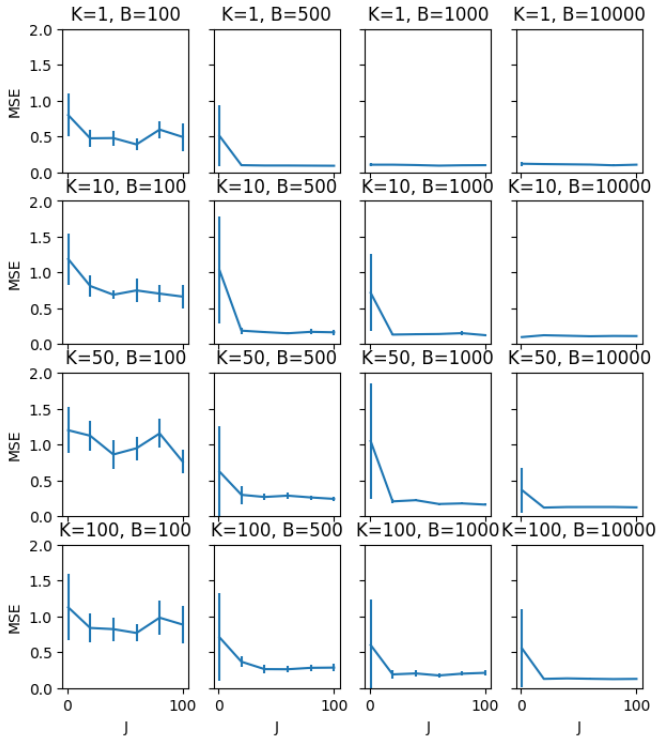


Figure 7: Evaluation on synthetic data of the effect of the number of samples J on the test set MSE of IS-MOE (with 95% confidence intervals), for various values of B and K.

as above, with $J = 10$, $K = 10$ and $B = 1000$), using importance samples with reweighted likelihood minibatches results in better predictive performance than either not upweighting the minibatched likelihood or using uniform weights to combine predictions.

A final difference from the RBCM is the choice of the distribution over partitions that the IS-MOE is able to explore. The IS-MOE uses a distribution based on covariate location, while the RBCM generates its single partition uniformly. To evaluate the impact of using covariate location to guide the partitioning, we explore two variants of the IS-MOE model: one that uses a Gaussian mixture model to partition data, and one which uses random, uniform partitions. We tested these two methods on two synthetic datasets. The first was the synthetic dataset described above, with 100-dimensional inputs sampled from a mixture of 50 Gaussians. The second used the same kernel, but used 100-dimensional inputs with each dimension sampled uniformly from a Uniform$(-1, 1)$ distribution. For both cases, our

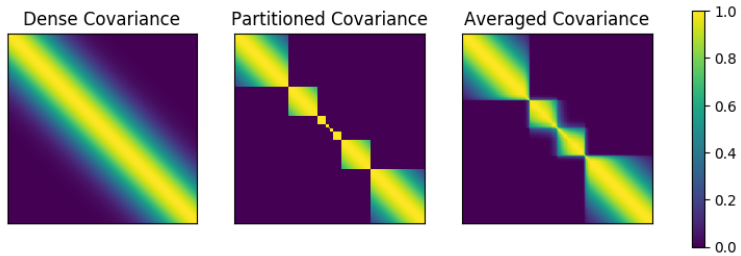


Figure 8: Dense, partitioned and averaged covariance matrices for a latent function with a short lengthscale. "Dense Covariance" is the dense covariance matrix $\Sigma(X, X')$. "Partitioned Covariance" is one instance of a block diagonal covariance matrix. "Averaged Covariance" is an averaging of several block diagonal partitioned covariance matrices.

Table 4: Test set log likelihood and MSE for various weighting schemes. Standard errors are in parentheses

Setting	LL	MSE
IS with SA	-354.96 (28.59)	0.096 (0.003)
IS without SA	-423.93 (41.24)	0.097 (0.002)
Unif. with SA	-726.67 (14.19)	0.19 (0.019)
Unif. without SA	-842.88 (9.82)	0.25 (0.337)

IS-MOE model used $J = 10$, $K = 10$ and $B = 1000$; note that the number of clusters used does not match the number of generating clusters. As we can see in Table 5, when the input data exhibits clustering behavior, we perform better when we place structure in the input clustering as opposed to purely random partitioning. When there is no structure (i.e. the inputs are sampled from a uniform distribution) there is no significant difference between the two models. This suggests that using a GMM is a reasonable default choice, since it can

Table 5: Test set log likelihood and MSE for two different covariate partitioning schemes when the input space is generated according to a GMM, and where the inputs are i.i.d Uniform. Standard errors are in parentheses.

Generating mechanism	IS-MOE partitioning scheme	LL	MSE
GMM	GMM	-429.19 (41.57)	0.14 (0.01)
	Random Clusters	-789.37 (37.43)	0.53 (0.02)
Uniform	GMM	-2962.01 (6.45)	0.62 (0.01)
	Random Clusters	-2964.29 (7.00)	0.62 (0.01)

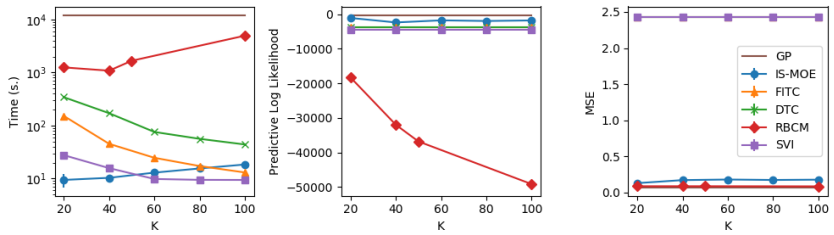


Figure 9: Wall clock time, predictive log likelihood and MSE results for large synthetic dataset with increasing K. SVI, FITC and DTC obtain very similar predictive performances.

.

take advantage of any structure in the input space but does not degrade performance in the absense of structure.

Lastly, we are interested to see how different GP methods compare with regards to the total wall clock time for fitting the model. Using the same data set, with $J = 128$ and $B = 1000$ for SVI and IS-MOE, we evaluate the speed and predictive performance of IS-MOE against the other methods compared in this paper. As seen in Figure 9, we can see that IS-MOE is faster than all other methods (most notably, SVI) at low values of K while maintaining good predictive log likelihood and MSE performance. Only RBCM can outperform IS-MOE compared to the full GP in terms of MSE, but, as cited earlier, RBCM will systematically produce overconfident predictions. Counter-intuitively, we would expect local methods to perform faster with respect to wall time as K increases, but the results of IS-MOE and RBCM show that increasing K actually can increase wall time. We claim that the reason for this is because there is additional computational overhead involved with increasing the number of experts, as we must increase the number of GP models instantiated, whereas increasing K for sparse methods corresponds to reducing M, meaning there are less parameters to fit in the sparse model.

4.2. Evaluation on real data

As seen in our experiments on synthetic data, the IS-MOE is applicable to many different data regimes where other approximations may fail. Its inherently parallelizable nature also makes it an appealing choice for larger, real-world datasets where use of a full GP is computationally infeasible. To evaluate performance in this "big data" regime, we used an empirical dataset consisting of 209,631 mid-tropospheric CO2 measurements over space and time from the Atmospheric Infrared Sounder (AIRS)[4]. First, in Section 4.2.1, we use this dataset to explore the sensitivity of our model to different parameter values, to show how we can trade off between predictive accuracy and computational cost. Then, in Section 4.2.2 we compare its performance against competing approaches.

4. Available in the R package `FRK` as `AIRS_05_2003`

4.2.1. Sensitivity to model settings

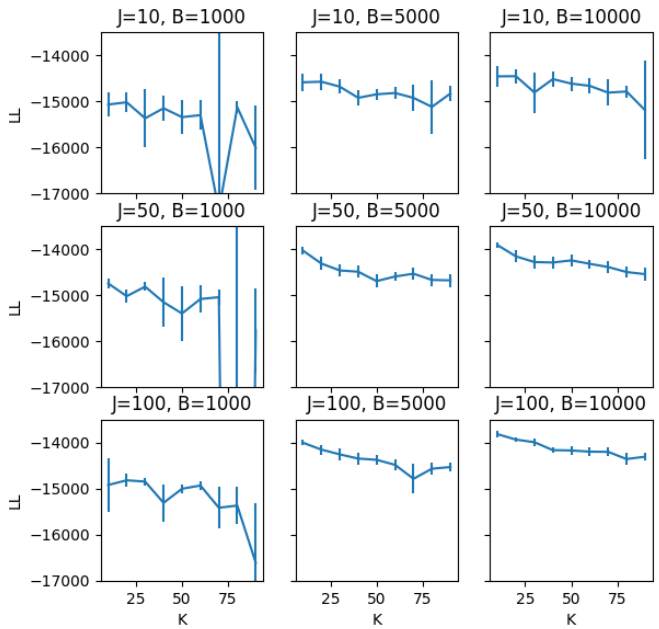


Figure 10: Evaluation on the AIRS dataset of the effect of B, K, and J on the test set log likelihood of IS-MOE (with 95% confidence intervals).

Clearly, both the performance and the cost of inference of our model will depend on the number of blocks K in our approximation, the number of importance samples J, and the minibatch size B. On the one hand, inference scales as $O(JB^3/K^2)$, so we can speed up inference by decreasing J or B or by increasing K. On the other hand, a smaller number of blocks will allow us to better approximate a dense covariance matrix; a larger number of importance samples helps us explore the full posterior; larger minibatches reduce the noise in our estimators.

In order to pick values for K, J and B, we must understand how they affect our overall estimates. We trained the IS-MOE using a range of values for B, K and J, over 20 cross-validation splits. As expected, we find that as we increase K or decrease J and B our performance deteriorates. Figure 10 shows that as the B and J increases, the average predictive log likelihood increases and the variance of the log likelihood decreases, and that as K increases the quality of our inference method degrades. However, looking at Figure 10, we see the deterioration in predictive likelihood is fairly gradual for most values: we only see a dramatic degradation when we have both a small minibatch size and a large number of

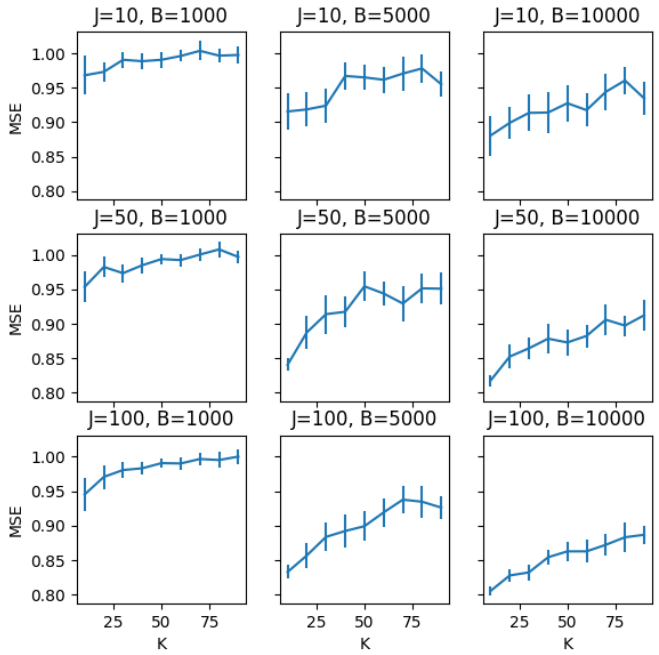


Figure 11: Evaluation on the AIRS dataset of the effect of B, K, and J on the test set MSE of IS-MOE (with 95% confidence intervals).

partitions. This suggests that the practitioner can modify B, J and K within a wide range to achieve acceptable computational costs without a dramatic drop in quality.

4.2.2. Comparison with competing methods

Using the same CO2 dataset and squared exponential kernel as before, we compare the IS-MOE with SVI – the only other method that would scale to this dataset.[5] For IS-MOE, we set $J = 100$ and $B = 1000$ and explored a range of values of K; for SVI we chose values for inducing points that gave a comparable level of computational complexity. We evaluated performance over 20 cross-validation splits. Our importance sampling method provides for a richer predictive model due to the averaging over importance proposals, and we see the benefit of this in our results. As Figure 12 shows, the IS-MOE typically performs comparably to SVI at equal levels of computational complexity in predictive performances using both

5. While RBCM is designed to scale to large data, we were unable to run the available Python package `gptf` due to memory issues.

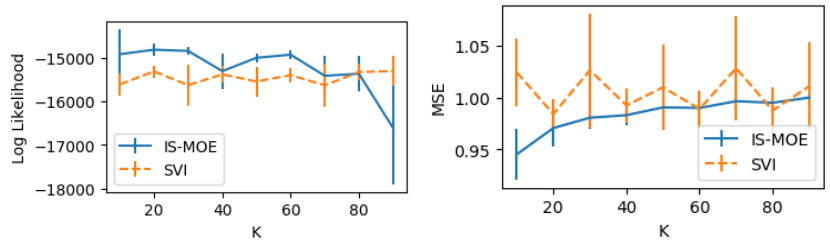


Figure 12: Comparison of IS-MOE and SVI on AIRS dataset for various K, with $J = 100$ and $B = 1000$. SVI parameters chosen to have equivalent computational cost.

metrics until approximately 90 clusters, in which our method performs notably worse due to the long range correlation present in this dataset.

4.2.3. Applications beyond regression

Finally, to highlight that our method is not limited to a specific GP model, we apply our method on a binary classification task, using a Laplace approximation with a squared exponential kernel. We compare with the full GP (Williams and Barber, 1998) and the sparse GP (Hernández-Lobato and Hernández-Lobato, 2016) on three classification datasets from the UCI repository: the Pima Indians diabetes dataset; the Parkinsons dataset; and the Wisconsin diagnostic breast cancer (WDBC) dataset.[6] As Table 6 shows, the IS-MOE can approximate the full GP results very well, with comparable area under the curve (AUC) scores and log likelihood to a full GP and a sparse approximation.

In addition, we also ran IS-MOE in comparison with the sparse variational GP (Hensman et al., 2013) on a binary classification dataset to distinguish background processes from Higgs-Boson particles. Our training data contains one million observations and 28 features and a test set of 100,000 observations with $J = 128$, $K = 20$ and $B = 1000$. Figure 14 shows the comparison of IS-MOE against SVI for this large scale classification task. We see that IS-MOE obtains better predictive log likelihood and AUC scores, while performing slower in comparison to SVI in terms of wall time. Given this fact, it may be possible that the latent generating process for the Higgs-Boson data is non-stationary leading to better performance for IS-MOE over SVI. On the other hand, SVI's apparent superior performance in terms of wall time but not predictive performance could be explained by the fact that there is a local optima that SVI tends to converge to quickly which produces sub-optimal results, whereas IS-MOE must spend more time exploring the posterior space but can find a better result. Nevertheless, the fact that IS-MOE inference procedure can obtain a wall time of about six minutes on a training set of size one million is still quite impressive and it does suggest that IS-MOE can perform well with marginal likelihood approximations to the full model as well.

6. All empirical classification datasets are available in the UCI repository at `http://archive.ics.uci.edu/ml/`.

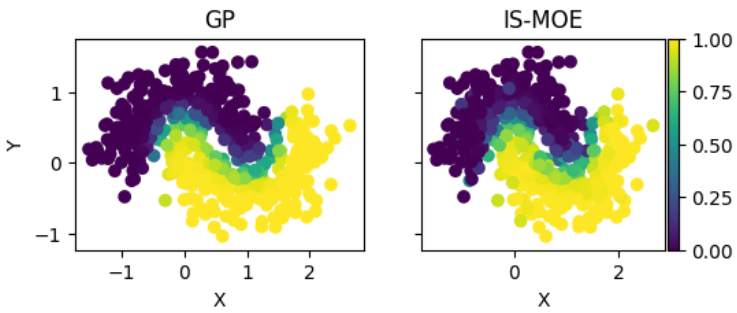


Figure 13: Binary classification task: label probabilities obtained using the full GP and the IS-MOE.

Table 6: Test set log likelihood and AUC on three classification datasets.

	Log Likelihood			AUC		
Data	Full GP	IS-MOE	FITC	Full GP	IS-MOE	FITC
Pima	-128.79	-135.09	-128.61	0.83	0.81	0.83
Parkinsons	-17.00	-22.76	-28.42	0.86	0.93	0.88
WDBC	-15.50	-12.62	-18.01	0.83	0.91	0.81

5. Summary and future work

While Gaussian processes provide a flexible framework for a wide variety of modeling scenarios, their use has been limited in the "big data" regime, since most implementations scale cubically with the number of data points. As discussed in Section 2, a number of approximations have been proposed to reduce this cost but these approximations come with notable failure modes. The IS-MOE avoids these pitfalls, using parallelizable importance sampling to explore a

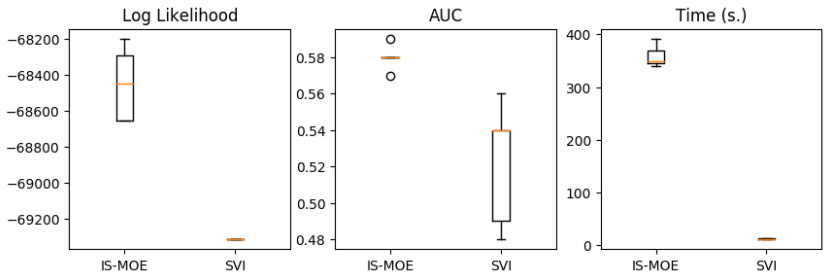


Figure 14: Results for Higgs-Boson classification experiment.

mixture of block-diagonal, easily invertible matrices. However, importance sampling can be a rather rudimentary approach to inference. For more sophisticated settings, we may need to resort to particle filters or other sequential Monte Carlo techniques. Of course in such inference methods, the proposal distribution is crucial for performance but difficult to choose in practice. We are interested in investigating the theoretical behavior of proposal distributions in this setting.

In this paper, we have focused on regression models using a Gaussian mixture model on the covariates, but the scope of the IS-MOE is much broader. For example, we could use alternative distributions over partitions, or embed the IS-MOE within a more complex model–particularly in deep Gaussian process models and non-Gaussian likelihoods. We are also interested in refinements to importance sampling to improve their inferential quality, such as alternative proposal distributions. Another potential avenue for future research is to explore whether we can achieve further speed-ups by using GPU-based computation (Dai et al., 2014; Gramacy et al., 2014; Gramacy and Apley, 2015, for example). We leave such explorations for future work.

Acknowledgments

Michael Zhang and Sinead Williamson were supported by NSF grant 1447721. Sinead Williamson's contribution was written before her employment at Amazon.

References

Sudipto Banerjee, Alan E. Gelfand, Andrew O. Finley, and Huiyan Sang. Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848, 2008.

Yanshuai Cao and David J. Fleet. Generalized product of experts for automatic and principled fusion of Gaussian process predictions. In *Modern Nonparametrics 3: Automating the Learning Pipeline workshop at NIPS*, 2014.

Zhenwen Dai, Andreas Damianou, James Hensman, and Neil D. Lawrence. Gaussian process models with parallelization and GPU acceleration. *arXiv preprint arXiv:1410.4984*, 2014.

Lisandro Dalcín, Rodrigo Paz, and Mario Storti. MPI for Python. *Journal of Parallel and Distributed Computing*, 65(9):1108 – 1115, 2005.

Marc P. Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490, 2015.

GPy. GPy: A Gaussian process framework in Python. `http://github.com/SheffieldML/GPy`, 2012.

Robert B. Gramacy and Daniel W. Apley. Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

Robert B. Gramacy and Herbert K. H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

Robert B. Gramacy, Jarad Niemi, and Robin M. Weiss. Massively parallel approximate Gaussian process regression. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1): 564–584, 2014.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, 2013.

Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable Gaussian process classification via expectation propagation. In *Artificial Intelligence and Statistics*, pages 168–176, 2016.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems*, pages 4080–4088, 2016.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Herman Kahn and Andy W. Marshall. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

Cari G. Kaufman, Mark J. Schervish, and Douglas W. Nychka. Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555, 2008.

Augustine Kong. A note on importance sampling using standardized weights. Technical Report 348, University of Chicago, Dept. of Statistics, 1992.

Haitao Liu, Jianfei Cai, Yi Wang, and Yew-Soon Ong. Generalized robust Bayesian committee machine for large-scale Gaussian process regression. In *International Conference on Machine Learning*, pages 3137–3146, 2018.

Kian Hsiang Low, Jiangbo Yu, Jie Chen, and Patrick Jaillet. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *AAAI Conference on Artificial Intelligence*, 2015.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

Edward Meeds and Simon Osindero. An alternative infinite mixture of Gaussian process experts. In *Neural Information Processing Systems*, pages 883–890, 2006.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David B. Dunson. Scalable and robust Bayesian inference via the median posterior. In *International Conference on Machine Learning*, pages 1656–1664, 2014.

Jun Wei Ng and Marc P. Deisenroth. Hierarchical mixture-of-experts model for large-scale Gaussian process regression. *arXiv preprint arXiv:1412.3078*, 2014.

Chiwoo Park, Jianhua Z. Huang, and Yu Ding. Domain decomposition approach for fast Gaussian process regression of large spatial data sets. *Journal of Machine Learning Research*, 12(May):1697–1728, 2011.

Carl E. Rasmussen and Zoubin Ghahramani. Infinite mixtures of Gaussian process experts. In *Neural Information Processing Systems*, pages 881–888, 2002.

Carl E. Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. *Gaussian Processes for Machine Learning*, 2006.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics*, 2003.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Neural Information Processing Systems*, pages 1257–1264, 2005.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Neural Information Processing Systems*, pages 2951–2959, 2012.

Sanvesh Srivastava, Volkan Cevher, Quoc Dinh, and David B. Dunson. WASP: Scalable Bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pages 912–920, 2015.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, volume 5, pages 567–574, 2009.

Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Pareto smoothed importance sampling. *arXiv preprint arXiv:1507.02646*, 2015.

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Neural Information Processing Systems*, 2005.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, pages 681–688, 2011.

Christopher K.I. Williams and David Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

Chao Yuan and Claus Neubauer. Variational mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems*, pages 1897–1904, 2009.

Journal of Machine Learning Research 20 (2019) 1-50 Submitted 7/18; Revised 5/19; Published 9/19

DBSCAN: Optimal Rates For Density-Based Cluster Estimation

Daren Wang darenw@uchicago.edu
Department of Statistics
University of Chicago
Chicago, IL 60637, USA

Xinyang Lu xlu8@lakeheadu.ca
Mathematical Sciences Department
Lakehead University
Thunder Bay, ON P7B 5E1, Canada

Alessandro Rinaldo arinaldo@cmu.edu
Department of Statistics and Data Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Ingo Steinwart

Abstract

We study the problem of optimal estimation of the density cluster tree under various smoothness assumptions on the underlying density. Inspired by the seminal work of Chaudhuri et al. (2014), we formulate a new notion of clustering consistency which is better suited to smooth densities, and derive minimax rates for cluster tree estimation under Hölder smooth densities of arbitrary degree. We present a computationally efficient, rate optimal cluster tree estimator based on simple extensions of the popular DBSCAN algorithm of Ester et al. (1996). Our procedure relies on kernel density estimators and returns a sequence of nested random geometric graphs whose connected components form a hierarchy of clusters. The resulting optimal rates for cluster tree estimation depend on the degree of smoothness of the underlying density and, interestingly, match the minimax rates for density estimation under the sup-norm loss. Our results complement and extend the analysis of the DBSCAN algorithm in Sriperumbudur and Steinwart (2012). Finally, we consider level set estimation and cluster consistency for densities with jump discontinuities. We demonstrate that the DBSCAN algorithm attains the minimax rate in terms of the jump size and sample size in this setting as well.

Keywords: DBSCAN; density-based clustering; cluster tree; minimax optimality; Hölder smooth density.

1. Introduction

Clustering is one of the most basic and fundamental tasks in statistics and machine learning, used ubiquitously and extensively in the exploration and analysis of data. The literature on this topic is vast, and practitioners have at their disposal a multitude of algorithms and heuristics to perform clustering on data of virtually all types. However, despite its

©2019 Daren Wang, Xinyang Lu and Alessandro Rinaldo.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-470.html.

importance and popularity, rigorous statistical theories for clustering, leading to inferential procedures with provable theoretical guarantees, have been traditionally lacking in the literature. As a result, the practice of clustering, a central tasks in the analysis and manipulation of data, still relies in many cases on methods and heuristics of unknown or even dubious scientific validity. One of the most striking instances of such a disconnect is the DBSCAN algorithm of Ester et al. (1996), an extremely popular and relatively efficient (see Gan and Tao, 2015; Wang et al., 2015) clustering methodology whose statistical properties have been properly analyzed only very recently: see Sriperumbudur and Steinwart (2012), Jiang (2017a) and Steinwart et al. (2017).

In this paper, we provide a complementary and thorough study of DBSCAN, and show that this simple algorithm can deliver optimal statistical performance in *density-based* clustering. Density-based clustering (see, e.g., Hartigan, 1981) provides a general and rigorous probabilistic framework in which the clustering task is well-defined and amenable to statistical analysis. Given a probability distribution P on $\mathbb{R}^d$ with a corresponding continuous density p and a fixed threshold $\lambda \geq 0$, the λ-*clusters* of p are the connected components of the upper λ-level set of p, the set $\{x \in \mathbb{R}^d \colon p(x) \geq \lambda\}$ of all points whose density values exceed the level λ. With this definition, clusters are the high-density regions, subsets of the support of P with the largest probability content among all sets of the same volume.

As noted in Hartigan (1981), the hierarchy of inclusions of all clusters of p is a tree structure indexed by $\lambda > 0$, called the *cluster tree* of p. The chief goal of density clustering is to estimate the cluster tree of p, given an i.i.d. sequence $\{X_i\}_{i=1}^n$ of points with common distribution P. A cluster tree estimator is also a tree structure, consisting of a hierarchy of nested subsets of the sample points, and typically relies on non-parametric estimators of p in order to determine which sample points belong to high-density regions of p. A cluster tree estimator is deemed accurate if, with high probability, the hierarchy of clusters it encodes is close to the hierarchy that would have been obtained should p be known.

Density-based clustering, an instance of hierarchical clustering, enjoys several advantages: (1) it imposes virtually no restrictions on the shape, size and number of clusters, at any level of the tree; (2) unlike *flat* (i.e. non-hierarchical) clustering, it does not require a pre-specified number of clusters as an input and in fact the number of clusters itself is a quantity that may change depending on the level of the tree; (3) it provides a multi-resolution representation of all the clustering features of p across all levels λ at the same time; (4) it allows for an efficient representation and storage of the entire tree of clusters with a compact data structure that can be easily accessed and queried, and (5) the main object of interest for inference, namely the cluster tree of p, is a well-defined quantity.

Despite the appealing properties of the density-based clustering framework, a rigorous quantification of the statistical performance of this type of algorithms has proved difficult. Previous results by Hartigan (1981) and then Penrose (1995) have demonstrated a weaker notion of consistency achieved by the popular single-linkage algorithm. More recently Chaudhuri et al. (2014) have developed a general framework for defining consistency of cluster tree estimators based on a separation criterion among clusters. The authors further demonstrated that two graph-based algorithms, both based on k-nearest neighbors graphs over the sample points, achieve such consistency and provided minimax optimal consistency rates with respect to the parameters specifying the amount of cluster separation. Such results hold with virtually no assumptions on the underlying density. However,

because of this generality, these consistency rates do not directly reflect any degree of regularity or smoothness of the underlying density. In particular, it remains unclear whether cluster tree estimation would be easier with smoother densities.

In this paper we provide further contributions to the theory of density based clustering by deriving novel, nearly minimax-optimal rates for cluster tree estimation that depend explicitly on the smoothness of the underlying density function. Our results further confirm that the smoother the density the faster the rate of consistency for the cluster tree estimation problem, a finding that is consistent with analogous results about non-parametric density estimation. Interestingly, our rates match those for estimating smooth densities in the L_∞ norm. To the best of our knowledge, this finding and the implication that density based clustering is no easier – at least in our setting – than density estimation, has not been rigorously shown before. In order to account explicitly for the smoothness of the density, we have developed a new criterion for cluster consistency that is better suited for smooth densities. In terms of procedures, we consider cluster tree estimators that arise from applying a very simple generalization of the well-known DBSCAN algorithm and are computationally efficient. Furthermore, our DBSCAN-based estimator is minimax optimal over arbitrary smooth densities according to our notion of consistency under appropriate conditions.

Related work

The idea of using the probability density function in order to study clustering structure dates back to Hartigan (1981), who formalized the notion of clusters as the connected components of high density regions and of cluster tree. Much of the subsequent theoretical work focused on consistency for "flat" clustering at a fixed level, which effectively reduces to level set estimation. The literature on this topic is vast and offers a multitude of results covering different settings and metric for consistency. See, e.g., Penrose (1995) Polonik (1995), Tsybakov et al. (1997), Cuevas and Fraiman (1997), BaÍllo et al. (2000), Klemelä (2004), Willett and Nowak (2007), Singh et al. (2009), Rigollet and Vert (2009), Rinaldo and Wasserman (2010). In contrast, there have been fewer contributions to the theory of practice of cluster tree estimation: see, e.g., Stuetzle (2003); Stuetzle and Nugent (2010), Klemelä (2009) and Rinaldo et al. (2012). The work of Chaudhuri et al. (2014) (see also Kpotufe and Luxburg (2011)) represented a significant advance in the theory of density-based clustering, as it derived a new framework and consistency rates for cluster tree estimation. Balakrishnan et al. (2012) generalized these results to the probability distributions supported over well-behaved manifolds, with consistency rates depending on the reach of the manifold and its intrinsic dimension. Corresponding guarantees in Hausdorff distance have been recently obtained by Jiang (2017a). Eldridge et al. (2015) developed a unified theory for consistency in cluster tree estimation that encompasses the original framework of Hartigan while Kim et al. (2016) investigated the challenging problems of defining adequate metrics over the space of cluster tree and of constructing confidence sets for cluster tree structures. Chen et al. (2017) provides bootstrap-based methods for constructing confidence sets for density level sets and for visualization of high-density clusters. Recently, Jang and Jiang (2018) proposed a variant of the DBSCAN algorithm with both minimax clustering rate and sub-quadratic computational complexity while Jiang et al. (2019) studied DBSCAN under possibly adversarial contamination of the input data.

In a parallel and important line of work, Steinwart (2011, 2015) developed a rigorous, measure-theoretic approach to density-based clustering whereby the cluster tree is recovered by estimating the lowest split level of the density and then proceeding recursively. The corresponding results demonstrate a direct link between density based clustering and optimal level set estimation. This approach was applied in Sriperumbudur and Steinwart (2012) to show that the DBSCAN algorithm yield consistent estimator of density trees, a result that was then extended in Steinwart et al. (2017) to allow for more general, KDE-based procedures. Our work built directly upon the contributions of Chaudhuri et al. (2014) and Steinwart (2015).

Organization of the paper

The rest of the paper is organized as follows. In Section 3, we describe the DBSCAN algorithm and establish its connections with non-parametric density estimation. In Section 4 we introduce a new notion of cluster consistency, called δ-consistency that is tailored to Hölder-continuous densities. We describe a DBSCAN-based algorithm for clustered tree estimation that is computational efficient and delivers nearly optimal minimax rates that depend explicitly on the degree of smoothness of the underlying density, whereby cluster tree of smoother densities can be estimated at faster rates. Interestingly and, perhaps surprisingly, for the class of DBSCAN-based algorithms we consider, we observe a trade-off between statistical optimality and computational cost for smoother Hölder densities of degree $\alpha > 1$. In these situations, minimax rates can still be achieved by our computationally efficient algorithm provided that the underlying density satisfies additional geometric regularity conditions around the split levels. Such conditions are relatively mild and have been exploited before; see in particular Steinwart (2015). Finally, in Section 5 we consider a different scenario in which the underlying density exhibits jump discontinuities. We are particularly interested in level set and cluster estimation at the jump, with the assumption that the size of the discontinuity is vanishing when $n \to \infty$ so that clustering becomes increasingly difficult. We show that, with suitable inputs, the DBSCAN algorithm returns a Devroye-Wise type of estimator which is minimax optimal for cluster recovery and level set estimation. In addition, we derive the minimax scaling for the size of the jump discontinuity.

Notation

We denote with p a density for the distribution P of the i.i.d. sample $\{X_i\}_{i=1}^n \subset \mathbb{R}^d$. For a constant $\lambda > 0$, we set $L(\lambda) = \{p \geq \lambda\}$ to be the λ-upper upper level set of the density p. We use T_p to denote the cluster tree generated by the density p and $\widehat{T}$ to density any estimator of T_p. We use subscript n to emphasize any global variable which may change with respect to n. $\mathcal{L}$ represents the Lebesgue measure in $\mathbb{R}^d$ and $B(x, r)$ the closed d dimensional Euclidean ball centered at x with radius r and $V_d = \mathcal{L}(B(0, 1))$ the volume of the unit ball $B(0, 1)$. For a vector x we denote with $\|x\|$ and $\|x\|_\infty$ its Euclidean and L_∞ norms, respectively. With a slight abuse of notation, if f is a real valued function defined over a subset S of $\mathbb{R}^d$, we let $\|f\|_\infty = \sup_{x \in S} |f(x)|$ its L_∞ norm. For any $h > 0$ and a measurable

set $A \subset \mathbb{R}^d$ we set

$$A_h = \bigcup_{x \in A} B(x,h) \quad \text{and} \quad A_{-h} = \{x \in A : B(x,h) \subset A\}. \tag{1}$$

For any two real sequences $\{a_n\}_{n=1}^\infty$ and $\{b_n\}_{n=1}^\infty$ we write $a_n = O(b_n)$ if there exists $C > 0$ such that $\limsup_{n\to\infty} |a_n/b_n| < C$ and write $a_n = \Theta(b_n)$ if $a_n = O(b_n)$ and $b_n = O(a_n)$. For any two closed subsets A and B of $\mathbb{R}^n$, we use $d(A,B) = \inf_{x\in A, y\in B} \|x-y\|$ to represent the ordinary distance between them.

2. Cluster Trees Estimation

Let P be a probability distribution with a continuous[1] Lebesgue density p and with support $\Omega \subset \mathbb{R}^d$. For any $\lambda \geq 0$, let $\{x \in \Omega \colon p(x) \geq \lambda\}$ be the λ-upper level set of p and the λ-cluster of p are the connected components of $L(\lambda)$. See Appendix A for definition of connectedness. Notice that the set of all clusters is an indexed collection of subsets of Ω, whereby each cluster of p is assigned the index λ associated to the corresponding super-level set $L(\lambda)$, and that many clusters may be indexed by the same level λ. The cluster tree of p is the collection T_p of all clusters of p, that is

$$T_p = \{L(\lambda)\}_{\lambda \geq 0}.$$

We can think of the cluster tree of p as the function defined on $[0,\infty)$ and for each $\lambda \geq 0$, it returns the set of λ-clusters of p. Thus, $T_p(\lambda)$ consists of disjoint connected subsets of Ω. We remark that, since the density p is unique only up to sets of Lebesgue measure zero, the cluster tree T_p is also not unique. In fact, Steinwart (2015) shows that there exists a well-defined notion of cluster tree for the distribution P that is independent of the choice of the density. Furthermore, if P admits an upper semi-continuous density p, then the cluster tree is in fact composed of the hierarchy of the (closures of the) upper level sets of such density. As P is assumed, throughout most of the article, to have a density that is continuous everywhere on its support, when we speak of "the" density of P, we will refer to this canonical choice.

The concept of cluster tree owes its name to the easily verifiable property (see Hartigan (1981)) that if A and B are elements of T_p, i.e. distinct clusters of p, then $A \cap B = \emptyset$ or $A \subseteq B$ or $B \subseteq A$. This induces a partial order on the set of clusters. In particular, for any $\lambda_1 \geq \lambda_2 \geq 0$, if $A \in T_p(\lambda_1)$ and $B \in T_p(\lambda_2)$ then either $A \cap B = \emptyset$ or $B \subseteq A$. As a result, T_p can be represented as a dendrogram with height indexed by $\lambda \geq 0$. We refer to Kim et al. (2016) for a formal definition of the dendrogram encoding a cluster tree.

Let $\{X_i\}_{i=1}^n$ be i.i.d. samples from P. In order to estimate the cluster tree of p we will consider tree-valued estimators, defined below.

Definition 1. *A cluster tree estimator of T_p is a collection $\widehat{T}_n$ of subsets of $\{X_i\}_{i=1}^n$ indexed by $[0,\infty)$ such that*
- *for each $\lambda \geq 0$, $\widehat{T}_n(\lambda)$ consists of disjoint subsets of $\{X_i\}_{i=1}^n$ (including, possibly the empty set), called clusters, and*
- *$\widehat{T}_n$ satisfies the tree property: for any $\lambda_1 \geq \lambda_2 \geq 0$, if $A \in \widehat{T}_n(\lambda_1)$ and $B \in \widehat{T}_n(\lambda_2)$ then either $A \cap B = \emptyset$ or $A \subseteq B$.*

1. Density based clustering does not require in general continuous densities.

It is important to realize that, while the cluster tree T_p is a collection of connected subsets of the support of p, the cluster tree estimators considered in this paper are collections of subsets of the sample points.

In order to quantify how well a cluster-tree estimator approximates the true cluster tree, we will make use of the notion of cluster tree consistency put forward by Chaudhuri et al. (2014).

In detail, let $\mathcal{A}_n$ denote a collection of connected subsets of the support of p, which may depend on n. A cluster tree estimator $\widehat{T}_n$ is consistent with respect to $\mathcal{A}_n$ if, with probability tending to 1 as $n \to \infty$, the following holds simultaneously over all A and A' in $\mathcal{A}_n$: the smallest clusters in $\widehat{T}_n$ containing $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ are disjoint. The requirement for consistency outlined above is rather natural: if a cluster tree is deemed consistent with respect to the sequence $\mathcal{A}_n$, then it should, with probability tending to 1, cluster the sample points perfectly well, as if we had the ability of verifying, for each pair of sample points X_i and X_j and each connected set $A \in \mathcal{A}_n$, whether both X_i and X_j are in A.

We allow $\mathcal{A}_n$ to grow larger and more complex with n, so that the cluster tree estimator will be able to discriminate among clusters of p that are barely distinguishable given the size of the sample. An example of a sequence $\{\mathcal{A}_n\}_{n=1}^\infty$ is the set of δ_n-separated clusters according to Definition 2, where the parameter δ_n is taken to be vanishing as $n \to \infty$. The sequence of target subsets $\{\mathcal{A}_n\}_{n=1}^\infty$ may not be chosen to be too large: for example if $\mathcal{A}_n$ equals to the set of all clusters of p, then, depending on the complexity of p, no cluster tree estimator need to be consistent. A natural way to define $\{\mathcal{A}_n\}_{n=1}^\infty$ is by specifying a *separation criterion* for sets, which may become less strict as n grows, and then populate each $\mathcal{A}_n$ using only the connected subsets of the support of p fulfilling such a criterion. In particular, Chaudhuri et al. (2014) develop a criterion known as the (ϵ,σ)-separation, which requires two connected subsets A and A' to be far apart from each other in terms of their "horizontal" distance $d(A,B)$ and their "vertical" distance, in the sense that the smallest cluster containing both A and B should belong to a level set of p indexed by a value of λ significantly smaller to the values indexing the level sets of A and B. See Definition 10 below for details. One of the main contributions of this paper is to replace this rather general notion of separation by a simpler one, the δ-separation criterion in Definition 2, which is better suited deal with smooth densities. This allows us to derive new rates of consistency that depend explicitly on the smoothness of the density.

As explained in Eldridge et al. (2015), the cluster tree consistency guarantees based on separation criteria can be fairy coarse, as they only require $\widehat{T}_n$ to preserve the connectivity of all the sets in $\mathcal{A}_n$. In particular, a tree estimator that is consistent with respect to such definition needs not yield a good clustering of the sample points. Concretely, $\widehat{T}_n$ might have additional unwanted clusters, referred to as *false* in Chaudhuri et al. (2014), that do not correspond to any disjoints sets in $\mathcal{A}_n$, a phenomenon referred to as *over-segmentation* by Eldridge et al. (2015). Similarly, $\widehat{T}_n$ might not conform to the partial order of inclusions among the clusters of p, an issue called *improper nesting*. In fact, the estimators developed in this paper do not suffer from such shortcomings and are consistent in the merge distortion metric of Eldridge et al. (2015), a more refined stronger notion of consistency for cluster trees. See Sections 4.6 and 6 below.

3. The DBSCAN Algorithm

The DBSCAN algorithm, first introduced in Ester et al. (1996), is an extremely popular methodology for "flat" clustering. In this section we introduce a simple generalization of DBSCAN, shown below in Algorithm 1, that yields cluster tree estimators and establish its connections with kernel density estimation.

Algorithm 1 The DBSCAN algorithm.

INPUT: i.i.d sample $\{X_i\}_{i=1}^n$, and $h > 0$.

1. For each $k \in \mathbb{N}$, construct a graph $\mathbb{G}_{h,k}$ with nodes $\{X_i : |B(X_i, h) \cap \{X_j\}_{j=1}^n| \geq k\}$ and edges (X_i, X_j) if $\|X_i - X_j\| < 2h$.
2. Compute $\mathbb{C}(h, k)$, the graphical connected components of $\mathbb{G}_{h,k}$.

OUTPUT: $\{\mathbb{C}(h, k), k \in \mathbb{N}\}$.

For a fixed value of k, Algorithm 1 is in fact a simplified version of the original DBSCAN procedure of Ester et al. (1996), where the parameters h and k are called instead Eps and MinPts, respectively. Notice that, unlike in the original formulation of DBSCAN, we do not distinguish between core and border points and, furthermore, we evaluate connectivity among the sample points using balls of radius $2h$ instead of h. Such modifications have no impact on the rates of consistency we obtain but simplify the derivations.

Assuming $h > 0$ fixed, by sweeping through all the possible values of k, Algorithm 1 produces a sequence of nested geometric graphs $\widehat{T}_n = \{\mathbb{C}(h, k)\}_{k \in \mathbb{N}}$. It is immediate to see that $\widehat{T}_n$ forms a cluster tree estimator over the sample points $\{X_i\}_{i=1}^n$; see Definition 1. This is because, for each $k_1 \leq k_2$,

$$\bigcup_{\{X_i:\ |B(X_i,h)\cap\{X_j\}_{j=1}^n| \geq k_2\}} B(X_i, h) \subseteq \bigcup_{\{X_i:\ |B(X_i,h)\cap\{X_j\}_{j=1}^n| \geq k_1\}} B(X_i, h).$$

In practice, Algorithm 1 can be efficiently implemented using a union-find structure in such a way that the set $\mathbb{C}(h, k)$ of the maximal connected components of $\mathbb{G}_{h,k}$ can be computed without using the potentially expensive breadth-first search or depth-first search algorithms. The resulting cluster tree algorithm is simpler than the estimator based on Wishart's algorithm proposed in Chaudhuri et al. (2014). Indeed, the DBSCAN-based estimator is obtained from a sequence of node-induced sub-graphs of the $2h$-neighborhood graph over the sample points. In contrast, Wishart's algorithm entails taking node and edge-induced sub-graphs of the k-nearest neighborhood graph over $\{X_1, \ldots, X_n\}$, which has higher computational complexity.

As explained in Sriperumbudur and Steinwart (2012), DBSCAN is implicitly using a kernel density estimator with kernel corresponding to the indicator function of the unit d-dimensional Euclidean ball to cluster the points. In detail, consider the density estimator $\widehat{p}_h$ given by

$$x \in \mathbb{R}^d \mapsto \widehat{p}_h(x) = \frac{|B(x, h) \cap \{X_i\}_{i=1}^n|}{nh^d V_d} = \frac{1}{nh^d V_d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right), \tag{2}$$

where

$$K(x) = \begin{cases} 1 & \text{if } x \in B_d(0, 1), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

It is easy to see that $\widehat{p}_h$ is a Lebesgue density, i.e. $\widehat{p}_h(x)$ is a measurable, non-negative function and $\int_{\mathbb{R}^d} \widehat{p}_h(x)dx = 1$. Furthermore, $\mathbb{E}[\widehat{p}_h(x)] = p_h(x)$ for all $x \in \mathbb{R}^d$, where

$$p_h(x) = \frac{1}{h^d V_d} \int_{\mathbb{R}^d} K\left(\frac{x - z}{h}\right) p(z)dz = \frac{P(B(x, h))}{h^d V_d}. \tag{4}$$

For any $\lambda \geq 0$, set $\widehat{D}(\lambda) = \{x \colon \widehat{p}_h(x) \geq \lambda\} \cap \{X_i\}_{i=1}^n$ and

$$\widehat{L}(\lambda) = \bigcup_{X_j \in \widehat{D}(\lambda)} B(X_j, h). \tag{5}$$

Then, setting, for $k > 0$, $\lambda_k = \frac{k}{nh^d V_d}$, one can see that, for clustering purpose, $\mathbb{C}(h, k)$ and $\widehat{L}(\lambda_k)$ convey the same information. Indeed from the definition of $\widehat{L}(\lambda_k)$, it is straightforward to see that

Lemma 1. *Two data points X_i and X_j are in the same connected component of the d-dimensional set $\widehat{L}(\lambda_k)$ if and only if they are in the same connected component of the graph $\mathbb{C}(h, k)$.*

The union of balls $\widehat{L}(\lambda)$ is a renown estimator in the literature on level set estimation, originally studied in Devroye and Wise (1980) (see also Cuevas and Rodríguez-Casal (2004)). In particular, with a suitable choice of the bandwidth parameter h and as n grows unbounded, $\widehat{L}(\lambda)$ is a rate-optimal estimator of the level set $L(\lambda)$ under various loss functions and appropriate assumptions on the underlying density.

4. Clustering Consistency for Hölder Continuous Densities

In this section we show that the DBSCAN algorithm 1 is consistent under Hölder smooth densities. Towards that end, we introduce a new notion of cluster tree consistency, called δ-consistency (see Section 4.2 below), which is well-suited to study cluster trees generated by smooth densities. We will show that DBSCAN, with suitable inputs, will return cluster tree estimators that nearly attain the corresponding minimax optimal rates and that those rates depend on the degree of smoothness of the density.

4.1. Hölder smooth densities

Below we give a recap of well-known results on non-parametric density estimation. Given vectors $s = (s_1, \ldots, s_d)$ in $\mathbb{N}^d$ and $x = (x_1, \ldots, x_d)$ in $\mathbb{R}^d$, set $|s| = s_1 + \cdots + s_d$ and $x^s = x_1^{s_1} \ldots x_d^{s_d}$, and let

$$D^s = \frac{\partial^{s_1 + \cdots + s_d}}{\partial x_1^{s_1} \ldots \partial x_d^{s_d}}$$

denote the differential operator. A function $p : \mathbb{R}^d \to \mathbb{R}$ is said to belong the Hölder class $\Sigma(L, \alpha)$ with parameters $\alpha > 0$ and $L > 0$ if p is $\lfloor \alpha \rfloor$-times continuously differentiable and, for all $x, y \in \mathbb{R}^d$ and all $s \in \mathbb{N}^d$ with $|s| = \lfloor \alpha \rfloor$,

$$|D^s p(x) - D^s p(y)| \leq L\|x - y\|^{\alpha - s}.$$

Notice that, when $0 < \alpha \leq 1$, the Hölder condition reduces to the Lipschitz condition

$$|p(x) - p(y)| \leq L\|x - y\|^{\alpha}, \quad \forall x, y \in \mathbb{R}^d.$$

Let $\widehat{p}_h$ denote a kernel density estimator with bandwidth h and kernel K, that is

$$\widehat{p}_h(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right).$$

Then, we obtain the standard bias-variance decomposition for the KDEs, namely

$$\|\widehat{p}_h - p\|_\infty \leq \|\widehat{p}_h - p_h\|_\infty + \|p_h - p\|_\infty.$$

In order to control the stochastic component $\|\widehat{p}_h - p_h\|_\infty$ we will invoke well-known concentration bounds from density estimation to conclude that, under appropriate and very mild assumptions on K, there exists a constant $C_1 > 0$, depending on $\|p\|_\infty$, d and K, such that, for any $\gamma > 0$ and all n large enough and assuming $nh^d \geq 1$,

$$\mathbb{P}\left(\|\widehat{p}_h - p_h\|_\infty \leq a_n\right) \geq 1 - e^{-\gamma}, \tag{6}$$

where

$$a_n = C_1 \sqrt{\frac{(\gamma + \log(1/h))}{nh^d}}. \tag{7}$$

The verification of this bound can be found in many places in the literature; see, e.g., Giné and Guillou (2002), Sriperumbudur and Steinwart (2012), Jiang (2017b) and Kim et al. (2019). See Appendix B for details. As for the bias term $\|p_h - p\|_\infty$, if K is chosen to be a α-valid kernel[2] (see, e.g., Rigollet and Vert (2009)), then standard calculations yield that

$$\|p_h - p\|_\infty \leq C_2 h^{\alpha}, \tag{8}$$

for an appropriate constant $C_2 > 0$ depending on L and α. In particular, since this type of kernels are polynomials supported on $[0, 1]^d$, they automatically satisfy the VC condition (see lemma 22 of Nolan and Pollard, 1987, for a justification).

Thus combining the bias in (6) and the variance in (8), we conclude that, with probability at least $1 - e^{-\gamma}$, with γ any positive number,

$$\|\widehat{p}_h - p\|_\infty \leq C_1 \sqrt{\frac{\gamma + \log(1/h)}{nh^d}} + C_2 h^{\alpha}. \tag{9}$$

Setting, e.g., $\gamma = \log n$, the optimal choice of the bandwidth is

$$h \asymp \left(\frac{\log n}{n}\right)^{\frac{2\alpha+d}{}}, \tag{10}$$

leading to the final rate of $\left(\frac{\log n}{n}\right)^{\frac{\alpha}{2\alpha+d}}$, which is in fact minimax optimal.

2. For a fixed $\alpha > 0$, a function $K : \mathbb{R}^d \to \mathbb{R}$ is an α-valid kernel if $\int_{\mathbb{R}^d} K(x)dx = 1$, has finite L_p norm for all $p \geq 1$, $\int_{\mathbb{R}^d} \|x\|^{\alpha} K(x)dx < \infty$ and $\int_{\mathbb{R}^d} x^s K(x)dx = 0$ for all $s = (s_1, \ldots, s_d) \in \mathbb{Z}^d$ such that $1 \leq \sum_{i=1}^{d} s_i \leq \lfloor \alpha \rfloor$, where for $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $x^s = \prod_{i=1}^{d} x_i^{s_i}$. See Definition 1 in Rigollet and Vert (2009).

4.2. The δ-Separation Criterion

We now formulate a notion of cluster separation that is naturally suited to smooth densities and, for continuous densities is equivalent to separation in the merge distance of Eldridge et al. (2015) (see also Kim et al., 2016).

Definition 2. *Let A and A' be subsets the support of the density p and set $\lambda := \inf_{x \in A \cup A'} p(x)$. For $\delta \in (0, \lambda)$, A and A' are said to be δ-separated if they belong to distinct connected components of the level set $\{x \colon p(x) > \lambda - \delta\}$.*

Unlike the separation criterion of Chaudhuri et al. (2014), which requires the specification of two parameters quantifying the horizontal and vertical displacement between clusters, δ-separation only uses one parameter The intuition behind the notion of δ-separation is simple: due to the smoothness of the density, the degrees of "vertical" and "horizontal" separation between clusters are coupled. This is illustrated in Figure 1 and best explained for the case of a density in $\Sigma(L, \alpha)$ with $\alpha \leq 1$. If A and A' are δ-separated, then their distance is at least $\left(\frac{\delta}{L}\right)^{1/\alpha}$. As a result, the degree of separation between clusters of smooth densities can be described using only one parameter, a feature that we will exploit to derive new notion of consistency for clustering.

Definition 3. *Let $\delta > 0$ and $\gamma \in (0, 1)$. A cluster tree estimator based on an i.i.d. sample $\{X_i\}_{i=1}^n$ is (δ, γ)-accurate if, with probability no smaller than $1-\gamma$, for any pair of connected subsets A and A' of the support that are δ-separated, exactly one of the following conditions holds:*

1. *at least one of $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ is empty;*
2. *the smallest clusters in the cluster tree estimator containing $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ are disjoint.*

Let $\{\delta_n\}_n$ be a vanishing sequence of positive numbers and a $\{\gamma_n\}$ a vanishing sequence in $(0, 1)$. We say that the sequence of cluster tree estimators $\{T_n\}_n$, where T_n is based on an i.i.d. sample $\{X_i\}_{i=1}^n$, is δ-consistent with rate δ_n if, for all n large enough, T_n is (δ_n, γ_n)-accurate where γ_n decays polynomially in n.

It is important to realize that the notion of δ-consistency is a *uniform* notion of consistency that is required to hold simultaneously over all possibly pairs of δ-separated connected subsets of the support.

The δ-separation criterion is closely related to the concept of the *merge height* introduced by Eldridge et al. (2015). In the context of hierarchical clustering, the merge height is used to describe the "height" at which two points or two clusters merge into one cluster; see Definition 9. In particular we show below in Lemma 4 that if two subsets A and A' of the support are δ-separated and $\inf_{x \in A \cup A'} p(x) = \lambda$, then their merge height is no larger than $\lambda - \delta$. To further emphasize how similar the two approaches are, we mention that our results about cluster consistency still hold for a slightly stronger notion of cluster consistency, whereby condition 2 in Definition 3 is replaced by the condition

2. there exists a level $\lambda \in [\inf_{x \in A \cup A'} f(x) - \delta, \inf_{x \in A \cup A'} f(x))$ such that $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ are contained in two different λ-clusters of the cluster tree estimator.

The key difference between δ-consistency and this stronger version of δ-consistency is that in the latter case we further constrain the split level for A and A' in the cluster tree estimator to occur at a value less than $\inf_{x \in A \cup A'} f(x)$ by an amount no larger than δ_n. This is precisely what is required for merge distance consistency; see Eldridge et al. (2015). We provide more detailed comparison in Section 6, where we further elucidate the differences between our notion of δ-separation and the (ϵ, σ)-separation criterion of Chaudhuri et al. (2014).

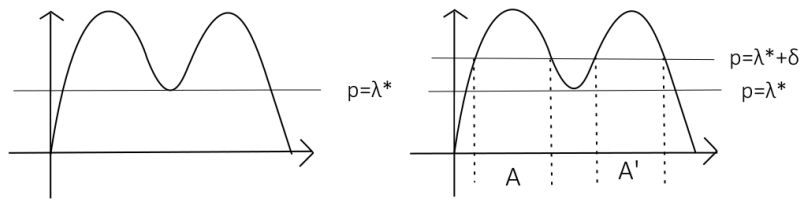


Figure 1: The left figure depicts a split level (defined in λ^* Section 4.3) of the density p. The right figure depicts two sets A and A' being δ separated with respect to λ^*.

4.3. The split levels

One of the most impoertant features of a cluster tree is the collections of levels λ at which the clusters split into two or more disjoint sub-clusters, which we refer to as *split levels.* Such levels belong to the well-known class of "critical levels" in differential topology, which identify critical changes in the topology of the upper level sets of p. See, for example, Hirsch (2012) for more details. In particular, the estimation of split levels is a central theme in the contributions of Sriperumbudur and Steinwart (2012) and in Steinwart (2015). Below, we provide a slightly different characterization of the split levels of continuous densities and relate it to the criterion of δ-separation of clusters. The notion of split levels will be important below in Section 4.4.2 in formalizing conditions under which computationally efficient and statistically optimal cluster tree estimation is feasible for Hölder densities with smoothness degree $\alpha > 1$. It will also be used to demonstrate that we can easily remove false clusters from the cluster tree estimations returned by our algorithms (see Section 4.6).

Definition 4. *Let $p : \mathbb{R}^d \to \mathbb{R}$ be a continuous density function. For a fixed $\lambda^* > 0$, let $\{\mathcal{C}_k\}_{k=1}^K$ be the connected components of $\{p \geq \lambda^*\}$. The value λ^* is said to be a split level of p if there exists a $\mathcal{C}_k$ such that $\mathcal{C}_k \cap \{p > \lambda^*\}$ has two or more connected components.*

The following simple result illustrates the main topological properties of split levels.

Proposition 1. *Suppose that $p : \mathbb{R}^d \to \mathbb{R}$ is compactly supported and that A and A' are subsets of two distinct connected components of $\{p \geq \lambda_1\}$. If A and A' belongs to the same connected components of $\{p \geq \lambda_2\}$, where $\lambda_2 < \lambda_1$, then there is a unique split level $\lambda^* \in [\lambda_2, \lambda_1)$ such that A and A' belong to one connected component of $\{p \geq \lambda^*\}$ and to distinct connected components of $\{p > \lambda^*\}$.*

Proposition 1 suggests that if two connected components merge into one as the density level λ decreases, then there exists one and only one split level at which the corresponding merge takes place. Therefore, the following definition, which characterizes the split level of any two distinct clusters in a cluster tree, seems natural.

Definition 5. *Suppose A and A' are two open subsets of the support of p. Then A and A' are said to split at level λ^* if A and A' belong to one connected component of $\{p \geq \lambda^*\}$ and to two distinct connected components of $\{p > \lambda^*\}$.*

In our next result we illustrate a direct link between the notion of split levels and the criterion of δ-separation introduced above. We will exploit this fact later in Section 4.6 to demonstrate how to prune the cluster tree estimators to yield accurate estimates of the split levels without producing false clusters.

Corollary 1. *Let A and A' be δ-separated. Then there exists a split level λ^* of the density, with*

$$\lambda^* \leq \inf_{x \in A \cup A'} f(x) - \delta,$$

such that A and A' belong to one connected component of $\{p \geq \lambda^*\}$ and to two distinct connected components of $\{p > \lambda^*\}$.

4.4. Rate of consistency for the DBSCAN algorithm

We are now ready to present the main results of the paper, and derive rates of consistency for DBSCAN-based cluster tree estimators with respect to the criterion of δ-separation and for Hölder smooth densities. Specifically, we will show that these estimators are δ-consistent with rate

$$\delta_n \geq C \left(\frac{\log(n)}{n} \right)^{\frac{\alpha}{2\alpha + d}}, \tag{11}$$

for an appropriate constant C that depends on $\|p\|_\infty$, L, K and α. The above rates depend on the smoothness of the underlying density, with smoother densities leading to faster rates, and, as shown in Section 4.5, are in fact minimax optimal. This is one of the main contributions of this article and delivers an extension of the cluster consistency results of Chaudhuri et al. (2014), which are agnostic to the smoothness of p.

4.4.1. Consistency for $\alpha \leq 1$

We first show that, when $\alpha \leq 1$, the DBSCAN algorithm is δ-consistent with rate of order (11). We remark that this type of result can be deduced from several contributions in the literature on density-based clustering, which show that variants of the DBSCAN algorithm lead to some form of cluster consistency when $\alpha \leq 1$. See, e.g., Rinaldo and Wasserman

(2010), Sriperumbudur and Steinwart (2012), Jiang (2017a) and Steinwart et al. (2017). We provide the details for completeness.

In order to demonstrate that DBSCAN is δ-consistent, it will be sufficient to show that the procedure provides an approximation to the upper level sets of $\widehat{p}_h$. This is done in the next result, which relies on general, well-known, finite sample concentration bounds for KDEs along with standard calculations for the bias of a KDE; see Lemma 6 and Proposition 7 in Appendix B.

Lemma 2. *Assume that $p \in \Sigma(L, \alpha)$, where $\alpha \in (0, 1]$, and let K be the spherical kernel. Then, there exist constants C_2, depending on C_1, $\|p\|_\infty$, L and d such that, if $h = C_1 n^{-\frac{1}{2\alpha+d}}$ then uniformly over all $\lambda > 0$, with probability at least $1 - 1/n$,*

$$\left\{p \geq \lambda + C_2 \left(\frac{\log(n)}{n}\right)^{\alpha/(2\alpha+d)} + Lh^\alpha\right\} \subset \bigcup_{X_j \in \widehat{D}(\lambda)} B(X_j, h) \subset \left\{p \geq \lambda - C_2 \left(\frac{\log(n)}{n}\right)^{\alpha/(2\alpha+d)} - Lh^\alpha\right\}. \tag{12}$$

As a direct corollary, we see that the DBSCAN algorithm, with an appropriate choice of the bandwidth h, outputs a δ-consistent cluster tree with consistency rates that depend on α.

Corollary 2. *Assume that $p \in \Sigma(L, \alpha)$, where $\alpha \in (0, 1]$, and let K be the spherical kernel. Then, there exist constants C_1 depending on $\|p\|_\infty$, L and d such that, if $h = C_1 \left(\frac{\log(n)}{n}\right)^{\frac{1}{2\alpha+d}}$, the cluster tree returned by the DBSCAN Algorithm 1 is δ-consistent with rate $\delta_n \geq C \left(\frac{\log(n)}{n}\right) n^{\frac{\alpha}{2\alpha+d}}$, where $C = C(\|p\|_\infty, L, d)$.*

4.4.2. Consistency for $\alpha > 1$

When $\alpha > 1$, Algorithm 1 no longer delivers the optimal rate displayed in (11), for two reasons. The first reason stems from standard non-parametric density estimation considerations: when $\alpha > 1$ it becomes necessary to rely on smoother kernels, namely α-valid kernels as indicated before. This will lead to a bias $\|p - p_h\|_\infty$ of the correct order $O(h^\alpha)$. The second reason is more subtle: the straightforward arguments we used to handle the case of $\alpha \leq 1$ do not lead to optimal clustering rates even if the kernel K is chosen to be α-valid. To exemplify, suppose we would like to cluster the sample points $\{X_i\}_{i=1}^n \cap \{x : \widehat{p}_h(x) \geq \lambda\}$ for some $\lambda > 0$. The computationally efficient linkage rule implemented by DBSCAN is to cluster the points based on the connected components of the union-of-balls around them, i.e based on the connected components of

$$\widehat{L}(\lambda) = \bigcup_{X_j \in \{\widehat{p}_h \geq \lambda\}} B(X_j, h).$$

Assume now that the gradient of p has norm uniformly bounded by a constant D for all $x \in L(\lambda)$. Then,

$$\max_{X_j \in \{\widehat{p}_h \geq \lambda\}} \sup_{x \in B(X_j, h)} |p(x) - p(X_j)| \leq Dh, \tag{13}$$

and, as a result,

$$\left\{p \geq \lambda + C\left(\sqrt{\frac{\log(n)}{nh^d}} + h^\alpha\right) + Dh\right\} \subset \widehat{L}(\lambda) \subset \left\{p \geq \lambda - C\left(\sqrt{\frac{\log(n)}{nh^d}} - h^\alpha\right) - Dh\right\}, \tag{14}$$

where $C\left(\frac{\log(n)}{\sqrt{nh^d}} + h^\alpha\right)$ comes from the L_∞ error bound of α-valid kernels as in (9) and Dh is due to (13). As $h \to 0$, the term Dh dominates the bias term Ch^α, so that the optimal choice of h is of the order $h \asymp \left(\frac{\log n}{n}\right)^{1/(2+d)}$, which in turn yields a worse rate than (11) when $\alpha > 1$. What is more, $\|\nabla p(X_j)\| > 0$ for any X_j away from the critical points of p. This would mean that

$$\min_{X_j \in \{\widehat{p}_h \geq \lambda\}} \|\nabla p(X_j)\| > 0,$$

Then, as $h \to 0$, $\sup_{x \in B(X_j, h)} |p(x) - p(X_j)| \approx \|\nabla p(X_j)\| h = \Theta(h)$. Thus, the inclusions in (14) are tight, showing that the sub-optimal choice of h cannot be ruled out. We believe that this phenomenon is not specific to DBSCAN only, but applies more broadly to the the class of single-linkage-type clustering algorithms. That is, it seems to us that such algorithms are in general unable, like our DBSCAN-based procedure in Algorithm 1, to take advantage of a higher degree of smoothness of the underlying density.

The issue outlined above can be handled in more than one way. A possible solution, which is nearly trivial but impractical, is to deploy a computationally inefficient algorithm that assumes the ability to evaluate the connected components of the upper level set of $\widehat{p}_h$ exactly: see Algorithm 3 in the appendix. It is immediate to see that this approach produces optimal δ-consistency; see Corollary 3 in Algorithm 3. Unfortunately, this procedure will require evaluating $\widehat{p}_h$ on a fine grid, which is computationally infeasible even in small dimensions. The second, more interesting and novel solution which we describe next, is to further assume that p satisfies additional mild regularity conditions around the split levels. The conditions are of geometric and analytic nature and are reminiscent of low-noise type assumptions in classification. Under these conditions, the modified DBSCAN Algorithm 2 will achieve the optimal rate (11) while remaining computationally efficient. This finding, stated formally in Theorem 1 below, is the main result of this section.

Algorithm 2 The modified DBSCAN

INPUT: i.i.d sample $\{X_i\}_{i=1}^n$, a α-valid kernel K and $h > 0$.

1. Compute $\{\widehat{p}_h(X_i), i = 1, \ldots, n\}$.
2. For each $\lambda \geq 0$, construct a graph $\mathbb{G}_{h,\lambda}$ with node set

$$\widehat{D}(\lambda) = \{X_i : \widehat{p}_h(X_i) \geq \lambda\}$$

and edge set $\{(X_i, X_j) : X_i, X_j \in \widehat{D}(\lambda) \text{ and } \|X_i - X_j\| < 2h\}$.

3. Compute $\mathbb{C}(h, \lambda)$, the graphical connected components of $\mathbb{G}_{h,\lambda}$.

OUTPUT: $\widehat{T}_n = \{\mathbb{C}(h, \lambda), \lambda \geq 0\}$.

Remark 1. *Despite its seemingly different form, Algorithm 2 is nearly identical to Algorithm 1. The only difference is the use of an α-valid kernel K instead of a spherical kernel. Furthermore, the procedures only require evaluating at most $n+1$ different graphs:*

$$\mathbb{G}_{h,0}, \mathbb{G}_{h,\widehat{p}_h(X_{\sigma_1})}, \ldots, \mathbb{G}_{h,\widehat{p}_h(X_{\sigma_n})},$$

where $(\sigma_1, \ldots, \sigma_n)$ is a permutation of $(1, \ldots, n)$ such that

$$\widehat{p}_h(X_{\sigma_1}) \leq \widehat{p}_h(X_{\sigma_2}) \leq \ldots \widehat{p}_h(X_{\sigma_n})$$

And, again just like with Algorithm 1, the connected components of each $\mathbb{C}(h, \lambda)$ can be easily evaluated by maintaining a union-find structure.

To formulate the the extra regularity conditions on the geometry of the density $p \in \Sigma(\alpha, L)$ around the split levels that guarantee optimality of the clustering Algorithm 2 we first recall some notions commonly used in the literature on level set and support estimation. Below, Ω denotes a generic subset of $\mathbb{R}^d$ of dimension d.

C1. (The Inner Cone Condition) The subset Ω satisfies the inner cone conditions if ihere exist constants $r_I, c_I > 0$ such that, for any $0 \leq r \leq r_I$ and $x \in \Omega$,

$$\mathcal{L}(B(x, r) \cap \Omega) \geq c_I V_d r^d,$$

where $\mathcal{L}$ denotes the Lebesgue measure of $\mathbb{R}^d$.

C2. (The Covering Condition) The subset Ω satisfies the covering condition if there exists a constant C_I such that, for any $0 < r \leq r_I$, there exists a collection of points $\mathcal{N}_r \subset \Omega$ such that $card(\mathcal{N}_r) \leq C_I r^{-d}$ and

$$\bigcup_{y \in \mathcal{N}_r} B(y, r) \supset \Omega.$$

Both assumptions **C1, C2** are rather mild. If Ω is a compact manifold of dimension $b \leq d$ with piecewise Lipschitz boundary, both assumptions are automatically verified with the dimension d replaced by the intrinsic dimension b. (see, e.g. Do Carmo, 1992). The inner cone condition **C1** is used in Korostelev and Tsybakov (1993) and is well-known as the standard condition (Cuevas, 2009) or, more recently, the (a, b) condition of (Chazal et al., 2015). It is essentially equivalent to the level set regularity condition [B] in Singh et al. (2009). The covering condition **C2** holds automatically if Ω is compact. See, e.g., Rinaldo and Wasserman (2010) and Balakrishnan et al. (2012).

Since $p \in \Sigma(L, \alpha)$ with $\alpha > 1$, any level set $\{p \geq \lambda\}$ is a union of connected d dimensional manifolds with C^1 boundary. Therefore it is natural to require both **C1** and **C2** to hold simultaneously for all the upper level-sets of p *right above the split levels.* Specifically, we will assume the following.

C. There exists a $\delta_0 > 0$ such that, for any split level λ^* of p and any $0 < \delta \leq \delta_0$, the set $\{x \colon p(x) \geq \lambda^* + \delta\}$ satisfies conditions **C1** and **C2** with constants r_I,c_I and C_I only depending on p.

We also need the connected components of the upper level sets right above the split levels to satisfy a low-noise condition as follows.

S(α). There exist positive constants δ_S and c_S such that, for each split level λ^* of the density p, the following holds. Let $\{\mathcal{C}_k\}_{k=1}^K$ be the connected components of $\{x \colon p(x) > \lambda^*\}$. Then,

$$\min_{k \neq k'} d(\mathcal{C}_k \cap \{p \geq \lambda^* + \delta\}, \mathcal{C}_{k'} \cap \{p \geq \lambda^* + \delta\}) \geq c_S \delta^{1/\alpha}, \quad \forall \delta \in (0, \delta_S]. \tag{15}$$

Condition **S(α)** constrains the behavior of the density only around the split levels. It is a fairly common assumption in the literature: it coincides with the *separation exponent* condition of Steinwart (2015) (see Definition 4.2 therein), which quantifies the separation of distinct connected components right above the split levels. Furthermore, **S(α)** is implied by *the local density regularity* conditions of Singh et al. (2009), which in turn is used in Jiang (2017a) to define the *β-regularity* condition for cluster separation. Steinwart (2015) provides several specific examples of densities satisfying the **S(α)** condition. In fact, we prove that conditions **C** and **S(α)** are verified in a large non-parametric class of functions. This class consists of Morse density functions, which are widely used in the density based clustering and mode estimation and topological data analysis; see, e.g., Chacón (2015), Arias-Castro et al. (2016) and references therein. We recall that a function p is Morse if all its critical points have a non-degenerate Hessian. An equivalent and more intuitive condition is that p behaves like a quadratic function around its critical points.

Proposition 2. *Suppose $p : \mathbb{R}^d \to \mathbb{R}$ is a Morse function. Then p satisfies* **C** *and* **S(2)**.

Another interesting class of density functions satisfying conditions **C** and **S(α)** can be obtained as follows. Let $\alpha \geq 2$ be any integer and $f_1 : [0, 1] \to \mathbb{R}$ be such that $f_1(x) = (x-2)^\alpha$. Then, there exists a polynomial f_2 of degree α such that the function on $\mathbb{R}$ defined point-wise as

$$f(x) = \begin{cases} f_1(x), & x \in [1, 2] \\ f_2(x), & x \in [0, 1] \\ 0, \text{ otherwise}, \end{cases}$$

has continuous derivatives up to order $\alpha - 1$ and is such that $f(0) = f'(0) =, \ldots, = f^{(\alpha-1)} = 0$. When $\alpha = 3$, f is a natural spline. For any integer $d \geq 1$, let $F : \mathbb{R}^d \to \mathbb{R}$ be such that $F(x) = f(\|x\|_2)$. Then $F \in \Sigma(\alpha, L)$. Denote $x_0 = (2, 0, \ldots, 0)$. Let $G(x) = F(x - x_0) + F(x + x_0)$. It is easy to see that for any $0 < \delta \leq 1$

$$\{G(x) \geq \delta\} = B(x_0, 2 - \delta^{1/\alpha}) \cup B(-x_0, 2 - \delta^{1/\alpha}).$$

As a result, conditions **C** and **S(α)** are trivially satisfied in this simple case.

Our main result of this section is to prove that that the conclusion of Corollary 2 still holds for $\alpha > 1$, provided that the conditions **C** and **S(α)** are met.

Theorem 1. *Let $p \in \Sigma(\alpha > 1, L)$ be any density function with compact and connected support and finitely many split levels. Suppose that conditions* **C** *and* **S(α)** *hold for p. If*

$h \asymp \left(\frac{\log(n)}{n}\right)^{1/(2\alpha+d)}$), *then, with probability at least* $1 - \frac{1}{n} - O(h^{-d}\exp(-cn^{\alpha/(2\alpha+d)}))$, *the cluster tree returned by the modified DBSCAN Algorithm 2 is* δ*-consistent with rate*

$$\delta_n \geq 2a_n + (4h/c_S)^\alpha$$

where c *is a constant that depends on* p *only,* $a_n = C_1\sqrt{\frac{(\log n + \log(1/h))}{nh^d}} + C_2 h^\alpha$ *is the right hand side of the inequality in* (9) *and* c_S *is defined in* **C**.

The choice of the parameter h in Theorem 1 yields that Algorithm 2 is δ consistent with rate given by (11).

4.5. Lower bounds

Next, we show that the consistent rates of the DBSCAN algorithm derived in the previous sections are nearly minimax optimal, save for a $\log(n)$ term. We point out that the lower bound results by Chaudhuri et al. (2014) are not directly applicable to our problem, since they rely on discontinuous densities.

Theorem 2. *Suppose* $d \geq 1$ *and* $\alpha > 0$. *There exists a finite family* $\mathcal{F}$ *of* d*-dimensional probability density functions belonging to the Hölder class* $\Sigma(L, \alpha)$ *satisfying the conditions* **C**, $\mathbf{S}(\alpha)$ *and uniformly bounded from above by* C_0, *and a constant* $\mathcal{K}$, *depending on* L *and* α, *such that when*

$$n \geq \frac{4^d 8\log(32)}{V_d} \quad \text{and} \quad \delta \leq \min\left\{\left(\frac{\mathcal{K}}{16^\alpha (7C_0)^{\alpha/d}}\right), \|p\|_\infty/(2^{d/2+1})\right\},$$

where V_d *denote the volume of a* d *dimensional ball, the following holds. If cluster tree estimator is* $(\delta_n, 1/4)$*-accurate when presented with an i.i.d. sample from a density function in* $\mathcal{F}$, *then it must be the case that*

$$n \geq \frac{C_0 \mathcal{K}^{d/\alpha}}{C\delta_n^{2+d/\alpha}}, \tag{16}$$

for some constant C *only depends on* d.

Therefore, with the constant C_0 in the previous theorem and the dimension d fixed, the bounds obtained in Theorem 1 and Corollary 2 match the minimax bound in (16), up to a $\log(n)$ factor. Thus, together they show that, up to log factors, the optimal rate for δ-consistency of density functions in $\Sigma(L, \alpha)$ is of order $\left(\frac{\log(n)}{n}\right)^{\alpha/(2\alpha+d)}$.

Interestingly, the minimax clustering rates we derived match the rates for estimation of a Hölder density p under L_∞ norm; see Section 4.1. While this result may not be entirely surprising in light of the findings of, e.g., Eldridge et al. (2015) and Kim et al. (2016), such a a connection has never been formally established, to the best of our knowledge. In particular, our results seem to settle, at least for the class of Hölder-continuous densities and with respect to the criterion of δ-consistency, a long-standing open problem of how density-based clustering compares to density estimation: both problems exhibit the same degree of statistical difficulty.

4.6. Consistent Estimate of The Split Levels

In this section, we present a simple pruning strategy, leading to consistent estimators of the split levels of the the density. While pruning strategies and consistent estimation of split levels have been considered by several authors, such as Sriperumbudur and Steinwart (2012); Steinwart (2015), Chaudhuri et al. (2014) and Jiang (2017a), the existing results do not yield error bounds that depend on the degree of smoothness α for density $p \in \Sigma(L, \alpha)$ with $\alpha > 1$.

The following definition provides a way to identify significant split levels in the cluster tree estimator returned by Algorithm 2.

Definition 6. *Let* $\Delta > 0$. *The random variable* $\widehat{\lambda^*} \in (0, \infty)$ *is said to be a* Δ*-significant split level of the cluster tree estimator if there exist two data points* $X_i, X_j \in D(\widehat{\lambda^*} + \Delta)$ *such that*

$$\widehat{\lambda^*} = \sup\{\lambda > 0 : X_i \text{ and } X_j \text{ are in the same connected component of } \mathbb{C}(h, \lambda).\} \tag{17}$$

Below, we show that there is a one to one correspondence between Δ-significant split levels of the modified DBSCAN cluster tree estimator from Algorithm 2 and the split levels of the population density under a slightly stronger covering condition than condition **C** given above. Specifically, we assume the following.

C'. There exists a constant $\delta_0 > 0$ such that, for any split level λ^* of p and any $\delta \in \mathbb{R}$ with $|\delta| \leq \delta_0$, $\{p \geq \lambda^* + \delta\}$ satisfies conditions **C1** and **C2** with constants r_I,c_I and C_I only depending on p.

The only difference between **C** and **C'** is that while condition **C** assumes some regularity of p only above split levels, **C'** requires the same type of regularity *around* split levels.

Proposition 3. *Suppose condition* **C'** *and* $\mathbf{S}(\alpha)$ *hold. Let* $\Delta = 2a_n + (4h/c_S)^\alpha$ *where* a_n *is defined in* (9) *and* $h = C_1 n^{-1/(2\alpha+d)}$. *Suppose* p *has finitely many split levels. Then, with probability at least* $1 - 1/n - O(h^{-d}\exp(-cn^{\alpha/(2\alpha+d)}))$, *the following additional results hold:*
1. *Let* λ^* *be a split level of the density* p. *Suppose* $\mathcal{C}$ *and* $\mathcal{C}'$ *are two open sets splitting at* λ^* *(see Definition 5) and that*

$$\min\{P(\mathcal{C} \cap \{p \geq \lambda^* + 2\Delta\}), P(\mathcal{C}' \cap \{p \geq \lambda^* + 2\Delta\})\} > 0. \tag{18}$$

Then, there exists a Δ*-significant split level* $\widehat{\lambda^*}$ *of the cluster tree estimator returned by the modified DBSCAN such that*

$$|\lambda^* - \widehat{\lambda^*}| \leq \Delta. \tag{19}$$

2. *Conversely, suppose that* $\widehat{\lambda^*}$ *is a* Δ*-significant split level of the cluster tree estimator. Then there exists a split level* λ^* *of* p *such that*

$$|\lambda^* - \widehat{\lambda^*}| \leq \Delta. \tag{20}$$

Proposition 3 says that, with high probability, every Δ-split level corresponds to a density split level and that conversely, any split level of the density p can be found if we have enough data. To prune the cluster tree returned by the modified DBSCAN algorithm, it suffices to remove all the split levels that are not Δ significant.

5. Densities with Gaps

We now consider the particular scenario where the density p exhibiting a jump discontinuity in such a way that, for all levels λ in a given interval of length ϵ, the upper level sets $\{x : p(x) \geq \lambda\}$ do not change. The value of ϵ is referred to as the *gap size*. We provide a formal definition next, which we formulate within the general measure-theoretic language of Steinwart (2015) since the underlying densities are not continuous. We recall that $\mathcal{L}$ denotes the Lebesgue measure on $\mathbb{R}^d$.

Definition 7 (Distribution with a gap)**.** *Let P a probability measure on $\mathbb{R}^d$, absolutely continuous with respect to the Lebesgue measure. For any $\lambda^* > 0$, let S_{λ^*} be the support of the sub-probability measure*

$$A \mapsto P(A \cap \{x : p'(x) \geq \lambda^*\}), \quad A \text{ Lebesgue measurable,}$$

where p' is any density of P. Then, P is said to have a gap at λ_* of size ϵ, where $0 < \epsilon < \lambda^*$, if $P(S_{\lambda^*}) > 0$ with $\mathcal{L}(\partial S_{\lambda^*}) = 0$ and

$$S_{\lambda_* - \eta} \backslash S_{\lambda_*} = \emptyset, \quad \forall \eta \in (0, \epsilon). \tag{21}$$

We impose the condition that $\mathcal{L}(\partial S_{\lambda^*}) = 0$ in order to avoid pathological cases.

The above definition is independent of the choice of the density of P. At the same time, it also implies that P admits a Lebesgue density p such that

$$S_{\lambda^*} = \mathrm{cl}\left(\{x : p(x) \geq \lambda^*\}\right) \quad \text{and} \quad \mathrm{cl}(S^c_{\lambda^*}) = \mathrm{cl}\left(\{x : p(x) \leq \lambda^* - \epsilon\}\right). \tag{22}$$

Here, given any set A, $\mathrm{cl}(A)$ denotes the closure of A. Indeed, it is not hard to see that, if p' is any Lebesgue density of P, then the function

$$p(x) = \begin{cases} \max\{p'(x), \lambda^*\} & x \in S_{\lambda^*} \\ \min\{p'(x), \lambda^* - \epsilon\} & x \in S^c_{\lambda^*} \end{cases}$$

is also a density of P, and satisfies (22). Thus, with a slight abuse of notation, we may also speak of "the" density p even in this case, with the understanding that we are referring to any density of P for which (22) holds. See Figure 2 for an illustration.

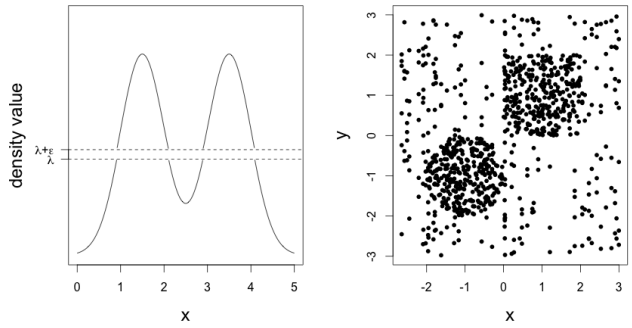


Figure 2: The left plot depicts a one dimensional density with gap of size ϵ at level λ. It is clear that $\{\lambda < p \leq \lambda + \epsilon\}$ is an empty set. The right plot depicts 500 i.i.d sampling from a two dimensional density with a gap. It is clear that the density is low in the background and high in the disk centered at (-1,-1) and in the square centered at (1,1). Finding the samples points with low density values can be thought of as outliers detection in this case.

Though fairly restrictive, the scenario of a distribution with a gap is quite interesting for the purpose of both clustering and level set estimation. Indeed, this situation encompasses the ideal clustering scenario, depicted as examples in Figures 1 and 5 in the original DBSCAN paper Ester et al. (1996), of a piece-wise constant density that is low everywhere on its support with the exception of a few connected, full-dimensional regions, or clusters, where it is higher by a certain amount (in our case the gap ϵ). The size of the gap parameter ϵ and the minimal distance among clusters both affect the difficulty of the clustering task, which becomes harder as the parameters get smaller.

To formally capture the dependence on the distance between clusters we set

$$S_{\lambda^*} = \bigcup_{i=1}^{I} \mathcal{C}_i,$$

where $(\mathcal{C}_1, \ldots, \mathcal{C}_I)$ are disjoint, connected, (necessarily) closed sets of positive P measure and let

$$\sigma = \min_{i \neq j} \mathrm{dist}(\mathcal{C}_i, \mathcal{C}_j) > 0 \tag{23}$$

be the minimal distance between them. The separation parameter σ captures an aspect of the intrinsic difficulty of the clustering task that is complementary to the one quantified by gap parameter ϵ: clusters that are at a small distance σ from each other are hard to separate, for any given value of ϵ. In our analysis, we let both the gap parameter ϵ and

the separation parameter σ vary with n (though we do not make this dependence in out notation for ease of readability), thus allowing for harder clustering problems as a function of the sample size.

In the next simple result, we show that a flat version of the vanilla DBSCAN algorithm given in Algorithm 1, with suitable choices for the input parameters, can optimally estimate the clusters at λ^* at a rate that depend explicitly on both ϵ and σ.

Proposition 4. *Let $\{X_1, \ldots, X_n\}$ be an i.i.d. sample from a probability distribution P that has a gap of size ϵ at level λ^*. Set $a_n = C_1\sqrt{\frac{\log(n)+\log(1/h)}{nh^d}}$ as in (7) and suppose the input parameters h and k of the DBSCAN algorithm are such that*

$$\sigma/4 \geq h \geq C\left(\frac{\log(n)}{n\epsilon^2}\right)^{1/d} \quad \text{and} \quad k = \lceil nh^d V_d \lambda \rceil, \tag{24}$$

for any $C > 0$ such that $2a_n < \epsilon$ and any λ in $(\lambda^* - \epsilon + a_n, \lambda^* - a_n]$. Then with probability at least $1 - 1/n$:

i. simultaneously over all connected sets A such that $A_{2h} \subset \mathcal{C}_i$, for some i, all the sample points in A, if any, belong to the same connected component of $\mathbb{G}_{k,h}$;

ii. simultaneously over all connected sets A and A' such that $A_{2h} \subset \mathcal{C}_i$ and $A'_{2h} \subset \mathcal{C}_j$, for some $i \neq j$, the sample points in A and A', if any, belong to distinct connected components of $\mathbb{G}_{k,h}$.

The definition of A_{2h} and A_{-2h} can be found in (1). Notice that the above results hold true for all n large enough such that $a_n < \epsilon/2$. Proposition 4 implies the DBSCAN algorithm will yield clustering consistency, in the sense of Chaudhuri et al. (2014), provided that its input parameters fulfill (24) holds. In particular, this result requires that the sample size relates to the gap parameter ϵ and the separation parameter σ according to the inequality

$$n \geq C\frac{1}{\epsilon^2\sigma^d},$$

for some constant C, depending on d. In fact, such scaling is nearly minimax optimal: no other clustering algorithms can guarantee cluster consistency under the same assumptions and with a better sample complexity as a function of both ϵ and σ. This results follows from the lower bound guarantee given in Theorem VI.1 of Chaudhuri et al. (2014), where we take notice that the parameters σ and ϵ have different, though related, meaning; see Appendix C.4. In fact, the results in Proposition 4 can be further extended to hold over more general settings of arbitrary densities; see Appendix D.

We conclude by noting that the gap size ϵ and the separation parameters σ quantify two very separate notions of intrinsic difficulty of clustering that are unrelated to each other, and the clustering problem becomes impossible whenever either one of them becomes so small to violate the lower bound (53), regardless of the other. In particular, it is easy to give examples in which the clustering task is impossible to solve because ϵ is too small even if σ is large, and the other way around. As a result, the overall hardness of the clustering problem around the gap is a combination of these two parameters. This is in contrast with the settings considered earlier, where, due to the smoothness of the underlying density, only one parameter is sufficient to capture separation among clusters.

5.1. The Devroye-Wise estimator of the Level Sets

The assumption of a density with gap allows us to carry out a further analysis of the DBSCAN algorithm, showing that it is also minimax optimal for estimating the level set itself S_{λ^*}. For this purpose, the DBSCAN algorithm reduces to the renown Devroye-Wise estimator: see Devroye and Wise (1980). Below we provide a novel, sharper analysis of this estimator, where we allow the size of the gap ϵ to decrease with n, and demonstrate that its rate-optimal (again, we will not explicitly express this dependence in our notation for simplicity). To the best of our knowledge, such scaling has not been previously established.

Recall that the DBSCAN algorithm with inputs k and h outputs a set of nodes $\mathbb{G}_{h,k}$. One then may construct the estimator

$$\widehat{S}_h = \bigcup_{X_j \in \mathbb{G}_{h,k}} B(X_j, h) \tag{25}$$

comprised of a union of balls around such points, and use it as an estimator of the corresponding high density region $S = S_{\lambda^*} = \cup_{i=1}^{I}\mathcal{C}_i$ consisting of all the clusters.

We measure the performance of any estimator $\widehat{S}$ with the Lebesgue measure of its symmetric difference with S:

$$\mathcal{L}\left(S\Delta\widehat{S}\right) = \mathcal{L}\left(S \cap \widehat{S}^c\right) + \mathcal{L}\left(S^c \cap \widehat{S}\right).$$

We will in addition impose the following condition:

R. (Level set regularity). There exists constants $h_0 > 0$ and $C_0 > 0$ such that, for all $h \in (0, h_0)$,

$$\mathcal{L}\left(S_h \setminus S_{-h}\right) \leq C_0 h,$$

where S_h and S_{-h} are defined in (1).

Condition **R** is very mild. Indeed, if ∂S is C^2, then the set $N_h := S_h \backslash S_{-h}$ is the tubular neighborhood (see, e.g. Hirsch, 2012) of ∂S in $\mathbb{R}^d$. In particular, every compact domain in $\mathbb{R}^d$ with C^2 boundary satisfies condition **R**. In this case $C_0 \preceq V_{d-1}|\partial S|$, where $|\partial S|$ denotes the surface volume of S. We also note that **R** is equivalent to the "smooth boundary" condition in Steinwart (2015).

Proposition 5. *Let $\{X_1, \ldots, X_n\}$ be an i.i.d. sample from a probability distribution P that has a gap of size ϵ at level λ^*. Let $a_n = C\sqrt{\frac{\log(n)+\log(1/h)}{nh^d}}$ be defined as in (7). Suppose the input parameters (h, k) of the DBSCAN algorithm satisfy*

$$h_0 \geq h \geq C\left(\frac{\log(n)}{n\epsilon^2}\right)^{1/d} \quad \text{and} \quad k = \lceil nh^d, V_d\lambda \rceil \tag{26}$$

for any $C > 0$ such that $2a_n < \epsilon$ and any λ in $(\lambda^* - \epsilon + a_n, \lambda^* - a_n]$. Then with probability at least $1 - 1/n$,

$$\mathcal{L}(S\triangle\widehat{S}_h) \leq 2C_0 h,$$

*where C_0 is defined in **R** and*

$$\widehat{S}_h = \bigcup_{X_j \in \mathbb{G}_{h,k}} B(X_j, h) \tag{27}$$

Similarly to Proposition 4, the above results hold true for all n large enough such that $a_n < \epsilon/2$. If $P(X \in S) = 1$, the level set estimator $\widehat{S}_h$ is also a support estimator and $\epsilon = \inf_{x \in S} p(x)$. In this case, Proposition 5 says that if the lower bound on the density vanishes no faster than $O(n^{-1/2})$, then support estimation is still possible.

Below we show that the error bound given in Proposition 5 is minimax optimal up to log factors. Consider $\mathcal{P}^n(h_0, \epsilon)$, the class of probability distribution of n i.i.d. random vectors in $\mathbb{R}^d$ whose common density exhibit a gap of size ϵ such that condition (22) holds, and satisfying condition **R** with parameter $h_0 > 0$. Then Proposition 5 shows that, for all n large enough,

$$\sup_{P \in \mathcal{P}^n(h_0,\epsilon)} \mathbb{E}_P \left(S \triangle \widehat{S}_h \right) = O\left(\left(\frac{\log(n)}{n\epsilon^2} \right)^{1/d} \right),$$

provided that h is of the order $\left(\frac{\log(n)}{n\epsilon^2}\right)^{1/d}$. Our next result provides a nearly-matching lower bound.

Proposition 6. *There exist constants h_0 and c, depending only on d such that for any $\epsilon \leq 1/4$, for all n large enough here exist probability distributions $\{P_1, \ldots, P_M\}$ in $\mathcal{P}^n(h_0, \epsilon)$ such that*

$$\inf_{\widehat{S}} \sup_{i=1,\ldots,M} \mathbb{E}_{P_i} \left(\mathcal{L}(\widehat{S} \triangle S) \right) \geq c \min \left\{ \left(\frac{1}{n\epsilon^2} \right)^{1/d}, 1 \right\},$$

where the infimum is with respect to all estimators of S.

Thus, if $\frac{\log(n)}{n\epsilon^2} \to 0$ as $n \to \infty$ (so that condition (26) is eventually satisfied), then the bounds given in Proposition 5 and Proposition 6 match, up to a $\log(n)$ factor. That is, with suitable choice of input, DBSCAN can optimally estimate the level set S at the gap.

The performance of the Devroye-Wise estimator is a well-established topic in the literature: see, e.g., Theorem 4 and 5 in Cuevas and Rodríguez-Casal (2004). Our contribution in this regard is two fold: we allow for an explicit dependence on the gap size parameter ϵ and deliver minimax lower bounds. Our rate of convergence confirms the intuition that a smaller gap size leads to a harder estimation problem.

6. Discussion

In this article we propose a new notion of consistency for estimating the clustering structure under various conditions. Our analysis shows that the DBSCAN algorithm is minimax optimal. Interestingly, the rates match, up to log terms, minimax rates for density estimation in the supreme norm for Höloder smooth densities. In particular, our results provide a complete, rigorous justification to the plausible belief, commonly held in density-based clustering, that clustering is as difficult as density estimation. In the rest of the discussion section, we will compare our notion of separation with other existing ones in the literature. For the sake of exposition, we will follow the convention used in much of the literature on density-based clustering of assuming that the cluster tree of the data generating distribution in fact corresponds to the hierarchy of the upper level sets of a canonical density p. As explained in Steinwart (2015), this definition is in general not well-posed, since different densities will yield different trees.

6.1. Hartigan consistency in Hartigan (1981)

We follow Chaudhuri and Dasgupta (2010) and Eldridge et al. (2015) in defining Hartigan consistency in terms of the density cluster tree.

Definition 8 (Hartigan consistency)**.** *Let $\widehat{T}_n$ be a cluster tree estimator constructed from i.i.d. data $\{X_i\}_{i=1}^n$ from a disribution P with Lebesgue density p. For any pair of subsets A and A', let denote A_n and A'_n be the smallest clusters of $\widehat{T}_n$ containing $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$, respectively. The cluster tree estimator $\widehat{T}_n$ is Hartigan consistent if, for any pair of sets A and A' belonging to distinct connected components of $\{x : p(x) \geq \lambda\}$ for some λ, $P(A_n \cap A'_n = \emptyset) \to 1$ as $n \to \infty$.*

It is immediate from Definition 3 that a δ-consistent cluster tree is also Hartigan consistent. While Hartigan consistency is a simple form of *point-wise* cluster tree consistency, which holds for each fixed pairs of disjoint clusters, δ-consistency is a stronger guarantee, as it yields *uniform* consistency over all δ-separated clusters and, furthermore, gives consistency rates depending on the value of the separation parameter δ.

6.2. Comparison with the Merge distortion metric

The notion of δ-separation is closely related to the notion of *merge distance* introduced by Eldridge et al. (2015), which we present next.

Definition 9. *Let p and q be Lebesgue densities in $\mathbb{R}^d$ and let T_p and T_q be the corresponding cluster density trees. The merge distortion distance between T_p and T_q is defined as*

$$d_M(T_p, T_q) = \sup_{x,y \in \mathbb{R}^d} |m_p(x,y) - m_q(x,y)|,$$

where, for a Lebesgue density p,

$$m_p(x,y) = \sup\{\lambda > 0 \in \mathbb{R} : \textit{there exists } C \in T_p(\lambda) \textit{ such that } \{x,y\} \subset C\}.$$

The original definition of merge distortion metric is, in fact, more general but, when specialized to our settings, reduces to the one given above.

The merge distortion distance is closely related to the L_∞ distance between densities. In fact, by Theorem 17 in Eldridge et al. (2015), $d_M(T_p, T_q) \leq \|p - q\|_\infty$, so that, if $\{p_n\}_n$ is a sequence of Lebesgue densities, then $\|p_n - p\| \to 0$ implies that $d_M(T_{p_n}, T_p) \to 0$. In fact, Lemma 1 in Appendix F of Kim et al. (2016) shows that, if p and q are continuous, then $d_M(T_p, T_q) = \|p - q\|_\infty$. As a result, for the class of continuous (and, in particular, Hölder smooth) densities, cluster consistency in the merge distortion distance is equivalent to cluster consistency based on the δ-separation criterion, which in turn is equivalent to estimation consistency of the underlying density in the L_∞ norm.

The above statement immediately applies to the näive cluster tree estimator $T_{\widehat{p}_{h_n}}$ built using the level sets of any density estimator $\widehat{p}_{h_n}$ that is continuous and, as $h_n \to \infty$, consistent in the L_∞ norm. Such estimator is of course computationally unfeasible even in low dimension. In fact, the DBSCAN-based procedures described above in Algorithms 1 and 2, which are applicable in high-dimensional settings, are also consistent in the merge-distortion metric. To see this, and following the arguments of Eldridge et al. (2015), it

is sufficient to demonstrate the properties of minimality and separation, as defined in that reference, for the DBSCAN cluster-tree estimators. The separation property, which prevents the emergence of false clusters or over-segmentation, follows directly from the definition of δ-separation; see also the pruning results of Section 4.6. On the other hand, minimality avoids the occurrence of improper nesting and holds for the DBSCAN procedures we consider here in virtue of Lemma 1 and Lemma 6. The fact that our algorithms produce cluster tree estimators that are consistent in the merge distance should not be surprising, since δ-consistency is directly tied to consistency for density estimation in in L_∞. As shown in Eldridge et al. (2015), the robust single-linkage clustering algorithm of Chaudhuri et al. (2014) is also consistent in the merge distance.

6.3. Comparison with the (ϵ, σ)-separation criterion of Chaudhuri et al. (2014)

The criterion of δ-separation we introduce in this paper is most useful when studying smooth densities. Nonetheless, it will be helpful to compare it to the notion of (ϵ, σ)-separation defined in Chaudhuri et al. (2014), which is applicable to arbitrary densities.

Definition 10 ((ϵ, σ)-separation criterion in Chaudhuri et al. (2014)).

1. *Let f be a density supported on $X \subset \mathbb{R}^d$. We say that $A, A' \subset X$ are (ϵ, σ)-separated if there exists $S \subset X$ (the separator set) such that (i) any path in X from A to A' intersects S, and (ii) $\sup_{x \in S_\sigma} f(x) < (1-\epsilon) \inf_{x \in A_\sigma \cup A'_\sigma} f(x)$.*
2. *Suppose an i.i.d samples $\{X_i\}_{i=1}^n$ is given. An estimate of the cluster tree is said to be (ϵ, σ) consistent if for any pair A and A' being (ϵ, σ) separated, the smallest cluster containing $A \cap \{X_i\}_{i=1}^n$ is disjoint from the smallest cluster containing $A' \cap \{X_i\}_{i=1}^n$.*

In the following result we make a straightforward connection between the δ-separation and (ϵ, σ)-separation.

Lemma 3. *Assume that $p \in \Sigma(L, \alpha)$ with $\alpha \leq 1$ and that A and A' are δ-separated. Then, A and A' are (ϵ, σ)-separated with*

$$S = \{x : p(x) \leq \lambda - \delta\}, \quad \epsilon = \delta/(3\lambda) \quad \text{and} \quad \sigma^\alpha = \delta/(3L), \tag{28}$$

where $\lambda = \inf_{z \in A \cup A'} p(z)$.

Proof of lemma 3. Denote $\lambda = \inf_{z \in A \cup A'} p(z)$. Suppose for the sake of contradiction that there is a path l connects A and A' and that $l \cap \{p \leq \lambda - \delta\} = \emptyset$. Then by the continuity of p and the compactness of l, there exist $\gamma > 0$ such that $l \subset \{p \geq \lambda - \delta + \gamma\}$. Thus A and A' belongs to the same path connected component of $\{p \geq \lambda - \delta + \gamma\}$. Since $\{p \geq \lambda - \delta + \gamma\} \subset \{p > \lambda - \delta\}$, A and A' belongs to the same path connected component of $\{p > \lambda\}$. Since $\{p > \lambda\}$ is an open set, A and A' be belongs to the same connected component of $\{p > \lambda\}$. This is a contradiction.

Let $\sigma^\alpha = \delta/(3L)$ and $\epsilon = \delta/3$, then for any $x \in S_\sigma$, $p(x) \leq \lambda - \delta + L\sigma^\alpha = \lambda - 2\delta/3$. Similarly if $x \in A_\sigma \cup A'_\sigma$, $p(x) \geq \lambda - L\sigma^\alpha = \lambda - \delta/3$. Thus

$$(1-\epsilon) \inf_{x \in A_\sigma \cup A'_\sigma} f(x) > (1-\epsilon)\lambda - \delta/3 = \lambda - 2\delta/3 > \sup_{x \in S_\sigma} f(x)$$

□

According to the separation criterion in Definition 10, two clusters can be (ϵ, σ)-separated for many the values of ϵ and σ. In particular, by taking the separator set to be larger, it is easy to produce examples of δ-separated clusters that are also (ϵ, σ)-separated such that δ is big but σ is small. This is simply because σ is heavily associated with S. And conversely, by taking an almost flat density function, it is possible to have a very large σ and very small δ.

We remark that when $\alpha > 1$, there is no obvious relationship between the parameter (σ, ϵ) in Definition 10 and δ in Definition 3 as that in Lemma 3. For $\alpha > 1$, while $p \in \Sigma(L, \alpha)$ implies that p is Lipschitz continuous, the Lipschitz constant in this case does not depend on L and α in a simple manner. As a result, the parameter σ, representing the distance between connected components of upper level sets of p, is not straightforwardly related to δ.

References

Ery Arias-Castro, David Mason, and Bruno Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *Journal of Machine Learning Research*, 17(1):1487–1514, 2016.

Amparo BaÍllo, Antonio Cuevas, and Ana Justel. Set estimation and nonparametric detection. *Canadian Journal of Statistics*, 28(4):765–782, 2000.

Sivaraman Balakrishnan, Srivatsan Narayanan, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Cluster trees on manifolds. In *Advances in Neural Information Processing Systems*, 2012.

José E Chacón. A population background for nonparametric density-based clustering. *Statistical Science*, 30(4):518–532, 2015.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for the cluster tree. In *Advances in Neural Information Processing Systems*, pages 343–351, 2010.

Kamalika Chaudhuri, Sanjoy Dasgupta, Samory Kpotufe, and Ulrike von Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912, 2014.

Frédéric Chazal, Marc Glisse, Catherine Labruère, and Bertrand Michel. Convergence rates for persistence diagram estimation in topological data analysis. *Journal of Machine Learning Research*, 16(1):3603–3635, 2015.

Yen-Chi Chen, Christopher R Genovese, and Larry Wasserman. Density level sets: Asymptotics, inference, and visualization. *Journal of the American Statistical Association*, 112 (520):1684–1696, 2017.

Antonio Cuevas. Set estimation: Another bridge between statistics and geometry. *Boletín de Estadística e Investigación Operativa*, 25(2):71–85, 2009.

Antonio Cuevas and Ricardo Fraiman. A plug-in approach to support estimation. *The Annals of Statistics*, pages 2300–2312, 1997.

Antonio Cuevas and Alberto Rodríguez-Casal. On boundary estimation. *Advances in Applied Probability*, 36(2):340–354, 2004.

Luc Devroye and Gary L. Wise. Detection of abnormal behavior via nonparametric estimation of the support. *SIAM Journal on Applied Mathematics*, 38(3):480–488, 1980.

Manfredo Perdigao Do Carmo. *Riemannian geometry*. Birkhauser, 1992.

Justin Eldridge, Mikhail Belkin, and Yusu Wang. Beyond hartigan consistency: Merge distortion metric for hierarchical clustering. In *Proceedings of The 28th Conference on Learning Theory*, pages 588–606, 2015.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Kdd, pages 226–231, 1996.

Junhao Gan and Yufei Tao. Dbscan revisited: Mis-claim, un-fixability, and approximation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, 2015.

Evarist Giné and Armelle Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l'IHP Probabilités et statistiques*, 38:907–921, 2002.

John A Hartigan. Consistency of single linkage for high-density clusters. *Journal of the American Statistical Association*, 76(374):388–394, 1981.

Morris W Hirsch. *Differential topology*, volume 33. Springer Science & Business Media, 2012.

Jennifer Jang and Heinrich Jiang. Dbscan++: Towards fast and scalable density clustering. *arXiv preprint arXiv:1810.13105*, 2018.

Heinrich Jiang. Density level set estimation on manifolds with dbscan. *arXiv preprint arXiv:1703.03503*, 2017a.

Heinrich Jiang. Uniform convergence rates for kernel density estimation. In *International Conference on Machine Learning*, pages 1694–1703, 2017b.

Heinrich Jiang, Jennifer Jang, and Ofir Nachum. Robustness guarantees for density clustering. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3342–3351, 2019.

Jisu Kim, Yen-Chi Chen, Sivaraman Balakrishnan, Alessandro Rinaldo, and Larry Wasserman. Statistical inference for cluster trees. In *Advances in Neural Information Processing Systems 29*, pages 1839–1847. 2016.

Jisu Kim, Jaehyeok Shin, Alessandro Rinaldo, and Larry Wasserman. Uniform convergence rate of the kernel density estimator adaptive to intrinsic volume dimension. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3398–3407, 2019.

Jussi Klemelä. Complexity penalized support estimation. *Journal of multivariate analysis*, 88(2):274–297, 2004.

Jussi Klemelä. *Smoothing of Multivariate Data: Density Estimation and Visualization*. Wiley, 2009.

Aleksandr Petrovich Korostelev and Alexandre B Tsybakov. *Minimax theory of image reconstruction*, volume 82. Springer Science & Business Media, 1993.

Samory Kpotufe and Ulrike V Luxburg. Pruning nearest neighbor cluster trees. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 225–232, 2011.

James R Munkres. *Topology*. Prentice Hall, 2000.

Deborah Nolan and David Pollard. U-processes: rates of convergence. *The Annals of Statistics*, pages 780–799, 1987.

Mathew D. Penrose. Single linkage clustering and continuum percolation. *Journal of Multivariate Analysis*, 53:94–109, 1995.

Wolfgang Polonik. Measuring mass concentrations and estimating density contour clusters-an excess mass approach. *The Annals of Statistics*, pages 855–881, 1995.

Philippe Rigollet and Régis Vert. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, pages 1154–1178, 2009.

Alessandro Rinaldo and Larry Wasserman. Generalized density clustering. *The Annals of Statistics*, pages 2678–2722, 2010.

Alessandro Rinaldo, Aarti Singh, Rebecca Nugent, and Larry Wasserman. Stability of density-based clustering. *The Journal of Machine Learning Research*, 13:905–948, 2012.

Arrti Singh, Clayton Scott, Robert Nowak, and Aarti Singh. Adaptive hausdorff estimation of density level sets. *The Annals of Statistics*, 37:2760–2782, 2009.

Bharath K Sriperumbudur and Ingo Steinwart. Consistency and rates for clustering with dbscan. In *AISTATS*, pages 1090–1098, 2012.

Ingo Steinwart. Adaptive density level set clustering. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 703–738, 2011.

Ingo Steinwart. Fully adaptive density-based clustering. *The Annals of Statistics*, 43(5): 2132–2167, 2015.

Ingo Steinwart, Bharath K. Sriperumbudur, and Philipp Thomann. Adaptive clustering using kernel density estimators. arXiv preprint arXiv:1708.05254, 2017.

Werner Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of classification*, 20(1):025–047, 2003.

Werner Stuetzle and Rebecca Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19(2), 2010.

Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Series in Statistics, 2009.

Alexandre B Tsybakov et al. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969, 1997.

Bingchen Wang, Chenglong Zhang, Lei Song, Lianhe Zhao, Yu Dou, and Zihao Yu. Design and optimization of dbscan algorithm based on cuda. *arXiv preprint arXiv:1506.02226*, 2015.

RM Willett and Robert D Nowak. Minimax optimal level-set estimation. *IEEE Transactions on Image Processing*, 16(12):2965–2979, 2007.

Appendix A. Topological Preliminaries

For completeness, we review the definition of connectedness from the general topology.

Definition 11 (Munkres (2000) Chapter 3)**.** *Let U be any nonempty subset in $\mathbb{R}^d$. Then U is said to be connected, if, for every pair of open subsets A, A' of U such that $A \cup A' = U$, we have either $A = \emptyset$ or $A' = \emptyset$. The maximal connected subsets of U are called the connected components of U.*

We briefly explain why the connected components naturally introduce a hierarchical structure to the level sets of p. Let $\lambda_1 > \lambda_2$, so we have $\{p \geq \lambda_1\} \subset \{p \geq \lambda_2\}$.

- Suppose A is any subset of $\mathbb{R}^d$, and A belongs to the same connected component of $\{p \geq \lambda_1\}$. Then A is contained in the same connected component of $\{p \geq \lambda_2\}$.
- Suppose $A \cup A' \subset \{p \geq \lambda_1\}$ and they belong to distinct connected components of $\{p \geq \lambda_2\}$ Then A and A' are not contained in the same connected component of $\{p \geq \lambda_1\}$.

We also review a closed related concepts, which is call the path connectedness in general topology.

Definition 12. *We say that a subset $U \subset \mathbb{R}^d$ is path connected if for any $x, y \in U$, there exists a path continuous $\mathcal{P} : [0,1] \to U$ such that $\mathcal{P}(0) = x$ and $\mathcal{P}(1) = y$.*

The main reason we introduce the path connectedness is that if U is an open set in $\mathbb{R}^d$, then U is connected if and only if it is path connected. Therefore a simple but useful consequence is that for any λ, the connected components of $\{p > \lambda\}$ are also the path connected components.
We will repeatedly use these topological properties in our analysis without further mentioning. The proofs of them are omitted and can be found in Munkres (2000) or any other books on general topology.

Appendix B. Proofs from Section 4

We begin by justifying (6). Since this is a well known result, we simply use a result of Sriperumbudur and Steinwart (2012). We will assume the following condition for the kernel K which is fairly standard in the non-parametric literature.

VC. The kernel $K : \mathbb{R}^d \to \mathbb{R}$ has bounded support and integrates to 1. Let $\mathcal{F}$ be the class of functions of the form

$$z \in \mathbb{R}^d \mapsto K(x - z), \quad z \in \mathbb{R}^d.$$

Then, $\mathcal{F}$ is a uniformly bounded VC class: there exist positive constants A and v such that

$$\sup_P \mathcal{N}(\mathcal{F}, L^2(P), \epsilon \|F\|_{L^2(P)}) \leq (A/\epsilon)^v,$$

where $\mathcal{N}(T, d, \epsilon)$ denotes the ϵ-covering number of the metric space (T, d), F is the envelope function of $\mathcal{F}$ and the sup is taken over the set of all probability measures on $\mathbb{R}^d$. The constants A and v are called the VC characteristics of the kernel.

The assumption VC holds for a large class of kernels, including any compact supported polynomial kernel and the Gaussian kernel. See Nolan and Pollard (1987) and Giné and Guillou (2002).

Proposition 7 (Sriperumbudur and Steinwart (2012))**.** *Let P be the probability measure on $\mathbb{R}^d$ with Lebesgue density bounded by $\|p\|_\infty$ and assume that the kernel K belongs to $L^\infty(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ satisfies the VC assumption. Then for any $\gamma > 0$ and $h > 0$, there exists an absolute constant C depending on the VC characteristic of K such that, with probability no smaller than $1 - e^{-\gamma}$,*

$$\|p_h - \widehat{p}_h\|_\infty \leq \frac{C}{nh^d}\left(\gamma + v \log \frac{2A}{\sqrt{h^d \|p\|_\infty \|K\|_2^2}}\right) + C\sqrt{\frac{2\|p\|_\infty}{nh^d}}\left(\gamma \|K\|_\infty^2 + v\|K\|_2^2 \log \frac{2A}{\sqrt{h^d \|p\|_\infty \|K\|_2^2}}\right)$$

Lemma 4. *Suppose A' and A' are two clusters of $T_p(\lambda)$ and they are δ-separated. Then their merge height $m_f(A, A')$ satisfies*

$$m_f(A, A') \leq \lambda - \delta.$$

Proof. By Definition 2, there A and A' belong distinct connected components of $\{p > \lambda - \delta\}$. For the sake of contradiction, suppose

$$m_f(A, A') > \lambda - \delta.$$

Therefore there exists λ' such that $m_f(A, A') > \lambda' > \lambda - \delta$ and that by Definition 9 A and A' belong the same connected component of $\{p \geq \lambda'\}$. This is a contradiction because $\{p \geq \lambda'\} \subset \{p > \lambda - \delta\}$. □

B.1. Proofs in Section 4.3

Proof of proposition 1. To show proposition 1, we begin by introducing a standard topology lemma.

Lemma 5. *Suppose $p : \mathbb{R}^d \to \mathbb{R}$ are compactly supported. If A and A' are in the same connected components of $\{p \geq \lambda_i\}$ for $i = 1, 2, \ldots, \infty$ and that $\lambda_i \leq \lambda_{i+1}$, then A and A' are in the same connected components of $\{p \geq \lambda_0\}$, where $\lambda_0 = \sup_i \lambda_i$.*

Proof of lemma 5. Let $\mathcal{C}_i$ be the connected component of $\{p \geq \lambda_i\}$ that contains A and A'. Thus $\mathcal{C}_i$ are compact and connected. Since $\mathcal{C}_{i+1} \subset \mathcal{C}_i$ for all $i \geq 1$, $\bigcap_{i=1}^{\infty} \mathcal{C}_i$ is connected. Thus $A, A' \subset \bigcap_i \mathcal{C}_i \subset \{p \geq \lambda_0\}$. □

Consider

$$\lambda^* = \sup\{\lambda : A, A' \text{ belongs to the same connected components of } \{p \geq \lambda\}\}$$

Then $\lambda_2 \leq \lambda^* \leq \lambda_1$. By lemma 5 , A and A' are in the same connected components of $\{p \geq \lambda^*\}$. Thus $\lambda_2 \leq \lambda^* < \lambda_1$.
In order to show that λ^* is split level, it suffices to show that A and A' are in the different connected components of $\{p > \lambda^*\}$. Suppose for the sake of contradiction that A and A' are connected in $\{p > \lambda^*\}$. Then A and A' are path connected as $\{p > \lambda^*\}$ is open. Thus there exist $\mathcal{P}$ connects A and A' in $\{p > \lambda^*\}$. Since $\mathcal{P}$ is compact, $p(\mathcal{P}) > \lambda^*$ implies that there exists $a > 0$ such that $\lambda^* + a < \lambda_1$ and $\mathcal{P}, A, A' \subset \{p \geq \lambda^* + a\}$. Thus A and A' belong to the same connected component of $\{p \geq \lambda^* + a\}$. This is a contradiction because by construction of λ^*, A and A' belongs to the different connected components of $\{p \geq \lambda^* + a\}$. □

Proof of corollary 1. Suppose A and A' are δ separated with respect to λ. Then A and A' belongs to distinct connected components of $\{p > \lambda - \delta\}$ where $\lambda = \inf_{x \in A \cup A'} f(x)$. Let $0 < \epsilon \leq \delta$ be given. Then since $\{p \geq \lambda - \delta + \epsilon\} \subset \{p > \lambda - \delta\}$, A and A' belongs to distinct connected components of $\{p \geq \lambda - \delta + \epsilon\}$.
Since $\mathbb{R}^d = \{p \geq 0\}$ is connected, A and A' belongs to the same connected component of $\{p \geq 0\}$. By proposition 1, there exists $0 \leq \lambda^* < \lambda - \delta + \epsilon$ such that A and A' in the same connected component of $\{p \geq \lambda^*\}$ and in different connected components of $\{p > \lambda^*\}$. By taking $\epsilon \to 0$, the claimed result follows.

□

B.2. Proofs in sections 4.4

Proof of Lemma 2. From the proof of Lemma 6, it can be see that

$$\left\{p \geq \lambda + C\frac{\log(n)}{n^{\alpha/(2\alpha+d)}}\right\} \cap \{X_i\}_{i=1}^n \subset \widehat{D}(\lambda) \subset \left\{p \geq \lambda - C\frac{\log(n)}{n^{\alpha/(2\alpha+d)}}\right\}.$$

Thus for any $y \in B(X_j, h)$ for some $X_j \in \hat{D}(\lambda)$,

$$p(y) \geq p(X_j) - Lh^\alpha \geq \lambda - C\frac{\log(n)}{n^{\alpha/(2\alpha+d)}} - Lh^\alpha,$$

where the first inequality follows from $|p(y) - p(X_j)| \leq Lh^\alpha$. Therefore the above display implies

$$\bigcup_{X_j \in \hat{D}(\lambda)} B(X_j, h) \subset \left\{p \geq \lambda - C\frac{\log(n)}{n^{\alpha/(2\alpha+d)}} - Lh^\alpha\right\}.$$

For the other inclusion, let $x \in \left\{p \geq \lambda + C\frac{\log(n)}{n^{\alpha/(2\alpha+d)}} + Lh^\alpha\right\}$. Then $\hat{p}_h(x) \geq \lambda + Lh^\alpha$. Thus $B(x, h) \cap \{X_i\}_{i=1}^n \neq \emptyset$, or else $\hat{p}_h(x) = 0$. Let $X_j \in B(x, h)$. Therefore

$$p(X_j) \geq p(x) - Lh^\alpha \geq \lambda + \frac{\log(n)}{n^{\alpha/(2\alpha+d)}}.$$

Thus $\widehat{p}_h(X_j) \geq \lambda$, which means that $X_j \in \widehat{D}(\lambda)$. So $x \in \bigcup_{X_j \in \hat{D}(\lambda)} B(X_j, h)$ and the first inclusion follows.

□

Proof of Theorem 1. Let $\mathcal{B}$ be the event that

$$\mathcal{B} = \{\sup_{x \in \mathbb{R}^d} |\widehat{p}_h(x) - p(x)| \leq a_n\}$$

By Proposition 7, we can choose a_n so that $P(\mathcal{B}) \geq 1 - 1/n$ and that $a_n = O\left(\left(\frac{\log(n)}{n}\right)^{\alpha/(d+2\alpha)}\right)$.
All the argument will be made on the good event $\mathcal{B}$.

Observe that p has connected support. Therefore $\lambda = 0$ is not a split level. Assume that $\lambda_0 = \min\{\lambda^* : \lambda^* \text{ is a split level of } p\}$. Then $\lambda_0 > 0$. If $h = O(n^{-1/(2\alpha+d)})$, for large n, we have $2a_n + (4h/c_S)^\alpha < \min\{\delta_S, \delta_0\}$. Take

$$\delta \geq 2a_n + (4h/c_S)^\alpha / c_S.$$

Let A and A' are two sets being δ-separated and let $\lambda = \inf_{x \in A \cup A'} p(x)$. Since A and A' are in distinct connected components of $\{p > \lambda - \delta\}$, by proposition 1 there exists λ^* being a split level of p such that $\lambda^* \leq \lambda - \delta$ and that A and A' belongs to distinct connected components of $\{p > \lambda^*\}$. Thus A and A' belong to distinct connected components of $\{p > \lambda'\}$, where

$$\lambda' = \lambda^* + 2a_n + (4h/c_S)^\alpha / c_S.$$

Let $\{\mathcal{C}_k\}_{k=1}^K$ be the collection of connected components of $\{p > \lambda'\}$. Thus we have $A \subset \mathcal{C}_k$ and $A' \subset \mathcal{C}_{k'}$ for some $k \neq k'$. In order to show the smallest cluster containing $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ are disjoint with high probability, it suffices to show the following statement.

• Let A and A' be two connected subsets of $\{p > \lambda'\}$ and belong to two distinct connected components of $\{p > \lambda^*\}$. Then the smallest cluster containing $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ are disjoint with high probability.

Note that this observation reduce the original statement which concerns with generic δ-separated sets to the current statement which only concerns with one level near the split

level. Since there are finitely many split levels, a simple union bound will suffice to show the δ consistency of the cluster tree returned by Algorithm 2.

The proof will be completed by the following two claims.

Claim 1. If A is a connected subset of $\{p > \lambda'\}$, then $A \cap \{X_i\}_{i=1}^n$ is in the same connected component of

$$\widehat{L}(\lambda' - a_n) := \bigcup_{\{X_j : \widehat{D}(\lambda' - a_n)\}} B(X_j, 2h). \tag{29}$$

Proof. It suffices to show that

$$\{p > \lambda'\} \subset \widehat{L}(\lambda' - a_n). \tag{30}$$

Since for large n,

$$a_n + (4h/c_S)^\alpha \le \delta_0,$$

By **C2** there exists $\mathcal{N}_h \subset \{p > \lambda'\}$ with $card(\mathcal{N}_h) \le A_c(h)^{-d}$ such that $\mathcal{N}_h$ is a h cover. Since $\{p > \lambda'\}$ satisfies the inner cone condition **C1**,

$$P(B(x,h) \cap \{p > \lambda'\}) \ge \lambda^* c_I V_d h^d \ge \lambda_0 c_I V_d h^d.$$

So there exists c_I' only depending on d and c_I such that

$$P(\{\{X_i\}_{i=1}^n \cap B(x,h) \cap \{p > \lambda'\} = \emptyset\}) \le (1 - \lambda_0 c_I V_d h^d)^n \le \exp(-c_I' \lambda_0 n^{2\alpha/(\alpha+d)}) = o(n^{-2}),$$

where the second inequality follows from $h = O(n^{1/(2\alpha+d)})$ and the equality follows from $\lambda_0 n^{2\alpha/(2\alpha+d)}/\log(n) \to \infty$ and n being large enough. Consider the event

$$\mathcal{A} = \{\{X_i\}_{i=1}^n \cap B(x,h) \cap \{p > \lambda'\} \ne \emptyset \text{ for all } x \in \mathcal{N}_h\}.$$

By the union bound

$$P(\mathcal{A}^c) \le card(\mathcal{N}_h) \exp(-c_I' \lambda_0 n^{2\alpha/(2\alpha+d)}) = A_c h^{-d} \exp(-c_I' \lambda_0 n^{2\alpha/(2\alpha+d)}) = o(1). \tag{31}$$

So for any $y \in \{p > \lambda'\}$, there exists $x \in \mathcal{N}_h$ such that $|y - x| \le h$. Under event $\mathcal{A}$ there exists $X_j \in \{p > \lambda'\}$ such that $|X_j - x| \le h$. Therefore $y \in B(X_j, 2h)$. Since

$$X_j \in \{X_i\}_{i=1}^n \cap \{p > \lambda'\} \subset \widehat{D}(\lambda' - a_n),$$

the claim follows. □

To finish the proof of the theorem, we still need to show at level $\lambda' - a_n$ the data points $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ are contained in distinct clusters. Therefore the following claim finish the proof.

Claim 2. There exists a partition $\{S_i\}_{i=1}^I$ of $\widehat{D}(\lambda' - a_n)$ such that $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$ belong to distinct subsets of the partition and that data points in distinct subsets of the partition are mutually disconnected.

Proof. Let $\{B_i\}_{i=1}^I$ be the collection of connected components of $\{p \ge (4h/c_S)^\alpha + \lambda^*\}$. Since A and A' belong to distinct connected components of $\{p > \lambda^*\}$, and $\lambda^* < (4h/c_S)^\alpha + \lambda^*$, A and A' are contained in distinct elements of $\{B_i\}_{i=1}^I$. From condition **S** ,

$$\min_{i \ne j} d(B_i, B_j) \ge 4h. \tag{32}$$

Note that $\widehat{D}(\lambda' - a_n) \subset \{p \ge (4h/c_S)^\alpha + \lambda^*\}$ as a consequence of event $\mathcal{B}$. Thus $S_i = B_i \cap \widehat{D}(\lambda' - a_n)$ form a partition of $\widehat{D}(\lambda' - a_n)$. Let

$$L_i = \bigcup_{X_j \in S_i} B(X_j, 2h).$$

By (32), $L_i \cap L_j = \emptyset$ if $i \ne j$. This shows that data points in distinct subsets of the partition $\{S_i\}_{i=1}^I$ are mutually disconnected at the graph $\mathbb{C}(h, \lambda' - a_n)$. □

□

Proof of Proposition 2.
Step 1. In this step we show that condition **S(2)** holds. Consider an arbitrary split level λ, and two connected components C_1, C_2. If

$$\inf_{\delta > 0} d(C_1 \cap \{p \ge \lambda + \delta\}, C_2 \cap \{p \ge \lambda + \delta\}) > 0$$

then we have $d(C_1 \cap \{p \ge \lambda\}, C_2 \cap \{p \ge \lambda\}) > 0$, and the thesis is trivial. Thus assume that

$$\inf_{\delta > 0} d(C_1 \cap \{p \ge \lambda + \delta\}, C_2 \cap \{p \ge \lambda + \delta\}) = 0,$$

i.e.

$$\lim_{\delta \to 0} d(C_1 \cap \{p \ge \lambda + \delta\}, C_2 \cap \{p \ge \lambda + \delta\}) = 0.$$

Thus there exists $y_0 \in \{p = \lambda\}$, and points $y_{1,2}^\delta \in C_{1,2} \cap \{p \ge \lambda + \delta\}$ such that

$$y_{1,2}^\delta \overset{\delta \to 0}{\to} y_0, \qquad |y_1^\delta - y_2^\delta| = d(C_1 \cap \{p \ge \lambda + \delta\}, C_2 \cap \{p \ge \lambda + \delta\}).$$

It is straightforward to check that $p(y_1^\delta) = p(y_2^\delta) = \lambda + \delta$.

The thesis is now rewritten as $|y_1^\delta - y_2^\delta| \ge c_S \delta^{1/2}$ for some constant $c_S > 0$ and all sufficiently small δ. Since split levels are also critical, $\nabla p(y_0) = 0$; since p is a Morse function, $\nabla^2 p(y_0)$ is non-degenerate. By Taylor formula we have

$$\delta = p(y_j^\delta) - p(y_0) = (y_j^\delta - y_0)^T \nabla^2 p(y_0)(y_j^\delta - y_0)/2 + O(|y_j^\delta - y_0|^3), \qquad j = 1, 2, \tag{33}$$

and, as $\nabla^2 p(y_0)$ is non-degenerate, it follows $|y_j^\delta - y_0| = O(\delta^{1/2})$, i.e. there exist constants $c_1, c_2, \delta_0 > 0$ such that

$$c_1 \delta^{1/2} \le |y_j^\delta - y_0| \le c_2 \delta^{1/2} \qquad \text{for all } \delta \in (0, \delta_0).$$

We can estimate c_2 from below: denoting by

$$a := \max\{|e_1(y_0)|, |e_2(y_0)|\}, \qquad e_1(y_0), e_2(y_0) = \text{eigenvalues of } \nabla^2 p(y_0),$$

(33) gives

$$(y_j^\delta - y_0)^T \nabla^2 p(y_0)(y_j^\delta - y_0) \le a c_2^2 |y_j^\delta - y_0|^2,$$

hence $c_2 \ge \sqrt{2/a}$. By the Lipschitz regularity of the gradient, i.e. hypothesis

$$|\nabla p(x) - \nabla p(y)| \le L|x-y|$$

for some $L > 0$, we have

$$|\nabla p(y_1^\delta) - \nabla p(y_0)| = |\nabla p(y_1^\delta)| \le L|y_1^\delta - y_0| \le L c_2 \delta^{1/2}. \tag{34}$$

Consider now the segment $[y_1^\delta, y_2^\delta]$ between y_1^δ and y_2^δ: since $y_j^\delta \in C_j \cap \{p \ge \lambda + \delta\}$ $(j = 1, 2)$, and $C_j \cap \{p \ge \lambda + \delta\}$ are disconnected for all $\delta > 0$, there exists some point $z \in [y_1^\delta, y_2^\delta]$ such that $p(z) < \lambda + \delta/2$. By Taylor's formula we then have

$$p(z) = p(y_1^\delta) + \nabla p(y_1^\delta) \cdot (z - y_1^\delta) + (z - y_1^\delta)^T \nabla^2 p(y_1^\delta)(z - y_1^\delta)/2 + O(|z - y_1^\delta|^3)$$

If inequality $|y_1^\delta - y_2^\delta| \le k\delta^{1/2}$ were to holds for some $k > 0$, then since the domain is compact and $\nabla^2 p \in C^2$, denoting by

$$A := \sup_x \Big(\max\{|e_1(x)|, |e_2(x)|\} \Big), \qquad e_1(x), e_2(x) = \text{eigenvalues of } \nabla^2 p(x),$$

we have

$$\begin{aligned} |p(z) - p(y_1^\delta)| &\le |\nabla p(y_1^\delta)| \cdot |z - y_1^\delta| + |\nabla^2 p(y_1^\delta)| \cdot |z - y_1^\delta|^2/2 \\ &\le |\nabla p(y_1^\delta)| \cdot |y_1^\delta - y_2^\delta| + |\nabla^2 p(y_1^\delta)| \cdot |y_1^\delta - y_2^\delta|^2/2 \overset{(34)}{\le} (Lkc_2 + k^2 A/2)\delta. \end{aligned}$$

Since $p(y_1^\delta) = \lambda + \delta$, and $p(z) < \lambda + \delta/2$, we need $Lkc_2 + k^2 A/2 > 1/2$, hence

$$k \ge A^{-1}(\sqrt{L^2 c_2^2 + A} - Lc2),$$

i.e.

$$\begin{aligned} |y_1^\delta - y_2^\delta| &= d(C_1 \cap \{p \ge \lambda + \delta\}, C_2 \cap \{p \ge \lambda + \delta\}) \\ &\ge A^{-1}(\sqrt{L^2 c_2^2 + A} - Lc_2)\delta^{1/2} \ge A^{-1}(\sqrt{2L^2/a + A} - L\sqrt{2/a})\delta^{1/2}. \end{aligned}$$

Step 2. In this step we show that condition **C** holds.

Proof of C1. Since a Morse function has only isolated non degenerate critical points, and an isolated set in a compact domain is also finite, we infer that $\nabla p(x) = 0$ only for finitely many x. In particular, since λ^* are split levels, and $\{p = \lambda^*\}$ contains a critical point, there exist sufficiently small $\delta_1, \delta_2 > 0$ such that $\{\lambda^* + \delta_1 \le p \le \lambda^* + \delta_2\}$ contains no critical points (since there are only finitely many critical points). Since the level sets are orthogonal to the gradient, we infer that $\{p = \lambda^* + \delta_1\}$ is smooth. In particular, $\{p = \lambda^* + \delta_1\}$ it satisfies the inner cone property with $c_I = 1/2$.

The key difficulty in extending the above argument to $\{p > \lambda^*\}$ (instead of just $\{p \ge \lambda^* + \delta_1\}$ with $\delta_1 > 0$) is that the norm of gradient $|\nabla p|$ can approach zero as $\delta_1 \to 0$, since $\{p = \lambda^*\}$ is a split level, hence it contains critical points.

The Morse function requirement, however, gives the "bare minimum" regularity to ensure C1. We aim to prove, by contradiction, that $\{p \ge \lambda^*\}$ also satisfies C1, i.e. the boundary $\{p = \lambda^*\}$ does not exhibit cusps. If a cusp were to appear, then there exist arc-length parameterized curves $\gamma_j : [0, \epsilon] \longrightarrow \Omega$, $j = 1, 2$, such that $x_0 = \gamma_1(0) = \gamma_2(0)$ and the angle $\angle \gamma_1(s) x_0 \gamma_2(s) \to 0$ as $s \to 0$.

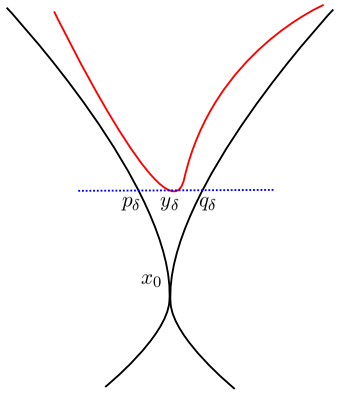


Figure 3: Construction in the proof of Morse function case.

Consider a level set $\{p = \lambda^* + \delta\}$, for small $\delta > 0$. Let $y_\delta \in \{p = \lambda^* + \delta\}$ be the point on $\{p = \lambda^* + \delta\}$ closest to x_0, i.e. $|x_0 - y_\delta| = \min_{y \in \{p = \lambda^* + \delta\}} |x_0 - y|$, and we proved that $|x_0 - y_\delta| = O(\sqrt{\delta})$. Let p_δ, q_δ be the intersection between $\{p = \lambda^*\}$ and the tangent line to $\{p = \lambda^* + \delta\}$ through y_δ. Clearly, as $\{p = \lambda^*\}$ has a cusp at x_0, we get $\lim_{\delta \to 0} \angle p_\delta x_0 q_\delta = 0$. Thus

$$d(y_\delta, \{p = \lambda^*\}) \le |p_\delta - y_\delta| = o(\sqrt{\delta}).$$

Since $\nabla p(x_0) = 0$, and the gradient ∇p is L-Lipschitz continuous for some constant L, we infer $|\nabla p(y)| \le A\sqrt{\delta}$ for some $A > 0$ and all y on the segment $[p_\delta, y_\delta]$. Thus it follows

$$\delta = |p(p_\delta) - p(y_\delta)| \le AL\sqrt{\delta}|p_\delta - y_\delta| = o(\delta).$$

This is a contradiction.

Proof of C2. Let $U = \{p \ge \lambda^* + \delta\}$. Fix an arbitrary r. Clearly $U \subseteq \bigcup_{x \in U} B(x, r/3)$. Since $U = \{p \ge \lambda^* + \delta\}$ is closed, and the domain Ω is compact, we infer $U = \{p \ge \lambda^* + \delta\}$ is also compact. Thus we can extract a covering $U \subseteq \bigcup_{i=1}^{C_r} B(x_i, r/3)$ with finitely many balls. By Vitali covering lemma, we can further extract mutually disjoint balls $B(x_{i_j}, r/3)$

such that

$$U \subseteq \bigcup_{j=1}^{C'_r} B(x_{i_j}, r)$$

Since $B(x_{i_j}, r/3) \subset \Omega_{r/3}$, and $\{B(x_{i_j}, r/3)\}_{j=1}^{C'_r}$ are pairwise disjoint, we have

$$V_d C'_r (r/3)^{-d} \leq \mathcal{L}^d(\Omega_{r/3}).$$

Thus we can choose $\mathcal{N}_r = \{x_{i_j}\}$, $j = 1, \cdots, C'_r$.

□

B.2.1. Proofs in Section 4.5

Proof of lemma 2. Let $\lambda > 0$ be given. Later in the proof, it can be seen that $\lambda = C_0$, being the common upper bound of $f_i \in F$. Define $a > 0$ to be such that

$$56\lambda \cdot 8^{d-1} a^d = 1. \tag{35}$$

Consider

$$f(x) = \begin{cases} \lambda, & x \in [0, 56a] \times [0, 8a]^{d-1} = \Omega \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

Let $b = \left(\frac{\log(32)}{n\lambda V_d}\right)^{1/d}$. For $0 < \alpha < 1$, define

$$g(r) = \begin{cases} 0, & 0 \leq r \leq b \\ \mathcal{K}\left(\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} - |r - b - \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}|\right)^{\alpha}, & |r - b - \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}| \leq \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} \\ 0, & \text{otherwise,} \end{cases}$$

and for $\alpha \geq 1$,define

$$g(r) = \begin{cases} 0, & 0 \leq r \leq b \\ 2^{1-\alpha}\delta - \mathcal{K}|r - b - \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}|^{\alpha}, & 0 \leq |r - b - \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}| \leq \frac{1}{2}\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} \\ \left(\delta^{1/\alpha} - \mathcal{K}^{1/\alpha}|r - b - \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}|\right)^{\alpha}, & \frac{1}{2}\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} \leq |r - b - \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}| \leq \left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{K} \leq 1$ is chosen so that $g \in \Sigma(L, \alpha)$. By construction $0 \leq g(r) \leq \delta$.
Consider the inequality

$$\begin{aligned} b + 2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} &= \left(\frac{\log(32)}{n\lambda V_d}\right)^{1/d} + 2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} \\ &\leq \left(\frac{1}{4^d 8\lambda}\right)^{1/d} + 2\left(\frac{1}{16(7\lambda)^{1/d}}\right) \\ &\leq \frac{1}{4}\left(\frac{1}{7}\right)^{1/d}\lambda^{-1/d} + \frac{1}{8}\left(\frac{1}{7}\right)^{1/d}\lambda^{-1/d} = 3a \end{aligned} \tag{37}$$

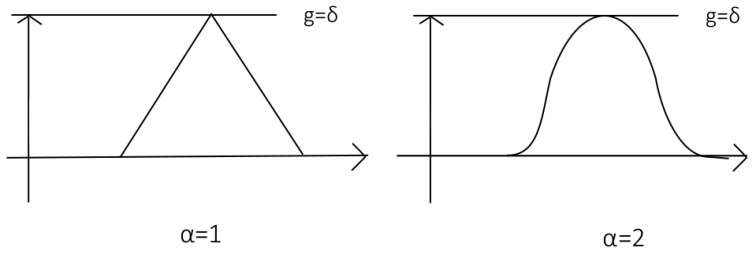


Figure 4: The radial function g for $\alpha = 1, 2$.

where the first inequality follows from $n \geq 4^d \frac{8\log(32)}{V_d}$ and $\delta \leq \left(\frac{\mathcal{K}}{16^{\alpha}(7\lambda)^{\alpha/d}}\right)$. So by construction, 9 disjoint ball $\left\{B\left(x_i, b + 2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}\right)\right\}_{i=0}^{8}$ can be placed in Ω. For $i = 1, \ldots, 8$, let $f_i = f(x) - g(|x - x_i|) + g(|x - x_0|)$. Thus $f_i \in \Sigma(L, \alpha)$ for $, i = 1, \ldots, 8$ and the common upper bound of f_i is $\|f_i\|_\infty = \lambda$. Since $\int f = 1$, by symmetry each f_i also integrates to 1. The fact that $f_i \geq 0$ follows from $0 \leq g(r) \leq \delta \leq p_{\max}$. Since for any $1 \leq i, j \leq 8$, $f_j(x) = \lambda$ for any $x \in B(x_i, b)$.

$$\begin{aligned} P(\text{There exists a point in } B(x_i, b) \text{ for any } i) &\geq 1 - (1 - \lambda V_d b^d)^n \\ &\geq 1 - \exp(-V_d b^d \lambda n) = 1 - 1/32 \end{aligned}$$

where $b = \left(\frac{\log(32)}{n\lambda V_d}\right)^{1/d}$ is used in the last equality. Thus

$$P(\text{There exists a point in every } \ B(x_i, b) \text{ for } \ i = 1 \ldots 8) \geq 3/4.$$

Suppose the family $F = \{f_i\}_{i=1}^{8}$ is given ahead. One wants to show that any algorithm being δ consistent with probability 3/4 can identify f_i with probability at least 1/2. To begin consider $B_i = \{f_i \geq \lambda\}$. B_i has exactly two connected components and one is $B(x_i, b)$. Denote the other connected component of B_i by V_i. Thus $B_i = V_i \cup B(x_i, b)$, where $V_i \cap B(x_i, b) = \emptyset$. Define the three events $\mathcal{E}_1$, $\mathcal{E}_1$ and $\mathcal{E}_3$ as following

$$\begin{aligned} \mathcal{E}_1 &= \{\text{There exists a point in every } \ B(x_i, b) \text{ for } \ i = 1 \ldots 8\} \\ \mathcal{E}_2 &= \{\text{The algorithm is } (\delta, \epsilon) \text{ consistent }\} \\ \mathcal{E}_3 &= \{\text{The algorithm can indentify the true density}\} \end{aligned} \tag{38}$$

Then one has $\mathcal{E}_1 \cap \mathcal{E}_2 \subset \mathcal{E}_3$. This is because if an algorithm is δ, consistent and every $B(x_i, b)$ contains at least one point , the algorithm will assign points in $\cup_{j\neq i} B(x_j, b)$ and points $B(x_i, b)$ into different clusters before joining them into the same cluster. In this way, the algorithm can identify the true density. Since $P(\mathcal{E}_1) \geq 3/4$ and $P(\mathcal{E}_2) \geq 3/4$, $P(\mathcal{E}_3) \geq 1/2$

It remains to compute the KL divergent between f_1 and f_2 and apply Fano's lemma. Using spherical coordinate centering at x_1 and x_2, the KL divergent is given by

$$\begin{aligned}
\mathrm{KL}(f_1, f_2) &= dV_d \int_b^{b+2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}} (\lambda) \log\left(\frac{\lambda}{\lambda - g(r)}\right) r^{d-1} + (\lambda - g(r)) \log\left(\frac{\lambda - g(r)}{\lambda}\right) r^{d-1} dr \\
&= dV_d \int_b^{b+2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}} g(r) \log\left(\frac{\lambda}{\lambda - g(r)}\right) r^{d-1} dr \\
&= dV_d \int_b^{b+2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}} g(r) \log\left(1 + \frac{g(r)}{\lambda - g(r)}\right) r^{d-1} dr \\
&\leq dV_d \int_b^{b+2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}} g(r) \frac{g(r)}{\lambda - g(r)} r^{d-1} dr \leq dV_d \int_b^{b+2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}} g(r) \frac{g(r)}{\lambda} r^{d-1} dr \\
&\leq d\lambda^{-1}\delta^2 V_d \int_b^{b+2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}} r^{d-1} dr \\
&\leq \frac{d\delta^2 V_d}{\lambda d} \left(b + 2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}\right)^d
\end{aligned}$$

Thus by Fano's lemma

$$n \geq \frac{(1/2)\log_2(8) - 1}{\mathrm{KL}(f_1, f_2)} = \frac{1}{2\mathrm{KL}(f_1, f_2)} \tag{39}$$

and this implies

$$\left(\frac{\lambda}{2\delta^2 V_d n}\right)^{1/d} \leq 2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} + \left(\frac{\log(32)}{n\lambda V_d}\right)^{1/d}. \tag{40}$$

Since $\delta \leq \lambda/(2^{d/2+1})$, this gives

$$\frac{\lambda}{2^{d+1}\delta^2 V_d n} \geq \frac{\log(32)}{n\lambda V_d}. \tag{41}$$

Combines equation (40) and equation (41) one has

$$2\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} \geq \left(\frac{\lambda}{2\delta^2 V_d n}\right)^{1/d} (1 - \frac{1}{2}) \tag{42}$$

This gives

$$n \geq \frac{\lambda \mathcal{K}^{d/\alpha}}{C(d)\delta^{2+d/\alpha}}, \tag{43}$$

where $C(d) = 2^{2d+1}V_d$.

To justify that the collection of functions $\{f_i\}_{i=1}^8$ constructed in the previous proof satisfies condition **C** and $\mathbf{S}(\alpha)$, observe that λ is the only split level of f_i for all $1 \leq i \leq 8$. The case of $\alpha > 1$ is only provided as the case of $\alpha < 1$ is simpler. Straight forward computations shows that for any $t \leq 2^{-\alpha}$, $\{x : f_i(x) \geq \lambda + t\}$ has two connected components: $B\left(x_i, b_0 + \frac{1}{2}\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha} - \left(\frac{t}{\mathcal{K}}\right)^{1/\alpha}\right)$ and $\left(B\left(x_i, b_0 + \frac{1}{2}\left(\frac{\delta}{\mathcal{K}}\right)^{1/\alpha}\right)\right)^c \cap \Omega$. Therefore condition **C** and $\mathbf{S}(\alpha)$ are trivially satisfied. □

B.3. Proofs in Section 4.6

Poof proposition 3. By (9), with probability at least γ, we have

$$\|\widehat{p}_h - p\|_\infty \leq a_n.$$

Step 1. In this step, we show that for any split level λ^* satisfying (18), there exists $\widehat{\lambda^*}$ being Δ-significant and that

$$|\widehat{\lambda^*} - \lambda^*| \leq \Delta$$

for large n. Let $\mathcal{C}$ and $\mathcal{C}'$ be two sets split at λ^*. Thus there exists $\mathcal{B}$ being the connected component of $\{p \geq \lambda^*\}$ containing both $\mathcal{C}$ and $\mathcal{C}'$.

By (18), for large n, neither $\{X_i\}_{i=1}^n \cap \mathcal{C} \cap \{p \geq \lambda^* + 2\Delta\}$ nor $\{X_i\}_{i=1}^n \cap \mathcal{C}' \cap \{p \geq \lambda^* + 2\Delta\}$ is empty. Let $X_i \in \mathcal{C} \cap \{p \geq \lambda^* + 2\Delta\}$ and $X_j \in \mathcal{C}' \cap \{p \geq \lambda^* + 2\Delta\}$.

• By the same argument that gives (30)

$$\{p \geq \lambda^*\} \subset \widehat{L}(\lambda^* - a_n) := \bigcup_{\{X_j : \widehat{D}(\lambda^* - a_n)\}} B(X_j, 2h). \tag{44}$$

Since $\mathcal{B} \subset \{p \geq \lambda^*\}$ and that $\mathcal{B}$ is connected, X_i and X_j have the same label in $\mathbb{C}(h, \lambda^* - a_n)$.

• Since $X_i \in \mathcal{C}, X_j \in \mathcal{C}'$ and that $\mathcal{C}$ and $\mathcal{C}'$ are split exactly at λ^*, X_i, X_j are contained in the distinct connected components of $\{p \geq \lambda^* + \Delta\}$. By **Claim 2** in the proof of Theorem 1, X_i and X_j belong to distinct connected components of $\mathbb{C}(h, \lambda^* + \Delta - a_n)$. Let $\widehat{\lambda^*}$ be defined as in (17). By the above two bullet points,

$$\lambda^* - a_n \leq \widehat{\lambda^*} \leq \lambda^* + \Delta - a_n.$$

The fact that $\widehat{\lambda^*}$ is Δ-significant follows from the observation that

$$X_i, X_j \in \mathbb{C}(h, \lambda^* + 2\Delta - a_n).$$

Step 2. In this step, we show that if $\widehat{\lambda^*}$ is a Δ-significant level of the cluster tree constructed using modified DBSCAN, then there exists λ^* being a split level of p such that

$$|\widehat{\lambda^*} - \lambda^*| \leq \Delta.$$

So suppose X_i ,X_j and $\widehat{\lambda^*}$ satisfies (17) and that $X_i, X_j \in \mathbb{C}(h, \widehat{\lambda^*} + \Delta)$. Let

$$\lambda^* := \sup\{\lambda \geq 0 : X_i \text{ and } X_j \text{ are in the same connected component of } \{p \geq \lambda\}\}.$$

• By (44), X_i and X_j have the same label in $\mathbb{C}(h, \lambda^* - a_n)$. Therefore,

$$\lambda^* - a_n \leq \widehat{\lambda^*}.$$

• For the sake of contradiction, suppose that

$$\widehat{\lambda^*} > \lambda^* + \Delta.$$

Then by **Claim 2** in the proof of Theorem 1, X_i and X_j belong to distinct connected components of $\mathbb{C}(h, \lambda^* + \Delta - a_n)$. By definition of $\widehat{\lambda^*}$, this implies

$$\lambda^* + \Delta - a_n \geq \widehat{\lambda^*},$$

which is a contradiction. This finishes the proof.

□

B.4. A Side result: consistency of the KDE tree

As a side result, we also compute the upper bound of cluster tree estimators generated by kernel density estimators. We acknowledge that KDE clustering algorithms have been studied by many authors including Rinaldo and Wasserman (2010), Rigollet and Vert (2009) and Kim et al. (2016). For completeness, we provide δ-consistency results for KDE cluster tree returned by Algorithm 3, but we do not claim the novelty of these results.

Algorithm 3 Clustering based on connected components

INPUT: i.i.d sample $\{X_i\}_{i=1}^n$, the kernel $K : \mathbb{R}^d \to \mathbb{R}$, the level λ and $h > 0$

1. Compute $\widehat{L}(\lambda) = \{x : \widehat{p}_h(x) \geq \lambda\}$.
2. Construct a graph $\mathbb{G}_{h,k}$ with nodes

$$\widehat{D}(\lambda) = \{X_i\}_{i=1}^n \cap \widehat{L}(\lambda)$$

and edges (X_i, X_j) if X_i and X_j belong to the same connected component of $\widehat{L}(\lambda)$.

3. Compute $\mathbb{C}(h, \lambda)$, the graphical connected components of $\mathbb{G}_{h,\lambda}$.

OUTPUT: $\widehat{T}_n = \{\mathbb{C}(h, \lambda), \lambda \geq 0\}$.

We start by showing that for generic $\alpha > 0$, if $p \in \Sigma(L, \alpha)$, level sets of KDE estimator are good approximations of the corresponding population quantities.

Lemma 6. *Assume that $p \in \Sigma(L, \alpha)$, where $\alpha > 0$, and let K be a α-valid kernel. Then, there exist constants C_1 and C_2, depending on $\|p\|_\infty$, K, L and d such that if $h = C_1 \frac{1}{n^{1/(2\alpha+d)}}$, then with probability $1 - 1/n$, uniformly over all $\lambda > 0$,*

$$\left\{x : p(x) \geq \lambda + C_2 \left(\frac{\log(n)}{n}\right)^{\alpha/(2\alpha+d)}\right\} \subset \{x : \widehat{p}_h(x) \geq \lambda\} \subset \left\{x : p(x) \geq \lambda - C_2 \left(\frac{\log(n)}{n}\right)^{\alpha/(2\alpha+d)}\right\}. \tag{45}$$

As a direct corollary of Lemma 6, we show that algorithm 3 is consistent with the optimal rate.

Corollary 3. *Let h be chosen as in Lemma 6. Under the assumptions of Lemma 6, the cluster tree returned by Algorithm 3 is δ-consistent with probability at least $1 - 1/n$, where*

$$\delta \geq 3C_2 \left(\frac{\log n}{n}\right)^{\alpha/(2\alpha+d)}, \tag{46}$$

with $C_2 = C_2(\|p\|_\infty, K, L, d)$ a constant independent of n and δ.

We remark that Algorithm 3 is computationally infeasible even in small dimensions. This is mainly because it requires to compute the level set $\{x : \widehat{p}_h(x) \geq \lambda\}$ exactly to determine the clustering structure of the data points. However Algorithm 3 does not require additional regularity conditions such as $\mathbf{S}(\alpha)$ and $\mathbf{C}$ to attain the minimax optimal rates.

B.4.1. Proofs in Appendix B.4

Proof of lemma 6. For any $x \in \mathbb{R}^d$, with probability at least $1 - 1/n$

$$\begin{aligned} |\hat{p}_h(x) - p(x)| \leq & |\hat{p}_h(x) - p_h(x)| + |p_h(x) - p(x)| \\ \leq & C_1(K, d, \|p\|_\infty)\sqrt{\frac{\log n}{nh^d}} + C_2(K, \alpha, L)h^\alpha \end{aligned} \tag{47}$$

where the second inequality follows from proposition 7 and standard calculations for the bias. By taking

$$h = h_n = \Theta\left(\frac{\log n}{n}\right)^{1/(2\alpha+d)}$$

in (47),

$$\sup_{x \in \mathbb{R}^d} |\hat{p}_h(x) - p(x)| \leq C(\|p\|_\infty, K, L, \alpha, d)\left(\frac{\log(n)}{n}\right)^{\alpha/(2\alpha+d)}$$

This completes the proof. □

Proof of Corollary 3. Let A and A' be two given connected subsets of $\mathbb{R}^d$. Suppose $\lambda > 0$ satisfies $\lambda + 3\delta = \inf_{x \in A \cup A'} f(x)$ and that A and A' are contained in two distinct connected components of $\{p > \lambda\}$. It suffices to show that the estimate cluster tree at $\{\hat{p}_h \geq \lambda + 2\delta\}$ gives correct labels to $A \cap \{X_i\}_{i=1}^n$ and $A' \cap \{X_i\}_{i=1}^n$, where

$$h = h_n = \Theta\left(\frac{\log n}{n}\right)^{1/(2\alpha+d)}$$

- Since A, A' are connected and

$$A,\ A' \subset \{p \geq 3\delta + \lambda\} \subset \{\hat{p}_h \geq 2\delta + \lambda\},$$

A and A' each belongs to the connected component of $\{\hat{p}_h \geq 2\delta + \lambda\}$. Therefore the cluster tree at $\{\hat{p}_h \geq 2\delta + \lambda\}$ will assign $A \cap \{X_i\}_{i=1}^n$ the same label. This is also true for $A' \cap \{X_i\}_{i=1}^n$.

- It remains to show that A and A' are in the two distinct connected components of $\{\hat{p}_h \geq 2\delta + \lambda\}$. For the sake of contradiction, suppose that A and A' are in the same connected components of $\{\hat{p}_h \geq 2\delta + \lambda\}$. Since

$$\{\hat{p}_h \geq \lambda + 2\delta\} \subset \{p \geq \lambda + \delta\} \subset \{p > \lambda\},$$

A and A' are in the same connected components of $\{p > \lambda\}$. This is a contradiction.

□

Appendix C. Proofs from Section 5

C.1. Proof of Proposition 4

We first prove two simple technical lemmas, that will also be used in the proof of Proposition 6.

Lemma 7. *Suppose that $\epsilon > 2a_n$, where*

$$\sup_{x \in \mathbb{R}^d} |\hat{p}_h(x) - p_h(x)| \leq a_n,$$

and let $\lambda \in (\lambda_* + a_n, \lambda^* - a_n)$. Then,

$$S_{-h} \cap \{X_i\}_{i=1}^n \subset \hat{D}_h(\lambda) \subset S_h,$$

where $S = \{p \geq \lambda^*\}$ and

$$\hat{D}_h(\lambda) = \{x : \hat{p}_h(x) \geq \lambda\} \bigcap \{X_i\}_{i=1}^n.$$

Proof of lemma 7. For the first inclusion, suppose $X_j \in S_{-h} \cap \{X_i\}_{i=1}^n$. Then $B(X_j, h) \subset S$. Since K is supported on $B(0,1)$,

$$p_h(X_j) = \frac{1}{V_d h^d} \int_{B(X_j,h)} p(y)dy \geq \lambda^*. \tag{48}$$

As a result,

$$\hat{p}_h(X_j) \geq p_h(X_j) - a_n \geq \lambda^* - a_n \geq \lambda,$$

which implies that $X_j \in \hat{D}_h(\lambda)$. For the second inclusion, if $X_j \in \hat{D}_h(\lambda)$, then $\hat{p}_h(X_j) \geq \lambda$. So

$$p_h(X_j) \geq \hat{p}_h(X_j) - a_n \geq \lambda - a_n > \lambda_*$$

However, for any point $x \in S_h^c$, since $B(x,h) \subset S^c$, $p_h(x) \leq \lambda_*$ (see (48)). So $X_j \in \hat{D}_h(\lambda)$ implies $X_j \in S_h$. □

Lemma 8. *Under the same assumption as in Lemma 7, suppose further that $\lambda^* > a_n$. Let $\hat{L}(\lambda) = \bigcup_{X_i \in \hat{D}_h(\lambda)} B(X_i, h)$ and $\mathcal{C}$ be any connected components of S. Then $\mathcal{C}_{-2h} \subset \hat{L}(\lambda)$.*

Proof of lemma 8. Let $x \in \mathcal{C}_{-2h}$. Then, $B(x,h) \subset S$, which implies, by (48), that $p_h(x) \geq \lambda^*$ and therefore that

$$\hat{p}_h(x) \geq p_h(x) - a_n \geq \lambda^* - a_n > 0.$$

Therefore, $B(x,h) \cap \{X_i\}_{i=1}^n$ is not empty – otherwise $\hat{p}_h(x) = 0$ – so that there exists a sample point, say X_j, in $B(x,h)$. Since $B(x,h) \subset S_{-h}$, we conclude that $X_j \in S_{-h}$. By lemma 7 we then have that $X_j \in \hat{D}_h(\lambda)$. This shows that if $x \in \mathcal{C}_{-2h}$, then there exists some $X_j \in \hat{D}_h(\lambda)$ such that $x \in B(X_j, h)$. This finishes the lemma. □

Proof of Proposition 4. Let

$$a_n = C_1 \sqrt{\frac{\log(n) + \log(1/h)}{nh^d}}$$

be defined as in (7). Then by (6),

$$P\left(\sup_{x \in \mathbb{R}^d} |\hat{p}_h(x) - p_h(x)| \leq a_n\right) \geq 1 - 1/n. \tag{49}$$

Denote $h = C(\frac{\log(n)}{n\epsilon^2})^{1/d}$ where C is chosen such that $3a_n \leq \epsilon$.

Step 1. Suppose $A_{2h} \subset \mathcal{C}_i$. Then $A \subset \mathcal{C}_{i,-2h}$. Then by Lemma 8, one has $A \subset \mathcal{C}_{i,-2h} \subset \hat{L}(\lambda)$. Since data points in connected components of $\hat{L}(\lambda)$ have the same label, and A is contained in only one connected components of $\hat{L}(\lambda)$, points in $A \cap \{X_i\}_{i=1}^n$ have the same labels.

Step 2. Suppose A1 holds. Since $d(\mathcal{C}_i, \mathcal{C}_j) > 4h$, $\{\mathcal{C}_{i,2h}\}_{i=1}^I$ are pairwise disjoint. Since $\hat{D}_h(\lambda_k) \subset \bigcup_{i=1}^I \mathcal{C}_{i,h}$, this means for any i, j there is no edges connect $\hat{D}_h(\lambda_k) \cap \mathcal{C}_{i,h}$ and $\hat{D}_h(\lambda_k) \cap \mathcal{C}_{j,h}$. Since A and A' belong to distinct members of $\{\mathcal{C}_i\}_{i=1}^I$, labels in $A \cap \{X_i\}_{i=1}^n$ and in $A' \cap \{X_i\}_{i=1}^n$ are different.

□

C.2. Proof of Proposition 5

Proof of Proposition 5. Let

$$a_n = C_1 \sqrt{\frac{\log(n) + \log(1/h)}{nh^d}}$$

be defined in (9). Then by (6).

$$P\left(\sup_{x \in \mathbb{R}^d} |\hat{p}_h(x) - p_h(x)| \leq a_n\right) \geq 1 - 1/n. \tag{50}$$

Denote $h = C(\frac{\log(n)}{n\epsilon^2})^{1/d}$ where C is chosen such that $3a_n \leq \epsilon$. Denote $\lambda_k = \frac{k}{nh^d V_d}$. Therefore $\lambda^* - \lambda_* \geq 3a_n$.
Consequently λ is well defined and one has

$$\lambda_* + a_n \leq \lambda < \lambda^* - a_n.$$

By lemma 7, the nodes of $\mathbb{G}_{h,k}$ are contained in S_h. Thus

$$\bigcup_{X_j \in \mathbb{G}_{h,k}} B(X_j, h) \subset S_{2h}. \tag{51}$$

By lemma 8,

$$S_{-2h} \subset \bigcup_{X_j \in \mathbb{G}_{h,k}} B(X_j, h). \tag{52}$$

Since $h_0 \geq h$ using assumption A3 then

$$\mathcal{L}(\hat{S} \triangle S) \leq \mathcal{L}(S_{2h} \backslash S) + \mathcal{L}(S_h \backslash S_{-2h}) \leq C_0 h.$$

□

C.3. Proof of Proposition 6

We begin by constructing of a well-behaved class of sets satisfying the boundary regularity condition **(R)**. These sets will then be used to define high-density clusters in the proof of Proposition 6. Sets satisfying the properties given in the next definition are well known in the literature on support estimation: see, e.g., Korostelev and Tsybakov (1993). For completeness we also show that they satisfy the boundary regularity condition **(R)**.

Definition 13. *Denote by $\mathcal{G}_d(L)$ the class of all domains in $[0,1]^d$ satisfying*

$$\left\{(x_1, \ldots, x_d) : (x_1, \ldots, x_{d-1}) \in [0,1]^{d-1}, \quad 0 \leq x_d \leq g(x_1, \ldots, x_{d-1})\right\},$$

where $g : \mathbb{R}^{d-1} \to \mathbb{R}$ satisfies

- $1/2 \leq |g(x)| \leq 3/2$ *for all* $x \in [0,1]^{d-1}$
- $|g(x) - g(x')| \leq L|x - x'|$ *for all* $x, x' \in \mathbb{R}^{d-1}$.

Lemma 9. *There exist a constants h_0 only depending only on L such that for any $\Omega \in \mathcal{G}_d(L)$, one has for any $0 \leq h \leq h_0$,*

$$\mathcal{L}(\Omega_h \backslash \Omega_{-h}) \leq C_0 h,$$

where $\Omega_h = \bigcup_{x \in \Omega} B(x,h)$, $\Omega_{-h} = \{x \in \Omega : B(x,h) \subset \Omega\}$ and C_0 is some constant depending on d.

Proof of lemma 9. Given $\Omega \in \mathcal{G}_d(L)$, let g be the corresponding map as in definition 13. Denote $\underline{x}$ be a generic point in $\mathbb{R}^{d-1}$. Consider the change of coordinate map $\phi : \mathbb{R}^d \to \mathbb{R}^d$ defined as

$$\phi(\underline{x}, x_d) = (\underline{x},\ x_d g(\underline{x})).$$

The inverse map $\phi^{-1} : \mathbb{R}^d \to \mathbb{R}^d$ where $\phi^{-1}(\underline{x}, x_d) = (\underline{x},\ x_d/g(\underline{x}))$ is also well defined as $g > 0$.
Observe that $\phi([0,1]^d) = \Omega$, and there exists a constant $C(d)$ depending only on d such that $[0,1]^d$ satisfies condition A3 with $h_0 = 1/2$ and $C_0 = C(d)$. Thus in order to justify the lemma, it suffices to show that the maps ϕ and ϕ^{-1} only distort the distance and volume by factors depending on L only.
To be more precise, it suffices to show that for some constant L' depending on L and some absolute constant C,

$$|\phi^{-1}(x) - \phi^{-1}(x')| \leq L'|x - x'| \text{ and } |\phi(x) - \phi(x')| \leq L'|x - x'| \text{ for all } x, x' \in [-2,2]^d$$
$$\mathcal{L}(\phi^{-1}(B)) \leq C\mathcal{L}(B) \text{ and } \mathcal{L}(\phi(B)) \leq C\mathcal{L}(B) \text{ for any } B \subset [-2,2]^d\ .$$

Since the calculations of ϕ are similar to that of ϕ^{-1}, only the former one is shown in this case.

Step 1. To show that $\phi(x)$ is Lipschitz, it suffices to bound $\|\nabla\phi\|_{op}$.

$$\nabla\phi(\underline{x}, x_d) = (\frac{\partial \phi_i}{\partial x_j}) = \begin{bmatrix} 1 & 0 & \ldots & 0 & x_d \frac{\partial g(\underline{x})}{\partial x_1} \\ 0 & 1 & \ldots & 0 & x_d \frac{\partial g(\underline{x})}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & x_d \frac{\partial g(\underline{x})}{\partial x_{d-1}} \\ 0 & 0 & \ldots & 0 & g(\underline{x}) \end{bmatrix}$$

A straight forward calculations shows that for any $(\underline{x}, x^d) \in [-2,2]^d$,

$$\|\nabla\phi\|_{op} \leq 1 + x_0\|\nabla g(\underline{x})\|_2 + g(\underline{x}) \leq 5/2 + 2L.$$

Step 2. The change of variables equations gives

$$\mathcal{L}(\phi(B)) = \int_{\phi(B)} d\mathcal{L} = \int_B |\det(\nabla\phi(x))| dx.$$

Since $\det(\nabla\phi(\underline{x}, x_d)) = g(\underline{x})$ which is bounded above by $3/2$, one has $\mathcal{L}(\phi(B)) \leq (3/2)\mathcal{L}(B)$.

□

Proof of Proposition 6. Let $0 < \delta \leq 1/16$ be depending on ϵ which will be specified later. For some constant depending $C(d)$ only depending on d, it is desired to construct $\{S_i\}_{i=1}^M \in \mathcal{G}_d(C(d))$ such that $\mathcal{L}(S_i \triangle S_j) \geq \delta$ and that M is of order δ^{-d+1}.

Step 1. Consider a hyper rectangle $[0,2\delta]\times[0,\delta]^{d-2}$ in $\mathbb{R}^{d-1}$. One can place $N=\lfloor\delta^{-1}\rfloor^{d-1}/2$ such hyper rectangles into $[0,1]^{d-1}$ without having any two intersect. Denote theses hype rectangles by $\{R_i\}_{i=1}^N$. R_i is composed of two hypercube of dimension $[0,\delta]^{d-1}$, which are denoted as R_i^0 and R_i^1.

Denote $\underline{x}$ to be a generic point in $\mathbb{R}^{d-1}$. One can defined a map $g:[-\delta/2,\delta/2]^{d-1}\to\mathbb{R}$ by

$$g(\underline{x})=\begin{cases}C(d)\left(\delta/2-\|\underline{x}\|_{\mathbb{R}^{d-1}}\right), & \text{if } \|\underline{x}\|_{\mathbb{R}^{d-1}}\le\delta/2\\ 0, & \text{otherwise}.\end{cases}$$

The region

$$\mathcal{C}=\{(\underline{x},x_d):\underline{x}\in[-\delta/2,\delta/2]^{d-1},0\le x_d\le g(\underline{x})\}$$

defines a region of hyper cone in $\mathbb{R}^d$ and $C(d)$ is set so that the cone volume

$$\int_{[-\delta/2,\delta/2]^{d-1}} g(\underline{x})d\underline{x}=\delta^d.$$

Let g_i^0 and g_i^1 be the corresponding map on R_i^0 and R_i^1, as the later ones are copies of $[-\delta/2,\delta/2]^{d-1}$.

Step 2. Let $W=\{w=(w_1,\dots,w_N),w_j\in\{0,1\}\}$. By Varshamov-Gilbert lemma, there exist $w^1,\dots,w^M\in W$ such that (i) $M\ge 2^{N/8}$ (ii) $H(w^i,w^j)\ge N/8$, where H denote the hamming distance.
For $1\le j\le M$, let $G_j:[0,1]^{d-1}\to\mathbb{R}^d$ be defined as

$$G_j(\underline{x})=1/2+\sum_{i=1}^N g_i^{w_i^j}(\underline{x}).$$

Consider

$$S_j=\{(\underline{x},x_d):\underline{x}\in[0,1]^{d-1},0\le x_d\le G_j(\underline{x})\}$$

Thus by construction $G_j\in\mathcal{G}_d(C(d))$ in definition 13. For $l=0,1$ and $1\le i\le N$, define

$$\mathcal{C}_i^l=\{(\underline{x},x_d):\underline{x}\in R_i^l,0\le x_d\le g_i^l(\underline{x})\}.$$

So $\mathcal{C}_j^l$ are non-overlapping cones with volume being δ^d, which are indexical copies of $\mathcal{C}$.

Step 3. Let $\{f_j\}_{j=1}^M$ be such that

$$f_j=\begin{cases}1/4, & \text{if } x\in[0,a]^d\backslash S_j\\ 1/4+\epsilon, & \text{if } x\in S_j\\ 0, & \text{otherwise}\end{cases}$$

Since $1=\int f_j=a^d/4+(1/4+\epsilon)(3/4)^d\le a^d/4+(1/2)(3/4)^d$, a has to be greater than 1. Thus $S_j\subset[0,a]^d$ and so S_j can be viewed as the support of f_i at the gap.

Step 5. For any i and j Since f_i and f_j are only possibly different on $\{\mathcal{C}_k^0\cup\mathcal{C}_k^1\}_{k=1}^N$. Also $f_i\ne f_j$ within $\mathcal{C}_k^0\cup\mathcal{C}_k^1$ if and only if $w_k^i\ne w_k^j$. Thus the $KL(f_i,f_j)$ is determined by

$$KL(f_i,f_j)=\sum_{k=1}^N\int_{\mathcal{C}_k^0\cup\mathcal{C}_k^1}f_i\log\left(\frac{f_i}{f_j}\right)=\sum_{k:w_k^i\ne w_k^j}\int_{\mathcal{C}_k^0\cup\mathcal{C}_k^1}f_i\log\left(\frac{f_i}{f_j}\right)$$

Suppose $w_k^i\ne w_k^j$, then

$$\int_{\mathcal{C}_k^0\cup\mathcal{C}_k^1}f_i\log\left(\frac{f_i}{f_j}\right)=\int_{\mathcal{C}}(1/4+\epsilon)\log\left(\frac{1/4+\epsilon}{1/4}\right)+(1/4)\log\left(\frac{1/4}{1/4+\epsilon}\right)\le 4\delta^d\epsilon^2.$$

So $KL(f_i,f_j)\le H(w^i,w^j)\delta^d\epsilon^2\le N\delta^d\epsilon^2$.

Step 6. To apply Fano's lower bound lemma (see Tsybakov (2009)) it suffices to have

$$\max_{i\ne j}KL(P_i,P_j)\le\frac{\log M}{16n}$$

Since $M\ge 2^N/8$ it suffices to have $nN\delta^d\epsilon^2\le N\log(2)/128$. Thus it suffices to have $\delta^d=\min\{a\frac{1}{n\epsilon^2},1/16\}$ for some absolute constant a.

Step 7. By Fano's lemma, the minimax rate is bounded from below above by

$$\mathcal{L}(S_i\triangle S_j)=H(w^i,w^j)2\mathcal{L}(\mathcal{C})\ge(N/8)2\delta^d=c\delta,$$

for some absolute constant c.

□

C.4. Lower bounds of clustering at the Gap

We cite Theorem VI.1 of Chaudhuri et al. (2014) to demonstrate that the scaling in Proposition 4 is minimax optimal.

Proposition 8. *Consider a finite family of density functions $F=\{f_j\}$. Suppose all $f_j\in F$ have gap of size $\epsilon>0$ at level λ_*. This means that for any j, $\{f_j\ge\lambda_*+\epsilon\}\cup\{f_j\le\lambda_*\}=\mathbb{R}^d$. For any j, let $\{\mathcal{C}_j^i\}_{i=1}^{I_j}$ be the connected components of $\{f_j\ge\lambda_*+\epsilon\}$ and $d(\mathcal{C}_j^i,\mathcal{C}_j^{i'})\ge\sigma$ for $i\ne i'$.*
There exists subsets A_j and A_j' for density f_j such that $A_{j,\sigma}\subset\mathcal{C}_j^i$ and $A_{j,\sigma}'\subset\mathcal{C}_j^{i'}$ with the following additional property.
Consider any algorithm that is given $n\ge 100$ i.i.d. samples $\{X_i\}_{i=1}^n$ from some $f_j\in F$ and, with probability at least 3/4, outputs a tree in which the smallest cluster containing $A_j\cap\{X_i\}_{i=1}^n$ is disjoint from the smallest cluster containing $A_j'\cap\{X_i\}_{i=1}^n$. Then there exists a constant $C(d)$ only depending on d such that

$$n\ge\frac{C(d)}{\sigma^d\lambda^*\epsilon^2}\log\frac{1}{\sigma^d\lambda^*}. \tag{53}$$

The proof of the proposition can be found in Theorem VI.1 of Chaudhuri et al. (2014). We omit the details of the proof for brevity.

Appendix D. A Cluster Consistency Resut for General Densities

The type of cluster consistency result we have obtained for densities with gaps can be easily generalized to arbitrary densities. To that end, we will introduce the notion of (ϵ, σ)-separated clusters and of h-thick clusters, where ϵ, σ and h are positive numbers..

Definition 14. *Let ϵ and σ be positive numbers. Two connected subsets A and A' of the support of P are said to be (ϵ, σ)-separated when*

- *they belong to different connected components of $L(\lambda^* - \epsilon)$, where $\lambda^* = \inf_{x \in A \cup A} f(x) > \epsilon$, and*
- $\min_{k \neq l} \operatorname{dist}(\mathcal{C}_k, \mathcal{C}_l) > \sigma$, *where $\mathcal{C}_1, \ldots, \mathcal{C}_m$ are the connected components of $L(\lambda^* - \epsilon)$.*

In addition to being well-separated, we further require the clusters to be *thick*, in a sense made precise below.

Definition 15. *A subset A is h-thick if $A_{-h} \neq \emptyset$.*

Remark 2. Comparison with the separation criterion of Chaudhuri et al. (2014). *The notion of (ϵ, σ)-separated clusters is analogous to the corresponding notion of separated clusters introduced in Chaudhuri et al. (2014), with ϵ and σ quantifying the degree of "vertical" and "horizontal" separation among clusters. There are however, two main differences. First, in our definition the parameter ϵ is on the same scale as the density p and, therefore, represents vertical separation among level sets in an additive and not multiplicative way. Secondly, we use different parameters to measure the degree of horizontal separation among clusters (σ) and the degrees of thickness of the clusters (h); in contrast, Chaudhuri et al. (2014) rely on just one parameter (σ in their notation) to express both separation and thickness.*

With these general notions of thick and well-separated clusters in place, we provide the following uniform consistency results for clustering, which applies to arbitrary densities.

Corollary 4. *Let $\{X_1, \ldots, X_n\}$ be an i.i.d. sample from a probability distribution P with an arbitrary density p and let $\epsilon > 0$ and $\sigma > 0$. Set $a_n = C_1 \sqrt{\frac{\log(n) + \log(1/h)}{nh^d}}$ as in (7) and suppose the input parameters h satisfying*

$$\sigma/4 \geq h \geq C \left(\frac{\log(n)}{n\epsilon^2} \right)^{1/d} \quad \text{and,} \tag{54}$$

for any $C > 0$ such that $2a_n < \epsilon$. Then with probability at least $1 - 1/n$, uniformly over all clusters A' and A that are h thick and (ϵ, σ) separated,

i. $A_{-h} \cap \{ X_1, \ldots, X_n\}$ and $A'_{-h} \cap \{ X_1, \ldots, X_n\}$, if non-empty, belong to distinct connected components of $\mathbb{G}_{k,h}$;

ii. all the sample points in A_{-h}, if any, belong to the same connected component of $\mathbb{G}_{k,h}$

where $k = \lceil nh^d V_d \lambda \rceil$, with $\lambda \in (\lambda^* - \epsilon + a_n, \lambda^* - a_n]$ and $\lambda^* = \inf_{xA \cup A'} f(x)$.

The proof of the previous corollary is almost the identical to the proof of Chaudhuri et al. (2014) and is omitted.

Remark 3. Optimality. *The scaling of the parameters (ϵ, σ) is minimax optimal, since the construction used in Chaudhuri et al. (2014) yields (ϵ, σ)-separated clusters that are also h thick with $h = \sigma/4$, so that the resulting lower bound applies to this case as well. Thus, DBSCAN delivers nearly minimax optimal cluster consistency with respect to our definitions of separated and thick clusters.*

Journal of Machine Learning Research 20 (2019) 1-37 Submitted 7/18; Published 10/19

Shared Subspace Models for Multi-Group Covariance Estimation

Alexander M. Franks AFRANKS@PSTAT.UCSB.EDU
Department of Probability and Applied Statistics Statistics
University of California, Santa Barbara
Santa Barbara, CA 93106, USA

Peter Hoff PETER.HOFF@DUKE.EDU
Department of Statistical Science
Duke University
Durham, NC 27708, USA

Editor: Barbara Engelhardt

Abstract

We develop a model-based method for evaluating heterogeneity among several $p \times p$ covariance matrices in the large p, small n setting. This is done by assuming a spiked covariance model for each group and sharing information about the space spanned by the group-level eigenvectors. We use an empirical Bayes method to identify a low-dimensional subspace which explains variation across all groups and use an MCMC algorithm to estimate the posterior uncertainty of eigenvectors and eigenvalues on this subspace. The implementation and utility of our model is illustrated with analyses of high-dimensional multivariate gene expression.

Keywords: covariance estimation; spiked covariance model; Stiefel manifold; large p, small n; high-dimensional data; empirical Bayes; gene expression data.

1. Introduction

Multivariate data can often be partitioned into groups, each of which represent samples from populations with distinct but possibly related distributions. Although historically the primary focus has been on identifying mean-level differences between populations, there has been a growing need to identify differences in population covariances as well. For instance, in case-control studies, mean-level effects may be small relative to subject variability; distributional differences between groups may still be evident as differences in the covariances between features. Even when mean-level differences are detectable, better estimates of the covariability of features across groups may lead to an improved understanding of the mechanisms underlying these apparent mean-level differences. Further, accurate covariance estimation is an essential part of many prediction tasks (e.g. quadratic discriminant analysis). Thus, evaluating heterogeneity between covariance matrices can be an important complement to more traditional analyses for estimating differences in means across groups.

To address this need, we develop a novel method for multi-group covariance estimation. Our method exploits the fact that in many natural systems, high dimensional data is often

©2019 Alexander M. Franks and Peter Hoff.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-484.html.

very structured and thus can be best understood on a lower dimensional subspace. For example, with gene expression data, we may be interested how the covariability between expression levels differs in subjects with and without a particular disease phenotype (e.g, how does gene expression covariability differ in different subtypes of leukemia? See Section 6). In these applications, the effective dimensionality is thought to scale with the number of gene regulatory modules, not the number of genes themselves (Heimberg et al., 2016). As such, differences in gene expression across groups should be expressed in terms of differences between these regulatory modules rather than strict differences between expression levels. Such differences can be examined on a subspace that reflects the correlations resulting from these modules. In contrast to most existing approaches for group covariance estimation, our approach is to directly infer such subspaces from groups of related data.

Some of the earliest approaches for multi-group covariance estimation focus on estimation in terms of spectral decompositions. Flury (1987) developed estimation and testing procedures for the "common principal components" model, in which a set of covariance matrices were assumed to share the same eigenvectors. Schott (1991, 1999) considered cases in which only certain eigenvectors are shared across populations, and Boik (2002) described an even more general model in which eigenvectors can be shared between some or all of the groups. More recently, Hoff (2009a), noting that eigenvectors are unlikely to be shared exactly between groups, introduced a hierarchical model for eigenvector shrinkage based on the matrix Bingham distribution. There has also been a significant interest in estimating covariance matrices using Gaussian graphical models. For Gaussian graphical models, zeros in the precision matrix correspond to conditional independence relationships between pairs of features given the remaining features (Meinshausen and Bühlmann, 2006). Danaher et al. (2014) extended existing work in this area to the multi-group setting, by pooling information about the pattern of zeros across precision matrices.

Another popular method for modeling relationships between high-dimensional multivariate data is partial least squares regression (PLS) (Wold et al., 2001). This approach, which is a special case of a bilinear factor model, involves projecting the data onto a lower dimensional space which maximizes the similarity of the two groups. This technique does not require the data from each group to share the same feature set. A common variant for prediction, partial least squares discriminant analysis (PLS-DA) is especially common in chemometrics and bioinformatics (Barker and Rayens, 2003). Although closely related to the approaches we will consider here, the primarily goal of PLS-based models is to create regression or discrimination models, not to explicitly infer covariance matrices from multiple groups of data. Nevertheless, the basic idea that data can often be well represented on a low dimensional subspace is an appealing one that we leverage.

The high-dimensional multi-group covariance estimation problem we explore in this work is also closely related to several important problems in machine learning. In particular, it can be viewed as an extension of distance metric learning methods (Bellet et al., 2012; Wang and Sun, 2015) to the multiple-metric setting. Multi-group covariance estimation also has applications in multi-task learning (Zhang et al., 2016; Liu et al., 2009), manifold and kernel learning tasks (Kanamori and Takeda, 2012), computer vision (Vemulapalli et al., 2013; Pham and Venkatesh, 2008) and compressed sensing and signal processing (Romero et al., 2016). Recently, covariance matrix and subspace learning has been used in deep learning applications (Huang and Van Gool, 2017).

In this paper we propose a multi-group covariance estimation model by sharing information about the subspace spanned by group-level eigenvectors. Our approach is closely related to the covariance reducing model proposed by Cook and Forzani (2008), but their model is applicable only when $n \gg p$. In this work we focus explicitly on high-dimensional inference in the context of the "the spiked covariance model" (also known as the "partial isotropy model"), a well studied variant of the factor model (Mardia et al., 1980; Johnstone, 2001). Unlike most previous methods for multi-group covariance estimation, our shared subspace model can be used to improve high-dimensional covariance estimates, facilitates exploration and interpretation of differences between covariance matrices, and incorporates uncertainty quantification. It is also straightforward to integrate assumptions used in previous approaches (e.g. eigenvector shrinkage) to the shared subspace model.

In Section 2 we briefly review the behavior of spiked covariance models for estimating a single covariance matrix and then introduce our extension to the multi-group setting. In Section 3 we describe an efficient empirical Bayes algorithm for inferring the shared subspace and estimating the posterior distribution of the covariance matrices of the data projected onto this subspace. In Section 4 we investigate the behavior of this class of models in simulation and demonstrate how the shared subspace assumption is widely applicable, even when there is little similarity in the covariance matrices across groups. In particular, independent covariance estimation is equivalent to shared subspace estimation with a sufficiently large shared subspace. In Section 5 we use an asymptotic approximation to describe how shared subspace inference reduces bias when both p and n are large. Finally, In Section 6 we demonstrate the utility of a shared subspace model in an analysis of gene expression data from juvenile leukemia patients . Despite the large feature size ($p > 3000$) relative to the sample size ($n < 100$ per group), we identify interpretable similarities and differences in gene covariances on a low dimensional subspace.

2. A Shared Subspace Spiked Covariance Model

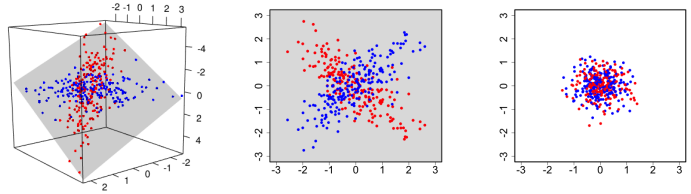
Suppose a random matrix S has a possibly degenerate Wishart(Σ, n) distribution with density given by

$$p(S|\Sigma, n) \propto l(\Sigma : S) = |\Sigma|^{-n/2}\text{etr}(-\Sigma^{-1}S/2), \tag{1}$$

where etr is the exponentiated trace, the covariance matrix is a positive definite matrix, i.e. $\Sigma \in \mathcal{S}_p^+$, and n may be less than p. Such a likelihood results from S being, for example, a residual sum of squares matrix from a multivariate regression analysis. In this case, n is the number of independent observations minus the rank of the design matrix.

In this paper we consider multi-group covariance estimation based on K matrices, $Y_1, ..., Y_K$, where Y_k is assumed to be an n_k by p matrix of mean-zero normal data, typically with $n_k \ll p$. Then, $Y_k^T Y_k = S_k$ has a (degenerate) Wishart distribution as in Equation 1. To improve estimation, we seek estimators of each covariance matrix, $\hat{\Sigma}_k$, that may depend on data from all groups. Specifically, we posit that the covariance matrix for each group can be written as

$$\Sigma_k = \sigma_k^2(V\Psi_k V^T + I), \tag{2}$$



(a) Projection in $\mathbb{R}^3$ (b) $Y_k V$ (c) $Y_k V_\perp$

Figure 1: Two groups of four-dimensional data (red and blue) projected into different subspaces. a) To visualize Y_k we can project the data into $\mathbb{R}^3$. In this illustration, the distributional differences between the groups are confined to a two-dimensional shared subspace (VV^T, grey plane). b) The data projected onto the two-dimensional shared subspace, $Y_k V$, have covariances Ψ_k that differ between groups. c) The orthogonal projection, $Y_k V_\perp$ has isotropic covariance, $\sigma_k^2 I$, for all groups.

where V is a $p \times s$ semi-orthogonal matrix whose columns form the basis vectors for subspace of variation shared by all groups. Ψ_k is a non-isotropic $s \times s$ covariance matrix for each group on this subspace of variation and it is assumed that $s \ll p$.

Our model extends the spiked principal components model (spiked PCA), studied extensively by Johnstone (2001) and others, to the multi-group setting. Spiked PCA assumes that

$$\Sigma = \sigma^2(U\Lambda U^T + I) \tag{3}$$

where for $s \ll p$, Λ is an $s \times s$ diagonal matrix and $U \in \mathcal{V}_{p,s}$, where $\mathcal{V}_{p,s}$ is the Stiefel manifold consisting of all $p \times s$ semi-orthogonal matrices in $\mathbb{R}^p$, so that $U^T U = I_s$. The spiked covariance formulation is appealing because it explicitly partitions the covariance matrix into a tractable low rank "signal" and isotropic "noise".

Classical results for parametric models (e.g., Kiefer and Schwartz (1965)) imply that asymptotically in n for fixed p, an estimator will be consistent for a spiked population covariance as long as the assumed number of spikes (eigenvalues larger than σ^2) is greater than or equal to the true number. However, when p is large relative to n, as is the case for the examples considered here, things are more difficult. Under the spiked covariance model, it has been shown that if $p/n \to \alpha > 0$ as $n \to \infty$, the kth largest eigenvalue of $S/(n\sigma^2)$ will converge to an upwardly biased version of $\lambda_k + 1$ if λ_k is greater than $\sqrt{\alpha}$ (Baik and Silverstein, 2006; Paul, 2007). This has led several authors to suggest estimating Σ via shrinkage of the eigenvalues of the sample covariance matrix. In particular, in the setting where σ^2 is known, Donoho et al. (2013) propose estimating all eigenvalues whose sample estimates are smaller than $\sigma^2(1 + \sqrt{\alpha})^2$ by σ^2, and shrinking the larger eigenvalues in a way that depends on the particular loss function being used. These shrinkage functions are shown to be asymptotically optimal in the $p/n \to \alpha$ setting.

Single-group covariance estimators of the spiked PCA form are equivariant with respect to rotations and scale changes, but the situation should be different, when we are interested in estimating multiple covariance matrices from distinct but related groups with shared features. Here, equivariance to distinct rotations in each group is an unreasonable assumption; both eigenvalue *and* eigenvector shrinkage can play an important role in improving covariance estimates.

In the multi-group setting, we account for similarity between group-level eigenvectors by positing that the anisotropic variability from each group occurs on a common low dimensional subspace. Throughout this paper we will denote to the shared subspace as $VV^T \in \mathcal{G}_{p,s}$, where $\mathcal{G}_{p,s}$ is the Grassmannian manifold consisting of all s-dimensional linear subspaces of $\mathbb{R}^p$ (Chikuse, 2012). Although V is only identifiable up to right rotations, the matrix VV^T, which defines the plane of variation shared by all groups, is identifiable for a fixed dimension, s. To achieve the most dimension reduction, we target the shared subspace of minimal dimension, e.g. the shared subspace for which all Ψ_k are full rank. Such a minimal subspace is known as the *central* subspace (Cook, 2009). Later, to emphasize the connection to the spiked PCA model (3), we will write Ψ_k in terms of its eigendecomposition, $\Psi_k = O_k \Lambda_k O_k$, where O_k are eigenvectors and Λ_k are the eigenvalues of Ψ_k (see Section 3.2).

For the shared subspace model, $V^T \Sigma_k V = \sigma_k^2(\Psi_k + I)$ is an anisotropic s-dimensional covariance matrix for the projected data, $Y_k V$. In contrast, the data projected onto the orthogonal space, $Y_k V_\perp$, is isotropic for all groups. In Figure 1 we provide a simple illustration using simulated 4-dimensional data from two groups. In this example, the differences in distribution between the groups of data can be expressed on a two dimensional subspace spanned by the columns of $V \in \mathcal{V}_{4,2}$. Differences in the correlations between the two groups manifest themselves on this shared subspace, whereas only the magnitude of the isotropic variability can differ between groups on the orthogonal space. Thus, a shared subspace model can be viewed as a covariance partition model, where one partition includes the anisotropic variability from all groups and the other partition is constrained to the isotropic variability from each group. This isotropic variability is often characterized as measurement noise.

3. Empirical Bayes Inference

In this section we outline an empirical Bayes approach for estimating a low-dimensional shared subspace and the covariance matrices of the data projected onto this space. As we discuss in Section 4, if the spiked covariance model holds for each group individually, then the shared subspace assumption also holds, where the shared subspace is simply the span of the group-specific eigenvectors, $U_1, ..., U_K$. In practice, we can usually identify a shared subspace of dimension $s \ll p$ that preserves most of the variation in the data. Our primary objective is to identify the "best" shared subspace of fixed dimension $s < p$. Note that this subspace accounts for the across-group similarity, and thus can be viewed as a hyperparameter in a hierarchical model. Although a fully Bayesian approach may be preferable in the absence of computational limitations, in this paper we propose computationally tractable empirical Bayes inference. In the empirical Bayes approach, hyperparameters are first estimated via maximum marginal likelihood, often using the expectation-maximization

algorithm (Lindstrom and Bates, 1988). In many settings such an approach yields group-level inferences that are close to that which would be obtained if the correct across-groups model were known (see for example Efron and Morris, 1973). In Section 3.1 we describe the expectation-maximization algorithm for estimating the maximum marginal likelihood of the shared subspace, VV^T. This approach is computationally tractable for high-dimensional data sets. Given an inferred subspace, we then seek estimators for the covariance matrices of the data projected onto this space. Because seemingly large differences in the point estimates of covariance matrices across groups may not actually reflect statistically significant differences, in Section 3.2 we also describe a Gibbs sampler that can be used to generate estimates of the projected covariance matrices, Ψ_k, and their associated uncertainty. Later, in Section 4 we discuss strategies for inferring an appropriate value for s and explore how shared subspace models can be used for exploratory data analysis by visualizing covariance heterogeneity on two or three dimensional subspaces.

3.1. Estimating the Shared Subspace

In this section we describe a maximum marginal likelihood procedure for estimating the shared subspace, VV^T, based on the expectation-maximization (EM) algorithm. The full likelihood for the shared subspace model can be written as

$$\begin{aligned}
p(S_1, ...S_k|\Sigma_k, n_k) &\propto \prod_{k=1}^K |\Sigma_k|^{-n_k/2} \text{etr}(-\Sigma_k^{-1} S_k/2) \\
&\propto \prod_{k=1}^K |\Sigma_k|^{-n_k/2} \text{etr}(-(\sigma_k^2(V\Psi_k V^T + I))^{-1} S_k/2) \\
&\propto \prod_{k=1}^K |\Sigma_k|^{-n_k/2} \text{etr}(-\left[V(\Psi_k + I)^{-1}/\sigma_k^2 V^T + (I - VV^T)/\sigma_k^2\right] S_k/2) \\
&\propto \prod_{k=1}^K (\sigma_k^2)^{-n_k(p-s)/2} |M_k|^{-n_k/2} \text{etr}(-\left[VM_k^{-1}V^T + \frac{1}{\sigma_k^2}(I - VV^T)\right] S_k/2),
\end{aligned} \tag{4}$$

where we define $M_k = \sigma_k^2(\Psi_k + I)$. The log-likelihood in V (up to an additive constant) is

$$\begin{aligned}
l(V) &= \sum_k \text{tr}\left(-(VM_k^{-1}V^T + VV^T/\sigma_k^2)S_k/2\right) \\
&= \frac{1}{2}\sum_k \text{tr}\left((\frac{1}{\sigma_k^2}I - M_k^{-1})V^T S_k V\right).
\end{aligned} \tag{5}$$

We maximize the marginal likelihood of V with an EM algorithm, where M_k^{-1} and $\frac{1}{\sigma_k^2}$ are considered the "missing" parameters. We assume independent Jeffreys prior distributions for both σ_k^2 and M_k. The Jeffreys prior distributions for these quantities correspond to $p(\sigma_k^2) \propto 1/\sigma_k^2$ and $p(M_k) \propto |M_k|^{-(s+1)/2}$. From the likelihood it can easily be shown that the conditional posterior for M_k is

$$p(M_k|V) \propto |M_k|^{-(n_k+s+1)/2} \text{etr}(-(M_k^{-1}V^T S_k V)/2)$$

Algorithm 1: Shared Subspace EM Algorithm

Initialize $V_0 \in \mathcal{V}_{p,s}$;
while $||V_t - V_{t-1}||_F > \epsilon$ **do**
  E-step:
  for $k \leftarrow 1$ ***to*** K **do**
    $\phi_t^{(k)} \leftarrow E[M_k^{-1}|V_{(t-1)}] = n_k(V_{(t-1)}^T S_k V_{(t-1)})^{-1}$;
    $\tau_t^{(k)} \leftarrow E[\frac{1}{\sigma_k^2}|V_{(t-1)}] = \frac{n_k(p-s)}{\mathrm{tr}[(I - V_{(t-1)}V_{(t-1)}^T)S_k]}$;
  end
  M-step:
  $V_t \leftarrow \arg\max_{V \in \mathcal{V}_{p,s}} \sum_k \mathrm{tr}\left(-(V\phi_t^{(k)}V^T + \tau_t^{(k)}VV^T)S_k/2\right)$;
end

which is an inverse-Wishart$(V^T S_k V,\, n_k)$ distribution. The conditional posterior distribution of σ_k^2 is simply

$$p\left(\sigma^2|V\right) \propto (\sigma_k^2)^{-n_k(p-s)/2-1}\mathrm{etr}\left(-(I - VV^T)S_k/[2\sigma_k^2]\right)$$

which is an inverse-gamma$(n_k(p-s)/2,\, \mathrm{tr}[(I - VV^T)S_k]/2)$ distribution. We summarize our approach in Algorithm 1 below.

For the M-step, we use a numerical algorithm for optimization over the Stiefel manifold. The algorithm uses the Cayley transform to preserve the orthogonality constraints in V and has computationally complexity that is dominated by the dimension of the shared subspace, not the number of features (Wen and Yin, 2013). Specifically, the optimization routine has time complexity $\mathrm{O}(ps^2 + s^3)$, and consequently, our approach is computationally efficient for relatively small values of s, even when p is large. Run times are typically on the order of minutes for values of p as large as 10,000 and moderate values of s (e.g. < 50). See Figure 10 in Appendix B for a plot with typical run times in simulations with a range of values of p and s.

Initialization and Convergence: The Stiefel manifold is compact and the marginal likelihood is continuous, so the likelihood is bounded. Thus, the EM algorithm, which increases the likelihood at each iteration, will converge to a stationary point (Wu, 1983). However, maximizing the marginal likelihood of the shared subspace model corresponds to a non-convex optimization problem over the Grassmannian manifold and may converge to a sub-optimal local mode or stationary point. Other work involving optimization on the Grassmannian has found convergence to non-optimal stationary values problematic and emphasized the importance of good (e.g. $\sqrt{n}$-consistent) starting values (Cook et al., 2016). Our empirical results on simulated data confirms that randomly initialized starting values converge to sub-optimal stationary values, and so in practice we initialize the algorithm at a carefully chosen starting value based on the eigenvectors of a pooled covariance estimate. We give the details for this initialization strategy below.

First, note that when the shared subspace model holds, the first s eigenvectors, from *any* of the groups can be used to construct a $\sqrt{n}$-consistent estimator of VV^T. In particular, if

$\hat{U}^{(k)}\hat{U}^{(k)^T}$ is the eigenprojection matrix for the subspace spanned by the first s eigenvectors of S_k then it can be shown that $\sqrt{n}\ \mathrm{vec}(\hat{U}^{(k)}\hat{U}^{(k)^T} - VV^T)$ converges in distribution to a mean-zero normal (Kollo, 2000). In the large p, small n setting, such classical asymptotic guarantees give little assurance that the resulting estimators would be reasonable, but they nevertheless suggest useful strategies for identifying starting value for the EM algorithm.

In this work, we choose a subspace initialization strategy based on sample eigenvectors of the data pooled from all groups. Let $Z = \sum_k \pi_k \frac{Z_k}{\sigma_k}$ where Z_k is a mean-zero normal with covariance Σ_k and $\pi_k = n_k / \sum_k n_k$. Then Z is a mixture of mean-zero normal distributions with covariance

$$\begin{aligned}\Sigma_Z &= \sum_k \frac{\pi_k}{\sigma_k^2}\Sigma_k \\ &= V^T(\sum_k \frac{\pi_k}{\sigma_k^2}\Psi_k)V + I,\end{aligned}$$

Clearly, the first s eigenvectors of Σ_Z span the shared subspace, VV^T. This suggests that we can estimate the shared subspace using the scaled and pooled data, $Y_{\mathrm{pool}} = [\frac{1}{\sigma_1}Y_1; \frac{1}{\sigma_2}Y_2; ...; \frac{1}{\sigma_k}Y_k]$, where Y_{pool} has dimension $(\sum_k n_k) \times p$. We use $\hat{U}_{\mathrm{pool}}\hat{U}_{\mathrm{pool}}^T$ as the initial value for subspace estimation algorithm where $\hat{U}_{\mathrm{pool}}$ are the first s eigenvectors of $S_{\mathrm{pool}} = Y_{\mathrm{pool}}^T Y_{\mathrm{pool}}$. If we treat Y_{pool} as an i.i.d. sample from the mixture distribution Z, then it is known that $\hat{U}_{\mathrm{pool}}\hat{U}_{\mathrm{pool}}^T$ is not consistent when both n and p growing at the same rate. For an arbitrary p-vector η, the asymptotic bias of $\eta^T\hat{U}_{\mathrm{pool}}\hat{U}_{\mathrm{pool}}^T\eta$ is well characterized as a function of the eigenvalues of Σ_Z (Mestre, 2008). If either the eigenvalues of $\sum_k \frac{\pi_k}{\sigma_k^2}\Psi_k$ or the total sample size $\sum_k n_k$ are large, $\hat{U}_{\mathrm{pool}}\hat{U}_{\mathrm{pool}}^T$ will accurately estimate the shared subspace and likelihood based optimization may not be necessary. However, when either the eigenvalues are small or the sample size is small the likelihood based analysis can significantly improve inference and $\hat{U}_{\mathrm{pool}}\hat{U}_{\mathrm{pool}}^T$ is a useful starting value for the EM algorithm.

Evaluating Goodness of Fit: Tests for evaluating whether eigenvectors from multiple groups span a common subspace were explored extensively by Schott (1991). These tests can be useful for assessing whether a shared subspace model is appropriate, but cannot be used to test whether a particular subspace explains variation across groups. These results are also based on classical asymptotics and are thus less accurate when $n \ll p$

Our goodness of fit measure is based on the fact that when V is a basis for a shared subspace, then for each group, most of the non-isotropic variation in Y_k should be preserved when projecting the data onto this space. To characterize the extent to which this is true for different groups, we propose a simple estimator for the proportion of "signal" variance that lies on a given subspace. Specifically, we use the following statistic for the ratio of the sum of the first s eigenvalues of $V^T\Sigma_k V$ to the sum of the first s eigenvalues of Σ_k:

$$\gamma(Y_k : V, \sigma_k^2) = \frac{||Y_k V||_F^2/n_k}{\max_{\widetilde{V} \in \mathcal{V}_{p,s}} ||Y_k \widetilde{V}||_F^2/n_k - B_k} \tag{6}$$

where $||\cdot||_F$ is the Frobenius norm and B_k is a bias correction where $B_k = \sigma_k^2 p/n_k \sum_k \left(\frac{m_i^{(k)}}{m_i^{(k)} - \sigma_k^2}\right)$ with $m_i^{(k)}$ the positive solution to the quadratic equation

$$(m_i^{(k)})^2 + m_i^{(k)}(\sigma_k^2 p/n_k - \sigma_k^2 - \hat{\lambda}_i^{(k)}) - \hat{\lambda}_i^{(k)}\sigma_k^2 = 0. \tag{7}$$

and $\hat{\lambda}_i^{(k)}$ is the i-th eigenvalue of S_k/n_k.

Theorem 1 *Assume $p/n_k \to \alpha_k$ and s is fixed. If $\Sigma_k = V\Psi_k V^T + \sigma_k^2 I$, then $\gamma(Y_k : V, \sigma_k^2) \overset{a.s}{\to} 1$ as $n_k, p \to \infty$.*

Proof Since s is fixed and n_k is growing, the numerator, $||Y_k V||_F^2/n_k$, is a consistent estimator for the sum of the eigenvalues of $V^T\Sigma_k V$. In the denominator, $\max_{\widetilde{V} \in \mathcal{V}_{p,s}} ||Y_k \widetilde{V}||_F^2/n_k$ is equivalent to the sum of the first s eigenvalues of the sample covariance matrix S_k/n_k. Baik and Silverstein (2006) and others have demonstrated that asymptotically as $p, n_k \to \infty$ and $p/n_k = \alpha_k$, $\hat{\lambda}_i^{(k)}$ is positively biased. Specifically,

$$\hat{\lambda}_i^{(k)} \overset{a.s.}{\to} \lambda_i^{(k)}\left(1 + \frac{\sigma_k^2 \alpha_k}{\lambda_i^{(k)} - \sigma_k^2}\right) \tag{8}$$

Replacing $\lambda_i^{(k)}$ by $m_i^{(k)}$ and assuming equality in 8 yields the quadratic equation 7. The solution, $m_i^{(k)}$, is an asymptotically (in n and p) unbiased estimator of $\lambda_i^{(k)}$ and

$$\max_{\widetilde{V} \in \mathcal{V}_{p,s}} ||Y_k \widetilde{V}||_F^2/n_k - B_k \overset{a.s.}{\to} \sum_i^s \lambda_i^{(k)} \tag{9}$$

As such, when the shared subspace model holds both the numerator and denominator of the goodness of fit statistic converge almost surely to $\sum_{i=1}^s \lambda_i^{(k)}$. Therefore $\gamma(Y_k : V, \sigma_k^2) \to 1$. ■

The goodness of fit statistic will be close to one for all groups when VV^T is a shared subspace for the data and typically smaller if not. The metric provides a useful indicator of which groups can be reasonably compared on a given subspace and which groups cannot. In practice, we estimate a shared subspace $\hat{V}$ and the isotropic variances $\hat{\sigma}_k^2$ using EM and compute the plug-in estimate $\gamma(Y_k : \hat{V}, \hat{\sigma}_k^2)$. When this statistic is small for some groups, it may suggest that the rank s of the inferred subspace needs to be larger to capture the variation in all groups. If $\gamma(Y_k : \hat{V}, \hat{\sigma}_k^2)$ is substantially larger than 1 for a particular group, it suggests that the inferred subspace is too similar to the sample principal components from group k. We investigate these issues in Section 4, by computing the goodness of fit statistic for inferred subspaces of different dimensions on a single data set. In Section 6, we compute the estimates for subspaces inferred with real biological data.

3.2. Inference for Projected Covariance Matrices

The EM algorithm presented in the previous section yields point estimates for VV^T, Ψ_k, and σ_k^2 but does not lead to natural uncertainty quantification for these estimates. In this section, we assume that the subspace VV^T is fixed and known and demonstrate how we can estimate the posterior distribution for Ψ_k. Note that when the subspace is known, the posterior distribution of Σ_k is conditionally independent from the other groups, so that we can independently estimate the conditional posterior distributions for each group.

There are many different ways in which we could choose to parameterize Ψ_k. Building on recent interest in the spiked covariance model (Donoho et al., 2013; Paul, 2007) we propose a tractable MCMC algorithm by specifying priors on the eigenvalues and eigenvectors of Ψ_k. By modeling the eigenstructure, we can now view each covariance Σ_k in terms of the original spiked principal components model. Equation 2, written as a function of V, becomes

$$\begin{aligned} \Psi_k &= O_k \Lambda_k O_k^T \\ \Sigma_k &= V\Psi_k V^T + \sigma_k^2 I. \end{aligned} \tag{10}$$

Here, we allow Ψ_k to be of rank $r \leq s$ dimensional covariance matrix on the s-dimensional subspace. Thus, Λ_k is an $r \times r$ diagonal matrix of eigenvalues, and $O_k \in \mathcal{V}_{s,r}$ is the matrix of eigenvectors of Ψ_k. For any individual group, this corresponds to the original spiked PCA model (Equation 3) with $U_k = VO_k \in \mathcal{V}_{p,r}$. Note that the V and O_k are jointly unidentifiable because for any $s \times s$ orthonormal matrix $W, VO = VW^T WO = \tilde{V}\tilde{O}$. Once we fix a basis for the shared subspace, O_k is identifiable. As such, O_k should only be interpreted relative to the basis V, as determined by the EM algorithm described in Section 3.1. Differentiating the ranks r and s is helpful because it enables us to independently specify a subspace common to all groups and the possibly lower rank features on this space that are specific to individual groups.

Although our model is most useful when the covariance matrices are related across groups, we can also use this formulation to specify models for multiple unrelated spiked covariance models. We explore this in detail in Section 4. In Section 6 we introduce a shared subspace model with additional structure on the eigenvectors and eigenvalues of Ψ_k to facilitate interpretation of covariance heterogeneity on a two-dimensional subspace.

The likelihood for Σ_k given the sufficient statistic $S_k = Y_k^T Y_k$ is given in Equation 1. For the spiked PCA formulation, we must rewrite this likelihood in terms of V, O_k, Λ_k and σ_k^2. First note that by the Woodbury matrix identity

$$\begin{aligned} \Sigma_k^{-1} &= (\sigma_k^2(U_k \Lambda_k U_k^T + I))^{-1} \\ &= \frac{1}{\sigma_k^2}(U_k \Lambda_k U_k^T + I)^{-1} \\ &= \frac{1}{\sigma_k^2}(I - U_k \Omega_k U_k^T), \end{aligned} \tag{11}$$

where the diagonal matrix $\Omega = \Lambda(I+\Lambda)^{-1}$, e.g. $\omega_i = \frac{\lambda_i}{\lambda_i+1}$. Further,

$$\begin{aligned}|\Sigma_k| &= (\sigma_k^2)^p |U_k \Lambda_k U_k^T + I| \\ &= (\sigma_k^2)^p |\Lambda_k + I| \\ &= (\sigma_k^2)^p \prod_{i=1}^r (\lambda_i + 1) \\ &= (\sigma_k^2)^p \prod_{i=1}^r (1-\omega_i), \end{aligned} \tag{12}$$

where the second line is due to Sylvester's determinant theorem. Now, the likelihood of V, O_k, Λ_k and σ_k^2 is available from Equation 1 by substituting the appropriate quantities for Σ_k^{-1} and $|\Sigma_k|$ and replacing U_k with VO_k:

$$L(\sigma_k^2, V, O_k\Omega_k : Y_k) \propto (\sigma_k^2)^{-n_k p/2} \mathrm{etr}(-\frac{1}{2\sigma_k^2} S_k) \left(\prod_{i=1}^r (1-\omega_{ki})\right)^{n_k/2} \mathrm{etr}(\frac{1}{2\sigma_k^2}(VO_k\Omega_k O_k^T V^T) S_k). \tag{13}$$

We use conjugate and semi-conjugate prior distributions for the parameters O_k, σ_k^2 and Ω_k to facilitate inference via a Gibbs sampling algorithm. In the absence of specific prior information, invariance considerations suggest the use of priors that lead to equivariant estimators. Below we describe our choices for the prior distributions of each parameter and the resultant conditional posterior distributions. We summarise the Gibbs Sampler in Algorithm 2.

Conditional distribution of σ_k^2: From Equation 13 it is clear that the inverse-gamma class of prior distributions is conjugate for σ_k^2. We chose a default prior distribution for σ_k^2 that is equivariant with respect to scale changes. Specifically, we use the Jeffreys prior distribution, an improper prior with density $p(\sigma_k^2) \propto 1/\sigma_k^2$. Under this prior, straightforward calculations show that the full conditional distribution of σ_k^2 is inverse-gamma$(n_k p/2, \mathrm{tr}[S_k(I - U_k\Omega_k U_k^T)/2])$, where $U_k = VO_k$.

Conditional distribution of O_k: Given the likelihood from Equation 13, it is easy to show that the class of Bingham distributions are conjugate for O_k (Hoff, 2009a,b). Again, invariance considerations lead us to use a rotationally invariant uniform probability measure on $\mathcal{V}_{s,p}$. Under this uniform prior, the full conditional distribution of O_k has a density proportional to the likelihood

$$p(O_k|\sigma_k^2, U_k, \Omega_k) \propto \mathrm{etr}(\Omega_k O_k^T V^T [S_k/(2\sigma_k^2)] V O_k). \tag{14}$$

This is a Bingham$(\Omega, V^T S_k V/(2\sigma^2))$ distribution on $\mathcal{V}_{s,r}$ (Khatri and Mardia, 1977). A Gibbs sampler to simulate from this distribution is given in Hoff (2009b).

Together, the prior for σ_k^2 and O_k leads to conditional (on V) Bayes estimators $\hat{\Sigma}(V^T S_k V)$ that are equivariant with respect to scale changes and rotations on the subspace spanned by V, so that $\hat{\Sigma}(aWV^T S_k V W^T) = aW\hat{\Sigma}(V^T S_k V)W$ for all $a > 0$ and $W \in \mathcal{O}_s$ (assuming an invariant loss function). Interestingly, if Ω_k were known (which it is not), then for a given invariant loss function the Bayes estimator under this prior minimizes the (frequentist) risk among all equivariant estimators (Eaton, 1989).

Algorithm 2: Gibbs Sampler for Projected Data Covariance Matrices

```
Estimate V̂ using EM (Algorithm 1). Initialize O_k, Λ_k, σ_k^2;
for s ← 1 to number of samples do
    for k ← 1 to K do
        Sample σ_k^2 from an inverse-gamma(n_k p/2, tr[S_k(I − V̂ O_k Ω_k V̂^T O_k^T)/2]);
        Sample O_k from a Bingham(Ω, V̂^T S_k V̂/(2σ^2));
        for i ← 1 to r do
            Sample (1 − ω_ki) from a gamma(n_k/2 + 1, c_ki n_k/2) truncated at 1;
            λ_ki ← ω_ki/(1 − ω_ki)
        end
    end
end
```

Conditional distribution for Ω_k: Here we specify the conditional distribution of the diagonal matrix $\Omega_k = \Lambda_k(I+\Lambda_k)^{-1} = \mathrm{diag}(\omega_{k1}, ...\omega_{kr})$. We consider a uniform(0,1) prior distribution for each element of Ω, or equivalently, an $F_{2,2}$ prior distribution for the elements of Λ. The full conditional distribution of an element ω_i of Ω is proportional to the likelihood function

$$p(\omega_{ki}|V, O_k, S_k) \propto_{\omega_{ki}} \left(\prod_{i=1}^r (1-\omega_{ki})^{n_k/2}\right) \mathrm{etr}(\frac{1}{2\sigma_k^2}(VO_k\Omega_k O_k^T V^T) S_k) \tag{15}$$

$$\propto (1-\omega_{ki})^{n_k/2} e^{c_{ki}\omega_{ki} n_k/2}, \tag{16}$$

where $c_{ki} = u_{ki}^T S_k u_{ki}/(n_k\sigma_k^2)$ and u_{ki} is column i of $U_k = VO_k$. It is straightforward to show that the density for $(1-\omega_{ki})$ is proportional to a gamma$(n_k/2+1, c_{ki}n_k/2)$ truncated at 1. Thus, we can easily sample from this distribution using inversion sampling. The behavior of the distribution for ω_{ki} is straightforward to understand: if $c_{ki} \leq 1$, then the function has a maximum at $\omega_{ki} = 0$, and decays monotonically to zero as $\omega_{ki} \to 1$. If $c_{ki} > 1$ then the function is uniquely maximized at $(c_{ki}-1)/c_{ki} \in (0,1)$. To see why this makes sense, note that the likelihood is maximized when the columns of U_k are equal to the eigenvectors of S_k corresponding to its top r eigenvalues (Tipping and Bishop, 1999). At this value of U_k, c_{ki} will then equal one of the top r eigenvalues of $S_k/(n_k\sigma_k^2)$. In the case that $n_k \gg p$, we expect $S_k/(n_k\sigma_k^2) \approx \Sigma_k/\sigma_k^2$, the true (scaled) population covariance, and so we expect c_{ki} to be near one of the top r eigenvalues of Σ_k/σ_k^2, say $\lambda_{ki}+1$. If indeed Σ_k has r spikes, then $\lambda_{ki} > 0$, $c_{ki} \approx \lambda_{ki}+1 > 1$, and so the conditional mode of w_{ki} is approximately $(c_{ki}-1)/c_{ki} = \lambda_{ki}/(\lambda_{ki}+1)$, the correct value. On the other hand, if we have assumed the existence of a spike when there is none, then $\lambda_{ki} = 0$, $c_{ki} \approx 1$ and the Bayes estimate of w_{ki} will be shrunk towards zero, as it should be. We summarise the full Gibbs sampling algorithm below.

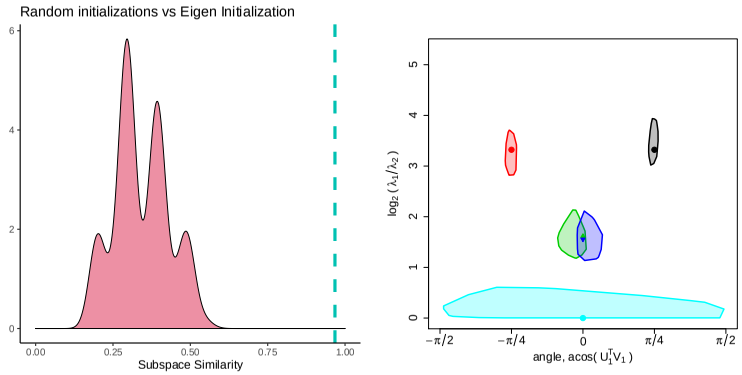
4. Simulation Studies

We start with an example demonstrating how a shared subspace model can be used to identify statistically significant differences between covariance matrices on a low dimensional

subspace. Here, we simulate $K = 5$ groups of data from the shared subspace spiked covariance model with $p = 20000$ features, a shared subspace dimension of $s = r = 2$, $\sigma_k^2 = 1$, and $n_k = 100$. We fix the first eigenvalue of Ψ_k from each group to $\lambda_1 = 1000$ and vary λ_2. We generate the basis for the shared subspace and the eigenvectors of Ψ_k by sampling uniformly from the Stiefel manifold. First, in Figure 2(a) we demonstrate the importance of the eigen-based initialization strategy proposed in Section 3.1. As an accuracy metric, we study the behavior of $\mathrm{tr}(\hat{V}\hat{V}^T VV^T)/s$ which is bounded by zero and one and achieves a maximum of one if and only if $\hat{V}\hat{V}^T$ corresponds to the true shared subspace. In this high dimensional problem, with random initialization, we typically converge to an estimated subspace that has a similarity between 0.25 and 0.5. With the eigen-based initialization we achieve nearly perfect estimation accuracy (> 0.95).

Next, we summarize estimates of Ψ_k inferred using Algorithm 2 in terms of its eigendecomposition by computing posterior distributions for the log eigenvalue ratio, $\log(\frac{\lambda_1}{\lambda_2})$, with $\lambda_1 > \lambda_2$, and the angle of the first eigenvector on this subspace, $\arctan(\frac{O_{12}}{O_{11}})$, relative to the first column of V. In Figure 2(b), we depict the 95% posterior regions for these quantities from a single simulation. Dots correspond to the true log ratios and orientations of $\hat{V}^T\Sigma_k\hat{V}$, where $\hat{V}$ is the maximum marginal likelihood for V. To compute the posterior regions, we iteratively remove posterior samples corresponding to the vertices of the convex hull until only 95% of the original samples remain. Non-overlapping posterior regions provide evidence that differences in the covariances are "statistically significant" between groups. In this example, the ratio of the eigenvalues of the true covariance matrices were 10 (black and red groups), 3 (green and blue groups) and 1 (cyan group). Larger eigenvalue ratios correspond to more correlated contours and a value of 1 implies isotropic covariance. Note that for the smaller eigenvalue ratio of 3, there is more uncertainty about the orientation of the primary axis. When the ratio is one, as is the case for the cyan colored group, there is no information about the orientation of the primary axis since the contours are spherical. In this simulation, the 95% regions all include the true data generating parameters. As we would hope, we find no evidence of a difference between the blue and green groups, since they have overlapping posterior regions. This means that a 95% posterior region for the difference between the groups (0,0), i.e. the model in which the angles and ratios are the same in both groups.

To demonstrate the overall validity of the shared subspace approach, we compute the frequentist coverage of these 95% Bayesian credible regions for the eigenvalue ratio and primary axis orientation using one thousand simulations. For the two groups with eigenvalue ratio $\lambda_1/\lambda_2 = 3$ the frequentist coverage was close to nominal at approximately 0.94. For the groups with $\lambda_1/\lambda_2 = 10$ the coverage was approximately 0.92. We did not evaluate the coverage for the group with $\lambda_1/\lambda_2 = 1$ (cyan) since this value is on the edge of the parameter space and is not covered by the 95% posterior regions as constructed. The slight under coverage for the other groups is likely due to the fact that we infer VV^T using maximum marginal likelihood, and thus ignore the extra variability due to the uncertainty about the shared subspace estimate.



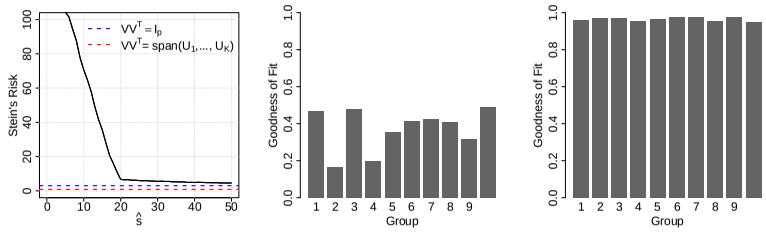
(a) Random vs eigen-based initialization

(b) Posterior eigen summaries

Figure 2: a) Accuracy of shared subspace estimation, $\mathrm{tr}(\hat{V}\hat{V}^T VV^T)/s$, for randomly initialized (density) and eigen-initialized value of V (dashed line). If V is initialized uniformly at random from the Stiefel manifold, then typically Algorithm 1 produces a subspace estimate that is sub-optimal. By contrast, using the initialization strategy described in Section 3.1, we achieve excellent accuracy. b) 95% posterior regions for the log of the ratio of eigenvalues, $\log(\frac{\lambda_1}{\lambda_2})$, of Ψ_k and the orientation of the principal axis on the space spanned by $\hat{V}$ cover the truth in this simulation. Dots correspond to true data generating parameter values on $\hat{V}^T\Sigma_k\hat{V}$. Since V is only identifiable up to rotation, for this figure we find the Procrustes rotation that maximizes the similarity of $\hat{V}$ to the true data generating basis. True eigenvalue ratios were 10 (red and black), 3 (green and blue) and 1 (cyan). True orientations were $\pi/4$ (black), $-\pi/4$ (red) and 0 (blue, green, and cyan). Note that the dark blue and green groups were generated with identical covariance matrices. Their posterior regions overlap, which suggests that a 95% region for the difference in eigenvalue ratios and angle would include (0,0).

4.1. Rank Selection and Model Misspecification

Naturally, shared subspace inference works well when the model is correctly specified. What happens when the model is not well specified? We explore this question *in silico* by simulating data from different data generating models and evaluating the efficiency of various covariance estimators. In all of the following simulations we evaluate covariance estimates using Stein's loss, $L_S(\Sigma_k, \hat{\Sigma}_k) = \mathrm{tr}(\Sigma_k^{-1}\hat{\Sigma}_k) - \log|\Sigma_k^{-1}\Sigma_k| - p$. Since we compute multi-group estimates, we report the average Stein's loss $L(\Sigma_1, ..., \Sigma_K; \hat{\Sigma}_1, ..., \hat{\Sigma}_K) = \frac{1}{K}\sum_k L_S(\Sigma_k, \hat{\Sigma}_k)$. Under Stein's loss, the Bayes estimator is the inverse of the posterior mean of the precision matrix, $\hat{\Sigma}_k = \mathrm{E}[\Sigma_k^{-1}|S_k]^{-1}$ which we estimate using MCMC samples.



(a) Stein's risk vs $\hat{s}$ (b) $\hat{s} = 5$ (c) $\hat{s} = 20$

Figure 3: a) Stein's risk as a function of the shared subspace dimension (solid black line). Data from ten groups, with U_k generated uniformly on the Stiefel manifold $\mathcal{V}_{200,2}$. As $\hat{s} \to p$, the risk converges to the risk from independently estimated spiked covariance matrices (dashed blue line). The data also fit a shared subspace model with $s = rK$. If $VV^T = \text{span}(U_1, ..., U_k)$ were known exactly, shared subspace estimation yields lower risk than independent covariance estimation (dashed red line). b) For a single simulated data set, the goodness of fit statistic, $\gamma(Y_k : \hat{V}, \hat{\sigma_k}^2)$, when the assumed shared subspace is dimension $\hat{s} = 5$. c). For the same data set, goodness of fit when the assumed shared subspace is dimension $\hat{s} = 20$. We can capture nearly all of the variability in each of the 10 groups using an $\hat{s} = rK = 20$ dimensional shared subspace.

We start by investigating the behavior of our model when we underestimate the true dimension of the shared subspace. In this simulation, we generate $K = 10$ groups of mean-zero normally distributed data with $p = 200$, $r = 2$, $s = p$ and $\sigma_k^2 = 1$. We fix the eigenvalues of Ψ_k to $(\lambda_1, \lambda_2) = (250, 25)$. Although the signal variance from each group individually is preserved on a two dimensional subspace, these subspaces are not similar across groups since the eigenvectors from each group are generated uniformly from the Stiefel manifold, $U_k \in \mathcal{V}_{p,r}$.

We use these data to evaluate how well the shared subspace estimator performs when we fit the data using a shared subspace model of dimension $\hat{s} < s$. In Figure 3(a) we plot Stein's risk as a function of $\hat{s}$, estimating the risk empirically using ten independent simulations per value of $\hat{s}$. The dashed blue line corresponds to Stein's risk for covariance matrices estimated independently. Independent covariance estimation is equivalent to shared subspace inference with $\hat{s} = p$ because this implies $VV^T = I_p$. Although the risk is large for small values of $\hat{s}$, as the shared subspace dimension increases to the dimension of the feature space, that is $\hat{s} \to p$, the risk for the shared subspace estimator quickly decreases. Importantly, it is always true that $\text{rank}([U_1, ..., U_K]) \leq rK$ so it can equivalently be assumed that the data were generated from a shared subspace model with dimension $s = rK < p$. As such, even when there is little similarity between the eigenvectors from each group, the shared subspace estimator with $\hat{s} = rK$ will perform well, provided that we can identify a subspace, $\hat{V}\hat{V}^T$

that is close to $\text{span}([U_1, ..., U_K])$. When $\hat{V}\hat{V}^T = \text{span}([U_1, ..., U_K])$ exactly, shared subspace estimation outperforms independent covariance estimation (3(a), dashed red line).

From this simulation, it is clear that correctly specifying the dimension of the shared subspace is important for efficient covariance estimation. When the dimension of the shared subspace is too small, we accrue higher risk. The goodness of fit statistic, $\gamma(Y_k : \hat{V}, \hat{\sigma_k}^2)$, can be used to identify when a larger shared subspace is warranted. When $\hat{s}$ is too small, $\gamma(Y_k : \hat{V}, \hat{\sigma_k}^2)$ will be substantially smaller than one for at least some of the groups, regardless of $\hat{V}$ (e.g. Figure 3(b)). When $\hat{s}$ is large enough, we are able to use maximum marginal likelihood to identify a shared subspace which preserves most of the variation in the data for all groups (Figure 3(c)). Thus, for any estimated subspace, the goodness of fit statistic can be used to identify the groups that can be fairly compared on this subspace and whether we would benefit from fitting a model with a larger value of $\hat{s}$.

Finally, in the appendix, we include a some additional misspecification results. In particular, we consider two cases in a 10 group analysis: one case in which 7 groups share a common subspace but the other three do not, and a second case in which five groups share one common two dimensional subspace, and the other five groups share a different two dimensional subspace (see Figures 8 and 9). Briefly, these results indicate that when only some of the groups share a common subspace, we can still usually identify both the existence of the subspace(s) shared by those groups. We can also identify which groups do not share the space, using the goodness of fit metric. When there are multiple relevant shared subspaces, we can often identify those distinct modes using a different subspace initialization for the EM algorithm.

Model Comparison and Rank Estimation: Clearly, correct specification for the rank of the shared subspace is important for efficient inference. So far in this section, we have assumed that the group rank, r, and shared subspace dimension, s, are fixed and known. In practice this is not the case. Prior to fitting a model we should estimate these quantities. Standard model selection methods can be applied to select the both s and r. Common approaches include cross validation and information criteria like AIC and BIC. However, these approaches are computationally intensive since they require fitting the model for each value of s and r. Here, we estimate the model dimensions by applying an asymptotically optimal (in mean squared error) singular value threshold for low rank matrix recovery with noisy data (Gavish and Donoho, 2014). This rank estimator is a function of the median singular value of the data matrix and the ratio $\alpha_k = p/n_k$. Note that under the shared subspace model, the scaled and pooled data described in section 3.1 can be expressed as $Y_{\text{pooled}} = X + Z$ where V are the left singular values of X and Z is a noise matrix with zero mean and variance one. This is the setting in which Gavish and Donoho (2014) develop a rank estimation algorithm, and so it can be appropriately applied to Y_{pooled} to estimate s.

Using this rank estimation approach, we conduct a simulation which demonstrates the relative performance of shared subspace group covariance estimation under different data generating models. We consider three different shared subspace data models: 1) a low dimensional shared subspace model with $s = r$; 2) a model in which the spiked covariance matrices from all groups are identical, e.g. $\Sigma_k = \Sigma = U\Lambda U^T + \sigma^2 I$; and 3) a full rank shared subspace model with $s = p$.

Table 1: Stein's risk (and 95% loss intervals) for different inferential models and data generating models with varying degrees of between-group covariance similarity. For each of $K = 10$ groups, we simulate data from three different types of shared subspace models. For each of these models, $p = 200$, $r = 2$, $\sigma_k^2 = 1$ and $n_k = 50$. We also fit the data using three different shared subspace models: a model in which s, r and VV^T are all estimated from the data ("adaptive"), a spiked covariance model in which the covariance matrices from each group are assumed to be identical ($\hat{\Sigma}_k = \hat{\Sigma}$) and a model in which we assume the data do *not* share a lower dimensional subspace across groups (i.e. $\hat{s} = p$). The estimators which most closely match the data generating model have the lowest risk (diagonal) but the adaptive estimator performs well relative to the alternative misspecified model.

		Inferential Model		
		Adaptive	$\hat{\Sigma}_k = \hat{\Sigma}$	$\hat{s} = p$
Data Model	$s = r = 2$	0.8 (0.7, 0.9)	2.1 (1.7, 2.6)	3.0 (2.9, 3.2)
	$s = r = 2$, $\Sigma_k = \Sigma$	0.8 (0.7, 0.9)	0.7 (0.6, 0.8)	3.0 (2.9, 3.2)
	$s = p = 200$	7.1 (6.2, 8.0)	138.2 (119, 153)	3.0 (2.9, 3.2)

We estimate group-level covariance matrices from simulated data using three different variants of the shared subspace model. For each of these fits we estimate r. First, we estimate a single spiked covariance matrix from the pooled data and let $\hat{\Sigma}_k = \hat{\Sigma}$. Second, we fit the full rank shared subspace model. This corresponds to a procedure in which we estimate each spiked covariance matrix independently, since $s = p$ implies $VV^T = I_p$. Finally, we use an "adaptive" shared subspace estimator, in which we estimate both s, r and VV^T.

Since full rank estimators do not scale well, we compare the performance of various estimators on a simulated data set with only $p = 200$ features. We also assume for $r = 2$ spikes, $\sigma_k^2 = 1$, and $n_k = 50$. We fix the non-zero eigenvalues of Ψ_k to $(\lambda_1, \lambda_2) = (250, 25)$. We simulate 100 independent data sets for each data generating mechanisms. In Table 1 we report the average Stein's risk and corresponding 95% loss intervals for the estimates derived from each of these inferential models.

As expected, the estimates with the lowest risk are derived from the inferential model that most closely match the data generating specifications. However, the adaptive estimator has small risk under model misspecification relative to the alternatives. For example, when $\Sigma_k = \Sigma$, the adaptive shared subspace estimator has almost four times smaller risk than the full rank estimator, in which each covariance matrix is estimated independently. When the data come from a model in which $s = p$, that is, the eigenvectors of Ψ_k are generated uniformly from $\mathcal{V}_{p,r}$, the adaptive estimator is over an order of magnitude better than the estimator which assumes no differences between groups. These results suggest that empirical Bayes inference for VV^T combined with the rank estimation procedure suggested by Gavish and Donoho (2014) can be widely applied to group covariance estimation because the estimator adapts to the amount of similarity across groups. Thus, shared subspace estimation can be an appropriate and computationally efficient choice when the similarity between groups is not known *a priori*.

Finally, in addition to potential statistical efficiency gains, the empirical Bayes shared subspace estimator has significant computational advantages. In particular, the total run time for empirical Bayes inference of the shared subspace is significantly smaller than full Bayesian inference for a $p \times r$ dimensional subspace (e.g. Bayesian probabilistic PCA with $s = p$), in particular for larger values of p. Given the difficulty of Bayesian inference on the Stiefel manifold, for large p, probabilistic principal component analysis quickly becomes infeasible. Empirical Bayes inference enables efficient optimization for $\hat{V}$ and Bayesian inference on the lower dimensional shared subspace (See Figure 10, Appendix B, for typical run times).

5. Reduction of Asymptotic Bias Via Pooling

Recently, there has been an interest in the asymptotic behavior of PCA-based covariance estimators in the setting in which $p, n \to \infty$ with $p/n = \alpha$ fixed. Specifically, in the spiked covariance model it is known that when p and n are both large, the leading eigenvalues of the sample covariance matrix are positively biased and the empirical eigenvectors form a non-zero angle with the true eigenvectors (Baik and Silverstein, 2006; Paul, 2007). Although this fact also implies that the shared subspace estimators are biased, a major advantage of shared subspace inference over independent estimation of multiple covariance matrices is that we reduce the asymptotic bias, relative to independently estimated covariance matrices, by pooling information across groups. The bias reduction appears to be especially large when there is significant heterogeneity in the first s eigenvectors of the projected covariance matrices.

Throughout this section we assume K groups of data each with $n_k = n$ observations per group and s a fixed constant. First, note that if $\hat{V}\hat{V}^T$ corresponds to the true shared subspace, then estimates $\hat{\psi}_k$ derived using the methods presented in Section 3.2 will consistently estimate ψ_k as $n \to \infty$ regardless of whether p increases as well because $Y_k V$ has a fixed number of columns. For this reason, we focus explicitly on the accuracy of $\hat{V}\hat{V}^T$ (derived using the maximum marginal likelihood algorithm presented in Section 3.1) as a function of the number of groups K when both p and n are of the same order of magnitude and much larger than s. As an accuracy metric, we again study the behavior of $\text{tr}(\hat{V}\hat{V}^T VV^T)/s$ which is bounded by zero and one and achieves a maximum of one if and only if $\hat{V}\hat{V}^T$ corresponds to the true shared subspace.

Conjecture 2 *Assume that the first s eigenvalues from each of K groups are identical with $\lambda_i > \sigma^2(1 + \sqrt{\alpha})$. Then, for $p/n \to \alpha$ and $p, n \to \infty$, $tr(\hat{V}\hat{V}^T VV^T)/s \overset{a.s.}{\to} \xi$ with*

$$1 > \xi \geq \frac{1}{s} \sum_{i=1}^{s} \left(1 - \frac{\alpha}{K(\lambda_i - 1)^2}\right) \Big/ \left(1 + \frac{\alpha}{K(\lambda_i - 1)}\right). \tag{17}$$

We prove that the lower bound in 17 is in fact achieved when Y_k are identically distributed and show in simulation that the subspace accuracy exceeds this bound when there is variation in the eigenvectors across groups. In the case of i.i.d. groups, let the covariance matrix $\Sigma_k = \Sigma$ have the shared-subspace form given in Equation 2 and without loss of generality let $\psi_k = \psi$ be a diagonal matrix (e.g assume the columns of V align with the

eigenvectors of Σ). In this case, the complete data likelihood of V (Equation 5) can be rewritten as

$$\begin{aligned}\ell(V) &= \frac{1}{2}\sum_k \operatorname{tr}\left((\frac{1}{\sigma^2}I - M^{-1})V^T S_k V\right)\\ &= \frac{1}{2}\operatorname{tr}\left(DV^T(\sum_k S_k)V\right).\end{aligned}$$

where $\sum_{k=1}^K S_k \sim \text{Wish}(\Sigma, Kn)$. Since ψ is diagonal and $\sigma^2 = 1$, $M = \sigma^2(\psi + I)$ is diagonal and thus $D = (\frac{1}{\sigma^2}I - M^{-1})$ is also diagonal with entries $0 < d_i < 1$ of decreasing magnitude.

Then, the solution to

$$\hat{V}^{(k)} = \underset{\widetilde{V}\in\mathcal{V}_{p,s}}{\operatorname{argmax}} \operatorname{tr}\left(D\widetilde{V}^T\sum_k (S_k)\widetilde{V}\right).$$

has $\hat{V}^{(k)}$ equal to the first s eigenvectors of $\sum_k S_k$. This is maximized when the columns of V match the first empirical eigenvectors of $\sum_k S_k$ and has a maximum of $\sum_{i=1}^r d_i\ell_i$ where ℓ_i is the ith eigenvalue of $\sum_k S_k$. Using a result from Paul (2007), it can be shown that as long as $\lambda_i > \sigma^2(1+\sqrt{\alpha})$ where λ_i is the ith eigenvalue of Σ_k, the asymptotic inner product between the ith sample eigenvector and the ith population eigenvector approaches a limit that is almost surely less than one

$$|\langle \hat{V}_i, V_i\rangle| \overset{a.s.}{\to} \sqrt{\left(1 - \frac{\alpha}{K(\lambda_i - 1)^2}\right) / \left(1 + \frac{\alpha}{K(\lambda_i - 1)}\right)}$$

As such, we can express asymptotic shared subspace accuracy for the identical groups model as

$$\begin{aligned}\operatorname{tr}(\hat{V}\hat{V}^T VV^T)/s &= \frac{1}{s}\sum_{i=1}^s |\langle \hat{V}_i, V_i\rangle|^2\\ &\overset{a.s.}{\to} \frac{1}{s}\sum_{i=1}^s \left(1 - \frac{\alpha}{K(\lambda_i - 1)^2}\right) / \left(1 + \frac{\alpha}{K(\lambda_i - 1)}\right).\end{aligned} \tag{18}$$

Here, the accuracy of the estimate depends on α, K and the magnitude of the eigenvalues, with the bias naturally decreasing as the number of groups increases. Most importantly, Equation 18 provides a useful benchmark for understanding the bias of shared subspace estimates in the general setting in which ψ_k varies across groups. Our conjecture that the subspace accuracy is larger than the lower bound when the eigenvectors between groups are variable is consistent with our simulation results.

In Figure 4 we depict the subspace accuracy metric $\operatorname{tr}(\hat{V}\hat{V}^T VV^T)/s$ and benchmark $\frac{1}{s}\sum_{i=1}^s \left(1 - \frac{\alpha}{K(\lambda_i-1)^2}\right) / \left(1 + \frac{\alpha}{K(\lambda_i-1)}\right)$ for simulated multi-group data generated under the shared subspace model with $s = 2$, $n = 50$, $p = 200$ and three different sets of eigenvalues. For each covariance matrix, the eigenvectors of ψ_k were sampled uniformly from Stiefel manifold $\mathcal{V}_{2,2}$. When ψ_k is isotropic (green) the subspace similarity metric closely matches the

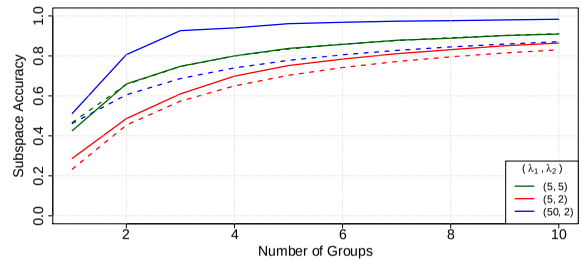


Figure 4: Subspace accuracy $\operatorname{tr}(\hat{V}\hat{V}^T VV^T)/s$ (solid) and the asymptotics-based benchmark (dashed) as a function of K. When $\lambda_1 = \lambda_2$ (green), the assumptions used to derive the benchmark (identically distributed groups) are met and thus the subspace accuracy matches the benchmark. However, when the ratio λ_1/λ_2 is large, the subspace accuracy metric can far exceed this benchmark if there is significant variation in the eigenvectors of ψ_k across groups. Small increases in accuracy over the benchmark are seen for moderately anisotropic data (red) and large increases for highly anisotropic data (blue).

benchmark since the assumptions used to derive this asymptotic result are met. However, when the eigenvectors of ψ_k vary significantly across groups and $\lambda_1 \gg \lambda_2$, the subspace accuracy can far exceed this benchmark (blue). Intuitively, when the first eigenvectors of two different groups are nearly orthogonal, each group provides a lot of information about orthogonal directions on VV^T and so the gains in accuracy exceed those that you would get by estimating the subspace from a single group with K times the sample size. In general the accuracy of shared subspace estimates depends on the variation in the eigenvectors of ψ_k across groups as well as the magnitude of the eigenvalues and matrix dimensions p and n_k. Although the shared subspace estimator improves on the accuracy of individually estimated covariance matrices, estimates can still be biased when α is very large or the eigenvalues of Σ_k are very small for all k. In practice, one should estimate the approximate magnitude of the bias using the inferred eigenvalues of Σ_k. When these inferred eigenvalues are significantly larger than $\hat{\sigma}_k^2(1 + \sqrt{\alpha/K})$ the bias will likely be small.

6. Analysis of Gene Expression Data

We demonstrate the utility of the shared subspace covariance estimator for exploring differences in the covariability of gene expression levels in young adults with different subtypes of pediatric acute lymphoblastic leukemia (ALL) (Yeoh et al., 2002). Quantifying biological variation across different subtypes of leukemia is important for assigning patients to risk groups, proposing appropriate treatments, and developing a deeper understanding of the mechanisms underlying these different types of cancer. The majority of studies have focused

on mean level differences between expression levels. In particular, mean-level differences can be useful for identifying leukemia subtypes. However, differences in the covariance structure across groups can be induced by interactions between important unobserved variables. Covariance analysis is particularly important when the effects of unobserved variables, like disease severity, disease progression or unmeasured genetic confounders, dominate mean level differences across groups. In this analysis, we explicitly remove the mean from the data and look for differences in the covariance structure of the gene expression levels.

The data we analyze were generated from 327 bone marrow samples analyzed on an Affymetrix oligonucleotide microarray with over 12,000 probe sets. Preliminary analysis using mean differences identified clusters corresponding to distinct leukemia subtypes: BCR-ABL, E2A-PBX1, hyperdiploid, MLL, T-ALL, TEL-AML1. 79 patients were assigned to a seventh group for unidentified subtypes ("Others"). We use these labels to stratify the observations into seven groups with corresponding sample sizes of $n = (15, 27, 64, 20, 43, 79, 79)$.

Although there are over 12,000 probes on the microarray, the vast majority of gene expression levels are missing. Thus, we restrict our attention to the genes for which less than half of the values are missing and use Amelia, a software package for missing value imputation, to fill in the remaining missing values (Honaker et al., 2011). Amelia assumes the data is missing at random and that each group is normally distributed with a common covariance matrix. Since imputation is done under the assumption of covariance homogeneity, any inferred differences between groups are unlikely to be an artifact of the imputation process. We leave it to future work to incorporate missing data imputation into the shared subspace inference algorithm. After removing genes with very high percentages of missing values, $p = 3124$ genes remain. Prior to analysis, we de-mean both the rows and columns of the gene expression levels in each group.

We apply the rank selection criteria discussed in Section 4.1 and proposed by Gavish and Donoho (2014) to the pooled expression data (i.e. data from all groups combined) to decide on an appropriate value for the shared subspace. This procedure yields $s = 45$ dimensions[1]. We run Algorithm 1 to estimate the shared subspace, and then use Bayesian inference (Algorithm 2) to identify differences between groups on the inferred subspace. Together, the run time for the full empirical Bayes procedure (both algorithms) took less than 10 minutes on a 2017 Macbook Pro.

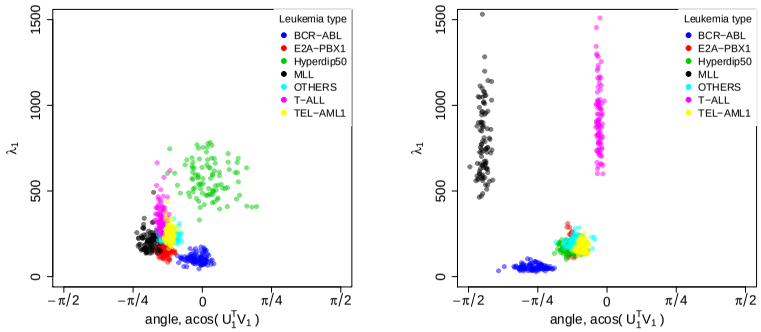
Using the goodness of fit metric, we find that a 45-dimensional shared subspace dimension that explains over 90% of the estimated variation in the top s eigenvectors of Σ_k, suggesting that the rank selection procedure worked reasonably well (Figure 11(a), Appendix B). To further validate the utility of shared subspace modeling, we look at how informative the projected data covariance matrices are for predicting group membership. For an observation Y_i, we compute the probability, assuming uniform prior distribution over group membership, that Y_i came from group k as $P(Y_i \text{ from group } k) = \frac{|\Psi_k|^{-1/2}\text{etr}(-1/2(Y_i\hat{V})^T\Psi_k^{-1}Y_i\hat{V})}{\sum_j(|\Psi_j|^{-1/2}\text{etr}(-1/2(Y_i\hat{V})^T\Psi_j^{-1}Y_i\hat{V}))}$. We correctly identified the leukemia type in all samples, which provides further confirmation that this subspace provides enough predictive power to easily distinguish groups.

In addition, we quantified differences amongst the projected data covariances using the Frobenius norm, $||\Psi_k - \Psi_j||_F$ for all pairs of the seven groups. We use these distances to compute a hierarchical clustering dendrogram of the groups (Figure 11(b), Appendix B). The hierarchical clustering reveals that BCR-ABL, E2A-PBX1, TEL-AML1 and hyperdiploid, which correspond to B lineage leukemias, cluster together. T-ALL, the T lineage leukemia, and MLL, the mixed lineage leukemia, appear the most different (Dang, 2012). To further verify that the inferred subspace relates to relevant biological processes, we conducted gene set enrichment analysis using the observed magnitudes of the loadings for the genes on the 45 basis vectors (Subramanian et al., 2005) and using gene sets defined by the Gene Ontology Consortium (Consortium et al., 2004). Gene set analysis on the magnitudes of gene loadings identified dozens of pathways (FDR < 0.01, (Storey et al., 2003)). Nearly every identified pathway relates to the immune response or cell growth (Figure 12, Appendix B), for example B and T cell proliferation (GO:0042100, GO:0042102), immunoglobin receptor binding (GO:0034987) and cellular response to cytokine stimulus (GO:0071345) to name only a few. Together, all of these results suggest that in this application there is indeed significant differences in the covariability between genes for each the of groups, with biologically plausible underpinnings. Consequently, there is value in exploring what underlies those differences.

1. Note that for some groups, $n_k < 45$, in which case we infer the rank $r = \min(n_k, s)$ $s \times s$ matrix Ψ_k.

We next demonstrate how we can explore significant *a posteriori* differences between the groups which might lead to scientifically meaningful insights. In order to visualize differences in the posterior distributions of the 45×45 dimensional matrices Ψ_k, we examine the distribution of eigenvalues and eigenvectors between the groups on a variety of two-dimensional subspaces of the shared space. We propose two different methods for identifying potentially interesting sub-subspaces to visualize. First, we summarize variation on a two dimensional subspace whose axes are approximately aligned to the first two eigenvectors of $\hat{\Sigma}_k$, for a specific group k. This subspace corresponds to the subspace of maximal variability within group k. For example, in Figure 5(a) we plot posterior summaries about the principal eigenvector and eigenvalues for each group on a two dimensional space spanned by the first two eigenvectors of the inferred covariance matrix for the hyperdiploid group. The x-axis corresponds to the orientation of the first eigenvector and the y-axis corresponds the magnitude of the first eigenvalue. In this subspace, we can see that the first eigenvector for most groups appear to have similar orientations, but that the hyperdiploid group has significantly larger variance along the first principal component direction than all other groups (with the exception of perhaps T-ALL, for which the posterior samples overlap). The first eigenvector for the BCR-ABL subgroup appears to be the least variable on this subspace.

As an alternative approach to summarizing the posterior distribution, we examine the posterior eigen-summaries on a two dimensional subspace which is chosen to maximize the difference between any two chosen groups. To achieve this, we look at spaces in which the axes correspond to the first two eigenvectors of $\hat{\Sigma}_k - \hat{\Sigma}_j$ for any $k \neq j$. As an example, in Figure 5(b) we plot posterior summaries corresponding to the subspace for which the difference between the T-ALL and MLL subgroups is large. On this subspace, the groups cluster into four distinct subgroups which appear significantly different *a posteriori*: the T-ALL subtype, the MLL subtype, the BCR subtype and the all other groups. Roughly, along the first dimension, there is large variability in the T-ALL group that is not matched in other groups, whereas the second dimension there is large variability in the MLL group that is not matched in the other groups.



(a) Subspace for hyperdiploid subtype

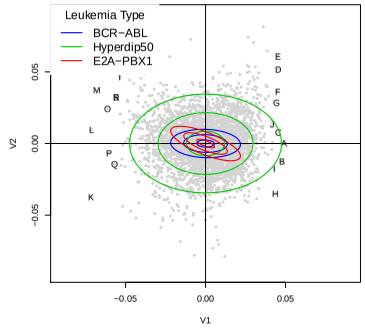
(b) T-ALL vs MLL subspace

Figure 5: Posterior samples for the first eigenvalue and orientation of the first eigenvector on the a dimensional subspace. a) The two dimensional subspace was chosen to approximately span the first two eigenvectors for the hyperdiploid group. The orientation of first eigenvector is similar for all groups, but the variance significantly larger for the hyperdiploid subgroup. b) The two dimensional subspace was chosen to maximize the difference between the T-ALL and MLL groups. Along the first dimension of this subspace, there is large variability in the T-ALL group that is not matched in other groups, whereas the second dimension there is large variability in the MLL group that is not matched in the other groups.

Scientific insights underlying the significant differences that were identified in Figure 5 can be understood in the biplots in Figures 6 and 7. In each figure, we plot the contours of the two dimensional covariance matrices for a few leukemia subtypes. The 20 genes with the largest loadings for one of the component directions are indicated with letters and the remaining loadings plotted with light grey dots. The gene names for the genes with the largest loadings are listed in the corresponding table. In both biplots, the identified genes have known connections to cancer, leukemia, and the immune system.

For example, for the subspace of maximal variability in the hyperdiploid group, gene set analysis identified two gene sets with large magnitude loadings on the first principal component: a small group of proteins corresponding to the MHC class II protein complex (GO:0006955) as well as a larger group of genes corresponding to genes generally involved in immune response (GO:0006955). MHC class II proteins are known to play an essential role in the adaptive immune system (Reith et al., 2005) and are correlated with leukemia patient outcomes (Rimsza et al., 2004). Our analysis indicates these proteins have especially variable levels in the hyperdiploid subtype relative to the other leukemia subtypes.

For the subspace chosen to maximize the difference between T-ALL and MLL groups, gene set analysis associated with large loadings in the second dimension (associated with

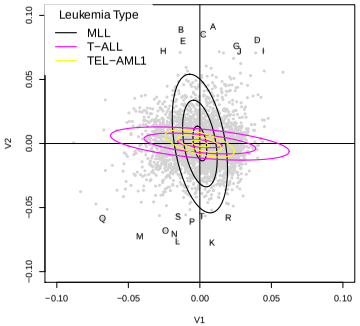


	Positive		Negative
A	HLA-DQB1	K	BCL11A
B	HBG1	L	HHLA1
C	SASH3	M	
D	HLA-DPB1	N	CSHL1
E	MME	O	NF2
F	HLA-DQB1	P	SKAP2
G	DPYSL2	Q	TRDV2
H	PRPF6	R	EIF2AK2
I	ADA	S	
J	ATP6V0E2	T	PMF1

Figure 6: Left) Variant of a biplot for the hyperdiploid subspace. We include contours for three leukemia subtypes and the loadings for each gene on the first two columns of $\hat{V}$. We plot contours for three leukemia subtypes and the loadings for genes with the most postive (A-J) and most negative (K-T) values on the first principal axis. The loadings for all of the genes are displayed in light gray. There is significant correlated variability amongst genes A-T in the TEL and hyperdiploid subgroups, and a factor of two less variability amongst these genes in the E2A subgroup. Right) List of the gene's with the largest loadings along the first axis.

high variance in the MLL subgroup) included "regulation of myeloid cell differentiation" (GO:0045637), "positive regulation of B cell receptor signaling pathway" (GO:0098609) and "immunoglobulin V(D)J recombination" (GO:0033152). Most of the individual genes with large loadings are known in the leukemia literature including WASF1 ("F") which plays an important role in apoptosis (Kang et al., 2010), LEF1 ("D") which is linked to the pathogenesis of leukemia (Gutierrez et al., 2010) and LMO2 ("M") which was shown to initiate leukemia in mice (McCormack et al., 2010), to name only a few. In contrast to the MLL group, these genes in the T-ALL and TEL-AML1 subgroups have relatively little variability.

These insights would be overlooked in more conventional mean-based analyses, particularly when mean-level differences are small relative to the residual variance. Further, we have shown how the shared subspace reveals sets of interpretable genes that are most important for describing difference between leukemia subtypes; these discoveries would less evident with alternative covariance estimation methods which do not explicitly include the assumption about differences manifesting on a common low dimensional subspace. All told, these results highlight the value of shared subspace covariance matrix inference for both predicting leukemia subtypes as well as for exploring scientifically meaningful differences between the groups.



	Positive		Negative
A	SELL	K	AHNAK
B	CD24	L	NR3C1
C	SH3BP5	M	LMO2
D	LEF1	N	NR3C1
E	CCR7	O	GSN
F	WASF1	P	SERPINB1
G	LSP1	Q	CSHL1
H	FADS3	R	DPYSL2
I	LCK	S	NKG7
J	LCK	T	DAD1

Figure 7: Left) Variant of a biplot for the MLL vs TEL-AML1 subspace. We plot contours for three leukemia subtypes and the loadings for genes with the most positive (A-J) and most negative (K-T) values on the second axis. The loadings for all of the genes are displayed in light gray. There is significant correlated variability among genes with large loadings (e.g. letters A through T) in the MLL subgroup, and a significantly less variability in the TEL-AML1 and T-ALL groups. Although the TEL and T-ALL groups have similar variance in the "V2" direction, T-ALL has significantly more variance in the "V1" direction. Right) List of the gene's with the largest loadings along the V2 axis.

7. Discussion

In this paper, we proposed a class of models for estimating and comparing differences in covariance matrices across multiple groups on a common low dimensional subspace. We described an empirical Bayes algorithm for estimating this common subspace and a Gibbs sampler for inferring the projected covariance matrices and their associated uncertainty. Estimates of both the shared subspace and the projected covariance matrices can both be useful summaries of the data. For example, with the leukemia data, the shared subspace highlights the full set of genes that are correlated across groups. Differences between group covariance matrices can be understood in terms of differences in these sets of correlated molecules. In this analysis, we demonstrated how we can use these notions to visualize and contrast the posterior distributions of covariance matrices projected onto a particular subspace and interpret these differences biologically.

In simulation, we showed that the shared subspace model can still be a reasonable choice for modeling multi-group covariance matrices even when the groups may be largely dissimilar. When there is little similarity between groups, the shared subspace model can still be appropriate as long as the dimension of the shared subspace is large enough. However, selecting the rank of the shared subspace remains a practical challenge. Although we propose a useful heuristic for choosing the dimension of the shared subspace based on the rank

selection estimators of Gavish and Donoho (2014), a more principled approach is warranted. Improved rank estimators would further improve the performance of the adaptive shared subspace estimator discussed in Section 4.

It is also a challenging problem to estimate the "best" subspace once the rank of the space is specified. We used maximum marginal likelihood to estimate VV^T and then used MCMC to infer Ψ_k. By focusing on group differences for Ψ_k on a *fixed* subspace, it is much simpler to interpret similarities and differences. Nevertheless, full uncertainty quantification for VV^T can be desirable. We found MCMC inference for VV^T to be challenging for the problems considered in this paper and leave it for future work to develop an efficient fully Bayesian approach for estimating the joint posterior of VV^T and Ψ_k. Recently developed Markov chain Monte Carlo algorithms, like Riemannian manifold Hamilton Monte Carlo, which can exploit the geometry of the Grassmannian manifold, may be useful here (Byrne and Girolami, 2013; Girolami and Calderhead, 2011). It may also be possible, though computationally intensive, to jointly estimate s and VV^T using for instance, a reversible-jump MCMC algorithm.

Fundamentally, our approach is quite general and can be integrated with existing approaches for multi-group covariance estimation. In particular, we can incorporate additional shrinkage on the projected covariance matrices Ψ_k. As in Hoff (2009a) we can employ non-uniform Bingham prior distributions for the eigenvectors of Ψ_k or we can model Ψ_k as a function of continuous covariates as in Yin et al. (2010) and Hoff and Niu (2012). Alternatively, we can summarize the estimated covariance matrices by thresholding entries of the precision matrix, Ψ_k^{-1} to visualize differences between groups using a graphical model (Meinshausen and Bühlmann, 2006). We can also incorporate sparsity to the estimated eigenvectors of the shared subspace to add in interpretation (Ročková and George, 2016, e.g). Finally, we can consider variants in which some eigenvectors are assumed to be identical across groups, whereas others are allowed to vary on the shared subspace. This can further improve estimation efficiency, particularly when the common eigenvectors are associated with the largest eigenvalues and differences appear in lower variance components (Cook and Forzani, 2008). Such an approach would further aid in identifying the relevant sub-subspace of variability that describes prominent differences between groups . The specifics of the problem at hand should dictate which extensions are appropriate, but the shared subspace assumption can be useful in a wide range of analyses, especially when the number of features is very large. A repository for the replication code is available on GitHub (Franks, 2016).

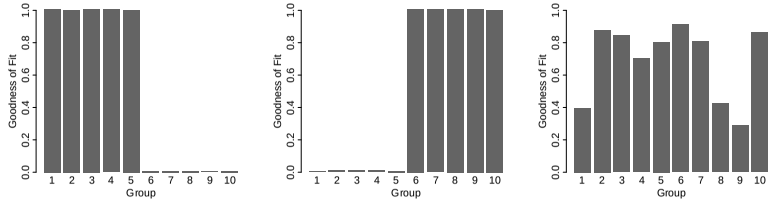
Acknowledgements

This work was partially supported by the Washington Research Foundation Fund for Innovation in Data-Intensive Discovery, the Moore/Sloan Data Science Environments Project at the University of Washington, NSF grant DMS-1505136, and NIH grant R03-CA211160-01. The authors are grateful to Dr. Daniel Promislow (Department of Pathology, University of Washington), Dr. Jessica Hoffman (University of Alabama at Birmingham) and Dr. Julie Josse (INRIA) for sharing data and ideas which contributed to framing of this paper.

Appendix A. Additional Misspecification Results

Following the simulation set up of 4.1 we generate data from 10 groups with $(\lambda_1, \lambda_2) = (250, 25)$, $p = 200$ and $\sigma_k^2 = 1$. In this section, we consider two model misspecification simulations. First, we consider data in which the first two eigenvectors for the first five groups share a two-dimensional subspace, and the eigenvectors for the last five groups share a different two-dimensional subspace. We then fit all ten groups assuming a two-dimensional shared subspace model. In Figure 8 we plot the goodness of fit metric for all ten groups for subspaces identified in different local modes of the likelihood. Specifically, we empirically identified three local modes: one mode identifies the shared subspace for the first give groups, the other mode corresponds to the shared subspace for the second five groups, and the third mode corresponds to subspace shares some commonalities across all 10 groups. This last mode is the one discovered by the eigen-based initialization strategy proposed in Section 4.1.

In the second simulation we generate the first two eigenvectors for the first seven groups from a common two dimensional subspace. The eigenvectors from the last three groups were generated uniformly at random from the $p - 2$ dimensional null space of the shared subspace. In repeated simulations with V initialized uniformly at random on the Stiefel manifold, the resulting we empirically discovered four modes. In Figure 9 we plot goodness of fit metrics for the 10 groups at these modes. The first mode corresponds to the shared subspace for the first 7 groups. The other three modes identify subspaces shared by two of the last three groups.



(a) (b) (c)

Figure 8: Goodness of fit for 10 groups using a two dimensional shared subspace model. In truth, the eigenvectors of the first five groups share a 2 dimensional subspace and the eigenvectors of the last five groups share a different 5 dimensional subspace. Empirically, by initializing the shared subspace uniformly at random, we found that there were three local modes. a) This mode corresponds to the shared subspace of the first five groups. b) This mode corresponds to the shared subspace of the second five groups. c) The third mode corresponds to a "shared subspace" across all groups. This is the mode discovered when using the eigen-based initialization strategy suggested in Section 4.1. Note that in truth variation in all ten groups could be captured using a 4 dimensional shared subspace.

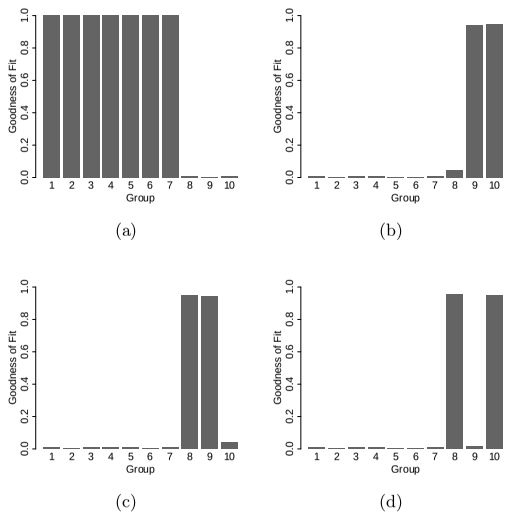


Figure 9: Goodness of fit for 10 group shared subspace model. The eigenvectors of the first seven groups share a 2 dimensional subspace and the eigenvectors of the last three groups were generated uniformly on the null space. Empirically, by initializing the shared subspace uniformaly at random, we found that there were three local modes. a) We discover the subspace shared by the first seven groups using the eigen-based initialization (Section 4.1). We also identify that the last three groups have small variance on this subspace, indicating that they do not share the subspace. b-d) Additional local (non-global) modes can be identified which in which 2 of the last three groups approximately share a two-dimensional subspace.

Appendix B. Run time results

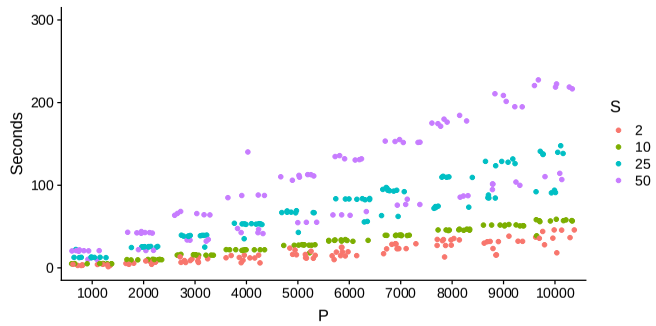
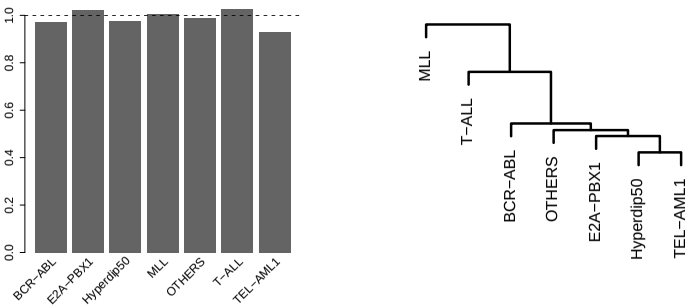


Figure 10: Run time results for subspace inference (Algorithm 1) as a function of the subspace dimension, S, and the number of features p. $s = 2, 10, 25, 50$ and $p = 1000, 2000, ...10000$. Points are jittered for visibility. In this simulation we assume $K = 5$ groups, $n_k = 50$ observations per group, $\sigma_k^2 = 1$ and the eigenvalues of ψ_k are samples from an Expo(1/4) (e.g. have mean 4). For each value of S and P we run subspace inference 10 times and plot the resulting run times. In each simulation we initialize the optimization routine uniformly at random on $\mathcal{V}_{p,s}$ to get a conservative estimate for run times. Using the intelligent initialization routine discussed in Section 4.1 typically increases time to convergence. Convergence time is on the order of minutes, even for relatively large values of s and p.

Appendix C. Addition Results From Leukemia Analysis



(a) $\gamma(Y_k : \hat{V}, \hat{\sigma_k}^2)$ (b) Hierarchical clustering

Figure 11: a) Goodness of shared subspace fit for each of the seven Leukemia groups. The inferred $s = 45$ dimensional subspace explains over 90% of estimated total variation in Σ_k in each of the seven groups. b) Complete-linkage hierarchical clustering of inferred projected data leukemia covariance matrices using Frobenius norm distance metric. The right sub-branches, which includes BCR-ABL, E2A-PBX1, hyperdiploid and TEL-AML1, are the B lineage leukemias of the seven types. T-All is a T lineage leukemia and MLL is a mixed lineage leukemia (Dang, 2012).

	Name	Q-value	Number of Genes
GO:0005751	mitochondrial respiratory chain complex IV	0.00	7
GO:0044388	small protein activating enzyme binding	0.01	7
GO:0022624	proteasome accessory complex	0.00	16
GO:0048025	negative regulation of nuclear mRNA splicing, via spliceosom	0.01	21
GO:0004298	threonine-type endopeptidase activity	0.01	19
GO:0010498	proteasomal protein catabolic process	0.01	26
GO:0006405	RNA export from nucleus	0.00	33
GO:0031124	mRNA 3'-end processing	0.00	36
GO:0030336	negative regulation of cell migration	0.01	22
GO:0038083	peptidyl-tyrosine autophosphorylation	0.01	20
GO:0043235	receptor complex	0.00	18
GO:0045766	positive regulation of angiogenesis	0.00	36
GO:0048661	positive regulation of smooth muscle cell proliferation	0.00	18
GO:0035690	cellular response to drug	0.00	27
GO:0060337	type I interferon-mediated signaling pathway	0.00	32
GO:0000786	nucleosome	0.00	27
GO:0004888	transmembrane signaling receptor activity	0.00	27
GO:0030183	B cell differentiation	0.00	30
GO:0030890	positive regulation of B cell proliferation	0.01	15
GO:0060333	interferon-gamma-mediated signaling pathway	0.00	35
GO:0030198	extracellular matrix organization	0.00	23
GO:0002053	positive regulation of mesenchymal cell proliferation	0.01	9
GO:0071345	cellular response to cytokine stimulus	0.01	19
GO:0007159	leukocyte cell-cell adhesion	0.00	16
GO:0034113	heterotypic cell-cell adhesion	0.00	10
GO:0042102	positive regulation of T cell proliferation	0.00	19
GO:0042605	peptide antigen binding	0.00	15
GO:0030658	transport vesicle membrane	0.00	13
GO:0071556	integral to lumenal side of endoplasmic reticulum membrane	0.01	17
GO:0042613	MHC class II protein complex	0.00	10
GO:0004896	cytokine receptor activity	0.00	9
GO:0005001	transmembrane receptor protein tyrosine phosphatase activity	0.01	7
GO:0030669	clathrin-coated endocytic vesicle membrane	0.00	10
GO:0042100	B cell proliferation	0.00	14
GO:0042742	defense response to bacterium	0.00	35
GO:0031668	cellular response to extracellular stimulus	0.00	13
GO:0001916	positive regulation of T cell mediated cytotoxicity	0.01	10
GO:0019731	antibacterial humoral response	0.00	19
GO:0001915	negative regulation of T cell mediated cytotoxicity	0.00	6
GO:0072562	blood microparticle	0.00	32
GO:0035456	response to interferon-beta	0.01	7
GO:0050829	defense response to Gram-negative bacterium	0.00	13
GO:0003823	antigen binding	0.00	24
GO:0071757	hexameric IgM immunoglobulin complex	0.00	6
GO:0006911	phagocytosis, engulfment	0.00	23
GO:0042834	peptidoglycan binding	0.00	7
GO:0071756	pentameric IgM immunoglobulin complex	0.00	7
GO:0006958	complement activation, classical pathway	0.00	19
GO:0006910	phagocytosis, recognition	0.00	16
GO:0042571	immunoglobulin complex, circulating	0.00	16
GO:0050871	positive regulation of B cell activation	0.00	16
GO:0003094	glomerular filtration	0.00	7
GO:0034987	immunoglobulin receptor binding	0.00	17
GO:0001895	retina homeostasis	0.00	11

Figure 12: Gene set enrichment analysis based on the magnitude of gene-loadings on the inferred 45 dimensional shared subspace (Section 6).

References

Jinho Baik and Jack W. Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6):1382–1408, 2006. ISSN 0047259X. doi: 10.1016/j.jmva.2005.08.003. URL `http://linkinghub.elsevier.com/retrieve/pii/S0047259X0500134X`.

Matthew Barker and William Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17(3):166–173, 2003. ISSN 08869383. doi: 10.1002/cem.785. URL `http://dx.doi.org/10.1002/cem.785`.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Similarity learning for provably accurate sparse linear classification. *arXiv preprint arXiv:1206.6476*, 2012.

Robert J Boik. Spectral models for covariance matrices. *Biometrika*, 89(1):159–182, 2002.

Simon Byrne and Mark Girolami. Geodesic Monte Carlo on embedded manifolds. *Scandinavian Journal of Statistics*, 40(4):825–845, 2013.

Yasuko Chikuse. *Statistics on special manifolds*, volume 174. Springer Science & Business Media, 2012.

Gene Ontology Consortium et al. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1):D258–D261, 2004.

R Dennis Cook. *Regression graphics: ideas for studying regressions through graphics*, volume 482. John Wiley & Sons, 2009.

R Dennis Cook and Liliana Forzani. Covariance reducing models: An alternative to spectral modelling of covariance matrices. *Biometrika*, 95(4):799–812, 2008.

R. Dennis Cook, Liliana Forzani, and Zhihua Su. A note on fast envelope estimation. *Journal of Multivariate Analysis*, 150:42–54, 2016. ISSN 10957243. doi: 10.1016/j.jmva.2016.05.006. URL `http://dx.doi.org/10.1016/j.jmva.2016.05.006`.

Patrick Danaher, Pei Wang, and Daniela M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 76(2):373–397, 2014. ISSN 13697412. doi: 10.1111/rssb.12033.

Chi V Dang. Myc on the path to cancer. *Cell*, 149(1):22–35, 2012.

David L Donoho, Matan Gavish, and Iain M Johnstone. Optimal shrinkage of eigenvalues in the spiked covariance model. *arXiv preprint arXiv:1311.0851*, 2013.

Morris L Eaton. Group invariance applications in statistics. In *Regional conference series in Probability and Statistics*, pages i–133. JSTOR, 1989.

Bradley Efron and Carl Morris. Stein's estimation rule and its competitors—an empirical bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973.

BK Flury. Two generalizations of the common principal component model. *Biometrika*, 74(1):59–69, 1987. ISSN 00063444. doi: 10.2307/2336021. URL `http://biomet.oxfordjournals.org/content/74/1/59.short`.

Alexander M. Franks. Replication code for shared subspace estimation. `https://github.com/afranks86/shared-subspace`, 2016.

Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *Information Theory, IEEE Transactions on*, 60(8):5040–5053, 2014.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 73:123–214, 2011. ISSN 13697412. doi: 10.1111/j.1467-9868.2010.00765.x.

Alejandro Gutierrez, Takaomi Sanda, Wenxue Ma, Jianhua Zhang, Ruta Grebliunaite, Suzanne Dahlberg, Donna Neuberg, Alexei Protopopov, Stuart S Winter, Richard S Larson, et al. Inactivation of lef1 in t-cell acute lymphoblastic leukemia. *Blood*, 115(14):2845–2851, 2010.

Graham Heimberg, Rajat Bhatnagar, Hana El-Samad, and Matt Thomson. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell Systems*, 2(4):239–250, 2016.

Peter D Hoff. A hierarchical eigenmodel for pooled covariance estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):971–992, 2009a.

Peter D Hoff. Simulation of the matrix Bingham–von Mises–Fisher distribution, with applications to multivariate and relational data. *Journal of Computational and Graphical Statistics*, 2009b.

Peter D Hoff and Xiaoyue Niu. A covariance regression model. *Statistica Sinica*, 22:729–753, 2012.

James Honaker, Gary King, and Matthew Blackwell. Amelia II: A program for missing data. *Journal of Statistical Software*, 45(7):1–47, 2011. URL `http://www.jstatsoft.org/v45/i07/`.

Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pages 295–327, 2001.

Takafumi Kanamori and Akiko Takeda. Non-convex optimization on stiefel manifold and applications to machine learning. In *International Conference on Neural Information Processing*, pages 109–116. Springer, 2012.

R Kang, D Tang, Y Yu, Z Wang, T Hu, H Wang, and L Cao. Wave1 regulates bcl-2 localization and phosphorylation in leukemia cells. *Leukemia*, 24(1):177, 2010.

CG Khatri and KV Mardia. The von Mises-Fisher matrix distribution in orientation statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 95–106, 1977.

J Kiefer and R Schwartz. Admissible Bayes character of T2-, R2-, and other fully invariant tests for classical multivariate normal problems. *The Annals of Mathematical Statistics*, pages 747–770, 1965.

Tõnu Kollo. Asymptotic inference based on eigenprojections of covariance and correlation matrices. In *Innovations in Multivariate Statistical Analysis*, pages 207–222. Springer, 2000.

Mary J Lindstrom and Douglas M Bates. Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data. *Journal of the American Statistical Association*, 83(404):1014–1022, 1988.

Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 339–348. AUAI Press, 2009.

Kantilal Varichand Mardia, John T Kent, and John M Bibby. *Multivariate analysis*. Academic press, 1980.

Matthew P McCormack, Lauren F Young, Sumitha Vasudevan, Carolyn A de Graaf, Rosalind Codrington, Terence H Rabbitts, Stephen M Jane, and David J Curtis. The lmo2 oncogene initiates leukemia in mice by inducing thymocyte self-renewal. *Science*, 327 (5967):879–883, 2010.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462, 2006.

Xavier Mestre. On the asymptotic behavior of the sample estimates of eigenvalues and eigenvectors of covariance matrices. *IEEE Transactions on Signal Processing*, 56(11): 5353–5368, 2008.

Debashis Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, pages 1617–1642, 2007.

Duc-Son Pham and Svetha Venkatesh. Robust learning of discriminative projection for multicategory classification on the stiefel manifold. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008.

Walter Reith, Salomé LeibundGut-Landmann, and Jean-Marc Waldburger. Regulation of mhc class ii gene expression by the class ii transactivator. *Nature Reviews Immunology*, 5(10):793, 2005.

Lisa M Rimsza, Robin A Roberts, Thomas P Miller, Joseph M Unger, Michael LeBlanc, Rita M Braziel, Dennis D Weisenberger, Wing C Chan, H Konrad Muller-Hermelink, Elaine S Jaffe, et al. Loss of mhc class ii gene and protein expression in diffuse large

b-cell lymphoma is related to decreased tumor immunosurveillance and poor patient survival regardless of other prognostic factors: a follow-up study from the leukemia and lymphoma molecular profiling project. *Blood*, 103(11):4251–4258, 2004.

Veronika Ročková and Edward I George. Fast bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association*, 111(516):1608–1622, 2016.

Daniel Romero, Dyonisius Dony Ariananda, Zhi Tian, and Geert Leus. Compressive covariance sensing: Structure-based compressive sensing beyond sparsity. *IEEE signal processing magazine*, 33(1):78–93, 2016.

James R Schott. Some tests for common principal component subspaces in several groups. *Biometrika*, 78(4):771–777, 1991.

James R Schott. Partial common principal component subspaces. *Biometrika*, 86(4):899–908, 1999.

John D Storey et al. The positive false discovery rate: a bayesian interpretation and the q-value. *The Annals of Statistics*, 31(6):2013–2035, 2003.

Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102 (43):15545–15550, 2005.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Raviteja Vemulapalli, Jaishanker K Pillai, and Rama Chellappa. Kernel learning for extrinsic classification of manifold features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1782–1789, 2013.

Fei Wang and Jimeng Sun. Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, 29(2):534–564, 2015.

Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130, 2001.

CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

Eng-Juh Yeoh, Mary E Ross, Sheila A Shurtleff, W Kent Williams, Divyen Patel, Rami Mahfouz, Fred G Behm, Susana C Raimondi, Mary V Relling, Anami Patel, et al. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer cell*, 1(2):133–143, 2002.

Jianxin Yin, Zhi Geng, Runze Li, and Hansheng Wang. Nonparametric covariance model. *Statistica Sinica*, 20:469, 2010.

Xiaotong Zhang, Xianchao Zhang, Han Liu, and Xinyue Liu. Multi-task multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3324–3338, 2016.

Journal of Machine Learning Research 20 (2019) 1-59 Submitted 9/18; Revised 10/19; Published 11/19

Optimization with Non-Differentiable Constraints with Applications to Fairness, Recall, Churn, and Other Goals

Andrew Cotter ACOTTER@GOOGLE.COM
Heinrich Jiang HEINRICHJ@GOOGLE.COM
Maya Gupta MAYAGUPTA@GOOGLE.COM
Serena Wang SERENAWANG@GOOGLE.COM
Taman Narayan TAMANN@GOOGLE.COM
Google Research
1600 Amphitheatre Pkwy
Mountain View, CA, USA
Seungil You SEUNGIL.YOU@GMAIL.COM
Kakao Mobility
Seongnam-si, Gyeonggi-do, South Korea
Karthik Sridharan SRIDHARAN@CS.CORNELL.EDU
Cornell University
Ithaca, NY, USA

Editor: Shivani Agarwal

Abstract

We show that many machine learning goals can be expressed as "rate constraints" on a model's predictions. We study the problem of training non-convex models subject to these rate constraints (or other non-convex or non-differentiable constraints). In the non-convex setting, the standard approach of Lagrange multipliers may fail. Furthermore, if the constraints are non-differentiable, then one cannot optimize the Lagrangian with gradient-based methods. To solve these issues, we introduce a new "proxy-Lagrangian" formulation. This leads to an algorithm that, assuming access to an optimization oracle, produces a stochastic classifier by playing a two-player non-zero-sum game solving for what we call a semi-coarse correlated equilibrium, which in turn corresponds to an approximately optimal and feasible solution to the constrained optimization problem. We then give a procedure that shrinks the randomized solution down to a mixture of at most $m+1$ deterministic solutions, given m constraints. This culminates in a procedure that can solve non-convex constrained optimization problems with possibly non-differentiable and non-convex constraints, and enjoys theoretical guarantees. We provide extensive experimental results covering a broad range of policy goals, including various fairness metrics, accuracy, coverage, recall, and churn.

Keywords: constrained optimization, non-convex, fairness, churn, swap regret, non-zero-sum game

1. Introduction

We seek to provide better ways to control machine learning to meet societal, legal, and practical goals, and to take advantage of different kinds of side information and intuition that practitioners may have about their machine learning problem. In this paper, we show that many real-world goals and

©2019 Andrew Cotter, Heinrich Jiang, Maya Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-616.html.

side information can be expressed as constraints on the model's prediction rates on different datasets, which we refer to as *rate constraints*, turning training into a constrained optimization problem. A simple example of a rate constraint is requiring a binary classifier to make positive predictions on at least 80% of examples. One can incorporate that rate constraint into training. That is, if h is a classifier parameterized by $\theta \in \Theta$, $\{(x_j, y_j)\}$ is a classifier training set with $j = 1, \ldots, N$, ℓ is the loss, and I is the usual indicator, the constrained optimization to minimize the empirical risk subject to this 80% positive rate constraint is:

$$\min_\theta \frac{1}{N} \sum_{j=1}^{N} \ell(h(x_j;\theta), y_j) \qquad (1)$$
$$\text{s.t. } \frac{1}{N} \sum_{j=1}^{N} I_{h(x_j;\theta)\geq 0} \;\geq\; 0.8.$$

1.1. The Broad Applicability Of Rate Constraints

One can express a surprisingly large set of real-world goals using rate constraints. Here we preview some categories of goals, with more details in Section 3.

Fairness: Many fairness goals can be expressed as rate constraints, including the popular fairness goal of *statistical parity*. For example, one can constrain a classifier so that its positive prediction rate for men and women differs by no more than ten percent. Other fairness goals that can be expressed as rate constraints are *equal opportunity* and *equal odds* (Hardt et al., 2016). In Section 3 we introduce some other fairness goals that we have encountered in real-world problems but have not previously seen in the machine learning literature, such as *no worse off*.

Performance Measures: Some standard performance metrics can be expressed as rate constraints, for example, one can lower-bound the recall, or constrain the model to have some minimal accuracy on specific slices of the data. Precision and win-loss ratio (WLR) compared to a baseline classifier can be expressed with rate constraints; however, there are some caveats about how satisfying constraints on these metrics will generalize to test samples (details below). AUC can be approximated as a set of rate constraints, using the approximation proposed in Eban et al. (2017).

Churn: Given a current classifier, the churn of a new classifier on a specific distribution of examples is the probability that the new classifier decision differs from the current classifier's decision (Cormier et al., 2016). Reducing classifier *churn* is important in many practical machine learning systems to improve overall system stability and to make changes easier to measure and test (Cormier et al., 2016). Churn can be expressed with rate constraints (Goh et al., 2016), thus one can constrain the churn of a new classifier to some desired level.

Multiple Training Datasets: Sometimes one has multiple labeled sets of varying quality and size. For example, one might have only a small set of data labeled by experts, but a large set of noisy training data. One can train the classifier to minimize errors on large noisy data, with a rate constraint that it must achieve at least a certain accuracy on the small expert-labeled dataset.

Unlabeled Datasets: Many of the rate constraints we discuss do not require labels, such as constraints on the positive rate of the classifier, or churn constraints. Training with these rate constraints enables one to take advantage of large unlabeled datasets, which are cheaper to obtain than labeled data.

1.2. Why Constrain? Why Not Penalize?

Rather than expressing each goal as a constraint during training, one could instead add an additive penalty to the loss. If there are multiple goals, one could use a linear combination of such additive penalties. However, the penalty approach requires the practitioner to determine the right weight for each penalty. In practice we find this gets difficult fast: there may be multiple constraints, possibly defined over multiple datasets, and the weights on the multiple penalties may interact with each other. In conclusion, it may be difficult to determine how much to weight each penalty.

We have found that specifying goals using constraints is in practice a cleaner and easier interface for practitioners. The key reason is that a constraint has an absolute meaning, making it possible for a practitioner to specify their goal as a constraint without regard for the presence of other constraints. For example, the meaning of a constraint that the classifier have 80% recall in India does not change if someone else adds other locale-specific constraints on the classifier. We also found that using hard constraints leads to a more understandable machine learning model because it is clearer what the model was trained to do, and it is clearer to measure and verify whether the training sufficiently achieved the practitioner's intent for each individual goal.

1.3. Training With Constraints:

Training with rate constraints poses some difficult challenges:

1. **Non-convex:** For nonlinear function classes, such as neural networks, the objective and constraint functions will be non-convex, *even with convex loss functions*.
2. **Non-differentiable:** Rate constraints are linear combinations of positive and negative classification rates. That is, they are made up of *indicator* functions (0-1 losses), and therefore have zero gradients almost everywhere.
3. **Data-dependent:** The constraints are data-dependent, so for large datasets it may be impractical to fully evaluate the constraints at every iteration—we'd prefer to work with *minibatches*.

While our motivating optimization problem is training with rate constraints, the analysis and algorithms we present will apply generally to constrained optimization problems of the form:

$$\begin{aligned} &\min_{\theta\in\Theta} g_0(\theta) \\ &\text{s.t. } \; g_i(\theta) \le 0 \text{ for } i = 1, \ldots, m, \end{aligned} \tag{2}$$

where the real-valued functions g_0 and the g_is may be non-convex. Furthermore, each of the m constraint functions g_i may be data-dependent, non-convex and even non-differentiable.

1.4. The Lagrangian May Have No Pure Equilibrium For Non-Convex Problems

A popular approach to constrained optimization problems of the form in Equation 2 is the method of Lagrange multipliers. Define the Lagrangian:

$$\mathcal{L}(\theta, \lambda) \triangleq g_0(\theta) + \sum_{i=1}^{m} \lambda_i g_i(\theta), \tag{3}$$

where λ is an m-dimensional non-negative vector of Lagrange multipliers. The method of Lagrange multipliers can be viewed as a two-player zero-sum game where one player minimizes Equation 3 with respect to the model parameters $\theta \in \Theta$, and the other player maximizes it with respect to the

Lagrange multipliers $\lambda \in \Lambda$. If the objective and constraints are all convex in θ, and the action spaces Λ and Θ are compact and convex, then this is a convex game, and it has a pure Nash equilibrium (von Neumann, 1928), i.e. there exists a θ for the first player and a λ for the second player such that neither player has the incentive to change their choice given the other player's choice. Furthermore, a pure Nash equilibrium (equivalently, a saddle point of the Lagrangian) gives us an optimal and feasible solution to the original constrained optimization problem specified in Equation 2.

On a constrained non-convex problem the Lagrangian might not even *have* a pure Nash equilibrium (see Figure 1 for an example). Hence, instead of converging, an iterative first-order algorithm may oscillate between different solutions, or it may converge to a locally optimal point—but not a Nash equilibrium—for which it is difficult to establish optimality and feasibility properties. However, if we allow each player to choose a *distribution* over their respective spaces Θ and Λ, and take the value of the Lagrangian to be the expectation over these distributions, then, under general conditions, the resulting *mixed* Nash equilibrium will exist.

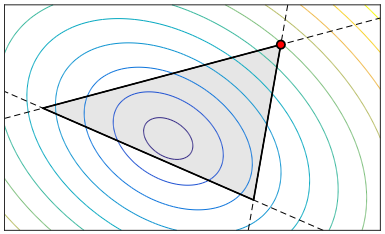


Figure 1: **Example of when no pure Nash equilibrium exists for the Lagrangian**: The plotted rectangular region is the domain Θ, the contours are those of the *strictly concave* minimization objective function g_0, and the shaded triangle is the feasible region determined by the three linear inequality constraints g_1, g_2, g_3. The red dot is the optimal feasible point. The Lagrangian $\mathcal{L}(\theta, \lambda)$ is strictly concave in θ for any choice of λ, so the optimal choice(s) for the θ-player will always lie on the four corners of the plotted rectangle. However, these points are infeasible, and therefore suboptimal for the λ-player, assuming that $\lambda \in \Lambda = \mathbb{R}^3_+$.

In this paper, we provide algorithms that approximately find such mixed equilibria, and we show that these correspond to nearly-feasible and nearly-optimal *stochastic* solutions to the original constrained optimization problem given in Equation 2. Such a stochastic solution is a random model: every time we classify an example x, we will independently sample a θ from the equilibrium distribution over Θ. Our guarantees will be expressed in terms of expectations with respect to this random θ.

1.5. The Lagrangian Is Impractical For Non-differentiable Constraints

Next, consider the issue of non-differentiable constraints (such as rate constraints). A major shortcoming of the Lagrangian is that one cannot use gradient-based methods to optimize non-differentiable constraints. One approach is to use the Lagrangian but replace non-differentiable constraints with differentiable surrogates (e.g. Davenport et al., 2010; Gasso et al., 2011; Eban et al., 2017). However changing the constraint functions may lead to solutions which either over-constrain or fail to satisfy the original constraints, as shown in Figure 2.

To address this, we introduce what we call the *proxy-Lagrangian* formulation, where the key idea is to relax the non-differentiable constraints *only when necessary*. Solving the proxy-Lagrangian poses technical challenges but leads to a number of interesting insights, and we provide algorithms

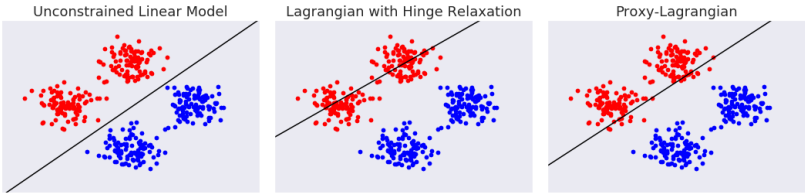


Figure 2: Mixture of Gaussians simulation: we generate 400 datapoints from a mixture of four Gaussians in two dimensions centered at $(0, 1), (-1, 0), (0, -1), (1, 0)$ in equal proportion each with covariance matrix $0.05\mathbf{I}$ where $\mathbf{I}$ denotes the identity matrix. The two top-left Gaussians are class red while the two bottom-right Gaussians are class blue. **Left**: The black line is the decision boundary for a linear model trained without any constraints. **Middle**: We trained this linear model subject to the rate constraint that the classifier must predict at least 55% of examples as blue. Here, we use the classical Lagrangian formulation and a hinge relaxation of the indicators in the constraints. Since the hinge relaxation is overly conservative, this classifier ends up overconstraining and actually predicts blue for 80.5% of the 400 examples in order to satisfy the *relaxed* constraint, at the cost of unnecessary loss of accuracy. **Right**: We trained this linear model subject to the same rate constraint that it must predict 55% of examples as blue, but this time we trained the model using the proposed proxy-Lagrangian formulation, and took the last iterate as the model. This model exactly enforces the requested 55% prediction rate for blue examples.

which attain solutions with optimality and feasibility guarantees on the original non-differentiable constraints.

Overall, we give an end-to-end recipe to provably (given access to an optimization oracle) and efficiently solve non-convex optimization problems with non-differentiable constraints, for which the solution will be a mixture of at most $m + 1$ deterministic solutions. In practice, we use SGD in place of the oracle. To our knowledge, this is the first time such a procedure has been found to provably solve such non-convex problems with such irregular constraints and return a sparse solution.

In addition, for those practical situations where a stochastic model is unappealing, we also experimentally consider algorithms that do produce deterministic models, though they do not come with guarantees.

1.6. Main Contributions And Organization

The main contributions of this paper are:

- We show that training with rate constraints can be used to address many real-world goals and capture realistic prior knowledge in the training.
- We give a new *proxy-Lagrangian* formulation for optimizing non-convex objectives with non-differentiable constraints.

- We provide an algorithm that outputs a $m + 1$ *sparse* stochastic classifier with theoretical guarantees, where m is the number of constraints.
- We show that our *proxy-Lagrangian* formulation can also be used to produce a deterministic classifier that may be more practical for some applications, but without guarantees.
- We provide an open-source Tensor Flow library that implements the presented algorithms.
- We experimentally demonstrate that the proposed optimization can be used to train classifiers with rate constraints, on both benchmark datasets and for real-world case studies.

Although our motivation and experimental focus is on the problem of training classifiers with rate constraints, our proposed proxy-Lagrangian formulation and theoretical results have broader application to other constrained optimization problems.

We next review related work. Then in Section 3 we detail many different goals that can be expressed with rate constraints. We then turn to the question of how to actually optimize with constraints, proposing new algorithms and theoretical results in Section 4. Section 5 presents a diverse set of experiments on benchmark and real datasets to illustrate the applicability of rate constraints and the proposed optimization. We close with a discussion of conclusions in Section 6 and open questions in Section 7.

2. Related Work

We begin by reviewing our own prior work which this paper builds upon, then other work that considers specific rate constraints, and then related work in constrained optimization.

2.1. Related Work On Specifying And Optimizing With Rate Constraints

Goh et al. (2016) showed that many different types of policy goals and side information can be expressed as constraints on the classifier's decisions on targeted datasets, and that one can then train the classifier to respect these constraints as part of the empirical risk minimization. Goh et al. (2016) referred to this class of constraints as *dataset constraints*, but we use the more precise term *rate constraints* to reflect that these constraints are functions of the classifier's positive and negative decision rates. In this paper, we will present many more goals and types of prior information that can be expressed as rate constraints that are useful in practice but have not previously appeared in the literature, such as *no lost benefit*, *not worse off*, and *loss-only churn*. Further, we provide more insight and analysis on how to use rate constraints in practice.

To optimize models with rate constraints, Goh et al. (2016) proposed a constrained optimization algorithm that was limited to linear classifiers, and used a new cutting-plane algorithm to iteratively upper-bound the ramp loss with a convex loss, then solved the resulting inner-loop minimizations using an SVM solver. While amenable to theoretical analysis, this strategy is a bit slow and difficult to scale to more than a handful of constraints. In contrast, in this paper we show we can effectively and efficiently train *nonlinear* classifiers with rate constraints using the more popular and scalable approach of stochastic gradients.

An important classic special case of rate constraints is Neyman-Pearson classification, which constrains the false positive rate (Scott and Nowak, 2005). Davenport et al. (2010) optimized Neyman-Pearson support vector machines with hinge loss relaxations using coordinate descent. Gasso et al. (2011) relaxed the indicators to the ramp loss (both in the objective and constraints). Eban et al. (2017) optimized the model parameters and Lagrangian multiplier using stochastic gradients with a hinge approximation for the indicators in the empirical loss and constraints, and took the last training

iterate as their solution. We compare to that optimization strategy in our experiments (listed as *Hinge Last* in the result tables).

Mann and McCallum (2007) and follow-on work (Bellare et al., 2009; Mann and McCallum, 2010) optimized probabilistic models with constraints in order to incorporate side information about the prior priors on class labels, which in the context of a binary classifier is a special case of rate constraints that we call a *coverage constraint*. They note their strategy could also be applied to any constraints that can be written as an expectation over a score on the random (X, Y) samples. They incorporated this side information as an additive regularizer and penalized the relative entropy between the given priors and estimated multi-class logistic regression models. They noted their approximation for the indicator could lead to degenerate solutions, which they indirectly addressed by additional regularization.

Agarwal et al. (2018) recently addressed training classifiers with fairness constraints that can be expressed as rate constraints. Like this work, their proposed algorithm is based on the two-player game perspective. Unlike this paper, they assume a *zero-sum* game, which works because they also assume oracle solvers for the two players, side-stepping the practical issues of dealing with the non-differentible non-convex indicators in the constraints, which is the focus of our algorithmic and theoretical contributions. Similar to this work, they output a stochastic classifier, but do not provide the sparse $m + 1$ solution that we present in this work. They also consider a deterministic solution, which they produce by searching over a grid of values for λ for the best λ. They noted in their experimental section that the resulting deterministic solution was generally as good as their stochastic solutions on test data for those experiments they tried it on. As they note, a grid-search over λ is less ideal as the number of constraints grows.

Some other work in training with fairness constraints has used weaker constraints or relaxed them immediately to weaker constraints such as correlation, e.g. Zafar et al. (2015, 2017). Another set of work in fair classification only corrects a model post-training by optimizing additive group-specific bias parameters, e.g. Hardt et al. (2016) and Woodworth et al. (2017). Donini et al. (2018) studies optimization of fairness constraints for kernel methods by formulating the fairness constraints as orthogonality constraints. The goal of *equal accuracy* has also been explored recently in Buolamwini and Gebru (2018) in the context of matching the accuracies of male/female classifiers across race.

2.2. Other Types Of Constraints On Machine Learned Models

We focus on rate constraints in this paper, which have tend to have the following properties. First, because rate constraints depend on $f(x)$, they generally depend on all the model parameters θ. Second, rate constraints are usually relatively expensive to compute. Third, we do not generally expect to have a very large number of rate constraints.

These qualities are different from the popular constrained machine learning problem of *shape constraints*, which requires that the model is restricted to functions with a certain shape such as monotonic functions (e.g. Barlow et al. (1972); Groeneboom and Jongbloed (2014); Gupta et al. (2016); Canini et al. (2016); Luss and Rosset (2017); You et al. (2017); Bonakdarpour et al. (2018)), or other shapes (e.g. Chetverikov et al. (2018); Pya and Wood (2015); Chen and Samworth (2016); Gupta et al. (2018); Cotter et al. (2019b)).

In contrast to rate constraints, shape constraints generally require adding many sparse, cheap-to-evaluate constraints. For example, for isotonic regression on N training examples, there are $O(N)$ constraints, and each is a function of only two model parameters (Barlow et al., 1972). Similarly,

in some of the experiments of Canini et al. (2016), the models are trained with around 100,000 constraints, but each constraint only touches two model parameters. Problems like that with many cheap sparse constraints can be well-handled by stochastic sampling of the constraints, as in Cotter et al. (2016), but that strategy is less well-suited to rate constraints because there tend to be fewer constraints, and each constraint is expensive to evaluate.

Another type of constrained machine learning aims to constrain the model parameters to obey known physical limits on the learned system (e.g. Long et al. (2018); Stewart and Ermon (2017)). These constraints generally do not take the form of rate constraints, but such constrained machine learning models may also benefit from the presented algorithms and theory.

Some fairness constraints are more complicated than can be handled as rate constraints. For example, Heidari et al. (2018) give a new individual fairness notion which ensures the expected utility an individual receives as a result of the model.

2.3. Related Work In Constrained Optimization As A Two Player Game

Our strategy for treating non-differentiable problems as a non-zero sum two-player game using a proxy Lagrangian formulation was first presented in our conference paper, Cotter et al. (2019c). This journal paper extends that work with more discussion of how a broad set of goals can be expressed as rate constraints, much more comprehensive experiments, some additional theoretical perspectives, and more advice for practitioners.

Our constrained optimization algorithms and analyses build on the long history of treating constrained optimization as a two-player game: Arora et al. (2012) surveys some such work, and there are several more recent examples (e.g. Agarwal et al. (2018); Kearns et al. (2018); Narasimhan (2018)). We extend that prior work in three key ways. First, to handle non-differentiable constraints, we propose a new proxy-Lagrangian *non-zero-sum* formulation, whereas prior work formulates the optimization as a zero-sum game. Second, we introduce a shrinking procedure that significantly simplifies a "T-stochastic" solution (i.e. a stochastic classifier supported on all T iterates) to a sparse "m-stochastic" solution (a stochastic classifier supported on only $m + 1$ iterates, where m is the number of constraints). Third, we consider a broader set of problems than prior work.

Our contributions also apply to robust optimization problems of the form:

$$\min_{\theta \in \Theta} \max_{i \in [m]} g_i(\theta),$$

where each $g_i : \Theta \to \mathbb{R}$. The most related work addressing non-convex robust optimization is Chen et al. (2017). Like both Agarwal et al. (2018) and this paper, Chen et al. (2017) (i) model the problem as a two-player game where one player chooses a mixture of objective functions, and the other player minimizes the loss of the mixture, and (ii) they find a *distribution* over solutions rather than a pure equilibrium. These similarities are unsurprising in light of the fact that robust optimization can be reformulated as constrained optimization via the introduction of a slack variable:

$$\begin{aligned} \min_{\theta \in \Theta, \xi \in \mathbb{R}} \; & \xi \\ \text{s.t. } & \xi \geq g_i(\theta) \quad \text{for all } i \in 1, \ldots, m. \end{aligned} \tag{4}$$

Correspondingly, one can transform a robust problem to a constrained one at the cost of an extra bisection search (e.g. Christiano et al., 2011; Rakhlin and Sridharan, 2013). As this relationship

suggests, our main contributions can be adapted to the robust optimization setting. In particular: (i) our proposed shrinking procedure can be applied to Equation 4 to yield a distribution over only $m+1$ solutions, and (ii) one could perform robust optimization over non-differentiable (even discontinuous) losses using "proxy objectives," just as we use proxy constraints.

2.4. Other Strategies For Constrained Optimization

There are other strategies for constrained optimization, each of which we argue is not well-suited to the problem of training classifiers with rate constraints.

The computational complexity of rate constraints makes them generally unattractive to try to optimize with approaches that require projections, such as projected SGD, or optimization of constrained subproblems, such as Frank-Wolfe (Hazan and Kale, 2012; Jaggi, 2013; Garber and Hazan, 2013)).

Another strategy for constrained optimization is to penalize violations of the constraints (e.g. Arora et al., 2012; Rakhlin and Sridharan, 2013; Mahdavi et al., 2012; Cotter et al., 2016; Yang et al., 2017), for example by adding $\gamma \max_{i\in[m]} \max\{0, g_i(\theta)\}$ to the objective, where $\gamma \in \mathbb{R}_+$ is a hyperparameter, and optimizing the resulting problem using a first-order method. This strategy is not ideal for rate constraints for two reasons. First, rate constraints are non-(semi)differentiable. Second, each rate constraint is data-dependent, so evaluating g_i, or even determining whether it is positive (as is necessary for such methods, due to the max with 0), requires enumerating over the entire constraint dataset, making this incompatible with the use with a computationally-cheap stochastic gradient optimizer.

3. What Are Rate Constraints Good For?

In this section, we first present the mathematical formulation of rate constraints and the resulting constrained empirical risk minimization training. Table 1 provides a handy reference for key notation. Then, we provide a list of metrics that can be expressed as rate constraints in Table 2, and detail in the following subsections how these rate constraints can be used to impose a broad set of policy goals and take advantage of side information.

Given a classifier $h : \mathcal{X} \times \Theta \to \mathbb{R}$ (where $\mathcal{X}$ is the feature space and Θ is the parameter space) a dataset D, and using I to denote the usual indicator, define the classifier's positive classification rate on D as $p^+(D;\theta)$, and the classifier's negative classification rate on D as $p^-(D;\theta)$, where

$$p^+(D;\theta) \triangleq \frac{1}{|D|}\sum_{x\in D} I_{h(x;\theta)\geq 0} \quad \text{and} \quad p^-(D;\theta) \triangleq \frac{1}{|D|}\sum_{x\in D} I_{h(x;\theta)<0}. \tag{5}$$

We call a constraint a *rate constraint* if it can be expressed in terms of a non-negative linear combination of positive classification rates $p^+(D_k;\theta)$ and negative classification rates $p^-(D_k;\theta)$ over different datasets $\{D_k\}$. That is, a *rate constraint* is a constraint expressible as:

$$\sum_{k=1}^{K} \alpha_k p^+(D_k;\theta) + \beta_k p^-(D_k;\theta) \leq \kappa, \tag{6}$$

for some $\alpha \in \mathbb{R}^+$, $\beta \in \mathbb{R}^+$, and $\kappa \in \mathbb{R}^+$.

Table 2 shows how different choices of scalars $\alpha_k, \beta_k, \kappa \in \mathbb{R}$ and datasets $\{D_k\}$ correspond to different standard performance metrics like accuracy and recall. One can add m rate constraints to

Table 1: Basic Notation

D	Set of examples
$D[*]$	Subset of D that satisfies expression *, e.g. $D[x \in \text{male}]$ is the subset of D of male examples, $D[y=1]$ is the subset of D whose label is 1, etc.
$\tilde{h} : \mathcal{X} \times \Theta \to \mathbb{R}$	A given classifier to which a candidate classifier h can be compared
$p^+(D;\theta) \in [0,1]$	Proportion of D classified positive
$p^-(D;\theta) \in [0,1]$	Proportion of D classified negative
$c^+(D;\theta) \in \mathbb{N}$	Count of D classified positive: $c^+(D;\theta) = \lvert D\rvert\, p^+(D;\theta)$
$c^-(D;\theta) \in \mathbb{N}$	Count of D classified negative: $c^-(D;\theta) = \lvert D\rvert\, p^-(D;\theta)$

the standard structural risk minimization to train a classifier with parameters $\theta \in \Theta$ on train dataset D_0, producing the constrained empirical risk minimization:

$$\min_{\theta\in\Theta} \frac{1}{|D_0|}\sum_{(x,y)\in D_0} \ell(h(x;\theta),y) + R(\theta) \tag{7}$$
$$\text{s.t.} \sum_{k=1}^{K_i} \alpha_{ik} p^+(D_{ik};\theta) + \beta_{ik} p^-(D_{ik};\theta) \leq \kappa_i \quad \text{for } i = 1,\ldots,m,$$

where $\alpha_{ik}, \beta_{ik} \in \mathbb{R}$, D_{ik} is the kth dataset for the ith constraint, K_i is the number of datasets used to specify the ith constraint, and $\kappa_i \in \mathbb{R}$.

For some applications it is notationally more convenient to drop the normalization on the rate constraints and express the constraint in terms of counts, let $c^+(D;\theta)$ and $c^-(D;\theta)$ denote the count of the positive and negative classifications:

$$c^+(D;\theta) \triangleq \sum_{x\in D} I_{h(x;\theta)\geq 0} \quad \text{and} \quad c^-(D;\theta) \triangleq \sum_{x\in D} I_{h(x;\theta)<0}. \tag{8}$$

Throughout this work, we focus on inequality constraints, for lower-bounding or upper-bounding some rate. Equality constraints can be imposed by using both a lower-bound and upper-bound inequality constraint, though we suggest doing so with some margin between the lower and upper bound to make the optimization problem easier.

In the rest of this section we show how different rate constraints can be used to impose various policy goals or capture side information. A key insight is that one can add constraints just on specific groups or subsets of the dataset by the choice of the datasets used for a constraint, which makes this approach particularly useful for fairness goals or other slice-specific metrics that are measured in terms of statistics on different datasets (see Table 3 and further details below).

3.1. Coverage Constraints

Coverage is the proportion of classifications that are positive: $p^+(D;\theta)$ (a variant is *negative coverage* $p^-(D;\theta)$). For example, if a company wants to train a classifier to identify the best 10% of all customers to receive a printed catalog, then one could train the classifier with a 10% coverage constraint.

Table 2: Examples of Metrics Expressed With Rates And Notation From Table 1

Recall	$p^+(D[y=1];\theta)$
Precision	$c^+(D[y=1];\theta)/c^+(D;\theta)$
Accuracy	$(c^+(D[y=1];\theta)+c^-(D[y=-1];\theta)/\lvert D\rvert$
AUCROC	$\lim_{L,J\to\infty}\frac{1}{L}\sum_{\ell=1}^{L}\ \max_{j\in[J]:p^+_{\alpha_j}(D[y=-1];\theta)\le\frac{\ell}{L}}\ p^+_{\alpha_j}(D[y=1];\theta)$
Wins Compared to $\tilde{h}$	$c^+(D[\tilde{h}=-1,y=1];\theta)+c^-(D[\tilde{h}=1,y=-1];\theta)$
Losses Compared to $\tilde{h}$	$c^+(D[\tilde{h}=-1,y=-1];\theta)+c^-(D[\tilde{h}=1,y=1];\theta)$
Win Loss Ratio (WLR)	Wins Compared to $\tilde{h}$ / Losses Compared to $\tilde{h}$
Churn	$(c^+(D[\tilde{h}=-1];\theta)+c^-(D[\tilde{h}=1];\theta))/\lvert D\rvert$
Loss-only Churn	$(c^+(D[\tilde{h}=-1,y=-1];\theta)+c^-(D[\tilde{h}=1,y=1];\theta)/\lvert D[\tilde{h}=y]\rvert$

Coverage constraints can also be used to capture prior knowledge in the training. For example, if training a model to classify Americans as male or female, one can regularize the classifier by incorporating the prior knowledge that 51% of examples should be predicted to be female, by using a 51% coverage constraint.

Using slice-specific coverage constraints can capture more side information. For example, for the American male/female classifier, in addition to the overall coverage constraint of 51%, one could also add constraints capturing prior information about state sex distributions, such as constraining 51.5% of examples from New York to be classified as women, but constraining only 47.6% of examples from Alaska to be classified as women.

A key advantage of coverage constraints is that they do not require labeled examples. This enables one to train on labeled training examples from a convenient distribution (such as actively-sampled examples), but then add a coverage constraint to ensure the classifier is optimized to positively classify the desired proportion of positive classifications on a larger unlabeled dataset drawn *i.i.d.* from the true underlying distribution. This usage of a coverage constraint forms a semi-supervised regularization of the classifier.

Another good use case for coverage constraints is to help make a controlled comparison of two model structures. For example, suppose one has a model type A (say, a kernel SVM), and wonders if an alternative B (say, a DNN) is better, where A makes positive predictions on 40% of test examples, while B appears to be more accurate, but only predicts the positive class for 35% of test examples. If precision errors are worse than recall errors, we cannot be sure that B is better than A. We can try to quantify the misclassification costs of a false negative vs. a false positive, but that may be difficult to agree upon. It would be simpler to compare B to A at the same coverage as A, or at some other relevant coverage. Coverage-matching B to A can be done by tuning the decision threshold of B post-training, but including the coverage constraint in the training can help B learn to be a better classifier when tested at the desired coverage.

3.2. Constraints On Accuracy, Recall, Precision, AUC

As shown in Table 2, classifier accuracy can be expressed in terms of rates, and thus accuracy on auxiliary datasets or slices of the training data can be constrained with rate constraints.

Recall, defined as TP / (TP + FN), can be written as $p^+(D[y=1];\theta)$, and thus one can put a lower-bound constraint on recall $p^+(D[y=1];\theta)>\kappa$ for the user's choice of $\kappa\in[0,1]$. For

example, one may wish to train a classifier that awards free meals to poor students, but constrain it to obtain at least 95% recall.

Precision can be expressed in rates as $c^+(D[y=1];\theta)/c^+(D;\theta)$, and thus to get precision of at least κ, one can add a rate constraint:

$$c^+(D[y=1];\theta)-\kappa c^+(D;\theta)\ge 0. \tag{9}$$

If (9) holds, then mathematically the precision is lower-bounded by κ on the dataset D. However, since the expectation of a ratio does not equal the ratio of the expected numerator and denominator, analyzing how well the empirical constraint holding generalizes to new *i.i.d.* samples is not straightforward, and violating the constraint (9) by some $\epsilon>0$ does not translate directly into a precision error of ϵ.

The ROC AUC (Area under the ROC curve) can be approximated using a rate constraint, as in Eban et al. (2017). The ROC curve is obtained by plotting the true positive rate (TPR) vs. the false positive rate (FPR). First, slice up the FPR-axis into L slices (to approximate the required Riemann integral). Then for the ℓth slice, consider J different decision thresholds and choose the threshold that maximizes TPR and satisfies the ℓth slice FPR bound ℓ/L, and then the averaged maximum precision across the L FPR slices is bounded:

$$\frac{1}{L}\sum_{\ell=1}^{L}\ \max_{j\in[J]:p^+_{\alpha_j}(D[y=-1];\theta)\le\frac{\ell}{L}}\ p^+_{\alpha_j}(D[y=1];\theta)\ge\kappa. \tag{10}$$

where $p^+_\alpha(D;\theta)\triangleq\frac{1}{\lvert D\rvert}\sum_{x\in D}I_{h(x;\theta)\ge\alpha}$, $c^+_\alpha(D;\theta)\triangleq\sum_{x\in D}I_{h(x;\theta)\ge\alpha}$, and $\alpha_j:=\frac{2j-1}{2J}$ for $j\in[J]$. In particular, $p^+_0\equiv p^+$. Taking $L\to\infty, J\to\infty$ will have the expression on the LHS of (10) converge to the exact ROC AUC.

3.3. Churn And Win Loss Ratio Constraints

In practice, a new classifier is often being trained to replace an existing classifier $\tilde{h}$, in which case the new classifier may be evaluated using metrics that compare the new classifier to the old classifier $\tilde{h}$.

One common metric to compare two classifiers is the win-loss ratio (WLR), which is the number of times the new classifier is right and the old classifier is wrong, divided by the number of times the new classifier is wrong and the old classifier is right.

A WLR constraint can be expressed in terms of rates as in Table 2, where we use $D[\tilde{h}=-1]$ to denote the subset of D that is labeled negatively by the classifier $\tilde{h}$, and $D[\tilde{h}=-1,y=1]$ to denote the subset of D of whose training label y is 1, so that $c^+(D[\tilde{h}=-1,y=1];\theta)$ is the number of wins of the new classifier over h, and so on. Re-arranging terms, one can constrain for WLR using a rate constraint:

$$\begin{aligned}&c^+(D[\tilde{h}=-1,y=1];\theta)+c^-(D[\tilde{h}=1,y=-1];\theta)\\&-\kappa(c^+(D[\tilde{h}=-1,y=-1];\theta)+c^-(D[\tilde{h}=1,y=1];\theta))\ge 0,\end{aligned} \tag{11}$$

where $\kappa\in\mathbb{R}^+$ is the lower-bound on the WLR. However, enforcing this constraint on a training dataset D does not necessarily guarantee that the desired WLR threshold will be achieved on fresh *i.i.d.* samples, not only due to the potential for overfitting, but also because the expectation of a ratio does not equal the ratio of expectations.

WLR constraints on different slices of the data can ensure that a new classifier's gains are not coming at the expense of an important subset of examples. (See also our discussion of *no worse off* and *no lost benefits* for related fairness constraints).

In practice, labeling data can be expensive, so it is common to test a new classifier by drawing a fresh test set including only examples on which the new classifier and previous classifier $\tilde{h}$ disagree. We refer to this as a *fresh test*. Fresh tests reduce the chance of overfitting to a fixed test set that is used over many model iterations. Fresh tests only incur labeling costs for those examples whose decisions have changed. Thus with a fresh test, higher WLR means fewer fresh test examples have to be rated to statistically significantly confirm that the new classifier is better than the old classifier $\tilde{h}$ (Cormier et al., 2016).

Note that two new classifiers can have the same accuracy but different WLRs compared to the previous classifier h. The proportion of a dataset D on which the classifications change when one changes classifiers is called *churn* (Cormier et al., 2016; Goh et al., 2016). When using a fresh test, the labeling costs scale linearly with the churn (and the size of the test set D). High churn also causes more instability for follow-on systems, and can confuse users. Goh et al. (2016) proposed explicitly constraining the churn, which can be directly expressed as the rate constraint:

$$c^+(D[\tilde{h} = -1]; \theta) + c^-(D[\tilde{h} = 1]; \theta) \leq \kappa|D|, \tag{12}$$

where $\kappa \in [0, 1]$ is the proportion of D on which the classification decision is allowed to change. Constraining churn on different slices of the data, with tighter and looser constraints, can be useful. For example, if the classifier is to be used worldwide, but labeling is more expensive in Norway than in Vietnam, or if there is known to be less headroom to improve on examples from Norway, then it could be beneficial to constrain the churn more tightly on examples from Norway, but more loosely on examples from Vietnam.

Of course, constraining churn too tightly limits the potential accuracy gains. Thus we also propose considering loss-only churn constraints, which only penalizes new losses:

$$c^+(D[\tilde{h} = -1, y = -1]; \theta) + c^-(D[\tilde{h} = 1, y = 1]; \theta) \leq \kappa|D|, \tag{13}$$

where $\kappa \in [0, 1]$ is the proportion of D whose classification decision is allowed to flip.

One disadvantage of constraining loss-only churn is it requires labeled examples, whereas churn constraints can be more conveniently used on a dataset of unlabeled examples.

3.4. Fairness Goals And Other Group-Specific Goals

An important use case for rate constraints is enforcing metrics for different groups or categories of examples. For example, ensuring that a classifier for identifying family-friendly videos works roughly equally well at filtering different types of objectionable adult content. Rate constraints can be used to enforce a broad set of such group-specific goals, as detailed in Table 3, where k indexes the K different groups of interest.

A special case of group-specific goals are those that are designed to improve some fairness metric. In these cases the groups are usually defined as different groups of people, e.g. different genders or age brackets. Table 3 shows that many of the fairness goals already studied in the machine learning literature can be expressed with rate constraints. With that said, fairness is a complex moral and policy problem, and depending on the context and application, different formulations may be appropriate, with some such formulations not being group-based at all.

Many fairness goals are designed for applications where positive classification endows a benefit, such as being awarded a loan, a job, or a free meal. For example, the goal of *statistical parity* reflects that a bank might be legally required to give loans at equal rates to different groups to alleviate disparity (Bocian et al., 2008), that is, the classifier is required to provide equal positive rates of classification across groups (see e.g. Zafar et al. (2015); Fish et al. (2016); Hardt et al. (2016); Goh et al. (2016)). Statistical parity is also known as *demographic parity* (Hardt et al., 2016), and *equal coverage* (Goh et al., 2016). Notice that a statistical parity constraint ignores the labels of the training data. We introduce the related goal of *minimum coverage*, which enforces some minimal benefit rate for each group.

Next we detail some fairness goals we find useful in practice, but have not seen previously formalized in the literature. The goal of *accurate coverage* requires the classifier to give free meals to each group to match that group's positive training label rate. This goal ignores whether the individual predictions are accurate, but tries to ensure that each group overall receives a rate of benefits that it is labeled as deserving.

No lost benefits: which requires a model to classify examples positively from each group at least as often as the classifier $\tilde{h}$ that it is replacing. *No lost benefits* is a type of churn goal (see Sec. 3.3) that is measured for the whole group (rather than for individual decisions).

The other fairness goals in Table 3 depend on the training labels. Our *not worse off* fairness goal requires that accuracy with the new classifier for each group is not worse than it was under the classifier $\tilde{h}$ that it would replace. For example, suppose someone invents a new driving test, and shows that it is more accurate than the current written driving test at diagnosing whether illiterate people are safe drivers, then *not worse off* requires that the new driving test not reduce accuracy compared to the old test for other groups, e.g. senior citizens and teenagers. *Not worse off* is a label-dependent group-specific churn goal.

Minimum accuracy, which requires that every group experience some pre-set level of accuracy. *Minimum accuracy* ensures that no group is left behind, but respects that for some problems some groups may be much easier to classify than other groups. For such problems, constraining accuracy to be similar across groups can lead to degenerate solutions, as the only way to make all groups have equal metrics may be to produce a degenerate classifier.

Equal opportunity and *equal odds* (Hardt et al., 2016) also rely on the training labels. For example, *equal opportunity* requires that *if* a classifier awards free meals (positive classification) to half of the east-side children who are labeled as deserving free meals, then it should also award free meals to half of the west-side children who are labeled as deserving free meals. Notice that *equal opportunity* imposes no conditions whatsoever on the negatively-labeled examples (in this case, those students who are not labeled as deserving of free meals). In contrast, the fairness goal of *equal odds* requires both the true positive rate and the false positive rate to be the same for all groups. Variations are *equal accuracy*, *equal recall*, *equal precision*, and so on, all of which aim to make the classifier equally good at some metric for different groups.

Fairness goals that depend on the training labels are most compelling when the training examples and labels are believed to have been fairly sampled and labeled. These goals are less compelling when the training dataset is not entirely trusted, or thought to be misaligned with the policy goals, a situation referred to as *negative legacy* (Kamishima et al., 2012). There are many reasons why training data and labels may not be fully trustworthy. Selection biases on the training examples and raters labeling them can affect the training distribution in ways that negatively impact certain groups. Further, training labels can have unbiased noise due to all sorts of cognitive biases. One key

Table 3: Group-Specific and Fairness Goals Expressed As Rate Constraints for Groups $k = 1, \ldots, K$

Statistical Parity	$p^+(D_k;\theta) = p^+(D;\theta)\ \forall k$
Minimum Coverage	$p^+(D_k;\theta) \geq \kappa\ \forall k$ and user-specified $\kappa \in [0, 1]$
No Lost Benefits	$p^+(D_k;\theta) \geq \lvert(D_k[\tilde{h} = 1])\rvert/\lvert D_k\rvert\ \forall k$
Accurate Coverage	$p^+(D_k;\theta) = \lvert D_k[y = 1]\rvert/\lvert D_k\rvert\ \forall k$
Equal Opportunity	$p^+(D_k[y = 1];\theta) = p^+(D[y = 1];\theta)\ \forall k$
Equal Odds	$p^+(D_k[y = 1];\theta) = p^+(D[y = 1];\theta)\ \forall k$
	and $p^+(D_k[y = -1];\theta) = p^+(D[y = -1];\theta)\ \forall k$
Equal Accuracy	$(c^+(D_k[y = 1]) + c^-(D_k[y = -1]))/\lvert D_k\rvert$
	$= (c^+(D[y = 1]) + c^-(D[y = -1]))/\lvert D\rvert\ \forall k$
Minimum Accuracy	$(c^+(D_k[y = 1]) + c^-(D_k[y = -1]))/\lvert D_k\rvert \geq \kappa\ \forall k$
Not Worse Off	$(c^+(D_k[y = 1]) + c^-(D_k[y = -1]))/\lvert D_k\rvert \geq \lvert(D_k[y = \tilde{h}])\rvert/\lvert D_k\rvert\ \forall k$

problem is raters are generally more accurate at labeling examples they are more familiar to them, for example, consider a situation where adult raters are asked to label whether children will find a video interesting.

3.5. Egregious Examples And Steering Examples

Another use of rate constraints is to constrain the performance on auxiliary labeled datasets to control the classifier. For example, Goh et al. (2016) proposed constraining the classifier for high accuracy on a small set of particularly *egregious examples* that should definitely not be mislabeled. Egregious examples act as an *integrated unit test*: as the classifier trains, it actively tests itself to make sure it satisfies the constraint on the egregious examples and is able to correct the training accordingly.

Another practical example of using an auxiliary labeled dataset is what we term *steering examples*, which we define as a set of labeled examples that are more accurately labeled than the training set. For example, one may have access to a large but noisy training set of clicks on news articles. However, an article might be clicked either because it was relevant news, or because it had a catchy headline. We can try to steer the classifier to focus on the relevant news articles by providing a smaller but expertly-labeled curated set of examples that mark catchy headlines as negative, and then constrain the classifier to achieve some reasonable minimal accuracy on these steering examples. Such a rate constraint will steer the classifier to be consistent with the steering examples, and helping it disregard any badly labeled training examples. A second example is a classifier that attempts to determine whether an online store should advertise to a given customer. Suppose that there is a large dataset of training examples with the positive label, "customer clicked advertisement and *visited* website", but a relatively small set of examples where the positive label is, "customer clicked advertisement and made a *purchase*." It may be better to train on the large set of "visited" examples due to its much larger size and coverage, but also to constrain at least some specified accuracy on the smaller "purchase" examples in order to steer the classifier towards prioritizing clicks that lead to purchases.

3.6. Decision Rule Priors

Machine learning practitioners often have prior knowledge about a classification problem that they can communicate as a decision rule on a tiny set of features. For example, "Don't recommend a book to a user if it is in a language they haven't purchased before." We propose a simple way of incorporating such decision rule priors into the structural risk minimization problem by creating an auxiliary dataset consisting of many unlabeled samples, labeling it with the desired decision rule, and adding an accuracy rate constraint on that auxiliary data set.

Such decision rule priors can act as regularizers against noisy and poorly-sampled training examples, and can produce a classifier that is more interpretable because it is known to (probably) obey the given decision rules (like all rate constraints, this depends on whether one constrains with slack or not, and also on exactly how well the satisfied constraint generalizes, which depends on whether the examples observed at evaluation time will truly be drawn *i.i.d.* from the same distribution as the training data, as well as the function class, and how hard the constraint is to satisfy).

This proposal is similar to Bayesian Rule Lists (BRL) (Letham et al., 2015) in that a decision rule (or set of decision rules) is given a priori to training the model. However, BRL training takes as input a large set of decision rules and outputs a posterior over the rules, rather than incorporating a decision rule into a structural risk minimization problem.

3.7. How To Best Specify Rate Constraints

For any rate constraint, we recommend allowing some slack in order to find a feasible solution. For example, statistical parity could be written as a constraint with an *additive slack* of κ like this:

$$p^+(D;\theta) - p^+(D_k;\theta) \leq \kappa,$$

or instead with *multiplicative slack* of κ like this:

$$p^+(D;\theta) - \kappa p^+(D_k;\theta) \geq 0,$$

where k is the index of the kth subset of interest.

Our experience is that additive slack tends to be more likely to produce reasonable solutions than multiplicative slack for many constraints. The danger to watch out for is whether the constraint is specified in a way that encourages the training to satisfy the constraint in a suboptimal way. For example, if one constrains the false positive rate of each group to be no worse than 125% of the overall false positive rate (multiplicative slack), then the training is incentivized to increase the overall false positive rate because that loosens the constraint further (the same effect occurs for additive slack, but the effect tends to be larger for multiplicative slack).

Constraints can also be expressed *pairwise* between groups, instead of against the global rate:

$$p^+(D_j;\theta) - p^+(D_k;\theta) \leq \kappa,$$

for all j, k pairs. Our experience is that constraints that involve a larger dataset are generally preferable, and that the smaller the dataset generally the greater the risk of overfitting the constraint or ending up with a degenerate solution to achieve feasibility.

Equality constraints can be expressed by using both a lower-bound and an upper-bound inequality constraint. In practice, we suggest allowing some slack between the lower and upper bounds in order to increase the number of feasible solutions, thereby making stochastic gradient optimization more stable.

4. Optimizing With Constraints

For nonlinear function classes, training a classifier with rate constraints as per Equation 7 is a non-convex optimization over a non-convex constraint set. In this section we provide new theoretical insights and algorithms to optimize general non-convex problems with non-convex constraints, then demonstrate our algorithmic proposals work well in practice with multiple real-world constraints in Section 5. We first outline our two main contributions for this section below.

A Minimal Stochastic Solution: Algorithms that solve non-convex constrained optimization problems based on regret minimization, which includes our approach as well as previous work (e.g. Chen et al., 2017; Agarwal et al., 2018) will output a distribution over θs which has discrete support over T different θ (resulting from the T different epochs of the training algorithm), requiring us to store and sample from T different models. In practice, large T may be problematic to store and serve. Surprisingly, we prove that there always exists an equilibrium that has *sparse support* on at most $m+1$ choices of model parameters, where m is the number of constraints. We use this result to provide a new practical algorithm to shrink the approximated equilibrium down to a nearly-optimal and nearly-feasible solution supported on at most $m+1$ models, which is guaranteed to be at least as good as the original stochastic classifier supported on T models.

Handling Non-Differentiable Constraints: A key issue for Equation 7 is the non-differentiability of the constraints due to the indicators in the rate constraints. To handle this, in Section 4.3, we introduce a new formulation we call the *proxy-Lagrangian* that changes the standard two-player *zero-sum* game to a two-player *non-zero-sum* game, which presents new challenges to analysis. In fact, solving a Nash equilibrium is PPAD-complete in the non-zero-sum setting (Chen and Deng, 2006). We prove that a particular game theory solution concept, which we call *semi-coarse correlated equilibrium*, results in a stochastic classifier that is feasible and optimal. This is surprising because the semi-coarse correlated equilibrium is a weaker notion of equilibrium than Nash equilibrium. We go on to provide a novel algorithm that converges to such an equilibrium. To our knowledge, we give the first reduction to this particular solution concept and the first practical use for it, which may be of independent interest. Interestingly, the θ-player needs to only minimize the usual external regret, but the λ-player must minimize the *swap regret* (Blum and Mansour, 2007), a stronger notion of regret. While the resulting distribution is supported on (a possibly large number of) (θ, λ) pairs, applying the same "shrinking" procedure as before yields a distribution over only $m+1$ of the θs that is at least as good as the original.

In Section 4.1, we handle the optimization of the zero-sum Lagrangian game with an oracle-based algorithm and introduce our proposed "shrinking" procedure. Then, in Section 4.2 we introduce the concept of proxy constraints, describe how it is useful to handle non-differentiable constraints, and formulate the non-zero-sum modification of the Lagrangian, which we call the proxy-Lagrangian. Section 4.3 describes the equilibrium required out of this non-zero-sum game so that it will correspond to an approximately feasible and optimal solution to the constrained optimization problem. Section 4.4 gives an oracle-based procedure for solving for such an equilibrium. Section 4.5 gives a more practical stochastic gradient-based optimizer along with improved guarantees in the convex setting. Finally, Section 4.6 shows that the "shrinking" procedure holds for the non-zero-sum solution as well.

Algorithm 1 Optimizes the Lagrangian formulation (Equation 3) in the non-convex setting via the use of an approximate Bayesian optimization oracle $\mathcal{O}_\rho$ (Definition 1) for the θ-player. The parameter R is the radius of the Lagrange multiplier space $\Lambda := \left\{\lambda \in \mathbb{R}_+^m : \|\lambda\|_1 \le R\right\}$, and the function Π_Λ projects its argument onto Λ w.r.t. the Euclidean norm.

OracleLagrangian $(R \in \mathbb{R}_+, \mathcal{L} : \Theta \times \Lambda \to \mathbb{R}, \mathcal{O}_\rho : (\Theta \to \mathbb{R}) \to \Theta, T \in \mathbb{N}, \eta_\lambda \in \mathbb{R}_+)$:

Initialize $\lambda^{(1)} = 0$

For $t \in [T]$:

  Let $\theta^{(t)} = \mathcal{O}_\rho\left(\mathcal{L}\left(\cdot, \lambda^{(t)}\right)\right)$ *// Oracle optimization*

  Let $\Delta_\lambda^{(t)}$ be a gradient of $\mathcal{L}\left(\theta^{(t)}, \lambda^{(t)}\right)$ w.r.t. λ

  Update $\lambda^{(t+1)} = \Pi_\Lambda\left(\lambda^{(t)} + \eta_\lambda \Delta_\lambda^{(t)}\right)$ *// Projected gradient update*

Return $\theta^{(1)}, \ldots, \theta^{(T)}$ and $\lambda^{(1)}, \ldots, \lambda^{(T)}$

4.1. Lagrangian Optimization In The Non-convex Setting

We start by assuming an approximate Bayesian optimization oracle (defined in Section 4.1.1), which enables us to use the Lagrangian formulation and not relax the non-convex and/or non-differentiable constraints. This setting is a slight generalization of that presented in Agarwal et al. (2018). Algorithm 1 solves for a stochastic solution to the non-convex constrained optimization problem. It proceeds by playing the following for T rounds: the model parameter player plays best-response (that is, the θ which minimizes the Lagrangian given the last choice of Lagrange multipliers), and the Lagrange multiplier player plays a regret minimizing strategy (here we use projected SGD).

Our first contribution of this section (in Section 4.1.2) is showing that the resulting stochastic classifier is provably approximately feasible and optimal in expectation. This extends the fair classification work of Agarwal et al. (2018) to our slightly more general setting. Our second contribution comes in Section 4.1.4: we will show how the support of the stochastic solution can be efficiently "shrunk" to one that is *at least as good*, but is supported on only $m+1$ solutions and is shown to also have a considerable gain empirically.

4.1.1. ORACLE FOR UNCONSTRAINED NON-CONVEX MINIMIZATION (ADDITIVE APPROXIMATION)

Algorithm 1, like the robust optimization algorithm of Chen et al. (2017), requires an *oracle* for performing approximate non-convex minimization. The oracle is a simply a function that takes in a function f and returns an approximate minimizer of f.

Definition 1 *A ρ-approximate Bayesian optimization oracle is a function $\mathcal{O}_\rho : (\Theta \to \mathbb{R}) \to \Theta$ such that:*

$$f\left(\mathcal{O}_\rho\left(f\right)\right) \le \inf_{\theta^* \in \Theta} f\left(\theta^*\right) + \rho$$

for any $f : \Theta \to \mathbb{R}$ that can be written as a nonnegative linear combination of the objective and constraint functions $g_0, g_1, \ldots, g_m$.

The oracle will be used by the θ-player, and the λ-player will use projected gradient ascent. We note that this is a standard assumption in order to obtain theoretical guarantees. (e.g. see Chen et al. (2017), which uses a multiplicative instead of additive approximation).

4.1.2. APPROXIMATE MIXED NASH EQUILIBRIUM

We characterize the relationship between an approximate Nash equilibrium of the Lagrangian game, and a nearly-optimal nearly-feasible solution to the non-convex constrained problem (Equation 2) in our theorem below. This theorem has a few differences from the more typical equivalence between Nash equilibria and optimal feasible solutions in the convex setting. First, it characterizes *mixed* equilibria, in that uniformly sampling from the sequences $\theta^{(t)}$ and $\lambda^{(t)}$ can be interpreted as defining distributions over Θ and Λ. Second, we require compact domains in order to prove convergence rates (below) so Λ is taken to consist only of sets of Lagrange multipliers with bounded 1-norm. In Appendix A, this is generalized to p-norms..

As a reminder, a mixed Nash equilibrium to a two-player game is a pair of distributions over the strategy spaces, one distribution assigned to each player such that neither player can improve their expected payoff (over these distributions) by changing their distribution given that the other player uses their assigned distribution. An *ϵ-approximate* mixed Nash equilibrium is where neither player can improve by more than ϵ by changing their assigned distribution given that the other player uses their assigned distribution.

Finally, as a consequence of the compact domains, the feasibility guarantee of Theorem 2 only holds if the Lagrange multipliers are, on average, smaller than the maximum 1-norm radius R. Thankfully, as is shown by the final result of Theorem 2, if there exists a point satisfying the constraints with some margin $\gamma > 0$, then there will exist an R that is large enough to guarantee feasibility to within $O(\epsilon)$.

Theorem 2 *Define:*

$$\Lambda \triangleq \{\lambda \in \mathbb{R}_+^m : \|\lambda\|_1 \leq R\} \tag{14}$$

and let $\theta^{(1)}, \ldots, \theta^{(T)} \in \Theta$ and $\lambda^{(1)}, \ldots, \lambda^{(T)} \in \Lambda$ be sequences of parameter vectors and Lagrange multipliers that comprise an approximate mixed Nash equilibrium, i.e.:

$$\max_{\lambda^* \in \Lambda} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}\left(\theta^{(t)}, \lambda^*\right) - \inf_{\theta^* \in \Theta} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}\left(\theta^*, \lambda^{(t)}\right) \leq \epsilon.$$

Define $\bar{\theta}$ as a random variable for which $\bar{\theta} = \theta^{(t)}$ with probability $1/T$, and let $\bar{\lambda} \triangleq \left(\sum_{t=1}^{T} \lambda^{(t)}\right)/T$. Then $\bar{\theta}$ is nearly-optimal and nearly-feasible in expectation:

$$\mathbb{E}_{\bar{\theta}}\left[g_0\left(\bar{\theta}\right)\right] \leq \inf_{\theta^* \in \Theta : \forall i, g_i(\theta^*) \leq 0} g_0\left(\theta^*\right) + \epsilon \quad \text{and} \quad \max_{i \in [m]} \mathbb{E}_{\bar{\theta}}\left[g_i\left(\bar{\theta}\right)\right] \leq \frac{\epsilon}{R - \|\bar{\lambda}\|_1}.$$

Additionally, if there exists a $\theta' \in \Theta$ that satisfies all of the constraints with margin γ (i.e. $g_i(\theta') \leq -\gamma$ for all $i \in [m]$), then:

$$\|\bar{\lambda}\|_1 \leq \frac{\epsilon + B_{g_0}}{\gamma},$$

where $B_{g_0} \geq \sup_{\theta \in \Theta} g_0(\theta) - \inf_{\theta \in \Theta} g_0(\theta)$ is a bound on the range of the objective function g_0.

Proof This is a special case of Theorem 9 and Lemma 10 in Appendix A. ■

4.1.3. CONVERGENCE OF ALGORITHM 1

Algorithm 1's convergence rate is given by the following lemma:

Lemma 3 ***(Algorithm 1)*** *Suppose that Λ and R are as in Theorem 2, and define $B_\Delta \geq \max_{t \in [T]} \left\|\Delta_\lambda^{(t)}\right\|_2$. If we run Algorithm 1 with the step size $\eta_\lambda := R/B_\Delta\sqrt{2T}$, then the result satisfies Theorem 2 for:*

$$\epsilon = \rho + RB_\Delta\sqrt{\frac{2}{T}},$$

where ρ is the error associated with the oracle $\mathcal{O}_\rho$.

Combined with Theorem 2, we therefore have that if R is sufficiently large, then Algorithm 1 will converge to a distribution over Θ that is, in expectation, $O(\rho)$-far from being optimal and feasible at a $O(1/\sqrt{T})$ rate, where ρ is defined in Section 4.1.1.

4.1.4. SHRINKING THE STOCHASTIC SOLUTION

A disadvantage of Algorithm 1 is that it results in a mixture of T solutions, which may be large and thus undesirable in practice. However, we can show that much smaller mixed Nash equilibria exist:

Lemma 4 *If Θ is a compact Hausdorff space, Λ is compact, and the objective and constraint functions $g_0, g_1, \ldots, g_m$ are continuous, then the Lagrangian game (Equation 3) has a mixed Nash equilibrium pair (θ, λ) where θ is a random variable supported on at most $m+1$ elements of Θ, and λ is non-random.*

Proof Follows from Theorem 13 in Appendix B. ■

We do not content ourselves with merely having shown the existence of such an equilibrium. Fortunately, we can re-formulate the problem of finding the optimal ϵ-feasible mixture of the $\theta^{(t)}$s as a linear program (LP) that can be solved to *shrink* the support set to $m+1$ solutions. We must first evaluate the objective and constraint functions for every $\theta^{(t)}$, yielding a T-dimensional vector of objective function values, and m such vectors of constraint function evaluations, which are then used to specify the LP.

Lemma 5 *Let $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(T)} \in \Theta$ be a sequence of T "candidate solutions" of Equation 2. Define $\vec{g_0}, \vec{g_i} \in \mathbb{R}^T$ such that $(\vec{g_0})_t = g_0\left(\theta^{(t)}\right)$ and $(\vec{g_i})_t = g_i\left(\theta^{(t)}\right)$ for $i \in [m]$, and consider the linear program:*

$$\begin{aligned} \min_{p \in \Delta^T} \quad & \langle p, \vec{g_0} \rangle \\ \text{s.t.} \quad & \langle p, \vec{g_i} \rangle \leq \epsilon \text{for all } i \in 1, \ldots, m, \end{aligned}$$

where Δ^T is the T-dimensional simplex. Then every vertex p^* of the feasible region—in particular an optimal one—has at most $m^* + 1 \leq m + 1$ nonzero elements, where m^* is the number of active $\langle p^*, \vec{g_i} \rangle \leq \epsilon$ constraints.

This lemma suggests a two-phase approach to actually finding the $m+1$ stochastic solution. In the first phase, apply Algorithm 1, yielding a sequence of iterates for which the uniform distribution over the $\theta^{(t)}$s is approximately feasible and optimal. Then apply the procedure of Lemma 5 to find the *best* distribution over these iterates, which in particular can be no worse than the uniform distribution, and is supported on at most $m+1$ iterates.

4.2. Proxy Constraints And A Non-Zero Sum Game

Most real-world machine learning implementations use first-order methods (even on non-convex problems, e.g. DNNs); however, to use these methods, one must have gradients which are unavailable for rate constraints due to the indicators. Since the constraint functions are piecewise-constant, their gradients are zero almost everywhere, and a gradient-based method cannot be expected to succeed. In general, for constrained optimization problems in the form of Equation 2, non-differentiable constraints arise naturally when one wishes to constrain counts or proportions.

The obvious solution is to use a surrogate for each indicator. For example, we might consider replacing the indicators defining a rate with sigmoids, and then optimizing the Lagrangian. This solves the differentiability problem, but introduces a new one: a (mixed) Nash equilibrium would correspond to a solution satisfying the sigmoid-relaxed constraints, instead of the *actual* constraints. Interestingly, it turns out that we can seek to satisfy the original un-relaxed constraints, even while using a surrogate. Our proposal is motivated by the observation that, while differentiating the Lagrangian (Equation 3) w.r.t. θ requires differentiating the constraint functions $g_i(\theta)$, to differentiate it w.r.t. λ we only need to *evaluate* them. Hence, a surrogate is only necessary for the θ-player; the λ-player can continue to use the original constraint functions.

We refer to a surrogate that is used by only one of the two players as a "proxy", and introduce the notion of "proxy constraints" by taking $\tilde{g}_i(\theta)$ to be a sufficiently-smooth upper bound on $g_i(\theta)$ for $i \in [m]$, and formulating two functions that we call "proxy-Lagrangians":

$$\mathcal{L}_\theta(\theta,\lambda) \triangleq \lambda_1 g_0(\theta) + \sum_{i=1}^{m} \lambda_{i+1}\tilde{g}_i(\theta) \tag{15}$$

$$\mathcal{L}_\lambda(\theta,\lambda) \triangleq \sum_{i=1}^{m} \lambda_{i+1} g_i(\theta),$$

where we restrict Λ to be the $(m+1)$-dimensional simplex Δ^{m+1}. The θ-player seeks to minimize $\mathcal{L}_\theta(\theta,\lambda)$, while the λ-player seeks to maximize $\mathcal{L}_\lambda(\theta,\lambda)$. Notice that the $\tilde{g}_i$s are *only* used by the θ-player. Intuitively, the λ-player chooses how much to weigh the proxy constraint functions, but—and this is the key to our proposal—does so in such a way as to satisfy the *original* constraints.

Viewed as a two-player game, what we have changed is that now the θ and λ players each have their own payoff functions $\mathcal{L}_\theta(\theta,\lambda)$ and $\mathcal{L}_\lambda(\theta,\lambda)$ respectively, making the game *non-zero sum*. Finding a Nash equilibrium of a non-zero-sum game is much more difficult than for a zero-sum game—in fact, it's PPAD-complete even in the finite setting (Chen and Deng, 2006). We will present a procedure which approximates a *weaker* type of equilibrium: instead of converging to a Nash equilibrium, it converges to a new solution concept, which we call a *semi-coarse correlated equilibrium*. Despite being weaker than a Nash equilibrium, we show that it still corresponds to a nearly-optimal and nearly-feasible solution to constrained optimization in expectation.

The proxy-Lagrangian formulation leads to a tighter approximation than the popular approach of using a surrogate for *both* players, as has been previously proposed, e.g. for Neyman-Pearson

classification (Davenport et al., 2010; Gasso et al., 2011), and AUC optimization (Eban et al., 2017). Those proposals optimize a simpler zero-sum game, but one that is a worse reflection of the true goal. In the experimental section, we will provide evidence that the proposed proxy-Lagrangian formulation can provide higher accuracy while still satisfying the constraints. This is especially important when the rate constraints express real-world restrictions on how the learned model is permitted to behave.

4.3. Proxy-Lagrangian Equilibrium

For the proxy-Lagrangian game (Equation 15), we cannot expect to find a Nash equilibrium, at least not efficiently, since it is non-zero-sum. However, the analogous result to Theorem 2 requires a *weaker* type of equilibrium: a joint distribution over Θ and Λ w.r.t. which the θ-player can only make a negligible improvement compared to the best constant strategy, and the λ-player compared to the best action-swapping strategy; this is a type of Φ-correlated equilibrium (Rakhlin et al., 2011). We call this semi-coarse-correlated equilibrium because it exhibits properties of a coarse-correlated equilibrium for one player (θ-player) and that of correlated equilibrium for the other player (λ-player). In a coarse-correlated equilibrium, each player is assigned a distribution over their respective strategy spaces where these distributions can be mutually dependent and no player can improve their payoff by switching to any *fixed* strategy given that the other players use their assigned distributions. In a correlated equilibrium, again each player is assigned a distribution over their respective strategy spaces where these distributions can be mutually dependent, but no player can improve their payoff by changing their assigned distribution given that the other players use their assigned distributions.

We present our theorem showing the achievability of this type of equilibrium, then we present Algorithm 2 to satisfy the theorem.

Theorem 6 *Define $\mathcal{M}$ as the set of all left-stochastic $(m+1)\times(m+1)$ matrices, $\Lambda \triangleq \Delta^{m+1}$ as the $(m+1)$-dimensional simplex, and assume that each $\tilde{g}_i$ upper bounds the corresponding g_i. Let $\theta^{(1)},\ldots,\theta^{(T)} \in \Theta$ and $\lambda^{(1)},\ldots,\lambda^{(T)} \in \Lambda$ be sequences satisfying:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_\theta\left(\theta^{(t)},\lambda^{(t)}\right) - \inf_{\theta^*\in\Theta}\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_\theta\left(\theta^*,\lambda^{(t)}\right) \le \epsilon_\theta$$

$$\max_{M^*\in\mathcal{M}}\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_\lambda\left(\theta^{(t)},M^*\lambda^{(t)}\right) - \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_\lambda\left(\theta^{(t)},\lambda^{(t)}\right) \le \epsilon_\lambda.$$

Define $\bar{\theta}$ as a random variable for which $\bar{\theta} = \theta^{(t)}$ with probability $\lambda_1^{(t)}/\sum_{s=1}^{T}\lambda_1^{(s)}$, and let $\bar{\lambda} \triangleq \left(\sum_{t=1}^{T}\lambda^{(t)}\right)/T$. Then $\bar{\theta}$ is nearly-optimal and nearly-feasible in expectation:

$$\mathbb{E}_{\bar{\theta}}\left[g_0\left(\bar{\theta}\right)\right] \le \inf_{\theta^*\in\Theta:\forall i.\tilde{g}_i(\theta^*)\le 0} g_0\left(\theta^*\right) + \frac{\epsilon_\theta+\epsilon_\lambda}{\bar{\lambda}_1} \tag{16}$$

and,

$$\max_{i\in[m]}\mathbb{E}_{\bar{\theta}}\left[g_i\left(\bar{\theta}\right)\right] \le \frac{\epsilon_\lambda}{\bar{\lambda}_1}. \tag{17}$$

Additionally, if there exists a $\theta' \in \Theta$ that satisfies all of the proxy constraints with margin γ (i.e. $\tilde{g}_i(\theta') \le -\gamma$ for all $i \in [m]$), then:

$$\bar{\lambda}_1 \ge \frac{\gamma-\epsilon_\theta-\epsilon_\lambda}{\gamma+B_{g_0}},$$

Algorithm 2 Optimizes the proxy-Lagrangian formulation (Equation 15) in the non-convex setting via the use of an approximate Bayesian optimization oracle $\mathcal{O}_\rho$ (Definition 1, but with $\tilde{g}_i$s instead of g_is in the linear combination defining f) for the θ-player, with the λ-player minimizing swap regret. The $\pi(M)$ operation on line 3 results in a stationary distribution of M (i.e. a $\lambda \in \Lambda$ such that $M\lambda = \lambda$, which can be derived from the top eigenvector).

OracleProxyLagrangian $\left(\mathcal{L}_\theta, \mathcal{L}_\lambda : \Theta \times \Delta^{m+1} \to \mathbb{R}, \mathcal{O}_\rho : (\Theta \to \mathbb{R}) \to \Theta, T \in \mathbb{N}, \eta_\lambda \in \mathbb{R}_+\right)$:

1 Initialize $M^{(1)} \in \mathbb{R}^{(m+1)\times(m+1)}$ with $M_{i,j} = 1/(m+1)$
2 For $t \in [T]$:
3 Let $\lambda^{(t)} = \pi\left(M^{(t)}\right)$ *// Stationary distribution of* $M^{(t)}$
4 Let $\theta^{(t)} = \mathcal{O}_\rho\left(\mathcal{L}_\theta\left(\cdot, \lambda^{(t)}\right)\right)$ *// Oracle optimization*
5 Let $\Delta_\lambda^{(t)}$ be a gradient of $\mathcal{L}_\lambda\left(\theta^{(t)}, \lambda^{(t)}\right)$ w.r.t. λ
6 Update $\tilde{M}^{(t+1)} = M^{(t)} \odot . \exp\left(\eta_\lambda \Delta_\lambda^{(t)} \left(\lambda^{(t)}\right)^T\right)$ *// $\odot$ and $. \exp$ are element-wise*
7 Project $M_{:,i}^{(t+1)} = \tilde{M}_{:,i}^{(t+1)} / \left\|\tilde{M}_{:,i}^{(t+1)}\right\|_1$ for $i \in [m+1]$ *// Column-wise projection*
8 Return $\theta^{(1)}, \ldots, \theta^{(T)}$ and $\lambda^{(1)}, \ldots, \lambda^{(T)}$

where $B_{g_0} \geq \sup_{\theta\in\Theta} g_0(\theta) - \inf_{\theta\in\Theta} g_0(\theta)$ is a bound on the range of the objective function g_0.

Proof This is a special case of Theorem 11 and Lemma 12 in Appendix A. ■

Notice that Equation 17 guarantees feasibility w.r.t. the original constraints, while Equation 16 shows that the solution minimizes the objective approximately as well as the best solution that's feasible w.r.t. the *proxy* constraints. Hence, the guarantee for minimizing the objective is no better than what we would have obtained if we took $g_i \triangleq \tilde{g}_i$ for all $i \in [m]$, and optimized the Lagrangian as in Section 4.1. However, because the feasible region w.r.t. the original constraints is larger (perhaps significantly so) than that w.r.t. the proxy constraints, the proxy-Lagrangian approach has more "room" to find a better solution in practice (this is demonstrated in the experiments).

One key difference between this result and Theorem 2 is that the R bound on λ is now gone. Instead, its role, and that of $\left\|\bar{\lambda}\right\|_1$, is played by the first coordinate of $\bar{\lambda}$. Inspection of Equation 15 reveals that, if one or more of the constraints are violated, then the λ-player would prefer the corresponding entries in λ to be higher, which in turn causes λ_1 to become closer to 0 from our procedures. Likewise, if they are satisfied (with some margin), then it would prefer the entries after the first in λ to be 0 which causes λ_1 to be one in our procedures. In other words, the first coordinate of $\lambda^{(t)}$ encodes the λ-player's belief about the feasibility of $\theta^{(t)}$, for which reason $\theta^{(t)}$ is weighted by $\lambda_1^{(t)}$ in the density defining $\bar{\theta}$.

4.4. Proxy-Lagrangian Optimization Algorithm

To optimize the proxy-Lagrangian formulation, we present Algorithm 2, which is motivated by the observation that, while Theorem 6 only requires that the $\theta^{(t)}$ sequence suffer low external regret w.r.t. $\mathcal{L}_\theta\left(\cdot, \lambda^{(t)}\right)$, the condition on the $\lambda^{(t)}$ sequence is stronger, requiring it to suffer low *swap regret* (Blum and Mansour, 2007) w.r.t. $\mathcal{L}_\lambda\left(\theta^{(t)}, \cdot\right)$.

Algorithm 3 Optimizes the proxy-Lagrangian formulation (Equation 15) in the convex setting, with the θ-player minimizing external regret, and the λ-player minimizing swap regret. The $\pi(M)$ operation on line 4 outputs the stationary distribution of M (that is, a $\lambda \in \Lambda$ such that $M\lambda = \lambda$) which can be derived from the top eigenvector. The function Π_Θ projects its argument onto Θ w.r.t. the Euclidean norm.

StochasticProxyLagrangian $\left(\mathcal{L}_\theta, \mathcal{L}_\lambda : \Theta \times \Delta^{m+1} \to \mathbb{R}, T \in \mathbb{N}, \eta_\theta, \eta_\lambda \in \mathbb{R}_+\right)$:

1 Initialize $\theta^{(1)} = 0$ *// Assumes* $0 \in \Theta$
2 Initialize $M^{(1)} \in \mathbb{R}^{(m+1)\times(m+1)}$ with $M_{i,j} = 1/(m+1)$
3 For $t \in [T]$:
4 Let $\lambda^{(t)} = \pi\left(M^{(t)}\right)$ *// Stationary distribution of* $M^{(t)}$
5 Let $\check{\Delta}_\theta^{(t)}$ be a stochastic subgradient of $\mathcal{L}_\theta\left(\theta^{(t)}, \lambda^{(t)}\right)$ w.r.t. θ
6 Let $\Delta_\lambda^{(t)}$ be a stochastic gradient of $\mathcal{L}_\lambda\left(\theta^{(t)}, \lambda^{(t)}\right)$ w.r.t. λ
7 Update $\theta^{(t+1)} = \Pi_\Theta\left(\theta^{(t)} - \eta_\theta \check{\Delta}_\theta^{(t)}\right)$ *// Projected SGD update*
8 Update $\tilde{M}^{(t+1)} = M^{(t)} \odot . \exp\left(\eta_\lambda \Delta_\lambda^{(t)} \left(\lambda^{(t)}\right)^T\right)$ *// $\odot$ and $. \exp$ are element-wise*
9 Project $M_{:,i}^{(t+1)} = \tilde{M}_{:,i}^{(t+1)} / \left\|\tilde{M}_{:,i}^{(t+1)}\right\|_1$ for $i \in [m+1]$ *// Column-wise projection*
10 Return $\theta^{(1)}, \ldots, \theta^{(T)}$ and $\lambda^{(1)}, \ldots, \lambda^{(T)}$

Hence, the θ-player uses the oracle to minimize external regret, while the λ-player uses a swap-regret minimization algorithm of the type proposed by Gordon et al. (2008), yielding the convergence guarantee:

Lemma 7 ***(Algorithm 2)*** *Suppose that $\mathcal{M}$ and Λ are as in Theorem 6, and define the upper bound $B_\Delta \geq \max_{t\in[T]} \left\|\Delta_\lambda^{(t)}\right\|_\infty$.*

If we run Algorithm 2 with the step size $\eta_\lambda := \sqrt{(m+1)\ln(m+1)/TB_\Delta^2}$, then the result satisfies satisfies the conditions of Theorem 6 for:

$$\epsilon_\theta = \rho$$
$$\epsilon_\lambda = 2B_\Delta \sqrt{\frac{(m+1)\ln(m+1)}{T}},$$

where ρ is the error associated with the oracle $\mathcal{O}_\rho$.

4.5. Practical Stochastic Proxy-Lagrangian Algorithm

Algorithm 3 is designed for the setting of a convex objective, thus we can safely use SGD for the θ-updates instead of the oracle and enjoy a more practical procedure. We stress that this is a considerable improvement over previous Lagrangian methods in the convex setting, as they require both the loss and constraints to be convex in order to attain optimality and feasibility guarantees. Here, while we assume convexity of the objective and proxy-constraints, the *original* constraints do not need to be convex, but we are still able to prove similar guarantees.

Lemma 8 ***(Algorithm 3)*** *Suppose that Θ is a compact convex set, $\mathcal{M}$ and Λ are as in Theorem 6, and that the objective and proxy constraint functions $g_0, \tilde{g}_1, \ldots, \tilde{g}_m$ are convex (but not $g_1, \ldots, g_m$). Define the three upper bounds $B_\Theta \geq \max_{\theta \in \Theta} \|\theta\|_2$, $B_{\check{\Delta}} \geq \max_{t \in [T]} \left\|\check{\Delta}_\theta^{(t)}\right\|_2$, and $B_\Delta \geq \max_{t \in [T]} \left\|\Delta_\lambda^{(t)}\right\|_\infty$.*

If we run Algorithm 3 with the step sizes $\eta_\theta := B_\Theta / B_{\check{\Delta}} \sqrt{2T}$ and $\eta_\lambda := \sqrt{(m+1)\ln(m+1)/T B_\Delta^2}$, then the result satisfies the conditions of Theorem 6 for:

$$\epsilon_\theta = 2 B_\Theta B_{\check{\Delta}} \sqrt{\frac{1 + 16 \ln \frac{2}{\delta}}{T}}$$

$$\epsilon_\lambda = 2 B_\Delta \sqrt{\frac{2(m+1)\ln(m+1)\left(1 + 16 \ln \frac{2}{\delta}\right)}{T}},$$

with probability $1 - \delta$ over the draws of the stochastic (sub)gradients.

4.6. Shrinking The Stochastic Proxy Lagrangian Solution

Like Algorithm 1, Algorithms 2 and 3 return a stochastic model with support on T solutions. Again, we show that we can find just as good a stochastic model with minimal support on $m+1$ solutions.

It turns out that the same existence result that we provided for the Lagrangian game (Lemma 4)—of a *Nash* equilibrium—holds for the proxy-Lagrangian (this is Lemma 14 in Appendix B). Furthermore, the exact same linear programming procedure of Lemma 5 can be applied (with the $\vec{g}_i$s being defined in terms of the *original*—not proxy—constraints) to yield a solution with support size $m+1$, and works equally well. This is easy to verify: since $\bar{\theta}$, as defined in Theorem 6, is a distribution over the $\theta^{(t)}$s, and is therefore feasible for the LP, the *best* distribution over the iterates will be at least as good.

5. Experiments

We illustrate the broad applicability of rate constraints and investigate how well different optimization strategies perform. We use the experiments to investigate the following questions:

Do rate constraints help in practice?

- Can we effectively solve the rate-constrained optimization problem?
- Can we get good results *at test time* by training with rate constraints?
- Do rate constraints interact well with other types of constraints (e.g. data-independent monotonicity shape constraints)?

Does the proxy-Lagrangian better solve the constrained optimization problem?

- Does simply using a hinge surrogate for both players as done in prior work over-constrain in practice?
- Does the proposed proxy-Lagrangian formulation result in better solutions?
- With the proxy-Lagrangian, is it necessary in practice for the λ-player to minimize the swap regret or does simply minimizing the external regret work just as well?

Do we really need stochastic classifiers?

- Do the iterates oscillate due to non-existence of an equilibrium in the non-convex setting, causing the last iterate to sometimes be very bad?

- Does the proposed sparsely supported m-stochastic classifier work at least as well in practice as the T-stochastic classifier?
- Does the *best* iterate perform as well as the stochastic classifiers?

To investigate these questions, we compared twelve optimization algorithms for each of seven datasets. Table 4 lists the three benchmark and four real-world datasets we used, each randomly split into train, validation and test sets. We experimented with seven different rate constraints and monotonicity constraints (Groeneboom and Jongbloed, 2014) as described in Table 5 and the following subsections. The last column of Table 5 states whether the classifier has access to information about the different datasets used in the constraints, for example, if there are ten constraints defined on ten different countries, is country also in the feature vector x?

As listed in Table 4, we performed the experiments on linear models and two types of nonlinear models: standard two-layer ReLU neural nets (NN), and a two-layer calibrated ensemble of lattices (Lattices) (Canini et al., 2016).

The rest of this section delves deeper into experimental details and result tables. Then, Section 6 discusses the results and how they provide positive and negative evidence for the above research questions – the reader may prefer to skip to Section 6 and only consult the following experimental details as needed.

All of our experiments are on binary classification datasets and the objective used was hinge loss and the constraint type used (e.g. unconstrained, hinge relaxation or original 0-1) depends on the algorithm and is made clear in the results.

5.1. TensorFlow Implementation

Our experiments were all run using TensorFlow. We have open-sourced our implementation of Lagrangian and proxy-Lagrangian optimization in a library called TensorFlow Constrained Optimization: `https://github.com/google-research/tensorflow_constrained_optimization`.

Experiments with linear models and DNN models used standard TensorFlow functions. Experiments on lattice models used the open-source TensorFlow Lattice package, and consist of learned one-dimensional piecewise linear feature transformations followed by an ensemble of lattices; all model parameters were jointly trained. For more details on lattice models see Gupta et al. (2016); Canini et al. (2016); You et al. (2017). Lattice models can be efficiently constrained for *partial monotonicity* shape constraints, where the term *partial* refers to the practitioner specifying which features can only have a positive (or negative) impact on $f(x)$. To produce the desired partial monotonicity, a large number of data-independent linear inequality constraints are needed, each constraining a pair of model parameters. In the TensorFlow Lattice package, these monotonicity shape constraints are handled by a projection after each minibatch of stochastic gradients.

5.2. Hyperparameter Optimization

For each of the different datasets, we fix the number of loops and model architecture ahead of time to perform well for the unconstrained problem. For the unconstrained problem, we validated the ADAM learning rate. Then for each of the twelve constrained optimization algorithms, we validated two ADAM learning rates, one for optimizing the model parameters θ, and the other for optimizing the constraints parameters λ. All ADAM learning rates were varied by powers of 10 around the usual default of ADAM learning rate of 0.001.

Table 4: Datasets and Model Types Used in Experiments

Dataset	Features	Train	Valid	Test	Model Type	Model Size or # Parameters
Bank Marketing	60	31,647	4,521	9,042	Linear	61
Adult	122	34,189	4,884	9,768	Linear	123
COMPAS	31	4,320	612	1,225	2 Layer NN	10 hidden units
Business Entity	37	11,560	3,856	3,856	2 Layer NN	16 hidden units
Thresholding	7	70,874	10,125	20,250	2 Layer NN	32 hidden units
Map Intent	32	420,000	60,000	120,000	Lattice Ens.	93,600
Filtering	16	1,282,532	183,219	366,440	Lattice Ens.	3,305

The usual strategy of choosing hyperparameters that score best on the validation set is not satisfying in the constrained optimization setting, because now there are two metrics of interest: accuracy and constraint violation, and the appropriate trade-off between them may be problem dependent. One solution researchers turn to is to side-step the issue of choosing one set of hyperparameters, and instead present the Pareto frontier of results over many hyperparameters on the test set. While certainly valuable in a research setting, we must be mindful that in practice one cannot see the Pareto frontier on the test set, and must make a choice for hyperparameters based only on the training and validation sets (as is standard).

For our experiments, we investigate the practical setting in which one must choose one set of hyperparameters on which to evaluate the test set. For that, we need a heuristic to choose the best hyperparameters based only on the training and validation data. We analyzed a number of such heuristics that differently balance the validation accuracy and constraint violation, and were unable to find any heuristic that was perfect, but settled on the following strategy that has some nice properties. Rank each hyperparameter vector β by its validation loss $\text{LossRank}(\beta)$, and create a second ranking of each hyperparamter choice by its maximum constraint violation on the validation set $\text{WorstConstraintRank}(\beta)$. Then choose the hyperparameter vector β that satisfies:

$$\operatorname*{argmin}_{\beta} \max \left\{\text{LossRank}(\beta), \text{WorstConstraintRank}(\beta)\right\}, \tag{18}$$

with ties broken by the minimizing the validation loss.

This strategy chooses the hyperparameter set that has both low loss and small constraint violations, and guarantees that no other hyperparameter set choice would have both better validation accuracy and smaller constraint violations.

5.3. Algorithms Tested

We experimented with four groups of algorithms:

1. **Unconstrained:** the model is trained without any constraints.
2. **Hinge:** We use a hinge relaxation of the constraints in place of the actual constraints in the Lagrangian as per Algorithm 5.
3. **0-1 swap:** This refers to Algorithm 3, which directly uses the 0-1 constraint in the proxy-Lagrangian, the λ-player minimizes swap-regret and the θ-player minimizes external regret.

Table 5: Constraints Used in Experiments

Dataset	Constraints (# of constraints)	Constraint Group in x?
Bank Marketing	Demographic Parity (5)	Y
Adult	Equal Opportunity (4)	Y
COMPAS	Equal Opportunity (4)	Y
Business Entity Res.	Minimum Recall (18) and Equal Accuracy (1)	Y
Thresholding	Steering Examples Minimum Acc. (1)	N
Map Intent	Not Worse Off (10), Monotonicity (148,800)	Y
Filtering	Loss-only Churn (11), Monotonicity (9,740)	Y

Algorithm 4 Optimizes the Lagrangian formulation *with* proxy constraints. Like the proxy-Lagrangian, this is a non-zero-sum game, but unlike the proxy-Lagrangian, we have no theoretical justification for it. That said, it makes intuitive sense, and works well in practice. The λ-player optimizes based on the proxy-constraints and the θ-player optimizes based on the original constraints. The parameter R is the radius of the Lagrange multiplier space $\Lambda := \left\{\lambda \in \mathbb{R}_+^m : \|\lambda\|_1 \leq R\right\}$, and the functions Π_Θ and Π_Λ project their arguments onto Θ and Λ (respectively) w.r.t. the Euclidean norm. $\{g_i\}_{i=1}^m, \{\widetilde{g}_i\}_{i=1}^m$ are respectively the original constraints and proxy-constraints.

ProxyAdditiveExternalLagrangian ($R \in \mathbb{R}_+, g_0 : \Theta \to \mathbb{R}, \{g_i\}_{i=1}^m, \{\widetilde{g}_i\}_{i=1}^m, T \in \mathbb{N}, \eta_\theta, \eta_\lambda \in \mathbb{R}_+$):

Initialize $\theta^{(1)} = 0$, $\lambda^{(1)} = 0$ *// Assumes* $0 \in \Theta$

For $t \in [T]$:

  Let $\check{\Delta}_\theta^{(t)}$ be a stochastic subgradient of $g_0(\theta^{(t)}) + \sum_{i=1}^m \lambda_i^{(t)} \widetilde{g}_i(\theta)$ w.r.t. θ

  Let $\Delta_\lambda^{(t)}$ be a stochastic gradient of $g_0(\theta^{(t)}) + \sum_{i=1}^m \lambda_i^{(t)} g_i(\theta)$ w.r.t. λ

  Update $\theta^{(t+1)} = \Pi_\Theta\left(\theta^{(t)} - \eta_\theta \check{\Delta}_\theta^{(t)}\right)$ *// Projected SGD updates ...*

  Update $\lambda^{(t+1)} = \Pi_\Lambda\left(\lambda^{(t)} + \eta_\lambda \Delta_\lambda^{(t)}\right)$ *// ...*

Return $\theta^{(1)}, \ldots, \theta^{(T)}$ and $\lambda^{(1)}, \ldots, \lambda^{(T)}$

4. **0-1 ext:** This refers to Algorithm 4 training the non-zero-sum game where θ player minimizes the original Lagrangian but the λ-player minimizes external-regret on the Lagrangian with the original constraints replaced by the proxy constraints. This is the "obvious" non-zero-sum analogue of the Lagrangian, but does not enjoy the theoretical guarantees of the proxy-Lagrangian. This is used as a comparison to 0-1 swap to see whether minimizing external regret (instead of the more complex swap regret) suffices in practice.

Then, for each constrained optimization technique, we show the results for the following four solution types:

1. **T-stoch**: the stochastic solution that is the uniform distribution over the T iterates $\theta^{(1)}$, ..., $\theta^{(T)}$.
2. **m-stoch**: the stochastic solution obtained by applying the "shrinking" technique to the T-stoch solution on the training set, which will have support on at most $m+1$ deterministic solutions.
3. **Last**: the deterministic solution defined by the last iterate $\theta^{(T)}$.
4. **Best**: the deterministic solution defined by the "best" iterate out of all T iterates $\theta^{(1)}$, ..., $\theta^{(T)}$, where "best" is chosen by the heuristic given in Equation 18 applied on the training set.

Table 6: Bank Marketing Experiment Results

Algorithm	Train Err.	Valid Err.	Test Err.	Train Vio.	Valid Vio.	Test Vio.
Unconstrained	0.0948	0.0935	0.0937	0.0202	0.0220	0.0152
Hinge m-stoch.	0.0955	0.0954	0.0949	0	-0.0008	-0.0030
Hinge T-stoch.	0.1109	0.1114	0.1121	-0.0177	-0.0181	-0.0179
Hinge Best	0.0964	0.0969	0.0955	-0.0032	-0.0045	-0.0047
Hinge Last	0.1122	0.1129	0.1140	-0.02	-0.02	-0.02
0-1 swap. m-stoch.	0.0939	0.0943	0.0951	0	-0.0005	0.0019
0-1 swap. T-stoch.	0.0963	0.0955	0.0947	0.0004	-0.0003	-0.0031
0-1 swap. Best	0.0936	0.0935	0.0932	-0.0004	-0.0009	-0.0041
0-1 swap. Last	0.0963	0.0957	0.0954	-0.0007	-0.001	-0.0035
0-1 ext. m-stoch.	0.0946	0.0952	0.0946	0	-0.001	-0.0024
0-1 ext. T-stoch.	0.1083	0.1087	0.1085	-0.0135	-0.0146	-0.0139
0-1 ext. Best	0.0963	0.0964	0.0953	-0.0021	-0.0016	-0.0056
0-1 ext. Last	0.1029	0.1032	0.1010	-0.0046	-0.0072	-0.0056

We note that in the non-convex proxy-Lagrangian setting, the 0-1 swap algorithm's T-stoch or m-stoch solutions come with theoretical guarantees if we replace the SGD with the approximate optimization oracle. In contrast, the 0-1 ext algorithm has no such guarantees, but is simpler. Similarly, in the non-convex setting, the deterministic solutions will not have any guarantees, but are even simpler.

5.4. Bank Marketing

The Bank Marketing UCI benchmark dataset (Lichman, 2013) classifier predicts whether someone will sign up for the bank product being marketed. This dataset was used to test improving statistical parity for a linear model in Zafar et al. (2015) but with only one protected group based on age. We similarly use a linear model and age as a protected feature, but create 5 protected groups based on the five training set quantiles of age. We add a statistical parity rate constraint for each of the five age quantiles with an additive slack of 2%:

$$p^+(D_k;\theta) \leq p^+(D;\theta)) - .02,$$

where D_k are the training examples from the kth protected group for $k = 1, 2, \ldots, 5$, and D are all the training examples.

The results can be found in Table 6. We note that the Hinge Last solution is a *degenerate* solution in that it always predicts the a priori more probable class.

5.5. Adult

We used the benchmark Adult income UCI dataset (Lichman, 2013). The goal is to predict whether someone makes more than 50k per year, and also do well at the *equal opportunity* fairness metric. We used four protected groups: two race-based (Black or White) and two sex-based (Male or Female). We preprocessed the dataset consistent with Zafar et al. (2015) and Goh et al. (2016). Goh et al. (2016)

Table 7: Adult Experiment Results

Algorithm	Train Err.	Valid Err.	Test Err.	Train Vio.	Valid Vio.	Test Vio.
Unconstrained	0.1421	0.1348	0.1428	0.0803	0.0604	0.0555
Hinge m-stoch.	0.1431	0.1348	0.1442	0	-0.0088	0.0025
Hinge T-stoch.	0.1462	0.1394	0.1481	-0.0409	-0.0372	-0.0436
Hinge Best	0.1424	0.1333	0.1447	-0.0280	-0.0154	-0.0317
Hinge Last	0.1532	0.1490	0.1551	-0.0174	-0.0217	-0.0254
0-1 swap. m-stoch.	0.1431	0.1349	0.1432	0.0176	0.0023	0.0559
0-1 swap. T-stoch.	0.1428	0.1365	0.1436	0.0054	0.0354	0.0285
0-1 swap. Best	0.1426	0.1354	0.1440	-0.0016	0.0140	0.0154
0-1 swap. Last	0.1436	0.1358	0.1443	0.0069	0.0248	0.0221
0-1 ext. m-stoch.	0.1418	0.1348	0.1432	0	-0.0019	0.0059
0-1 ext. T-stoch.	0.1441	0.1369	0.1447	0.0034	0.0220	0.0174
0-1 ext. Best	0.1420	0.1348	0.1432	-0.0374	-0.0333	-0.0015
0-1 ext. Last	0.1436	0.1358	0.1448	-0.0116	0.0078	0.0028

showed that by explicitly constraining the difference in coverage and using a *linear* model, they could achieve higher p fairness and better accuracy than earlier work using correlation constraints of Zafar et al. (2015) by up to 0.5% on this dataset.

For these experiments, we added four rate constraints to the training to impose *equal opportunity* at 95%, that is for each of the protected groups (Black, White, Female and Male) the constraints force the classifier's coverage (the proportion classified positive) on the positively labeled examples for each protected group to be at least 95% of the overall coverage on the positively labeled examples:

$$p^+(D_k[y=1];\theta) \geq 0.95 p^+(D[y=1];\theta), \tag{19}$$

where D_k are the training examples from the kth protected group for $k = 1, 2, \ldots, 4$, and D are all the training examples.

We use a linear model. The results can be found in Table 7.

5.6. COMPAS

The positive label in the ProPublicas COMPAS recidivism data is a prediction the person will re-offend. The goal is to predict recidivism with fairness constraints and we preprocess this dataset in a similar manner as in the Adult dataset and the protected groups are also similar: two race-based (Black and White) and two sex-based (Male and Female). The classifier we use is a 2 layer neural network with 10 hidden units.

In this experiment, the goals are quite similar to that of the Adult experiment. Our protected groups are again two races (Black and White) and two sexes (Male and Female) and the goal is to constrain equal opportunity such that no group is unfairly getting targeted. However, instead of expressing the constraint with multiplicative slack as in the Adult experiments, we expressed it as an additive slack of 5%:

$$p^+(D_k[y=1];\theta) \leq p^+(D[y=1];\theta) + .05,$$

Table 8: COMPAS Experiment Results

Algorithm	Train Err.	Valid Err.	Test Err.	Train Vio.	Valid Vio.	Test Vio.
Unconstrained	0.3056	0.3160	0.3109	0.1151	0.2143	0.1082
Hinge m-stoch.	0.3711	0.3744	0.3676	0	0.0395	0.0284
Hinge T-stoch.	0.2880	0.3387	0.3198	0.1093	0.1779	0.0917
Hinge Best	0.2840	0.3322	0.3223	0.0803	0.1262	0.0800
Hinge Last	0.2882	0.3322	0.3231	0.1275	0.1968	0.0996
0-1 swap. m-stoch.	0.3132	0.3015	0.3174	0.0004	0.0851	0.0111
0-1 swap. T-stoch.	0.2968	0.3208	0.3219	0.0257	0.1286	0.0547
0-1 swap. Best	0.3009	0.3096	0.3125	0.0281	0.1084	0.0356
0-1 swap. Last	0.3023	0.3096	0.3158	0.0412	0.1153	0.0480
0-1 ext. m-stoch.	0.3145	0.3080	0.3146	0	0.0813	0.0147
0-1 ext. T-stoch.	0.2990	0.3128	0.3086	0.0323	0.1154	0.0321
0-1 ext. Best	0.3106	0.3160	0.3101	-0.0069	0.0797	-0.0085
0-1 ext. Last	0.2935	0.3160	0.3125	0.0330	0.1231	0.0325

where D_k are the training examples from the kth protected group for $k = 1, 2, \ldots, 4$, and D are all the training examples. That is, the positive prediction rate of the positively labeled examples for each protected class can exceed that of the overall dataset by at most 5%.

The results are shown in Table 8.

5.7. Business Entity Resolution

In this entity resolution problem from Google, the task is to classify whether a pair of business descriptions describe the same real-world business. For example, is *Siam Thai Restaurant at Main and 5th* the same business as *Old Siam Thai at 5070 Main St*? Features include measures of similarity of the two business titles, phone numbers, and so on. We add two types of constraints to the training. First, the dataset is world-wide, and for each of the 16 most frequent countries, we imposed a minimum recall rate constraint of 95 percent:

$$p^+(D_k[y = 1]; \theta) \geq .95,$$

where D_k are the training examples from the kth country for $k = 1, 2, \ldots, 16$. It is also known whether each example is a chain business or not. We impose the same minimum recall rate constraint on chain business examples and non-chain business examples. Additionally, we add an equal accuracy constraint that the accuracy on not-chain businesses should not be worse than the accuracy on chain businesses by more than ten percent, as a *proxy fairness constraint* (Gupta et al., 2019) to making sure large and small businesses receive similar performance from the model:

$$\frac{c^+(D_{\text{notCh}}[y = 1]; \theta) + c^-(D_{\text{notCh}}[y = -1]; \theta)}{|D_{\text{notCh}}|} \geq \frac{c^+(D_{\text{ch}}[y = 1]; \theta) + c^-(D_{\text{ch}}[y = -1]; \theta)}{|D_{\text{ch}}|} - 0.1,$$

where ch is an abbreviation for chain.

Table 9: Business Entity Resolution Experiment Results: 2 Layer NN

Algorithm	Train Err.	Valid Err.	Test Err.	Train Vio.	Valid Vio.	Test Vio.
Unconstrained	0.1223	0.1505	0.1520	0.1727	0.2172	0.2357
Hinge m-stoch.	0.2405	0.2509	0.2535	0	0.0341	0.0282
Hinge T-stoch.	0.3308	0.3351	0.3446	-0.0258	0.0196	-0.0082
Hinge Best	0.2657	0.2720	0.2786	-0.0083	0.0437	0.0026
Hinge Last	0.2483	0.2624	0.2617	-0.0175	0.0125	0.0421
0-1 swap. m-stoch.	0.1751	0.1953	0.1983	0	0.0745	0.0898
0-1 swap. T-stoch.	0.1506	0.1749	0.1760	0.0950	0.1427	0.1933
0-1 swap. Best	0.1407	0.1687	0.1696	0.0681	0.1224	0.1706
0-1 swap. Last	0.1699	0.1910	0.1927	0.0252	0.0864	0.0846
0-1 ext. m-stoch.	0.1891	0.2060	0.2063	0	0.0741	0.0752
0-1 ext. T-stoch.	0.1934	0.2082	0.2092	0.0011	0.0652	0.0770
0-1 ext. Best	0.1889	0.2053	0.2049	0.0026	0.0750	0.0750
0-1 ext. Last	0.1968	0.2118	0.2130	0.0008	0.0594	0.0750

We ran this experiment with a two-layer neural network, the results are shown in Table 9. In the top row, one sees that the unconstrained model has a very high maximum constraint violation, because it is very difficult to achieve 95% recall for all regions.

5.8. Thresholding

For this Google problem, a ranked list of hundreds of business results is given for a query, and the task is to threshold the list to return only the results worth showing a user. We use a 2 layer neural network with 32 hidden units as the classifier.

A medium-size labeled set is available with labels that are known to be noisy, and the label noise is not zero-mean and not homogeneous across the feature space. That set is broken uniformly and randomly into train/validation/test sets.

We also have an auxiliary independent set of $1,814$ *steering examples* (see Section 3.5) which were more carefully labeled by expert labelers, and were actively sampled to pinpoint key types of problems. If one only uses the steering examples (ignoring the noisy labeled data), previous experiments have shown that one can stably achieve a 33% cross-validation error rate on the steering examples. The goal is to have a model that gets that 33% error on the steering examples, but also works as well as possible on the larger noisy data.

The top row of Table 10 shows that only training on the noisy train data produces a error rate of 35% on the noisy test data, which violates our goal of 33% error on the steering examples by 3% (that is, it has an error rate 36% on the steering examples).

In the other extreme, only training on the steering examples is also unsatisfying: as reported in the second row of Table 10 that performs poorly on the large noisy test set with an error rate of 39%, because the steering example set does not cover the entire feature space.

Table 10: Thresholding Experiment Results

Algorithm	Train Err.	Valid Err.	Test Err.	Steering Violation
Unconstrained	0.3595	0.3491	0.3538	0.0316
Unconstrained Trained on Steering	0.3909	0.3924	0.3930	-0.0456
Hinge m-stoch.	0.3601	0.3512	0.3582	0
Hinge T-stoch.	0.3635	0.3558	0.3594	-0.0037
Hinge Best	0.3606	0.3509	0.3560	-0.0031
Hinge Last	0.3621	0.3542	0.3594	-0.0003
0-1 swap. m-stoch.	0.3574	0.3500	0.3557	-0.0025
0-1 swap. T-stoch.	0.3593	0.3513	0.3551	0.0010
0-1 swap. Best	0.3561	0.3484	0.3532	-0.0020
0-1 swap. Last	0.3584	0.3497	0.3543	-0.0020
0-1 ext. m-stoch.	0.3605	0.3504	0.3568	-0.0009
0-1 ext. T-stoch.	0.3602	0.352	0.3553	0.0010
0-1 ext. Best	0.3569	0.3486	0.3515	-0.0014
0-1 ext. Last	0.3579	0.3500	0.3539	-0.0009

For the rest of the rows in Table 10, we train on the noisy data with a minimum accuracy rate constraint for 67% accuracy on the steering examples:

$$\frac{c^+(D_{\text{steering}}[y=1];\theta) + c^-(D_{\text{steering}}[y=-1];\theta)}{|D_{\text{steering}}|} \geq 0.67.$$

All of the different optimization methods find essentially feasible solutions, with many able to achieve the same or better test set performance as the unconstrained training (top row).

5.9. Map Intent

For this Google problem, the task is to classify whether a query is seeking a result on a map. We add ten *not worse off* rate constraints for ten regions that constrain the new model training to be at least as accurate as the production classifier is for each of those ten regions.

$$\frac{c^+(D_{\text{region}}[y=1];\theta) + c^-(D_{\text{region}}[y=-1];\theta)}{|D_{\text{region}}|} \geq \kappa_{\text{region}},$$

where κ_{region} is the accuracy of the production classifier for that region. The feature vector x includes ten Bool features that indicate if x belongs to each of these ten regions (each example belongs to at most one region).

Thirty-two dense and categorical features are available. We train a model that is an ensemble of 300 calibrated lattices, where each lattice acts on 8 of the 32 features, with shared calibrators, and the lattices are interpolated using multi-linear interpolation, all implemented using the TensorFlow Lattice package. We enforce monotonicity constraints on 28 of the 32 features, resulting in an additional 148,800 constraints (each one is a linear inequality constraint on a pair of model parameters) applied during training; see Canini et al. (2016) for more technical details.

Table 11: Map Intent Experiment Results

Algorithm	Train Err.	Valid Err.	Test Err.	Train Vio.	Valid Vio.	Test Vio.
Unconstrained	0.3093	0.3122	0.3104	0.0187	0.0162	0.0319
Hinge m-stoch.	0.3130	0.3129	0.3124	0.0182	0.0176	0.0313
Hinge T-stoch.	0.3096	0.3136	0.3106	0.0194	0.0197	0.0210
Hinge Best	0.3056	0.3131	0.3104	0.0172	0.0194	0.0247
Hinge Last	0.3058	0.3130	0.3099	0.0177	0.0189	0.0220
0-1 swap. m-stoch.	0.2949	0.3002	0.2997	-0.0003	0.0025	0.0176
0-1 swap. T-stoch.	0.3004	0.3022	0.3024	0.0022	0.0061	0.0204
0-1 swap. Best	0.2949	0.3002	0.2997	-0.0003	0.0025	0.0176
0-1 swap. Last	0.2953	0.3004	0.3002	0.0013	0.0034	0.0192
0-1 ext. m-stoch.	0.3069	0.3115	0.3101	0.0094	0.0144	0.0231
0-1 ext. T-stoch.	0.3101	0.3121	0.3107	0.0132	0.0157	0.0243
0-1 ext. Best	0.3069	0.3115	0.3101	0.0094	0.0144	0.0231
0-1 ext. Last	0.3071	0.3111	0.3103	0.0096	0.0140	0.0242

5.10. Filtering

For this Google problem, the task is to classify whether a candidate result for a query should be immediately discarded as too irrelevant to be worthy of further processing. For this problem we take as given a base classifier $\tilde{h}$ and the goal is to maximize accuracy with minimal loss-only churn (see Section 3.3 for details). The base classifier h was trained as a regression model to minimize mean squared error with respect to a real-valued label on $[-1, 1]$, but then used as a classifier with decision threshold 0.0 to filter the results. The new classifier is trained on the same training data, but we pre-threshold the real-valued training labels to form binary classification labels, then train the new classifier to minimize the classification error rate. We add ten loss-only churn rate constraints to individually restrict the loss-only churn with respect to the production model for each of ten mutually-exclusive geographic regions to less than 5%:

$$\frac{c^+(D_{\text{region}}[y=-1, h=-1];\theta) + c^-(D_{\text{region}}[y=1, h=1];\theta)}{|D_{\text{region}}[h=y]|} \leq 0.05.$$

That is, we ask that no more than five percent of the base classifier's wins are lost for each of the ten regions. The feature vector x includes ten binary features that indicate if x belongs to one of these ten regions (some examples do not belong to any of the ten regions).

Both the given regression model $h(x)$ and the new classifier $f(x)$ use the same model architecture: both are lattice models that are an ensemble of 50 lattices, where each lattice acts on 6 of 16 continuous-valued features, each feature is calibrated by a monotonic piecewise linear transform that is shared across the lattices, the lattices are interpolated using multilinear interpolation, all model parameters trained jointly using the TensorFlow Lattice package. We enforce monotonicity constraints on 14 of the 16 features, resulting in an additional 9,740 constraints applied during training (each of these is simply a linear inequality constraint on a pair of model parameters); see Canini et al. (2016) for more technical details.

Table 12: Filtering Experiment Results

Algorithm	Train Err.	Valid Err.	Test Err.	Train Vio.	Valid Vio.	Test Vio.
Unconstrained	0.2747	0.2723	0.2761	0.3164	0.3107	0.3227
Hinge m-stoch.	0.3363	0.3362	0.3369	0	-0.0023	-0.0012
Hinge T-stoch.	0.3658	0.3656	0.3665	-0.0297	-0.0262	-0.0243
Hinge Best	0.3404	0.3403	0.3409	-0.0075	-0.0080	-0.0068
Hinge Last	0.3622	0.3618	0.3630	-0.0239	-0.0239	-0.0242
0-1 swap. m-stoch.	0.3230	0.3231	0.3239	0	0.0071	0.0130
0-1 swap. T-stoch.	0.3205	0.3208	0.3217	0.0096	0.0192	0.0227
0-1 swap. Best	0.3175	0.3178	0.3186	0.0081	0.0116	0.0156
0-1 swap. Last	0.3185	0.3189	0.3195	0.0112	0.0146	0.0118
0-1 ext. m-stoch.	0.3231	0.3234	0.3243	0	0.0048	0.0065
0-1 ext. T-stoch.	0.3300	0.3302	0.3309	0.0004	0.0008	0.0014
0-1 ext. Best	0.3180	0.3179	0.3190	0.0079	0.0116	0.0138
0-1 ext. Last	0.3268	0.3272	0.3278	0.0021	0.0055	0.0087

The production classifier $\tilde{h}$ had a test error rate of 39.72%. As hoped, by training specifically for this classification task, the new classifier $f(x)$ achieves lower test error rates: as low as 27.61% for the unconstrained training. However, the high test constraint violation of 32.27% (measured as the maximum violation over the ten regions) shows that the new unconstrained classifier loses a large number of the wins the base classifier had for at least one of the ten countries considered.

6. Discussion Of Experimental Results

Now that we have presented the experimental results, we return to discuss the experimental and theoretical evidence for and against the hypotheses and questions posed at the beginning of Section 5.

6.1. Do Rate Constraints Help In Practice?

Yes, overall the experiments show rate constraints are are a useful machine learning tool. Let us consider some more specific questions.

6.1.1. CAN WE EFFECTIVELY SOLVE THE RATE-CONSTRAINED OPTIMIZATION PROBLEM?

Yes, but the optimization algorithm does matter. Note here we are asking whether the optimization problem is well-solved, and thus we focus on the *training* error and the *training* violation.

The good news is that compared to unconstrained (top row in result tables) the 0-1 swap regret m-stochastic optimization (row 6 in result tables) consistently across all experiments did produce lower training constraint violations while still achieving reasonable training error compared with the unconstrained. Recall that each m-stochastic solves a linear program that sparsifies the corresponding T-stochastic such that the constraints are exactly satisfied if the T-stochastic solution is feasible, so it is by design that the m-stochastic solution train constraint violation is exactly 0.0 for many of the experiments. For Adult (see Table 7), the 0-1 swap m-stochastic train error is only .001 worse,

but the train violation drops from .0803 to .0176. For Bank Marketing (see Table 6), the train error is slightly better for , and the train violation drops from .0202 to 0.0. Similarly for COMPAS (see Table 8), the 0-1 swap m-stochastic has slightly higher training error but drops the train constraint violation from 0.1151 to almost zero. For Business Entity Resolution (Table 9), the training error does increase with 0-1 swap m-stochastic, but it is a reasonable price to pay in training accuracy for the huge reduction of the worst case equal-accuracy or min-recall constraint violation from 0.1727 to 0.0. For the Thresholding problem (Table 10), the 0-1 swap m-stochastic is again slightly better on training error and effectively reduces the constraint violation to 0.0, and similarly for the Map Intent experiment (Table 11), the training error is lower and the training constraint violation is lower. For Filtering (Table 12), the training error for 0-1 swap regret m-stochastic did go up significantly from 0.2747 to 0.3230, but the unconstrained training violation was horrendous at 0.3164 whereas the m-stochastic found a feasible solution. In conclusion for all experiments run, we found the 0-1 swap regret m-stochastic did a good or reasonable job at the optimization problem of minimizing training error and satisfying the constraints on the training set.

In contrast, one can see that using the baseline strategy of approximating all indicators with the hinge throughout the optimization can provide poor or even worse results than the unconstrained. For example, on the Map Intent experiment (see Table 11), the hinge T-stochastic solution manages to have slightly both worse training error and worse training constraint violation than the unconstrained. The other hinge optimizations are also un-compelling in this experiment. In contrast, the swap regret optimizations consistently find good solutions with lower training error and roughly zero training constraint violations. This is a challenging optimization problem because there are ten rate constraints on ten regions of differing sizes.

The baseline strategy of simply taking the last iterate often does a good job at solving the constrained problem, but sometimes is worse at optimizing the constrained problem than even the unconstrained solver. For example, on COMPAS (see Table 8) the Hinge Last training violation is actually bigger than the unconstrained training violation. While Hinge Last does achieve slightly better training error, it hasn't achieve better validation error (or test error), so we don't believe this was simply an unlucky validation of hyperparameter choice. For more details on why last iterate can perform badly, see Section 6.3.1.

While theory dictates a stochastic solution is necessary for guarantees, in practice the T-stochastic solutions can be quite poor, for example on Map Intent (Table 11) the Hinge T-stochastic solution is worse than unconstrained on both training error and training constraint violation. This may be due to bad early iterates, which would be diluted with a longer run time. Compared to the T-stochastic solutions, the m-stochastic solutions are always better on training error and never more violating, as designed.

The best iterate is by definition always at least as good as the last iterate on the training error and/or training violation. For all three optimization strategies (hinge, 0-1 swap regret, 0-1 external regret), the best iterate manages to consistently produce solutions that are better than the unconstrained in terms of training violations and have reasonable or good training errors.

6.1.2. CAN WE GET GOOD TEST RESULTS BY TRAINING WITH RATE CONSTRAINTS?

Yes, mostly. The m-stochastic and best iterate solutions do result in lower test violations and reasonable test errors for six of the seven experiments. However, for Adult (Table 7), the 0-1 swap m-stochastic failed to produce lower test violation nor lower test error than the unconstrained,

despite having much lower training and validation violations. Sadly, the good training and validation performance simply did not generalize to the test set. This case is hard in part because the Black constraint in the Adult dataset is based on a relatively small sample: only 345 positive training examples, 42 positive validation examples, and 179 positive test examples.

Overall, small constraint datasets can lead to poor generalization that can significantly hurt the overall metrics. The worst generalization happened with the Business Entity Resolution, where training violations for the proxy-Lagrangian methods ranged from $[0 - .095]$, but the test violations ranged from $[0.075 - 0.19]$. For that experiment, the hinge solutions generalized better, but at the cost of much higher test errors. Business Entity is a particularly hard problem because there are 16 constraints on different regions, some of which have very small datasets, and just like training a model, there is a greater risk of poor generalization if the datasets used in the constraints are small.

For the larger datasets (Map Intent in Table 11 and Filtering in Table 12), the classifier performance was much more similar on training and test sets.

For a further discussion of generalization for rate-constraints, with some theoretical results and practical strategies, see Cotter et al. (2019a).

6.1.3. DO RATE CONSTRAINTS INTERACT WELL WITH OTHER TYPES OF CONSTRAINTS?

We did not see any problems from combining rate constraints with monotonicity constraints. For Map Intent (Table 11), both 0-1 optimization strategies worked reasonably, with the 0-1 swap regret producing attractive solutions that both notably lowered training error and satisfied the constraints on the training set. This shows that the addition of the 148,800 sparse linear inequality constraints for monotonicity did not cause a problem in optimizing the rate constrained problem. Similarly, for the Filtering (Table 12), the addition of the 9,740 sparse linear inequality constraints for monotonicity did not keep the optimizers from satisfying the rate constraints.

6.2. Does the Proxy-Lagrangian Better Solve the Constrained Optimization Problem?

We break this question into a few specific questions.

6.2.1. DOES SIMPLY USING HINGE SURROGATE FOR BOTH PLAYERS OVERCONSTRAIN?

We hypothesized that using the hinge loss as a convex relaxation to the 0-1 indicators in the rate constraints would cause the constrained optimization to find overly-constrained solutions at the cost of more training accuracy than needed to satisfy the constraints. This was not as large an effect as we expected. However, it can be seen in the Business Entity Resolution (Table 9) experiment where the hinge training violations are negative and the training errors are relatively high, whereas the 0-1 m-stochastic solutions crisply achieve the constraint with much lower training errors.

6.2.2. DOES THE PROXY-LAGRANGIAN FORMULATION RESULT IN BETTER SOLUTIONS?

In most experiments, there were trade-offs between test constraint violation and test accuracy which make it difficult to compare the hinge solutions to the proxy-Lagrangian solutions (denoted 0-1 in the tables) on the test metrics.

On the training metrics, there is stronger evidence the $0 - 1$ m-stochastic optimization is in fact doing a better job solving the optimization problem than the hinge m-stochastic. For seven of the seven experiments, the $0 - 1$ ext. m-stochastic produced both lower train error and lower

train violation than the Hinge m-stochastic solution. This was also true for five out of the seven experiments for the $0 - 1$ swap m-stochastic, and the solutions were close for the remaining two experiments.

In the case of Map Intent (Table 11), we clearly see that the Hinge solutions perform worse in both accuracy and fairness constraints than the 0-1 proxy-Lagrangian procedures on both training and testing. In the case of Thresholding, we see that the Hinge procedures seem to do slightly worse in final accuracy at the cost of over-constraining. We see that in Business Entity Resolution (Table 9), the Hinge procedures attain significantly higher errors than the other methods but do attain better constraint satisfaction on testing. Thus, even though proxy-Lagrangian formulation may seem better on a few of the datasets, this effect was not seen consistently across the remaining datasets and thus, the question of whether the proxy-Lagrangian attains better solutions in practice remains inconclusive.

6.2.3. IS MINIMIZING SWAP REGRET NECESSARY, OR DOES EXTERNAL REGRET SUFFICE?

Our theoretical results show that in the proxy-Lagrangian setting, the appropriate type of equilibrium (i.e. semi-coarse correlated equilibrium) has optimality and feasibility guarantees for the original constrained optimization problem. In order to attain such an equilibrium, we needed the λ-player to minimize swap-regret (while the θ-player minimizes the classic external regret). However, minimizing swap-regret involves a more complicated procedure. We used the strategy of Gordon et al. (2008), who showed that any external regret minimizing procedure can be turned into one that minimizes swap regret by a meta-algorithm which runs m copies of the procedure. We questioned whether it would be just as good in practice to use the simpler external-regret minimizing procedure, which still leads to a coarse-correlated equilibrium (which is a *weaker* notion than semi-coarse correlated equilibrium).

Comparing the swap regret to the external regret for the same solution type (m-stochastic/T-stochastic/best/last), the external regret usually ends up with a solution with slightly lower test violations but slightly higher test error. The only exception was the Map Intent experiment in which the swap-regret solutions were both considerably more accurate and better at satisfying the constraints. In conclusion, we have not seen experimental evidence that the extra complexity of swap regret is warranted in practice.

6.3. Do We Really Need Stochastic Classifiers?

Next, we investigate some specific questions regarding the necessity of stochastic solutions over a deterministic classifier.

6.3.1. DO THE ITERATES OSCILLATE IN THE NON-CONVEX SETTING?

As noted in Section 6.1.1, simply taking the last iterate can produce worse constraint violations to the optimization problem then solving the unconstrained problem. Figure 3 plots the error and constraints for each of the iterates on the COMPAS dataset which shows such oscillation. This suggests that, as we showed in Section 1.4, the phenomenon of the non-convex Lagrangian having no pure Nash equilibrium to which it can converge, may occur in practice.

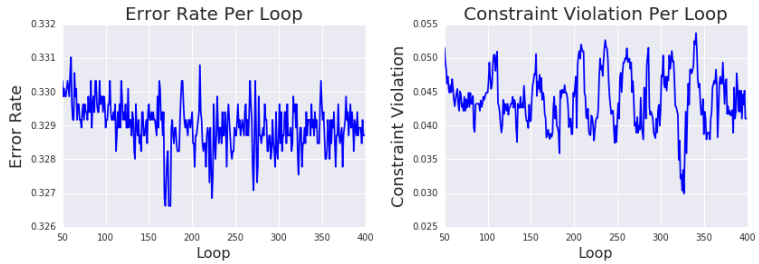


Figure 3: The plots for the errors and constraint violations for each iteration during training on the COMPAS dataset with equal opportunity constraints with an additive slack of 5%. The oscillation due to the conflicting goals of accuracy and constraints suggest that there may be no pure equilibrium to converge to in the non-convex setting.

6.3.2. DOES m-STOCHASTIC BEAT T-STOCHASTIC?

Our theoretical results guarantee that the m-stochastic solution (which is obtained through solving a simple LP on the T-stochastic iterates) will be no worse than the T-stochastic solution by forcing the m-stochastic solution to be at least as feasible as the T stochastic solution, while having no worse error (at least on the training set). Our hope is therefore that our "shrinking" procedure will find better solutions on *test* data.

We see consistently across datasets as well as optimization techniques that the m-stochastic is indeed better than the T-stochastic in terms of both error and constraint violation on training. Part of this effect may be due to the fact that many of the iterates of the T-stochastic perform poorly, for example the early iterates before our procedures are able to get to reasonable solutions. Or during phase-transitions if there is oscillation between satisfying constraints and satisfying error. Fortunately, the shrinking procedure seems to be able to choose a good re-weighting of the T-stochastic solution in order to attain well-performing final results.

We also see that in the vast majority of situations, the test performance for the m-stochastic either surpasses that of the T-stochastic, or there is an accuracy-fairness trade-off between the two (and hence, not straightforward to compare the two).

6.3.3. DOES THE BEST ITERATE PERFORM AS WELL AS THE STOCHASTIC CLASSIFIERS?

We have already established that a stochastic solution may be difficult to avoid, in theory (Section 1.4). However, stochastic solutions are unappealing in practice: they take more memory, are harder to test and debug due to their inherent randomness, and a randomized decision may feel less *fair* in certain contexts (even if the outcomes statistically improve the desired fairness metric). Here, we ask if a stochastic solution is needed *in practice*, based on test metrics.

First, we compare the 0-1 swap regret m-stochastic solution, which is our theoretically preferred stochastic solution, to the 0-1 swap regret best iterate. The 0-1 swap best iterate is never a strictly worse choice than the 0-1 swap m-stochastic. In some cases the m-stochastic solution puts all or most of its weight on the best iterate—for example, for the Map Intent problem (Table 11) the two

solutions are identical. In other experiments the solutions differ but both achieve reasonable different trade-offs of test error and test violation, for example on the Thresholding problem (Table 10) and COMPAS (see Table 8), the best iterate has a lower test error, but a higher test constraint violation.

Comparing the m-stochastic solution and best iterate solution for the 0-1 external regret optimization similarly suggests that much of the time the best iterate works just as well in practice.

6.3.4. DOES BEST ITERATE PERFORM BETTER IN PRACTICE THAN LAST ITERATE?

We have established that using the best iterate works well in practice. Now we discuss how much better best is than simply taking the last iterate. In fact, the last iterate is strictly worse at test metrics than the best iterate for 4 of the 7 experiments: Bank, Thresholding, Adult, and Compass; and the two solutions are similar for the other three experiments.

If there are oscillations on the loss and constraint violation (as shown in Figure 6.3 for COMPAS), then the last iterate could be highly unstable and could produce undesirable solutions. In practice, the strongest evidence for last being a risky choice is Hinge Last on COMPAS, where test error went up from 0.3109 to 0.3231, and training violation only went down from 0.1082 to 0.0996.

Overall, the experimental results suggest that the best iterate is preferable to the last iterate.

7. Conclusions, Advice To Practitioners, And Open Questions

In this paper, we provide the most comprehensive study to-date of training classifiers with a broad array of rate constraints, with new theoretical, algorithmic, and experimental results as well as practical insights and guidance for using rate constraints to solve real-world problems. Next, we provide some conclusions, specifically draw out our best advice to practitioners, and note some open questions.

7.1. Advice To Practitioners: How To Train Classifiers With Rate Constraints

Based on our experiments, our advice to practitioners is to optimize the rate-constrained training using either our proposed non-zero-sum variant of the normal Lagrangian formulation (0-1 external regret) and taking the last iterate.

The 0-1 external regret optimization procedure is simple: when optimizing the model parameters θ use stochastic gradient descent as usual with a hinge relaxation of the indicators in the constraints, and when optimizing the Lagrange multipliers λ use stochastic gradient descent, but do not relax the indicators in the rate constraints. If one needs a deterministic solution, ideally one would take the *best iterate*, but this requires storing all the candidate iterates on the Pareto frontier during training, in order to rank them by the training objective and training error at the end, and in the worst case that could be all candidate iterates. However one can control the number of candidate iterates, for example by sub-sampling them, or waiting until late in training to sample them. Simply taking the last iterate usually yielded reasonable results, but we do see in practice that the last iterate may perform strictly worse under all metrics than the best iterate.

We caution against relaxing the indicators for both the θ-player and λ-player (hinge last). It is hardly simpler than the 0-1 external regret optimization, and experimentally generally (but not always) produced worse test results, sometimes notably worse.

7.2. Advice To Practitioners: Plan To Overfit The Constraints

A key issue with using rate constraints is *generalization*: satisfying the constraints on the training examples does not necessarily mean that they will be satisfied on new test sets, and the generalization may be worse if the test examples are drawn from a different distribution. In expressing the rate constraints, one should add in some slack to account for generalization issues, especially if the constraints are optimized on small datasets.

In Cotter et al. (2019a), we extend the ideas of this paper with a focus on generalization. We show that providing different *datasets* to the two players, instead of (or in addition to) different constraint functions, can theoretically and practically improve generalization.

7.3. Advice To Practitioners: How The Constraints Are Specified Matters

We have learned that in practice that *how* one specifies the datasets and slack in a rate constraint is very important - see Section 3.7 for more discussion.

7.4. More Experimental Conclusions

The clearest experimental finding is that treating the optimization as a non-zero-sum two-player game where the λ-player does not relax the indicators in the rate constraints (notated as 0-1 in the experimental tables) does generally help, both in finding a better solution to the optimization problem (i.e. train metrics), and in practice (i.e. test metrics). Another fairly clear experimental finding is that the T-stochastic solution can effectively be sparsified to an m-stochastic solution, generally with improved metrics.

While the T-stochastic solution has better theoretical guarantees than any of our deterministic solutions, especially for large T, in practice we found the deterministic best iterate generally worked better than the T-stochastic solution. Other comparisons were more cloudy, see Section 6 for details.

7.5. On Making Stochastic Classifiers Deterministic

While it is clear that theoretically one needs a stochastic classifier, practitioners may prefer a deterministic classifier. Given that, what is the best way to convert a stochastic classifier into a deterministic one? Recently, Narasimhan et al. (2019b) investigate this question theoretically and experimentally.

7.6. Nonlinear Rate Constraints

We limited our focus to rate constraints that can be written as in Equation 6 as a linear non-negative combination of the positive and negative classification rates on datasets. We touched on the issues posed by nonlinear rate constraints in Section 3 in our discussion of win-loss ratio and precision. As we go to press, newer work shows promise extending these ideas to nonlinear rate constraints (Narasimhan et al., 2019a). However, many open questions remain in handling generalizations of rate constraints, both theoretically and experimentally.

7.7. Rate Constraints For Ranking or Regression Models

Recent work has shown that the presented rate constraint methodology can be intuitively extended to regression and ranking models by defining rate constraints on pairs of examples, forming, for example, analogous *pairwise fairness* definitions (Narasimhan et al., 2020).

7.8. Some Open Theoretical Questions

One open question is how tight our optimality and feasibility guarantees are for our procedures in the following aspects:

- The dependence on the number of iterations T for our guarantees is $O\left(\sqrt{\frac{1}{T}}\right)$. This rate is an artifact of our usage of regret-minimization procedures, but it could be improved through a number of possible techniques, such as variance reduction (e.g. Johnson and Zhang, 2013), or by making stronger assumptions (e.g. strong convexity and/or smoothness).
- The dependence on m, the number of constraints, is $O(\sqrt{m \log m})$, which also comes from the regret-minimization procedures. This is because the λ-player essentially chooses a distribution over $m+1$ actions and this dependence on the number of arms is tight in the context of regret-minimization, but the question remains of whether there are situations where this could be improved upon for constrained optimization for either feasibility or optimality.
- Our results also have a dependence on the model complexity in both feasibility and optimality guarantees. This may be undesirable in models with a large number of parameters, such as modern neural networks. We explored the question of whether we can improve upon this dependence further in follow-up work of Cotter et al. (2019a), which improves the feasibility guarantee. However, further investigation is required to either establish matching lower bounds and/or obtaining tighter results.

Appendix A. Proofs Of Sub{optimality, feasibility} Guarantees

Theorem 9 ***(Lagrangian Sub{optimality,feasibility})*** *Define* $\Lambda = \left\{\lambda \in \mathbb{R}_+^m : \|\lambda\|_p \leq R\right\}$*, and consider the Lagrangian of Equation 2 given in Equation 3. Suppose that* $\theta \in \Theta$ *and* $\lambda \in \Lambda$ *are random variables such that:*

$$\max_{\lambda^* \in \Lambda} \mathbb{E}_\theta \left[\mathcal{L}\left(\theta, \lambda^*\right)\right] - \inf_{\theta^* \in \Theta} \mathbb{E}_\lambda \left[\mathcal{L}\left(\theta^*, \lambda\right)\right] \leq \epsilon, \tag{20}$$

i.e. θ, λ *is an* ϵ*-approximate Nash equilibrium. Then* θ *is* ϵ*-suboptimal:*

$$\mathbb{E}_\theta \left[g_0\left(\theta\right)\right] \leq \inf_{\theta^* \in \Theta : \forall i \in [m], g_i(\theta^*) \leq 0} g_0\left(\theta^*\right) + \epsilon.$$

Furthermore, if λ *is in the interior of* Λ*, in the sense that* $\left\|\bar{\lambda}\right\|_p < R$ *where* $\bar{\lambda} := \mathbb{E}_\lambda\left[\lambda\right]$*, then* θ *is* $\epsilon / \left(R - \left\|\bar{\lambda}\right\|_p\right)$*-feasible:*

$$\left\|\left(\mathbb{E}_\theta\left[g_:\left(\theta\right)\right]\right)_+\right\|_q \leq \frac{\epsilon}{R - \left\|\bar{\lambda}\right\|_p},$$

where $g_:\left(\theta\right)$ *is the* m*-dimensional vector of constraint evaluations, and* $\left(\cdot\right)_+$ *takes the positive part of its argument, so that* $\left\|\left(\mathbb{E}_\theta\left[g_:\left(\theta\right)\right]\right)_+\right\|_q$ *is the* q*-norm of the vector of expected constraint violations.*

Proof First notice that $\mathcal{L}$ is linear in λ, so:

$$\max_{\lambda^* \in \Lambda} \mathbb{E}_\theta \left[\mathcal{L}\left(\theta, \lambda^*\right)\right] - \inf_{\theta^* \in \Theta} \mathcal{L}\left(\theta^*, \bar{\lambda}\right) \leq \epsilon. \tag{21}$$

Optimality: Choose θ^* to be the optimal *feasible* solution in Equation 21, so that $g_i\left(\theta^*\right) \leq 0$ for all $i \in [m]$, and also choose $\lambda^* = 0$, which combined with the definition of $\mathcal{L}$ (Equation 3) gives that:

$$\mathbb{E}_\theta \left[g_0\left(\theta\right)\right] - g_0\left(\theta^*\right) \leq \epsilon,$$

which is the optimality claim.

Feasibility: Choose $\theta^* = \theta$ in Equation 21. By the definition of $\mathcal{L}$ (Equation 3):

$$\max_{\lambda^* \in \Lambda} \sum_{i=1}^m \lambda_i^* \mathbb{E}_\theta \left[g_i\left(\theta\right)\right] - \sum_{i=1}^m \bar{\lambda}_i \mathbb{E}_\theta \left[g_i\left(\theta\right)\right] \leq \epsilon.$$

Then by the definition of a dual norm, Hölder's inequality, and the assumption that $\left\|\bar{\lambda}\right\|_p < R$:

$$R \left\|\left(\mathbb{E}_\theta \left[g_:\left(\theta\right)\right]\right)_+\right\|_q - \left\|\bar{\lambda}\right\|_p \left\|\left(\mathbb{E}_\theta \left[g_:\left(\theta\right)\right]\right)_+\right\|_q \leq \epsilon.$$

Rearranging terms gives the feasibility claim. ■

Lemma 10 *In the context of Theorem 9, suppose that there exists a $\theta' \in \Theta$ that satisfies all of the constraints, and does so with q-norm margin γ, i.e. $g_i\left(\theta'\right) \leq 0$ for all $i \in [m]$ and $\left\|g_:\left(\theta'\right)\right\|_q \geq \gamma$. Then:*

$$\left\|\bar{\lambda}\right\|_p \leq \frac{\epsilon + B_{g_0}}{\gamma},$$

where $B_{g_0} \geq \sup_{\theta \in \Theta} g_0\left(\theta\right) - \inf_{\theta \in \Theta} g_0\left(\theta\right)$ is a bound on the range of the objective function g_0.

Proof Starting from Equation 20 (in Theorem 9), and choosing $\theta^* = \theta'$ and $\lambda^* = 0$:

$$\begin{aligned}
\epsilon \geq & \mathbb{E}_\theta \left[g_0\left(\theta\right)\right] - \mathbb{E}_\lambda \left[g_0\left(\theta'\right) + \sum_{i=1}^m \lambda_i g_i\left(\theta'\right)\right] \\
\epsilon \geq & \mathbb{E}_\theta \left[g_0\left(\theta\right) - \inf_{\theta' \in \Theta} g_0\left(\theta'\right)\right] - \left(g_0\left(\theta'\right) - \inf_{\theta' \in \Theta} g_0\left(\theta'\right)\right) + \gamma \left\|\bar{\lambda}\right\|_p \\
\epsilon \geq & - B_{g_0} + \gamma \left\|\bar{\lambda}\right\|_p .
\end{aligned}$$

Solving for $\left\|\bar{\lambda}\right\|_p$ yields the claim. ■

We next give the optimality and feasibility guarantees for the proxy-Lagrangian formulation. The result shows that the approximate semi-coarse correlated equilibrium to the two-player non-zero sum game based on the proxy-Lagrangian will correspond to an approximately feasible solution to the constrained optimization problem w.r.t. the original constraints which is also approximately optimal compared to the solution which is optimal and feasible w.r.t. the proxy constraints. The conditions for semi-coarse correlated equilibrium are shown in Equation 22. The first line requires that the

solution is approximately as good for the θ-player compared to any fixed choice of θ (i.e. which comes from the θ-player using best-response or minimizing external-regret). The second line requires that the solution is approximately as good for the λ-player when compared to any left-stochastic linear transformation of that solution for the λ-player (which is a result from the λ-player optimizing for swap-regret).

The swap-regret guarantee is required to show feasibility. The key idea is to use the swap regret guarantee to show that the difference in the proxy-Lagrangian when shifting the weight λ_1 on the objective g_0 to any of the constraints $g_1, ..., g_m$ will result in only a small change, and thus the constraint violations themselves are small.

Theorem 11 ***(Proxy-Lagrangian Sub{optimality,feasibility})*** *Let*

$$\mathcal{M} := \left\{M \in \mathbb{R}^{(m+1)\times(m+1)} : \forall i \in [m+1] . M_{:,i} \in \Delta^{m+1}\right\}$$

be the set of all left-stochastic $(m+1) \times (m+1)$ matrices, and consider the "proxy-Lagrangians" of Equation 2 given in Equation 15. Suppose that $\theta \in \Theta$ and $\lambda \in \Lambda$ are jointly distributed random variables such that:

$$\begin{aligned}
\mathbb{E}_{\theta,\lambda} \left[\mathcal{L}_\theta\left(\theta, \lambda\right)\right] - \inf_{\theta^* \in \Theta} \mathbb{E}_\lambda \left[\mathcal{L}_\theta\left(\theta^*, \lambda\right)\right] \leq & \epsilon_\theta \\
\max_{M^* \in \mathcal{M}} \mathbb{E}_{\theta,\lambda} \left[\mathcal{L}_\lambda\left(\theta, M^*\lambda\right)\right] - \mathbb{E}_{\theta,\lambda} \left[\mathcal{L}_\lambda\left(\theta, \lambda\right)\right] \leq & \epsilon_\lambda.
\end{aligned} \tag{22}$$

Define $\bar{\lambda} := \mathbb{E}_\lambda \left[\lambda\right]$, let $(\Omega, \mathcal{F}, P)$ be the probability space, and define a random variable $\bar{\theta}$ such that:

$$\Pr\left\{\bar{\theta} \in S\right\} = \frac{\int_{\theta^{-1}(S)} \lambda_1\left(x\right) dP\left(x\right)}{\int_\Omega \lambda_1\left(x\right) dP\left(x\right)}.$$

In words, $\bar{\theta}$ is a version of θ that has been resampled with λ_1 being treated as an importance weight. In particular $\mathbb{E}_{\bar{\theta}} \left[f\left(\bar{\theta}\right)\right] = \mathbb{E}_{\theta,\lambda} \left[\lambda_1 f\left(\theta\right)\right] / \bar{\lambda}_1$ for any $f : \Theta \to \mathbb{R}$. Then $\bar{\theta}$ is nearly-optimal:

$$\mathbb{E}_{\bar{\theta}} \left[g_0\left(\bar{\theta}\right)\right] \leq \inf_{\theta^* \in \Theta : \forall i \in [m] . \tilde{g}_i(\theta^*) \leq 0} g_0\left(\theta^*\right) + \frac{\epsilon_\theta + \epsilon_\lambda}{\bar{\lambda}_1},$$

and nearly-feasible:

$$\left\|\left(\mathbb{E}_{\bar{\theta}} \left[g_:\left(\bar{\theta}\right)\right]\right)_+\right\|_\infty \leq \frac{\epsilon_\lambda}{\bar{\lambda}_1}.$$

Notice the optimality inequality is weaker than it may appear, since the comparator in this equation is not *the optimal solution w.r.t. the constraints g_i, but rather w.r.t. the* proxy *constraints $\tilde{g}_i$.*

Proof Optimality: If we choose M^* to be the matrix with its first row being all-one, and all other rows being all-zero, then $\mathcal{L}_\lambda\left(\theta, M^*\lambda\right) = 0$, which shows that the first term in the LHS of the second line of Equation 22 is nonnegative. Hence, $-\mathbb{E}_{\theta,\lambda} \left[\mathcal{L}_\lambda\left(\theta, \lambda\right)\right] \leq \epsilon_\lambda$, so by the definition of $\mathcal{L}_\lambda$ (Equation 15), and the fact that $\tilde{g}_i \geq g_i$:

$$\mathbb{E}_{\theta,\lambda} \left[\sum_{i=1}^m \lambda_{i+1} \tilde{g}_i\left(\theta\right)\right] \geq -\epsilon_\lambda.$$

Notice that $\mathcal{L}_\theta$ is linear in λ, so the first line of Equation 22, combined with the above result and the definition of $\mathcal{L}_\theta$ (Equation 15) becomes:

$$\mathbb{E}_{\theta,\lambda}\left[\lambda_1 g_0(\theta)\right] - \inf_{\theta^*\in\Theta}\left(\bar{\lambda}_1 g_0(\theta^*) + \sum_{i=1}^{m}\bar{\lambda}_{i+1}\tilde{g}_i(\theta^*)\right) \leq \epsilon_\theta + \epsilon_\lambda. \tag{23}$$

Choose θ^* to be the optimal solution that satisfies the *proxy* constraints $\tilde{g}$, so that $\tilde{g}_i(\theta^*) \leq 0$ for all $i \in [m]$. Hence:

$$\mathbb{E}_{\theta,\lambda}\left[\lambda_1 g_0(\theta)\right] - \bar{\lambda}_1 g_0(\theta^*) \leq \epsilon_\theta + \epsilon_\lambda,$$

which is the optimality claim.

Feasibility: We'll simplify our notation by defining $\ell_1(\theta) := 0$ and $\ell_{i+1}(\theta) := g_i(\theta)$ for $i \in [m]$, so that $\mathcal{L}_\lambda(\theta,\lambda) = \langle \lambda, \ell_:(\theta)\rangle$. Consider the first term in the LHS of the second line of Equation 22:

$$\begin{aligned}
\max_{M^*\in\mathcal{M}} \mathbb{E}_{\theta,\lambda}\left[\mathcal{L}_\lambda(\theta, M^*\lambda)\right] &= \max_{M^*\in\mathcal{M}} \mathbb{E}_{\theta,\lambda}\left[\langle M^*\lambda, \ell_:(\theta)\rangle\right] \\
&= \max_{M^*\in\mathcal{M}} \mathbb{E}_{\theta,\lambda}\left[\sum_{i=1}^{m+1}\sum_{j=1}^{m+1} M^*_{j,i}\lambda_i \ell_j(\theta)\right] \\
&= \sum_{i=1}^{m+1} \max_{M^*_{:,i}\in\Delta^{m+1}} \sum_{j=1}^{m+1} \mathbb{E}_{\theta,\lambda}\left[M^*_{j,i}\lambda_i\ell_j(\theta)\right] \\
&= \sum_{i=1}^{m+1} \max_{j\in[m+1]} \mathbb{E}_{\theta,\lambda}\left[\lambda_i \ell_j(\theta)\right],
\end{aligned}$$

where we used the fact that, since M^* is left-stochastic, each of its columns is a $(m+1)$-dimensional multinoulli distribution. For the second term in the LHS of the second line of Equation 22, we can use the fact that $\ell_1(\theta) = 0$:

$$\mathbb{E}_{\theta,\lambda}\left[\sum_{i=2}^{m+1}\lambda_i\ell_i(\theta)\right] \leq \sum_{i=2}^{m+1}\max_{j\in[m+1]}\mathbb{E}_{\theta,\lambda}\left[\lambda_i\ell_j(\theta)\right].$$

Plugging these two results into the second line of Equation 22, the two sums collapse, leaving:

$$\max_{i\in[m+1]}\mathbb{E}_{\theta,\lambda}\left[\lambda_1\ell_i(\theta)\right] \leq \epsilon_\lambda.$$

By the definition of ℓ_i, and the fact that $\ell_1 = 0$:

$$\left\|\left(\mathbb{E}_{\theta,\lambda}\left[\lambda_1 g_:(\theta)\right]\right)_+\right\|_\infty \leq \epsilon_\lambda,$$

which is the feasibility claim. ■

Lemma 12 *In the context of Theorem 11, suppose that there exists a $\theta' \in \Theta$ that satisfies all of the* proxy *constraints with margin γ, i.e. $\tilde{g}_i(\theta') \leq -\gamma$ for all $i \in [m]$. Then:*

$$\bar{\lambda}_1 \geq \frac{\gamma - \epsilon_\theta - \epsilon_\lambda}{\gamma + B_{g_0}},$$

where $B_{g_0} \geq \sup_{\theta\in\Theta} g_0(\theta) - \inf_{\theta\in\Theta} g_0(\theta)$ is a bound on the range of the objective function g_0.

Proof Starting from Equation 23 (in the proof of Theorem 11), and choosing $\theta^* = \theta'$:

$$\mathbb{E}_{\theta,\lambda}\left[\lambda_1 g_0(\theta)\right] - \left(\bar{\lambda}_1 g_0(\theta') + \sum_{i=1}^{m}\bar{\lambda}_{i+1}\tilde{g}_i(\theta')\right) \leq \epsilon_\theta + \epsilon_\lambda.$$

Since $\tilde{g}_i(\theta') \leq -\gamma$ for all $i \in [m]$:

$$\begin{aligned}
\epsilon_\theta + \epsilon_\lambda \geq& \mathbb{E}_{\theta,\lambda}\left[\lambda_1 g_0(\theta)\right] - \bar{\lambda}_1 g_0(\theta') + \left(1-\bar{\lambda}_1\right)\gamma \\
\geq& \mathbb{E}_{\theta,\lambda}\left[\lambda_1\left(g_0(\theta) - \inf_{\theta'\in\Theta} g_0(\theta')\right)\right] - \bar{\lambda}_1\left(g_0(\theta') - \inf_{\theta'\in\Theta} g_0(\theta')\right) + \left(1-\bar{\lambda}_1\right)\gamma \\
\geq& -\bar{\lambda}_1 B_{g_0} + \left(1-\bar{\lambda}_1\right)\gamma.
\end{aligned}$$

Solving for $\bar{\lambda}_1$ yields the claim. ■

Appendix B. Proofs Of Existence Of Sparse Equilibria

Theorem 13 *Consider a two player game, played on the compact Hausdorff spaces Θ and $\Lambda \subseteq \mathbb{R}^m$. Imagine that the θ-player wishes to minimize $\mathcal{L}_\theta : \Theta\times\Lambda \to \mathbb{R}$, and the λ-player wishes to maximize $\mathcal{L}_\lambda : \Theta\times\Lambda\to\mathbb{R}$, with both of these functions being continuous in θ and linear in λ. Then there exists a Nash equilibrium θ, λ:*

$$\begin{aligned}
\mathbb{E}_\theta\left[\mathcal{L}_\theta(\theta,\lambda)\right] &= \min_{\theta^*\in\Theta}\mathcal{L}_\theta(\theta^*,\lambda) \\
\mathbb{E}_\theta\left[\mathcal{L}_\lambda(\theta,\lambda)\right] &= \max_{\lambda^*\in\Lambda}\mathbb{E}_\theta\left[\mathcal{L}_\lambda(\theta,\lambda^*)\right].
\end{aligned}$$

where θ is a random variable placing nonzero probability mass on at most $m+1$ *elements of Θ, and $\lambda \in \Lambda$ is non-random.*

Proof There are some extremely similar (and in some ways more general) results than this in the game theory literature (e.g. Bohnenblust et al., 1950; Parthasarathy, 1975), but for our particular (Lagrangian and proxy-Lagrangian) setting it's possible to provide a fairly straightforward proof.

To begin with, Glicksberg (1952) gives that there exists a mixed strategy in the form of two random variables $\tilde{\theta}$ and $\tilde{\lambda}$:

$$\begin{aligned}
\mathbb{E}_{\tilde{\theta},\tilde{\lambda}}\left[\mathcal{L}_\theta\left(\tilde{\theta},\tilde{\lambda}\right)\right] &= \min_{\theta^*\in\Theta}\mathbb{E}_{\tilde{\lambda}}\left[\mathcal{L}_\theta\left(\theta^*,\tilde{\lambda}\right)\right] \\
\mathbb{E}_{\tilde{\theta},\tilde{\lambda}}\left[\mathcal{L}_\lambda\left(\tilde{\theta},\tilde{\lambda}\right)\right] &= \max_{\lambda^*\in\Lambda}\mathbb{E}_{\tilde{\theta}}\left[\mathcal{L}_\lambda\left(\tilde{\theta},\lambda^*\right)\right].
\end{aligned}$$

Since both functions are linear in $\tilde{\lambda}$, we can define $\lambda := \mathbb{E}_{\tilde{\lambda}}\left[\tilde{\lambda}\right]$, and these conditions become:

$$\mathbb{E}_{\tilde{\theta}}\left[\mathcal{L}_\theta\left(\tilde{\theta},\lambda\right)\right] = \min_{\theta^*\in\Theta}\mathcal{L}_\theta\left(\theta^*,\lambda\right) := \ell_{\min}$$
$$\mathbb{E}_{\tilde{\theta}}\left[\mathcal{L}_\lambda\left(\tilde{\theta},\lambda\right)\right] = \max_{\lambda^*\in\Lambda}\mathbb{E}_{\tilde{\theta}}\left[\mathcal{L}_\lambda\left(\tilde{\theta},\lambda^*\right)\right].$$

Let's focus on the first condition. Let $p_\epsilon := \Pr\left\{\mathcal{L}_\theta\left(\tilde{\theta},\lambda\right) \geq \ell_{\min}+\epsilon\right\}$, and notice that $p_{1/n}$ must equal zero for any $n \in \{1,2,\dots\}$ (otherwise we would contradict the above), implying by the countable additivity of measures that $\Pr\left\{\mathcal{L}_\theta\left(\tilde{\theta},\lambda\right) = \ell_{\min}\right\} = 1$. We therefore assume henceforth, without loss of generality, that the support of $\tilde{\theta}$ consists entirely of minimizers of $\mathcal{L}_\theta\left(\cdot,\lambda\right)$. Let $S \subseteq \Theta$ be this support set.

Define $G := \left\{\nabla_{\tilde{\lambda}}\mathcal{L}_\lambda\left(\theta',\lambda\right) : \theta' \in S\right\}$, and take $\bar{G}$ to be the closure of the convex hull of G. Since $\mathbb{E}_{\tilde{\theta}}\left[\nabla_{\tilde{\lambda}}\mathcal{L}_\lambda\left(\tilde{\theta},\lambda\right)\right] \in \bar{G} \subseteq \mathbb{R}^m$, we can write it as a convex combination of at most $m+1$ extreme points of $\bar{G}$, or equivalently of $m+1$ elements of G. Hence, we can take θ to be a discrete random variable that places nonzero mass on at most $m+1$ elements of S, and:

$$\mathbb{E}_\theta\left[\nabla_{\tilde{\lambda}}\mathcal{L}_\lambda\left(\theta,\lambda\right)\right] = \mathbb{E}_{\tilde{\theta}}\left[\nabla_{\tilde{\lambda}}\mathcal{L}_\lambda\left(\tilde{\theta},\lambda\right)\right].$$

Linearity in λ then implies that $\mathbb{E}_\theta\left[\mathcal{L}_\lambda\left(\theta,\cdot\right)\right]$ and $\mathbb{E}_{\tilde{\theta}}\left[\mathcal{L}_\lambda\left(\tilde{\theta},\cdot\right)\right]$ are the *same function* up to a constant, and therefore have the same maximizer(s). Correspondingly, θ is supported on S, which contains only minimizers of $\mathcal{L}_\theta\left(\cdot,\lambda\right)$ by construction. ■

Lemma 14 *If Θ is a compact Hausdorff space and the objective, constraint and proxy constraint functions $g_0, g_1, \dots, g_m, \tilde{g}_1, \dots, \tilde{g}_m$ are continuous, then the proxy-Lagrangian game (Equation 15) has a mixed Nash equilibrium pair (θ,λ) where θ is a random variable supported on at most $m+1$ elements of Θ, and λ is non-random.*

Proof Applying Theorem 13 directly would result in a support size of $m+2$, rather than the desired $m+1$, since Λ is $(m+1)$-dimensional. Instead, we define $\tilde{\Lambda} = \left\{\tilde{\lambda} \in \mathbb{R}^m_+ : \left\|\tilde{\lambda}\right\|_1 \leq 1\right\}$ as the space containing the last m coordinates of Λ. Then we can rewrite the proxy-Lagrangian functions $\tilde{\mathcal{L}}_\theta, \tilde{\mathcal{L}}_\lambda : \Theta \times \tilde{\Lambda} \to \mathbb{R}$ as:

$$\tilde{\mathcal{L}}_\theta\left(\theta,\tilde{\lambda}\right) = \left(1-\left\|\tilde{\lambda}\right\|_1\right)g_0\left(\theta\right) + \sum_{i=1}^m \tilde{\lambda}_i\tilde{g}_i\left(\theta\right)$$
$$\tilde{\mathcal{L}}_\lambda\left(\theta,\tilde{\lambda}\right) = \sum_{i=1}^m \tilde{\lambda}_i g_i\left(\theta\right).$$

These functions are linear in $\tilde{\lambda}$, which is a m-dimensional space, so the conditions of Theorem 13 apply, yielding the claimed result. ■

Proof [Proof of Lemma 5] The linear program contains not only the m explicit linearized functional constraints, but also, since $p \in \Delta^T$, the T nonnegativity constraints $p_t \geq 0$, and the sum-to-one constraint $\sum_{t=1}^T p_t = 1$.

Since p is T-dimensional, every vertex p^* of the feasible region must include T active constraints. Letting $m^* \leq m$ be the number of active linearized functional constraints, and accounting for the sum-to-one constraint, it follows that at least $T-m^*-1$ nonnegativity constraints are active, implying that p^* contains at most m^*+1 nonzero elements. ■

Appendix C. Proofs Of Convergence Rates

C.1. Non-Stochastic One-Player Convergence Rates

Theorem 15 ***(Mirror Descent)*** *Let $f_1, f_2, \dots : \Theta \to \mathbb{R}$ be a sequence of convex functions that we wish to minimize on a compact convex set Θ. Suppose that the "distance generating function" $\Psi : \Theta \to \mathbb{R}_+$ is nonnegative and 1-strongly convex w.r.t. a norm $\|\cdot\|$ with dual norm $\|\cdot\|_*$.*

Define the step size $\eta = \sqrt{B_\Psi / T B_{\check{\nabla}}^2}$, where $B_\Psi \geq \max_{\theta\in\Theta}\Psi\left(\theta\right)$ is a uniform upper bound on Ψ, and $B_{\check{\nabla}} \geq \left\|\check{\nabla} f_t\left(\theta^{(t)}\right)\right\|_*$ is a uniform upper bound on the norms of the subgradients. Suppose that we perform T iterations of the following update, starting from $\theta^{(1)} = \operatorname{argmin}_{\theta\in\Theta}\Psi\left(\theta\right)$:

$$\tilde{\theta}^{(t+1)} = \nabla\Psi^*\left(\nabla\Psi\left(\theta^{(t)}\right) - \eta\check{\nabla} f_t\left(\theta^{(t)}\right)\right)$$
$$\theta^{(t+1)} = \operatorname*{argmin}_{\theta\in\Theta} D_\Psi\left(\theta \mid \tilde{\theta}^{(t+1)}\right),$$

where $\check{\nabla} f_t\left(\theta\right) \in \partial f_t(\theta^{(t)})$ is a subgradient of f_t at θ, and $D_\Psi\left(\theta \mid \theta'\right) := \Psi\left(\theta\right) - \Psi\left(\theta'\right) - \langle\nabla\Psi\left(\theta'\right), \theta-\theta'\rangle$ is the Bregman divergence associated with Ψ. Then:

$$\frac{1}{T}\sum_{t=1}^T f_t\left(\theta^{(t)}\right) - \frac{1}{T}\sum_{t=1}^T f_t\left(\theta^*\right) \leq 2B_{\check{\nabla}}\sqrt{\frac{B_\Psi}{T}},$$

where $\theta^* \in \Theta$ is an arbitrary reference vector.

Proof Mirror descent (Nemirovski and Yudin, 1983; Beck and Teboulle, 2003) dates back to 1983, but this particular statement is taken from Lemma 2 of Srebro et al. (2011). ■

Corollary 16 ***(Gradient Descent)*** *Let $f_1, f_2, \dots : \Theta \to \mathbb{R}$ be a sequence of convex functions that we wish to minimize on a compact convex set Θ.*

Define the step size $\eta = B_\Theta / B_{\check{\nabla}}\sqrt{2T}$, where $B_\Theta \geq \max_{\theta\in\Theta}\|\theta\|_2$, and $B_{\check{\nabla}} \geq \left\|\check{\nabla} f_t\left(\theta^{(t)}\right)\right\|_2$ is a uniform upper bound on the norms of the subgradients. Suppose that we perform T iterations of the following update, starting from $\theta^{(1)} = \operatorname{argmin}_{\theta\in\Theta}\|\theta\|_2$:

$$\theta^{(t+1)} = \Pi_\Theta\left(\theta^{(t)} - \eta\check{\nabla} f_t\left(\theta^{(t)}\right)\right),$$

where $\check{\nabla} f_t(\theta) \in \partial f_t(\theta^{(t)})$ is a subgradient of f_t at θ, and Π_Θ projects its argument onto Θ w.r.t. the Euclidean norm. Then:

$$\frac{1}{T}\sum_{t=1}^{T} f_t\left(\theta^{(t)}\right) - \frac{1}{T}\sum_{t=1}^{T} f_t\left(\theta^*\right) \leq B_\Theta B_{\check{\nabla}}\sqrt{\frac{2}{T}},$$

where $\theta^* \in \Theta$ is an arbitrary reference vector.

Proof Follows from taking $\Psi(\theta) = \|\theta\|_2^2/2$ in Theorem 15. ■

Corollary 17 *Let $\mathcal{M} := \left\{M \in \mathbb{R}^{\tilde{m}\times\tilde{m}} : \forall i \in [\tilde{m}]\, .M_{:,i} \in \Delta^{\tilde{m}}\right\}$ be the set of all left-stochastic $\tilde{m}\times\tilde{m}$ matrices, and let $f_1, f_2, \ldots : \mathcal{M} \to \mathbb{R}$ be a sequence of concave functions that we wish to maximize.*

Define the step size $\eta = \sqrt{\tilde{m}\ln\tilde{m}/TB_{\hat{\nabla}}^2}$, where $B_{\hat{\nabla}} \geq \left\|\hat{\nabla} f_t\left(M^{(t)}\right)\right\|_{\infty,2}$ is a uniform upper bound on the norms of the supergradients, and $\|\cdot\|_{\infty,2} := \sqrt{\sum_{i=1}^{\tilde{m}}\|M_{:,i}\|_\infty^2}$ is the $L_{\infty,2}$ matrix norm. Suppose that we perform T iterations of the following update starting from the matrix $M^{(1)}$ with all elements equal to $1/\tilde{m}$:

$$\begin{aligned}\tilde{M}^{(t+1)} =& M^{(t)} \odot .\exp\left(\eta\hat{\nabla} f_t\left(M^{(t)}\right)\right)\\ M_{:,i}^{(t+1)} =& \tilde{M}_{:,i}^{(t+1)} / \left\|\tilde{M}_{:,i}^{(t+1)}\right\|_1,\end{aligned}$$

where $-\hat{\nabla} f_t\left(M^{(t)}\right) \in \partial\left(-f_t(M^{(t)})\right)$, i.e. $\hat{\nabla} f_t\left(M^{(t)}\right)$ is a supergradient of f_t at $M^{(t)}$, and the multiplication and exponentiation in the first step are performed element-wise. Then:

$$\frac{1}{T}\sum_{t=1}^{T} f_t\left(M^*\right) - \frac{1}{T}\sum_{t=1}^{T} f_t\left(M^{(t)}\right) \leq 2B_{\hat{\nabla}}\sqrt{\frac{\tilde{m}\ln\tilde{m}}{T}},$$

where $M^* \in \mathcal{M}$ is an arbitrary reference matrix.

Proof Define $\Psi : \mathcal{M} \to \mathbb{R} := \tilde{m}\ln\tilde{m} + \sum_{i,j\in[\tilde{m}]} M_{i,j}\ln M_{i,j}$ as $\tilde{m}\ln\tilde{m}$ plus the negative Shannon entropy, applied to its (matrix) argument element-wise ($\tilde{m}\ln\tilde{m}$ is added to make Ψ nonnegative on $\mathcal{M}$). As in the vector setting, the resulting mirror descent update will be (element-wise) multiplicative.

The Bregman divergence satisfies:

$$\begin{aligned}D_\Psi\left(M|M'\right) =& \Psi(M) - \Psi\left(M'\right) - \left\langle\nabla\Psi\left(M'\right), M - M'\right\rangle\\ =& \left\|M'\right\|_{1,1} - \|M\|_{1,1} + \sum_{i=1}^{\tilde{m}} D_{KL}\left(M_{:,i}\|M'_{:,i}\right),\end{aligned} \tag{24}$$

where $\|M\|_{1,1} = \sum_{i=1}^{\tilde{m}}\|M_{:,i}\|_1$ is the $L_{1,1}$ matrix norm. This incidentally shows that one projects onto $\mathcal{M}$ w.r.t. D_Ψ by projecting each column w.r.t. the KL divergence, i.e. by normalizing the columns.

By Pinsker's inequality (applied to each column of an $M \in \mathcal{M}$):

$$\left\|M - M'\right\|_{1,2}^2 \leq 2\sum_{i=1}^{\tilde{m}} D_{KL}\left(M_{:,i}\|M'_{:,i}\right),$$

where $\|M\|_{1,2} = \sqrt{\sum_{i=1}^{\tilde{m}}\|M_{:,i}\|_1^2}$ is the $L_{1,2}$ matrix norm. Substituting this into Equation 24, and using the fact that $\|M\|_{1,1} = \tilde{m}$ for all $M \in \mathcal{M}$, we have that for all $M, M' \in \mathcal{M}$:

$$D_\Psi\left(M|M'\right) \geq \frac{1}{2}\left\|M - M'\right\|_{1,2}^2,$$

which shows that Ψ is 1-strongly convex w.r.t. the $L_{1,2}$ matrix norm. The dual norm of the $L_{1,2}$ matrix norm is the $L_{\infty,2}$ norm, which is the last piece needed to apply Theorem 15, yielding the claimed result. ■

Lemma 18 *Let $\Lambda := \Delta^{\tilde{m}}$ be the $\tilde{m}$-dimensional simplex, define*

$$\mathcal{M} := \left\{M \in \mathbb{R}^{\tilde{m}\times\tilde{m}} : \forall i \in [\tilde{m}]\, .M_{:,i} \in \Delta^{\tilde{m}}\right\}$$

as the set of all left-stochastic $\tilde{m}\times\tilde{m}$ matrices, and take $f_1, f_2, \ldots : \Lambda \to \mathbb{R}$ to be a sequence of concave functions that we wish to maximize.

Define the step size $\eta = \sqrt{\tilde{m}\ln\tilde{m}/TB_{\hat{\nabla}}^2}$, where $B_{\hat{\nabla}} \geq \left\|\hat{\nabla} f_t\left(\lambda^{(t)}\right)\right\|_\infty$ is an uniform upper bound on the ∞-norms of the supergradients. Suppose that we perform T iterations of the following update, starting from the matrix $M^{(1)}$ with all elements equal to $1/\tilde{m}$:

$$\begin{aligned}&\lambda^{(t)} \text{ is any stationary distribution of } M^{(t)}\\ A^{(t)} =& \left(\hat{\nabla} f_t\left(\lambda^{(t)}\right)\right)\left(\lambda^{(t)}\right)^T\\ \tilde{M}^{(t+1)} =& M^{(t)} \odot .\exp\left(\eta A^{(t)}\right)\\ M_{:,i}^{(t+1)} =& \tilde{M}_{:,i}^{(t+1)} / \left\|\tilde{M}_{:,i}^{(t+1)}\right\|_1,\end{aligned}$$

where a stationary distribution of M (i.e. a $\lambda \in \Lambda$ such that $M\lambda = \lambda$) always exists because M is left-stochastic, $-\hat{\nabla} f_t\left(\lambda^{(t)}\right) \in \partial\left(-f_t(\lambda^{(t)})\right)$, i.e. $\hat{\nabla} f_t\left(\lambda^{(t)}\right)$ is a supergradient of f_t at $\lambda^{(t)}$, and the multiplication and exponentiation of the third step are performed element-wise. Then:

$$\frac{1}{T}\sum_{t=1}^{T} f_t\left(M^*\lambda^{(t)}\right) - \frac{1}{T}\sum_{t=1}^{T} f_t\left(\lambda^{(t)}\right) \leq 2B_{\hat{\nabla}}\sqrt{\frac{\tilde{m}\ln\tilde{m}}{T}},$$

where $M^* \in \mathcal{M}$ is an arbitrary left-stochastic reference matrix.

Proof This algorithm is an instance of that contained in Figure 1 of Gordon et al. (2008).

Define $\tilde{f}_t(M) := f_t\left(M^{(t)}\lambda^{(t)}\right)$. Observe that since $\hat{\nabla} f_t\left(\lambda^{(t)}\right)$ is a supergradient of f_t at $\lambda^{(t)}$, and $M^{(t)}\lambda^{(t)} = \lambda^{(t)}$:

$$\begin{aligned}f_t\left(\tilde{M}\lambda^{(t)}\right) \leq& f_t\left(M^{(t)}\lambda^{(t)}\right) + \left\langle\hat{\nabla} f_t\left(\lambda^{(t)}\right), \tilde{M}\lambda^{(t)} - M^{(t)}\lambda^{(t)}\right\rangle\\ \leq& f_t\left(M^{(t)}\lambda^{(t)}\right) + A^{(t)}\cdot\left(\tilde{M} - M^{(t)}\right),\end{aligned}$$

where the matrix product on the last line is performed element-wise. This shows that $A^{(t)}$ is a supergradient of $\tilde{f}_t$ at $M^{(t)}$, from which we conclude that the final two steps of the update are performing the algorithm of Corollary 17, so:

$$\frac{1}{T}\sum_{t=1}^{T}\tilde{f}_t\left(M^*\right)-\frac{1}{T}\sum_{t=1}^{T}\tilde{f}_t\left(M^{(t)}\right)\leq 2B_{\tilde{\nabla}}\sqrt{\frac{\tilde{m}\ln\tilde{m}}{T}},$$

where the $B_{\tilde{\nabla}}$ of Corollary 17 is a uniform upper bound on the $L_{\infty,2}$ matrix norms of the $A^{(t)}$s. However, by the definition of $A^{(t)}$ and the fact that $\lambda^{(t)}\in\Delta^{\tilde{m}}$, we can instead take $B_{\tilde{\nabla}}$ to be a uniform upper bound on $\left\|\hat{\nabla}^{(t)}\right\|_\infty$. Substituting the definition of $\tilde{f}_t$ and again using the fact that $M^{(t)}\lambda^{(t)}=\lambda^{(t)}$ then yields the claimed result. ■

C.2. Stochastic One-Player Convergence Rates

Theorem 19 ***(Stochastic Mirror Descent)*** *Let Ψ, $\|\cdot\|$, D_Ψ and B_Ψ be as in Theorem 15, and let $f_1,f_2,\ldots:\Theta\to\mathbb{R}$ be a sequence of convex functions that we wish to minimize on a compact convex set Θ.*

Define the step size $\eta=\sqrt{B_\Psi/TB_{\check{\Delta}}^2}$, where $B_{\check{\Delta}}\geq\left\|\check{\Delta}^{(t)}\right\|_*$ is a uniform upper bound on the norms of the stochastic subgradients. Suppose that we perform T iterations of the following stochastic *update, starting from $\theta^{(1)}=\operatorname{argmin}_{\theta\in\Theta}\Psi\left(\theta\right)$:*

$$\tilde{\theta}^{(t+1)}=\nabla\Psi^*\left(\nabla\Psi\left(\theta^{(t)}\right)-\eta\check{\Delta}^{(t)}\right)$$
$$\theta^{(t+1)}=\operatorname*{argmin}_{\theta\in\Theta}D_\Psi\left(\theta|\tilde{\theta}^{(t+1)}\right),$$

where $\mathbb{E}\left[\check{\Delta}^{(t)}\mid\theta^{(t)}\right]\in\partial f_t(\theta^{(t)})$, i.e. $\check{\Delta}^{(t)}$ is a stochastic subgradient of f_t at $\theta^{(t)}$. Then, with probability $1-\delta$ over the draws of the stochastic subgradients:

$$\frac{1}{T}\sum_{t=1}^{T}f_t\left(\theta^{(t)}\right)-\frac{1}{T}\sum_{t=1}^{T}f_t\left(\theta^*\right)\leq 2B_{\check{\nabla}}\sqrt{\frac{2B_\Psi\left(1+16\ln\frac{1}{\delta}\right)}{T}},$$

where $\theta^*\in\Theta$ is an arbitrary reference vector.

Proof This is nothing more than the usual transformation of a uniform regret guarantee into a stochastic one via the Hoeffding-Azuma inequality—we include a proof for completeness.

Define the sequence:

$$\tilde{f}_t\left(\theta\right)=f_t\left(\theta^{(t)}\right)+\left\langle\check{\Delta}^{(t)},\theta-\theta^{(t)}\right\rangle.$$

Then applying non-stochastic mirror descent to the sequence $\tilde{f}_t$ will result in exactly the same sequence of iterates $\theta^{(t)}$ as applying stochastic mirror descent (above) to f_t. Hence, by Theorem 15

and the definition of $\tilde{f}_t$ (notice that we can take $B_{\check{\nabla}}=B_{\check{\Delta}}$):

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\tilde{f}_t\left(\theta^{(t)}\right)-\frac{1}{T}\sum_{t=1}^{T}\tilde{f}_t\left(\theta^*\right)\leq&2B_{\check{\nabla}}\sqrt{\frac{B_\Psi}{T}}\\ \frac{1}{T}\sum_{t=1}^{T}f_t\left(\theta^{(t)}\right)-\frac{1}{T}\sum_{t=1}^{T}f_t\left(\theta^*\right)\leq&2B_{\check{\nabla}}\sqrt{\frac{B_\Psi}{T}}+\frac{1}{T}\sum_{t=1}^{T}\left(\tilde{f}_t\left(\theta^*\right)-f_t\left(\theta^*\right)\right)\\ \leq&2B_{\check{\nabla}}\sqrt{\frac{B_\Psi}{T}}+\frac{1}{T}\sum_{t=1}^{T}\left\langle\check{\Delta}^{(t)}-\check{\nabla}f_t\left(\theta^{(t)}\right),\theta^*-\theta^{(t)}\right\rangle,\end{aligned}\tag{25}$$

where the last step follows from the convexity of the f_ts. Consider the second term on the RHS. Observe that, since the $\check{\Delta}^{(t)}$s are stochastic subgradients, each of the terms in the sum is zero in expectation (conditioned on the past), and the partial sums therefore form a martingale. Furthermore, by Hölder's inequality:

$$\left\langle\check{\Delta}^{(t)}-\check{\nabla}f_t\left(\theta^{(t)}\right),\theta^*-\theta^{(t)}\right\rangle\leq\left\|\check{\Delta}^{(t)}-\check{\nabla}f_t\left(\theta^{(t)}\right)\right\|_*\left\|\theta^*-\theta^{(t)}\right\|\leq 4B_{\check{\Delta}}\sqrt{2B_\Psi},$$

where the last line holds because $\left\|\theta^*-\theta^{(t)}\right\|\leq\left\|\theta^*-\theta^{(1)}\right\|+\left\|\theta^{(t)}-\theta^{(1)}\right\|\leq 2\sup_{\theta\in\Theta}\sqrt{2D_\Psi\left(\theta\mid\theta^{(1)}\right)}\leq 2\sqrt{2B_\Psi}$, using the fact that D_Ψ is 1-strongly convex w.r.t. $\|\cdot\|$, and the definition of $\theta^{(1)}$. Hence, by the Hoeffding-Azuma inequality:

$$\Pr\left\{\frac{1}{T}\sum_{t=1}^{T}\left\langle\check{\Delta}^{(t)}-\check{\nabla}f_t\left(\theta^{(t)}\right),\theta^*-\theta^{(t)}\right\rangle\geq\epsilon\right\}\leq\exp\left(-\frac{T\epsilon^2}{64B_\Psi B_{\check{\Delta}}^2}\right).$$

Equivalently:

$$\Pr\left\{\frac{1}{T}\sum_{t=1}^{T}\left\langle\check{\Delta}^{(t)}-\check{\nabla}f_t\left(\theta^{(t)}\right),\theta^*-\theta^{(t)}\right\rangle\geq 8B_{\check{\Delta}}\sqrt{\frac{B_\Psi\ln\frac{1}{\delta}}{T}}\right\}\leq\delta.$$

Substituting this into Equation 25, and applying the inequality $\sqrt{a}+\sqrt{b}\leq\sqrt{2a+2b}$, yields the claimed result. ■

Corollary 20 ***(Stochastic Gradient Descent)*** *Let $f_1,f_2,\ldots:\Theta\to\mathbb{R}$ be a sequence of convex functions that we wish to minimize on a compact convex set Θ.*

Define the step size $\eta=B_\Theta/B_{\check{\Delta}}\sqrt{2T}$, where $B_\Theta\geq\max_{\theta\in\Theta}\|\theta\|_2$, and $B_{\check{\Delta}}\geq\left\|\check{\Delta}^{(t)}\right\|_2$ is a uniform upper bound on the norms of the stochastic subgradients. Suppose that we perform T iterations of the following stochastic *update, starting from $\theta^{(1)}=\operatorname{argmin}_{\theta\in\Theta}\|\theta\|_2$:*

$$\theta^{(t+1)}=\Pi_\Theta\left(\theta^{(t)}-\eta\check{\Delta}^{(t)}\right),$$

where $\mathbb{E}\left[\check{\Delta}^{(t)}\mid\theta^{(t)}\right]\in\partial f_t(\theta^{(t)})$, i.e. $\check{\Delta}^{(t)}$ is a stochastic subgradient of f_t at $\theta^{(t)}$, and Π_Θ projects its argument onto Θ w.r.t. the Euclidean norm. Then, with probability $1-\delta$ over the draws of the

stochastic subgradients:

$$\frac{1}{T}\sum_{t=1}^{T} f_t\left(\theta^{(t)}\right) - \frac{1}{T}\sum_{t=1}^{T} f_t\left(\theta^*\right) \leq 2B_\Theta B_{\tilde{\nabla}}\sqrt{\frac{1+16\ln\frac{1}{\delta}}{T}},$$

where $\theta^* \in \Theta$ is an arbitrary reference vector.

Proof Follows from taking $\Psi(\theta) = \|\theta\|_2^2/2$ in Theorem 19. ■

Corollary 21 *Let $\mathcal{M} := \left\{M \in \mathbb{R}^{\tilde{m}\times\tilde{m}} : \forall i \in [\tilde{m}]\, .M_{:,i} \in \Delta^{\tilde{m}}\right\}$ be the set of all left-stochastic $\tilde{m}\times\tilde{m}$ matrices, and let $f_1, f_2, \ldots : \mathcal{M} \to \mathbb{R}$ be a sequence of concave functions that we wish to maximize.*

Define the step size $\eta = \sqrt{\tilde{m}\ln\tilde{m}/TB_{\hat{\Delta}}^2}$, where $B_{\hat{\Delta}} \geq \left\|\hat{\Delta}^{(t)}\right\|_{\infty,2}$ is a uniform upper bound on the norms of the stochastic supergradients, and $\|\cdot\|_{\infty,2} := \sqrt{\sum_{i=1}^{\tilde{m}}\|M_{:,i}\|_\infty^2}$ is the $L_{\infty,2}$ matrix norm. Suppose that we perform T iterations of the following stochastic update starting from the matrix $M^{(1)}$ with all elements equal to $1/\tilde{m}$:

$$\begin{aligned}\tilde{M}^{(t+1)} =& M^{(t)} \odot .\exp\left(\eta\hat{\Delta}^{(t)}\right)\\ M_{:,i}^{(t+1)} =& \tilde{M}_{:,i}^{(t+1)} / \left\|\tilde{M}_{:,i}^{(t+1)}\right\|_1,\end{aligned}$$

where $\mathbb{E}\left[-\hat{\Delta}^{(t)} \mid M^{(t)}\right] \in \partial\left(-f_t(M^{(t)})\right)$, i.e. $\hat{\Delta}^{(t)}$ is a stochastic supergradient of f_t at $M^{(t)}$, and the multiplication and exponentiation in the first step are performed element-wise. Then with probability $1-\delta$ over the draws of the stochastic supergradients:

$$\frac{1}{T}\sum_{t=1}^{T} f_t\left(M^*\right) - \frac{1}{T}\sum_{t=1}^{T} f_t\left(M^{(t)}\right) \leq 2B_{\hat{\Delta}}\sqrt{\frac{2\left(\tilde{m}\ln\tilde{m}\right)\left(1+16\ln\frac{1}{\delta}\right)}{T}},$$

where $M^* \in \mathcal{M}$ is an arbitrary reference matrix.

Proof The same reasoning as was used to prove Corollary 17 from Theorem 15 applies here (but starting from Theorem 19). ■

Lemma 22 *Let $\Lambda := \Delta^{\tilde{m}}$ be the $\tilde{m}$-dimensional simplex, define*

$$\mathcal{M} := \left\{M \in \mathbb{R}^{\tilde{m}\times\tilde{m}} : \forall i \in [\tilde{m}]\, .M_{:,i} \in \Delta^{\tilde{m}}\right\}$$

as the set of all left-stochastic $\tilde{m}\times\tilde{m}$ matrices, and take $f_1, f_2, \ldots : \Lambda \to \mathbb{R}$ to be a sequence of concave functions that we wish to maximize.

Define the step size $\eta = \sqrt{\tilde{m}\ln\tilde{m}/TB_{\hat{\Delta}}^2}$, where $B_{\hat{\Delta}} \geq \left\|\hat{\Delta}^{(t)}\right\|_\infty$ is a uniform upper bound on the ∞-norms of the stochastic supergradients. Suppose that we perform T iterations of the following

update, starting from the matrix $M^{(1)}$ with all elements equal to $1/\tilde{m}$:

$$\begin{aligned}&\lambda^{(t)} \text{ is any stationary distribution of } M^{(t)}\\ A^{(t)} =& \hat{\Delta}^{(t)}\left(\lambda^{(t)}\right)^T\\ \tilde{M}^{(t+1)} =& M^{(t)} \odot .\exp\left(\eta A^{(t)}\right)\\ M_{:,i}^{(t+1)} =& \tilde{M}_{:,i}^{(t+1)} / \left\|\tilde{M}_{:,i}^{(t+1)}\right\|_1,\end{aligned}$$

where a stationary distribution of M (i.e. a $\lambda \in \Lambda$ such that $M\lambda = \lambda$) always exists because M is left-stochastic, $\mathbb{E}\left[-\hat{\Delta}^{(t)} \mid \lambda^{(t)}\right] \in \partial\left(-f_t(\lambda^{(t)})\right)$, i.e. $\hat{\Delta}^{(t)}$ is a stochastic supergradient of f_t at $\lambda^{(t)}$, and the multiplication and exponentiation of the third step are performed element-wise. Then with probability $1-\delta$ over the draws of the stochastic supergradients:

$$\frac{1}{T}\sum_{t=1}^{T} f_t\left(M^*\lambda^{(t)}\right) - \frac{1}{T}\sum_{t=1}^{T} f_t\left(\lambda^{(t)}\right) \leq 2B_{\hat{\Delta}}\sqrt{\frac{2\left(\tilde{m}\ln\tilde{m}\right)\left(1+16\ln\frac{1}{\delta}\right)}{T}},$$

where $M^* \in \mathcal{M}$ is an arbitrary left-stochastic reference matrix.

Proof The same reasoning as was used to prove Lemma 18 from Corollary 17 applies here (but starting from Corollary 21). ■

C.3. Two-Player Convergence Rates

Proof [Proof of Lemma 3] Applying Corollary 16 to the optimization over λ gives:

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}\left(\theta^{(t)}, \lambda^*\right) - \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}\left(\theta^{(t)}, \lambda^{(t)}\right) \leq B_\Lambda B_\Delta\sqrt{\frac{2}{T}}.$$

By the definition of $\mathcal{O}_\rho$ (Definition 1):

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}\left(\theta^{(t)}, \lambda^*\right) - \inf_{\theta^*\in\Theta}\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}\left(\theta^*, \lambda^{(t)}\right) \leq \rho + B_\Lambda B_\Delta\sqrt{\frac{2}{T}}.$$

Using the linearity of $\mathcal{L}$ in λ, the fact that $B_\Lambda = R$, and the definitions of $\bar{\theta}$ and $\bar{\lambda}$, yields the claimed result. ■

Lemma 23 ***(Algorithm 5)*** *Suppose that Θ is a compact convex set, Λ and R are as in Theorem 2, and that the objective and constraint functions $g_0, g_1, \ldots, g_m$ are convex. Define the three upper bounds $B_\Theta \geq \max_{\theta\in\Theta}\|\theta\|_2$, $B_{\check{\Delta}} \geq \max_{t\in[T]}\left\|\check{\Delta}_\theta^{(t)}\right\|_2$, and $B_\Delta \geq \max_{t\in[T]}\left\|\Delta_\lambda^{(t)}\right\|_2$.*

If we run Algorithm 5 with the step sizes $\eta_\theta := B_\Theta/B_{\check{\Delta}}\sqrt{2T}$ and $\eta_\lambda := R/B_\Delta\sqrt{2T}$, then the result satisfies the conditions of Theorem 2 for:

$$\epsilon = 2\left(B_\Theta B_{\check{\Delta}} + RB_\Delta\right)\sqrt{\frac{1+16\ln\frac{2}{\delta}}{T}},$$

with probability $1-\delta$ over the draws of the stochastic (sub)gradients.

Algorithm 5 Optimizes the Lagrangian formulation (Equation 3) in the convex setting. The parameter R is the radius of the Lagrange multiplier space $\Lambda := \left\{\lambda \in \mathbb{R}_+^m : \|\lambda\|_1 \leq R\right\}$, and the functions Π_Θ and Π_Λ project their arguments onto Θ and Λ (respectively) w.r.t. the Euclidean norm.

StochasticLagrangian $(R \in \mathbb{R}_+, \mathcal{L} : \Theta \times \Lambda \to \mathbb{R}, T \in \mathbb{N}, \eta_\theta, \eta_\lambda \in \mathbb{R}_+)$:
Initialize $\theta^{(1)} = 0$, $\lambda^{(1)} = 0$ *// Assumes* $0 \in \Theta$
For $t \in [T]$:
  Let $\check{\Delta}_\theta^{(t)}$ be a stochastic subgradient of $\mathcal{L}\left(\theta^{(t)}, \lambda^{(t)}\right)$ w.r.t. θ
  Let $\Delta_\lambda^{(t)}$ be a stochastic gradient of $\mathcal{L}\left(\theta^{(t)}, \lambda^{(t)}\right)$ w.r.t. λ
  Update $\theta^{(t+1)} = \Pi_\Theta\left(\theta^{(t)} - \eta_\theta \check{\Delta}_\theta^{(t)}\right)$ *// Projected SGD updates ...*
  Update $\lambda^{(t+1)} = \Pi_\Lambda\left(\lambda^{(t)} + \eta_\lambda \Delta_\lambda^{(t)}\right)$ *// ...*
Return $\theta^{(1)}, \ldots, \theta^{(T)}$ and $\lambda^{(1)}, \ldots, \lambda^{(T)}$

Proof Applying Corollary 20 to the two optimizations (over θ and λ) gives that with probability $1 - 2\delta'$ over the draws of the stochastic (sub)gradients:

$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}\left(\theta^{(t)}, \lambda^{(t)}\right) - \frac{1}{T}\sum_{t=1}^T \mathcal{L}\left(\theta^*, \lambda^{(t)}\right) \leq 2B_\Theta B_{\check{\Delta}} \sqrt{\frac{1 + 16\ln\frac{1}{\delta'}}{T}}$$
$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}\left(\theta^{(t)}, \lambda^*\right) - \frac{1}{T}\sum_{t=1}^T \mathcal{L}\left(\theta^{(t)}, \lambda^{(t)}\right) \leq 2B_\Lambda B_\Delta \sqrt{\frac{1 + 16\ln\frac{1}{\delta'}}{T}}.$$

Adding these inequalities, taking $\delta = 2\delta'$, using the linearity of $\mathcal{L}$ in λ, the fact that $B_\Lambda = R$, and the definitions of $\bar{\theta}$ and $\bar{\lambda}$, yields the claimed result. ■

Proof [Proof of Lemma 7] Applying Lemma 18 to the optimization over λ (with $\tilde{m} := m + 1$) gives:

$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}_\lambda\left(\theta^{(t)}, M^*\lambda^{(t)}\right) - \frac{1}{T}\sum_{t=1}^T \mathcal{L}_\lambda\left(\theta^{(t)}, \lambda^{(t)}\right) \leq 2B_\Delta \sqrt{\frac{(m+1)\ln(m+1)}{T}}.$$

By the definition of $\mathcal{O}_\rho$ (Definition 1):

$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}_\theta\left(\theta^{(t)}, \lambda^{(t)}\right) - \inf_{\theta^* \in \Theta} \frac{1}{T}\sum_{t=1}^T \mathcal{L}_\theta\left(\theta^*, \lambda^{(t)}\right) \leq \rho.$$

Using the definitions of $\bar{\theta}$ and $\bar{\lambda}$ yields the claimed result. ■

Proof [Proof of Lemma 8] Applying Corollary 20 to the optimization over θ, and Lemma 22 to that over λ (with $\tilde{m} := m + 1$), gives that with probability $1 - 2\delta'$ over the draws of the stochastic

(sub)gradients:

$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}_\theta\left(\theta^{(t)}, \lambda^{(t)}\right) - \frac{1}{T}\sum_{t=1}^T \mathcal{L}_\theta\left(\theta^*, \lambda^{(t)}\right) \leq 2B_\Theta B_{\check{\Delta}} \sqrt{\frac{1 + 16\ln\frac{1}{\delta'}}{T}}$$
$$\frac{1}{T}\sum_{t=1}^T \mathcal{L}_\lambda\left(\theta^{(t)}, M^*\lambda^{(t)}\right) - \frac{1}{T}\sum_{t=1}^T \mathcal{L}_\lambda\left(\theta^{(t)}, \lambda^{(t)}\right) \leq 2B_\Delta \sqrt{\frac{2(m+1)\ln(m+1)\left(1 + 16\ln\frac{1}{\delta'}\right)}{T}}.$$

Taking $\delta = 2\delta'$, and using the definitions of $\bar{\theta}$ and $\bar{\lambda}$, yields the claimed result. ■

References

A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach. A reductions approach to fair classification. In *ICML*, 2018.

S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

R. E. Barlow, D. J. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical Inference Under Order Restrictions; The Theory And Application Of Isotonic Regression*. Wiley, New York, USA, 1972.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, May 2003.

K. Bellare, G. Druck, and A. McCallum. Alternating projections for learning with expectation constraints. *UAI*, 2009.

A. Blum and Y. Mansour. From external to internal regret. *JMLR*, 8:1307–1324, 2007.

D. G. Bocian, K. S. Ernst, and W. Li. Race, ethnicity and subprime home loan pricing. *Journal of Economics and Business*, 60(1-2):110–124, 2008.

H. F. Bohnenblust, S. Karlin, and L. S. Shapley. Games with continuous, convex pay-off. *Contributions to the Theory of Games*, 1(24):181–192, 1950.

M. Bonakdarpour, S. Chatterjee, R. F. Barber, and J. D. Lafferty. Prediction rule reshaping. In *ICML*, 2018.

J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

K. Canini, A. Cotter, M. R. Gupta, M. Milani Fard, and J. Pfeifer. Fast and flexible monotonic functions with ensembles of lattices. In *NIPS*, pages 2919–2927, 2016.

R. S. Chen, B. Lucier, Y. Singer, and V. Syrgkanis. Robust optimization for non-convex objectives. In *NIPS*, 2017.

X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *FOCS'06*, pages 261–272. IEEE, 2006.

Y. Chen and R. J. Samworth. Generalized additive and index models with shape constraints. *Journal Royal Statistical Society B*, 2016.

D. Chetverikov, A. Santos, and A. M. Shaikh. The econometrics of shape restrictions. *Annual Review of Economics*, 2018.

P. Christiano, J. A. Kelner, A. Madry, C. A. Spielman, and S. Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *STOC*, pages 273–282, 2011.

Q. Cormier, M. Milani Fard, K. Canini, and M. R. Gupta. Launch and iterate: Reducing prediction churn. *NIPS*, 2016.

A. Cotter, M. R. Gupta, and J. Pfeifer. A Light Touch for heavily constrained SGD. In *COLT*, pages 729–771, 2016.

A. Cotter, M. Gupta, H. Jiang, N. Srebro, K. Sridharan, S. Wang, B. Woodworth, and S. You. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. In *ICML*, 2019a.

A. Cotter, M. R. Gupta, H. Jiang, E. Louidor, J. Muller, T. Narayan, S. Wang, and T. Zhu. Shape constraints for set functions. In *ICML*, 2019b.

A. Cotter, H. Jiang, and K. Sridharan. Two-player games for efficient non-convex constrained optimization. In *Algorithmic Learning Theory*, pages 300–332, 2019c.

M. Davenport, R. G. Baraniuk, and C. D. Scott. Tuning support vector machines for minimax and Neyman-Pearson classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

M. Donini, L. Oneto, S. Ben-David, J. Shawe-Taylor, and M. Pontil. Empirical risk minimization under fairness constraints. *NeurIPS*, 2018.

E. Eban, M. Schain, A. Mackey, A. Gordon, R. A. Saurous, and G. Elidan. Scalable learning of non-decomposable objectives. *AIStats*, 2017.

B. Fish, J. Kun, and A. D. Lelkes. A confidence-based approach for balancing fairness and accuracy. *SIAM ICDM*, 2016.

D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *FOCS*, pages 420–428. IEEE Computer Society, 2013.

G. Gasso, A. Pappaionannou, M. Spivak, and L. Bottou. Batch and online learning algorithms for nonconvex Neyman-Pearson classification. *ACM Transactions on Intelligent Systems and Technology*, 2011.

I. L. Glicksberg. A further generalization of the Kakutani fixed point theorem with application to Nash equilibrium points. *Proceedings American Mathematical Society*, 3:170–174, 1952.

G. Goh, A. Cotter, M. R. Gupta, and M. P. Friedlander. Satisfying real-world goals with dataset constraints. In *NIPS*, pages 2415–2423, 2016.

G. J. Gordon, A. Greenwald, and C. Marks. No-regret learning in convex games. In *ICML*, pages 360–367, 2008.

P. Groeneboom and G. Jongbloed. *Nonparametric Estimation Under Shape Constraints*. Cambridge Press, New York, USA, 2014.

M. R. Gupta, A. Cotter, J. Pfeifer, K. Voevodski, K. Canini, A. Mangylov, W. Moczydlowski, and A. van Esbroeck. Monotonic calibrated interpolated look-up tables. *JMLR*, 17(109):1–47, 2016.

M. R. Gupta, D. Bahri, A. Cotter, and K. Canini. Diminishing returns shape constraints for interpretability and regularization. *NeurIPS*, 2018.

M. R. Gupta, A. Cotter, M. Milani Fard, and S. Wang. Proxy fairness. In *arXiv:1806.11212*, 2019.

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *NIPS*, 2016.

E. Hazan and S. Kale. Projection-free online learning. In *ICML*, 2012.

H. Heidari, C. Ferrari, K. Gummadi, and A. Krause. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *NeurIPS*, pages 1265–1276, 2018.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. Fairness-aware classifier with prejudice remover regularizer. *Machine Learning and Knowledge Discovery in Databases*, pages 35–50, 2012.

M. Kearns, S. Neel, A. Roth, and Z. S. Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *ICML*, 2018.

B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. Interpretable classifiers using rules and Bayesian analysis: building a better stroke prediction model. *Annals of Applied Statistics*, 2015.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-Net: Learning PDEs from Data. In *ICML*, 2018.

R. Luss and S. Rosset. Bounded isotonic regression. *Electronic Journal of Statistics*, 11(2):4488–4514, 2017.

M. Mahdavi, T. Yang, R. Jin, S. Zhu, and J. Yi. Stochastic gradient descent with only one projection. In *NIPS*, pages 494–502, 2012.

G. S. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning with expectation regularization. In *ICML*, 2007.

G. S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 11, 2010.

H. Narasimhan. Learning with complex loss functions and constraints. In *AIStats*, 2018.

H. Narasimhan, A. Cotter, and M. R. Gupta. Optimizing generalized rate metrics through game equilibrium. In *NeurIPS*, 2019a.

H. Narasimhan, A. Cotter, and M. R. Gupta. On making stochastic classifiers deterministic. In *NeurIPS*, 2019b.

H. Narasimhan, A. Cotter, M. R. Gupta, and S. Wang. Pairwise fairness for ranking and regression. In *AAAI*, 2020.

A. Nemirovski and D. Yudin. *Problem Complexity And Method Efficiency In Optimization*. John Wiley & Sons Ltd, 1983.

T Parthasarathy. Equilibria of continuous two-person games. *Pacific Journal of Mathematics*, 57(1): 265–270, 1975.

N. Pya and S. N. Wood. Shape constrained additive models. *Statistics and Computing*, 2015.

A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. In *NIPS*, pages 3066–3074, 2013.

A. Rakhlin, K. Sridharan, and A. Tewari. Online learning: beyond regret. In *COLT*, pages 559–594, 2011.

C. D. Scott and R. D. Nowak. A Neyman-Pearson approach to statistical learning. *IEEE Transactions on Information Theory*, 2005.

N. Srebro, K. Sridharan, and A. Tewari. On the universality of online mirror descent. In *NIPS*, 2011.

R. Stewart and S. Ermon. Label-free supervision of neural networks with physics and domain knowledge. *AAAI*, 2017.

J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

B. E. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro. Learning non-discriminatory predictors. In *COLT*, pages 1920–1953, 2017.

T. Yang, Q. Lin, and L. Zhang. A richer theory of convex constrained optimization with reduced projections and improved rates. In *ICML*, pages 3901–3910, 2017.

S. You, K. Canini, D. Ding, J. Pfeifer, and M. R. Gupta. Deep lattice networks for learning partial monotonic functions. *NIPS*, 2017.

M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi. Fairness constraints: A mechanism for fair classification. In *ICML Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2015.

M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *AIStats*, pages 962–970, 2017.

Journal of Machine Learning Research 20 (2019) 1-27 Submitted 9/18; Revised 10/19; Published 11/19

Fast Automatic Smoothing for Generalized Additive Models

Yousra El-Bachir YOUSRA.ELBACHIR@GMAIL.COM
Anthony C. Davison ANTHONY.DAVISON@EPFL.CH
EPFL-FSB-MATH-STAT
Ecole Polytechnique Fédérale de Lausanne
Station 8, CH-1015 Lausanne, Switzerland

Editor: Manfred Opper

Abstract

Generalized additive models (GAMs) are regression models wherein parameters of probability distributions depend on input variables through a sum of smooth functions, whose degrees of smoothness are selected by L_2 regularization. Such models have become the de-facto standard nonlinear regression models when interpretability and flexibility are required, but reliable and fast methods for automatic smoothing in large data sets are still lacking. We develop a general methodology for automatically learning the optimal degree of L_2 regularization for GAMs using an empirical Bayes approach. The smooth functions are penalized by hyper-parameters that are learned simultaneously by maximization of a marginal likelihood using an approximate expectation-maximization algorithm. The latter involves a double Laplace approximation at the E-step, and leads to an efficient M-step. Empirical analysis shows that the resulting algorithm is numerically stable, faster than the best existing methods and achieves state-of-the-art accuracy. For illustration, we apply it to an important and challenging problem in the analysis of extremal data.

Keywords: Automatic L_2 Regularization, Empirical Bayes, Expectation-maximization Algorithm, Generalized Additive Model, Laplace Approximation, Marginal Maximum Likelihood

1. Introduction

State-of-the-art machine learning methods achieve impressive accuracy, but their opacity can make them unattractive when their results contradict intuition or lack a ready explanation. Methods that are interpretable by humans and easy to diagnose and debug are needed for difficult decision-making problems in healthcare, legal settings, finance, risk management, and many other areas. One possibility is the use of generalized additive models, a class of interpretable data-driven models used in smooth regression (Duvenaud et al., 2011; Ba et al., 2012; Tsang et al., 2018; Mutny and Krause, 2018).

Generalized additive models (GAMs) are a class of supervised learning tools that describe the relationship between output variables and inputs using a sum of smooth functions (Hastie et al., 2009, Chapter 9). They were introduced by Hastie and Tibshirani (1986), who represented the smooth functions by scatterplot smoothers and trained them sequentially by backfitting (Breiman and Friedman, 1985). The corresponding `R` (R Core Team, 2019) package `gam` implements the methods in Hastie and Tibshirani (1990), which select the level of smoothness by stepwise regression using approximate distributional results. Backfitting allows smooth terms to be represented by local regression smoothers (Cleveland et al., 1993), but inference based on the resulting fit is awkward. Yee and Wild (1996) later proposed modified vector backfitting, whereby several smooth functions are learned simultaneously. Their method, embodied in the `R` package `VGAM`, first learns the linear components and then learns the nonlinear part by training a vector additive model on the resulting partial residuals. In the `R` package `gamlss`, Rigby and Stasinopoulos (2005) learn the smooth functions sequentially by combining backfitting with two algorithms, which optimize the penalized likelihood of the regression weights. The first algorithm generalizes that of Cole and Green (1992), whereas the second generalizes that of Rigby and Stasinopoulos (1996) and is preferable when the parameters of the distribution are orthogonal with respect to the information matrix. All these approaches invoke backfitting, which dissociates learning of the regression model from that of the smoothing hyper-parameters. This may be statistically inefficient, and accuracy may be increased by learning the appropriate degree of smoothing as part of the regression training. An alternative representation of the smooth functions that enables automatic smoothing is via basis function expansion using reduced rank smoothing; this is the foundation upon which we build our methodology.

1.1. Model Set-up

We suppose that independent observations come from a probability distribution whose parameters depend on generalized additive models. Let Y_i denote a random variable with realized value y_i and probability distribution function $F_i(y_i; \boldsymbol{\theta}_i)$ that depends on a parameter vector $\boldsymbol{\theta}_i = (\theta_i^{(1)}, \ldots, \theta_i^{(P)}) \in \mathbb{R}^P$; so for the training set $\boldsymbol{y} = (y_1, \ldots, y_n)^T$, the full parameter vector is $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)^T \in \mathbb{R}^{nP}$ with sub-vectors $\boldsymbol{\theta}^{(p)} = (\theta_1^{(p)}, \ldots, \theta_n^{(p)})^T \in \mathbb{R}^n$ for $p = 1, \ldots, P$. In the Gaussian model for example, $P = 2$, $\boldsymbol{\theta}^{(1)} = \boldsymbol{\mu}$ is the mean and $\boldsymbol{\theta}^{(2)} = \boldsymbol{\sigma}$ is the standard deviation, and we have $\boldsymbol{\theta} = (\mu_1, \sigma_1, \ldots, \mu_n, \sigma_n)^T$.

For a probability distribution with P parameters, each $\boldsymbol{\theta}^{(p)}$ has an additive structure, which we now describe. Let $\boldsymbol{X}_i^{(p)^*}$ denote the i-th row of a feature matrix corresponding to a regression weight vector $\boldsymbol{w}^{(p)^*}$ that includes an offset. The superscript * refers to an intermediate notation for the regression elements that will become clear when we define the full feature matrices corresponding to the untransformed input variables and the smooth functions. Let $q_p \geqslant 0$ denote the number of unknown smooth functions $f_j^{(p)}$ contributing to $\boldsymbol{\theta}^{(p)}$, and let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ denote the vectors of input variables. In spatio-temporal settings for example, $\boldsymbol{x}_1$ may denote the spatial location of a site and $\boldsymbol{x}_2$ may denote the time. The components $\theta_i^{(p)}$ of $\boldsymbol{\theta}^{(p)}$ are represented using the additive structure

$$\theta_i^{(p)} = \boldsymbol{X}_i^{(p)^*} \boldsymbol{w}^{(p)^*} + \sum_{j=1}^{q_p} f_j^{(p)}(x_{i1}, x_{i2}, \ldots), \quad i = 1, \ldots, n. \tag{1}$$

Each of the $f_j^{(p)}$ can be a function of one or more inputs, and is parametrized by an expansion of basis functions $b_k^{(p)}(x)$,

$$f_j^{(p)}(x) = \sum_{k=1}^{K} \tilde{w}_k^{(p)} b_k^{(p)}(x) = \tilde{\boldsymbol{w}}^{(p)^T} \boldsymbol{b}^{(p)}(x), \tag{2}$$

©2019 Yousra El-Bachir and Anthony C. Davison.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-659.html.

where $\tilde{\boldsymbol{w}}^{(p)} = (\tilde{w}_1^{(p)}, \ldots, \tilde{w}_K^{(p)})^T \in \mathbb{R}^K$ is the column vector of unknown regression weights that expand $f_j^{(p)}$, and $\boldsymbol{b}^{(p)}(x) = \{b_1^{(p)}(x), \ldots, b_K^{(p)}(x)\}^T \in \mathbb{R}^K$ is a K-dimensional vector of basis functions selected from a dictionary of readily-available functions, such as cubic regression splines. The basis dimension K is chosen manually, and typically is allowed to grow slowly with the size n of the training set. We assume that the smooth functions are subject to a sum-to-zero identifiability constraint

$$\sum_{i=1}^{n} f_j^{(p)}(x_{ij}) = 0, \tag{3}$$

which leads to good coverage for confidence intervals (Nychka, 1988). In this setting, the components of $\boldsymbol{\theta}^{(p)}$ in (1) become $\theta_i^{(p)} = \boldsymbol{X}_i^{(p)} \boldsymbol{w}^{(p)}$, where $\boldsymbol{w}^{(p)} \in \mathbb{R}^{d_p}$ denotes the regression weights including their parametric part indicated with superscript * and smooth part indicated by a tilde in (2), and $\boldsymbol{X}^{(p)} \in \mathbb{R}^{n \times d_p}$ denotes the corresponding feature matrix. We assume that some of the columns of $\boldsymbol{X}^{(p)}$ have been centered to absorb the identifiability constraint (3) on the smooth functions before training.

A GAM is characterized by the additivity of its functions of inputs and by their smoothness. The latter is controlled by penalizing the curvature of the functions, using roughness penalties on the $f_j^{(p)}$ of the form

$$\mathrm{PEN}(\lambda_j^{(p)}) = \lambda_j^{(p)} \int \left\{ f_j^{(p)\prime\prime}(t) \right\}^2 \mathrm{d}t \in \mathbb{R}, \tag{4}$$

where the regularization hyper-parameter $\lambda_j^{(p)} > 0$ controls the degree of smoothness, and the integral is taken over the input range values. If the basis functions are twice differentiable, the element-wise integration

$$\boldsymbol{S}_j^{(p)} = \int \boldsymbol{b}^{(p)\prime\prime}(x) \boldsymbol{b}^{(p)\prime\prime T}(x) \, \mathrm{d}x$$

defines a known symmetric and semi-positive definite smoothing matrix $\boldsymbol{S}_j^{(p)} \in \mathbb{R}^{K \times K}$. The penalty in (4) thus becomes quadratic,

$$\mathrm{PEN}(\lambda_j^{(p)}) = \lambda_j^{(p)} \boldsymbol{w}^{(p)T} \boldsymbol{S}_j^{(p)} \boldsymbol{w}^{(p)}.$$

On defining analogous quantities for any of the parameter vectors $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(P)}$, and stacking the regression weights and the smoothing hyper-parameters to form $\boldsymbol{w} \in \mathbb{R}^d$ and $\boldsymbol{\lambda} \in \mathbb{R}^q$ with $d = \sum_{p=1}^{P} d_p$ and $q = \sum_{p=1}^{P} q_p$, the full functional parameter vector and curvature penalties are parametrized by

$$\boldsymbol{\theta}_{\boldsymbol{w}} = \boldsymbol{X}\boldsymbol{w} \in \mathbb{R}^{nP}, \quad \mathrm{PEN}(\boldsymbol{\lambda}) = \sum_{p=1}^{P} \sum_{j=1}^{q_p} \mathrm{PEN}(\lambda_j^{(p)}) = \boldsymbol{w}^T \boldsymbol{S}_{\boldsymbol{\lambda}} \boldsymbol{w} \in \mathbb{R}, \tag{5}$$

where the i-th row block of the full feature matrix $\boldsymbol{X} \in \mathbb{R}^{nP \times d}$ is

$$\boldsymbol{X}_i = \mathrm{diag}\left(\boldsymbol{X}_i^{(1)}, \ldots, \boldsymbol{X}_i^{(P)} \right) \in \mathbb{R}^{P \times d},$$

and the full smoothing matrix

$$\boldsymbol{S}_{\boldsymbol{\lambda}} = \mathrm{diag}\left(\lambda_1^{(1)} \boldsymbol{S}_1^{(1)}, \ldots, \lambda_{q_P}^{(P)} \boldsymbol{S}_{q_P}^{(P)} \right) \in \mathbb{R}^{d \times d} \tag{6}$$

is block diagonal.

Learning the regression weights involves balancing the conflicting goals of providing a good fit to the data and avoiding overfitting. For a given $\boldsymbol{\lambda}$, this is achieved by maximizing the penalized log-likelihood for $\boldsymbol{w}$,

$$\ell_{\mathrm{P}}(\boldsymbol{w}; \boldsymbol{y}, \boldsymbol{\lambda}) = \ell_{\mathrm{L}}(\boldsymbol{\theta}_{\boldsymbol{w}}; \boldsymbol{y}) - \frac{1}{2} \boldsymbol{w}^T \boldsymbol{S}_{\boldsymbol{\lambda}} \boldsymbol{w}, \tag{7}$$

where the log-likelihood ℓ_{L} may be written equivalently in terms of $\boldsymbol{\theta}$ or of $\boldsymbol{w}$. Let $\boldsymbol{U}_{\mathrm{L}}(\boldsymbol{w}; \boldsymbol{y}) \in \mathbb{R}^d$ and $\boldsymbol{H}_{\mathrm{L}}(\boldsymbol{w}; \boldsymbol{y}) \in \mathbb{R}^{d \times d}$, or $\boldsymbol{U}_{\mathrm{L}}(\boldsymbol{\theta}; \boldsymbol{y}) \in \mathbb{R}^{Pn}$ and $\boldsymbol{H}_{\mathrm{L}}(\boldsymbol{\theta}; \boldsymbol{y}) \in \mathbb{R}^{Pn \times Pn}$, denote the gradient and negative Hessian of ℓ_{L} with respect to $\boldsymbol{w}$ or to $\boldsymbol{\theta}$. The corresponding penalized quantities are

$$\begin{aligned} \boldsymbol{U}_{\mathrm{P}}(\boldsymbol{w}; \boldsymbol{y}, \boldsymbol{\lambda}) &= \boldsymbol{U}_{\mathrm{L}}(\boldsymbol{w}; \boldsymbol{y}) - \boldsymbol{S}_{\boldsymbol{\lambda}} \boldsymbol{w}, & \boldsymbol{H}_{\mathrm{P}}(\boldsymbol{w}; \boldsymbol{y}, \boldsymbol{\lambda}) &= \boldsymbol{H}_{\mathrm{L}}(\boldsymbol{w}; \boldsymbol{y}) + \boldsymbol{S}_{\boldsymbol{\lambda}}, \\ &= \boldsymbol{X}^T \boldsymbol{U}_{\mathrm{L}}(\boldsymbol{\theta}; \boldsymbol{y}) - \boldsymbol{S}_{\boldsymbol{\lambda}} \boldsymbol{w}, & &= \boldsymbol{X}^T \boldsymbol{H}_{\mathrm{L}}(\boldsymbol{\theta}; \boldsymbol{y}) \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}}. \end{aligned} \tag{8}$$

The negative Hessian is used to compute standard errors and confidence intervals. For simplicity we sometimes suppress the dependence upon $\boldsymbol{y}$ and $\boldsymbol{\lambda}$ from the notation. Maximization of the penalized log-likelihood (7) provides an estimate for $\boldsymbol{w}$ for a given value of the smoothing hyper-parameters $\boldsymbol{\lambda}$. In Section 1.2 we briefly summarize the main classical optimization methods for embodying learning of $\boldsymbol{\lambda}$ in that of the regression weights.

1.2. Classical Optimization

The two learning strategies for generalized additive models are performance iteration (Gu, 1992) and outer iteration (O'Sullivan et al., 1986), and optimize a criterion for the smoothing hyper-parameters whilst updating the regression weights. The updating step in performance iteration consists of one iteration for the regression weights, often performed by iterative least squares (Nelder and Wedderburn, 1972), followed by one full optimization for the smoothing hyper-parameters. Since the smoothness selection is applied to an intermediate estimate of the regression model, its score changes from iteration to iteration and convergence of the overall optimization is not guaranteed, as shown by Wood (2008, 2011). The updating step in outer iteration comprises one update for the smoothing hyper-parameters followed by one full optimization for the regression weights. Since the former are obtained from a converged and so a fixed regression model, the convergence of outer iteration is guaranteed, but each updating step is computationally more expensive, and the dependence between the regression weights and the smoothing hyper-parameters makes the calculations much more challenging.

Once the strategy for automatic smoothing is chosen, the classical approach for learning its regularization hyper-parameters is to minimize measures of prediction error such as the Akaike or Bayesian information criteria (AIC or BIC), or the generalized cross validation (GCV) criterion. The first criteria tend to overfit, and GCV can generate multiple minima and unstable estimates that may lead to substantial underfitting (Reiss and Ogden,

2009; Wood, 2011). Use of marginal likelihood overcomes these limitations but involves intractable integrals. Despite the widespread use of GAMs, automatic learning of their smoothing hyper-parameters remains an open problem. The reliable method (Wood, 2011) and its generalization (Wood et al., 2016), implemented in the `R` package `mgcv`, combine the advantages of the marginal likelihood approach with the good convergence of outer iteration. However, these approaches are challenging to set up, difficult to extend to new families of distributions, and computationally expensive for large samples. Methods specifically designed for large (Wood et al., 2015) and big (Wood et al., 2017) data sets are based on performance iteration, and so offer no guarantee of convergence to a local optimum. In this paper we overcome these limitations by presenting an approach that is simpler, faster and achieves state-of-the-art accuracy.

The rest of the paper is organized as follows. Section 2 introduces our proposed automatic smoothness selection procedure, which is based on an approximate expectation-maximization algorithm. Section 3 assesses its performance with a simulation study. Section 4 provides a real data analysis on extreme temperatures, and Section 5 closes the paper with a discussion.

2. Automatic Smoothing

The Bayesian perspective provides an interpretation for the roughness penalty that underlies the weighted L_2 regularization in the penalized log-likelihood (7), as we now describe. Let $\boldsymbol{S}_{\boldsymbol{\lambda}}^{-}$ denote the generalized inverse of $\boldsymbol{S}_{\boldsymbol{\lambda}}$, and suppose that the regression weights have an improper $\mathcal{N}(0, \boldsymbol{S}_{\boldsymbol{\lambda}}^{-})$ multivariate Gaussian prior density (Kimeldorf and Wahba, 1970; Silverman, 1985),

$$\pi(\boldsymbol{w}; \boldsymbol{\lambda}) = (2\pi)^{-(d-m)/2} \; |\boldsymbol{S}_{\boldsymbol{\lambda}}|_{+}^{1/2} \exp\left(-\frac{1}{2}\boldsymbol{w}^T \boldsymbol{S}_{\boldsymbol{\lambda}} \boldsymbol{w}\right), \tag{9}$$

where m is the number of zero eigenvalues of $\boldsymbol{S}_{\boldsymbol{\lambda}}$, and $|\boldsymbol{S}_{\boldsymbol{\lambda}}|_{+}$ is the product of its positive eigenvalues. With f here denoting the density of the data, the log-posterior density for $\boldsymbol{w}$ is

$$\begin{aligned} \ell(\boldsymbol{w} \mid \boldsymbol{y}; \boldsymbol{\lambda}) &= \log\left\{f(\boldsymbol{y} \mid \boldsymbol{w}; \boldsymbol{\lambda}) \; \pi(\boldsymbol{w}; \boldsymbol{\lambda})\right\} - \log f(\boldsymbol{y}; \boldsymbol{\lambda}) \\ &= \ell_{\mathrm{P}}(\boldsymbol{w}; \boldsymbol{y}, \boldsymbol{\lambda}) + \frac{1}{2}\log|\boldsymbol{S}_{\boldsymbol{\lambda}}|_{+} - \frac{d-m}{2}\log(2\pi) - \log f(\boldsymbol{y}; \boldsymbol{\lambda}). \end{aligned} \tag{10}$$

The roughness penalty in (5) now appears as the key component of the logarithm of the prior (9), and the penalized log-likelihood (7) appears as the log-posterior (10) (up to a constant depending upon $\boldsymbol{\lambda}$ but not upon $\boldsymbol{w}$). The smoothing hyper-parameters can hence be learned from the last term on the right of (10), the marginal density of $\boldsymbol{y}$,

$$L_{\mathrm{M}}(\boldsymbol{\lambda}; \boldsymbol{y}) = f(\boldsymbol{y}; \boldsymbol{\lambda}) = \int f(\boldsymbol{y}, \boldsymbol{w}; \boldsymbol{\lambda}) \, \mathrm{d}\boldsymbol{w} = \int f(\boldsymbol{y} \mid \boldsymbol{w}; \boldsymbol{\lambda}) \; \pi(\boldsymbol{w}; \boldsymbol{\lambda}) \, \mathrm{d}\boldsymbol{w}.$$

This integral is equivalent to $L_{\mathrm{M}}(\boldsymbol{\lambda}; \boldsymbol{y}) = E_{\pi(\boldsymbol{w};\boldsymbol{\lambda})}\{f(\boldsymbol{y} \mid \boldsymbol{w}; \boldsymbol{\lambda})\}$, which is the expectation of the likelihood function under the prior density of the regression weight vector.

A fully Bayesian approach would involve choosing a prior density for $\boldsymbol{\lambda}$ and integrating out over it, but instead we take an empirical Bayes approach and transform the smoothness

selection problem to an optimization problem, where the optimal $\boldsymbol{\lambda}$ maximize the log-marginal likelihood

$$\ell_{\mathrm{M}}(\boldsymbol{\lambda}; \boldsymbol{y}) \;=\; \log L_{\mathrm{M}}(\boldsymbol{\lambda}; \boldsymbol{y}) \equiv \frac{1}{2}\log|\boldsymbol{S}_{\boldsymbol{\lambda}}|_{+} + \log\int \exp \ell_{\mathrm{P}}(\boldsymbol{w}; \boldsymbol{y}, \boldsymbol{\lambda}) \, \mathrm{d}\boldsymbol{w}, \tag{11}$$

where $\equiv$ indicates equality up to a constant not depending upon $\boldsymbol{\lambda}$. The multidimensional integral over $\boldsymbol{w}$ is generally intractable, and its accurate and efficient computation is challenging. The two classical strategies for this are stochastic integration methods based on importance sampling, and deterministic methods based on quadrature or the Laplace method. Importance sampling is a Monte Carlo integration technique under which the integral is treated as an expectation, but its performance relies on the choice of the distribution from which to sample, and its accuracy increases only with the number of samples. Quadrature involves a discretization of the integrand over the domain of integration, and amounts to calculating a weighted sum of the values of the integrand. Both methods perform well when the dimension d of the regression weight vector $\boldsymbol{w}$ is small, but become computationally infeasible when d becomes large (Vonesh et al., 2002). The most commonly used deterministic integration method is Laplace approximation, which yields an analytical expression for (11) by exploiting a quadratic Taylor expansion of the log-integrand around the maximum penalized likelihood estimate,

$$\ell_{\mathrm{M}}(\boldsymbol{\lambda}; \boldsymbol{y}) \;=\; \frac{1}{2}\,\log|\boldsymbol{S}_{\boldsymbol{\lambda}}|_{+} + \ell_{\mathrm{P}}(\hat{\boldsymbol{w}}_{\boldsymbol{\lambda}}; \boldsymbol{y}, \boldsymbol{\lambda}) - \frac{1}{2}\log\det \boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_{\boldsymbol{\lambda}}; \boldsymbol{y}, \boldsymbol{\lambda}) + O(n^{-1}), \tag{12}$$

where $\hat{\boldsymbol{w}}_{\boldsymbol{\lambda}}$ denotes the maximizer of ℓ_{P} for a given $\boldsymbol{\lambda}$. However, optimization of the resulting approximate log-marginal likelihood (12) is awkward. Each updating step includes intermediate maximizations, involves unstable terms that need careful and computationally expensive decompositions, and requires the fourth derivatives of the log-likelihood to obtain the Hessian matrix of the log-marginal likelihood in the Newton–Raphson method (Wood, 2011; Wood et al., 2016). These difficulties make Laplace approximation computationally demanding for smoothness selection, and limit its extension to complex models. In Section 2.1 we present a simpler alternative that is easier to implement, faster and very accurate.

Although we illustrate our ideas on generalized additive models, our automatic smoothing method extends to the general setting of L_2 regularization if we replace the smoothing matrices with identity matrices. Hyper-parameter optimization based on marginal likelihood in other contexts is a long-standing problem addressed by MacKay (1992, 1999), Neal (1996), Tipping (1999, 2001), Faul and Tipping (2001), Qi et al. (2004), McHutchon and Rasmussen (2011) and many others.

2.1. Approximate Expectation-maximization

We directly maximize the log-marginal likelihood (11) with respect to the smoothing hyper-parameters using the expectation-maximization (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2008), which circumvents evaluation of the objective function. The EM algorithm is an iterative method for computing maximum likelihood estimators for difficult functions, by alternating between an expectation step, the E-step, and its maximization, the M-step, at every iteration until convergence. Ignoring the constant term, taking conditional expectations of expression (10) with respect to the posterior $\pi(\boldsymbol{w} \mid \boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\lambda}_k)$

at the best estimate $\boldsymbol{\lambda}_k$ at iteration k yields

$$\ell_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y}) \equiv Q(\boldsymbol{\lambda};\boldsymbol{\lambda}_k) - K(\boldsymbol{\lambda};\boldsymbol{\lambda}_k),$$

where $\equiv$ indicates equality up a constant, and

$$\begin{aligned} Q(\boldsymbol{\lambda};\boldsymbol{\lambda}_k) &= E_{\pi(\boldsymbol{w}|\boldsymbol{Y};\boldsymbol{\lambda}_k)}\left\{\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{Y},\boldsymbol{\lambda}) + \frac{1}{2}\log|\boldsymbol{S}_{\boldsymbol{\lambda}}|_+\right\}, \qquad (13)\\ K(\boldsymbol{\lambda};\boldsymbol{\lambda}_k) &= E_{\pi(\boldsymbol{w}|\boldsymbol{Y};\boldsymbol{\lambda}_k)}\left\{\ell(\boldsymbol{w}\mid\boldsymbol{Y};\boldsymbol{\lambda})\right\} = E_{\pi(\boldsymbol{w}|\boldsymbol{Y};\boldsymbol{\lambda}_k)}\left\{\log\pi(\boldsymbol{w}\mid\boldsymbol{Y};\boldsymbol{\lambda})\right\}. \end{aligned}$$

The E-step corresponds to the analytic calculation of the function Q, which is maximized with respect to $\boldsymbol{\lambda}$ at the M-step to provide $\boldsymbol{\lambda}_{k+1}$, as input for the next EM iteration. Jensen's inequality implies that $K(\boldsymbol{\lambda};\boldsymbol{\lambda}_k) \leqslant K(\boldsymbol{\lambda}_k;\boldsymbol{\lambda}_k)$ for all $\boldsymbol{\lambda}$, and since $Q(\boldsymbol{\lambda}_{k+1};\boldsymbol{\lambda}_k) \geqslant Q(\boldsymbol{\lambda}_k;\boldsymbol{\lambda}_k)$, we have $\ell_{\mathrm{M}}(\boldsymbol{\lambda}_{k+1};\boldsymbol{y}) \geqslant \ell_{\mathrm{M}}(\boldsymbol{\lambda}_k;\boldsymbol{y})$. The EM algorithm thus transfers optimization of the log-marginal likelihood ℓ_{M} to that of Q, and ensures that ℓ_{M} increases after every M-step. Under mild conditions, the algorithm is guaranteed to reach at least a local maximum (Dempster et al., 1977). We first construct the function Q used at the E-step.

2.2. E-step

Applying Bayes' rule to the posterior for $\boldsymbol{w}$, the non-trivial element of the function Q in (13) is

$$\begin{aligned} E_{\pi(\boldsymbol{w}|\boldsymbol{Y};\boldsymbol{\lambda}_k)}\left\{\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{Y},\boldsymbol{\lambda})\right\} &= \int \ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda})\ \pi(\boldsymbol{w}\mid\boldsymbol{Y}=\boldsymbol{y};\boldsymbol{\lambda}_k)\,\mathrm{d}\boldsymbol{w}\\ &= \frac{\int \ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda})\exp\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}_k)\,\mathrm{d}\boldsymbol{w}}{\int \exp\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}_k)\,\mathrm{d}\boldsymbol{w}}. \qquad (14) \end{aligned}$$

Both integrals are intractable. Since ℓ_{P} may not be positive, the numerator cannot be expressed as the integral of an exponential function, so direct Laplace approximation is impracticable. Tierney et al. (1989) overcome this by approximating similar ratios using the moment generating function, since (14) is the expectation of a scalar function, ℓ_{P}, of the regression weights, which are here treated as random variables with joint probability density proportional to $\exp\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}_k)$, the posterior density conditional on $\boldsymbol{y}$ as seen from (10). For any $\boldsymbol{w}$, let

$$\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k) = t\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}) + \ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}_k), \quad t\in\mathbb{R}.$$

The conditional moment generating function of $\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{Y},\boldsymbol{\lambda})$ is thus

$$\begin{aligned} \mathrm{M}(t) = E_{\pi(\boldsymbol{w}|\boldsymbol{Y};\boldsymbol{\lambda}_k)}\left[\exp\left\{t\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{Y},\boldsymbol{\lambda})\right\}\right] &= \frac{\int\exp\left\{t\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda})\right\}f(\boldsymbol{y},\boldsymbol{w};\boldsymbol{\lambda})\,\mathrm{d}\boldsymbol{w}}{\int f(\boldsymbol{y},\boldsymbol{w};\boldsymbol{\lambda})\,\mathrm{d}\boldsymbol{w}}\\ &= \frac{\int\exp\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)\,\mathrm{d}\boldsymbol{w}}{\int\exp\ell_0(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)\,\mathrm{d}\boldsymbol{w}}. \qquad (15) \end{aligned}$$

Expression (15) is a ratio of two intractable integrals, each of which can be approximated using the Laplace method. Let

$$\hat{\boldsymbol{w}}_t = \arg\max_{\boldsymbol{w}}\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k), \quad \hat{\boldsymbol{w}}_k = \arg\max_{\boldsymbol{w}}\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}_k) = \arg\max_{\boldsymbol{w}}\ell_0(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)$$

denote the maximizers of $\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)$ and $\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}_k)$, and write the negative Hessian matrix as $\boldsymbol{H}_t(\boldsymbol{w};\boldsymbol{\lambda},\boldsymbol{\lambda}_k) = t\boldsymbol{H}_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{\lambda}) + \boldsymbol{H}_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{\lambda}_k)$, where $\boldsymbol{H}_{\mathrm{P}}$ is given in (8). Second-order Taylor expansion of $\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)$ around $\hat{\boldsymbol{w}}_t$ yields the following approximation for the numerator of (15),

$$\begin{aligned} \int\exp\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)\,\mathrm{d}\boldsymbol{w} &= \exp\ell_t(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)\int\exp\left\{-\frac{1}{2}(\boldsymbol{w}-\hat{\boldsymbol{w}}_t)^T\boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)(\boldsymbol{w}-\hat{\boldsymbol{w}}_t)\right\}\mathrm{d}\boldsymbol{w}\ \{1+O(n^{-1})\}\\ &= (2\pi)^{d/2}\det\boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)^{-1/2}\exp\ell_t(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)\{1+O(n^{-1})\}, \end{aligned}$$

where the determinant is well-defined because $\boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)$ is positive definite at convergence. On similarly applying Laplace approximation to the denominator of (15), the conditional moment generating function becomes

$$\mathrm{M}(t) = \exp\left\{\ell_t(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k) - \ell_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{y},\boldsymbol{\lambda}_k)\right\}\frac{\det\boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)^{-1/2}}{\det\boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k)^{-1/2}}\{1+O(n^{-2})\}, \qquad (16)$$

where the relative approximation error is $O(n^{-2})$ rather than $O(n^{-1})$ because the error terms in the numerator and denominator almost cancel (Tierney et al., 1989, Theorem 1). The conditional expectation (14) is obtained by differentiating (16) with respect to t and evaluating the result at $t=0$.

Whereas Tierney et al. (1989) and Steele (1996) suggest numerical computation of such derivatives, we shall calculate them analytically. We need $\mathrm{d}\ell_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)/\mathrm{d}t$ and $\mathrm{d}\det\boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)/\mathrm{d}t$, both evaluated at $t=0$. To simplify the notation we use $\mathrm{d}_0\cdot/\mathrm{d}t$ to denote $\mathrm{d}\cdot/\mathrm{d}t\,|_{t=0}$, and similarly for $\partial_0\cdot/\partial t$.

Calculation of $\mathrm{d}_0\ell_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)/\mathrm{d}t$.

Since $\hat{\boldsymbol{w}}_t$ depends upon t,

$$\frac{\mathrm{d}\ell_t(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)}{\mathrm{d}t} = \ell_{\mathrm{P}}(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda}) + t\frac{\partial\hat{\boldsymbol{w}}_t}{\partial t}\cdot\frac{\partial\ell_{\mathrm{P}}(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda})}{\partial\boldsymbol{w}} + \frac{\partial\hat{\boldsymbol{w}}_t}{\partial t}\cdot\frac{\partial\ell_{\mathrm{P}}(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda}_k)}{\partial\boldsymbol{w}},$$

where $\cdot$ denotes the scalar product. Since $\hat{\boldsymbol{w}}_t = \hat{\boldsymbol{w}}_k$ at $t=0$, and $\hat{\boldsymbol{w}}_k$ maximizes $\ell_{\mathrm{P}}(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda}_k)$, we obtain

$$\frac{\mathrm{d}_0\ell_t(\hat{\boldsymbol{w}}_t;\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)}{\mathrm{d}t} = \ell_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{y},\boldsymbol{\lambda}). \qquad (17)$$

Calculation of $\mathrm{d}_0\det\boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)/\mathrm{d}t$.

This requires $\partial_0\hat{\boldsymbol{w}}_t/\partial t$, which we obtain by implicit differentiation of $\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)$. At $\boldsymbol{w}=\hat{\boldsymbol{w}}_t$, we have $\partial\ell_t(\boldsymbol{w};\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\lambda}_k)/\partial\boldsymbol{w}\,|_{\boldsymbol{w}=\hat{\boldsymbol{w}}_t} = \boldsymbol{0}_d\in\mathbb{R}^d$, so differentiating with respect to t and setting $t=0$ yields

$$\boldsymbol{U}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}) - \boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k)\cdot\frac{\partial_0\hat{\boldsymbol{w}}_t}{\partial t} = \boldsymbol{0}_d. \qquad (18)$$

As $\boldsymbol{U}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}) = \boldsymbol{U}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k) + \boldsymbol{S}_{\boldsymbol{\lambda}_k}\hat{\boldsymbol{w}}_k - \boldsymbol{S}_{\boldsymbol{\lambda}}\hat{\boldsymbol{w}}_k = \left(\boldsymbol{S}_{\boldsymbol{\lambda}_k} - \boldsymbol{S}_{\boldsymbol{\lambda}}\right)\hat{\boldsymbol{w}}_k$, we get from (18) that

$$\frac{\partial_0 \hat{\boldsymbol{w}}_t}{\partial t} = \boldsymbol{H}_{\mathrm{P}}^{-1}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k)\left(\boldsymbol{S}_{\boldsymbol{\lambda}_k} - \boldsymbol{S}_{\boldsymbol{\lambda}}\right)\hat{\boldsymbol{w}}_k. \tag{19}$$

Let $\hat{\boldsymbol{\theta}}_t = \boldsymbol{X}\hat{\boldsymbol{w}}_t \in \mathbb{R}^{nP}$ and $\hat{\boldsymbol{\theta}}_k = \boldsymbol{X}\hat{\boldsymbol{w}}_k$. Using the chain rule and (19), the derivative of the negative Hessian $\boldsymbol{H}_{\mathrm{L},i}$ of the unpenalized log-likelihood corresponding to a data-point y_i is

$$\frac{\mathrm{d}_0 \boldsymbol{H}_{\mathrm{L},i}(\hat{\boldsymbol{\theta}}_{t,i};y_i)}{\mathrm{d}t} = \sum_{p=1}^{P} \boldsymbol{X}_i^{(p)} \cdot \frac{\partial_0 \hat{\boldsymbol{w}}_t^{(p)}}{\partial t}\frac{\partial \boldsymbol{H}_{\mathrm{L},i}(\hat{\boldsymbol{\theta}}_{k,i};y_i)}{\partial \theta_i^{(p)}}, \quad i = 1,\ldots,n. \tag{20}$$

Applying Jacobi's formula to $\mathrm{d}\det \boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)/\mathrm{d}t$ and evaluating the result at $t = 0$ yields

$$\frac{\mathrm{d}_0 \det \boldsymbol{H}_t(\hat{\boldsymbol{w}}_t;\boldsymbol{\lambda},\boldsymbol{\lambda}_k)}{\mathrm{d}t} = \det \boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k) \times \mathrm{Tr}\left[\boldsymbol{H}_{\mathrm{P}}^{-1}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k)\left\{\boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}) + \frac{\mathrm{d}_0 \boldsymbol{H}_{\mathrm{L}}}{\mathrm{d}t}(\hat{\boldsymbol{w}}_t)\right\}\right], \tag{21}$$

where the last derivative term can be computed using (20).

On inserting (21) and (17) into the derivative of (16) with respect to t and evaluating the result at $t = 0$, we find after a little algebra that

$$Q(\boldsymbol{\lambda};\boldsymbol{\lambda}_k) = \frac{1}{2}\log|\boldsymbol{S}_{\boldsymbol{\lambda}}|_+ + \left\{-\frac{1}{2}\ \mathrm{Tr}\left[\boldsymbol{H}_{\mathrm{P}}^{-1}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k)\left\{\boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}) + \frac{\mathrm{d}_0 \boldsymbol{H}_{\mathrm{L}}(\hat{\boldsymbol{w}}_t)}{\mathrm{d}t}\right\}\right] + \ell_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{y},\boldsymbol{\lambda})\right\}\left\{1 + O(n^{-2})\right\}.$$

The order $O(n^{-2})$ of the relative error in Q over the usual $O(n^{-1})$ error for Laplace approximation may suggest that the approximate E-step provides a potentially better approximation to the function to be maximized for the smoothing hyper-parameters. We discuss this further in Appendix A.

The proposed approach is clearly an outer iteration optimization, since Q is defined in terms of the maximizer $\hat{\boldsymbol{w}}_k$ rather than its intermediate estimate, as in the performance iteration optimization; see Section 1.2. As we shall see, this approximate E-step greatly simplifies the M-step; the crux is that $\hat{\boldsymbol{w}}_k$ depends on $\boldsymbol{\lambda}_k$ alone, and not on $\boldsymbol{\lambda}$.

2.3. M-step

The M-step entails the calculation of the gradient and Hessian matrix of Q with respect to the smoothing hyper-parameters $\boldsymbol{\lambda}$. We first show that the derivative of $\partial_0 \hat{\boldsymbol{w}}_t/\partial t$ in (19) with respect to $\boldsymbol{\lambda}$ equals $\partial \hat{\boldsymbol{w}}_k/\partial \boldsymbol{\lambda}_k$. As $\hat{\boldsymbol{w}}_k$ is the solution to the equation $\boldsymbol{U}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k) = \boldsymbol{0}_d$, taking the derivative with respect to the j-th component $\lambda_{k,j}$ of $\boldsymbol{\lambda}_k$ yields

$$\frac{\partial \hat{\boldsymbol{w}}_k}{\partial \lambda_{k,j}} = -\boldsymbol{H}_{\mathrm{P}}^{-1}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k)\boldsymbol{S}_j\hat{\boldsymbol{w}}_k = \frac{\partial \partial_0 \hat{\boldsymbol{w}}_t}{\partial \lambda_j \partial t}, \tag{22}$$

since $\partial \boldsymbol{S}_{\boldsymbol{\lambda}_{k,j}}/\partial \lambda_{k,j} = \boldsymbol{S}_j = \partial \boldsymbol{S}_{\boldsymbol{\lambda}}/\partial \lambda_j$. Combining (22) with (20) yields

$$\frac{\partial \mathrm{d}_0 \boldsymbol{H}_{\mathrm{L},i}(\hat{\boldsymbol{\theta}}_{t,i};y_i)}{\partial \lambda_j \mathrm{d}t} = \sum_{p=1}^{P} \boldsymbol{X}_i^{(p)} \cdot \frac{\partial \hat{\boldsymbol{w}}_k^{(p)}}{\partial \lambda_{k,j}}\frac{\partial \boldsymbol{H}_{\mathrm{L},i}(\hat{\boldsymbol{\theta}}_{k,i};y_i)}{\partial \theta_i^{(p)}} = \frac{\partial \boldsymbol{H}_{\mathrm{L},i}(\hat{\boldsymbol{\theta}}_{k,i};y_i)}{\partial \lambda_{k,j}},$$

and

$$\frac{\partial \mathrm{d}_0 \boldsymbol{H}_{\mathrm{L}}}{\partial \lambda_j \mathrm{d}t}(\hat{\boldsymbol{w}}_t) = \frac{\partial \boldsymbol{H}_{\mathrm{L}}}{\partial \lambda_{k,j}}(\hat{\boldsymbol{w}}_k). \tag{23}$$

Let $\hat{\boldsymbol{w}}_k^{(j)}$ denote the block of $\hat{\boldsymbol{w}}_k$ corresponding to $\boldsymbol{S}_j$ and the smooth function f_j. Using (23), the components of the gradient of the function Q at the E-step are

$$G_j(\boldsymbol{\lambda};\boldsymbol{\lambda}_k) = \frac{\partial Q(\boldsymbol{\lambda};\boldsymbol{\lambda}_k)}{\partial \lambda_j} = \frac{1}{2}\left\{\mathrm{Tr}\left(\boldsymbol{S}_{\boldsymbol{\lambda}}^{-}\boldsymbol{S}_j\right) - c_{k,j}\right\}, \tag{24}$$

where

$$c_{k,j} = \hat{\boldsymbol{w}}_k^{(j)^T}\boldsymbol{S}_j\hat{\boldsymbol{w}}_k^{(j)} + \mathrm{Tr}\left[\boldsymbol{H}_{\mathrm{P}}^{-1}(\hat{\boldsymbol{w}}_k;\boldsymbol{\lambda}_k)\left\{\boldsymbol{S}_j + \frac{\partial \boldsymbol{H}_{\mathrm{L}}}{\partial \lambda_{k,j}}(\hat{\boldsymbol{w}}_k)\right\}\right] \in \mathbb{R}.$$

By construction in (6), $\boldsymbol{S}_{\boldsymbol{\lambda}}$ is a block-diagonal matrix whose blocks are of the general form $\boldsymbol{S} = \lambda_j \boldsymbol{S}_j$, which implies that $\mathrm{Tr}(\boldsymbol{S}_{\boldsymbol{\lambda}}^{-}\boldsymbol{S}_j) = \mathrm{rank}(\boldsymbol{S}_j)/\lambda_j$, and yields the closed form

$$\hat{\lambda}_{k+1,j} = \frac{\mathrm{rank}(\boldsymbol{S}_j)}{c_{k,j}}, \quad j = 1,\ldots,q, \tag{25}$$

where $c_{k,j} > 0$ by positivity of the smoothing hyper-parameters. The Hessian matrix of Q is therefore diagonal with negative elements $-\mathrm{rank}(\boldsymbol{S}_j)/(2\lambda_j^2) < 0$, so (25) are always maximizers. The components of $\boldsymbol{\lambda}$ are positive, so it might be thought necessary to set $\boldsymbol{\rho} = \log \boldsymbol{\lambda}$ component-wise before the approximate EM optimization, and then back-transform afterwards. This would have led to finding the roots of

$$\tilde{G}_j(\boldsymbol{\rho};\boldsymbol{\rho}_k) = \frac{\partial Q(\exp \boldsymbol{\rho};\exp \boldsymbol{\rho}_k)}{\partial \rho_j} = G_j(\boldsymbol{\lambda};\boldsymbol{\lambda}_k)\exp \rho_j,$$

which are also the roots of G_j in (24), with Hessian components $-(c_{k,j}\exp \rho_j)/2 < 0$, so the positivity constraint need not be explicitly included. The diagonality of the Hessian matrix of Q allows embarrassingly parallel computation of the M-step, which provides substantial speedup when q, the number of smooth functions, is large.

Overall, the k-th iteration of the approximate EM algorithm consists in

1. using the current best estimate $\boldsymbol{\lambda}_k$ to maximize the penalized log-likelihood (7) to get $\hat{\boldsymbol{w}}_k$;
2. computing $\boldsymbol{\lambda}_{k+1}$, possibly in parallel, using (25);
3. updating $k+1$ to k.

Learning of the regression weights is incorporated into Step 1, which is based on the Newton–Raphson algorithm. Given the trial value $\boldsymbol{w}_l$, each iteration involves

a) making $\boldsymbol{H}_{\mathrm{P}}(\boldsymbol{w}_l;\boldsymbol{\lambda}_k)$ positive definite;

b) evaluating the updating step

$$\boldsymbol{w}_{l+1} = \boldsymbol{w}_l + \alpha \boldsymbol{\Delta}_l, \quad \boldsymbol{\Delta}_l = \boldsymbol{H}_{\mathrm{P}}^{-1}(\boldsymbol{w}_l; \boldsymbol{\lambda}_k) \boldsymbol{U}_{\mathrm{P}}(\boldsymbol{w}_l; \boldsymbol{\lambda}_k),$$

where α is the learning rate. At step a), the positive definiteness of $\boldsymbol{H}_{\mathrm{P}}(\boldsymbol{w}_l; \boldsymbol{\lambda}_k)$ is guaranteed by increasing eigenvalues smaller than a certain positive tolerance to that tolerance. The stability of the algorithm is ensured by successively halving α at Step b) until the penalized log-likelihood increases.

At convergence of the Newton–Raphson algorithm, $\hat{\boldsymbol{w}}_k = \boldsymbol{w}_{l+1}$, and the identifiability of the regression weights must be checked to ensure that $\boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k; \boldsymbol{\lambda}_k)$ is invertible, since this matrix is required for calculating the smoothing hyper-parameters. By definition, the regression model is identifiable if and only if its weights are linearly independent, so we deal with lack of identifiability by keeping only the $r := \operatorname{rank}\left\{\boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k; \boldsymbol{\lambda}_k)\right\} \leqslant d$ linearly independent regression weights. An efficient and stable method to find these is QR decomposition with column pivoting (Golub and Van Loan, 2013, Section 5.4.2). The QR factorization finds a permutation matrix $\boldsymbol{P} \in \mathbb{R}^{d \times d}$ such that $\boldsymbol{H}_{\mathrm{P}}(\hat{\boldsymbol{w}}_k; \boldsymbol{\lambda}_k)\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{R}$, where the first r columns of $\boldsymbol{Q}$ form an orthonormal basis for $\boldsymbol{H}_{\mathrm{P}}$. As the permutation matrix tracks the moves of the columns of $\boldsymbol{H}_{\mathrm{P}}$, the r identifiable weights are the first r components of the re-ordered vector $\boldsymbol{P}^T \hat{\boldsymbol{w}}_k$. Thus the remaining $d - r$ weights are linearly dependent and should be excluded from the model, together with the corresponding columns of $\boldsymbol{X}$ and rows and columns of $\boldsymbol{S}_{\boldsymbol{\lambda}_k}$.

Steps 1–3 are iterated until the gradient of the approximate log-marginal likelihood is sufficiently small. Oakes (1999) showed that this gradient can be written in terms of that of Q, as $\partial \ell_{\mathrm{M}}(\boldsymbol{\lambda}; \boldsymbol{y})/\partial \boldsymbol{\lambda} = G(\boldsymbol{\lambda}; \boldsymbol{\lambda}_k)|_{\boldsymbol{\lambda}_k = \boldsymbol{\lambda}}$. Since $G(\boldsymbol{\lambda}_{k+1}; \boldsymbol{\lambda}_k) = 0$, the convergence criterion is equivalent to checking that for each j,

$$G_j(\lambda_{k+1,j}) = \frac{1}{2}(c_{k,j} - c_{k+1,j}) < \epsilon,$$

where ϵ is a small tolerance. Furthermore, the diagonality of the Hessian of Q allows one to check convergence independently for each smoothing hyper-parameter, so that only unconverged ones need be updated. In practice, the smoothing hyper-parameters may be large enough that significant changes in some components of $\boldsymbol{\lambda}$ yield insignificant changes in the penalized log-likelihood, which suggests deeming convergence when there is no significant change in ℓ_{P}. The full optimization is summarized in the three-step iteration, whose leading computational costs in the worst-case scenario are $O(nd^2)$ for the computation of the Hessian of the log-likelihood, $O(d^3)$ for its inversion, and $O(nd^2)$ for its derivative.

The approximate EM algorithm provides an elegant and straightforward approach to maximization of the log-marginal likelihood. We obtained an accurate E-step based on the approximation of Tierney et al. (1989) with relative error $O(n^{-2})$, and derived a closed form for the M-step that circumvents evaluation of the expensive and numerically unstable function Q. This indirect approach leads to an important simplification of the learning procedure compared to the direct Laplace approach. Since the M-step is upward, no learning rate tuning is required as in the Newton–Raphson algorithm: there is no need for intermediate evaluation of the log-marginal likelihood or its Hessian matrix. The former circumvents inner maximizations of the penalized log-likelihood and evaluation of the log-generalized

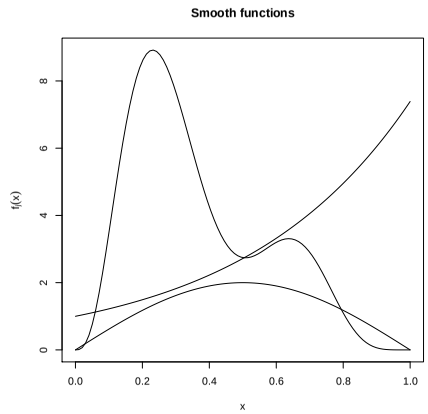


Figure 1: Original f_j for $j = 1, \ldots, 3$.

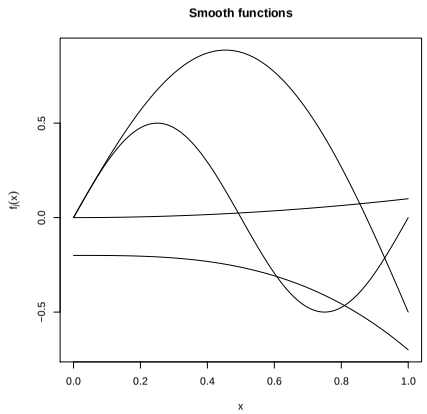


Figure 2: Original f_j for $j = 4, \ldots, 7$.

determinant in (12), which is unstable when the components of $\boldsymbol{\lambda}$ differ in magnitude. Furthermore, avoiding evaluation of the Hessian matrix of ℓ_{M} bypasses computation of the fourth derivatives of the log-likelihood ℓ_{L}, which may be difficult to calculate, computationally expensive and numerically unstable (Wood, 2011; Wood et al., 2016). Moreover, the diagonality of the Hessian matrix of Q allows parallelization of the M-step and update of the unconverged smoothing hyper-parameters only, giving an additional shortcut.

We assess the performance of the proposed methodology in Section 3.

3. Simulation Study

We generated $R = 100$ replicates of training sets of $n = 25000$ examples from a variety of probability distributions with parameters that depend on smooth functions of inputs. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_7$ be independent vectors of n identically distributed standard uniform variables. Figures 1 and 2 illustrate the seven smooth functions we considered,

$$\begin{aligned} f_1(x) &= 10^4 x^3 (1-x)^6 \left\{(1-x)^4 + 20x^8\right\}, \quad f_2(x) = 2\sin(\pi x), \quad f_3(x) = \exp(2x), \\ f_4(x) &= 0.1x^2, \quad f_5(x) = \sin(2\pi x)/2, \quad f_6(x) = -0.2 - x^3/2, \quad f_7(x) = -x^2/2 + \sin(\pi x). \end{aligned}$$

With the functional parameters

$$\mu(x_{i1}, x_{i2}, x_{i3}) = \sum_{j=1}^{3} f_j(x_{ij}), \quad \sigma(x_{i4}, x_{i5}, x_{i6}) = \sum_{j=4}^{6} f_j(x_{ij}), \quad \xi(x_{i7}) = f_7(x_{i7}), \quad i = 1, \ldots, n,$$

we generated n training examples from the following models:

- Gaussian distribution with mean $\mu(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ and standard deviation $\exp\{\sigma(\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6)/2\}$,

- Poisson distribution with rate $\exp\{\mu(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)/6\}$,
- Exponential distribution with rate $\exp\{\mu(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)/6\}$,
- Gamma distribution with shape $\exp\{\mu(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)/6\}$ and scale $\exp\{-\sigma(\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6)\}$,
- Binomial distribution with probability of success $1/[1 + \exp\{-\mu(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) + 5\}/6]$,
- Generalized extreme value (GEV) distribution with location $\mu(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$, scale $\exp \sigma(\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6)$ and shape $\xi(\boldsymbol{x}_7)$; see Section 4.1 for further details.

We fit the six models using cubic regression splines with evenly spaced knots in the input range values. We used ten basis functions for each of the smooth functions f_j. We computed the integrated mean squared error between the true and learned functional parameters, represented by hats, for each of the r replicates

$$\mathrm{MSE}(\hat{\boldsymbol{\theta}}^{(p)[r]}) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_i^{(p)[r]} - \hat{\theta}_i^{(p)[r]}\right)^2,$$

where $\hat{\boldsymbol{\theta}}^{(p)}$ is $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\sigma}}$ or $\hat{\boldsymbol{\xi}}$. Table 1 summarizes the results for the proposed approach, `multgam`, and three state-of-the-art methods implemented in the `R` packages `mgcv` gam (Wood, 2011; Wood et al., 2016) using the Newton–Raphson method, `mgcv` bam (Wood et al., 2015), and `INLA` (Rue et al., 2009). The routines used from `mgcv` are based on Version 1.8-22, and those used from `INLA` are based on Version 18.07.12 run with eight threads, using log-gamma priors and the random walk parametrization of order two. We also tried both Stan algorithms (Carpenter et al., 2017), a fully Bayesian approach with Markov chain Monte Carlo sampling and an approximate variational Bayes approach, through the `R` package `brms` (Burkner, 2017) Version 2.4.0, but a single replicate for a single functional parameter model run with four cores took five and three hours respectively, so the full simulation study would have taken more than four months, which is infeasible. Another widely-used `R` package, `VGAM`, does not offer automatic smoothing, and choosing $\boldsymbol{\lambda}$ manually for each f_j for each model would have been tedious and error-prone. Use of the `R` package `gamlss` Version 5.1-0 turned out to be infeasible. Some results for the Gauss, Gamma and GEV models are missing from Table 1 because the corresponding packages do not support them. Moreover, `multgam` failed on 17 replicates for the GEV model, whereas `mgcv` gam failed on 46 replicates, so the values shown are based on 83 and 54 training sets respectively.

Table 1 shows that `multgam` is the only package which supports all the classical models, and its small errors and low variances demonstrate the high accuracy and reliability of its estimates. The proposed method is competitive with both methods in `mgcv`, whereas `INLA` is less accurate. The new method is considerably better for the GEV model; it could fit 83 of the replicates, compared to 54 for `mgcv` gam, and the estimates themselves were more accurate and less variable. The only model where all the methods give equally poor results is the binomial.

Table 2 gives a timing comparison for training sets of different sizes generated from the models described above. The computations were performed on a 2.80 GHz Intel i7-7700HQ laptop using Ubuntu. The proposed method is always fastest, more so for large training sets, and substantially outperforms `mgcv` gam and `INLA`. Moreover, it can fit the GEV

Model	Package	$\hat{\boldsymbol{\mu}}$	$\hat{\boldsymbol{\sigma}}$	$\hat{\boldsymbol{\xi}}$
Gauss	`multgam`	$2.47_{2.93}$	$0.04_{0.01}$	–
	`mgcv` gam	$2.47_{2.93}$	$0.04_{0.01}$	–
Poisson	`multgam`	$1.63_{4.56}$	–	–
	`mgcv` gam	$1.61_{4.67}$	–	–
	`mgcv` bam	$1.62_{7.33}$	–	–
	`INLA`	$9.91_{7.48}$	–	–
Exponential	`multgam`	$3.67_{114.02}$	–	–
	`mgcv` gam	$3.75_{115.26}$	–	–
	`mgcv` bam	$3.69_{123.74}$	–	–
	`INLA`	$11.95_{87.06}$	–	–
Gamma	`multgam`	$1.78_{9.27}$	$0.02_{0.01}$	–
Binomial	`multgam`	$38.51_{0.99}$	–	–
	`mgcv` gam	$38.51_{0.99}$	–	–
	`mgcv` bam	$38.51_{0.99}$	–	–
	`INLA`	$38.51_{0.99}$	–	–
GEV	`multgam`	$3.58_{8.61}$	$0.15_{0.11}$	$0.41_{1.06}$
	`mgcv` gam	$3.63_{8.56}$	$0.27_{77.63}$	$0.67_{336.54}$

Table 1: Means ($\times 10^{-2}$) over 100 replicates of the integrated mean squared errors of the learned functional parameters for a variety of models and `R` packages. The variances ($\times 10^{-6}$) appear as subscripts.

model at sizes unmatched by existing software. The package `INLA` fails with a half-million observations for all the models. Surprisingly, the proposed method is faster than `mgcv` bam, which is specifically designed for large data sets and exploits parallel computing, whereas `multgam` performs the M-step serially for fair comparison. Furthermore, the speed of `mgcv` bam should be balanced by lack of reliability of the performance iteration algorithm; see Section 1.2. Table 2 shows that speed and reliability need not be exclusive. One reason why `mgcv` gam is slow is that it evaluates the fourth derivatives of the log-likelihood. Except for the GEV model, these are not difficult to compute, but they can entail significant computational overhead, shown by the difference in performance between `multgam` and `mgcv` gam.

Overall, the new approach gives a large gain in speed with no loss in accuracy, and in some cases, it is the sole approach feasible. In Section 4 we apply it to real data.

4. Data Analysis

We analyze monthly maxima of temperature, which are non-stationary. Using stationary models to fit them could result in underestimation of risk, with serious potential conse-

Model	Package	2.5^4	1^5	5^5	$R_{2.5^4}$	R_{1^5}	R_{5^5}
Gauss	`multgam`	3.38	17.40	33.92	1	1	1
	`mgcv` gam	51.22	220.96	1269.16	15.15	12.70	37.42
	`brms` MCMC	×	$359140^{(\times)}$	$387242^{(\times)}$	×	×	×
	`brms` VB	×	$384702^{(\times)}$	$394031^{(\times)}$	×	×	×
Poisson	`multgam`	0.33	4.05	16.39	1	1	1
	`mgcv` gam	4.25	60.27	2157.61	12.88	14.88	131.64
	`mgcv` bam	1.37	9.83	30.30	4.15	2.43	1.85
	`INLA`	459.71	12077.35	×	1393.06	2982.06	×
Exponential	`multgam`	0.71	5.03	19.89	1	1	1
	`mgcv` gam	4.67	56.55	340.15	6.58	11.24	17.10
	`mgcv` bam	1.49	6.83	33.13	2.10	1.36	1.67
	`INLA`	466.77	×	×	657.42	×	×
Gamma	`multgam`	2.67	30.75	97.04	1	1	1
Binomial	`multgam`	0.38	7.90	19.27	1	1	1
	`mgcv` gam	3.33	58.51	463.46	8.76	7.41	24.05
	`mgcv` bam	1.23	8.81	22.60	3.24	1.12	1.17
	`INLA`	299.91	11543.32	×	789.24	1461.18	×
GEV	`multgam`	8.24	440.22	×	1	1	×
	`mgcv` gam	167.13	×	×	20.28	×	×

Table 2: Times (s) for a variety of models and three training set sizes $n = 2.5 \times 10^4$, 5×10^5, 5×10^5. The times correspond to averages over 100 replicates when $n = 2.5 \times 10^4$. The notation x^y means $x \times 10^y$. The notation $t^{(\times)}$ means the computation failed to converge after t seconds, and $\times$ indicates failure to converge or that computations turned out to be infeasible. The ratios R_{x^y} are with respect to `multgam`, which does not benefit from the parallelization of the M-step.

quences. The generalized extreme-value distribution, widely used for modeling maxima and minima, will serve as our underlying probability model.

4.1. Model

Let $y_1, \ldots, y_n$ be the maxima of blocks of observations from an unknown probability distribution. Extreme value theory (Fisher and Tippett, 1928; de Haan and Ferreira, 2006) implies that as the block size increases and under mild conditions, each of the y_i follows a GEV(μ_i, σ_i, ξ_i) distribution with parameters the location $\mu_i \in \mathbb{R}$, the scale $\sigma_i > 0$ and the

shape $\xi_i \in \mathbb{R}$,

$$F(y_i; \mu_i, \sigma_i, \xi_i) = \begin{cases} \exp\left[-\left\{1 + \xi_i\left(\dfrac{y_i - \mu_i}{\sigma_i}\right)\right\}_+^{-1/\xi_i}\right], & \xi_i \neq 0, \\ \exp\left[-\exp\left\{-\left(\dfrac{y_i - \mu_i}{\sigma_i}\right)\right\}\right], & \xi_i = 0, \end{cases}$$

where $a_+ = \max(a, 0)$. This encompasses the three classical models for maxima (Jenkinson, 1955): if $\xi_i > 0$, the distribution is Fréchet; if $\xi_i < 0$, it is reverse Weibull; and if $\xi_i = 0$, it is Gumbel. The shape parameter is particularly important since it controls the tail properties of the distributions. The expectation of Y_i is

$$E(Y_i) = \begin{cases} \mu_i + \dfrac{\sigma_i}{\xi_i}\left\{\Gamma(1 - \xi_i) - 1\right\}, & \xi_i \neq 0,\ \xi_i < 1, \\ \mu_i + \gamma\sigma_i, & \xi_i = 0, \\ \infty, & \xi_i \geqslant 1, \end{cases} \tag{26}$$

where γ is Euler's constant. Non-stationarity of (26) could stem from changes in any of the parameters, and as interpretability is priority in risk assessment, a GEV model with functional parameters having flexible additive structure is well justified.

Most data analyses involving non-stationary extremes use a parametric or semi-parametric form in the location and/or scale parameters while keeping the shape a fixed scalar (Chavez-Demoulin and Davison, 2012, Section 4), even though it may be plausible that this varies. Seasonal effects, for example, may stem from different physical processes with different extremal behaviors. Fixing the shape parameter in such cases is a pragmatic choice driven by the difficulty of learning it from limited data in a numerically stable manner. Chavez-Demoulin and Davison (2005) learn a functional shape parameter for the generalized Pareto distribution, but their approach has some drawbacks. First, training is based on backfitting, which does not allow automatic smoothness selection, and so involves manual tuning of the smoothing hyper-parameters. Second, the optimization is in the spirit of performance iteration, with one updating step for the regression model followed by one full optimization for smoothing; limitations of this were discussed in Section 1.2. Third, optimization is sequential rather than simultaneous, by alternating a regression step for each smooth term when there are several, and alternating backfitting steps for each functional parameter separately. Fourth, convergence may only be guaranteed when the functional parameters are orthogonal, meaning that the methodology may not extend to more than two. Moreover, the smoothing method is applied to orthogonalized distribution parameters that may be awkward to interpret. We learn a functional shape in a generic and stable manner; this is of separate interest for the modeling of non-stationary extremes.

In our earlier general terms, $\theta_i^{(1)} = \mu_i$, $\theta_i^{(2)} = \exp\tau_i$, where $\tau_i = \log\sigma_i$ to ensure positivity of the scale, and $\theta_i^{(3)} = \xi_i$. Let Ω and Ω_0 denote the partition of the support as

$$\begin{aligned} \Omega &= \left\{y_i \in \mathbb{R} : \xi_i > 0, y_i > \mu_i - \exp\tau_i/\xi_i\right\} \cup \left\{y_i \in \mathbb{R} : \xi_i < 0, y_i < \mu_i - \exp\tau_i/\xi_i\right\}, \\ \Omega_0 &= \left\{(y_i, \xi_i) :\ y_i \in \mathbb{R}, \xi_i = 0\right\}. \end{aligned}$$

The corresponding log-likelihood is then

$$\ell_{\mathrm{L}}(\boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\xi}; \boldsymbol{y}) \quad = \quad \sum_{i=1}^{n} \ell_{\mathrm{L}}^{(i)}(\mu_i, \tau_i, \xi_i; y_i),$$

where the individual contributions are

$$\ell_{\mathrm{L}}^{(i)}(\mu_i, \tau_i, \xi_i; y_i) \quad = \quad \begin{cases} -\tau_i - \left(1 + \dfrac{1}{\xi_i}\right) \log(1 + z_i) - (1 + z_i)^{-1/\xi_i}, & y_i \in \Omega, \\ -\tau_i - \exp(-z_i) - z_i, & (y_i, \xi_i) \in \Omega_0, \end{cases}$$

with

$$z_i = \begin{cases} (y_i - \mu_i)\xi_i \exp(-\tau_i), & y_i \in \Omega, \\ (y_i - \mu_i) \exp(-\tau_i), & (y_i, \xi_i) \in \Omega_0. \end{cases}$$

This log-likelihood becomes numerically unstable when ξ_i and z_i are close to zero, while overflow is amplified as the order of the derivatives increases. The proposed approximate EM method requires the third derivatives of the log-likelihood, which involve terms like ξ_i^{-5}. When $\xi_i \approx 0$, the threshold below which the absolute value of the shape parameter should be set to zero is therefore troublesome. Its value should reflect the compromise between stability of the derivatives and the switch from the general GEV form to the Gumbel distribution. The numerical instability is even more problematic in the `mgcv` gam method, which requires the fourth log-likelihood derivatives; the lower the order of the derivatives, the fewer unstable computations. In our implementation, we set $\xi_i = 0$ whenever $|\xi_i| \leq \varpi^{3/10}$, with ϖ the machine precision. This sets the order of the threshold to 10^{-5}, while allowing negative exponents of ξ_i terms to grow up to 10^{25}, which is within the range of precision of all modern machines.

4.2. Application

We analyze monthly maxima of the daily Central England Temperature (CET)[1] series from January 1772 to December 2016. Figure 3 shows yearly maxima and suggests that the recent years are the warmest, while panel a) in Figure 4 indicates that any increase is most apparent at the end of the year. Figure 4 exhibits obvious seasonality, which we represent using 12 basis functions from cyclic cubic regression splines for each of the location, scale and shape parameters of the GEV model; we use ten basis functions from thin plate splines (Wood, 2003) in the location for the trend visible in Figure 3. We initially included long-term trends in the scale and shape, but they were not significant. We also tried more complex models with larger basis dimensions, but these did not improve the accuracy of the estimates. To our knowledge, this is the only paper modeling a variable shape parameter for this data set. Neither of the algorithms in Stan (Carpenter et al., 2017) using the `R` package `brms` (Burkner, 2017) converged, and the variational Bayes approach faced numerical instabilities.

Panel a) of Figure 5 shows annual variation of 11°C, similar to that seen in Figure 4, and panel b) of Figure 5 shows a non-linear trend with a drop from 1772 to 1800 and a sharp increase from the 1960s onwards. The pattern between is hard to discern in Figure 3,

1. The data can be downloaded at `https://www.metoffice.gov.uk/hadobs/hadcet/data/download.html`

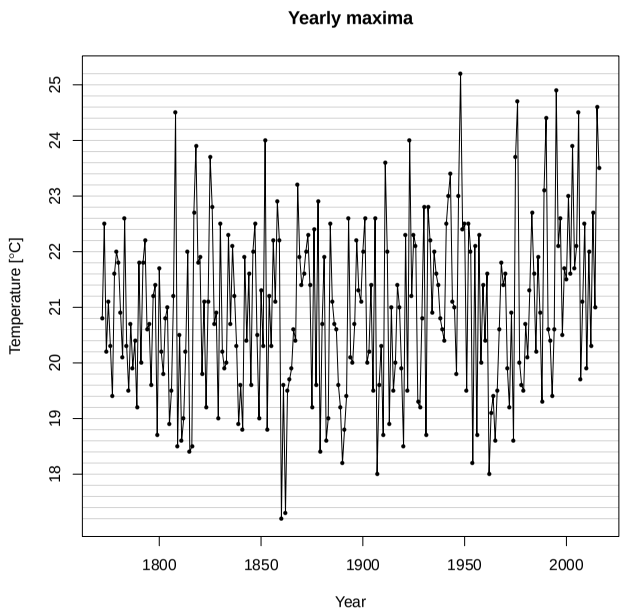


Figure 3: Annual maxima for the CET data.

but panel b) shows an overall increase of about 1.5°C from 1800 onwards and peaks over the last few decades.

The scale and shape parameters in Figure 5, whose functional forms vary significantly through the year, give insight into the seasonality. They are negatively correlated except in mid-June to September, where the increase in the shape is much slower and weaker than the drop in the scale. We can distinguish two cycles within the year, with similar patterns but different intensities: the extended winter from September to April, and the extended weak summer, from April to September. Each incorporates two antagonistic phases that are negatively correlated, alternating between decrease and increase for the shape, and vice-versa for the scale.

Figure 5 summarizes the influences of the scale and the shape parameters on the seasonality of the CET data as follows: whether the temperature is increasing or decreasing seems to be smoothly related to the direction of the shape in the winter, and to that of the scale in the summer. Since the former controls the tail of the distribution and is always significantly negative here, the temperature is bounded above throughout the year; the strongest

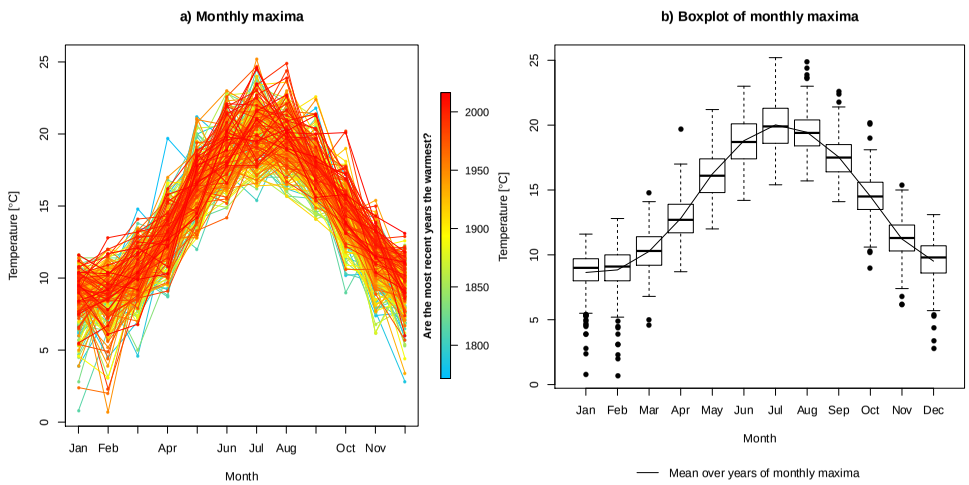


Figure 4: Monthly maxima for the CET data. Left: annual cycles with the year coded by color according to the scale to the right of the panel. Right: boxplots for monthly maxima.

increase of the shape occurs in February to mid-April, early spring, stabilizing around its highest values, -0.2 or so, in the summer. This stabilization and the negative correlation between the scale and the shape explain why the sharper fluctuations of the scale have more impact on the temperature in the summer than the near-constant shape. The rather narrow point-wise confidence intervals suggest that there is very strong evidence for seasonal variation of the shape, and less strong but still appreciable evidence of such variation for the scale. Davison and Ramesh (2000) fitted a local likelihood smooth model to the five largest daily data (after declustering) for each year up to 1996, using linear polynomials for the location and the scale parameters and a constant shape parameter. Their results are consistent with ours in that the changes in the upper extremes are driven by the changes in the scale parameter.

Figures 6 and 7 illustrate model fit diagnostics. Figure 6 shows that the true maxima are within the range of those simulated from the learned model. Figure 7 represents the predicted 0.95, 0.98 and 0.99 quantiles for monthly maxima. Based on the model for 1916, only one value from previous years, 24.5°C in July 1808, exceeded the maximum of the 0.99 quantile curve, 24.4°C, in July; all other exceedances occur after 1916. The maximum of the 0.99 quantile curve in 2016 occurred in July at 25.4°C, and no higher temperature has been observed.

Overall, the model does not seem unrealistic, although it may underestimate slightly the uncertainty, as it assumes independence of maxima in successive months. A possible

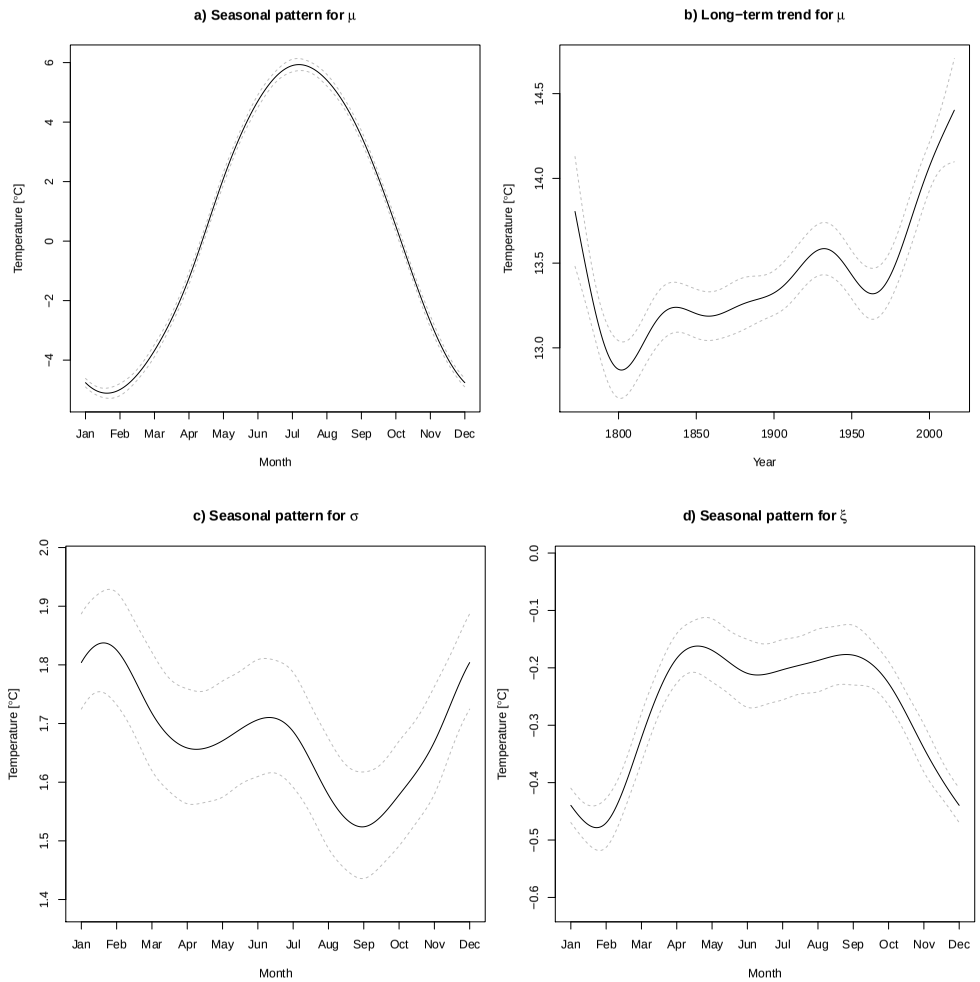


Figure 5: Learned functional parameters, with 95% point-wise confidence intervals (dashes) obtained from the Hessian of the penalized log-likelihood.

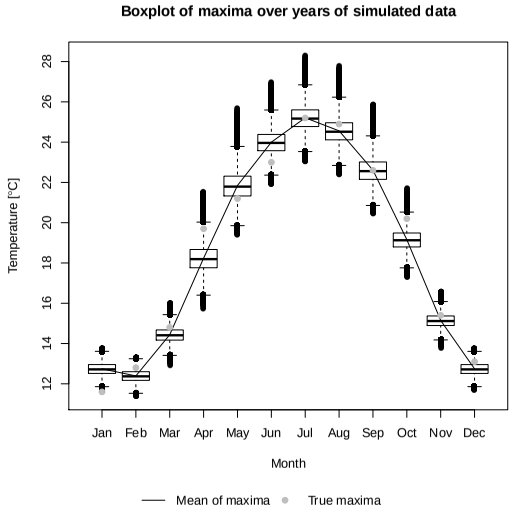


Figure 6: Monthly maxima simulated from the learned GEV model.

improvement would be the use of dependent errors, but this is outside the scope of the present paper.

5. Discussion

This paper makes contributions to optimal and automatic smoothing for generalized additive models using an empirical Bayes approach. The roughness penalty corresponds to a weighted L_2 regularization interpreted as a Gaussian prior on the regression weights, and the penalized log-likelihood is the associated posterior. We maximize the resulting log-marginal likelihood with respect to the smoothing hyper-parameters using an EM algorithm, made tractable via double Laplace approximation of the moment generating function underlying the E-step. The proposed approach transfers maximization of the log-marginal likelihood to a function whose maximizer has a closed form, and avoids evaluation of expensive and numerically unstable terms. The only requirement is that the log-likelihood has third derivatives. The proposed method is stable, accurate and fast. Its stability is ensured both by the EM algorithm and by its need for fewer derivatives, making the proposed method broadly applicable for complex models. Its high accuracy is established theoretically by Tierney et al. (1989), with an $O(n^{-2})$ relative error in the Laplace approximation at the E-step, and we show in Appendix A that this leads to an error of order $O(n^{-3})$ in the learned smoothing hyper-parameters. Its serial implementation is substantially faster than the best existing

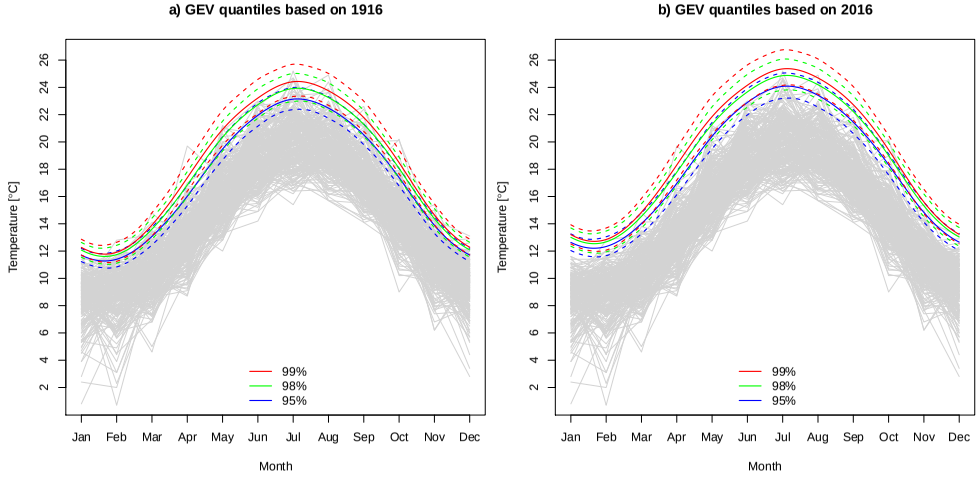


Figure 7: Superposition of the original data (grey) and quantiles of the GEV models, with 95% point-wise confidence intervals (dashes) obtained from the Hessian of the penalized log-likelihood.

methods and achieves state-of-the-art accuracy. It can easily be parallelized, making it appealing for extension to big-data settings, where no reliable method yet exists.

These advantages are balanced by potential difficulties. First, the EM algorithm can be slow around the optimum, though the simulations of Section 3 required no more than fifty iterations. Tests show that deceleration occurs when certain smoothing hyper-parameters become so large that their corresponding smooth functions are linear, and their updates no longer change the penalized log-likelihood. At that point, we declare convergence for those components of $\boldsymbol{\lambda}$, though they may keep changing without affecting the regression weights. Validating convergence for a portion of smoothing hyper-parameters and updating the remainder is supported by the diagonality of the Hessian matrix of the function Q at the E-step. Second, the EM algorithm is known to suffer from local optima, though we found none in the data sets and the simulated models we analyzed, perhaps because the log-likelihood is fairly quadratic for large samples.

The proposed method is implemented in a C++ library that uses Eigen (Guennebaud et al., 2018) for matrix decompositions, is integrated into the `R` package `multgam` through the interface `RcppEigen` (Bates and Eddelbuettel, 2013), and makes addition of further probability models straightforward.

Acknowledgments

The work was financially supported by the ETH domain Competence Center Environment and Sustainability (CCES) project Osper.

Appendix A. Approximation error of the smoothing hyper-parameters

We discuss the approximation error on the final estimate of the smoothing hyper-parameters implied by the approximate E-Step in Section 2.2. Let $\ell^*_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y})$ denote the Laplace approximation to the log-marginal likelihood $\ell_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y})$ based on n independent observations. If we write the relative error of the Laplace approximation for the corresponding density functions as $1 + A(\boldsymbol{\lambda};\boldsymbol{y})/n^a + O(n^{-a-1})$ for some positive a, then on taking logs we obtain

$$\ell_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y}) = \ell^*_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y}) + A(\boldsymbol{\lambda};\boldsymbol{y})/n^a + O(n^{-a-1}), \tag{27}$$

where $\ell_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y})$ and $\ell^*_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y})$ are each essentially sums of n terms, and thus are $O(n)$, and $A(\boldsymbol{\lambda};\boldsymbol{y})$ is $O(1)$.

Now suppose that $\ell_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y})$ has two continuous derivatives and an invertible Hessian matrix throughout an open convex subset $O \subset \mathbb{R}^q$ surrounding the maximum likelihood estimate $\hat{\boldsymbol{\lambda}}$. Then for any $\boldsymbol{\lambda}, \boldsymbol{\lambda}^* \in O$, the mean value theorem implies that we can write

$$\frac{\partial \ell_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y})}{\partial \lambda_j} = \frac{\partial \ell_{\mathrm{M}}(\boldsymbol{\lambda}^*;\boldsymbol{y})}{\partial \lambda_j} + \mathrm{D}\left\{\frac{\partial \ell_{\mathrm{M}}(\boldsymbol{\lambda}^\dagger_{(j)};\boldsymbol{y})}{\partial \lambda_j}\right\} \cdot (\boldsymbol{\lambda} - \boldsymbol{\lambda}^*), \quad j = 1, \ldots, q,$$

where $\mathrm{D}\{\partial \ell_{\mathrm{M}}(\boldsymbol{\lambda}^\dagger_{(j)};\boldsymbol{y})/\partial \lambda_j\} \in \mathbb{R}^q$ is a vector whose k-th element is $\partial^2 \ell_{\mathrm{M}}(\boldsymbol{\lambda}^\dagger_{(j)};\boldsymbol{y})/\partial \lambda_j \partial \lambda_k$, $\boldsymbol{\lambda}^\dagger_{(j)} = \boldsymbol{\lambda} + t_j(\boldsymbol{\lambda} - \boldsymbol{\lambda}^*)$ for $t_j \in [0,1]$, and $\cdot$ is the scalar product. If we put these equations together in matrix form we can write

$$\boldsymbol{U}_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y}) - \boldsymbol{U}_{\mathrm{M}}(\boldsymbol{\lambda}^*;\boldsymbol{y}) = \boldsymbol{J}(\boldsymbol{\lambda}^\dagger_{(1)}, \ldots, \boldsymbol{\lambda}^\dagger_{(q)})(\boldsymbol{\lambda} - \boldsymbol{\lambda}^*),$$

where $\boldsymbol{U}_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y}) \in \mathbb{R}^q$ is the vector whose j-th component is $\partial \ell_{\mathrm{M}}(\boldsymbol{\lambda};\boldsymbol{y})/\partial \lambda_j$, and $\boldsymbol{J}(\boldsymbol{\lambda}^\dagger_{(1)}, \ldots, \boldsymbol{\lambda}^\dagger_{(q)}) \in \mathbb{R}^{q \times q}$ is a matrix whose j-th row contains the vector $\mathrm{D}\{\partial \ell_{\mathrm{M}}(\boldsymbol{\lambda}^\dagger_{(j)};\boldsymbol{y})/\partial \lambda_j\}$ and is therefore of order $O(n)$ since each element is a sum of n log-likelihood derivatives.

The maximum likelihood estimate $\hat{\boldsymbol{\lambda}}$ and its counterpart $\hat{\boldsymbol{\lambda}}^*$ based on the Laplace approximation satisfy

$$\boldsymbol{U}_{\mathrm{M}}(\hat{\boldsymbol{\lambda}};\boldsymbol{y}) = \boldsymbol{0}_q, \quad \boldsymbol{U}^*_{\mathrm{M}}(\hat{\boldsymbol{\lambda}}^*;\boldsymbol{y}) = \boldsymbol{0}_q.$$

On writing $\dot{\boldsymbol{A}}(\boldsymbol{\lambda};\boldsymbol{y}) \in \mathbb{R}^q$ for the vector of components $\partial A(\boldsymbol{\lambda};\boldsymbol{y})/\partial \lambda_j$, and denoting the vector of error terms of order $O(n^{-a-1})$ by $\boldsymbol{O}(n^{-a-1}) \in \mathbb{R}^q$, we deduce from (27) that

$$\boldsymbol{0}_q = \boldsymbol{U}^*_{\mathrm{M}}(\hat{\boldsymbol{\lambda}}^*;\boldsymbol{y}) = \boldsymbol{U}_{\mathrm{M}}(\hat{\boldsymbol{\lambda}}^*;\boldsymbol{y}) - \dot{\boldsymbol{A}}(\hat{\boldsymbol{\lambda}}^*;\boldsymbol{y})/n^a + \boldsymbol{O}(n^{-a-1}),$$

which yields

$$\begin{aligned} \boldsymbol{J}(\hat{\boldsymbol{\lambda}}^\dagger_{(1)}, \ldots, \hat{\boldsymbol{\lambda}}^\dagger_{(q)})\,(\hat{\boldsymbol{\lambda}} - \hat{\boldsymbol{\lambda}}^*) &= \boldsymbol{U}_{\mathrm{M}}(\hat{\boldsymbol{\lambda}};\boldsymbol{y}) - \boldsymbol{U}_{\mathrm{M}}(\hat{\boldsymbol{\lambda}}^*;\boldsymbol{y}) \\ &= -\dot{\boldsymbol{A}}(\hat{\boldsymbol{\lambda}}^*;\boldsymbol{y})/n^a + \boldsymbol{O}(n^{-a-1}). \end{aligned}$$

Since $\boldsymbol{J}(\hat{\boldsymbol{\lambda}}^\dagger_{(1)}, \ldots, \hat{\boldsymbol{\lambda}}^\dagger_{(q)})$ is $O(n)$ and $\dot{\boldsymbol{A}}(\boldsymbol{\lambda};\boldsymbol{y})$ is $O(1)$, we must have $\hat{\boldsymbol{\lambda}} - \hat{\boldsymbol{\lambda}}^* = \boldsymbol{O}(n^{-a-1})$.

In a conventional Laplace approximation, we would have $a = 1$, and then $\hat{\boldsymbol{\lambda}} - \hat{\boldsymbol{\lambda}}^* = \boldsymbol{O}(n^{-2})$, but with the double Laplace approximation used in the proposed method we have $a = 2$, so $\hat{\boldsymbol{\lambda}} - \hat{\boldsymbol{\lambda}}^* = \boldsymbol{O}(n^{-3})$. Thus the maximizers of the true log-marginal likelihood and its Laplace approximation differ by much less than the 'statistical variation' of $\hat{\boldsymbol{\lambda}}$, which is $O_p(n^{-1/2})$ away from the 'true' $\boldsymbol{\lambda}$. Hence the difference $\hat{\boldsymbol{\lambda}} - \hat{\boldsymbol{\lambda}}^*$ is statistically negligible, and the same will apply to the corresponding values of $\hat{\boldsymbol{\beta}}$, which are smooth functions of $\boldsymbol{\lambda}$.

References

A. Ba, M. Sinn, Y. Goude, and P. Pompey. Adaptive learning of smoothing functions: application to electricity load forecasting. In *Advances in Neural Information Processing Systems 25*, pages 2510–2518. USA, 2012.

D. Bates and D. Eddelbuettel. Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software*, 52(5):1–24, 2013.

L. Breiman and J. H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598, 1985.

J. C. Burkner. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28, 2017.

B. Carpenter, D. Lee, M. A. Brubaker, A. Riddell, A. Gelman, B. Goodrich, J. Guo, M. Hoffman, M. Betancourt, and P. Li. Stan: a probabilistic programming language, 2017.

V. Chavez-Demoulin and A. C. Davison. Generalized additive modelling of sample extremes. *Journal of the Royal Statistical Society, Series C*, 54(1):207–222, 2005.

V. Chavez-Demoulin and A. C. Davison. Modelling time series extremes. *Revstat-Statistical Journal*, 10:109–133, 2012.

W. S. Cleveland, E. Grosse, and W. M. Shyu. *Local Regression Models.* Chapman & Hall, New York, 1993.

T. J. Cole and P. J. Green. Smoothing reference centile curves: the LMS method and penalized likelihood. *Statistics in Medicine*, 11(10):1305–1319, 1992.

A. C. Davison and N. I. Ramesh. Local likelihood smoothing of sample extremes. *Journal of the Royal Statistical Society, Series B*, 62:191–208, 2000.

L. de Haan and A. Ferreira. *Extreme Value Theory.* Springer-Verlag New York, 2006.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen. Additive Gaussian processes. In *Advances in Neural Information Processing Systems 24*, pages 226–234. USA, 2011.

A. C. Faul and M. E. Tipping. Analysis of sparse Bayesian learning. In *Advances in Neural Information Processing Systems 14*, pages 383–389. Cambridge, USA, 2001.

R. A. Fisher and L. H. C. Tippett. Limiting forms of the frequency distributions of the largest or smallest member of a sample. *Proceedings of the Cambridge Philosophical Society*, 24:180–190, 1928.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 4th edition, 2013.

C. Gu. Cross-validating non-Gaussian data. *Journal of Computational and Graphical Statistics*, 1(2):169–179, 1992.

G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2018.

T. J. Hastie and R. J. Tibshirani. Generalized additive models (with discussion). *Statistical Science*, 1:297–310, 1986.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman & Hall, 1990.

T. J. Hastie, R. J. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2nd edition, 2009.

A. F. Jenkinson. The frequency distribution of the annual maximum (or minimum) values of meteorological elements. *Journal of the Royal Meteorological Society*, 81:158–171, 1955.

G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2): 495–502, 1970.

D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068, 1999.

A. McHutchon and C. E. Rasmussen. Gaussian process training with input noise. In *Advances in Neural Information Processing Systems 24*, pages 1341–1349. USA, 2011.

G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2nd edition, 2008.

M. Mutny and A. Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features. In *Advances in Neural Information Processing Systems 31*, pages 9005–9016. USA, 2018.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996.

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135(3):370–384, 1972.

D. Nychka. Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association*, 83(404):1134–1143, 1988.

D. Oakes. Direct calculation of the information matrix via the EM. *Journal of the Royal Statistical Society, Series B*, 61(2):479–482, 1999.

F. O'Sullivan, B. S. Yandell, and W. J. Raynor. Automatic smoothing of regression functions in generalized linear models. *Journal of the American Statistical Association*, 81(393): 96–103, 1986.

Y. Qi, T. P. Minka, R. W. Picard, and Z. Ghahramani. Predictive automatic relevance determination by expectation propagation. In *International Conference on Machine Learning*, page 85, New York, USA, 2004.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

P. T. Reiss and R. T. Ogden. Smoothing parameter selection for a class of semiparametric linear models. *Journal of the Royal Statistical Society, Series B*, 71(2):505–523, 2009.

R. A. Rigby and D. M. Stasinopoulos. A semi-parametric additive model for variance heterogeneity. *Statistics and Computing*, 6(1):57–65, 1996.

R. A. Rigby and D. M. Stasinopoulos. Generalized additive models for location, scale and shape (with discussion). *Journal of the Royal Statistical Society, Series C*, 54(3):507–554, 2005.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society, Series B*, 71(2):319–392, 2009.

B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society, Series B*, 47(1):1–52, 1985.

B. M. Steele. A modified EM algorithm for estimation in generalized mixed models. *Biometrics*, 52(4):1295–1310, 1996.

L. Tierney, R. E. Kass, and J. B. Kadane. Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association*, 84(407):710–716, 1989.

M. E. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems 12*, pages 652–658, Cambridge, USA, 1999.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

M. Tsang, H. Liu, S. Purushotham, P. Murali, and Y. Liu. Neural interaction transparency: disentangling learned interactions for improved interpretability. In *Advances in Neural Information Processing Systems 31*, pages 5804–5813. USA, 2018.

E. F. Vonesh, H. Wang, L. Nie, and D. Majumdar. Conditional second-order generalized estimating equations for generalized linear and nonlinear mixed-effects models. *Journal of the American Statistical Association*, 97(457):271–283, 2002.

S. N. Wood. Thin plate regression splines. *Journal of the Royal Statistical Society, Series B*, 65(1):95–114, 2003.

S. N. Wood. Fast stable direct fitting and smoothness selection for generalized additive models. *Journal of the Royal Statistical Society, Series B*, 70(3):495–518, 2008.

S. N. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society, Series B*, 73(1):3–36, 2011.

S. N. Wood, Y. Goude, and S. Shaw. Generalized additive models for large data sets. *Journal of the Royal Statistical Society, Series C*, 64(1):139–155, 2015.

S. N. Wood, N. Pya, and B. Safken. Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association*, 111(516):1548–1563, 2016.

S. N. Wood, Z. Li, G. Shaddick, and N. H. Augustin. Generalized additive models for gigadata: modeling the UK black smoke network daily data. *Journal of the American Statistical Association*, 112(519):1199–1210, 2017.

T. W. Yee and C. J. Wild. Vector generalized additive models. *Journal of the Royal Statistical Society, Series B*, 58(3):481–493, 1996.

Journal of Machine Learning Research 20 (2019) 1-42 Submitted 10/18; Revised 11/19; Published 11/19

Learning Overcomplete, Low Coherence Dictionaries with Linear Inference

Jesse A. Livezey JLIVEZEY@LBL.GOV
Biological Systems and Engineering Division
Lawrence Berkeley National Laboratory
Berkeley, California 94720, USA
Redwood Center for Theoretical Neuroscience
University of California, Berkeley
Berkeley, California 94720, USA

Alejandro F. Bujan AFBUJAN@GMAIL.COM
Friedrich T. Sommer FSOMMER@BERKELEY.EDU
Redwood Center for Theoretical Neuroscience
University of California, Berkeley
Berkeley, California 94720, USA

Editor: Aapo Hyvärinen

Abstract

Finding overcomplete latent representations of data has applications in data analysis, signal processing, machine learning, theoretical neuroscience and many other fields. In an overcomplete representation, the number of latent features exceeds the data dimensionality, which is useful when the data is undersampled by the measurements (compressed sensing or information bottlenecks in neural systems) or composed from multiple complete sets of linear features, each spanning the data space. Independent Components Analysis (ICA) is a linear technique for learning sparse latent representations, which typically has a lower computational cost than sparse coding, a linear generative model which requires an iterative, nonlinear inference step. While well suited for finding complete representations, we show that overcompleteness poses a challenge to existing ICA algorithms. Specifically, the coherence control used in existing ICA and other dictionary learning algorithms, necessary to prevent the formation of duplicate dictionary features, is ill-suited in the overcomplete case. We show that in the overcomplete case, several existing ICA algorithms have undesirable global minima that maximize coherence. We provide a theoretical explanation of these failures and, based on the theory, propose improved coherence control costs for overcomplete ICA algorithms. Further, by comparing ICA algorithms to the computationally more expensive sparse coding on synthetic data, we show that the limited applicability of overcomplete, linear inference can be extended with the proposed cost functions. Finally, when trained on natural images, we show that the coherence control biases the exploration of the data manifold, sometimes yielding suboptimal, coherent solutions. All told, this study contributes new insights into and methods for coherence control for linear ICA, some of which are applicable to many other nonlinear models.

Keywords: independent components analysis, dictionary learning, coherence

©2019 Jesse A. Livezey, Alejandro F. Bujan, Friedrich T. Sommer.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-703.html.

1. Introduction

Mining the statistical structure of data is a central topic of machine learning and is also a principle for computational models in neuroscience. A prominent class of such algorithms is dictionary learning, which reveal a set of structural primitives in the data, the dictionary, and a corresponding latent representation, often regularized by sparsity. In this work, we focus on overcomplete dictionary learning (Olshausen and Field, 1997; Hyvärinen, 2005; Le et al., 2011), the case when the dimension of the latent representation exceeds the dimension of the data and therefore the linear filters (dictionary) generating the data cannot all be mutually orthogonal.

Independent Components Analysis (ICA) (Comon, 1994; Bell and Sejnowski, 1997) is a technique for learning the underlying non-Gaussian and independent sources, S, in a dataset, X. ICA is commonly used in the complete, noiseless case, although methods which can be run on noisy data exist Hyvarinen (1999). When run as a complete, noiseless model, ICA is computationally light-weight because the learned mappings between data and sources are linear in both directions. ICA can be formulated as a noiseless linear generative model

$$X_i = \sum_{j=1}^{L} A_{ij} S_j, \tag{1}$$

where $A \in \mathbb{R}^{D \times L}$ is referred to as the *mixing matrix* wherein D is the dimensionality of the data, X, and L is the dimensionality of the sources, S. In the complete case ($D = L$), the goal of ICA is to find the *unmixing matrix* $W \in \mathbb{R}^{L \times D}$ such that the sources for all M data samples can be recovered, $S_j^{(i)} = \sum_k W_{jk} X_k$ with $W = A^{-1}$. The unmixing matrix W can then be obtained by minimizing the negative log-likelihood of the model

$$-\log P(X; W) = \sum_{i=1}^{M} \sum_{j=1}^{L} g(\sum_k W_{jk} X_k^{(i)}) - M \log(\det(W)) \tag{2}$$

where $g(\cdot)$ specifies the shape of the negative log-prior of the latent variables S and is usually a smooth version of the L_1 norm such as the $\log(\cosh(\cdot))$, which encourages the projections of X to be sparse, $X^{(i)}$ is the ith element of the dataset, X, which has M samples, and where the bases are constrained to have unit-norm. The log-determinant comes from the multivariate change of variables in the likelihood from X to S

$$P(X) = P(S) |\det \frac{dS}{dX}| = P(W \cdot X) |\det W|. \tag{3}$$

If the data has been whitened, the unconstrained optimization (Eq 2) can be replaced by a constrained optimization where the second term in the cost function is replaced with the constraint $WW^T = I$ (Hyvärinen and Oja, 1997).

In complete ICA, the log-determinant (or the identity constraint) will prevent multiple elements of the dictionary, W, from learning the same feature. In overcomplete ICA, the linear generative model (Eq 1) cannot be inverted, and therefore, overcomplete versions of Eqs 2 and 3 cannot be derived. One alternative to maximum likelihood learning is to create

an objective function by adding a new cost, $C(W)$, to the sparsity prior (Hyvärinen and Inki, 2002; Le et al., 2011). The new unconstrained objective function becomes

$$\text{Objective}(W) = \lambda \sum_{i=1}^{M} \sum_{j=1}^{L} g(\sum_{k} W_{jk} X_k^{(i)}) + C(W). \tag{4}$$

This form is also similar to the log-posterior of the sources, S, which appears in sparse coding models and is used in maximum a posteriori inference, although here, the cost is used to optimize the dictionary, W, not estimate sources. Overcomplete ICA models of this form are one case of analysis dictionary learning methods, where the projections of the data are assumed to have sparse structure rather than assuming a sparse linear generative (synthesis) model. In complete ICA methods, there is no distinction between the synthesis and the analysis view (Bell and Sejnowski, 1997; Hyvärinen et al., 2001; Elad et al., 2007; Teh et al., 2003; Ophir et al., 2011; Rubinstein et al., 2013; Chun and Fessler, 2018)

$$\begin{aligned} \text{Synthesis: } X = AS,\ S \text{ sparse} \\ \text{Analysis: } P = WX,\ P \text{ sparse.} \end{aligned} \tag{5}$$

In overcomplete ICA, these two formulations are no longer equivalent to each other.

The cost, $C(W)$, should be chosen to exert coherence control on the dictionary, that is, to prevent the co-alignment of the bases. The coherence of a dictionary is defined as the maximum absolute value of the off-diagonal elements of the Gram matrix of a unit-normalized dictionary (Davenport et al., 2011), W,

$$\text{coherence}(W) \equiv \max_{i \neq j} |\sum_{k} W_{ik} W_{jk}| = \max_{i \neq j} |\cos \theta_{ij}| \tag{6}$$

where $\sum_k W_{ik} W_{jk} = \cos \theta_{ij}$ is the cosine similarity between the unit normalized dictionary elements W_i and W_j. A dictionary with high coherence (near 1) will have duplicated or nearly duplicated bases.

A number of methods for coherence control in complete and overcomplete methods have been proposed including a quasi-orthogonality constraint (Hyvärinen et al., 1999), a reconstruction cost (Le et al., 2011) which is equivalent to the L_2 coherence cost in Eq 8 (Ramırez et al., 2009; Sigg et al., 2012; Bao et al., 2014; Chun and Fessler, 2018; Bansal et al., 2018), and a Random Prior cost (Hyvärinen and Inki, 2002) (see Section 3 for details). However, a systematic analysis of the properties of proposed coherence control methods and a comparison with methods that extend more naturally to overcomplete representations, for example, sparse coding, is still missing in the literature. In particular, the L_2 cost is often claimed to promote diversity or incoherence in overcomplete dictionaries elements, which we will show is not the case.

Our first theoretical result is that although the global minima of the L_2 cost have minimal coherence (coherence = 0) for a complete basis, in the overcomplete case, it has global minima with maximal coherence (coherence = 1). We introduce an analytic framework for evaluating different coherence control costs, and propose several new costs, which fix deficiencies in previous methods. Our first novel cost is the L_4 cost on the difference between the identity matrix and the Gram matrix of the bases. The second is a cost which we call

the *Coulomb* cost because it is derived from the potential energy of a collection of charged particles bound to the surface of an n-sphere. We also propose modifications to previously proposed methods of coherence control which we show allows them to learn less coherent dictionaries.

In addition to controlling coherence, we show empirically that these costs will influence the entire distribution of the learned bases in an overcomplete dictionary. We investigate the impact of different coherence control costs on recovering overcomplete synthesis and analysis models. Finally, we evaluate the coherence and diversity of bases learned on a dataset of natural image patches.

1.1. Related work

Studying methods for learning overcomplete dictionaries is motivated from applications in data analysis and theoretical neuroscience. In data analysis, overcomplete dictionaries become essential if data are either undersampled (Hillar and Sommer, 2015), or have a sparse structure with respect to a combination of orthobases (Donoho and Elad, 2003). In neuroscience, dictionary learning has not only been proposed for data analysis (Delorme et al., 2007; Agarwal et al., 2014; Hirayama et al., 2015), but also as a computational model for understanding the formation of sensory representations (Bell and Sejnowski, 1997; Olshausen and Field, 1996; Klein et al., 2003; Smith and Lewicki, 2006; Rehn and Sommer, 2007; Zylberberg et al., 2011; Carlson et al., 2012). It has been estimated from anatomical data that in primary sensory areas the number of neurons by far exceeds the number of afferent inputs (Barlow, 1981; Spoendlin and Schrott, 1989; Curcio and Allen, 1990; Leuba and Kraftsik, 1994; Northern and Downs, 2002; DeWeese et al., 2005). Further, it has been shown that dictionary learning forms more diverse sets of features when overcomplete, which more closely matches the diversity of receptive fields found in sensory cortex (Rehn and Sommer, 2007; Carlson et al., 2012; Olshausen, 2013).

Sparse coding is a linear generative model for dictionary learning, which unlike typical ICA models, also includes an additive noise term to the mixtures

$$X_i = \sum_{j=1}^{L} A_{ij} S_j + \epsilon_i, \tag{7}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma)$. Sparse coding requires an iterative, computationally complex maximum a posteriori estimation, posterior estimation step, or an approximation to these (Olshausen and Field, 1996; Lewicki and Sejnowski, 2000; Rehn and Sommer, 2007; Rozell et al., 2008; Gregor and LeCun, 2010; Hu et al., 2014). However, unlike ICA, sparse coding extends naturally to the overcomplete setting without modification. During inference, latent features in overcomplete sparse coding models (Lewicki and Olshausen, 1999) have an explaining-away effect on each other which discourages them from learning coherent solutions. Methods for incoherent overcomplete dictionary learning which add additional coherence costs, including the L_2 cost, with nonlinear inference have also been proposed (Ramırez et al., 2009; Sigg et al., 2012; Mailhé et al., 2012; Bao et al., 2014). Score matching (Hyvärinen, 2005) is another alternative to maximum likelihood learning which can be used for overcomplete ICA models.

In overcomplete dictionary learning, a distinction is made between synthesis and analysis methods. Synthesis methods posit that data is formed from a linear combination (dictionary) of sparse sources, and methods are designed to both recover the dictionary and the sources (Olshausen and Field, 1997; Chen et al., 2001; Davenport et al., 2011). Analysis dictionary learning assume that the atoms which combine linearly to makeup a signal have sparse projections from some analysis matrix (Ophir et al., 2011; Rubinstein et al., 2013; Chun and Fessler, 2018). In the overcomplete case, these two problems will deviate (Elad et al., 2007). Methods have been proposed for inference in analysis models (Elad et al., 2007) as well as analysis dictionary learning (Ophir et al., 2011; Rubinstein et al., 2013; Chun and Fessler, 2018).

2. Results

In this section we first prove that the L_2 cost has global minima with coherence $= 1$. We then propose new coherence control costs and evaluate them on synthetic datasets and natural images.

2.1. The L_2 cost has high coherence global minima

Dictionary or representation learning methods often augment their cost functions with additional terms aimed at learning less coherent features (Ramırez et al., 2009; Le et al., 2011; Sigg et al., 2012; Bao et al., 2014; Chun and Fessler, 2018; Bansal et al., 2018) or making learning through optimization more efficient (Howard et al., 2008). The L_2 cost, defined for a unmixing matrix, W, as

$$C_{L_2}(W) = \frac{1}{2}\sum_{ij}(\delta_{ij} - \sum_k W_{ik}W_{jk})^2 = \frac{1}{2}\sum_{ij}(\delta_{ij} - \cos\theta_{ij})^2, \tag{8}$$

has been used to augment dictionary learning methods motivated by the desire to learn more incoherent or diverse dictionaries (Strohmer and Heath Jr, 2003; Davenport et al., 2011). However, we show that minimizing the L_2 cost is a necessary but not sufficient condition for finding *equiangular tight frames* (see Section 3.1.1 for details and definitions), a certain class of minimum coherence solutions. Moreover, we prove that the L_2 cost has global minima with maximum coherence. This shows that the L_2 cost and its related costs are not providing coherence control in overcomplete dictionaries.

For the L_2 cost, it can be shown that for integer overcompleteness, there exists a set of global minima in which the angle between many pairs of bases is exactly zero and the coherence is 1, the maximum attainable value. We prove the following theorem:

Theorem 1 *Let $W_0 \in \mathbb{R}^{L\times D}$ be an overcomplete unmixing matrix with data dimension D and latent dimension $L = n \times D$, with $n > 1$, $\in \mathbb{Z}$ and unit-norm rows. There exist dictionaries, W_0, that are global minima of the L_2 cost with coherence $= 1$.*

This shows that the L_2 cost has global minima that have the exact property it was proposed to prevent (high coherence). The proof of this theorem also shows that, in the complete case ($n = 1$), an orthonormal basis is a global minimum of the L_2 cost. We also prove that there are operators which transform the pathological solution (coherence $= 1$) into non-pathological solutions (coherence < 1) to which the L_2 cost is invariant:

Theorem 2 *Let P be a projection operator from the L dimensional space of dictionary elements to a D dimensional subspace and P^C its complement projection. Φ is constructed as the sum of any rotation, $R \in O(L)$, projected within the D-dimensional subspace of the dictionary elements and an identity applied to the complement subspace, $\Phi = PR + P^C$. There exist non-trivial continuous transformations: Φ, on W_0 to which the L_2 cost is invariant. These transformed dictionaries, $W_0\Phi$, have coherence ≤ 1 for non-identity rotations and are global minima of the L_2 cost.*

Appendices B.1 and B.4 contain the proofs of these theorems.

These high coherence global minima are illustrated with a two dimensional, two times overcomplete example in Fig 1. It can be shown that there are pathological (high coherence) minima (Fig 1A) which can be continuously rotated into other low coherence minima (Fig 1B). These configurations are equivalent in terms of the value of the L_2 cost and lie on a connected global minimum. These families of configurations are minima if it can be shown that the gradient of the cost is zero, that is, they are critical points of the cost, and that the Hessian is positive definite in all directions but the one that rotates the configuration within the family of solutions. We will show these two things through an explicit derivation in the 2 dimensional case.

In order to understand these minima, we evaluate the L_2 cost in a two dimensional example analytically. The global rotational symmetry of the L_2 cost allows us to parameterize all solutions with respect to one fixed dictionary element: $(1, 0)$, without loss of generality. The four dictionary elements, shown in Fig 1, are

$$(1, 0), (\cos\theta_1, \sin\theta_1), (\cos\theta_2, \sin\theta_2), (\cos\theta_2 + \theta_3, \sin\theta_2 + \theta_3). \tag{9}$$

Setting θ_1 and θ_3 to $\pi/2$, that is, creating two sets of orthonormal bases, forms a ring of minima as θ_2 is varied. This can be shown by computing the gradient and the eigenvalues of the Hessian of the L_2 cost at these points. The cost function, gradient, and Hessian are tabulated in Appendix A and the eigenvalues are plotted individually in Fig 1.

The value of the L_2 cost is constant as a function of θ_2 (Fig 1C, purple, dashed line) even though the coherence is drastically changing as a function of θ_2. The Hessian of the L_2 cost along this path has one eigenvalue that is 0 as a function of θ_2 whose eigenvector corresponds to changing θ_2 with fixed θ_1 and θ_3 (Fig 1D, see Appendix A for the exact functional form). The other two eigenvalues are greater than zero and greater then zero for $\theta_2 \neq 0$ respectively which shows that the cost is a minimum almost everywhere along this path. At $\theta_2 = 0$, the second eigenvalue becomes 0. This eigenvalue has eigenvector $(-1, 0, 1)$. If we evaluate the cost along this direction centered at $\theta_1 = \theta_2 = \theta_3 = 0$, we find that although the second derivative is zero, the fourth derivative is positive showing that indeed, this point is a minimum (see Appendix A.2 for a derivation).

In many previous studies, the L_2 cost or variations of it were proposed in order to learn dictionaries with lower coherence (Ramırez et al., 2009; Le et al., 2011; Sigg et al., 2012; Bao et al., 2014; Chun and Fessler, 2018; Bansal et al., 2018). The results in this section show that the L_2 cost function does not provide coherence control in the overcomplete regime. In fact, dictionaries that should be maxima are part of a set of global minima. This indicates that there is a need for new forms of coherence control.

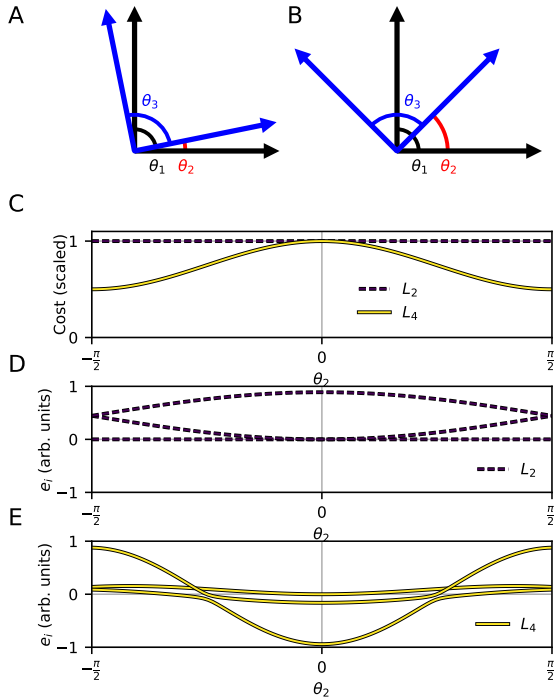


Figure 1: Structure of the pathological global minimum in the L_2 cost which the L_4 cost corrects. In **A** and **B**, each arrow represents a dictionary element in a 2-times overcomplete dictionary in a 2-dimensional space. **A** A dictionary with high coherence which has the same value of the cost as the dictionary in **B** for any θ_2 including the pathological solution $\theta_2 \to 0$. **B** A dictionary with low coherence. **C** The L_2 and L_4 costs are plotted at $\theta_1 = \theta_3 = \pi/2$ as a function of θ_2. The costs have been scaled so that their maximum value is 1. **D** The eigenvalues of the L_2 cost at $\theta_1 = \theta_3 = \pi/2$ as a function of θ_2 scaled between -1 and 1. **E** The eigenvalues of the L_4 cost at $\theta_1 = \theta_3 = \pi/2$ as a function of θ_2 scaled between -1 and 1.

2.2. Addressing high coherence solutions: L_4 and Coulomb costs

The rotational symmetry in the L_2 cost leads to its pathological (high coherence) global minima, and this insight motivates a simple modification which will not have high coherence minima. We propose a novel coherence control cost termed the L_4 cost, which removes the pathological minima of the L_2 cost. The motivation for this cost function is to more strongly penalize large inner products in the gram matrix. The L_4 cost function also acts on the gram matrix of W, but raises each off diagonal element to the fourth power which breaks the rotational symmetries which lead to the pathological minima

$$C_{L_4}(W) = \frac{1}{4}\sum_{ij}(\delta_{ij} - \sum_k W_{ik}W_{jk})^4 = \frac{1}{4}\sum_{ij}(\delta_{ij} - \cos\theta_{ij})^4. \tag{10}$$

Following the same analysis as in Section 2.1, we show that the pathological solutions are either reduced to saddle points at $\theta_2 = n\frac{\pi}{2}$ or local minima at $\theta_2 = (2n+1)\frac{\pi}{4}$, which correspond to incoherent solutions (Fig 1E). The L_4 cost as a function of θ_2 has a maximum at $\theta_2 = 0$ (coherent solutions) and minima at $\theta_2 = \frac{\pi}{2}$ (Fig 1C). The L_4 cost function, gradient, and Hessian are tabulated in Appendix A for this 2 dimensional example.

We also propose a second alternative cost, where the repulsion from high coherence is *Coulombic*: the Coulomb cost. Coherence control can then be related to the problem of characterizing the minimum potential energy states of L charged particles on an n-sphere, an open problem in electrostatics (Smale, 1998). The energy, E^{Coulomb}, of two charged point particles of the the same sign is proportional to the inverse of their distance, $\vec{r}_{ij}$

$$E_{ij}^{\text{Coulomb}} \propto \frac{1}{|\vec{r}_{ij}|}. \tag{11}$$

When constrained to the surface of the unit-radius n-sphere, the distance between two points can be written as a function of only the angle between the two points $|r_{ij}| = \sqrt{1-\cos^2(\theta_{ij}/2)}$. In the case of same-sign charged particles, the minimum energy is when the particles are at antipodal points. However, in ICA, there is no distinction between a dictionary element and its negative (the antipodal point). Instead, the minimum energy configuration should be when two elements are perpendicular. To map this problem onto ICA, the cost should be made symmetric around $\theta = \pi/2$ rather than $\theta = \pi$ since a dictionary element and its negative should have maximal pairwise energy, not minimal. This can be accomplished by replacing θ with 2θ, that is, $\sqrt{1-\cos^2(\theta_{ij}/2)} \to \sqrt{1-\cos^2\theta_{ij}}$. Therefore, the Coulomb cost can be formulated as

$$C_{\text{Coulomb}}(W) = \sum_{i\neq j}\frac{1}{\sqrt{1-\cos^2\theta_{ij}}} = \sum_{i\neq j}\frac{1}{\sqrt{1-(\sum_k W_{ik}W_{jk})^2}}. \tag{12}$$

In practice, we subtract the value of the cost for perpendicular bases, 1, for each pair $i \neq j$ to bring the cost into a better dynamic range. This cost diverges as coherence $\to 1$, which means it cannot have high coherence minima.

2.3. Numerical investigations of coherence control

The above analysis provides evidence of a failure of the L_2 cost to provide coherence control. The alternative coherence cost function can prevent high coherence solutions, but all costs

functions will act on the entire distribution of dictionary elements, not only the high coherence pairs. Deriving the distribution of pairwise angles in the minima of the cost functions is analytically difficult. However, understanding the influence of the coherence control cost function on the distribution of dictionary elements allows us to better understand their biases.

In order to understand the origin of the effects of the different coherence controls on the pairwise angle distributions, the coherence costs can be directly compared without the data dependent ICA sparsity prior. We use two different initializations of the bases and optimize the data-independent coherence costs. These initializations are: a noisy pathological initialization (as in Section 2.1) and a random uniform initialization on the surface of a n-sphere (Gaussian distributed entries normalized to unit-norm elements). We will numerically explore the minima of these cost function for a 2 times overcomplete dictionary in a 32 dimensional data space by minimizing the cost function with these two initializations.

The noisy pathological initialization tiles an orthonormal, complete basis two times and adds a relatively small ($\sigma = .01$) amount of zero-mean Gaussian noise to every basis element to create W. As shown by the red-dashed histogram in Fig 2A, most pairwise angles start close to either 0 or $\frac{\pi}{2}$ as shown in the two peaks in the initial distribution. Minimizing the L_2 cost (purple line) from this initialization gives a final solutions with high coherence, similar to the initial distribution. The other costs push the pairs of bases with initially small pairwise angles apart. This shows numerically that the L_2 does not provide coherence control for overcomplete dictionaries unlike other proposed methods. Appendix Fig C1 contains the same analysis for the full set of cost functions, and Appendix Fig C2 contains a comparison across powers from 1 to 6. Although the L_4 cost may have saddle points in the cost landscape (see Section 2.2), in practice they do not seem to be a problem for optimization (see Appendix Fig C3).

In the random uniform case, the elements of W are drawn independently from a uniform distribution on the unit n-sphere. The final distribution of pairwise angles for the L_2 cost peaks at $\frac{\pi}{2}$ but also has a longer tail towards small pairwise angles. The other costs have shorter tails and have varying amounts of density near $\frac{\pi}{2}$. Of all costs, the L_4 cost distributes the angles most evenly which is reflected by its distribution having the narrowest width and lowest coherence.

Together, these results show that the L_2 cost does not provide coherence control and is also sensitive to the initialization method. The proposed L_4 and Coulomb cost, as well as the previously proposed Random Prior (see Section 3), all provide coherence control. For these three costs, the distribution from which the dictionary was initialized does not have a large effect on the distributions at the numerical minima. These traits mean that they are better suited for providing coherence control in overcomplete dictionary learning methods.

2.4. Flattened costs

The previous analysis provides insight into why different cost function have different behavior for small angles (high coherence). However, the L_4, Coulomb, and Random Prior cost also show qualitatively different behavior in their distributions near $\frac{\pi}{2}$. Both the Coulomb and Random Prior have density near $\frac{\pi}{2}$ for the distribution of pairwise angles, meaning that a fraction of the bases are nearly orthogonal. The L_4 has much lower density near $\frac{\pi}{2}$,

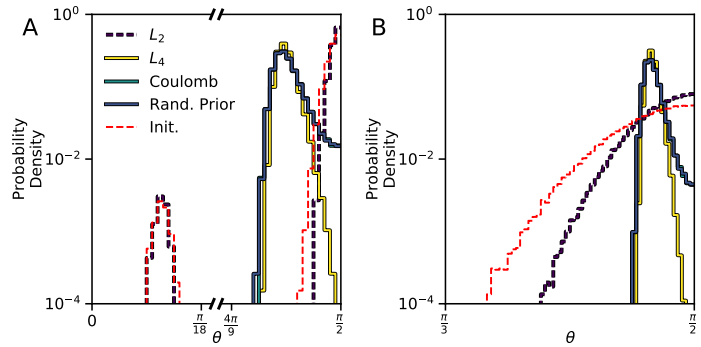


Figure 2: Coherence control costs have minima with varying coherence which can depend on initialization. Color legend is preserved across panels. For both panels a 2 times overcomplete dictionary with a data dimension of 32 was used and the distributions are averaged across 10 random initializations. **A** Distribution of pairwise angles (log scale) obtained by numerically minimizing a subset of the coherence cost functions for the pathological dictionary initialization. Red dotted line indicates the initial distribution of pairwise angles. Note that the horizontal axis is broken. **B** Angle distributions obtained through optimization from a uniform random dictionary initialization. Note that the horizontal axis only includes the range from $\frac{\pi}{3}$ to $\frac{\pi}{2}$. In both plots, the Coulomb and Random Prior lines are almost entirely overlapping.

and a correspondingly lower coherence (smallest pairwise angle). In order to achieve minimal coherence (equiangular tight frame) for an overcomplete dictionary, the distribution of pairwise angles should form a delta-function away from $\frac{\pi}{2}$. Therefore high density near $\frac{\pi}{2}$ may not be desirable for learning low coherence solutions.

In order to gain more insight into the causes of the qualitative differences in the distributions of angles, we analyze the behavior of the costs around $\theta = 0$ and $\theta = \frac{\pi}{2}$ (Fig 3A, B respectively). The gradient of the cost close to $|\cos\theta| = 1$ is proportional to the force the angles feel to stay away from zero which will influence the high coherence tail of the distribution. Taylor expanding all the costs near $\cos\theta = 0$ reveals that all cost functions have non-zero second order terms except for the L_4 cost which only has a fourth order term with linear and cubic terms in their gradients respectively as shown in Fig 3A. Cost with gradients that have lower-order Taylor expansions near $\cos\theta = 0$ encourage pairs of basis vectors to be more orthogonal at the expense of higher coherence. This may lead to distributions of pairwise angles which are less uniform over all pairs of elements of the dictionary. Since the L_4 cost only has higher order derivatives near $\cos\theta = 0$, there is no pressure to form exactly perpendicular pairs. This can also be seen in an L_1 cost which encourages many pair to be almost exactly perpendicular at the expense of many pairs with maximum coherence (see Appendix Fig C2).

We hypothesize that the quadratic terms are creating higher coherence minima with more pairwise angles close to $\frac{\pi}{2}$. This is additional motivation for the L_4 cost and leads us to propose modified versions of the Coulomb and Random Prior costs where the quadratic terms have been removed. The Random Prior cost (Hyvärinen and Inki, 2002) is derived from the distribution of angles expected between pairs of angles randomly drawn on the surface of an n-sphere and is described in Section 3. This can be done by subtracting the quadratic term in the Taylor series from the original cost function

$$C_{\text{Flat}}(\cos\theta_{ij}) = C(\cos\theta_{ij}) - \left.\frac{\partial^2 C(\cos\theta_{ij})}{\partial \cos\theta_{ij}^2}\right|_0 \cos^2\theta_{ij}. \tag{13}$$

This hypothesis can be validated numerically. We compared the distribution of pairwise angles when the Coulomb nad Random Prior costs were minimized with their flattened counterparts. Both the Flattened Coulomb and Random Prior costs (Fig 3C, dotted) show pairwise angle distributions which have lower coherence and fewer pairwise angles close to 90 degrees compared to the original costs (Fig 3C, solid). This shows that across costs, the quadratic terms dominate the behavior of the pairwise angle distributions near 90 degrees and can have a small effect on the coherence on the distributions.

These coherence control methods will also have different behaviors as a function of overcompleteness. To understand their behavior, we measured the coherence of their minima as a function of overcompleteness. Fig 3D shows the minimum pairwise angle (arccos of coherence, low coherence is high minimum pairwise angle) of these methods as a function of overcompleteness at fixed data dimensionality. The median over random initializations of the minimum pairwise angle between dictionary elements for numerically minimized coherence costs is shown. The cost functions evaluated here fall into three groups with quantitatively similar intra-group coherence as a function of overcompleteness. The L_2 cost has the highest coherence (smallest pairwise angle) for all overcompletenesses greater than

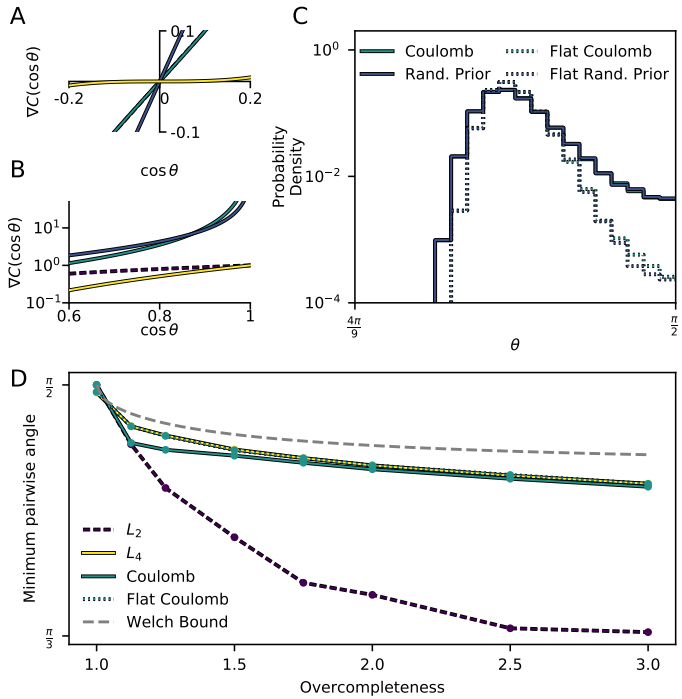


Figure 3: Quadratic terms dominate the minima of coherence control costs. **A** Gradient of the costs as a function of $\cos\theta$ near $\cos\theta = 0$. **B** Gradient of the costs as a function of $\cos\theta$ near $\cos\theta = 1$. **C** Distibution of pairwise angles for a 2 times overcomplete dictionary with a data dimension of 32 from 10 random uniform initializations. The Coulomb and Random Prior cost function distributions are shown (solid lines) along with their counterparts with quadratic terms removed ("flattened", dashed). The Coulomb and Random Prior lines are almost entirely overlapping. **D** The median minimum pairwise angle (arccosine of coherence) across 10 initializations is plotted as a function of overcompleteness for a dictionary with a data dimension of 32. The largest possible value (Welch Bound) is also shown as a function of overcompleteness. The L_4 and Flat Coulomb lines are almost entirely overlapping.

1. The L_4 cost and flattened versions of the Random Prior and Coulomb costs have the lowest coherence. The Random Prior and Coulomb costs behave similarly to the L_2 costs for low overcompleteness (less than 1.5) and then converge to be similar to the L_4 and flattened costs for high overcompletenesses (greater than 2). Fig C6 contains a detailed Coulomb and Random Prior comparison. The Welch Bound (Welch, 1974) is a lower bound for the smallest possible coherence (upper bound on the largest minimum pairwise angle) achievable (Fig 3D). The best coherence control cost functions approach, but do not saturate this bound. Note that constructing overcomplete dictionaries that saturate this bound for arbitrary overcompleteness is an open problem (Strohmer and Heath Jr, 2003; Fickus and Mixon, 2015). This shows that the quadratic terms in the cost function are dominating the coherence behavior of the cost functions and that removing the term as in the flattened costs or only including quartic terms as in the L_4 leads to lower coherence solutions.

These results show that proposed coherence control methods prevent high coherence to different degrees, and furthermore that the choice of coherence control, which is meant to affect the distribution of small pairwise angles, has an effect on the entire distribution of angles. Specifically, the L_2 cost does not provide coherence control and leads to solutions which are heavily biases by initialization unlike other proposed costs. These results also validate the relationship between second order terms in the cost function and the trade-off between coherence and orthogonality. Furthermore, since the costs were investigated without the data-dependent ICA prior, they should be useful for augmenting methods including sparse coding, deep learning, and anything that learns a overcomplete dictionary or weight matrix.

2.5. Recovery of the synthesis mixing matrix with overcomplete ICA

The previous analysis considered the data-independent coherence costs on their own. In ICA, the coherence costs will trade-off with the sparsity prior (Eq 4). Ideally, coherence costs would only prevent duplication of learned dictionary elements, but otherwise let the data shaping of the basis functions through the sparsity prior. In practice, we have shown that coherence control costs can have an effect on all dictionary elements, including those with large pairwise angles. It is not currently clear how these different costs will bias the learned dictionaries.

To investigate how the coherence control costs perform on data in overcomplete ICA, we compare different ICA cost functions and a sparse coding model on the task of recovering a known mixing matrix from k-sparse data with a Laplacian prior. We compare three classes of overcomplete dictionary recovery methods. The first is a sparse coding baseline (Olshausen and Field, 1997), the second are analysis ICA models described in Section 1 which combine the sparse prior from complete ICA and a coherence control cost, and the final is Score Matching (Hyvärinen, 2005), which is a non-maximum-likelihood method that can be used in overcomplete ICA. The data generated for this task comes from a noiseless overcomplete, sparse generative model. Sparse coding, as a dictionary recovery method, is designed to infer generative models. However, the overcomplete ICA models considered here are being fit assuming a sparse analysis model (Ophir et al., 2011; Rubinstein et al., 2013; Chun and Fessler, 2018). Therefore, these ICA models are mismatched to the underlying generative process of the data. Here, we evaluate in what regime overcomplete analysis ICA models

are able to recover a generative dictionary and what impact the coherence control cost has on recovery.

Overcomplete mixing matrices were generated from the Soft Coherence Cost (see Section 3) and used to generate a k-sparse dataset. The dictionary learning methods were then all trained on these datasets. Recovered unmixing matrices were compared to the ground-truth mixing matrix where the error for recovery is 0 for a perfect recovery ($W^T = A$) and 1 for a random recovery (see Section 3.5 for details). For a 32-dimensional data space, we vary the k-sparseness and overcompleteness of the data. For each of these datasets, where the number of dataset samples was 10-times the mixing matrix dimensionality, we fit all models to the data from 10 random initializations, for a range of sparsity weights: λ, if applicable, and then compare the recovery metric across models.

For a 12-sparse, 2-times overcomplete dataset, all methods can recover the mixing matrix well for some value of λ (Fig 4A). The L_2 and Score Matching costs perform slightly worse than the maximum-likelihood inspired ICA methods and sparse coding. All methods have a certain range of λ over which they recover the mixing matrix well and have differences in how they fail, for instance sparse coding has a very quick transition to poor recovery compared to ICA methods whose performance tends to decrease more slowly as λ moves outside of the optimal range.

At fixed k-sparsity ($k = 12$), we vary the overcompleteness and compare recovery costs (Fig 4B). As a function of overcompleteness, Score Matching recovers well in a smaller range of overcompleteness as compared to other ICA methods. Besides the L_2 cost, all other ICA methods have nearly identical recovery. The L_2 cost's performance breaks down at lower overcompleteness. All ICA methods fail to recover the mixing matrix once the overcompleteness becomes too large, while sparse coding continues to succeed in recovering the mixing matrix. Since the number of bases being recovered changes as the overcompleteness changes, it is not meaningful to compare the recovery metric between overcompletenesses, but it meaningful to compare different models at fixed overcompleteness.

At fixed overcompletenesss (OC=2), we vary the k-sparsity and compare recovery costs Fig (4C). Sparse coding performs well at all k-sparsenesses, but the ICA methods perform better with larger k-sparseness. The L_2 cost and Score Matching fails to recover well at a lower k-sparseness than other ICA methods. Since the number of bases being recovered is fixed as a function of the k-sparseness, the recovery metric can be compared across k-sparseness and models.

Fig 4D and E show the methods in a regime ($k = 6$ and 3-times overcomplete, respectively) where ICA methods do not recover the mixing matrix as well as sparse coding. Fig C4 contains the same analysis for the full set of cost functions.

In summary, we show that in general, ICA analysis methods have limited ability to recover generative dictionaries as a function of overcompleteness compared to sparse coding although the methods proposed here extend the range of applicability, which is consistent with Elad et al. (2007). Furthermore, we show that different ICA methods have different regimes of performance with Score Matching and the L_2 cost having the smallest ranges of applicability. Other ICA methods generally have similar performance. Score Matching did not always perform as well as other ICA methods as a function of overcompleteness or k-sparseness, although it is a hyperparameter-free cost (no λ hyperparameter). The more computationally costly sparse coding was able to recover the mixing matrix more

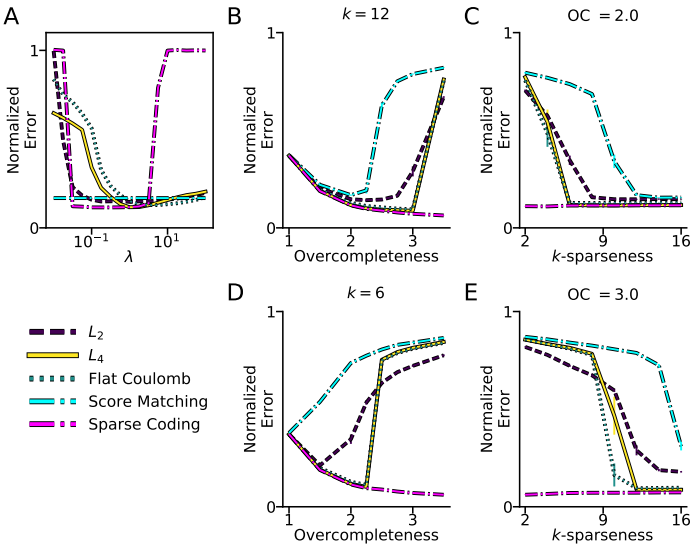


Figure 4: Coherence control costs do not all recover mixing matrices well. All ground truth mixing matrices were generated from the Soft Coherence cost and had a data dimension of 32. Color and line style legend are preserved across panels. **A** The normalized recovery error (see Section 3 for details) for a 2-times overcomplete mixing matrix and $k = 12$ as a function of the sparsity prior weight (λ). Since score matching does not have a λ parameter, it is plotted at a constant. **B** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of overcompleteness at $k = 12$. **C** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of k-sparseness at 2-times overcompleteness. **D, E** Same plots as **B** and **C** at a point where methods do not perform as well: $k = 6$ and 3-times overcomplete. In **B-E**, the L_4 and Flattened Coulomb lines are largely overlapping.

consistently than ICA models. This suggests that the linear inference in ICA models can only recover dictionaries for moderately overcomplete representations.

2.6. Recovery of the analysis matrix with overcomplete ICA

In analysis dictionary learning (Elad et al., 2007; Rubinstein et al., 2013; Ophir et al., 2011; Chun and Fessler, 2018), the goal is to find a set of dictionary elements such that each datapoint is orthogonal to many (or k) elements of the set, rather than to find a generative model for the data. Since overcomplete ICA models fit more naturally into an analysis framework rather than a synthesis framework, we perform a similar analysis as in Section 2.5, except here, we generate the data from an analysis model using the method described in Rubinstein et al. (2013) (see Section 3 for details).

Across overcompleteness (OC) and k, the methods are generally more similar than in Section 2.5. Note that in this analysis, k is the number of zeros in the projection per data point, which is different than k in the synthesis data, which is the number of elements included per data point. We find three main trends as a function of overcompleteness and k: the L_2 cost tends to perform worse or as well as the other costs, score matching can perform slightly better or slightly worse than other methods, and L_4, Coulomb, and Flattened Coulomb are not well separated. Although this analysis does not distinguish the costs proposed here, it does show that the L_2 cost is suboptimal for both the synthesis and analysis problems.

2.7. Experiments on natural images

When ICA is applied to real data, one typically does not know the exact sparse distribution of the data. For instance, for a natural images dataset, we no longer have a ground truth mixing matrix or known prior, and furthermore, it is not likely that natural image patches come from a simple generative model (Hyvärinen and Köster, 2007; Lücke et al., 2009). However, the effects of coherence control on the distribution of dictionary elements learned can be evaluated. Specifically, we can look at the coherence of learned dictionaries and whether different methods prevent duplicate features from being learned.

We train 2-times overcomplete ICA models on 8-by-8 whitened image patches from the Van Hateren database (van Hateren and van der Schaaf, 1998) at a fixed value of sparsity across costs found by binary search on λ. The score matching cost has no λ parameter to trade off sparsity versus coherence although it finds solutions of similar sparsity to the value chosen for the other costs. It is known that for natural images data sets, bases learned with ICA can be well-fit by Gabor filters (Bell and Sejnowski, 1997). Hence, we evaluate the distribution of the learned basis by inspecting the parameters obtained from fitting the bases to Gabor filters (see Section 3.6 for details).

The distributions of angles from the trained ICA models are in line with the theoretical results from Section 2.3. The L_2 cost has more pairwise angles close to zero compared to the other costs with the L_4 having the smallest coherence (Fig 6A). Although the probabilities for the L_2 cost in Fig 6A trend to 10^{-5} for small angles, there is a peak at zero which is closer to 10^{-3}. For the natural images analysis, each learned dictionary has $2 \times 8 \times 8 = 128$ elements which corresponds to 8128 pairwise angles. This means that for the L_2 cost there are approximately 8 pairs of elements that are nearly identical on average which means

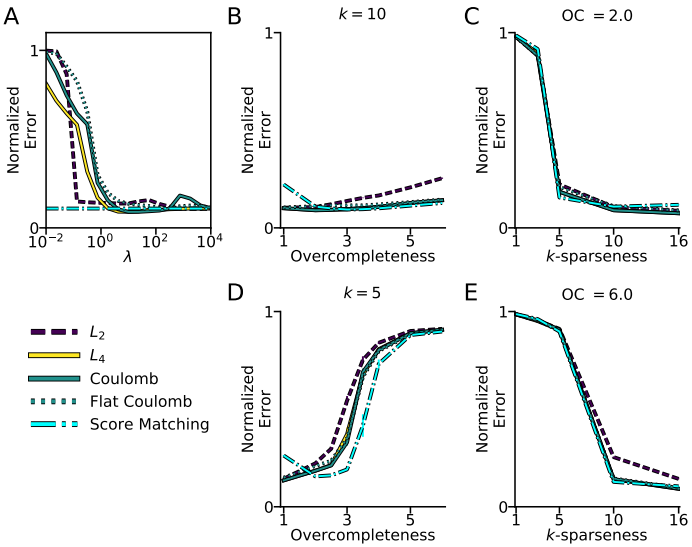


Figure 5: Coherence control costs do not all recover analysis matrices well. All ground truth analysis matrices were generated from the Soft Coherence cost and had a data dimension of 32. Color and line style legend are preserved across panels. **A** The normalized recovery error (see Section 3 for details) for a 2-times overcomplete analysis matrix and $k = 10$ as a function of the sparsity prior weight (λ). Since score matching does not have a λ parameter, it is plotted at a constant. **B** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of overcompleteness at $k = 10$. **C** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of k-sparseness at 2-times overcompleteness. **D, E** Same plots as **B** and **C** at a point where methods do not perform as well: $k = 5$ and 6-times overcomplete. In **B-E**, the L_4, Coulomb, and Flattened Coulomb lines are largely overlapping.

that about 5% of the elements are redundant. Similarly, as shown in Fig 6B, the Random Prior and Coulomb costs have lower coherence when the second order terms are removed and behave more similarly to the L_4 cost. These distributions also show that ICA models with the L_2 cost tend to learn duplicate bases from natural images.

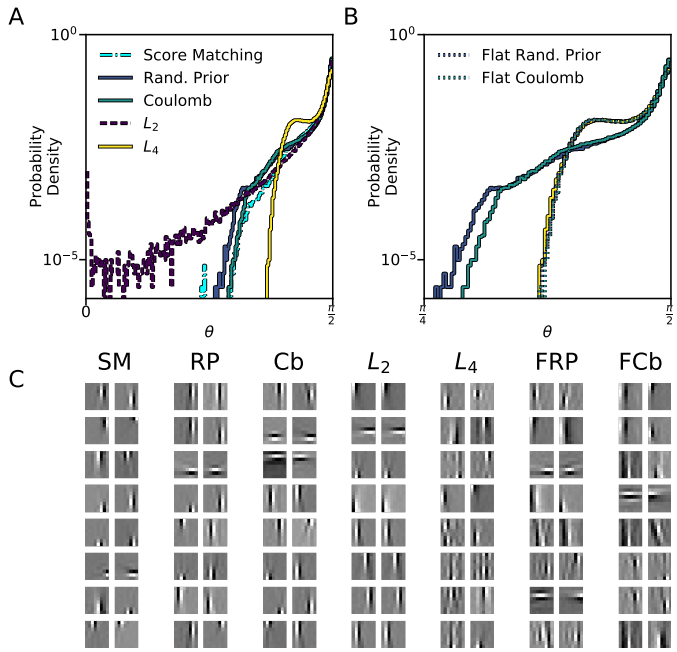


Figure 6: The coherence of an overcomplete dictionary learned from natural images depends on the coherence control cost. Results from fitting a 2-times overcomplete model on 8-by-8 natural image patches. **A, B** Pairwise angle distributions (log scale) across costs for the learned dictionaries for a fixed value of sparsity across costs. **B** Comparison between the Random Prior and Coulomb costs and their flattened versions. The L_4 distribution is also shown for comparison. Note that the horizontal axis covers 45 to 90 degrees and that the Flattened Random Prior and Flattened Coulomb lines largely overlap the L_4 line. **C** For each cost from **A** and **B**, the 8 pairs of bases with smallest pairwise angle are shown. Since the overall sign of a basis element is arbitrary, the bases have been inverted to have positive inner product, if needed, for visualization.

For the range of sparsities which were considered, the visual appearance of the individual bases is similar to results from previous ICA work and similar across costs (L_4 bases are shown in Fig 7A). The tiling properties of the learned dictionaries can also be visualized directly. The coordinates of the center of the fit Gabor filter, rotations, and scales tile the space for the L_2, L_4, and Flattened Coulomb costs (Fig 7B). The dimensions and rotation of the rectangle represent the envelope widths and planar rotation angle respectively. This is similarly true for the planar rotation angle against the oscillation wavelength of the Gabor (Fig 7C) and the envelope widths and wavelengths (Fig 7D). Although these distributions look qualitatively similar, the underlying dictionaries can have very different coherence.

These results demonstrate that the L_2 cost learns undesirable, high-coherence overcomplete dictionaries on real data. Visually inspecting the bases or even their tiling properties may not reveal the redundant set of basis functions. To reveal this type of redundancy one has to measure the coherence or the distribution of pairwise angles of a dictionary directly.

3. Methods

In this section we summarize previously proposed coherence control methods, our model implementations, and datasets used.

3.1. Previously proposed coherence control methods

3.1.1. Reconstruction cost and the L_2 cost

Le et al. (2011) propose adding a reconstruction cost to the ICA prior (RICA) as a form of coherence control, which they show is equivalent to a cost on the L_2 norm of the difference between the Gram matrix of the filters and an identity matrix for whitened data

$$\begin{aligned} C_{\text{RICA}} &= \frac{1}{N}\sum_{ij}(X_j^{(i)} - \sum_{kl} W_{kj}W_{kl}X_l^{(i)})^2 \\ &\propto C_{L_2} = \frac{1}{2}\sum_{ij}(\delta_{ij} - \sum_k W_{ik}W_{jk})^2 = \frac{1}{2}\sum_{ij}(\delta_{ij} - \cos\theta_{ij})^2, \end{aligned} \tag{14}$$

where W_{ij} is the component of the ith source for the jth mixture , $X_j^{(i)}$ is the jth element of the ith sample, θ_{ij} is the angle between pairs of basis, and δ_{ij} is the Kronecker delta.

The L_2 cost has also been proposed as a form of coherence control (Ramırez et al., 2009; Sigg et al., 2012; Bao et al., 2014; Chun and Fessler, 2018). Equiangular tight-frames (ETFs) are frames (overcomplete dictionaries) which have minimum coherence. The fact that an ETF has minimum coherence is used to motivate the L_2 cost as a form of coherence control. A matrix $W \in \mathbb{R}^{L\times D}$ is an ETF if

$$|\sum_k W_{ik}\cdot W_{jk}| = \cos\alpha,\ \forall i \neq j \tag{15}$$

for some angle, α, and

$$\sum_k W_{ki}W_{kj} = \frac{L}{D}\delta_{ij}. \tag{16}$$

The L_2 cost will encourage Eq 16 to be satisfied, but does not encourage Eq 15 to be satisfied as we show in Theorem 1.

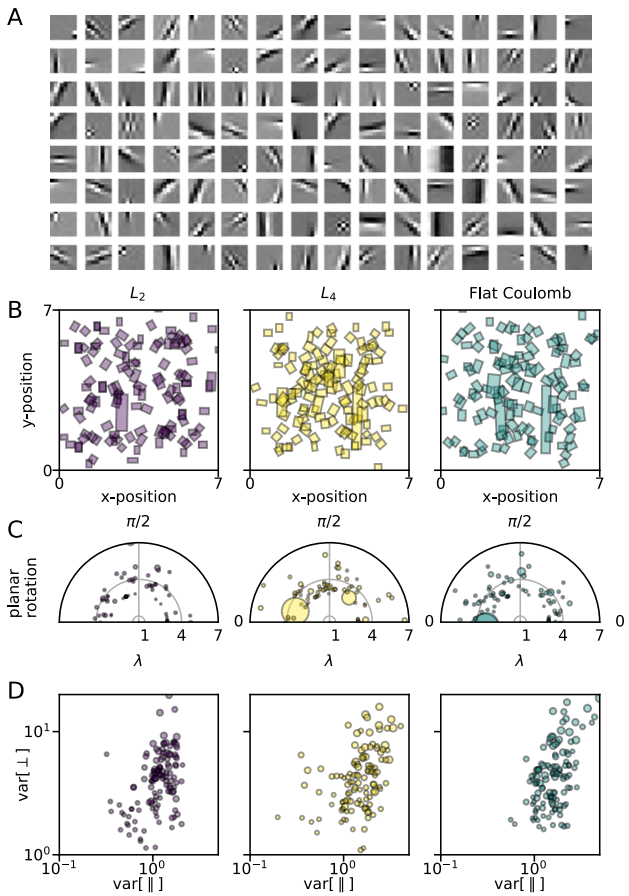


Figure 7: All coherence costs learn a dictionary that approximately tiles the space of Gabor Filters. **A** Dictionary learned using the L_4 cost on 8-by-8 natural image patches. **B** Distributions of locations, envelope scales, and rotations. Rectangle position: center of Gabor fit in pixel coordinates, rectangle rotation: planar-rotation of the Gabors, rectangle shape: envelope width parallel and perpendicular to the oscillation axis. **C** Distributions of rotations, log-wavelengths (λ), and envelope widths. Polar plots of planar-rotation angle and log-spatial wavelength of the Gabors. Marker size scales with geometric mean of envelope widths. **D** Distributions of envelope scales and log-wavelengths. Log-scale plot of envelope widths-squared parallel and perpendicular to the oscillation axis of the Gabors. Circle size scales with log-wavelength.

3.1.2. Quasi-orthogonality constraint

Hyvärinen et al. (1999) suggest a quasi-orthogonality update which approximates a symmetric Gram-Schmidt orthogonalization scheme for an overcomplete basis, W, which is formulated as

$$W \leftarrow \frac{3}{2}W - \frac{1}{2}WW^TW. \tag{17}$$

3.1.3. Random prior cost

A prior on the distribution of pairwise angles was proposed to encourage low coherence (Hyvärinen and Inki, 2002). The prior is the distribution of pairwise angles for two vectors drawn from a uniform distribution on the n-sphere[1]

$$C_{\text{Random prior}} = -\sum_{i \neq j} \log P(\cos\theta_{ij}) \propto -\sum_{i \neq j} \log(1 - \cos^2\theta_{ij}). \tag{18}$$

3.1.4. Score Matching

Score matching is a training objective function for non-normalized statistical models of continuous variables(Hyvärinen, 2005). It has been used to learn overcomplete ICA models. The score function is derivative of the log-likelihood of the model or data distribution with respect to the data

$$\psi(X;\Theta) = \nabla_X \log p(X;\Theta) \tag{19}$$

The score matching objective is the mean-squared error between the model score, $\psi(X;\Theta)$, and data score, $\psi_{\mathcal{D}}(X;\Theta)$ averaged over the data, $\mathcal{D}$

$$J(\Theta) = \frac{1}{2}\int_X p_{\mathcal{D}}(X)||\psi(X;\Theta) - \psi_{\mathcal{D}}(X;\Theta)||^2. \tag{20}$$

3.2. Coherence-based costs

The coherence of a dictionary is defined as the maximum absolute value of the off-diagonal elements of the Gram matrix (Davenport et al., 2011) as in Eq 6. Using the coherence as a cost is equivalent to using the L_∞ norm version of the L_p cost function. We find that this cost is difficult to numerically optimize with both second order methods and gradient descent since the derivative through the max operation will only act on one pair of bases at each optimization step, although it should find solution with local minima of coherence. An easier to optimize, but heuristic, version of this cost is the sum over all off-diagonal elements whose squares are larger than the mean squared value

$$C_{\text{Soft Coherence}} = \sum_{i \neq j \text{ s.t. } \cos\theta_{ij}^2 > \cos\hat{\theta}^2} |\cos\theta_{ij}|, \text{ with } \cos\hat{\theta}^2 = \underset{i \neq j}{\text{mean}}(\cos\theta_{ij}^2). \tag{21}$$

We find that this cost does not optimize well for coherence control in ICA when fit with data, but it can be used to create low-coherence mixing matrices for generating data with known structure in Section 2.5.

1. For both the Random Prior and the Coulomb cost, we regularize the costs and their derivatives near $|\cos\theta| = 1$ by adding a small positive constant in the objective: $1 - \cos\theta_{ij}^2 \to 1 + |\epsilon| - \cos\theta_{ij}^2$.

3.3. Model implementation

All models were implemented in Theano (Theano Development Team, 2016). ICA models were trained using the L-BFGS-B (Byrd et al., 1995) implementation in SciPy (Jones et al., 2001–2017). FISTA (Beck and Teboulle, 2009) was used for MAP inference in the sparse coding model and the weights were learned using L-BFGS-B. All weights were training with the norm-ball projection (Le et al., 2011) to keep the bases normalized. A repository with code to reproduce the results is available[2]. For ICA models with coherence costs, the coherence control cost with no sparsity penalty ($\lambda = 0$) was used as the objective for Figs 2 and 3.

3.4. Datasets

For all datasets and models, the number of samples in a dataset was equal to 10 times the number of model parameters, that is, $10 \times n_{\text{sources}} \times n_{\text{mixtures}}$. Datasets were mean-centered and whitened.

3.4.1. k-sparse and analysis datasets

For both datasets, dictionaries were generated by minimizing the Soft Coherence cost. The synthesis data was generated by keeping k random elements per data sample from draws of a diagonal multivariate Laplacian distribution, zeroing out the rest, and combining them with the mixing matrix, $X_j^{(i)} = \sum_{j=1}^{L} A_{jk} S_k^{(i)}$, with k-sparse S.

For the analysis dataset, we use the method proposed by Rubinstein et al. (2013). $X^{(i)}$ is initialized to i.i.d. Gaussian samples. Then, a subset of k elements, $W^{A_{\text{sub}}}$, of the analysis matrix, W^A, are chosen per data sample and are used to form a projection matrix which removes the subspace spanned by $W^{A_{\text{sub}}}$ from $X^{(i)}$, $X_j^{(i)} = \sum_{k=1}^{L}(I - \Omega^T\Omega)_{jk}^{(i)} X_k^{(i)}$, where Ω is a basis for the subspace spanned by $W^{A_{\text{sub}}}$. This ensures that at least k elements of $W^A X^{(i)}$ will be zero.

3.4.2. Natural images dataset

Images were taken from the Van Hateren database (van Hateren and van der Schaaf, 1998). We selected images where there was no evident motion blur and minimal saturated pixels. 8-by-8 patches were taken from these images and whitened using PCA.

3.5. Dictionary recovery error

If the mixing matrix A is recovered perfectly, W^T will be a permutation of A. To estimate the closeness to a permutation matrix, the matrix $P_{ij} = |A_i^T \cdot W_j|$ is created. For the analysis dictionaries W^A, $P_{ij} = |W_i^A \cdot W_j|$. The Hungarian method (Kuhn, 1955) is used to find the best assignment between A_i^T (or W_i^A) and W_j. Given this best assignment, the median angle between the elements is returned.

This error is normalized by calculating the same quantity for matrices, W^*, which were recovered from mixing matrices A^*, which were from the same distribution as A but

2. `https://github.com/JesseLivezey/oc_ica`

with different random initializations. After this normalization, perfect recovery gives a normalized error of 0 and a random recovery gives a normalized error of 1.

3.6. Fitting Gabor parameters

We fit the Gabor parameters (Ringach, 2002) to the learned bases using an iterative grid-search and optimization scheme which gave the best results on generated filters. The learned parameters were the center vector: $\{\mu_x, \mu_y\}$, planar-rotation angle: θ, phase: ϕ, oscillation wave-vector k, and envelope variances parallel and perpendicular to the oscillations: $\sigma_{\hat{x}}^2$ and $\sigma_{\hat{y}}^2$ respectively. Because they are constrained to be positive, the log of the parameters: $\sigma_{\hat{x}}^2$ and $\sigma_{\hat{y}}^2$ are optimized. To keep the wavelength of the Gabor larger than 2 pixels, instead of optimizing k directly we optimize ρ with $k = \frac{2\pi}{2\sqrt{2}+\exp(\rho)}$. Shorter wavelengths would be aliased by the pixel sampling.

$$\begin{aligned}
\hat{x} &= \cos(\theta)x + \sin(\theta)y \\
\hat{y} &= -\sin(\theta)x + \cos(\theta)y \\
\hat{\mu}_x &= \cos(\theta)\mu_x + \sin(\theta)\mu_y \\
\hat{\mu}_y &= -\sin(\theta)\mu_x + \cos(\theta)\mu_y \\
\mathrm{Gabor}(x, y; \mu_x, \mu_y, \theta, \sigma_{\hat{x}}, k, \sigma_{\hat{y}}, \phi) &= \exp\left(-\frac{(\hat{x}-\hat{\mu}_x)^2}{2\sigma_{\hat{x}}^2} - \frac{(\hat{y}-\hat{\mu}_y)^2}{2\sigma_{\hat{y}}^2}\right)\sin(k\hat{x}+\phi)
\end{aligned} \tag{22}$$

A global, gradient-based optimization leads to many local minia where one lobe of the Gabor would be well fit, but the other would not. We found that a combination of an iterative approach with gradient-based optimization of subsets of the parameters worked well. The procedure for finding the best Gabor kernel parameters was to save the parameter set with best mean-squared error after the following iterations

1. for different initial envelope widths, fit the center location for the envelope to the blurred, absolute value of the basis element,
2. for different initial planar rotations and frequencies, numerically optimize the planar rotation, phase, and frequency of the Gabor
3. for the best fit from above, re-optimize the centers, widths, and phases,
4. re-optimize all parameters from best previous fit.

A repository with code to fit the Gabor kernels is posted online [3].

4. Discussion

Learning overcomplete sparse representations of data is often an extremely informative first stage in analyzing multivariate data. In the field of neuroscience, overcomplete dictionary learning serves as a theoretical model of how the brain analyzes sensory inputs (Olshausen

3. `https://github.com/JesseLivezey/gabor_fit`

and Field, 1996; Klein et al., 2003; Smith and Lewicki, 2006; Rehn and Sommer, 2007; Zylberberg et al., 2011; Carlson et al., 2012), motivating the study of methods suitable in the overcomplete regime. For all of these purposes, the heavy computational cost of the nonlinear inference step involved in common sparse coding approaches can be an obstacle for large datasets. For learning complete sparse representations, ICA with just a linear inference mechanism is a viable alternative with drastically reduced computational demand. Here, we investigated the limitations of linear inference in overcomplete dictionary learning.

Any multidimensional method for extracting signal components needs a form of coherence control to prevent components from becoming co-aligned and therefore redundant. We first compared different coherence costs' ability to prevent the learning of coherent dictionary elements in the overcomplete case. We show theoretically and by simulation, that the L_2 cost, which successfully achieves minimum coherence (orthogonality) in the complete case, exhibits pathological global minima with maximum coherence in the overcomplete case. Encouraging diverse, incoherent, or orthogonal solutions has been proposed as a desirable additional cost function for dictionary learning and deep network applications (Le et al., 2011; Sigg et al., 2012; Ramırez et al., 2009; Bao et al., 2014; Brock et al., 2017; Chun and Fessler, 2018; Bansal et al., 2018), typically by applying a power/norm to the difference between an identity matrix and the gram matrix. However, the impact of previously proposed costs (L_1 and L_2) on coherence has not been directly explored, and we have shown that they do not encourage lower coherence.

We propose novel cost functions which do not suffer from pathological minima in the overcomplete case. Specifically, we propose the L_4 cost and the flattened versions of the Coulomb and Random Prior costs, and show that they yield dictionaries with lower coherence than the cost functions that have been proposed earlier. At the same time, these new cost functions have smaller effects on incoherent basis pairs, thus leading to dictionaries that reflect the structure of the data rather than effects from the coherence term.

We show that the methods of coherence control proposed here can extent the regime of overcompleteness and sparseness, in which ICA methods can successfully learn recover synthetic dictionaries with linear inference. However, this expansion of the regime of applicability is still limited. Even the improved methods begin to fail when overcompleteness grows beyond two-fold (for 32 dimensional data) or if the data is k-sparse with small k. The problem to deal with extremely k-sparse data is counterintuitive at first, because nonlinear inference methods usually do better as k is decreased because the combinatorial search for the best sparse support in the inference becomes easier (Davenport et al., 2011). However, linear inference in ICA models cannot recover extremely sparse sources unlike sparse coding models, which do not fail in the small-k limit.

Based on the observations in Section 2.4 that the leading terms in the Taylor expansion will have the largest influence on the distribution of angles, we proposed a modification to the Coulomb and Random Prior costs to make them more similar to the L_4 costs which resulted in lower coherence solutions. We do not have a strong way of distinguishing the L_4 cost and the flattened Coulomb and Random Prior costs. However, it may be possible that the higher order terms become important in specific situations we have not considered here and may distinguish these costs.

In this work, coherence costs were investigated as augmented costs for overcomplete ICA. Although, the proposed methods only modestly extended the ability of linear inference to

perform overcomplete model recovery, the proposed coherence control costs lead to learned dictionaries with significantly higher coherence. In the case of dictionaries learned on natural image patches, we show that the L_4 cost prevents the model from learning duplicated bases, unlike the L_2 cost. For other application where low coherence might be desirable (e.g., sparse coding, deep learning, the proposed cost functions could provide improved coherence control. We note that variations of the ICA sparsity prior and mismatch with data sparsity structure have not been systematically explored here and are another potential topic of further investigation. All told, our study explores the power and limitations of linear inference for overcomplete dictionary learning.

Acknowledgments

We thank Yubei Chen, Alexander Anderson, and Kristofer Bouchard for helpful discussions. We also thank the editor and reviewers for their helpful comments which improved the quality of the manuscript. We thank the reviewer for pointing out a simplified proof of Theorem 1. JAL was supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under U.S. Department of Energy Contract No. DE-AC02-05CH11231. JAL and AFB were supported by the Applied Mathematics Program within the Office of Science Advanced Scientific Computing Research of the U.S. Department of Energy under contract No. DE-AC02-05CH11231. FTS was supported by the National Science Foundation grants IIS1718991, IIS1516527, INTEL, and the Kavli Foundation. We acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

Appendix A. Minima analysis for the L_2 and L_4 costs for a 2-dimensional space.

We can write a 2 dimensional, 2 times overcomplete dictionary as

$$W = \begin{pmatrix} 1 & 0 \\ \cos\theta_1 & \sin\theta_1 \\ \cos\theta_2 & \sin\theta_2 \\ \cos(\theta_2+\theta_3) & \sin(\theta_2+\theta_3) \end{pmatrix} \tag{23}$$

If we restrict the dictionary to be composed of two orthogonal pairs, without loss of generality, we can write W as

$$W|_{\theta_1=\theta_3=\frac{\pi}{2}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2 \end{pmatrix} \tag{24}$$

For the general dictionary For the L_p cost for even p is

$$\begin{aligned}
C_{L_p} =& \frac{1}{p}\sum_{ij}(\delta_{ij} - \sum_k W_{ik}W_{jk})^p \\
=& \frac{1}{p}\sum_{ij}\left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & \cos\theta_1 & \cos\theta_2 & \cos(\theta_2+\theta_3) \\ \cos\theta_1 & 1 & \cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2 & \cos\theta_1\cos(\theta_2+\theta_3)+\sin\theta_1\sin(\theta_2+\theta_3) \\ \cos\theta_2 & \cos\theta_1\cos\theta_2+\sin\theta_1\sin\theta_2 & 1 & \cos\theta_2\cos(\theta_2+\theta_3)+\sin\theta_2\sin(\theta_2+\theta_3) \\ \cos(\theta_2+\theta_3) & \cos\theta_1\cos(\theta_2+\theta_3)+\sin\theta_1\sin(\theta_2+\theta_3) & \cos\theta_2\cos(\theta_2+\theta_3)+\sin\theta_2\sin(\theta_2+\theta_3) & 1 \end{pmatrix}\right)^p \\
=& \frac{1}{p}\left(2\cos^p\theta_1 + 2\cos^p\theta_2 + 2\cos^p(\theta_2+\theta_3)\right. \\
&+ 2(\cos\theta_1\cos\theta_2+\sin\theta_1\sin\theta_2)^p + 2(\cos\theta_1\cos(\theta_2+\theta_3)+\sin\theta_1\sin(\theta_2+\theta_3))^p \\
&\left.+2(\cos\theta_2\cos(\theta_2+\theta_3)+\sin\theta_2\sin(\theta_2+\theta_3))^p\right)
\end{aligned} \tag{25}$$

This can be simplified for the case of 2 orthogonal bases (Eq 24)

$$\begin{aligned}
C_{L_p}|_{\theta_1=\theta_3=\frac{\pi}{2}} &= \frac{1}{p}\left(2\cos^p\theta_2 + 2\sin^p\theta_2 + 2\sin^p\theta_2 + 2\cos^p\theta_2 + 2(-\cos\theta_2\sin\theta_2 + \sin\theta_2\cos\theta_2)^p\right) \\
&= \frac{1}{p}\left(4\sin^p\theta_2 + 4\cos^p\theta_2\right).
\end{aligned} \tag{26}$$

For $p=2$ and $p=4$ this is

$$\begin{aligned}
C_{L_2}|_{\theta_1=\theta_3=\frac{\pi}{2}} &= \frac{1}{2}(4\cos^2\theta_2 + 4\sin^2\theta_2) = 2 \\
C_{L_4}|_{\theta_1=\theta_3=\frac{\pi}{2}} &= \cos^4\theta_2 + \sin^4\theta_2.
\end{aligned} \tag{27}$$

Here we tabulate the full Hessian matrices, eigenvalues, and eigenvectors for the analysis in Sections 2.1 and 2.2.

A.1. L_2 cost

$$
\begin{aligned}
C_{L_2}(\theta_1,\theta_2,\theta_3)|_{\theta_1,\theta_3=\frac{\pi}{2}} &= 2 \\
\frac{\partial C_{L_2}(\theta_1,\theta_2,\theta_3)}{\partial\vec{\theta}}|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\
H(C_{L_2})|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \begin{pmatrix} 2 & 0 & 2\cos 2\theta_2 \\ 0 & 0 & 0 \\ 2\cos 2\theta_2 & 0 & 2 \end{pmatrix} \\
\text{EVal.}(H_{L_2})|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \begin{pmatrix} 0 \\ 4\sin^2\theta_2 \\ 4\cos^2\theta_2 \end{pmatrix} \\
\text{EVec.}(H_{L_2})|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}
\end{aligned}
\tag{28}
$$

A.2. The second eigenvalue of the L_2 cost

For the second eigenvalue and eigenvector, $4\sin^2\theta_2$ and (-1, 0, 1), centered at $\theta_1 = \theta_3 = \pi/2$, $\theta_2 = 0$, the cost along the direction of the eigenvector is

$$
\begin{aligned}
C(\text{EVal}_2,\Delta\theta) = \frac{1}{2}\big(&2\cos^2(\pi/2-\Delta\theta) + 2\cos^2 0 + 2\cos^2(0+\pi/2+\Delta\theta) \\
&+ 2(\cos(\pi/2-\Delta\theta)\cos 0 + \sin(\pi/2-\Delta\theta)\sin 0)^2 \\
&+ 2(\cos(\pi/2-\Delta\theta)\cos(0+\pi/2+\Delta\theta) \\
&+ \sin(\pi/2-\Delta\theta)\sin(0+\pi/2+\Delta\theta))^2 \\
&+2(\cos 0\cos(0+\pi/2+\Delta\theta) + \sin 0\sin(0+\pi/2+\Delta\theta))^2\big) \\
= 1 &+ 4\sin^2\Delta\theta + \sin^4\Delta\theta + \cos^4\Delta\theta - 2\sin^2\Delta\theta\cos^2\Delta\theta
\end{aligned}
\tag{29}
$$

which Taylor-expanded around $\Delta\theta = 0$ gives

$$
\begin{aligned}
C(\text{EVal}_2,\Delta\theta) &= 1 + 4(\Delta\theta^2 + \frac{\Delta\theta^4}{3}) + (\Delta\theta^4) + (1 - 2\Delta\theta^2 + \frac{5\Delta\theta}{3}) \\
&\quad - 2(\Delta\theta^2 - \frac{4\Delta\theta}{3}) + O(\Delta\theta^5) \\
&= 2 + \frac{4\Delta\theta^4}{3} + O(\Delta\theta^5).
\end{aligned}
\tag{30}
$$

This shows that although the second (and third) derivative is zero, the fourth derivative is positive, meaning this point is a minimum.

A.3. L_4 cost

$$
\begin{aligned}
C_{L_4}(\theta_1,\theta_2,\theta_3)|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \cos^4\theta_2 + \sin^4\theta_2 \\
\frac{\partial C_{L_4}(\theta_1,\theta_2,\theta_3)}{\partial\vec{\theta}}|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \frac{1}{2}\begin{pmatrix} \sin 4\theta_2 & -2\sin 4\theta_2 & -\sin 4\theta_2 \end{pmatrix} \\
H(C_{L_4})|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \begin{pmatrix} -2\cos 4\theta_2 & 2\cos 4\theta_2 & \cos 2\theta_2 + \cos 4\theta_2 \\ 2\cos 4\theta_2 & -4\cos 4\theta_2 & -2\cos 4\theta_2 \\ \cos 2\theta_2 + \cos 4\theta_2 & -2\cos 4\theta_2 & -2\cos 4\theta_2 \end{pmatrix} \\
\text{EVal.}(H_{L_4})|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \begin{pmatrix} \cos 2\theta_2 - \cos 4\theta_2 \\ -\frac{1}{2}\cos 2\theta_2 - \frac{7}{2}\cos 4\theta_2 - \ldots \\ \ldots\frac{1}{2\sqrt{2}}\sqrt{34 - 2\cos 2\theta_2 + \cos 4\theta_2 - 2\cos 6\theta_2 + 33\cos 8\theta_2} \\ -\frac{1}{2}\cos 2\theta_2 - \frac{7}{2}\cos 4\theta_2 + \ldots \\ \ldots\frac{1}{2\sqrt{2}}\sqrt{34 - 2\cos 2\theta_2 + \cos 4\theta_2 - 2\cos 6\theta_2 + 33\cos 8\theta_2} \end{pmatrix} \\
\text{EVec.}(H_{L_4})|_{\theta_1,\theta_3=\frac{\pi}{2}} &= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \\
&\begin{pmatrix} -1 \\ (\frac{\sqrt{2}}{8}\sqrt{\begin{matrix} 2\cos 2\theta_2 + \cos 4\theta_2 - \ldots \\ \ldots 2\cos 6\theta_2 + 33\cos 8\theta_2 + 34 \end{matrix}} - \ldots \\ \ldots - 2\cos 2\theta_2)\sec 4\theta_2 + \frac{1}{4} \\ 1 \end{pmatrix}, \\
&\begin{pmatrix} -1 \\ \frac{1}{4} - (2\cos\frac{1}{4}\theta_2 + \ldots \\ \ldots\frac{\sqrt{2}}{8}\sqrt{\begin{matrix} -2\cos 2\theta_2 + \cos 4\theta_2 - 2\cos 6\theta_2 + \ldots \\ \ldots 33\cos 8\theta_2 + 34 \end{matrix}})\sec 4\theta_2 \\ 1 \end{pmatrix}
\end{aligned}
\tag{31}
$$

Appendix B. Proofs of Theorems 1 and 2

B.1. L_2 cost minima and equiangular tight-frames: proof of Theorem 1

We can prove Theorem 1 in two ways. The first way is by showing that a dictionary of concatenated identity matrices is at the global minimum of the C_{L_2} cost and is a short proof. The second is longer, but makes an explicit connection to the C_{L_2} cost optimizing one but not both of the conditions of an equiangular tight-frame (Eqs 15 and 16).

B.2. Proof 1

Here we prove Theorem 1 by showing that a dictionary of stacked identity matrices is a global minimum of the C_{L_2} cost.

Proof [Proof 1 of Theorem 1] For an overcomplete dictionary $W \in \mathbb{R}^{L\times D}$, the expression for the minimum possible coherence is known (Strohmer and Heath Jr, 2003)

$$\text{Coherence}_{\min}(W) = \sqrt{\frac{L-D}{D(L-1)}}. \tag{32}$$

The C_{L_2} cost is the sum of the off-diagonal elements of the squared Gram matrix. The minimum possible value for each element is given by the square of Eq 32 and so the minimum of the C_{L_2} cost will be achieved if all (L^2-L) off-diagonal elements are equal to the square of the value in Eq 32

$$C_{L_2,\ \min} = (L^2-L)\sqrt{\frac{L-D}{D(L-1)}}^2 = \frac{L(L-D)}{D}. \tag{33}$$

For $W = W_0$ constructed as an identity matrix concatenated L/D times, the sum of the off-diagonal elements of the squared Gram matrix is

$$C_{L_2}(W_0) = (\frac{L}{D}^2 - \frac{L}{D})D = \frac{L(L-D)}{D} \tag{34}$$

which is the minimum value. However, W_0 has coherence $= 1$ since the stacked identity matrices contain identical elements. ■

B.3. Proof 2

Here we prove Theorem 1 in two steps: first we can show the equivalence, up to an additive constant, of minimizing the L_2 cost and minimizing the L_2 norm of the error of Eq 16. Then we show that the pathological solution (Section 2.1) is at the global minimum of this cost.

Proof [Proof 2 of Theorem 1] For a normalized ($\sum_k W_{ik}^2 = 1,\ \forall\ i$) matrix, W

$$\begin{aligned}
C_{L_2} &= \sum_{ij}(\sum_k W_{ik}W_{jk} - \delta_{ij})^2 \\
&= \sum_{ij}(\sum_k W_{ik}W_{jk} - \delta_{ij})(\sum_l W_{il}W_{jl} - \delta_{ij}) \\
&= \sum_{ijkl} W_{ik}W_{jk}W_{il}W_{jl} - 2\sum_{ijk} W_{ik}W_{jk}\delta_{ij} + \sum_{ij}\delta_{ij}^2 \\
&= \sum_{ijkl} W_{ik}W_{jk}W_{il}W_{jl} - 2\sum_{ik} W_{ik}^2 + \text{const.}(L) \\
&= \sum_{ijkl} W_{ik}W_{jk}W_{il}W_{jl} + \text{const.}(L)
\end{aligned} \tag{35}$$

$$\begin{aligned}
C_{\text{Eq 16}} &= \sum_{kl}(\sum_i W_{ik}W_{il} - \frac{L}{D}\delta_{kl})^2 \\
&= \sum_{kl}(\sum_i W_{ik}W_{il} - \frac{L}{D}\delta_{kl})(\sum_j W_{jk}W_{jl} - \frac{L}{D}\delta_{kl}) \\
&= \sum_{ijkl} W_{ik}W_{il}W_{jk}W_{jl} - 2\sum_{ikl}\frac{L}{D}W_{ik}W_{il}\delta_{kl} + \sum_{kl}(\frac{L}{D}\delta_{kl})^2 \\
&= \sum_{ijkl} W_{ik}W_{il}W_{jk}W_{jl} - 2\frac{L}{D}\sum_{ik} W_{ik}^2 + \text{const.}(L,D) \\
&= \sum_{ijkl} W_{ik}W_{il}W_{jk}W_{jl} + \text{const.}(L,D)
\end{aligned} \tag{36}$$

where $\sum_k W_{ik}^2 = 1,\ \forall\ i$ is used extensively and the index letters were initially chosen to make the comparison of the final lines more clear. Le et al. (2011) show this first equivalence and that the L_2 cost is also shown to be equivalent to the reconstruction cost with whitened data (Lemmas 3.1 and 3.2 in Le et al. (2011)).

Now we can show that the same dictionary that was described in Section 2.1: W_0, an integer overcomplete dictionary where each set of complete bases is an orthonormal basis, exactly satisfies Eq 16 and so is a minimum of the L_2 cost. This solution is very far away from an ETF in the sense of Eq 15. A dictionary of this form, $W \in \mathbb{R}^{L\times D}$, can be constructed as $W_{ij} = \delta_{(i \bmod D)j}$ with $L = n \times D,\ n > 1,\ \in \mathbb{Z}$, that is, a D dimensional identity matrix tiled n times.

This construction satisfies Eq 16 and therefore has a value of 0 for $C_{\text{Eq 16}}$. Since $C_{\text{Eq 16}}$ is a sum of quadratic, and therefore non-negative, terms, this construction is a global minimum of $C_{\text{Eq 16}}$ and the L_2 cost

$$\begin{aligned}
\sum_k W_{ki}W_{kj} &= \sum_k \delta_{(k \bmod D)i}\delta_{(k \bmod D)j} \\
&= n\delta_{ij} \\
&= \frac{L}{D}\delta_{ij} \\
&\Rightarrow C_{\text{Eq 16}} = 0
\end{aligned} \tag{37}$$

as $k \bmod D = i$ a total of n times when $i = j$.

However, this construction has off-diagonal Gram matrix elements that are either 0 or 1:

$$\begin{aligned}
\cos\theta_{ij} &= \sum_k W_{ik}W_{jk} \\
&= \sum_k \delta_{(i \bmod D)k}\delta_{(j \bmod D)k} \\
&= \delta_{(i \bmod D)(j \bmod D)},
\end{aligned} \tag{38}$$

which is not equal or close to an equiangular solution, that is, $\cos\theta_{ij} = \cos\alpha,\ \forall i \neq j$. ■

B.4. Invariance to continuous transformations: proof of Theorem 2

Here we prove Theorem 2: the L_2 cost, initialized from the pathological solution, is invariant to transformations, Φ, constructed as orthogonal rotations applied to any basis subset and an identity transformation on the remaining bases. This shows that low coherence and high coherence configurations are both global minima of the L_2 cost.

Proof [Proof of Theorem 2] For an D dimensional space with an n times overcomplete dictionary, with n an integer greater than 1, the pathological dictionary configuration is a orthonormal basis tiled n times. The dictionary elements can be labels as the sequential subsets of orthornormal subsets $W_1, \ldots, W_D, \ldots, W_{2D}, \ldots, W_{n\times D}$. So, bases W_1 through W_D form a full-rank, orthonormal basis and this basis is tiled n times.

Consider the following partition of the bases: partition $\mathcal{A}$ is the first orthonormal set, bases W_1 through W_D, and partition $\mathcal{B}$ the remainder of the bases, W_{D+1} through $W_{n\times D}$. Let P be a projection operator for $\mathcal{A}$ and P^C its complement projection operator for $\mathcal{B}$, that is, $P^C W_i = W_i,\ PW_i = 0\ \forall\ W_i \in \mathcal{B}$ and $PW_j = W_j,\ P^C W_j = 0\ \forall\ W_j \in \mathcal{A}$. Let $R \in \mathrm{O}(L)$ be any rotation and PR a rotation that only acts on the $\mathcal{A}$ subspace. The operator $\Phi = PR + P^C$ is a rotation applied to all elements of $\mathcal{A}$ which leaves elements of $\mathcal{B}$ unchanged. Under its action, only terms in the cost between elements of $\mathcal{A}$ and $\mathcal{B}$ will change. It is straightforward to show that the terms in the cost that have both elements within $\mathcal{A}$ or both within $\mathcal{B}$ are constant since the rotation does not alter the relative pairwise angles.

For $W_i \in \mathcal{B}$, we can write down the terms in the L_2 cost which contain itself and elements from $\mathcal{A}PR$

$$\begin{aligned} C_{W_i}(\mathcal{A}\Phi) &= \sum_{W_j\in\mathcal{A}} (R^T P^T W_j^T W_i)^2 + (W_i^T W_j PR)^2 \\ &= \sum_{W_j\in\mathcal{A}} (R^T W_j^T W_i)^2 + (W_i^T W_j R)^2 \\ &= 2\sum_{W_j\in\mathcal{A}} \mathrm{Proj}_{W_j R}(W_i)^2 \\ &= 2|W_i|^2 \\ &= C_{W_i}(\mathcal{A}). \end{aligned} \tag{39}$$

Since the $W_j \in \mathcal{A}$ remain an orthonormal basis under a rotation, the sum of the projections-squared is the L_2 norm-squared of W_i which is constant. Since this is true for every $W_i \in \mathcal{B}$, the entire cost is constant under this transformation. This argument holds for any subset which forms an orthonormal basis and so all orthonormal subsets can rotate arbitrarily with respect to each other without changing the value of the L_2 cost, but the coherence of the matrix does depend on the transformation, Φ. This shows that the L_2 global minimum contains dictionaries with coherence $= 1$ and < 1 which can be continuously transformed into each other. ■

Appendix C. Additional figures

C.1. Extended Fig 2

Fig C1 is identical analysis as Fig 2 with all cost functions included.

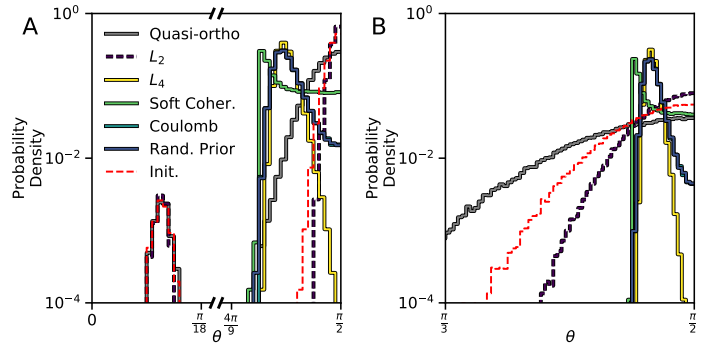


Figure C1: Coherence control costs have minima with varying coherence which can depend on initialization. Color legend is preserved across panels. For both panels a 2 times overcomplete dictionary with a data dimension of 64 was used. **A** Distribution of pairwise angles (log scale) obtained by numerically minimizing a subset of the coherence cost functions for the pathological dictionary initialization. Red dotted line indicates the initial distribution of pairwise angles. Note that the horizontal axis is broken at 10 and 80 degrees. **B** Angle distributions obtained (as in **A**) from a uniform random dictionary initialization. Note that the horizontal axis only includes $\frac{\pi}{3}$ to $\frac{\pi}{2}$.

C.2. Pairwise distributions for different powers

Fig C2 shows parwise angle distrubutions for cost functions based on powers of the difference between the gram matrix and the identity matrix for powers from 1 to 6 (skipping 2).

C.3. Local minima and saddle points are rare

Fig C3A shows the distribution of the mean-centered final cost values normalized by the average cost value at initialization

$$\text{Normalized Cost} = \frac{C_{\min} - \langle C_{\min}\rangle}{\langle C_{\text{init}}\rangle} \tag{40}$$

over 1000 uniform n-sphere initializations. All optimized cost functions are tighly bunched (1 part in $\sim 10^{-3}$). Minimizing the L_2 cost finds minima that are exactly equal to single precision floating point. This means that all costs are empirically converging to the same

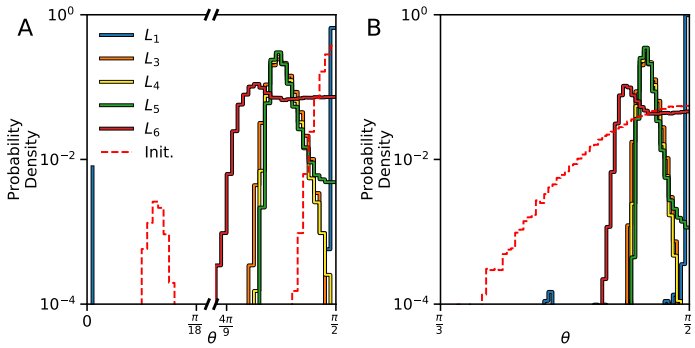


Figure C2: Coherence control costs based on powers of the difference between the Gram matrix and the identity matrix have highest coherence for powers 4 and 5. **A** Distribution of pairwise angles (log scale) obtained by numerically minimizing power-based coherence cost functions for the pathological dictionary initialization. Red dotted line indicates the initial distribution of pairwise angles. Note that the horizontal axis is broken at 10 and 80 degrees. **B** Angle distributions obtained (as in **A**) from a uniform random dictionary initialization. Note that the horizontal axis only includes 65 to 90 degrees.

value across many random re-initializations implying that they are not getting stuck in local minima or saddle points.

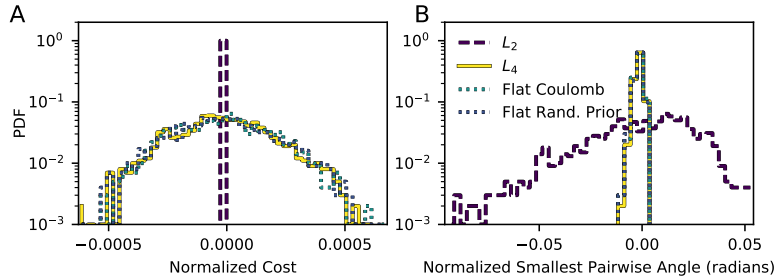


Figure C3: Local minima and saddle points do not impede optimization. **A** The distribution of the mean-centered final cost values normalized by the average cost value at initialization over 1000 uniform n-sphere initializations. **B** The distribution of the mean-centered final smallest pariwise angles over 1000 uniform n-sphere initializations.

Fig C3B shows the distribution of mean-centered final smallest pariwise angles

$$\text{Normalized Smallest Pairwise Angle} = \Theta_{\min} - \langle \Theta_{\min} \rangle \tag{41}$$

over 1000 uniform n-sphere initializations. Although all cost functions minimize to consistent values, they do not all minimize to consistent values of coherence (cosine of the smallest pairwise angle). The L_4, Flat Coulomb and Flat Random Prior costs all minimize to have tightly bunched coherence ($\sim .002$ radians, ~ 0.1 degrees) compared to the L_2 cost (~ 0.025 radians, ~ 1.5 degrees). This shows that all costs do not have problems with local minima or saddle points, but that minimizing the L_2 cost does not lead to consistently low coherence.

C.4. Extended Fig 4

Fig C4 is identical analysis as Fig 4 with all cost functions included.

C.5. Extended Fig 5

Fig C5 is identical analysis as Fig 5 with all cost functions included.

C.6. Supplemented Fig 3D.

Fig C6 is similar to Fig 3D for the Coulomb and Random Prior costs and their flattened versions.

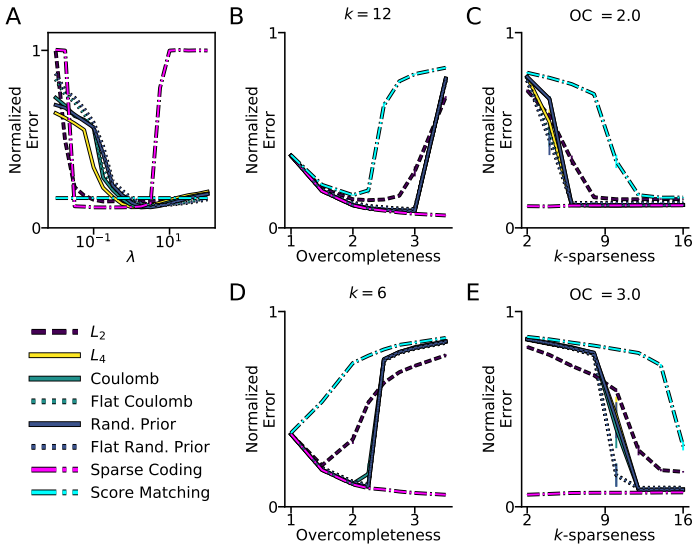


Figure C4: Coherence control costs do not all recover mixing matrices well. All ground truth mixing matrices were generated from the Soft Coherence cost and had a data dimension of 32. Color and line style legend are preserved across panels. **A** The normalized recovery error (see Section 3 for details) for a 2-times overcomplete mixing matrix and $k = 12$ as a function of the sparsity prior weight (λ). Since score matching does not have a λ parameter, it is plotted at a constant. **B** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of overcompleteness at $k = 12$. **C** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of k-sparseness at 2-times overcompleteness. **D, E** Same plots as **B** and **C** at a point where methods do not perform as well: $k = 6$ and 3-times overcomplete. In **B-E**, the L_4, Coulomb, Flattened Coulomb, Random, and Flattened Random lines are largely overlapping.

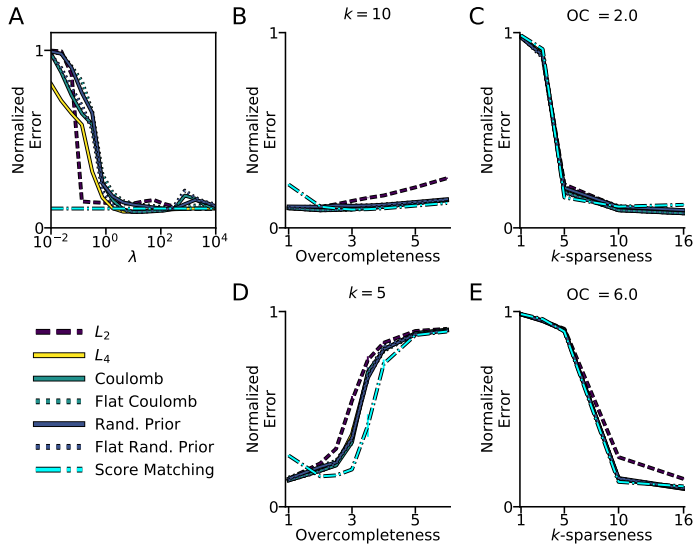


Figure C5: Coherence control costs do not all recover analysis matrices well. All ground truth analysis matrices were generated from the Soft Coherence cost and had a data dimension of 32. Color and line style legend are preserved across panels. **A** The normalized recovery error (see Section 3 for details) for a 2-times overcomplete analysis matrix and $k = 10$ as a function of the sparsity prior weight (λ). Since score matching does not have a λ parameter, it is plotted at a constant. **B** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of overcompleteness at $k = 10$. **C** Recovery performance ($\pm$ s.e.m., $n = 10$) at the best value of λ as a function of k-sparseness at 2-times overcompleteness. **D, E** Same plots as **B** and **C** at a point where methods do not perform as well: $k = 5$ and 6-times overcomplete. In **B-E**, the L_4, Coulomb, Flattened Coulomb, Random, and Flattened Random lines are largely overlapping.

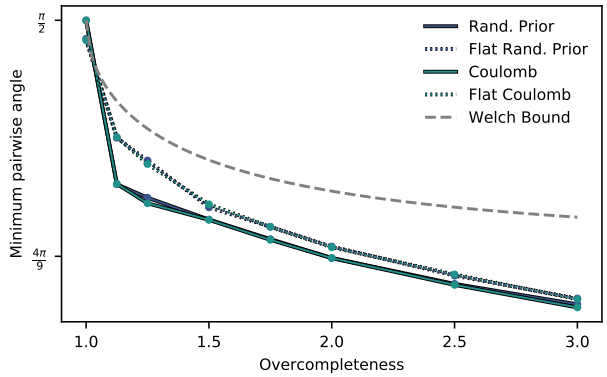


Figure C6: Quadratic terms dominate the minima of coherence control costs as a function of overcompleteness. The median minimum pairwise angle (arccosine of coherence) across 10 initializations is plotted as a function of overcompleteness for a dictionary with a data dimension of 32. The largest possible value (Welch Bound) is also shown as a function of overcompleteness.

References

Gautam Agarwal, Ian H Stevenson, Antal Berényi, Kenji Mizuseki, György Buzsáki, and Friedrich T Sommer. Spatially distributed local fields in the hippocampus encode rat position. *Science*, 344(6184):626–630, 2014.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? In *Advances in Neural Information Processing Systems*, pages 4261–4271, 2018.

Chenglong Bao, Yuhui Quan, and Hui Ji. A convergent incoherent dictionary learning algorithm for sparse coding. In *European Conference on Computer Vision*, pages 302–316. Springer, 2014.

Horace B Barlow. The ferrier lecture, 1980: Critical limiting factors in the design of the eye and visual cortex. *Proceedings of the Royal Society of London B: Biological Sciences*, 212(1186):1–34, 1981.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Anthony J Bell and Terrence J Sejnowski. The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

Andrew Brock, Theodore Lim, James Millar Ritchie, and Nicholas J Weston. Neural photo editing with introspective adversarial networks. In *5th International Conference on Learning Representations 2017*, 2017.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Nicole L Carlson, Vivienne L Ming, and Michael Robert DeWeese. Sparse codes for speech predict spectrotemporal receptive fields in the inferior colliculus. *PLoS Comput Biol*, 8 (7):e1002594, 2012.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

Il Yong Chun and Jeffrey A Fessler. Convolutional analysis operator learning: Acceleration, convergence, application, and neural networks. *arXiv preprint arXiv:1802.05584*, 2018.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3): 287–314, 1994.

Christine A Curcio and Kimberly A Allen. Topography of ganglion cells in human retina. *Journal of comparative Neurology*, 300(1):5–25, 1990.

Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing. *preprint*, 93(1):2, 2011.

Arnaud Delorme, Terrence Sejnowski, and Scott Makeig. Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis. *Neuroimage*, 34(4):1443–1449, 2007.

Michael R DeWeese, Tomáš Hromádka, and Anthony M Zador. Reliability and representational bandwidth in the auditory cortex. *Neuron*, 48(3):479–488, 2005.

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

Michael Elad, Peyman Milanfar, and Ron Rubinstein. Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947, 2007.

Matthew Fickus and Dustin G Mixon. Tables of the existence of equiangular tight frames. *arXiv preprint arXiv:1504.00253*, 2015.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 399–406. Omnipress, 2010.

Christopher J Hillar and Friedrich T Sommer. When can dictionary learning uniquely recover sparse data from subsamples? *IEEE Transactions on Information Theory*, 61 (11):6290–6297, 2015.

Jun-ichiro Hirayama, Takeshi Ogawa, and Aapo Hyvärinen. Unifying blind separation and clustering for resting-state eeg/meg functional connectivity analysis. *Neural computation*, 2015.

Stephen D Howard, A Robert Calderbank, and Stephen J Searle. A fast reconstruction algorithm for deterministic compressive sensing using second order reed-muller codes. In *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pages 11–15. IEEE, 2008.

Tao Hu, Cengiz Pehlevan, and Dmitri B Chklovskii. A hebbian/anti-hebbian network for online sparse dictionary learning derived from symmetric matrix factorization. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 613–619. IEEE, 2014.

Aapo Hyvarinen. Fast ica for noisy data using gaussian moments. In *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No. 99CH36349)*, volume 5, pages 57–61. IEEE, 1999.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. In *Journal of Machine Learning Research*, pages 695–709, 2005.

Aapo Hyvärinen and Mika Inki. Estimating overcomplete independent component bases for image windows. *Journal of Mathematical Imaging and Vision*, 17(2):139–152, 2002.

Aapo Hyvärinen and Urs Köster. Complex cell pooling and the statistics of natural images. *Network: Computation in Neural Systems*, 18(2):81–100, 2007.

Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural computation*, 9(7):1483–1492, 1997.

Aapo Hyvärinen, Razvan Cristescu, and Erkki Oja. A fast algorithm for estimating overcomplete ica bases for image windows. In *Neural Networks, 1999. IJCNN'99. International Joint Conference on*, volume 2, pages 894–899. IEEE, 1999.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2001.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–2017. URL `http://www.scipy.org/`. [Online; accessed 2017-03-15].

David J Klein, Peter König, and Konrad P Körding. Sparse spectrotemporal coding of sounds. *EURASIP Journal on Advances in Signal Processing*, 2003(7):902061, 2003.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Quoc V Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Y Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In *Advances in Neural Information Processing Systems*, pages 1017–1025, 2011.

Geneviève Leuba and Rudolf Kraftsik. Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual cortex from midgestation until old age. *Anatomy and embryology*, 190(4):351–366, 1994.

Michael S Lewicki and Bruno A Olshausen. Probabilistic framework for the adaptation and comparison of image codes. *JOSA A*, 16(7):1587–1601, 1999.

Michael S Lewicki and Terrence J Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000.

Jörg Lücke, Richard Turner, Maneesh Sahani, and Marc Henniges. Occlusive components analysis. In *Advances in Neural Information Processing Systems*, pages 1069–1077, 2009.

Boris Mailhé, Daniele Barchiesi, and Mark D Plumbley. Ink-svd: Learning incoherent dictionaries for sparse representations. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3573–3576. IEEE, 2012.

Jerry L Northern and Marion P Downs. *Hearing in children*. Lippincott Williams & Wilkins, 2002.

Bruno A Olshausen. Highly overcomplete sparse coding. In *IS&T/SPIE Electronic Imaging*, pages 86510S–86510S. International Society for Optics and Photonics, 2013.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Boaz Ophir, Michael Elad, Nancy Bertin, and Mark D Plumbley. Sequential minimal eigenvalues-an approach to analysis dictionary learning. In *2011 19th European Signal Processing Conference*, pages 1465–1469. IEEE, 2011.

Ignacio Ramırez, Federico Lecumberry, and Guillermo Sapiro. Sparse modeling with universal priors and learned incoherent dictionaries. Technical report, Citeseer, 2009.

Martin Rehn and Friedrich T Sommer. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of computational neuroscience*, 22(2):135–146, 2007.

Dario L Ringach. Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *Journal of neurophysiology*, 88(1):455–463, 2002.

Christopher J Rozell, Don H Johnson, Richard G Baraniuk, and Bruno A Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20 (10):2526–2563, 2008.

Ron Rubinstein, Tomer Peleg, and Michael Elad. Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model. *IEEE Transactions on Signal Processing*, 61(3): 661–677, 2013.

Christian D Sigg, Tomas Dikk, and Joachim M Buhmann. Learning dictionaries with bounded self-coherence. *IEEE Signal Processing Letters*, 19(12):861–864, 2012.

Steve Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998.

Evan C Smith and Michael S Lewicki. Efficient auditory coding. *Nature*, 439(7079):978–982, 2006.

H Spoendlin and A Schrott. Analysis of the human auditory nerve. *Hearing research*, 43 (1):25–38, 1989.

Thomas Strohmer and Robert W Heath Jr. Grassmannian frames with applications to coding and communication. *Applied and computational harmonic analysis*, 14(3):257–275, 2003.

Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E Hinton. Energy-based models for sparse overcomplete representations. *The Journal of Machine Learning Research*, 4: 1235–1260, 2003.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL `http://arxiv.org/abs/1605.02688`.

J Hans van Hateren and Arjen van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London B: Biological Sciences*, 265(1394):359–366, 1998.

Lloyd Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information theory*, 20(3):397–399, 1974.

Joel Zylberberg, Jason Timothy Murphy, and Michael Robert DeWeese. A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields. *PLoS computational biology*, 7(10):e1002250, 2011.

Journal of Machine Learning Research 20 (2019) 1-6 Submitted 11/18; Revised 9/19; Published 11/19

DataWig: Missing Value Imputation for Tables

Felix Bießmann FELIX.BIESSMANN@BEUTH-HOCHSCHULE.DE
Beuth University, Luxemburger Str. 10, 13353 Berlin (work done while at Amazon Research)

Tammo Rukat TAMMRUKA@AMAZON.COM
Phillipp Schmidt PHSCHMID@AMAZON.COM
Amazon Research, Krausenstr. 38, 10117 Berlin, Germany

Prathik Naidu PRATHIKN@STANFORD.EDU
Department of Computer Science, Stanford University
Stanford, CA 94305, USA (work done while at Amazon Research)

Sebastian Schelter SEBASTIAN.SCHELTER@NYU.EDU
Center for Data Science, New York University, New York, USA (work done while at Amazon Research)

Andrey Taptunov ANDREI.TAPTUNOV@GMAIL.COM
Snowflake, Stresemannstraße 123, 10963 Berlin (work done while at Amazon Research)

Dustin Lange LANGED@AMAZON.COM
Amazon Research, Krausenstr. 38, 10117 Berlin, Germany

David Salinas DAVID.SALINAS@NAVERLABS.COM
Naver Labs, 6 Chemin de Maupertuis, 38240 Meylan, France (work done while at Amazon Research)

Editor: Alexandre Gramfort

Abstract

With the growing importance of machine learning (ML) algorithms for practical applications, reducing data quality problems in ML pipelines has become a major focus of research. In many cases missing values can break data pipelines which makes completeness one of the most impactful data quality challenges. Current missing value imputation methods are focusing on numerical or categorical data and can be difficult to scale to datasets with millions of rows. We release `DataWig`, a robust and scalable approach for missing value imputation that can be applied to tables with heterogeneous data types, including unstructured text. `DataWig` combines deep learning feature extractors with automatic hyperparameter tuning. This enables users without a machine learning background, such as data engineers, to impute missing values with minimal effort in tables with more heterogeneous data types than supported in existing libraries, while requiring less glue code for feature engineering and offering more flexible modelling options. We demonstrate that `DataWig` compares favourably to existing imputation packages. Source code, documentation, and unit tests for this package are available at: `github.com/awslabs/datawig`

Keywords: missing value imputation, deep learning, heterogeneous data

1. Introduction

Machine learning (ML) algorithms have become a standard technology in production use cases. One of the main reasons for suboptimal predictive performance of such systems is low data

©2019 Felix Biessmann, Tammo Rukat, Phillipp Schmidt, Prathik Naidu, Sebastian Schelter, Andrey Taptunov, Dustin Lange.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/18-753.html`.

Data Type	Featurizers	Loss
Numerical	Normalization Neural Network	Regression
Categorical	Embeddings	Softmax
Text	Bag-of-Words LSTM	N/A

```
table = pandas.read_csv('products.csv')
missing = table[table['color'].isnull()]

# instantiate model and train imputer
model = SimpleImputer(
          input_columns=['description',
                         'product_type',
                         'size'],
          output_columns=['color'])
        .fit(table)

# impute missing values
imputed = model.predict(missing)
```

Figure 1: *Left:* Available featurizers and loss functions for different data types in `DataWig`. *Right:* Application example of `DataWig` API for the use case shown in Figure 2.

quality and one of the most frequent data quality problems are missing values. Imputation of missing values can help to increase data quality by filling gaps in training data. However automated and scalable imputations for tables with heterogeneous data types including free form text fields remains challenging. Here we present `DataWig`, a software package that aims at minimizing the effort required for missing value imputation in heterogeneous data sources. Most research in the field of imputation focuses on imputing missing values in *matrices*, that is imputation of numerical values from other numerical values (Mayer et al., 2019). Popular approaches include *k-nearest neighbors* (KNN) (Batista and Monard, 2003), *multivariate imputation by chained equations* (MICE) (Little and Rubin, 2002), *matrix factorization* (Koren et al., 2009; Mazumder et al., 2010; Troyanskaya et al., 2001) or deep learning methods (Gondara and Wang, 2017; Zhang et al., 2018; Mattei and Frellsen, 2019). While some recent work addresses imputation for more heterogeneous data types (Stekhoven and Bühlmann, 2012; Yoon et al., 2018; Nazabal et al., 2018), *heterogeneous* in those studies refers to binary, ordinal or categorical variables, which can be easily transformed into numerical representations. In practice also these simple transformations require glue code that can be difficult to adapt and maintain in a production setting. Writing such feature extraction code is out of scope for many engineers and can incur considerable technical debt on any data pipeline (Sculley et al., 2015; Schelter et al., 2018). We release `DataWig` to complement existing imputation libraries by an imputation solution for tables that contain not only numerical values or categorical values, but also more generic data types such as unstructured text. Extending the functionality of previous packages, `DataWig`'s imputation automatically selects from a number of feature extractors, including deep learning techniques, and learns all parameters in an end-to-end fashion using the symbolic API of Apache `mxnet` to ensure efficient execution on both CPUs and GPUs.

2. Imputation Model

The imputation model in `DataWig` is inspired by established approaches (van Buuren, 2018) and follows the approach of MICE, also referred to as *fully conditional specification*: for each to-be-imputed column (referred to as *output column*), the user can specify the columns which might contain useful information for imputation (referred to as *input columns*).

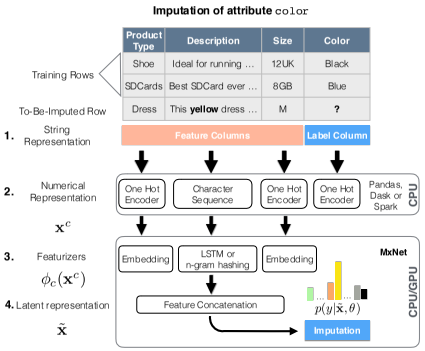


Figure 2: Imputation example on non-numerical data with deep learning.

Depending on its data type, each input column gets a dedicated *featurizer* denoted below as ϕ. Similarly, depending on the data type for the output column, `DataWig` uses a different loss function. The types of featurizers and loss functions currently available in `DataWig` are listed in Figure 1 (left). The code design enables users to extend these types easily to images or sequences. More formally, `DataWig` imputes values $\hat{y}_o = f(\tilde{\mathbf{x}}_{\mathcal{I}})$ in an output column o, where f refers to the imputation model learned on the observed values in column o and $\tilde{\mathbf{x}}_{\mathcal{I}}$ refers to the concatenation of the features extracted from all input columns $\tilde{\mathbf{x}}_{\mathcal{I}} = [\phi_1(\mathbf{x}^1), \phi_2(\mathbf{x}^2), \ldots, \phi_{C_I}(\mathbf{x}^{C_I})]$, see also Figure 2. Depending on the data type in the output column, f is fitted using either a regression or a cross-entropy loss. The API allows imputation of missing values in a table by simply passing in a `pandas` dataframe and specifying the input and output columns, see Figure 1 (right). Alternatively, all missing values in a dataframe can be imputed by calling `SimpleImputer.complete(df)`. Additionally `DataWig` has a number of features that help to automate end-to-end imputation for practitioners: The data types are detected using heuristics and the corresponding features are learned automatically during the training of the imputation model. All hyperparameters and neural architectures are optimized using random search (Bergstra and Bengio, 2012), which can be constrained to a specified time limit. Probabilistic model outputs are automatically calibrated on the validation set (Guo et al., 2017), and if requested explanations for the imputations can be computed for string input columns to better understand the imputations. Moreover, the model is equipped with functionality to compensate for label shift between the training and unlabelled production data using in the approach proposed by Lipton et al. (2018).

3. Evaluation

In Figure 3 we compare `DataWig` on *numerical missing value imputation* against three methods from the `fancyimpute` package (mean, KNN and matrix factorization) and two methods from the IterativeImputer of `sklearn` with the estimators RandomForestRegressor and LinearRegression, which are similar to the MissForest approach (Stekhoven and Bühlmann, 2012; van Buuren, 2018), and MICE with a linear model (Little and Rubin, 2002); iterative

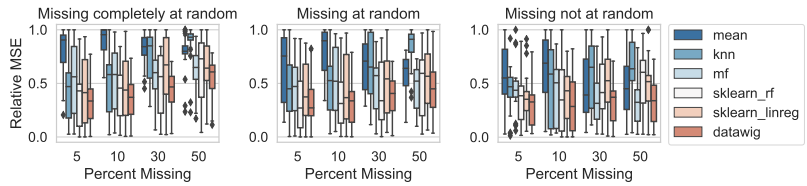


Figure 3: Comparison of imputation performance across several synthetic and real world data sets with varying amounts of missing data and missingness structure. Relative mean squared errors were normalized to the highest error in a condition.

imputation here means that 10 consecutive imputation rounds were performed for replacing the missing values in the input columns. All methods were evaluated on one synthetic linear and one synthetic non-linear problem and five real data sets available in `sklearn`. Values were discarded either completely at random, at random (conditioned on values in another randomly chosen column being in a random interval) or not at random (conditioned on values to be discarded). In Figure 3 the relative mean-squared error is shown, normalized to the highest MSE in a given condition. For `DataWig` the `SimpleImputer.complete` function with random search for hyperparameter tuning was used. For each baseline method, grid search was performed for hyperparameter optimization on a validation set, test errors were obtained on a separate test set, for details and unnormalized results see benchmarks github repository. We observe that `DataWig` compares favourably with other implementations for numeric imputation, even in the difficult missing-not-at-random condition. These experiments allow for a comparison of `DataWig` with existing packages designed for numeric data. For imputation with text data, standard numerical imputation methods cannot be used. When comparing `DataWig` with mode imputation and string matching (Dallachiesa et al., 2013) `DataWig` achieves a median F1-score of 60% across three tasks, imputation of the Wikipedia attributes *birth-place, genre* and *location*, with a simple n-gram model. Mode imputation reached a median F1-score of 0.7% and string matching 7.5% (Biessmann et al., 2018).

4. Conclusion

We present `DataWig`, a software package that enables practitioners such as data engineers to achieve state-of-the-art imputation results with minimal set up and maintenance. Our package complements the open source ecosystem by offering deep learning modules combined with neural architecture search and end-to-end optimization of the imputation pipeline, also for data types like free text fields. `DataWig` compares favorably to existing imputation approaches on numeric imputation problems, but also when imputing values in tables containing unstructured text. The software, unit tests, and all experiments are available under `github.com/awslabs/datawig`. While the present version of our software does not impute free form text or images, an interesting topic for future research is using generative models for these types of data building on recent advancements in neural missing value imputation (Zhang et al., 2018; Camino et al., 2019).

References

Gustavo Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

benchmarks github repository. `https://github.com/awslabs/datawig/blob/master/experiments/benchmarks.py`.

James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. URL `http://dl.acm.org/citation.cfm?id=2188395`.

Felix Biessmann, David Salinas, Sebastian Schelter, Philipp Schmidt, and Dustin Lange. "deep" learning for missing value imputation in tables with non-numerical data. In *International Conference on Information and Knowledge Management (CIKM)*, 2018.

Ramiro D. Camino, Christian A. Hammerschmidt, and Radu State. Improving Missing Data Imputation with Deep Generative Models. feb 2019. URL `http://arxiv.org/abs/1902.10666`.

Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed Elmagarmid, Ihab F Ilyas, Mourad Ouzzani, and Nan Tang. Nadeef: a commodity data cleaning system. In *ACM SIGMOD*, pages 541–552. ACM, 2013.

Lovedeep Gondara and Ke Wang. Multiple imputation using deep denoising autoencoders. *CoRR*, abs/1705.02737, 2017. URL `http://arxiv.org/abs/1705.02737`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning (ICML)*, 2017.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

Zachary C. Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and Correcting for Label Shift with Black Box Predictors. *International Conference on Machine Learning (ICML)*, 2018.

R. J. A. Little and D. B. Rubin. *Statistical analysis with missing data. 2nd ed.* Wiley-Interscience, Hoboken, NJ,, 2002.

Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning (ICML)*, 2019.

Imke Mayer, Julie Josse, Nicholas Tierney, and Nathalie Vialaneix. R-miss-tastic: a unified platform for missing values methods and workflows. art. arXiv:1908.04822, 2019.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010. URL `http://portal.acm.org/citation.cfm?id=1859931`.

A Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling Incomplete Heterogeneous Data using VAEs. 2018. URL `https://arxiv.org/pdf/1807.03653.pdf`.

Sebastian Schelter, Felix Biessmann, Tim Januschowski, David Salinas, Stephan Seufert, and Gyuri Szarvas. On challenges in machine learning model management. *IEEE Data Engineering Bulletin*, 41, 12 2018.

D Sculley, G Holt, D Golovin, E Davydov, T Phillips, D Ebner, V Chaudhary, M Young, and D Dennison. Hidden Technical Debt in Machine Learning Systems. *Neural Information Processing Systems (NeurIPS)*, 2015.

Daniel J Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

Olga G. Troyanskaya, Michael N. Cantor, Gavin Sherlock, Patrick O. Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

S. van Buuren. *Flexible Imputation of Missing Data. 2nd ed.* CRC/Chapman & Hall, 2018.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing Data Imputation using Generative Adversarial Nets. *International Conference on Machine Learning (ICML)*, 2018. URL `http://arxiv.org/abs/1806.02920`.

Hongbao Zhang, Pengtao Xie, and Eric P. Xing. Missing value imputation based on deep generative models. *CoRR*, abs/1808.01684, 2018. URL `http://arxiv.org/abs/1808.01684`.

Journal of Machine Learning Research 20 (2019) 1-49 Submitted 11/18; Revised 7/19; Published 11/19

New Convergence Aspects of Stochastic Gradient Algorithms

Lam M. Nguyen* LamNguyen.MLTD@ibm.com
IBM Research, Thomas J. Watson Research Center
Yorktown Heights, NY 10598, USA

Phuong Ha Nguyen* phuongha.ntu@gmail.com
Department of Electrical and Computer Engineering
University of Connecticut, Storrs, CT 06268, USA

Peter Richtárik peter.richtarik@kaust.edu.sa
Computer, Electrical and Math. Sciences and Engineering Division
King Abdullah University of Science and Technology, Thuwal, KSA

Katya Scheinberg ks2375@cornell.edu
School of Operations Research and Information Engineering
Cornell University, Ithaca, NY 14850, USA

Martin Takáč Takac.MT@gmail.com
Department of Industrial and Systems Engineering
Lehigh University, Bethlehem, PA 18015, USA

Marten van Dijk marten.van_dijk@uconn.edu
Department of Electrical and Computer Engineering
University of Connecticut, Storrs, CT 06268, USA

Editor: Inderjit Dhillon

Abstract

The classical convergence analysis of SGD is carried out under the assumption that the norm of the stochastic gradient is uniformly bounded. While this might hold for some loss functions, it is violated for cases where the objective function is strongly convex. In Bottou et al. (2018), a new analysis of convergence of SGD is performed under the assumption that stochastic gradients are bounded with respect to the true gradient norm. We show that for stochastic problems arising in machine learning such bound always holds; and we also propose an alternative convergence analysis of SGD with diminishing learning rate regime. We then move on to the asynchronous parallel setting, and prove convergence of Hogwild! algorithm in the same regime in the case of diminished learning rate. It is well-known that SGD converges if a sequence of learning rates $\{\eta_t\}$ satisfies $\sum_{t=0}^{\infty} \eta_t \to \infty$ and $\sum_{t=0}^{\infty} \eta_t^2 < \infty$. We show the convergence of SGD for strongly convex objective function without using bounded gradient assumption when $\{\eta_t\}$ is a diminishing sequence and $\sum_{t=0}^{\infty} \eta_t \to \infty$. In other words, we extend the current state-of-the-art class of learning rates satisfying the convergence of SGD.

Keywords: Stochastic Gradient Algorithms, Asynchronous Stochastic Optimization, SGD, Hogwild!, bounded gradient

. * Equally contributed. Corresponding author: Lam M. Nguyen.

©2019 Lam M. Nguyen, Phuong Ha Nguyen, Peter Richtárik, Katya Scheinberg, Martin Takáč, and Marten van Dijk.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-759.html.

1. Introduction

We are interested in solving the following stochastic optimization problem

$$\min_{w\in\mathbb{R}^d} \left\{F(w) = \mathbb{E}[f(w;\xi)]\right\}, \tag{1}$$

where ξ is a random variable obeying some distribution.

In the case of empirical risk minimization with a training set $\{(x_i, y_i)\}_{i=1}^n$, ξ_i is a random variable that is defined by a single random sample (x, y) pulled uniformly from the training set. Then, by defining $f_i(w) := f(w;\xi_i)$, empirical risk minimization reduces to

$$\min_{w\in\mathbb{R}^d} \left\{F(w) = \frac{1}{n}\sum_{i=1}^n f_i(w)\right\}. \tag{2}$$

Problem (2) arises frequently in supervised learning applications (Hastie et al., 2009). For a wide range of applications, such as linear regression and logistic regression, the objective function F is strongly convex and each f_i, $i \in [n]$, is convex and has Lipschitz continuous gradients (with Lipschitz constant L). Given a training set $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$, the ℓ_2-regularized least squares regression model, for example, is written as (2) with $f_i(w) \stackrel{\text{def}}{=} (\langle x_i, w\rangle - y_i)^2 + \frac{\lambda}{2}\|w\|^2$. The ℓ_2-regularized logistic regression for binary classification is written with $f_i(w) \stackrel{\text{def}}{=} \log(1 + \exp(-y_i\langle x_i, w\rangle)) + \frac{\lambda}{2}\|w\|^2$, $y_i \in \{-1, 1\}$. It is well established by now that solving this type of problem by gradient descent (GD) (Nesterov, 2004; Nocedal and Wright, 2006) may be prohibitively expensive and stochastic gradient descent (SGD) is thus preferable. Recently, a class of variance reduction methods (Le Roux et al., 2012; Defazio et al., 2014; Johnson and Zhang, 2013; Nguyen et al., 2017) has been proposed in order to reduce the computational cost. All these methods explicitly exploit the finite sum form of (2) and thus they have some disadvantages for very large scale machine learning problems and are not applicable to (1).

To apply SGD to the general form (1) one needs to assume existence of unbiased gradient estimators. This is usually defined as follows:

$$\mathbb{E}_\xi[\nabla f(w;\xi)] = \nabla F(w),$$

for any fixed w. Here we make an important observation: if we view (1) not as a general stochastic problem but as the expected risk minimization problem, where ξ corresponds to a random data sample pulled from a distribution, then (1) has an additional key property: for each realization of the random variable ξ, $f(w;\xi)$ is a convex function with Lipschitz continuous gradients. Notice that traditional analysis of SGD for general stochastic problem of the form (1) does not make any assumptions on individual function realizations. In this paper we derive convergence properties for SGD applied to (1) with these additional assumptions on $f(w;\xi)$ and also extend to the case when $f(w;\xi)$ are not necessarily convex.

Regardless of the properties of $f(w;\xi)$ we assume that F in (1) is strongly convex. We define the (unique) optimal solution of F as w_*.

Assumption 1 (μ-strongly convex) *The objective function $F : \mathbb{R}^d \to \mathbb{R}$ is a μ-strongly convex, i.e., there exists a constant $\mu > 0$ such that $\forall w, w' \in \mathbb{R}^d$,*

$$F(w) - F(w') \geq \langle \nabla F(w'), (w - w')\rangle + \frac{\mu}{2}\|w - w'\|^2. \tag{3}$$

It is well-known in literature (Nesterov, 2004; Bottou et al., 2018) that Assumption 1 implies

$$2\mu[F(w) - F(w_*)] \leq \|\nabla F(w)\|^2 \ , \ \forall w \in \mathbb{R}^d. \tag{4}$$

The classical theoretical analysis of SGD assumes that the *stochastic gradients are uniformly bounded*, i.e. there exists a finite (fixed) constant $\sigma < \infty$, such that

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq \sigma^2 \ , \ \forall w \in \mathbb{R}^d \tag{5}$$

(see e.g. Shalev-Shwartz et al. (2011); Nemirovski et al. (2009); Recht et al. (2011); Hazan and Kale (2014); Rakhlin et al. (2012), etc.). However, this assumption is clearly false if F is strongly convex. Specifically, under this assumption together with strong convexity, $\forall w \in \mathbb{R}^d$, we have

$$\begin{aligned} 2\mu[F(w) - F(w_*)] &\overset{(4)}{\leq} \|\nabla F(w)\|^2 = \|\mathbb{E}[\nabla f(w;\xi)]\|^2 \\ &\leq \mathbb{E}[\|\nabla f(w;\xi)\|^2] \overset{(5)}{\leq} \sigma^2. \end{aligned}$$

Hence,

$$F(w) \leq \frac{\sigma^2}{2\mu} + F(w_*) \ , \ \forall w \in \mathbb{R}^d.$$

On the other hand strong convexity and $\nabla F(w_*) = 0$ imply

$$F(w) \geq \mu\|w - w_*\|^2 + F(w_*) \ , \ \forall w \in \mathbb{R}^d.$$

The last two inequalities are clearly in contradiction with each other for sufficiently large $\|w - w_*\|^2$.

Let us consider the following example: $f_1(w) = \frac{1}{2}w^2$ and $f_2(w) = w$ with $F(w) = \frac{1}{2}(f_1(w) + f_2(w))$. Note that F is strongly convex, while individual realizations are not necessarily so. Let $w_0 = 0$, for any number $t \geq 0$, with probability $\frac{1}{2^t}$ the steps of SGD algorithm for all $i < t$ are $w_{i+1} = w_i - \eta_i$. This implies that $w_t = -\sum_{i=1}^t \eta_i$ and since $\sum_{i=1}^\infty \eta_i = \infty$ then $|w_t|$ can be arbitrarily large for large enough t with probability $\frac{1}{2^t}$. Noting that for this example, $\mathbb{E}[\|\nabla f(w_t;\xi)\|^2] = \frac{1}{2}w_t^2 + \frac{1}{2}$, we see that $\mathbb{E}[\|\nabla f(w_t;\xi)\|^2]$ can also be arbitrarily large.

Recently, in the review paper (Bottou et al., 2018), convergence of SGD for general stochastic optimization problem was analyzed under the following assumption: there exist constants M and N such that $\mathbb{E}[\|\nabla f(w_t;\xi_t)\|^2] \leq M\|\nabla F(w_t)\|^2 + N$, where w_t, $t \geq 0$, are generated by the SGD algorithm. This assumption does not contradict strong convexity, however, in general, constants M and N are unknown, while M is used to determine the learning rate η_t (see Bottou et al. (2018)). In addition, the rate of convergence of the SGD algorithm depends on M and N. In this paper we show that under the smoothness assumption on individual realizations $f(w,\xi)$ it is possible to derive the bound $\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq M_0[F(w) - F(w_*)] + N$ with specific values of M_0, and N for $\forall w \in \mathbb{R}^d$, which in turn implies the bound $\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq M\|\nabla F(w)\|^2 + N$ with specific M, by strong convexity of F. We also note that, in Moulines and Bach (2011), the convergence of SGD without bounded

gradient assumption is studied. We then use the new framework for the convergence analysis of SGD to analyze an asynchronous stochastic gradient method.

In Recht et al. (2011), an asynchronous stochastic optimization method called Hogwild! was proposed. Hogwild! algorithm is a parallel version of SGD, where each processor applies SGD steps independently of the other processors to the solution w which is shared by all processors. Thus, each processor computes a stochastic gradient and updates w without "locking" the memory containing w, meaning that multiple processors are able to update w at the same time. This approach leads to much better scaling of parallel SGD algorithm than a synchoronous version, but the analysis of this method is more complex. In Recht et al. (2011); Mania et al. (2017); De Sa et al. (2015) various variants of Hogwild! with a fixed step size are analyzed under the assumption that the gradients are bounded as in (5). In this paper, we extend our analysis of SGD to provide analysis of Hogwild! with diminishing step sizes and without the assumption on bounded gradients.

In Leblond et al. (2018) Hogwild! with fixed step size is analyzed without the bounded gradient assumption. We note that SGD with fixed step size only converges to a neighborhood of the optimal solution, while by analyzing the diminishing step size variant we are able to show convergence to the *optimal solution* with probability one (w.p.1). Both in Leblond et al. (2018) and in this paper, the version of Hogwild! with inconsistent reads and writes is considered.

It is well-known that SGD will converge if a sequence of learning rates $\{\eta_t\}$ satisfies the following conditions (1) $\sum_{t=0}^\infty \eta_t \to \infty$ and (2) $\sum_{t=0}^\infty \eta_t^2 < \infty$. As an important contribution of this paper, we show the convergence of SGD for strongly convex objective function without using bounded gradient assumption when $\{\eta_t\}$ is a diminishing sequence and $\sum_{t=0}^\infty \eta_t \to \infty$. In Moulines and Bach (2011), the authors also proved the convergence of SGD for $\{\eta_t = \mathcal{O}(1/t^q)\}, 0 < q \leq 1$, without using bounded gradient assumption and the second condition. Compared to Moulines and Bach (2011), we prove the convergence of SGD for $\{\eta_t = \mathcal{O}(1/t^q)\}$ which is $1/\mu$ times larger and our proposed class of learning rates satisfying the convergence of SGD is larger. Our proposed class of learning rates satisfying the convergence of SGD is larger than the current state-of-the art one.

We would like to highlight that this paper is originally from Nguyen et al. (2018) (Proceedings of the 35th International Conference on Machine Learning, 2018) but it presents a substantial extension by providing many new results for SGD and Hogwild!.

1.1. Contribution

We provide a new framework for the analysis of stochastic gradient algorithms in the strongly convex case under the condition of Lipschitz continuity of the individual function realizations, but **without requiring any bounds on the stochastic gradients**. Within this framework we have the following contributions:

- We prove the almost sure (w.p.1) convergence of SGD with diminishing step size. Our analysis provides a larger bound on the possible initial step size when compared to any previous analysis of convergence in expectation for SGD.

- We introduce a general recurrence for vector updates which has as its special cases (a) the Hogwild! algorithm with diminishing step sizes, where each update involves all

non-zero entries of the computed gradient, and (b) a position-based updating algorithm where each update corresponds to only one uniformly selected non-zero entry of the computed gradient.

- We analyze this general recurrence under inconsistent vector reads from and vector writes to shared memory (where individual vector entry reads and writes are atomic in that they cannot be interrupted by writes to the same entry) assuming that there exists a delay τ such that during the $(t+1)$-th iteration a gradient of a read vector w is computed which includes the aggregate of all the updates up to and including those made during the $(t-\tau)$-th iteration. In other words, τ controls to what extent past updates influence the shared memory.
  - Our upper bound for the expected convergence rate is $O(1/t)$, and its precise expression allows comparison of algorithms (a) and (b) described above.
  - For SGD we can improve this upper bound by a factor of 2 and also show that its initial step size can be larger.
  - We show that τ can be a function of t as large as $\sqrt{(t/lnt)(1-1/lnt)}$ without affecting the asymptotic behavior of the upper bound; we also determine a constant T_0 with the property that, for $t \geq T_0$, higher order terms containing parameter τ are smaller than the leading $O(1/t)$ term. We give intuition explaining why the expected convergence rate is not more affected by τ. Our experiments confirm our analysis.
  - We determine a constant T_1 with the property that, for $t \geq T_1$, the higher order term containing parameter $\|w_0 - w_*\|^2$ is smaller than the leading $O(1/t)$ term.
- All the above contributions generalize to the setting where we do not need to assume that the component functions $f(w;\xi)$ are convex in w.

Compared to Nguyen et al. (2018), we have following new results:

- We prove the almost sure (w.p.1) convergence of Hogwild! with a diminishing sequence of learning rates $\{\eta_t\}$.
- We prove the convergence of SGD for diminishing sequences of learning rates $\{\eta_t\}$ with condition $\sum_{t=0}^{\infty} \eta_t \to \infty$. In other words, we extend the current state-of-the-art class of learning rates satisfying the convergence of SGD.
- We prove the convergence of SGD for our extended class of learning rates in batch model.

1.2. Organization

We analyse the convergence rate of SGD in Section 2 and introduce the general recursion and its analysis in Section 3. Section 4 studies the convergence of SGD for our extended class of learning rates. Experiments are reported in Section 5.

Algorithm 1 Stochastic Gradient Descent (SGD) Method

Initialize: w_0
Iterate:
for $t = 0, 1, 2, \ldots$ **do**
  Choose a step size (i.e., learning rate) $\eta_t > 0$.
  Generate a random variable ξ_t.
  Compute a stochastic gradient $\nabla f(w_t;\xi_t)$.
  Update the new iterate $w_{t+1} = w_t - \eta_t \nabla f(w_t;\xi_t)$.
end for

2. New Framework for Convergence Analysis of SGD

We introduce SGD algorithm in Algorithm 1.

The sequence of random variables $\{\xi_t\}_{t\geq 0}$ is assumed to be i.i.d.[1] Let us introduce our key assumption that each realization $\nabla f(w;\xi)$ is an L-smooth function.

Assumption 2 (L-smooth) *$f(w;\xi)$ is L-smooth for every realization of ξ, i.e., there exists a constant $L > 0$ such that, $\forall w, w' \in \mathbb{R}^d$,*

$$\|\nabla f(w;\xi) - \nabla f(w';\xi)\| \leq L\|w - w'\|. \tag{6}$$

Assumption 2 implies that F is also L-smooth. Then, by a property of L-smooth functions in Nesterov (2004), we have, $\forall w, w' \in \mathbb{R}^d$,

$$F(w) \leq F(w') + \langle \nabla F(w'), (w - w')\rangle + \frac{L}{2}\|w - w'\|^2. \tag{7}$$

The following additional convexity assumption can be made, as it holds for many problems arising in machine learning.

Assumption 3 *$f(w;\xi)$ is convex for every realization of ξ, i.e., $\forall w, w' \in \mathbb{R}^d$,*

$$f(w;\xi) - f(w';\xi) \geq \langle \nabla f(w';\xi), (w - w')\rangle.$$

We first derive our analysis under Assumptions 2, and 3 and then we derive weaker results under only Assumption 2.

2.1. Convergence With Probability One

As discussed in the introduction, under Assumptions 2 and 3 we can now derive a bound on $\mathbb{E}\|\nabla f(w;\xi)\|^2$.

Lemma 1 *Let Assumptions 2 and 3 hold. Then, for $\forall w \in \mathbb{R}^d$,*

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \leq 4L[F(w) - F(w_*)] + N, \tag{8}$$

where $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$; ξ is a random variable, and $w_* = \arg\min_w F(w)$.

1. i.i.d. stands for independent and identically distributed.

Using Lemma 1 and Super Martingale Convergence Theorem from Bertsekas (2011) (Lemma 5 in Appendix A), we can provide the sufficient condition for almost sure convergence of Algorithm 1 in the strongly convex case without assuming any bounded gradients.

Theorem 1 (Sufficient conditions for almost sure convergence) *Let Assumptions 1, 2 and 3 hold. Consider Algorithm 1 with a stepsize sequence such that*

$$0 < \eta_t \leq \frac{1}{2L}\ ,\ \sum_{t=0}^{\infty} \eta_t = \infty \text{ and } \sum_{t=0}^{\infty} \eta_t^2 < \infty.$$

Then, the following holds w.p.1 (almost surely)

$$\|w_t - w_*\|^2 \to 0.$$

Note that the classical SGD proposed in Robbins and Monro (1951) has learning rate satisfying conditions

$$\sum_{t=0}^{\infty} \eta_t = \infty \text{ and } \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

However, the original analysis is performed under the bounded gradient assumption, as in (5). In Theorem 1, on the other hand, we do not use this assumption, but instead assume Lipschitz smoothness and convexity of the function realizations, which does not contradict the strong convexity of $F(w)$.

The following result establishes a sublinear convergence rate of SGD.

Theorem 2 *Let Assumptions 1, 2 and 3 hold. Let $E = \frac{2\alpha L}{\mu}$ with $\alpha = 2$. Consider Algorithm 1 with a stepsize sequence such that $\eta_t = \frac{\alpha}{\mu(t+E)} \leq \eta_0 = \frac{1}{2L}$. Then,*

$$\mathbb{E}[\|w_t - w_*\|^2] \leq \frac{4\alpha^2 N}{\mu^2} \frac{1}{(t - T + E)}$$

for

$$t \geq T = \frac{4L}{\mu} \max\{\frac{L\mu}{N}\|w_0 - w_*\|^2, 1\} - \frac{4L}{\mu},$$

where $N = 2\mathbb{E}[\|\nabla f(w_*; \xi)\|^2]$ and $w_* = \arg\min_w F(w)$.

2.2. Convergence Analysis without Convexity

In this section, we provide the analysis of Algorithm 1 without using Assumption 3, that is, $f(w; \xi)$ is not necessarily convex. We still do not need to impose the bounded stochastic gradient assumption, since we can derive an analogue of Lemma 1, albeit with worse constant in the bound.

Lemma 2 *Let Assumptions 1 and 2 hold. Then, for $\forall w \in \mathbb{R}^d$,*

$$\mathbb{E}[\|\nabla f(w; \xi)\|^2] \leq 4L\kappa[F(w) - F(w_*)] + N, \tag{9}$$

where $\kappa = \frac{L}{\mu}$ and $N = 2\mathbb{E}[\|\nabla f(w_*; \xi)\|^2]$; ξ is a random variable, and $w_* = \arg\min_w F(w)$.

Based on the proofs of Theorems 1 and 2, we can easily have the following two results (Theorems 3 and 4).

Theorem 3 (Sufficient conditions for almost sure convergence) *Let Assumptions 1 and 2 hold. Then, we can conclude the statement of Theorem 1 with the definition of the step size replaced by $0 < \eta_t \leq \frac{1}{2L\kappa}$ with $\kappa = \frac{L}{\mu}$.*

Theorem 4 *Let Assumptions 1 and 2 hold. Then, we can conclude the statement of Theorem 2 with the definition of the step size replaced by $\eta_t = \frac{\alpha}{\mu(t+E)} \leq \eta_0 = \frac{1}{2L\kappa}$ with $\kappa = \frac{L}{\mu}$ and $\alpha = 2$, and all other occurrences of L in E and T replaced by $L\kappa$.*

Remark 1 *By strong convexity of F, Lemma 2 implies $\mathbb{E}[\|\nabla f(w; \xi)\|^2] \leq 2\kappa^2\|\nabla F(w)\|^2 + N$, for $\forall w \in \mathbb{R}^d$, where $\kappa = \frac{L}{\mu}$ and $N = 2\mathbb{E}[\|\nabla f(w_*; \xi)\|^2]$. We can now substitute the value $M = 2\kappa^2$ into Theorem 4.7 in Bottou et al. (2018). We observe that the resulting initial learning rate in Bottou et al. (2018) has to satisfy $\eta_0 \leq \frac{1}{2L_F\kappa^2}$ while our results allows $\eta_0 = \frac{1}{2L\kappa}$. We notice that Bottou et al. (2018) only assumes that F has Lipschitz continuous gradients with Lipschitz constant L_F while we need the smoothness assumption on individual realizations. Therefore, L_F and L may be different. Both L_F and L values are hard to compare and L_F in Bottou et al. (2018) can potentially be much smaller, however, no general comparative statements can be made.*

By introducing Assumption 2, which holds for many ML problems, we are able to provide the values of M and N. Recall that under Assumption 3, our initial learning rate is $\eta_0 = \frac{1}{2L}$ (in Theorem 2). Thus Assumption 3 provides an improvement of the conditions on the learning rate.

3. Asynchronous Stochastic Optimization aka Hogwild!

Hogwild! (Recht et al., 2011) is an asynchronous stochastic optimization method where writes to and reads from vector positions in shared memory can be inconsistent (this corresponds to (13) as we shall see). However, as mentioned in Mania et al. (2017), for the purpose of analysis the method in Recht et al. (2011) performs single vector entry updates that are randomly selected from the non-zero entries of the computed gradient as in (12) (explained later) and requires the assumption of consistent vector reads together with the bounded gradient assumption to prove convergence. Both Mania et al. (2017) and De Sa et al. (2015) prove the same result for fixed step size based on the assumption of bounded stochastic gradients in the strongly convex case but now without assuming consistent vector reads and writes. In these works the fixed step size η must depend on σ from the bounded gradient assumption, however, one does not usually know σ and thus, we cannot compute a suitable η a-priori.

As claimed by the authors in Mania et al. (2017), they can eliminate the bounded gradient assumption in their analysis of Hogwild!, which however was only mentioned as a remark without proof. On the other hand, the authors of Leblond et al. (2018) formulate and prove, without the bounded gradient assumption, a precise theorem about the convergence rate of Hogwild! of the form

$$\mathbb{E}[\|w_t - w_*\|^2] \leq (1 - \rho)^t (2\|w_0 - w_*\|^2) + b,$$

where ρ is a function of several parameters but independent of the fixed chosen step size η and where b is a function of several parameters and has a linear dependency with respect to the fixed step size, i.e., $b = O(\eta)$.

In this section, we discuss the convergence of Hogwild! with **diminishing** stepsize where writes to and reads from vector positions in shared memory can be **inconsistent**. This is a slight modification of the original Hogwild! where the stepsize is fixed. In our analysis we also **do not use the bounded gradient assumption** as in Leblond et al. (2018). Moreover, (a) we focus on solving the **more general problem** in (1), while Leblond et al. (2018) considers the specific case of the "finite-sum" problem in (2), and (b) we show that our analysis generalizes to the **non-convex case** of the component functions, i.e., we do not need to assume functions $f(w;\xi)$ are convex (we only require $F(w) = \mathbb{E}[f(w;\xi)]$ to be strongly convex) as opposed to the assumption in Leblond et al. (2018).

3.1. Recursion

We first formulate a general recursion for w_t to which our analysis applies, next we will explain how the different variables in the recursion interact and describe two special cases, and finally we present pseudo code of the algorithm using the recursion.

The recursion explains which positions in w_t should be updated in order to compute w_{t+1}. Since w_t is stored in shared memory and is being updated in a possibly non-consistent way by multiple cores who each perform recursions, the shared memory will contain a vector w whose entries represent a mix of updates. That is, before performing the computation of a recursion, a core will first read w from shared memory, however, while reading w from shared memory, the entries in w are being updated out of order. The final vector $\hat{w}_t$ read by the core represents an aggregate of a mix of updates in previous iterations.

The general recursion is defined as follows: For $t \geq 0$,

$$w_{t+1} = w_t - \eta_t d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t), \tag{10}$$

where

- $\hat{w}_t$ represents the vector used in computing the gradient $\nabla f(\hat{w}_t;\xi_t)$ and whose entries have been read (one by one) from an aggregate of a mix of previous updates that led to w_j, $j \leq t$, and
- the $S_{u_t}^{\xi_t}$ are diagonal 0/1-matrices with the property that there exist real numbers d_ξ satisfying
$$d_\xi \mathbb{E}[S_u^\xi | \xi] = D_\xi, \tag{11}$$
where the expectation is taken over u and D_ξ is the diagonal 0/1 matrix whose 1-entries correspond to the non-zero positions in $\nabla f(w;\xi)$ in the following sense: The i-th entry of D_ξ's diagonal is equal to 1 if and only if there exists a w such that the i-th position of $\nabla f(w;\xi)$ is non-zero.

The role of matrix $S_{u_t}^{\xi_t}$ is that it filters which positions of gradient $\nabla f(\hat{w}_t;\xi_t)$ play a role in (10) and need to be computed. Notice that D_ξ represents the support of $\nabla f(w;\xi)$; by $|D_\xi|$ we denote the number of 1s in D_ξ, i.e., $|D_\xi|$ equals the size of the support of $\nabla f(w;\xi)$.

We will restrict ourselves to choosing (i.e., fixing a-priori) *non-empty* matrices S_u^ξ that "partition" D_ξ in D approximately "equally sized" S_u^ξ:

$$\sum_u S_u^\xi = D_\xi,$$

where each matrix S_u^ξ has either $\lfloor |D_\xi|/D \rfloor$ or $\lceil |D_\xi|/D \rceil$ ones on its diagonal. We uniformly choose one of the matrices $S_{u_t}^{\xi_t}$ in (10), hence, d_ξ equals the number of matrices S_u^ξ, see (11).

In other to explain recursion (10) we first consider two special cases. For $D = \bar{\Delta}$, where

$$\bar{\Delta} = \max_\xi \{|D_\xi|\}$$

represents the maximum number of non-zero positions in any gradient computation $f(w;\xi)$, we have that for all ξ, there are exactly $|D_\xi|$ diagonal matrices S_u^ξ with a single 1 representing each of the elements in D_ξ. Since $p_\xi(u) = 1/|D_\xi|$ is the uniform distribution, we have $\mathbb{E}[S_u^\xi|\xi] = D_\xi/|D_\xi|$, hence, $d_\xi = |D_\xi|$. This gives the recursion

$$w_{t+1} = w_t - \eta_t |D_\xi| [\nabla f(\hat{w}_t;\xi_t)]_{u_t}, \tag{12}$$

where $[\nabla f(\hat{w}_t;\xi_t)]_{u_t}$ denotes the u_t-th position of $\nabla f(\hat{w}_t;\xi_t)$ and where u_t is a uniformly selected position that corresponds to a non-zero entry in $\nabla f(\hat{w}_t;\xi_t)$.

At the other extreme, for $D = 1$, we have exactly one matrix $S_1^\xi = D_\xi$ for each ξ, and we have $d_\xi = 1$. This gives the recursion

$$w_{t+1} = w_t - \eta_t \nabla f(\hat{w}_t;\xi_t). \tag{13}$$

Recursion (13) represents Hogwild!. In a single-core setting where updates are done in a consistent way and $\hat{w}_t = w_t$ yields SGD.

Algorithm 2 gives the pseudo code corresponding to recursion (10) with our choice of sets S_u^ξ (for parameter D).

Algorithm 2 Hogwild! general recursion

Input: $w_0 \in \mathbb{R}^d$
for $t = 0, 1, 2, \ldots$ **in parallel do**
  read each position of shared memory w denoted by $\hat{w}_t$ **(each position read is atomic)**
  draw a random sample ξ_t and a random "filter" $S_{u_t}^{\xi_t}$
  for positions h where $S_{u_t}^{\xi_t}$ has a 1 on its diagonal **do**
    compute g_h as the gradient $\nabla f(\hat{w}_t;\xi_t)$ at position h
    add $\eta_t d_{\xi_t} g_h$ to the entry at position h of w in shared memory **(each position update is atomic)**
  end for
end for

3.2. Analysis

Besides Assumptions 1, 2, and for now 3, we assume the following assumption regarding a parameter τ, called the delay, which indicates which updates in previous iterations have certainly made their way into shared memory w.

Assumption 4 (Consistent with delay τ) *We say that shared memory is consistent with delay τ with respect to recursion (10) if, for all t, vector $\hat{w}_t$ includes the aggregate of the updates up to and including those made during the $(t-\tau)$-th iteration (where (10) defines the $(t+1)$-st iteration). Each position read from shared memory is atomic and each position update to shared memory is atomic (in that these cannot be interrupted by another update to the same position).*

In other words in the $(t+1)$-th iteration, $\hat{w}_t$ equals $w_{t-\tau}$ plus some subset of position updates made during iterations $t-\tau, t-\tau+1, \ldots, t-1$. We assume that there exists a constant delay τ satisfying Assumption 4.

3.3. Convergence With Probability One

Appendix D.5 proves the following theorem

Theorem 5 (Sufficient conditions for almost sure convergence for Hogwild!) *Let Assumptions 1, 2, 3 and 4 hold. Consider Hogwild! method described in Algorithm 2 with a stepsize sequence such that*

$$0 < \eta_t = \frac{1}{LD(2+\beta)(k+t)} < \frac{1}{4LD}, \beta > 0, k \geq 3\tau.$$

Then, the following holds w.p.1 (almost surely)

$$\|w_t - w_*\| \to 0.$$

3.4. Convergence in Expectation

Appendix D.2 proves the following theorem where

$$\bar{\Delta}_D \stackrel{\text{def}}{=} D \cdot \mathbb{E}[\lceil |D_\xi|/D \rceil].$$

Theorem 6 *Suppose Assumptions 1, 2, 3 and 4 and consider Algorithm 2 for sets S_u^ξ with parameter D. Let $\eta_t = \frac{\alpha_t}{\mu(t+E)}$ with $4 \leq \alpha_t \leq \alpha$ and $E = \max\{2\tau, \frac{4L\alpha D}{\mu}\}$. Then, the expected number of single vector entry updates after t iterations is equal to*

$$t' = t\bar{\Delta}_D/D,$$

and

$$\begin{aligned}
\mathbb{E}[\|\hat{w}_t - w_*\|^2] &\leq \tfrac{4\alpha^2 DN}{\mu^2}\tfrac{t}{(t+E-1)^2} + O\left(\tfrac{\ln t}{(t+E-1)^2}\right), \\
\mathbb{E}[\|w_t - w_*\|^2] &\leq \tfrac{4\alpha^2 DN}{\mu^2}\tfrac{t}{(t+E-1)^2} + O\left(\tfrac{\ln t}{(t+E-1)^2}\right),
\end{aligned}$$

where $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$ and $w_* = \arg\min_w F(w)$.

In terms of t', the expected number single vector entry updates after t iterations, $\mathbb{E}[\|\hat{w}_t - w_*\|^2]$ and $\mathbb{E}[\|w_t - w_*\|^2]$ are at most

$$\frac{4\alpha^2\bar{\Delta}_D N}{\mu^2}\frac{1}{t'} + O\left(\frac{\ln t'}{t'^2}\right).$$

Remark 2 *In (12) $D = \bar{\Delta}$, hence, $\lceil |D_\xi|/D \rceil = 1$ and $\bar{\Delta}_D = \bar{\Delta} = \max_\xi\{|D_\xi|\}$. In (13) $D = 1$, hence, $\bar{\Delta}_D = \mathbb{E}[|D_\xi|]$. This shows that the upper bound in Theorem 6 is better for (13) with $D = 1$. If we assume no delay, i.e. $\tau = 0$, in addition to $D = 1$, then we obtain SGD. Theorem 2 shows that, measured in t', we obtain the upper bound*

$$\frac{4\alpha_{SGD}^2\bar{\Delta}_D N}{\mu^2}\frac{1}{t'}$$

with $\alpha_{SGD} = 2$ as opposed to $\alpha \geq 4$.

With respect to parallelism, SGD assumes a single core, while (13) and (12) allow multiple cores. Notice that recursion (12) allows us to partition the position of the shared memory among the different processor cores in such a way that each partition can only be updated by its assigned core and where partitions can be read by all cores. This allows optimal resource sharing and could make up for the difference between $\bar{\Delta}_D$ for (12) and (13). We hypothesize that, for a parallel implementation, D equal to a fraction of $\bar{\Delta}$ will lead to best performance.

Remark 3 *Surprisingly, the leading term of the upper bound on the convergence rate is independent of delay τ. On one hand, one would expect that a more recent read which contains more of the updates done during the last τ iterations will lead to better convergence. When inspecting the second order term in the proof in Appendix D.2, we do see that a smaller τ (and/or smaller sparsity) makes the convergence rate smaller. That is, asymptotically t should be large enough as a function of τ (and other parameters) in order for the leading term to dominate.*

Nevertheless, in asymptotic terms (for larger t) the dependence on τ is not noticeable. In fact, Appendix D.4 shows that we may allow τ to be a monotonic increasing function of t with

$$\frac{2L\alpha D}{\mu} \leq \tau(t) \leq \sqrt{t \cdot L(t)},$$

where $L(t) = \frac{1}{\ln t} - \frac{1}{(\ln t)^2}$ (this will make $E = \max\{2\tau(t), \frac{4L\alpha D}{\mu}\}$ also a function of t). The leading term of the convergence rate does not change while the second order terms increase to $O(\frac{1}{t\ln t})$. We show that, for

$$t \geq T_0 = \exp[2\sqrt{\Delta}(1 + \frac{(L+\mu)\alpha}{\mu})],$$

where $\Delta = \max_i \mathbb{P}(i \in D_\xi)$ measures sparsity, the higher order terms that contain $\tau(t)$ (as defined above) are at most the leading term.

Our intuition behind this phenomenon is that for large τ, all the last τ iterations before the t-th iteration use vectors $\hat{w}_j$ with entries that are dominated by the aggregate of updates

that happened till iteration $t-\tau$. Since the average sum of the updates during the last τ iterations is equal to

$$-\frac{1}{\tau}\sum_{j=t-\tau}^{t-1}\eta_j d_{\xi_j} S_{u_j}^{\xi_j}\nabla f(\hat{w}_j;\xi_t) \tag{14}$$

and all $\hat{w}_j$ look alike in that they mainly represent learned information before the $(t-\tau)$-th iteration, (14) becomes an estimate of the expectation of (14), i.e.,

$$\sum_{j=t-\tau}^{t-1}\frac{-\eta_j}{\tau}\mathbb{E}[d_{\xi_j} S_{u_j}^{\xi_j}\nabla f(\hat{w}_j;\xi_t)] = \sum_{j=t-\tau}^{t-1}\frac{-\eta_j}{\tau}\nabla F(\hat{w}_j). \tag{15}$$

This looks like GD which in the strong convex case has convergence rate $\leq c^{-t}$ for some constant $c>1$. This already shows that larger τ could help convergence as well. However, estimate (14) has estimation noise with respect to (15) which explains why in this thought experiment we cannot attain c^{-t} but can only reach a much smaller convergence rate of e.g. $O(1/t)$ as in Theorem 6.

Experiments in Section 5 confirm our analysis.

Remark 4 *The higher order terms in the proof in Appendix D.2 show that, as in Theorem 2, the expected convergence rate in Theorem 6 depends on $\|w_0-w_*\|^2$. The proof shows that, for*

$$t \geq T_1 = \frac{\mu^2}{\alpha^2 ND}\|w_0-w_*\|^2,$$

the higher order term that contains $\|w_0-w_*\|^2$ is at most the leading term. This is comparable to T in Theorem 2 for SGD.

Remark 5 *Step size $\eta_t = \frac{\alpha_t}{\mu(t+E)}$ with $4\leq\alpha_t\leq\alpha$ can be chosen to be fixed during periods whose ranges exponentially increase. For $t+E\in[2^h,2^{h+1})$ we define $\alpha_t=\frac{4(t+E)}{2^h}$. Notice that $4\leq\alpha_t<8$ which satisfies the conditions of Theorem 6 for $\alpha=8$. This means that we can choose*

$$\eta_t = \frac{\alpha_t}{\mu(t+E)} = \frac{4}{\mu 2^h}$$

as step size for $t+E\in[2^h,2^{h+1})$. This choice for η_t allows changes in η_t to be easily synchronized between cores since these changes only happen when $t+E=2^h$ for some integer h. That is, if each core is processing iterations at the same speed, then each core on its own may reliably assume that after having processed $(2^h-E)/P$ iterations the aggregate of all P cores has approximately processed 2^h-E iterations. So, after $(2^h-E)/P$ iterations a core will increment its version of h to $h+1$. This will introduce some noise as the different cores will not increment their h versions at exactly the same time, but this only happens during a small interval around every $t+E=2^h$. This will occur rarely for larger h.

3.5. Convergence Analysis without Convexity

In Appendix D.3, we also show that the proof of Theorem 6 can easily be modified such that Theorem 6 with $E\geq\frac{4L\kappa\alpha D}{\mu}$ also holds in the non-convex case of the component functions, i.e., we do not need Assumption 3. Note that this case is not analyzed in Leblond et al. (2018).

Theorem 7 *Let Assumptions 1 and 2 hold. Then, we can conclude the statement of Theorem 6 with $E\geq\frac{4L\kappa\alpha D}{\mu}$ for $\kappa=\frac{L}{\mu}$.*

Theorem 8 (Sufficient conditions for almost sure convergence) *Let Assumptions 1 and 2 hold. Then, we can conclude the statement of Theorem 5 with the definition of the step size replaced by $0<\eta_t=\frac{1}{LD\kappa(2+\beta)(k+t)}$ with $\kappa=\frac{L}{\mu}$.*

4. Convergence of Large Stepsizes

In Robbins and Monro (1951), the authors proved the convergence of SGD for step size sequences $\{\eta_t\}$ satisfying conditions

$$\sum_{t=0}^{\infty}\eta_t=\infty \text{ and } \sum_{t=0}^{\infty}\eta_t^2<\infty.$$

In Moulines and Bach (2011), the authors studied the expected convergence rates for another class of step sizes of $\mathcal{O}(1/t^p)$ where $0<p\leq 1$. This class has many large step sizes in comparison with Robbins and Monro (1951). For example $\eta_t=\mathcal{O}(1/t^p)$ does not satisfy the second condition (i.e., $\sum_{t=0}^{\infty}\eta_t^2\to\infty$) where $0<p<1/2$. In this section, we prove that SGD will converge without using bounded gradient assumption if $\{\eta_t\}$ is a diminishing sequence and $\sum_{t=0}^{\infty}\eta_t\to\infty$. Compared to Moulines and Bach (2011), we prove the convergence of SGD for step sizes $\eta_t=\mathcal{O}(1/t^q)$ which is $1/\mu$ times larger. Our proposed class is much larger than the classes in Robbins and Monro (1951) and Moulines and Bach (2011).

4.1. Convergence of Large Stepsizes

The proofs of all theorems and lemmas in this subsection are provided in Appendix D.6.

Theorem 9 *Let Assumptions 1, 2, and 3 hold. Consider Algorithm 1 with a step size sequence such that: $\eta_t\leq\frac{1}{2L}$, $\eta_t\to 0$, $\frac{d}{dt}\eta_t\leq 0$ and $\sum_{t=0}^{\infty}\eta_t\to\infty$. Then,*

$$\mathbb{E}[\|w_{t+1}-w_*\|^2]\to 0.$$

Theorem 9 only discusses about the convergence of SGD for the given step size sequence $\{\eta_t\}$ above. The expected convergence rate of SGD with the setup in Theorem 9 is analysed in Theorem 10.

Theorem 10 *Let Assumptions 1, 2, and 3 hold. Consider Algorithm 1 with a step size sequence such that $\eta_t\leq\frac{1}{2L}$, $\eta_t\to 0$, $\frac{d}{dt}\eta_t\leq 0$, and $\sum_{t=0}^{\infty}\eta_t\to\infty$. Then,*

$$\begin{aligned}\mathbb{E}[\|w_{t+1}-w_*\|^2] \leq{}& N\exp(n(0))2n(M^{-1}(\ln[\frac{n(t+1)}{n(0)}]+M(t+1)))\\ &+\exp(-M(t+1))[\exp(M(1))n^2(0)N+\mathbb{E}[\|w_0-w_*\|^2]],\end{aligned}$$

where $n(t)=\mu\eta_t$ and $M(t)=\int_{x=0}^{t}n(x)dx$.

The upper bound in Theorem 10 can be interpreted as being approximately equivalent to $\mathbb{E}[\|w_t - w_*\|^2] \leq Un(t-\Delta) + V\exp(-M(t))$ where $U = N\exp(n(0))$, $V = \exp(M(1))n^2(0)N + \mathbb{E}[\|w_0 - w_*\|^2]$ and Δ is a delay computed from $\ln[\frac{n(t+1)}{n(0)}]$. Since $n(t)$ decreases and $M(t)$ increases when t approaches to infinity, $\mathbb{E}[\|w_t - w_*\|^2]]$ decreases in the same way as $n(t)$ to 0, except for some delay Δ.

As shown in (53) (see also Appendix D.6), we have

$$\mathbb{E}[\|w_t - w_*\|^2] \leq AC(t) + B\exp(-M(t)),$$

where A and B are constants and $C(t)$ is defined in (16) below. We show that an alternative proof for the convergence of SGD with the setup above based on the study of $C(t)$ can be developed.

Lemma 3 *Let*

$$C(t) = \exp(-M(t)) \int_{x=0}^{t} \exp(M(x))n(x)^2 dx, \tag{16}$$

where $\frac{d}{dx}M(x) = n(x)$ *with function* $n(x)$ *satisfying the following conditions:*

1. $\frac{d}{dx}n(x) < 0$,
2. $\frac{d}{dx}n(x)$ *is continuous.*

Then, there is a moment T *such that for all* $t > T$, $C(t) > n(t)$.

Proof We take the derivative of $C(t)$, i.e.,

$$\begin{aligned}\frac{d}{dt}C(t) &= -\exp(-M(t))n(t)\int_{x=0}^{t}\exp(M(x))n(x)^2dx + \exp(-M(t))\exp(M(t))n(t)^2 \\ &= n(t)[n(t) - C(t)]\end{aligned}$$

This shows that

$$C(t) \text{ is decreasing if and only if } C(t) > n(t).$$

Initially $C(0) = 0$ and $n(0) > 0$, hence, $C(t)$ starts increasing from $t \geq 0$. Since $n(t)$ decreases for all $t \geq 0$, we know that there must exist a first cross over point x:

- There exists a value x such that $C(t)$ increases for $0 \leq t < x$, and
- $C(x) = n(x)$ with derivative $dC(t)/dt|_{t=x} = 0$.

Since $n(x)$ has a derivative < 0, we know that $C(t) > n(t)$ immediately after x. Suppose that $C(y) = n(y)$ for some $y > x$ with $C(t) > n(t)$ for $x < t < y$. This implies that $dC(t)/dt|_{t=y} = 0$ and since $dC(t)/dt$ is continuous

$$C(y-\epsilon) = C(y) + O(\epsilon^2).$$

Also,

$$n(y-\epsilon) = n(y) - \epsilon dn(t)/dt|_{t=y} + O(\epsilon^2).$$

Since $dn(t)/dt|_{t=y} < 0$, we know that there exists an ϵ small enough (close to 0) such that

$$C(y-\epsilon) < n(y-\epsilon).$$

This contradicts $C(t) > n(t)$ for $x < t < y$. We conclude that there does not exist a $y > x$ such that $C(y) = n(y)$:

- For $t > x$, $C(t) > n(t)$ and $C(t)$ is strictly decreasing.

We conclude that for any given $n(t)$, there exists a time T such that $C(t) < n(t)$ for all $t \in [0, T)$, $C(T) = n(T)$ after which $C(t) > n(t)$ when $t \in (T, \infty]$. Note that $C(t)$ is always bigger then zero. ■

As proved above, $C(t)$ decreases for $t > T$. In addition to this we note that $C(t)$ converges to zero when t goes to infinity (see the proof of Theorem 9 in Appendix D.6). In addition to $C(t) \to 0$ when $t \to \infty$, also $\exp(-M(t)) \to 0$ when $t \to \infty$ because $\sum_{t=0}^{\infty} n(t) \to \infty$. Based on these two results we conclude that $\mathbb{E}[\|w_t - w_*\|^2] \to 0$ when $t \to \infty$. This is an alternative proof for the convergence of SGD as shown in Theorem 9.

Theorem 11 *Among all stepsizes* $\eta_{q,t} = 1/(K+t)^q$ *where* $q > 0$, K *is a constant such that* $\eta_{q,t} \leq \frac{1}{2L}$, *SGD algorithm enjoys the fastest convergence with stepsize* $\eta_{1,t} = 1/(2L+t)$.

4.2. Convergence of Large Stepsizes in Batch Mode

We define

$$\mathcal{F}_t = \sigma(w_0, \xi_0', u_0, \ldots, \xi_{t-1}', u_{t-1}),$$

where

$$\xi_i' = (\xi_{i,1}, \ldots, \xi_{i,k_i}).$$

We consider the following general algorithm with the following gradient updating rule:

$$w_{t+1} = w_t - \eta_t d_{\xi_t'} S_{u_t}^{\xi_t'} \nabla f(w_t; \xi_t'), \tag{17}$$

where $f(w_t; \xi_t') = \frac{1}{k_t}\sum_{i=1}^{k_t} f(w_t; \xi_{t,i})$.

Theorem 12 *Let Assumptions 1, 2 and 3 hold,* $\{\eta_t\}$ *is a diminishing sequence with conditions* $\sum_{t=0}^{\infty}\eta_t \to \infty$ *and* $0 < \eta_t \leq \frac{1}{2LD}$ *for all* $t \geq 0$. *Then, the sequence* $\{w_t\}$ *converges to* w_* *where*

$$w_{t+1} = w_t - \eta_t d_{\xi_t'} S_{u_t}^{\xi_t'} \nabla f(w_t; \xi_t').$$

The proof of Theorem 12 is provided in Appendix D.7.

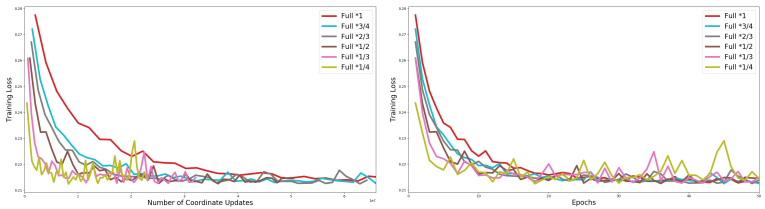


Figure 1: ijcnn1 data set for different fraction of non-zero set

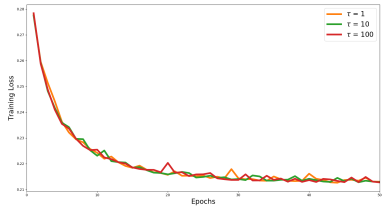


Figure 2: ijcnn1 data set for different τ with the whole non-zero set

5. Numerical Experiments

For our numerical experiments, we consider the finite sum minimization problem in (2). We consider ℓ_2-regularized logistic regression problems with

$$f_i(w) = \log(1 + \exp(-y_i \langle x_i, w \rangle)) + \frac{\lambda}{2}\|w\|^2,$$

where the penalty parameter λ is set to $1/n$, a widely-used value in literature (Le Roux et al., 2012).

We conducted experiments on a single core for Algorithm 2 on two popular data sets `ijcnn1` ($n = 91,701$ training data) and `covtype` ($n = 406,709$ training data) from LIBSVM (Chang and Lin, 2011) data sets. Since we are interested in the expected convergence rate with respect to the number of iterations, respectively number of single position vector updates, we do not need a parallelized multi-core simulation to confirm our analysis. The impact of efficient resource scheduling over multiple cores leads to a performance improvement complementary to our analysis of (10) (which, as discussed, lends itself for an efficient parallelized implementation). We experimented with 10 runs and reported the average results. We choose the step size based on Theorem 6, i.e, $\eta_t = \frac{4}{\mu(t+E)}$ and $E = \max\{2\tau, \frac{16LD}{\mu}\}$. For each fraction $v \in \{1, 3/4, 2/3, 1/2, 1/3, 1/4\}$ we performed the following experiment: In Algorithm 2 we choose each "filter" matrix $S_{u_t}^{\xi_t}$ to correspond with a random subset of size $v|D_{\xi_t}|$ of the non-zero positions of D_{ξ_t} (i.e., the support of the gradient corresponding to ξ_t). In addition we use $\tau = 10$. For the two data sets,

Figures 1 and 3 plot the training loss for each fraction with $\tau = 10$. The top plots have t', the number of coordinate updates, for the horizontal axis. The bottom plots have the

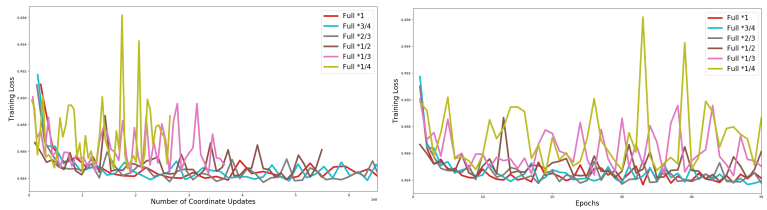


Figure 3: covtype data set for different fraction of non-zero set

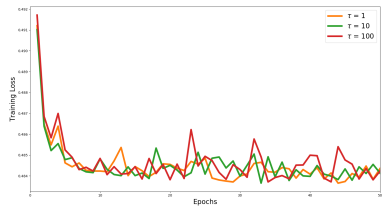


Figure 4: covtype data set for different τ with the whole non-zero set

number of epochs, each epoch counting n iterations, for the horizontal axis. The results show that each fraction shows a sublinear expected convergence rate of $O(1/t')$; the smaller fractions exhibit larger deviations but do seem to converge faster to the minimum solution.

In Figures 2 and 4, we show experiments with different values of $\tau \in \{1, 10, 100\}$ where we use the whole non-zero set of gradient positions (i.e., $v = 1$) for the update. Our analysis states that, for $t = 50$ epochs times n iterations per epoch, τ can be as large as $\sqrt{t \cdot L(t)} = 524$ for `ijcnn1` and 1058 for `covtype`. The experiments indeed show that $\tau \leq 100$ has little effect on the expected convergence rate.

6. Conclusion

We have provided the analysis of stochastic gradient algorithms with diminishing step size in the strongly convex case under the condition of Lipschitz continuity of the individual function realizations, but without requiring any bounds on the stochastic gradients. We showed almost sure convergence of SGD and provided sublinear upper bounds for the expected convergence rate of a general recursion which includes Hogwild! for inconsistent reads and writes as a special case. We also provided new intuition which will help understanding convergence as observed in practice.

Acknowledgement

Phuong Ha Nguyen and Marten van Dijk were supported in part by AFOSR MURI under award number FA9550-14-1-0351. Katya Scheinberg was partially supported by NSF Grants

CCF 16-18717 and CCF 17-40796. Martin Takáč was partially supported by the NSF Grant CCF-1618717, CMMI-1663256 and CCF-1740796.

References

Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

Christopher M De Sa, Ce Zhang, Kunle Olukotun, and Christopher Ré. Taming the wild: A unified analysis of hogwild-style algorithms. In *Advances in neural information processing systems*, pages 2674–2682, 2015.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, 2nd edition, 2009.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Nicolas Le Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in neural information processing systems*, pages 2663–2671, 2012.

Rémi Leblond, Fabian Pedregosa, and Simon Lacoste-Julien. Improved asynchronous parallel optimization analysis for stochastic incremental methods. *The Journal of Machine Learning Research*, 19(1):3140–3207, 2018.

Horia Mania, Xinghao Pan, Dimitris Papailiopoulos, Benjamin Recht, Kannan Ramchandran, and Michael I Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017.

Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, 2004. ISBN 1-4020-7553-7.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621, 2017.

Lam M. Nguyen, Phuong Ha Nguyen, Marten van Dijk, Peter Richtarik, Katya Scheinberg, and Martin Takac. SGD and Hogwild! convergence without the bounded gradients assumption. In *Proceedings of the 35th International Conference on Machine Learning-Volume 80*, pages 3747–3755, 2018.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1571–1578, 2012.

Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical programming*, 127(1):3–30, 2011.

Appendix A. Review of Useful Theorems

Lemma 4 (Generalization of the result in Johnson and Zhang (2013)) *Let Assumptions 2 and 3 hold. Then, $\forall w \in \mathbb{R}^d$,*

$$\mathbb{E}[\|\nabla f(w;\xi) - \nabla f(w_*;\xi)\|^2] \le 2L[F(w) - F(w_*)], \tag{18}$$

where ξ is a random variable, and $w_* = \arg\min_w F(w)$.

Lemma 5 (Bertsekas (2011)) *Let Y_k, Z_k, and W_k, $k = 0, 1, \ldots$, be three sequences of random variables and let $\{\mathcal{F}_k\}_{k\ge 0}$ be a filtration, that is, σ-algebras such that $\mathcal{F}_k \subset \mathcal{F}_{k+1}$ for all k. Suppose that:*

- *The random variables Y_k, Z_k, and W_k are nonnegative, and $\mathcal{F}_k$-measurable.*
- *For each k, we have $\mathbb{E}[Y_{k+1}|\mathcal{F}_k] \le Y_k - Z_k + W_k$.*
- *There holds, w.p.1,*

$$\sum_{k=0}^{\infty} W_k < \infty.$$

Then, we have, w.p.1,

$$\sum_{k=0}^{\infty} Z_k < \infty \text{ and } Y_k \to Y \ge 0.$$

Appendix B. Proofs of Lemmas 1 and 2

The proofs of Lemmas 1 and 2 can be done in the following sub-sections.

B.1. Proof of Lemma 1

Lemma 1. *Let Assumptions 2 and 3 hold. Then, for $\forall w \in \mathbb{R}^d$,*

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \le 4L[F(w) - F(w_*)] + N,$$

where $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$; ξ is a random variable, and $w_* = \arg\min_w F(w)$.

Proof Note that

$$\|a\|^2 = \|a - b + b\|^2 \le 2\|a - b\|^2 + 2\|b\|^2, \tag{19}$$

$$\Rightarrow \frac{1}{2}\|a\|^2 - \|b\|^2 \le \|a - b\|^2. \tag{20}$$

Hence,

$$\begin{aligned}\frac{1}{2}\mathbb{E}[\|\nabla f(w;\xi)\|^2] - \mathbb{E}[\|\nabla f(w_*;\xi)\|^2] &= \mathbb{E}\left[\frac{1}{2}\|\nabla f(w;\xi)\|^2 - \|\nabla f(w_*;\xi)\|^2\right] \\ &\overset{(20)}{\le} \mathbb{E}[\|\nabla f(w;\xi) - \nabla f(w_*;\xi)\|^2] \\ &\overset{(18)}{\le} 2L[F(w) - F(w_*)] \end{aligned} \tag{21}$$

Therefore,

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \overset{(19)(21)}{\le} 4L[F(w) - F(w_*)] + 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2].$$

■

B.2. Proof of Lemma 2

Lemma 2. *Let Assumptions 1 and 2 hold. Then, for $\forall w \in \mathbb{R}^d$,*

$$\mathbb{E}\|\nabla f(w;\xi)\|^2 \le 4L\kappa[F(w) - F(w_*)] + N,$$

where $\kappa = \frac{L}{\mu}$ and $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$; ξ is a random variable, and $w_* = \arg\min_w F(w)$.

Proof Analogous to the proof of Lemma 1, we have

Hence,

$$\begin{aligned}\frac{1}{2}\mathbb{E}[\|\nabla f(w;\xi)\|^2] - \mathbb{E}[\|\nabla f(w_*;\xi)\|^2] &= \mathbb{E}\left[\frac{1}{2}\|\nabla f(w;\xi)\|^2 - \|\nabla f(w_*;\xi)\|^2\right] \\ &\overset{(20)}{\le} \mathbb{E}[\|\nabla f(w;\xi) - \nabla f(w_*;\xi)\|^2] \\ &\overset{(6)}{\le} L^2\|w - w_*\|^2 \\ &\overset{(3)}{\le} \frac{2L^2}{\mu}[F(w) - F(w_*)] = 2L\kappa[F(w) - F(w_*)]. \end{aligned} \tag{22}$$

Therefore,

$$\mathbb{E}[\|\nabla f(w;\xi)\|^2] \overset{(19)(22)}{\le} 4L\kappa[F(w) - F(w_*)] + 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2].$$

■

Appendix C. Analysis for Algorithm 1

In this Section, we provide the analysis of Algorithm 1 under Assumptions 1, 2, and 3.

We note that if $\{\xi_i\}_{i\ge 0}$ are i.i.d. random variables, then $\mathbb{E}[\|\nabla f(w_*;\xi_0)\|^2] = \cdots = \mathbb{E}[\|\nabla f(w_*;\xi_t)\|^2]$. We have the following results for Algorithm 1.

Theorem 1 (Sufficient condition for almost sure convergence). *Let Assumptions 1, 2 and 3 hold. Consider Algorithm 1 with a stepsize sequence such that*

$$0 < \eta_t \le \frac{1}{2L}\ ,\ \sum_{t=0}^{\infty}\eta_t = \infty \text{ and } \sum_{t=0}^{\infty}\eta_t^2 < \infty.$$

Then, the following holds w.p.1 (almost surely)

$$\|w_t - w_*\|^2 \to 0.$$

Proof Let $\mathcal{F}_t = \sigma(w_0, \xi_0, \ldots, \xi_{t-1})$ be the σ-algebra generated by $w_0, \xi_0, \ldots, \xi_{t-1}$, i.e., $\mathcal{F}_t$ contains all the information of $w_0, \ldots, w_t$. Note that $\mathbb{E}[\nabla f(w_t; \xi_t)|\mathcal{F}_t] = \nabla F(w_t)$. By Lemma 1, we have

$$\mathbb{E}[\|\nabla f(w_t; \xi_t)\|^2|\mathcal{F}_t] \leq 4L[F(w_t) - F(w_*)] + N, \tag{23}$$

where $N = 2\mathbb{E}[\|\nabla f(w_*; \xi_0)\|^2] = \cdots = 2\mathbb{E}[\|\nabla f(w_*; \xi_t)\|^2]$ since $\{\xi_i\}_{i\geq 0}$ are i.i.d. random variables. Note that $w_{t+1} = w_t - \eta_t \nabla f(w_t; \xi_t)$. Hence,

$$\begin{aligned}
\mathbb{E}[\|w_{t+1} - w_*\|^2|\mathcal{F}_t] &= \mathbb{E}[\|w_t - \eta_t \nabla f(w_t; \xi_t) - w_*\|^2|\mathcal{F}_t] \\
&= \|w_t - w_*\|^2 - 2\eta_t \langle \nabla F(w_t), (w_t - w_*)\rangle + \eta_t^2 \mathbb{E}[\|\nabla f(w_t; \xi_t)\|^2|\mathcal{F}_t] \\
&\overset{(3)(23)}{\leq} \|w_t - w_*\|^2 - \mu\eta_t \|w_t - w_*\|^2 - 2\eta_t[F(w_t) - F(w_*)] \\
&+ 4L\eta_t^2[F(w_t) - F(w_*)] + \eta_t^2 N \\
&= \|w_t - w_*\|^2 - \mu\eta_t \|w_t - w_*\|^2 - 2\eta_t(1 - 2L\eta_t)[F(w_t) - F(w_*)] + \eta_t^2 N \\
&\leq \|w_t - w_*\|^2 - \mu\eta_t \|w_t - w_*\|^2 + \eta_t^2 N.
\end{aligned}$$

The last inequality follows since $0 < \eta_t \leq \frac{1}{2L}$. Therefore,

$$\mathbb{E}[\|w_{t+1} - w_*\|^2|\mathcal{F}_t] \leq \|w_t - w_*\|^2 - \mu\eta_t \|w_t - w_*\|^2 + \eta_t^2 N. \tag{24}$$

Since $\sum_{t=0}^{\infty} \eta_t^2 N < \infty$, we could apply Lemma 5. Then, we have w.p.1,

$$\|w_t - w_*\|^2 \to W \geq 0,$$
$$\text{and } \sum_{t=0}^{\infty} \mu\eta_t \|w_t - w_*\|^2 < \infty.$$

We want to show that $\|w_t - w_*\|^2 \to 0$, w.p.1. Proving by contradiction, we assume that there exist $\epsilon > 0$ and t_0, s.t. $\|w_t - w_*\|^2 \geq \epsilon$ for $\forall t \geq t_0$. Hence,

$$\sum_{t=t_0}^{\infty} \mu\eta_t \|w_t - w_*\|^2 \geq \mu\epsilon \sum_{t=t_0}^{\infty} \eta_t = \infty.$$

This is a contradiction. Therefore, $\|w_t - w_*\|^2 \to 0$ w.p.1. ■

Theorem 2. *Let Assumptions 1, 2 and 3 hold. Let $E = \frac{2\alpha L}{\mu}$ with $\alpha = 2$. Consider Algorithm 1 with a stepsize sequence such that $\eta_t = \frac{\alpha}{\mu(t+E)} \leq \eta_0 = \frac{1}{2L}$. Then,*

$$\mathbb{E}[\|w_t - w_*\|^2] \leq \frac{4\alpha^2 N}{\mu^2} \frac{1}{(t - T + E)}$$

for $t \geq T = \frac{4L}{\mu} \max\{\frac{L\mu}{N}\|w_0 - w_*\|^2, 1\} - \frac{4L}{\mu}$, where $N = 2\mathbb{E}[\|\nabla f(w_*; \xi)\|^2]$ and $w_* = \arg\min_w F(w)$.

Proof Using the beginning of the proof of Theorem 1, taking the expectation to (24), with $0 < \eta_t \leq \frac{1}{2L}$, we have

$$\mathbb{E}[\|w_{t+1} - w_*\|^2] \leq (1 - \mu\eta_t)\mathbb{E}[\|w_t - w_*\|^2] + \eta_t^2 N.$$

We first show that

$$\mathbb{E}[\|w_t - w_*\|^2] \leq \frac{N}{\mu^2} G \frac{1}{(t+E)}, \tag{25}$$

where $G = \max\{I, J\}$, and

$$I = \frac{E\mu^2}{N} \mathbb{E}[\|w_0 - w_*\|^2] > 0,$$
$$J = \frac{\alpha^2}{\alpha - 1} > 0.$$

We use mathematical induction to prove (25) (this trick is based on the idea from Bottou et al. (2018)). Let $t = 0$, we have

$$\mathbb{E}[\|w_0 - w_*\|^2] \leq \frac{NG}{\mu^2 E},$$

which is obviously true since $G \geq \frac{E\mu^2}{N}\|w_0 - w_*\|^2$.

Suppose it is true for t, we need to show that it is also true for $t+1$. We have

$$\begin{aligned}
\mathbb{E}[\|w_{t+1} - w_*\|^2] &\leq \left(1 - \frac{\alpha}{t+E}\right) \frac{NG}{\mu^2(t+E)} + \frac{\alpha^2 N}{\mu^2(t+E)^2} \\
&= \left(\frac{t+E-\alpha}{\mu^2(t+E)^2}\right) NG + \frac{\alpha^2 N}{\mu^2(t+E)^2} \\
&= \left(\frac{t+E-1}{\mu^2(t+E)^2}\right) NG - \left(\frac{\alpha-1}{\mu^2(t+E)^2}\right) NG + \frac{\alpha^2 N}{\mu^2(t+E)^2}.
\end{aligned}$$

Since $G \geq \frac{\alpha^2}{\alpha-1}$,

$$-\left(\frac{\alpha-1}{\mu^2(t+E)^2}\right) NG + \frac{\alpha^2 N}{\mu^2(t+E)^2} \leq 0.$$

This implies

$$\begin{aligned}
\mathbb{E}[\|w_{t+1} - w_*\|^2] &\leq \left(\frac{t+E-1}{\mu^2(t+E)^2}\right) NG \\
&= \left(\frac{(t+E)^2 - 1}{(t+E)^2}\right) \frac{NG}{\mu^2(t+E+1)} \\
&\leq \frac{NG}{\mu^2(t+E+1)}.
\end{aligned}$$

This proves (25) by induction in t.

Notice that the induction proof of (25) holds more generally for $E \geq \frac{2\alpha L}{\mu}$ with $\alpha > 1$ (this is sufficient for showing $\eta_t \leq \frac{1}{2L}$. In this more general interpretation we can see that

the convergence rate is minimized for I minimal, i.e., $E = \frac{2\alpha L}{\mu}$ and for this reason we have fixed E as such in the theorem statement.

Notice that

$$G = \max\{I, J\} = \max\{\frac{2\alpha L\mu}{N}\mathbb{E}[\|w_0 - w_*\|^2], \frac{\alpha^2}{\alpha - 1}\}.$$

We choose $\alpha = 2$ such that η_t only depends on known parameters μ and L. For this α we obtain

$$G = 4\max\{\frac{L\mu}{N}\mathbb{E}[\|w_0 - w_*\|^2], 1\}.$$

For $T = \frac{4L}{\mu}\max\{\frac{L\mu}{N}\mathbb{E}[\|w_0 - w_*\|^2], 1\} - \frac{4L}{\mu}$, we have that according to (25)

$$\begin{aligned} \frac{L\mu}{N}\mathbb{E}[\|w_T - w_*\|^2] &\leq \frac{L\mu}{N}\frac{N}{\mu^2}\frac{G}{(T+E)} \\ &= \frac{L}{\mu}\frac{4\max\{\frac{L\mu}{N}\mathbb{E}[\|w_0 - w_*\|^2], 1\}}{\frac{4L}{\mu}\max\{\frac{L\mu}{N}\mathbb{E}[\|w_0 - w_*\|^2], 1\}} = 1. \end{aligned} \tag{26}$$

Applying (25)with w_T as starting point rather than w_0 gives, for $t \geq \max\{T, 0\}$,

$$\mathbb{E}[\|w_t - w_*\|^2] \leq \frac{N}{\mu^2}G\frac{1}{(t - T + E)},$$

where G is now equal to

$$4\max\{\frac{L\mu}{N}\mathbb{E}[\|w_T - w_*\|^2], 1\},$$

which equals 4, see (26). For any given w_0, we prove the theorem. ■

Appendix D. Analysis for Algorithm 2

The analysis for Algorithm 2 can be done in below.

D.1. Recurrence and Notation

We introduce the following notation: For each ξ, we define $D_\xi \subseteq \{1, \ldots, d\}$ as the set of possible non-zero positions in a vector of the form $\nabla f(w; \xi)$ for some w. We consider a fixed mapping from $u \in U$ to subsets $S_u^\xi \subseteq D_\xi$ for each possible ξ. In our notation we also let D_ξ represent the diagonal $d \times d$ matrix with ones exactly at the positions corresponding to D_ξ and with zeroes elsewhere. Similarly, S_u^ξ also denotes a diagonal matrix with ones at the positions corresponding to D_ξ.

We will use a probability distribution $p_\xi(u)$ to indicate how to randomly select a matrix S_u^ξ. We choose the matrices S_u^ξ and distribution $p_\xi(u)$ so that there exist d_ξ such that

$$d_\xi \mathbb{E}[S_u^\xi | \xi] = D_\xi, \tag{27}$$

where the expectation is over $p_\xi(u)$.

We will restrict ourselves to choosing *non-empty* sets S_u^ξ that partition D_ξ in D approximately equally sized sets together with uniform distributions $p_\xi(u)$ for some fixed D. So, if $D \leq |D_\xi|$, then sets have sizes $\lfloor |D_\xi|/D \rfloor$ and $\lceil |D_\xi|/D \rceil$. For the special case $D > |D_\xi|$ we have exactly $|D_\xi|$ singleton sets of size 1 (in our definition we only use non-empty sets).

For example, for $D = \bar{\Delta}$, where

$$\bar{\Delta} = \max_\xi\{|D_\xi|\}$$

represents the maximum number of non-zero positions in any gradient computation $f(w; \xi)$, we have that for all ξ, there are exactly $|D_\xi|$ singleton sets S_u^ξ representing each of the elements in D_ξ. Since $p_\xi(u) = 1/|D_\xi|$ is the uniform distribution, we have $\mathbb{E}[S_u^\xi|\xi] = D_\xi/|D_\xi|$, hence, $d_\xi = |D_\xi|$. As another example at the other extreme, for $D = 1$, we have exactly one set $S_1^\xi = D_\xi$ for each ξ. Now $p_\xi(1) = 1$ and we have $d_\xi = 1$.

We define the parameter

$$\bar{\Delta}_D \stackrel{\text{def}}{=} D \cdot \mathbb{E}[\lceil |D_\xi|/D \rceil],$$

where the expectation is over ξ. We use $\bar{\Delta}_D$ in the leading asymptotic term for the convergence rate in our main theorem. We observe that

$$\bar{\Delta}_D \leq \mathbb{E}[|D_\xi|] + D - 1$$

and $\bar{\Delta}_D \leq \bar{\Delta}$ with equality for $D = \bar{\Delta}$.

For completeness we define

$$\Delta \stackrel{\text{def}}{=} \max_i \mathbb{P}\left(i \in D_\xi\right).$$

Let us remark, that $\Delta \in (0, 1]$ measures the probability of collision. Small Δ means that there is a small chance that the support of two random realizations of $\nabla f(w; \xi)$ will have an intersection. On the other hand, $\Delta = 1$ means that almost surely, the support of two stochastic gradients will have non-empty intersection.

With this definition of Δ it is an easy exercise to show that for iid ξ_1 and ξ_2 in a finite-sum setting (i.e., ξ_i and ξ_2 can only take on a finite set of possible values) we have

$$\begin{aligned} &\mathbb{E}[|\langle \nabla f(w_1; \xi_1), \nabla f(w_2; \xi_2)\rangle|] \\ &\leq \frac{\sqrt{\Delta}}{2}\left(\mathbb{E}[\|\nabla f(w_1; \xi_1)\|^2] + \mathbb{E}[\|\nabla f(w_2; \xi_2)\|^2]\right) \end{aligned} \tag{28}$$

(see Proposition 10 in Leblond et al. (2018)). We notice that in the non-finite sum setting we can use the property that for any two vectors a and b, $\langle a, b\rangle \leq (\|a\|^2 + \|b\|^2)/2$ and this proves (28) with Δ set to $\Delta = 1$. In our asymptotic analysis of the convergence rate, we will show how Δ plays a role in non-leading terms — this, with respect to the leading term, it will not matter whether we use $\Delta = 1$ or Δ equal the probability of collision (in the finite sum case).

We have

$$w_{t+1} = w_t - \eta_t d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t), \tag{29}$$

where $\hat{w}_t$ represents the vector used in computing the gradient $\nabla f(\hat{w}_t; \xi_t)$ and whose entries have been read (one by one) from an aggregate of a mix of previous updates that led to w_j, $j \leq t$. Here, we assume that

- updating/writing to vector positions is atomic, reading vector positions is atomic, and
- there exists a "delay" τ such that, for all t, vector $\hat{w}_t$ includes all the updates up to and including those made during the $(t-\tau)$-th iteration (where (29) defines the $(t+1)$-st iteration).

Notice that we do **not assume consistent reads and writes of vector positions**. We only assume that up to a "delay" τ all writes/updates are included in the values of positions that are being read.

According to our definition of τ, in (29) vector $\hat{w}_t$ represents an inconsistent read with entries that contain all of the updates made during the 1st to $(t-\tau)$-th iteration. Furthermore each entry in $\hat{w}_t$ includes some of the updates made during the $(t-\tau+1)$-th iteration up to t-th iteration. Each entry includes its own subset of updates because writes are inconsistent. We model this by "masks" $\Sigma_{t,j}$ for $t-\tau \leq j \leq t-1$. A mask $\Sigma_{t,j}$ is a diagonal 0/1-matrix with the 1s expressing which of the entry updates made in the $(j+1)$-th iteration are included in $\hat{w}_t$. That is,

$$\hat{w}_t = w_{t-\tau} - \sum_{j=t-\tau}^{t-1} \eta_j d_{\xi_j} \Sigma_{t,j} S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j). \tag{30}$$

Notice that the recursion (29) implies

$$w_t = w_{t-\tau} - \sum_{j=t-\tau}^{t-1} \eta_j d_{\xi_j} S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j). \tag{31}$$

By combining (31) and (30) we obtain

$$w_t - \hat{w}_t = - \sum_{j=t-\tau}^{t-1} \eta_j d_{\xi_j} (I - \Sigma_{t,j}) S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j), \tag{32}$$

where I represents the identity matrix.

D.2. Main Analysis

We first derive a couple lemmas which will help us deriving our main bounds. In what follows let Assumptions 1, 2, 3 and 4 hold for all lemmas. We define

$$\mathcal{F}_t = \sigma(w_0, \xi_0, u_0, \sigma_0, \ldots, \xi_{t-1}, u_{t-1}, \sigma_{t-1}),$$

where

$$\sigma_{t-1} = (\Sigma_{t,t-\tau}, \ldots, \Sigma_{t,t-1}).$$

When we subtract τ from, for example, t and write $t-\tau$, we will actually mean $\max\{t-\tau, 0\}$.

Lemma 6 *We have*

$$\mathbb{E}[\|d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t)\|^2 | \mathcal{F}_t, \xi_t] \leq D \|\nabla f(\hat{w}_t; \xi_t)\|^2$$

and

$$\mathbb{E}[d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t) | \mathcal{F}_t] = \nabla F(\hat{w}_t).$$

Proof For the first bound, if we take the expectation of $\|d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t)\|^2$ with respect to u_t, then we have (for vectors x we denote the value if its i-th position by $[x]_i$)

$$\mathbb{E}[\|d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t)\|^2 | \mathcal{F}_t, \xi_t] = d_{\xi_t}^2 \sum_u p_{\xi_t}(u) \|S_u^{\xi_t} \nabla f(\hat{w}_t; \xi_t)\|^2 =$$

$$= d_{\xi_t}^2 \sum_u p_{\xi_t}(u) \sum_{i \in S_u^{\xi_t}} [\nabla f(\hat{w}_t; \xi_t)]_i^2 d_{\xi_t} \sum_{i \in D_{\xi_t}} [\nabla f(\hat{w}_t; \xi_t)]_i^2 = d_{\xi_t} \|f(\hat{w}_t; \xi_t)\|^2 \leq D \|\nabla f(\hat{w}_t; \xi_t)\|^2,$$

where the transition to the second line follows from (27).

For the second bound, if we take the expectation of $d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t)$ wrt u_t, then we have:

$$\mathbb{E}[d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t) | \mathcal{F}_t, \xi_t] = d_{\xi_t} \sum_u p_{\xi_t}(u) S_u^{\xi_t} \nabla f(\hat{w}_t; \xi_t) = D_{\xi_t} \nabla f(\hat{w}_t; \xi_t) = \nabla f(\hat{w}_t; \xi_t),$$

and this can be used to derive

$$\mathbb{E}[d_{\xi_t} S_{u_t}^{\xi_t} f(\hat{w}_t; \xi_t) | \mathcal{F}_t] = \mathbb{E}[\mathbb{E}[d_{\xi_t} S_{u_t}^{\xi_t} f(\hat{w}_t; \xi_t) | \mathcal{F}_t, \xi_t] | \mathcal{F}_t] = \nabla F(\hat{w}_t).$$

■

As a consequence of this lemma we derive a bound on the expectation of $\|w_t - \hat{w}_t\|^2$.

Lemma 7 *The expectation of $\|w_t - \hat{w}_t\|^2$ is at most*

$$\mathbb{E}[\|w_t - \hat{w}_t\|^2] \leq (1 + \sqrt{\Delta}\tau) D \sum_{j=t-\tau}^{t-1} \eta_j^2 (2L^2 \mathbb{E}[\|\hat{w}_j - w_*\|^2] + N).$$

Proof As shown in (32),

$$w_t - \hat{w}_t = - \sum_{j=t-\tau}^{t-1} \eta_j d_{\xi_j} (I - \Sigma_{t,j}) S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j).$$

This can be used to derive an expression for the square of its norm: $\|w_t - \hat{w}_t\|^2$

$$\begin{aligned} &= \|\sum_{j=t-\tau}^{t-1} \eta_j d_{\xi_j} (I - \Sigma_{t,j}) S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j)\|^2 \\ &= \sum_{j=t-\tau}^{t-1} \|\eta_j d_{\xi_j} (I - \Sigma_{t,j}) S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j)\|^2 \\ &\quad + \sum_{i \neq j \in \{t-\tau, \ldots, t-1\}} \langle \eta_j d_{\xi_j} (I - \Sigma_{t,j}) S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j), \eta_i d_{\xi_i} (I - \Sigma_{t,j}) S_{u_i}^{\xi_i} \nabla f(\hat{w}_i; \xi_i) \rangle. \end{aligned}$$

Applying (28) to the inner products implies

$$\begin{aligned}\|w_t-\hat{w}_t\|^2 &\le \sum_{j=t-\tau}^{t-1}\|\eta_j d_{\xi_j}(I-\Sigma_{t,j})S_{u_j}^{\xi_j}\nabla f(\hat{w}_j;\xi_j)\|^2\\ &\quad+\sum_{i\neq j\in\{t-\tau,\dots,t-1\}}[\|\eta_j d_{\xi_j}(I-\Sigma_{t,j})S_{u_j}^{\xi_j}\nabla f(\hat{w}_j;\xi_j)\|^2\\ &\quad+\|\eta_i d_{\xi_i}(I-\Sigma_{t,j})S_{u_i}^{\xi_i}\nabla f(\hat{w}_i;\xi_i)\|^2]\sqrt{\Delta}/2\\ &= (1+\sqrt{\Delta}\tau)\sum_{j=t-\tau}^{t-1}\|\eta_j d_{\xi_j}(I-\Sigma_{t,j})S_{u_j}^{\xi_j}\nabla f(\hat{w}_j;\xi_j)\|^2\\ &\le (1+\sqrt{\Delta}\tau)\sum_{j=t-\tau}^{t-1}\eta_j^2\|d_{\xi_j}S_{u_j}^{\xi_j}\nabla f(\hat{w}_j;\xi_j)\|^2.\end{aligned}$$

Taking expectations shows

$$\mathbb{E}[\|w_t-\hat{w}_t\|^2]\le(1+\sqrt{\Delta}\tau)\sum_{j=t-\tau}^{t-1}\eta_j^2\mathbb{E}[\|d_{\xi_j}S_{u_j}^{\xi_j}\nabla f(\hat{w}_j;\xi_j)\|^2].$$

Now, we can apply Lemma 15: We first take the expectation over u_j and this shows

$$\mathbb{E}[\|w_t-\hat{w}_t\|^2]\le(1+\sqrt{\Delta}\tau)\sum_{j=t-\tau}^{t-1}\eta_j^2 D\mathbb{E}[\|\nabla f(\hat{w}_j;\xi_j)\|^2].$$

From Lemma 1 we infer

$$\mathbb{E}[\|\nabla f(\hat{w}_j;\xi_j)\|^2]\le 4L\mathbb{E}[F(\hat{w}_j)-F(w_*)]+N \tag{33}$$

and by L-smoothness, see Equation 7 with $\nabla F(w_*)=0$,

$$F(\hat{w}_j)-F(w_*)\le\frac{L}{2}\|\hat{w}_j-w_*\|^2.$$

Combining the above inequalities proves the lemma. ■

Together with the next lemma we will be able to start deriving a recursive inequality from which we will be able to derive a bound on the convergence rate.

Lemma 8 *Let $0<\eta_t\le\frac{1}{4LD}$ for all $t\ge 0$. Then,*

$$\mathbb{E}[\|w_{t+1}-w_*\|^2|\mathcal{F}_t]\le\left(1-\frac{\mu\eta_t}{2}\right)\|w_t-w_*\|^2+[(L+\mu)\eta_t+2L^2\eta_t^2D]\|\hat{w}_t-w_t\|^2+2\eta_t^2DN.$$

Proof Since $w_{t+1}=w_t-\eta_t d_{\xi_t}S_{u_t}^{\xi_t}\nabla f(\hat{w}_t;\xi_t)$, we have

$$\|w_{t+1}-w_*\|^2=\|w_t-w_*\|^2-2\eta_t\langle d_{\xi_t}S_{u_t}^{\xi_t}\nabla f(\hat{w}_t;\xi_t),(w_t-w_*)\rangle+\eta_t^2\|d_{\xi_t}S_{u_t}^{\xi_t}\nabla f(\hat{w}_t;\xi_t)\|^2.$$

We now take expectations over u_t and ξ_t and use Lemma 15:

$$\begin{aligned}&\mathbb{E}[\|w_{t+1}-w_*\|^2|\mathcal{F}_t]\\ \le\;&\|w_t-w_*\|^2-2\eta_t\langle\nabla F(\hat{w}_t),(w_t-w_*)\rangle+\eta_t^2D\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t]\\ =\;&\|w_t-w_*\|^2-2\eta_t\langle\nabla F(\hat{w}_t),(w_t-\hat{w}_t)\rangle-2\eta_t\langle\nabla F(\hat{w}_t),(\hat{w}_t-w_*)\rangle\\ &+\eta_t^2D\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t].\end{aligned}$$

By (3) and (7), we have

$$-\langle\nabla F(\hat{w}_t),(\hat{w}_t-w_*)\rangle\le-[F(\hat{w}_t)-F(w_*)]-\frac{\mu}{2}\|\hat{w}_t-w_*\|^2,\text{ and} \tag{34}$$

$$-\langle\nabla F(\hat{w}_t),(w_t-\hat{w}_t)\rangle\le F(\hat{w}_t)-F(w_t)+\frac{L}{2}\|\hat{w}_t-w_t\|^2 \tag{35}$$

Thus, $\mathbb{E}[\|w_{t+1}-w_*\|^2|\mathcal{F}_t]$ is at most

$$\begin{aligned}&\overset{(34)(35)}{\le}\|w_t-w_*\|^2+2\eta_t[F(\hat{w}_t)-F(w_t)]+L\eta_t\|\hat{w}_t-w_t\|^2-2\eta_t[F(\hat{w}_t)-F(w_*)]\\ &\quad-\mu\eta_t\|\hat{w}_t-w_*\|^2+\eta_t^2D\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t]\\ &=\|w_t-w_*\|^2-2\eta_t[F(w_t)-F(w_*)]+L\eta_t\|\hat{w}_t-w_t\|^2-\mu\eta_t\|\hat{w}_t-w_*\|^2\\ &\quad+\eta_t^2D\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t].\end{aligned}$$

Since

$$-\|\hat{w}_t-w_*\|^2=-\|(w_t-w_*)-(w_t-\hat{w}_t)\|^2\overset{(20)}{\le}-\frac{1}{2}\|w_t-w_*\|^2+\|w_t-\hat{w}_t\|^2,$$

$\mathbb{E}[\|w_{t+1}-w_*\|^2|\mathcal{F}_t,\sigma_t]$ is at most

$$(1-\frac{\mu\eta_t}{2})\|w_t-w_*\|^2-2\eta_t[F(w_t)-F(w_*)]+(L+\mu)\eta_t\|\hat{w}_t-w_t\|^2+\eta_t^2D\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t].$$

We now use $\|a\|^2=\|a-b+b\|^2\le 2\|a-b\|^2+2\|b\|^2$ for $\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t]$ to obtain

$$\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t]\le 2\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)-\nabla f(w_t;\xi_t)\|^2|\mathcal{F}_t]+2\mathbb{E}[\|\nabla f(w_t;\xi_t)\|^2|\mathcal{F}_t]. \tag{36}$$

By Lemma 1, we have

$$\mathbb{E}[\|\nabla f(w_t;\xi_t)\|^2|\mathcal{F}_t]\le 4L[F(w_t)-F(w_*)]+N. \tag{37}$$

Applying (6) twice gives

$$\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)-\nabla f(w_t;\xi_t)\|^2|\mathcal{F}_t,\sigma_t]\le L^2\|\hat{w}_t-w_t\|^2$$

and together with (36) and (37) we obtain

$$\mathbb{E}[\|\nabla f(\hat{w}_t;\xi_t)\|^2|\mathcal{F}_t]\le 2L^2\|\hat{w}_t-w_t\|^2+4L[F(w_t)-F(w_*)]+N.$$

Plugging this into the previous derivation yields

$$\begin{aligned}\mathbb{E}[\|w_{t+1}-w_*\|^2|\mathcal{F}_t] &\le (1-\frac{\mu\eta_t}{2})\|w_t-w_*\|^2 - 2\eta_t[F(w_t)-F(w_*)] + (L+\mu)\eta_t\|\hat{w}_t-w_t\|^2 \\ &\quad + 2L^2\eta_t^2 D\|\hat{w}_t-w_t\|^2 + 8L\eta_t^2 D[F(w_t)-F(w_*)] + 2\eta_t^2 DN \\ &= (1-\frac{\mu\eta_t}{2})\|w_t-w_*\|^2 + [(L+\mu)\eta_t + 2L^2\eta_t^2 D]\|\hat{w}_t-w_t\|^2 \\ &\quad - 2\eta_t(1-4L\eta_t D)[F(w_t)-F(w_*)] + 2\eta_t^2 DN.\end{aligned}$$

Since $\eta_t \le \frac{1}{4LD}$, $-2\eta_t(1-4L\eta_t D)[F(w_t)-F(w_*)] \le 0$ (we can get a negative upper bound by applying strong convexity but this will not improve the asymptotic behavior of the convergence rate in our main result although it would improve the constant of the leading term making the final bound applied to SGD closer to the bound of Theorem 2 for SGD),

$$\mathbb{E}[\|w_{t+1}-w_*\|^2|\mathcal{F}_t] \le \left(1-\frac{\mu\eta_t}{2}\right)\|w_t-w_*\|^2 + [(L+\mu)\eta_t + 2L^2\eta_t^2 D]\|\hat{w}_t-w_t\|^2 + 2\eta_t^2 DN$$

and this concludes the proof. ■

Assume $0 < \eta_t \le \frac{1}{4LD}$ for all $t \ge 0$. Then, after taking the full expectation of the inequality in Lemma 8, we can plug Lemma 7 into it which yields the recurrence

$$\begin{aligned}\mathbb{E}[\|w_{t+1}-w_*\|^2] \quad \le \quad & \left(1-\frac{\mu\eta_t}{2}\right)\mathbb{E}[\|w_t-w_*\|^2] + \\ & [(L+\mu)\eta_t + 2L^2\eta_t^2 D](1+\sqrt{\Delta}\tau)D\sum_{j=t-\tau}^{t-1}\eta_j^2(2L^2\mathbb{E}[\|\hat{w}_j-w_*\|^2]+N) \\ & +2\eta_t^2 DN. \end{aligned} \tag{38}$$

This can be solved by using the next lemma. For completeness, we follow the convention that an empty product is equal to 1 and an empty sum is equal to 0, i.e.,

$$\prod_{i=h}^{k} g_i = 1 \text{ and } \sum_{i=h}^{k} g_i = 0 \text{ if } k < h. \tag{39}$$

Lemma 9 *Let Y_t, β_t and γ_t be sequences such that $Y_{t+1} \le \beta_t Y_t + \gamma_t$, for all $t \ge 0$. Then,*

$$Y_{t+1} \le (\sum_{i=0}^{t}[\prod_{j=i+1}^{t}\beta_j]\gamma_i) + (\prod_{j=0}^{t}\beta_j)Y_0. \tag{40}$$

Proof We prove the lemma by using induction. It is obvious that (40) is true for $t=0$ because $Y_1 \le \beta_1 Y_0 + \gamma_1$. Assume as induction hypothesis that (40) is true for $t-1$. Since

$Y_{t+1} \le \beta_t Y_t + \gamma_t$,

$$\begin{aligned}Y_{t+1} &\le \beta_t Y_t + \gamma_t \\ &\le \beta_t[(\sum_{i=0}^{t-1}[\prod_{j=i+1}^{t-1}\beta_j]\gamma_i) + (\prod_{j=0}^{t-1}\beta_j)Y_0] + \gamma_t \\ &\overset{(39)}{=} (\sum_{i=0}^{t-1}\beta_t[\prod_{j=i+1}^{t-1}\beta_j]\gamma_i) + \beta_t(\prod_{j=0}^{t-1}\beta_j)Y_0 + (\prod_{j=t+1}^{t}\beta_j)\gamma_t \\ &= [(\sum_{i=0}^{t-1}[\prod_{j=i+1}^{t}\beta_j]\gamma_i) + (\prod_{j=t+1}^{t}\beta_j)\gamma_t] + (\prod_{j=0}^{t}\beta_j)Y_0 \\ &= (\sum_{i=0}^{t}[\prod_{j=i+1}^{t}\beta_j]\gamma_i) + (\prod_{j=0}^{t}\beta_j)Y_0.\end{aligned}$$

■

Applying the above lemma to (38) will yield the following bound.

Lemma 10 *Let $\eta_t = \frac{\alpha_t}{\mu(t+E)}$ with $4 \le \alpha_t \le \alpha$ and $E = \max\{2\tau, \frac{4L\alpha D}{\mu}\}$. Then,*

$$\begin{aligned}\mathbb{E}[\|w_{t+1}-w_*\|^2] \le \frac{\alpha^2 D}{\mu^2}\frac{1}{(t+E-1)^2}\cdot & \\ \cdot\left(\sum_{i=1}^{t}\left[4a_i(1+\sqrt{\Delta}\tau)[N\tau + 2L^2\sum_{j=i-\tau}^{i-1}\mathbb{E}[\|\hat{w}_j-w_*\|^2] + 2N\right]\right) & \\ + \frac{(E+1)^2}{(t+E-1)^2}\mathbb{E}[\|w_0-w_*\|^2], & \end{aligned}$$

where $a_i = (L+\mu)\eta_i + 2L^2\eta_i^2 D$.

Proof Notice that we may use (38) because $\eta_t \le \frac{1}{4LD}$ follows from $\eta_t = \frac{\alpha_t}{\mu(t+E)} \le \frac{\alpha}{\mu(t+E)}$ combined with $E \ge \frac{4L\alpha D}{\mu}$. From (38) with $a_t = (L+\mu)\eta_t + 2L^2\eta_t^2 D$ and η_t being decreasing in t we infer $\mathbb{E}[\|w_{t+1}-w_*\|^2]$

$$\begin{aligned}\le \quad & (1-\tfrac{\mu\eta_t}{2})\,\mathbb{E}[\|w_t-w_*\|^2] + a_t(1+\sqrt{\Delta}\tau)D\eta_{t-\tau}^2\textstyle\sum_{j=t-\tau}^{t-1}(2L^2\mathbb{E}[\|\hat{w}_j-w_*\|^2]+N) \\ & +2\eta_t^2 DN \\ = \quad & (1-\tfrac{\mu\eta_t}{2})\,\mathbb{E}[\|w_t-w_*\|^2] + a_t(1+\sqrt{\Delta}\tau)D\eta_{t-\tau}^2[N\tau + 2L^2\textstyle\sum_{j=t-\tau}^{t-1}\mathbb{E}[\|\hat{w}_j-w_*\|^2] \\ & +2\eta_t^2 DN.\end{aligned}$$

Since $E \ge 2\tau$, $\frac{1}{t-\tau+E} \le \frac{2}{t+E}$. Hence, together with $\eta_{t-\tau} = \frac{\alpha_{t-\tau}}{\mu(t-\tau+E)} \le \frac{\alpha}{\mu(t-\tau+E)}$ we have

$$\eta_{t-\tau}^2 \le \frac{4\alpha^2}{\mu^2}\frac{1}{(t+E)^2}. \tag{41}$$

This translates the above bound into

$$\mathbb{E}[\|w_{t+1}-w_*\|^2] \leq \beta_t \mathbb{E}[\|w_t - w_*\|^2] + \gamma_t,$$

for

$$\beta_t = 1 - \frac{\mu\eta_t}{2},$$
$$\gamma_t = 4a_t(1+\sqrt{\Delta}\tau)D\frac{\alpha^2}{\mu^2}\frac{1}{(t+E)^2}[N\tau + 2L^2\sum_{j=t-\tau}^{t-1}\mathbb{E}[\|\hat{w}_j - w_*\|^2] + 2\eta_t^2 DN, where$$
$$a_t = (L+\mu)\eta_t + 2L^2\eta_t^2 D.$$

Application of Lemma 9 for $Y_{t+1} = \mathbb{E}[\|w_{t+1}-w_*\|^2]$ and $Y_t = \mathbb{E}[\|w_t - w_*\|^2]$ gives

$$\mathbb{E}[\|w_{t+1}-w_*\|^2] \leq \left(\sum_{i=0}^{t}\left[\prod_{j=i+1}^{t}\left(1-\frac{\mu\eta_j}{2}\right)\right]\gamma_i\right) + \left(\prod_{j=0}^{t}\left(1-\frac{\mu\eta_j}{2}\right)\right)\mathbb{E}[\|w_0 - w_*\|^2].$$

In order to analyze this formula, since $\eta_j = \frac{\alpha_j}{\mu(j+E)}$ with $\alpha_j \geq 4$, we have

$$1 - \frac{\mu\eta_j}{2} = 1 - \frac{\alpha_j}{2(j+E)} \leq 1 - \frac{2}{j+E},$$

Hence (we can also use $1-x \leq e^{-x}$ which leads to similar results and can be used to show that our choice for η_t leads to the tightest convergence rates in our framework),

$$\begin{aligned}\prod_{j=i}^{t}\left(1-\frac{\mu\eta_j}{2}\right) &\leq \prod_{j=i}^{t}\left(1-\frac{2}{j+E}\right) = \prod_{j=i}^{t}\frac{j+E-2}{j+E}\\ &= \frac{i+E-2}{i+E}\frac{i+E-1}{i+E+1}\frac{i+E}{i+E+2}\frac{i+E+1}{i+E+3}\cdots\frac{t+E-3}{t+E-1}\frac{t+E-2}{t+E}\\ &= \frac{(i+E-2)(i+E-1)}{(t+E-1)(t+E)} \leq \frac{(i+E-1)^2}{(t+E-1)(t+E)} \leq \frac{(i+E)^2}{(t+E-1)^2}.\end{aligned}$$

From this calculation we infer that

$$\mathbb{E}[\|w_{t+1}-w_*\|^2] \leq \left(\sum_{i=0}^{t}\left[\frac{(i+E)^2}{(t+E-1)^2}\right]\gamma_i\right) + \frac{(E+1)^2}{(t+E-1)^2}\mathbb{E}[\|w_0-w_*\|^2]. \tag{42}$$

Now, we substitute $\eta_i \leq \frac{\alpha}{\mu(i+E)}$ in γ_i and compute

$$\begin{aligned}&\frac{(i+E)^2}{(t+E-1)^2}\gamma_i\\ &= \frac{(i+E)^2}{(t+E-1)^2}4a_i(1+\sqrt{\Delta}\tau)D\frac{\alpha^2}{\mu^2}\frac{1}{(i+E)^2}[N\tau + 2L^2\sum_{j=i-\tau}^{i-1}\mathbb{E}[\|\hat{w}_j - w_*\|^2]\\ &+ \frac{(i+E)^2}{(t+E-1)^2}2ND\frac{\alpha^2}{\mu^2(i+E)^2}\\ &= \frac{\alpha^2 D}{\mu^2}\frac{1}{(t+E-1)^2}\left[4a_i(1+\sqrt{\Delta}\tau)[N\tau + 2L^2\sum_{j=i-\tau}^{i-1}\mathbb{E}[\|\hat{w}_j - w_*\|^2] + 2N\right].\end{aligned}$$

Substituting this in (42) proves the lemma. ■

As an immediate corollary we can apply the inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ to $\mathbb{E}[\|\hat{w}_{t+1} - w_*\|^2]$ to obtain

$$\mathbb{E}[\|\hat{w}_{t+1}-w_*\|^2] \leq 2\mathbb{E}[\|\hat{w}_{t+1} - w_{t+1}\|^2] + 2\mathbb{E}[\|w_{t+1}-w_*\|^2], \tag{43}$$

which in turn can be bounded by the previous lemma together with Lemma 7: $\mathbb{E}[\|\hat{w}_{t+1} - w_*\|^2]$

$$\begin{aligned}\leq\ & 2(1+\sqrt{\Delta}\tau)D\sum_{j=t+1-\tau}^{t}\eta_j^2(2L^2\mathbb{E}[\|\hat{w}_j - w_*\|^2] + N) +\\ & 2\frac{\alpha^2 D}{\mu^2}\frac{1}{(t+E-1)^2}\left(\sum_{i=1}^{t}\left[4a_i(1+\sqrt{\Delta}\tau)[N\tau + 2L^2\sum_{j=i-\tau}^{i-1}\mathbb{E}[\|\hat{w}_j - w_*\|^2] + 2N\right]\right) +\\ & \frac{(E+1)^2}{(t+E-1)^2}\mathbb{E}[\|w_0 - w_*\|^2].\end{aligned}$$

Now assume a decreasing sequence Z_t for which we want to prove that $\mathbb{E}[\|\hat{w}_t - w_*\|^2] \leq Z_t$ by induction in t. Then, the above bound can be used together with the property that Z_t and η_t are decreasing in t to show

$$\begin{aligned}\sum_{j=t+1-\tau}^{t}\eta_j^2(2L^2\mathbb{E}[\|\hat{w}_j - w_*\|^2] + N) &\leq \tau\eta_{t-\tau}^2(2L^2 Z_{t+1-\tau} + N)\\ &\leq 4\tau\frac{\alpha^2}{\mu^2}\frac{1}{(t+E-1)^2}(2L^2 Z_{t+1-\tau} + N)\end{aligned}$$

where the last inequality follows from (41), and

$$\sum_{j=i-\tau}^{i-1}\mathbb{E}[\|\hat{w}_j - w_*\|^2] \leq \tau Z_{i-\tau}.$$

From these inequalities we infer

$$\begin{aligned}\mathbb{E}[\|\hat{w}_{t+1}-w_*\|^2] \leq\ & 8(1+\sqrt{\Delta}\tau)\tau D\frac{\alpha^2}{\mu^2}\frac{1}{(t+E-1)^2}(2L^2 Z_{t+1-\tau} + N) +\\ & 2\frac{\alpha^2 D}{\mu^2}\frac{1}{(t+E-1)^2}\left(\sum_{i=1}^{t}\left[4a_i(1+\sqrt{\Delta}\tau)[N\tau + 2L^2\tau Z_{i-\tau}] + 2N\right]\right) +\\ & \frac{(E+1)^2}{(t+E-1)^2}\mathbb{E}[\|w_0 - w_*\|^2].\end{aligned} \tag{44}$$

Even if we assume a constant $Z \geq Z_0 \geq Z_1 \geq Z_2 \geq \ldots$, we can get a first bound on the convergence rate of vectors $\hat{w}^t$: Substituting Z gives

$$\begin{aligned}\mathbb{E}[\|\hat{w}_{t+1} - w_*\|^2] \;\leq\;& 8(1+\sqrt{\Delta}\tau)\tau D\frac{\alpha^2}{\mu^2}\frac{1}{(t+E-1)^2}(2L^2 Z + N) + \\ & 2\frac{\alpha^2 D}{\mu^2}\frac{1}{(t+E-1)^2}\left(\sum_{i=1}^{t}\Big[4a_i(1+\sqrt{\Delta}\tau)[N\tau + 2L^2\tau Z] + 2N\Big]\right) + \\ & \frac{(E+1)^2}{(t+E-1)^2}\mathbb{E}[\|w_0 - w_*\|^2]. \end{aligned} \tag{45}$$

Since $a_i = (L+\mu)\eta_i + 2L^2\eta_i^2 D$ and $\eta_i \leq \frac{\alpha}{\mu(i+E)}$, we have

$$\begin{aligned}\sum_{i=1}^{t} a_i &= (L+\mu)\sum_{i=1}^{t}\eta_i + 2L^2 D\sum_{i=1}^{t}\eta_i^2 \\ &\leq (L+\mu)\sum_{i=1}^{t}\frac{\alpha}{\mu(i+E)} + 2L^2 D\sum_{i=1}^{t}\frac{\alpha^2}{\mu^2(i+E)^2} \\ &\leq \frac{(L+\mu)\alpha}{\mu}\sum_{i=1}^{t}\frac{1}{i} + \frac{2L^2\alpha^2 D}{\mu^2}\sum_{i=1}^{t}\frac{1}{i^2} \\ &\leq \frac{(L+\mu)\alpha}{\mu}(1+\ln t) + \frac{L^2\alpha^2 D\pi^2}{3\mu^2}, \end{aligned} \tag{46}$$

where the last inequality is a property of the harmonic sequence $\sum_{i=1}^{t}\frac{1}{i} \leq 1 + \ln t$ and $\sum_{i=1}^{t}\frac{1}{i^2} \leq \sum_{i=1}^{\infty}\frac{1}{i^2} = \frac{\pi^2}{6}$.

Substituting (46) in (45) and collecting terms yields

$$\begin{aligned}&\mathbb{E}[\|\hat{w}_{t+1} - w_*\|^2] \leq \\ &2\frac{\alpha^2 D}{\mu^2}\frac{1}{(t+E-1)^2}\left(2Nt + 4(1+\sqrt{\Delta}\tau)\tau[N + 2L^2 Z]\left\{\frac{(L+\mu)\alpha}{\mu}(1+\ln t) + \frac{L^2\alpha^2 D\pi^2}{3\mu^2+1}\right\}\right) \\ &+\frac{(E+1)^2}{(t+E-1)^2}\mathbb{E}[\|w_0 - w_*\|^2]. \end{aligned} \tag{47}$$

Notice that the asymptotic behavior in t is dominated by the term

$$\frac{4\alpha^2 DN}{\mu^2}\frac{t}{(t+E-1)^2}.$$

If we define Z_{t+1} to be the right hand side of (47) and observe that this Z_{t+1} is decreasing and a constant Z exists (since the terms with Z decrease much faster in t compared to the dominating term), then this Z_{t+1} satisfies the derivations done above and a proof by induction can be completed.

Our derivations prove our main result: The expected convergence rate of read vectors is

$$\mathbb{E}[\|\hat{w}_{t+1} - w_*\|^2] \;\leq\; \frac{4\alpha^2 DN}{\mu^2}\frac{t}{(t+E-1)^2} + O\left(\frac{\ln t}{(t+E-1)^2}\right).$$

We can use this result in Lemma 10 in order to show that the expected convergence rate $\mathbb{E}[\|w_{t+1} - w_*\|^2]$ satisfies the same bound.

We remind the reader, that in the $(t+1)$-th iteration at most $\leq \lceil |D_{\xi_t}|/D \rceil$ vector positions are updated. Therefore the expected number of single vector entry updates is at most $\bar{\Delta}_D/D$.

Theorem 6. *Suppose Assumptions 1, 2, 3 and 4 and consider Algorithm 2. Let $\eta_t = \frac{\alpha_t}{\mu(t+E)}$ with $4 \leq \alpha_t \leq \alpha$ and $E = \max\{2\tau, \frac{4L\alpha D}{\mu}\}$. Then, $t' = t\bar{\Delta}_D/D$ is the expected number of single vector entry updates after t iterations and*

$$\begin{aligned}\mathbb{E}[\|\hat{w}_t - w_*\|^2] &\leq \tfrac{4\alpha^2 DN}{\mu^2}\tfrac{t}{(t+E-1)^2} + O\left(\tfrac{\ln t}{(t+E-1)^2}\right), \\ \mathbb{E}[\|w_t - w_*\|^2] &\leq \tfrac{4\alpha^2 DN}{\mu^2}\tfrac{t}{(t+E-1)^2} + O\left(\tfrac{\ln t}{(t+E-1)^2}\right),\end{aligned}$$

where $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$ and $w_* = \arg\min_w F(w)$.

D.3. Convergence without Convexity of Component Functions

For the non-convex case of the component functions, L in (33) must be replaced by $L\kappa$ and as a result L^2 in Lemma 7 must be replaced by $L^2\kappa$. Also L in (37) must be replaced by $L\kappa$. We now require that $\eta_t \leq \frac{1}{4L\kappa D}$ so that $-2\eta_t(1 - 4L\kappa\eta_t D)[F(w_t) - F(w_*)] \leq 0$. This leads to Lemma 8 where no changes are needed except requiring $\eta_t \leq \frac{1}{4L\kappa D}$. The changes in Lemmas 7 and 8 lead to a Lemma 10 where we require $E \geq \frac{4L\kappa\alpha D}{\mu}$ and where in the bound of the expectation L^2 must be replaced by $L^2\kappa$. This perculates through to inequality (47) with a similar change finally leading to Theorem 7, i.e., Theorem 6 where we only need to strengthen the condition on E to $E \geq \frac{4L\kappa\alpha D}{\mu}$ in order to remove Assumption 3.

D.4. Sensitivity to τ

What about the upper bound's sensitivity with respect to τ? Suppose τ is not a constant but an increasing function of t, which also makes E a function of t:

$$\frac{2L\alpha D}{\mu} \leq \tau(t) \leq t \text{ and } E(t) = 2\tau(t).$$

In order to obtain a similar theorem we increase the lower bound on α_t to

$$12 \leq \alpha_t \leq \alpha.$$

This allows us to modify the proof of Lemma 10 where we analyse the product

$$\prod_{j=i}^{t}\left(1 - \frac{\mu\eta_j}{2}\right).$$

Since $\alpha_j \geq 12$ and $E(j) = 2\tau(j) \leq 2j$,

$$1 - \frac{\mu\eta_j}{2} = 1 - \frac{\alpha_j}{2(j+E(j))} \leq 1 - \frac{12}{2(j+2j)} = 1 - \frac{2}{j} \leq 1 - \frac{2}{j+1}.$$

The remaining part of the proof of Lemma 10 continues as before where constant E in the proof is replaced by 1. This yields instead of (42)

$$\mathbb{E}[\|w_{t+1}-w_*\|^2] \le \left(\sum_{i=1}^{t}\left[\frac{(i+1)^2}{t^2}\right]\gamma_i\right) + \frac{4}{t^2}\mathbb{E}[\|w_0-w_*\|^2].$$

We again substitute $\eta_i \le \frac{\alpha}{\mu(i+E(i))}$ in γ_i, realize that $\frac{(i+1)}{(i+E(i))} \le 1$, and compute

$$\begin{aligned}
&\frac{(i+1)^2}{t^2}\gamma_i \\
&= \frac{(i+1)^2}{t^2}4a_i(1+\sqrt{\Delta}\tau(i))D\frac{\alpha^2}{\mu^2}\frac{1}{(i+E(i))^2}[N\tau(i)+2L^2\sum_{j=i-\tau(i)}^{i-1}\mathbb{E}[\|\hat{w}_j-w_*\|^2] \\
&+ \frac{(i+1)^2}{t^2}2ND\frac{\alpha^2}{\mu^2(i+E(i))^2} \\
&\le \frac{\alpha^2 D}{\mu^2}\frac{1}{t^2}\left[4a_i(1+\sqrt{\Delta}\tau(i))[N\tau(i)+2L^2\sum_{j=i-\tau(i)}^{i-1}\mathbb{E}[\|\hat{w}_j-w_*\|^2]+2N\right].
\end{aligned}$$

This gives a new Lemma 10:

Lemma 11 *Assume $\frac{2L\alpha D}{\mu} \le \tau(t) \le t$ with $\tau(t)$ monotonic increasing. Let $\eta_t = \frac{\alpha_t}{\mu(t+E(t))}$ with $12 \le \alpha_t \le \alpha$ and $E(t) = 2\tau(t)$. Then,*

$$\begin{aligned}
\mathbb{E}[\|w_{t+1}-w_*\|^2] \le \frac{\alpha^2 D}{\mu^2}\frac{1}{t^2}\cdot \\
\cdot\left(\sum_{i=1}^{t}\left[4a_i(1+\sqrt{\Delta}\tau(i))[N\tau(i)+2L^2\sum_{j=i-\tau(i)}^{i-1}\mathbb{E}[\|\hat{w}_j-w_*\|^2]+2N\right]\right) \\
+\frac{4}{t^2}\mathbb{E}[\|w_0-w_*\|^2],
\end{aligned}$$

where $a_i = (L+\mu)\eta_i + 2L^2\eta_i^2 D$.

Now we can continue the same analysis that led to Theorem 6 and conclude that there exists a constant Z such that, see (45),

$$\begin{aligned}
\mathbb{E}[\|\hat{w}_{t+1}-w_*\|^2] \le\ & 8(1+\sqrt{\Delta}\tau(t))\tau(t)D\frac{\alpha^2}{\mu^2}\frac{1}{t^2}(2L^2Z+N) + \\
& 2\frac{\alpha^2 D}{\mu^2}\frac{1}{t^2}\left(\sum_{i=1}^{t}\left[4a_i(1+\sqrt{\Delta}\tau(i))[N\tau(i)+2L^2\tau(i)Z]+2N\right]\right) + \\
& \frac{4}{t^2}\mathbb{E}[\|w_0-w_*\|^2]. \qquad (48)
\end{aligned}$$

Let us assume

$$\tau(t) \le \sqrt{t\cdot L(t)}, \qquad (49)$$

where

$$L(t) = \frac{1}{\ln t} - \frac{1}{(\ln t)^2}$$

which has the property that the derivative of $t/(\ln t)$ is equal to $L(t)$. Now we observe

$$\begin{aligned}
\sum_{i=1}^{t} a_i\tau(i)^2 &= \sum_{i=1}^{t}[(L+\mu)\eta_i + 2L^2\eta_i^2 D]\tau(i)^2 \le \sum_{i=1}^{t}[(L+\mu)\frac{\alpha}{\mu i} + 2L^2\frac{\alpha^2}{\mu^2 i^2}D]\cdot iL(i) \\
&= \frac{(L+\mu)\alpha}{\mu}\sum_{i=1}^{t}L(i) + O(\ln t) = \frac{(L+\mu)\alpha}{\mu}\frac{t}{\ln t} + O(\ln t)
\end{aligned}$$

and

$$\begin{aligned}
\sum_{i=1}^{t} a_i\tau(i) &= \sum_{i=1}^{t}[(L+\mu)\eta_i + 2L^2\eta_i^2 D]\tau(i) \le \sum_{i=1}^{t}[(L+\mu)\frac{\alpha}{\mu i} + 2L^2\frac{\alpha^2}{\mu^2 i^2}D]\cdot\sqrt{i} \\
&= O(\sum_{i=1}^{t}\frac{1}{\sqrt{i}}) = O(\sqrt{t}).
\end{aligned}$$

Substituting both inequalities in (48) gives $\mathbb{E}[\|\hat{w}_{t+1}-w_*\|^2]$

$$\begin{aligned}
\le\ & 8(1+\sqrt{\Delta}\tau(t))\tau(t)D\frac{\alpha^2}{\mu^2}\frac{1}{t^2}(2L^2Z+N) + \\
& 2\frac{\alpha^2 D}{\mu^2}\frac{1}{t^2}\left(2Nt + 4\sqrt{\Delta}[\frac{(L+\mu)\alpha}{\mu}\frac{t}{\ln t} + O(\ln t)][N+2L^2Z] + O(\sqrt{t})\right) + \\
& \frac{4}{t^2}\mathbb{E}[\|w_0-w_*\|^2] \\
\le\ & 2\frac{\alpha^2 D}{\mu^2}\frac{1}{t^2}\left(2Nt + 4\sqrt{\Delta}[(1+\frac{(L+\mu)\alpha}{\mu})\frac{t}{\ln t} + O(\ln t)][N+2L^2Z] + O(\sqrt{t})\right) \\
& + \frac{4}{t^2}\mathbb{E}[\|w_0-w_*\|^2] \qquad (50)
\end{aligned}$$

Again we define Z_{t+1} as the right hand side of this inequality. Notice that $Z_t = O(1/t)$, since the above derivation proves

$$\mathbb{E}[\|\hat{w}_{t+1}-w_*\|^2] \le \frac{4\alpha^2 DN}{\mu^2}\frac{1}{t} + O(\frac{1}{t\ln t}).$$

Summarizing we have the following main lemma:

Lemma 12 *Let Assumptions 1, 2, 3 and 4 hold and consider Algorithm 2. Assume $\frac{2L\alpha D}{\mu} \le \tau(t) \le \sqrt{t\cdot L(t)}$ with $\tau(t)$ monotonic increasing. Let $\eta_t = \frac{\alpha_t}{\mu(t+2\tau(t))}$ with $12 \le \alpha_t \le \alpha$. Then, the expected convergence rate of read vectors is*

$$\mathbb{E}[\|\hat{w}_{t+1}-w_*\|^2] \le \frac{4\alpha^2 DN}{\mu^2}\frac{1}{t} + O(\frac{1}{t\ln t}),$$

where $L(t) = \frac{1}{\ln t} - \frac{1}{(\ln t)^2}$. The expected convergence rate $\mathbb{E}[\|w_{t+1}-w_*\|^2]$ satisfies the same bound.

Notice that we can plug $Z_t = O(1/t)$ back into an equivalent of (44) where we may bound $Z_{i-\tau(i)} = O(1/(i-\tau(i))$ which replaces Z in the second line of (45). On careful examination this leads to a new upper bound (50) where the $2L^2Z$ terms gets absorped in a higher order term. This can be used to show that, for

$$t \geq T_0 = \exp[2\sqrt{\Delta}(1 + \frac{(L+\mu)\alpha}{\mu})],$$

the higher order terms that contain $\tau(t)$ (as defined above) are at most the leading term as given in Lemma 12.

Upper bound (50) also shows that, for

$$t \geq T_1 = \frac{\mu^2}{\alpha^2 ND}\|w_0 - w_*\|^2,$$

the higher order term that contains $\|w_0 - w_*\|^2$ is at most the leading term.

D.5. Convergence of Hogwild! with probability 1

Lemma 13 *Let us consider the sequence $w_0, w_1, w_2, \ldots, w_t, \ldots, w_n$ generated by (29):*

$$w_{t+1} = w_t - \eta_t d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t),$$

and define

$$m_t = \max_{0 \leq i \leq n, 0 \leq t' \leq t} \|\nabla f(w_{t'}; \xi_i)\|.$$

Then,

$$m_t \leq m_0 \exp(LD \sum_{i=0}^{t-1} \eta_i), \tag{51}$$

where $d_{\xi_t} \leq D$ for all ξ_t.

Proof From the L-smoothness assumption (i.e., $\|\nabla f(w;\xi) - \nabla f(w';\xi)\| \leq L\|w - w'\|$, $\forall w, w' \in \mathbb{R}^d$), for any given ξ_i:

$$\begin{aligned}\|\nabla f(w_{t+1}; \xi_i) - \nabla f(w_t; \xi_i)\| &\leq L\|w_{t+1} - w_t\| = L\|\eta_t d_{\xi_t} S_{u_t}^{\xi_t} \nabla f(\hat{w}_t; \xi_t)\| \\ &\leq LD\eta_t \|\nabla f(\hat{w}_t; \xi_t)\|.\end{aligned}$$

Since

$$m_t = \max_{0 \leq i \leq n, 0 \leq t' \leq t} \|\nabla f(w_{t'}; \xi_i)\|,$$

we have

$$\|\nabla f(w_{t+1}; \xi_i) - \nabla f(w_t; \xi_i)\| \leq LD\eta_t m_t$$

for any $i \in [n]$ and t. Using the triangular inequality, we obtain

$$\begin{aligned}\|\nabla f(w_{t+1}; \xi_i)\| &\leq \|\nabla f(w_t; \xi_i)\| + \|\nabla f(w_{t+1}; \xi_i) - \nabla f(w_t; \xi_i)\| \\ &\overset{\forall i, \|\nabla f(w_t;\xi_i)\| \leq m_t}{\leq} m_t + LD\eta_t \|\nabla f(\hat{w}_t; \xi_t)\| \\ &\overset{\forall i, \|\nabla f_i(\hat{w}_t;\xi_i)\| \leq m_t}{\leq} (1 + LD\eta_t) m_t.\end{aligned}$$

Moreover, the result above implies $m_{t+1} \leq (1 + LD\eta_t)m_t$ and unrolling m_t yields

$$m_{t+1} \leq m_0 \prod_{i=0}^{t} (1 + LD\eta_i).$$

For all $x \geq 0$, it is always true that $1 + x \leq \exp(x)$. Hence, we have

$$m_{t+1} \leq m_0 \prod_{i=0}^{t} (1 + LD\eta_i) \leq m_0 \prod_{i=0}^{t} \exp(LD\eta_i) \leq m_0 \exp(LD[\sum_{i=0}^{t} \eta_i]).$$

■

Theorem 5 (Sufficient conditions for almost sure convergence for Hogwild!) *Let Assumptions 1, 2, 3 and 4 hold. Consider Hogwild! method described in Algorithm 2 with a stepsize sequence such that*

$$0 < \eta_t = \frac{1}{LD(2+\beta)(k+t)} < \frac{1}{4LD}, \beta > 0, k \geq 3\tau.$$

Then, the following holds w.p.1 (almost surely)

$$\|w_t - w_*\| \to 0.$$

Proof As shown in Lemma 8, for $0 < \eta_t \leq \frac{1}{4LD}$, we have

$$\begin{aligned}\mathbb{E}[\|w_{t+1} - w_*\|^2 | \mathcal{F}_t] &\leq \left(1 - \frac{\mu\eta_t}{2}\right)\|w_t - w_*\|^2 + [(L+\mu)\eta_t + 2L^2\eta_t^2 D]\|\hat{w}_t - w_t\|^2 + 2\eta_t^2 DN \\ &= \|w_t - w_*\|^2 - \frac{\mu\eta_t}{2}\|w_t - w_*\|^2 + [(L+\mu)\eta_t + 2L^2\eta_t^2 D]\|\hat{w}_t - w_t\|^2 \\ &+ 2\eta_t^2 DN.\end{aligned}$$

If we can show that $\sum_{t=0}^{\infty}[(L+\mu)\eta_t + 2L^2\eta_t^2 D]\|\hat{w}_t - w_t\|^2$ is finite, then it is straight forward to apply the proof technique from Theorem 1 to show that $\|w_t - w_*\|^2 \to 0$ w.p.1. From the proof of Lemma 7, we know $\|w_t - \hat{w}_t\|^2$ is at most

$$\begin{aligned}(1 + \sqrt{\Delta}\tau) \sum_{j=t-\tau}^{t-1} \eta_j^2 \|d_{\xi_j} S_{u_j}^{\xi_j} \nabla f(\hat{w}_j; \xi_j)\|^2 &\leq (1 + \sqrt{\Delta}\tau) D^2 \sum_{j=t-\tau}^{t-1} \eta_j^2 \|\nabla f(\hat{w}_j; \xi_j)\|^2 \\ &\leq (1 + \sqrt{\Delta}\tau) D^2 \tau m_t^2 \eta_{t-\tau}^2.\end{aligned}$$

Since $\eta_{t-\tau} = (1 - \frac{\tau}{k+t-\tau})\eta_t < \frac{1}{2}\eta_t$ when $k \geq 3\tau$ for all $t \geq 0$, it yields $\|w_t - \hat{w}_t\|^2 < (1 + \sqrt{\Delta}\tau) D^2 \tau \frac{1}{4}\eta_t^2 m_t^2$. Hence $\sum_{t=0}^{\infty}[(L+\mu)\eta_t + 2L^2\eta_t^2 D]\|\hat{w}_t - w_t\|^2$ is at most

$$[(L+\mu)\eta_0 + 2L^2\eta_0^2 D](1 + \sqrt{\Delta}\tau) D^2 \tau \sum_{t=0}^{\infty} \eta_t^2 m_t^2.$$

Combining $m_t \le m_0 \exp(LD\sum_{i=0}^t \eta_i)$ (see (51)) and $\eta_i = \frac{1}{LD(2+\beta)(k+i)}$ yields

$$m_t \le m_0 \exp(\frac{1}{2+\beta}\sum_{i=1}^t \frac{1}{i}) \le m_0 \exp(\frac{1}{2+\beta}(1+\ln t)) \le m_0 \exp(\frac{1}{2+\beta})t^{\frac{1}{2+\beta}}.$$

The second inequality is a property of harmonic number $H_t = \sum_{i=1}^t \frac{1}{i} \le 1 + \ln t$. Hence,

$$\eta_t m_t \le \frac{1}{L(2+\beta)t} m_0 \exp(\frac{1}{2+\beta})t^{\frac{1}{2+\beta}} \le \frac{m_0 \exp(\frac{1}{2+\beta})}{L(2+\beta)} t^{\frac{-(1+\beta)}{2+\beta}}.$$

Hence, we obtain

$$(\eta_t m_t)^2 \le [\frac{m_0 \exp(\frac{1}{2+\beta})}{L(2+\beta)}]^2 t^{\frac{-(2+2\beta)}{2+\beta}} \le [\frac{m_0 \exp(\frac{1}{2+\beta})}{L(2+\beta)}]^2 t^{-(1+\rho)},$$

where $\rho = \frac{\beta}{2+\beta}$. Due to the property of over-harmonic series, $\sum_{t=1}^\infty \frac{1}{t^{1+\rho}}$ converges for any $\rho > 0$. In other words, $\sum_{t=0}^\infty (\eta_t m_t)^2$ is finite or $\sum_{t=0}^\infty [(L+\mu)\eta_t + 2L^2\eta_t^2 D]\|\hat{w}_t - w_t\|^2$ is finite. ■

D.6. Convergence of Large Stepsizes

Theorem 9 *Let Assumptions 1, 2, and 3 hold. Consider Algorithm 1 with a stepsize sequence such that $\eta_t \le \frac{1}{2L}$, $\eta_t \to 0$, $\frac{d}{dt}\eta_t \le 0$ and $\sum_{t=0}^\infty \eta_t \to \infty$. Then,*

$$\mathbb{E}[\|w_{t+1} - w_*\|^2] \to 0.$$

Proof As shown in (24)

$$\mathbb{E}[\|w_{t+1} - w_*\|^2] \le (1-\mu\eta_t)\mathbb{E}[\|w_t - w_*\|^2] + \eta_t^2 N,$$

when $\eta_t \le \frac{1}{2L}$.

Let $Y_{t+1} = \mathbb{E}[\|w_{t+1} - w_*\|^2]$, $Y_t = \mathbb{E}[\|w_t - w_*\|^2]$, $\beta_t = 1 - \mu\eta_t$ and $\gamma_t = \eta_t^2 N$. As proved in Lemma 9, if $Y_{t+1} \le \beta_t Y_t + \gamma_t$, then

$$\begin{aligned} Y_{t+1} &\le \sum_{i=0}^t [\prod_{j=i+1}^t \beta_j]\gamma_i + (\prod_{i=0}^t \beta_i) Y_1 \\ &= \sum_{i=0}^t [\prod_{j=i+1}^t (1-\mu\eta_j)]\gamma_i + [\prod_{i=0}^t (1-\mu\eta_i)]\mathbb{E}[\|w_1 - w_*\|^2] \end{aligned}$$

Let us define

$$n(j) = \mu\eta_j. \tag{52}$$

Since $1 - x \le \exp(-x)$ for all $x \ge 0$,

$$\prod_{j=i+1}^t (1-\mu\eta_j) \le \exp(-\sum_{j=i+1}^t (\mu\eta_j)) = \exp(-\sum_{j=i+1}^t n(j)).$$

Furthermore, since $n(j)$ is decreasing in j, we have

$$\sum_{j=i+1}^t n(j) \ge \int_{x=i+1}^{t+1} n(x)dx.$$

These two inequalities can be used to derive

$$\begin{aligned} Y_{t+1} &\le \sum_{i=0}^t \exp(-\sum_{j=i+1}^t n(j))n(i)^2 N + \exp(-\sum_{j=0}^t n(j))Y_0 \\ &\le \sum_{i=0}^t \exp(-\int_{x=i+1}^{t+1} n(x)dx)n(i)^2 N + \exp(-\int_{x=0}^{t+1} n(x)dx)Y_0 \\ &= \sum_{i=0}^t \exp(-[M(t+1) - M(i+1)])n(i)^2 N + \exp(-M(t+1))Y_0, \end{aligned}$$

where

$$M(y) = \int_{x=0}^y n(x)dx \text{ and } \frac{d}{dy}M(y) = n(y).$$

We focus on

$$F = \sum_{i=0}^t \exp(-[M(t+1) - M(i+1)])n(i)^2.$$

We notice that

$$F = \exp(-M(t+1))\sum_{i=0}^t \exp(M(i+1))n(i)^2.$$

We know that $\exp(M(x+1))$ increases and $n(x)^2$ decreases, hence, in the most general case either their product first decreases and then starts to increase or their product keeps on increasing. We first discuss the decreasing and increasing case. Let $a(x) = \exp(M(x+1))n(x)^2$ denote this product and let integer $j \ge 0$ be such that $a(0) \ge a(1) \ge \ldots \ge a(j)$ and $a(j) \le a(j+1) \le a(j+2) \le \ldots$ (notice that $j = 0$ expresses the situation where $a(i)$ only increases). Function $a(x)$ for $x \ge 0$ is minimized for some value h in $[j, j+1)$. For $1 \le i \le j$, $a(i) \le \int_{x=i-1}^i a(x)dx$, and for $j+1 \le i$, $a(i) \le \int_{x=i}^{i+1} a(x)dx$. This yields the upper bound

$$\begin{aligned} \sum_{i=0}^t a(i) &= a(0) + \sum_{i=1}^j a(i) + \sum_{i=j+1}^t a(i) \\ &\le a(0) + \int_{x=0}^j a(x)dx + \int_{x=j+1}^{t+1} a(x)dx, \\ &\le a(0) + \int_{x=0}^{t+1} a(x)dx. \end{aligned}$$

The same upper bound holds for the other case as well, i.e., if $a(i)$ is only decreasing. We conclude

$$F \le \exp(-M(t+1))[\exp(M(2))n(0)^2 + \int_{x=0}^{j+1} \exp(M(x+1))n(x)^2 dx].$$

Combined with

$$M(x+1) = \int_{y=0}^{x+1} n(y)dy \le \int_{y=0}^{x} n(y)dy + n(x) = M(x) + n(x)$$

we obtain

$$\begin{aligned} F &\le \exp(-M(t+1))[\exp(M(2))n(0)^2 + \int_{x=0}^{t+1} \exp(M(x))n(x)^2 \exp(n(x))dx] \\ &\le \exp(-M(t+1))[\exp(M(2))n(0)^2 + \exp(n(0)) \int_{x=0}^{t+1} \exp(M(x))n(x)^2 dx]. \end{aligned}$$

This gives

$$\begin{aligned} Y_{t+1} &\le \exp(-M(t+1))[\exp(M(1))n(0)^2 + \exp(n(0)) \int_{x=0}^{t+1} \exp(M(x))n(x)^2 dx]N \\ &\quad + \exp(-M(t+1))Y_0 \\ &= N\exp(n(0))C(t+1) + \exp(-M(t+1))[\exp(M(1))n(0)^2 N + Y_0], \end{aligned} \tag{53}$$

where

$$C(t) = \exp(-M(t)) \int_{x=0}^{t} \exp(M(x))n(x)^2 dx.$$

For $y \le t$, we derive (notice that $n(x)$ is decreasing)

$$\begin{aligned} C(t) &= \exp(-M(t)) \int_{x=0}^{t} \exp(M(x))n(x)^2 dx \\ &= \exp(-M(t)) \int_{x=0}^{y} \exp(M(x))n(x)^2 dx + \exp(-M(t)) \int_{x=y}^{t} \exp(M(x))n(x)^2 dx \\ &\le \exp(-M(t)) \int_{x=0}^{y} \exp(M(x))n(x)^2 dx + \exp(-M(t)) \int_{x=y}^{t} \exp(M(x))n(x)n(y) dx \\ &= \exp(-M(t)) \int_{x=0}^{y} \exp(M(x))n(x)^2 dx + \exp(-M(t))n(y) \int_{x=y}^{t} \exp(M(x))n(x) dx \\ &= \exp(-M(t)) \int_{x=0}^{y} \exp(M(x))n(x)^2 dx + n(y)[1 - \exp(-M(t))\exp(M(y))] \\ &\le \exp(-M(t)) \int_{x=0}^{y} \exp(M(x))n(x)^2 dx + n(y). \end{aligned}$$

Let $\epsilon > 0$. Since $n(y) \to 0$ as $y \to \infty$, there exists a y such that $n(y) \le \epsilon/2$. Since $M(t) \to \infty$ as $t \to \infty$, $\exp(-M(t)) \to 0$ as $t \to \infty$. Hence, there exists a T such that for $t \ge T$, $\exp(-M(t)) \int_{x=0}^{y} \exp(M(x))n(x)^2 dx \le \epsilon/2$. This implies $C(t) \le \epsilon$ for $t \ge T$. This proves $C(t) \to 0$ as $t \to \infty$, and we conclude $Y_t \to 0$ as $t \to \infty$. ■

Theorem 10 *Let Assumptions 1, 2, and 3 hold. Consider Algorithm 1 with a stepsize sequence such that $\eta_t \le \frac{1}{2L}$, $\eta_t \to 0$, $\frac{d}{dt}\eta_t \le 0$, and $\sum_{t=0}^{\infty} \eta_t \to \infty$. Then,*

$$\begin{aligned} \mathbb{E}[\|w_{t+1} - w_*\|^2] &\le N\exp(n(0))2n(M^{-1}(\ln[\frac{n(t+1)}{n(0)}] + M(t+1))) \\ &\quad + \exp(-M(t+1))[\exp(M(1))n(0)^2 N + \mathbb{E}[\|w_0 - w_*\|^2]], \end{aligned}$$

where $n(t) = \mu\eta_t$ and $M(t) = \int_{x=0}^{t} n(x)dx$.

Proof We are ready to compute the convergence rate of $Y_t = \mathbb{E}[\|w_t - w_*\|^2]$ for a given $M(t)$. We have shown that $C(t) \le \exp(-M(t)) \int_{x=0}^{y} \exp(M(x))n(x)^2 dx + n(y)$. We are interested in the following problem: finding the largest $y \le t$ such as

$$\exp(-M(t))[\int_{x=0}^{y} \exp(M(x))n(x)^2 dx] \le n(y).$$

The solution is equal to

$$y = \sup\{y \le t : \exp(-M(t))[\int_{x=0}^{y} \exp(M(x))n(x)^2 dx] \le n(y)\}.$$

Since $M(x)$ always decreases,

$$\begin{aligned} y &\ge \sup\{y \le t : \exp(-M(t))[\int_{x=0}^{y} \exp(M(x))n(x)n(0)dx] \le n(y)\} \\ &= \sup\{y \le t : \exp(-M(t))n(0)[\exp(M(y)) - \exp(M(0))] \le n(y)\} \\ &= \sup\{y \le t : \exp(M(y)) \le \exp(M(0)) + \frac{n(y)}{n(0)}\exp(M(t))\} \\ &\ge \sup\{y \le t : \exp(M(y)) \le \exp(M(0)) + \frac{n(t)}{n(0)}\exp(M(t))\} \\ &\ge \sup\{y \le t : \exp(M(y)) \le \frac{n(t)}{n(0)}\exp(M(t))\} \\ &= \sup\{y \le t : M(y) \le \ln[\frac{n(t)}{n(0)}] + M(t)\} \\ &= M^{-1}(\ln[\frac{n(t)}{n(0)}] + M(t)), \end{aligned}$$

where $M^{-1}(t)$ exists for $t \in (0, n(0)]$ (since $M(y)$ strictly increases and maps into $(0, n(0)]$ for $y \ge 0$).

Therefore,

$$C(t) \le 2n(M^{-1}(\ln[\frac{n(t)}{n(0)}] + M(t)))$$

and

$$Y_{t+1} \le N\exp(n(0))2n(M^{-1}(\ln[\frac{n(t+1)}{n(0)}] + M(t+1))) + \exp(-M(t+1))[\exp(M(1))n(0)^2 N + Y_0].$$

■

Theorem 11 *Among all stepsizes $\eta_{q,t} = 1/(K+t)^q$ where $q > 0$, K is a constant such that $\eta_{q,t} \le \frac{1}{2L}$, SGD algorithm enjoys the fastest convergence with stepsize $\eta_{1,t} = 1/(2L+t)$.*

Proof In (53) we have

$$\mathbb{E}[\|w_t - w_*\|^2] \le AC(t) + B\exp(-M(t)),$$

where $A = N\exp(n(0))$ and $B = \exp(M(1))n(0)^2N + \mathbb{E}[\|w_0 - w_*\|^2]$. Let us denote $C_q(t) = C(t), n_q(t) = \mu/(K+t)^q$ where $\eta_{q,t} = 1/(K+t)^q$. It is obvious that $n_q(t) > n_1(t)$ for all t and $q < 1$. It implies for any $n_q(t)$ with $q < 1$,

$$\exp(-M(t)) < \exp(-\int_{x=0}^{t} \mu 1/(K+x)dx) < \exp(-\int_{x=0}^{t} \mu 1/xdx) < 1/t.$$

Therefore, we always have $\exp(-M(t)) < 1/t = n_1(t) < n_q(t) < C_q(t)$. Now, we consider the following case. We find $n(t)$ such that $n(t) = C(t)/2$. We rewrite this as

$$C(t) = \exp(-M(t))\int_{x=0}^{t} \exp(M(x))n(x)^2dx = 2n(t).$$

Taking derivatives of both sides, we have:

$$-n^2 = n(n-2n) = n(n-C) = \frac{d}{dt}C = 2\frac{d}{dt}n.$$

This is solved for $1/(at) : -1/(a^2t^2) = -2/(at^2)$ Hence, $a = 1/2$ and $n(t) = 2/t$. It means, $C_q(t) > C_1(t)$ and thus, the stepsize $\eta_{1,t} = 1/(K+t)$ enjoys the fastest convergence. ■

D.7. Convergence of Large Stepsizes in Batch Mode

We first derive a couple lemmas which will help us deriving our main bounds. In what follows let Assumptions 1, 2 and 3 hold for all lemmas.

Lemma 14 *Let us define $f(w;(\xi_1,\dots,\xi_k)) = \frac{1}{k}\sum_{i=1}^k f(w;\xi_i)$, then we have the following properties:*

$$\mathbb{E}[f(w;(\xi_1,\dots,\xi_k))] = F(w),$$
$$\mathbb{E}[\|\nabla f(w_*;(\xi_1,\dots,\xi_k))\|^2] = \frac{\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]}{k},$$

and

$$\mathbb{E}[\|\nabla f(w;(\xi_1,\dots,\xi_k))\|^2] \le 4L[F(w) - F(w_*)] + \frac{N}{k}$$

Proof The expectation of $f(w;(\xi_1,\dots,\xi_k))$ is equal to

$$\mathbb{E}[f(w;(\xi_1,\dots,\xi_k))] = \frac{1}{k}\sum_{i=1}^k \mathbb{E}[f(w;\xi_i)] = F(w). \tag{54}$$

Now we write $\mathbb{E}[\|\nabla f(w_*;(\xi_1,\dots,\xi_k))\|^2]$ as $\mathbb{E}[\sum_{j=1}^d(\frac{1}{k}\sum_{i=1}^k[\nabla f(w_*;\xi_i)]_j)^2]$. This is equal to

$$\mathbb{E}[\sum_{j=1}^d\{\frac{1}{k^2}\sum_{i=1}^k[\nabla f(w_*;\xi_i)]_j^2 + \frac{1}{k}\sum_{i_0\neq i_1}[\nabla f(w_*;\xi_{i_0})]_j[\nabla f(w_*;\xi_{i_1})]_j\}]$$
$$=\mathbb{E}[\sum_{j=1}^d\frac{1}{k^2}\sum_{i=1}^k[\nabla f(w_*;\xi_i)]_j^2] + \mathbb{E}[\sum_{j=1}^d\frac{1}{k}\sum_{i_0\neq i_1}[\nabla f(w_*;\xi_{i_0})]_j[\nabla f(w_*;\xi_{i_1})]_j].$$

The first term $\mathbb{E}[\sum_{j=1}^d\frac{1}{k^2}\sum_{i=1}^k[\nabla f(w_*;\xi_i)]_j^2]$ is equal to

$$\sum_{j=1}^d\frac{1}{k^2}\sum_{i=1}^k\mathbb{E}[[\nabla f(w_*;\xi_i)]_j^2] = \frac{1}{k^2}\sum_{i=1}^k\mathbb{E}[\sum_{j=1}^d[\nabla f(w_*;\xi_i)]_j^2] = \frac{1}{k^2}\sum_{i=1}^k\mathbb{E}[\|\nabla f(w_*;\xi_i)\|^2]$$
$$= \frac{1}{k}E[\|\nabla f(w_*;\xi)\|^2].$$

The second term $\mathbb{E}[\sum_{j=1}^d\frac{1}{k}\sum_{i_0\neq i_1}[\nabla f(w_*;\xi_{i_0})]_j[\nabla f(w_*;\xi_{i_1})]_j]$ is equal to

$$\sum_{j=1}^d\frac{1}{k}\sum_{i_0\neq i_1}\mathbb{E}[[\nabla f(w_*;\xi_{i_0})]_j]\cdot\mathbb{E}[[\nabla f(w_*;\xi_{i_1})]_j]$$
$$= \sum_{j=1}^d\frac{1}{k}\sum_{i_0\neq i_1}[\mathbb{E}[\nabla f(w_*;\xi_{i_0})]]_j\cdot[\mathbb{E}[\nabla f(w_*;\xi_{i_1})]]_j.$$

Note that $\mathbb{E}[\nabla f(w_*;\xi_{i_0})] = \nabla\mathbb{E}[f(w_*;\xi_i)] = \nabla F(w_*) = 0$. This means that the second term is equal to 0 and we conclude

$$\mathbb{E}[\|\nabla f(w_*;(\xi_1,\dots,\xi_k))\|^2] = \frac{\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]}{k}. \tag{55}$$

We have the following fact:

$$\|\nabla f(w;(\xi_1,\dots,\xi_k)) - \nabla f(w_*;(\xi_1,\dots,\xi_k))\|^2 = \frac{1}{k^2}[\|\sum_{i=1}^k(\nabla f(w;\xi_i) - \nabla f(w_*;\xi_i))\|^2]$$
$$\le \frac{1}{k}\sum_{i=1}^k\|\nabla f(w;\xi_i) - \nabla f(w_*;\xi_i)\|^2$$
$$= \|\nabla f(w;\xi_i) - \nabla f(w_*;\xi_i)\|^2.$$

Since $\mathbb{E}[\|\nabla f(w;\xi) - \nabla f(w_*;\xi)\|^2] \le 2L[F(w) - F(w_*)]$ (see (18)), we obtain

$$\mathbb{E}[\|\nabla f(w;(\xi_1,\dots,\xi_k)) - \nabla f(w_*;(\xi_1,\dots,\xi_k))\|^2] \le 2L[F(w) - F(w_*)].$$

By using a similar argument as in Lemma 1 we can derive

$$\mathbb{E}[\|\nabla f(w;(\xi_1,\dots,\xi_k))\|^2] \le 4L[F(w) - F(w_*)] + 2\mathbb{E}[\|\nabla f(w_*;(\xi_1,\dots,\xi_k))\|^2]$$
$$\overset{(55)}{\le} 4L[F(w) - F(w_*)] + 2\frac{\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]}{k}$$
$$= 4L[F(w) - F(w_*)] + \frac{N}{k}, \tag{56}$$

where $N = 2\mathbb{E}[\|\nabla f(w_*;\xi)\|^2]$. ■

We define

$$\mathcal{F}_t = \sigma(w_0, \xi_0', u_0, \dots, \xi_{t-1}', u_{t-1}),$$

where

$$\xi'_i = (\xi_{i,1}, \ldots, \xi_{i,k_i}).$$

We consider the following general algorithm with the following gradient updating rule:

$$w_{t+1} = w_t - \eta_t d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t),$$

where $f(w_t; \xi'_t) = \frac{1}{k_t} \sum_{i=1}^{k_t} f(w_t; \xi_{t,i})$.

Lemma 15 *We have*

$$\mathbb{E}[\|d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t)\|^2 | \mathcal{F}_t, \xi'_t] \leq D \|\nabla f(w_t; \xi'_t)\|^2$$

and

$$\mathbb{E}[d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t) | \mathcal{F}_t] = \nabla F(w_t).$$

Proof For the first bound, if we take the expectation of $\|d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t)\|^2$ with respect to u_t, then we have (for vectors x we denote the value of its i-th position by $[x]_i$)

$$\begin{aligned}
\mathbb{E}[\|d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t)\|^2 | \mathcal{F}_t, \xi'_t] &= d_{\xi'_t}^2 \sum_u p_{\xi'_t}(u) \|S_u^{\xi'_t} \nabla f(w_t; \xi'_t)\|^2 \\
&= d_{\xi'_t}^2 \sum_u p_{\xi'_t}(u) \sum_{i \in S_u^{\xi'_t}} [\nabla f(w_t; \xi'_t)]_i^2 \\
&= d_{\xi'_t} \sum_{i \in D_{\xi'_t}} [\nabla f(w_t; \xi'_t)]_i^2 = d_{\xi'_t} \|f(w_t; \xi'_t)\|^2 \leq D \|\nabla f(w_t; \xi'_t)\|^2,
\end{aligned}$$

where the transition to the second line follows from (27).

For the second bound, if we take the expectation of $d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t)$ wrt u_t, then we have:

$$\mathbb{E}[d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t) | \mathcal{F}_t, \xi'_t] = d_{\xi'_t} \sum_u p_{\xi'_t}(u) S_u^{\xi'_t} \nabla f(w_t; \xi'_t) = \nabla f(w_t; \xi'_t),$$

and this can be used to derive

$$\mathbb{E}[d_{\xi'_t} S_{u_t}^{\xi'_t} f(w_t; \xi'_t) | \mathcal{F}_t] = \mathbb{E}[\mathbb{E}[d_{\xi'_t} S_{u_t}^{\xi'_t} f(w_t; \xi'_t) | \mathcal{F}_t, \xi'_t] | \mathcal{F}_t] = \mathbb{E}[\nabla f(w_t; \xi'_t)] = \nabla F(w_t).$$

The last equality comes from (54). ■

Lemma 16 *Let Assumptions 1, 2 and 3 hold, $0 < \eta_t \leq \frac{1}{2LD}$ for all $t \geq 0$. Then,*

$$\mathbb{E}[\|w_{t+1} - w_*\|^2] \leq (1 - \mu\eta_t) \|w_t - w_*\|^2 + \eta_t^2 \frac{ND}{k_t}.$$

Proof Since $w_{t+1} = w_t - \eta_t d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t)$, we have

$$\|w_{t+1} - w_*\|^2 = \|w_t - w_*\|^2 - 2\eta_t \langle d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t), (w_t - w_*) \rangle + \eta_t^2 \|d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t)\|^2.$$

We now take expectations over u_t and ξ_t and use Lemmas 15 and 14:

$$\begin{aligned}
&\mathbb{E}[\|w_{t+1} - w_*\|^2 | \mathcal{F}_t] \\
&\leq \|w_t - w_*\|^2 - 2\eta_t \langle \nabla F(w), w_t - w_* \rangle + \eta_t^2 D \mathbb{E}[\|\nabla f(w_t; \xi'_t)\|^2 | \mathcal{F}_t]
\end{aligned}$$

By (1), we have

$$-\langle \nabla F(w), w_t - w_* \rangle \leq -[F(w) - F(w_*)] - \mu/2 \|w_t - w_*\|^2 \tag{57}$$

Thus, $\mathbb{E}[\|w_{t+1} - w_*\|^2 | \mathcal{F}_t]$ is at most

$$\overset{(57)}{\leq} \|w_t - w_*\|^2 - \mu\eta_t \|w_t - w_*\|^2 - 2\eta_t [F(w) - F(w_*)] + \eta_t^2 D \mathbb{E}[\|\nabla f(w_t; \xi'_t)\|^2 | \mathcal{F}_t].$$

Since $\mathbb{E}[\|\nabla f(w_t; \xi'_t)\|^2 | \mathcal{F}_t] \leq 4L[F(w) - F(w_*)] + \frac{N}{k_t}$ (see (56)), $\mathbb{E}[\|w_{t+1} - w_*\|^2 | \mathcal{F}_t]$ is at most

$$\overset{(56)}{\leq} (1 - \mu\eta_t) \|w_t - w_*\|^2 - 2\eta_t (1 - 2\eta_t LD)[F(w) - F(w_*)] + \eta_t^2 \frac{ND}{k_t}.$$

Using the condition $\eta_t \leq \frac{1}{2LD}$ yields the lemma. ■

As shown above,

$$\mathbb{E}[\|w_{t+1} - w_*\|^2] \leq (1 - \mu\eta_t) \mathbb{E}[\|w_t - w_*\|^2] + \frac{\eta_t^2 ND}{k_t},$$

when $\eta_t \leq \frac{1}{2LD}$.

Let $Y_{t+1} = \mathbb{E}[\|w_{t+1} - w_*\|^2]$, $Y_t = \mathbb{E}[\|w_t - w_*\|^2]$, $\beta_t = 1 - \mu\eta_t$ and $\gamma_t = \frac{\eta_t^2 ND}{k_t}$. As proved in Lemma 9, if $Y_{t+1} \leq \beta_t Y_t + \gamma_t$, then

$$\begin{aligned}
Y_{t+1} &\leq \sum_{i=0}^{t} [\prod_{j=i+1}^{t} \beta_j] \gamma_i + (\prod_{i=0}^{t} \beta_i) Y_0 \\
&= \sum_{i=0}^{t} [\prod_{j=i+1}^{t} (1 - \mu\eta_j)] \gamma_i + [\prod_{i=0}^{t} (1 - \mu\eta_i)] \mathbb{E}[\|w_0 - w_*\|^2]
\end{aligned}$$

Let us define $n(j) = \mu n_j$ and $M(y) = \int_{x=0}^{y} n(x) dx$ as in Section 4.

Theorem 12 *Let Assumptions 1, 2 and 3 hold, $\{\eta_t\}$ is a diminishing sequence with conditions $\sum_{t=0}^{\infty} \eta_t \to \infty$ and $0 < \eta_t \leq \frac{1}{2LD}$ for all $t \geq 0$. Then, the sequence $\{w_t\}$ converges to w_* where*

$$w_{t+1} = w_t - \eta_t d_{\xi'_t} S_{u_t}^{\xi'_t} \nabla f(w_t; \xi'_t).$$

Proof To prove the convergence of w_t, we only need to prove the convergence of

$$C(t) = \exp(-M(t)) \int_{x=0}^{t} \exp(M(x)) \frac{n(x)^2}{k(x)} dx.$$

Let T denote the total number of gradient computations and define $K(t) = \int_{x=1}^{t} k(x)dx = T$; we have $t = K^{-1}(T)$ and $\frac{dK(x)}{dx} = k(x)$. We define $y = K(x)$ or $x = K^{-1}(y)$ with $dy = k(x)dx$. We write

$$\begin{aligned}
C(t) &= \exp(-M(t)) \int_{x=0}^{t} \exp(M(x)) \frac{n(x)^2}{k(x)} dx \\
&= \exp(-M(K^{-1}(T))) \int_{K(0)}^{K^{-1}(T)} \exp(M(K^{-1}(y))) \frac{n(x)^2}{k(x)^2} k(x) dx \\
&\leq \exp(-M(K^{-1}(T))) \int_{1}^{K^{-1}(T)} \exp(M(K^{-1}(y))) \frac{n(x)^2}{k(x)^2} k(x) dx.
\end{aligned}$$

The last inequality is based on the fact that $K(0) \geq 1$.

Let us define $n'(x) = \frac{n(x)}{k(x)}$ and using the fact that $dy = k(x)dx$, we obtain

$$\begin{aligned}
C(t) &= C(K^{-1}(T)) \\
&\leq \exp(-M(K^{-1}(T))) \int_{1}^{K^{-1}(T)} \exp(M(K^{-1}(y)))[n'(K^{-1}(y))]^2 dy.
\end{aligned}$$

Since $K^{-1}(y) = x$, we have $\frac{dK^{-1}(y)}{dy} = \frac{1}{k(x)}$ where $y = K(x)$. This implies $\frac{dM(K^{-1}(y))}{dy} = \frac{n(K^{-1}(y))}{k(K^{-1}(y))} = n'(K^{-1}(y))$. Hence, by denoting

$$\begin{aligned}
C'(t) &= C(K^{-1}(t)), \\
n'(x) &= \frac{n(x)}{k(x)}, \\
M'(x) &= M(K^{-1}(x)),
\end{aligned}$$

we can convert the general problem into the problem of Section D.6. This implies that the analysis of $C(t)$ in Section D.6 can directly apply to analyze $C(K^{-1}(T))$. Since we already proved the convergence of $C(t)$ in Section D.6, we obtain the theorem. ■

Journal of Machine Learning Research 20 (2019) 1-81 Submitted 11/18; Revised 8/19; Published 12/19

All Models are Wrong, but *Many* are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously

Aaron Fisher AFISHE27@ALUMNI.JH.EDU
Takeda Pharmaceuticals
Cambridge, MA 02139, USA

Cynthia Rudin CYNTHIA@CS.DUKE.EDU
Departments of Computer Science and Electrical and Computer Engineering
Duke University
Durham, NC 27708, USA

Francesca Dominici FDOMINIC@HSPH.HARVARD.EDU
Department of Biostatistics
Harvard T.H. Chan School of Public Health
Boston, MA 02115, USA

(Authors are listed in order of contribution, with highest contribution listed first.)

Editor: Maya Gupta

Abstract

Variable importance (VI) tools describe how much covariates contribute to a prediction model's accuracy. However, important variables for one well-performing model (for example, a linear model $f(\mathbf{x}) = \mathbf{x}^T\beta$ with a fixed coefficient vector β) may be unimportant for another model. In this paper, we propose model class reliance (MCR) as the range of VI values across *all* well-performing model in a prespecified class. Thus, MCR gives a more comprehensive description of importance by accounting for the fact that many prediction models, possibly of different parametric forms, may fit the data well. In the process of deriving MCR, we show several informative results for permutation-based VI estimates, based on the VI measures used in Random Forests. Specifically, we derive connections between permutation importance estimates for a *single* prediction model, U-statistics, conditional variable importance, conditional causal effects, and linear model coefficients. We then give probabilistic bounds for MCR, using a novel, generalizable technique. We apply MCR to a public data set of Broward County criminal records to study the reliance of recidivism prediction models on sex and race. In this application, MCR can be used to help inform VI for unknown, proprietary models.

Keywords: Rashomon, permutation importance, conditional variable importance, U-statistics, transparency, interpretable models

1. Introduction

Variable importance (VI) tools describe how much a prediction model's accuracy depends on the information in each covariate. For example, in Random Forests, VI is measured by the decrease in prediction accuracy when a covariate is permuted (Breiman, 2001; Breiman et al., 2001; see also Strobl et al., 2008; Altmann et al., 2010; Zhu et al., 2015; Gregorutti et al., 2015; Datta et al., 2016; Gregorutti et al., 2017). A similar "Perturb" VI measure has been used for neural networks, where noise is added to covariates (Recknagel et al., 1997; Yao et al., 1998; Scardi and Harding, 1999; Gevrey et al., 2003). Such tools can be useful for identifying covariates that must be measured with high precision, for improving the transparency of a "black box" prediction model (see also Rudin, 2019), or for determining what scenarios may cause the model to fail.

However, existing VI measures do not generally account for the fact that many prediction models may fit the data almost equally well. In such cases, the model used by one analyst may rely on entirely different covariate information than the model used by another analyst. This common scenario has been called the "Rashomon" effect of statistics (Breiman et al., 2001; see also Lecué, 2011; Statnikov et al., 2013; Tulabandhula and Rudin, 2014; Nevo and Ritov, 2017; Letham et al., 2016). The term is inspired by the 1950 Kurosawa film of the same name, in which four witnesses offer different descriptions and explanations for the same encounter. Under the Rashomon effect, how should analysts give comprehensive descriptions of the importance of each covariate? How well can one analyst recover the conclusions of another? Will the model that gives the best predictions necessarily give the most accurate interpretation?

To address these concerns, we analyze the *set* of prediction models that provide near-optimal accuracy, which we refer to as a *Rashomon set*. This approach stands in contrast to training to select a *single* prediction model, among a prespecified class of candidate models. Our motivation is that Rashomon sets (defined formally below) summarize the range of effective prediction strategies that an analyst might choose. Additionally, even if the candidate models do not contain the true data generating process, we may hope that some of these models function in similar ways to the data generating process. In particular, we may hope there exist well performing candidate models that place the same importance on a variable of interest as the underlying data generating process does. If so, then studying sets of well-performing models will allow us to deduce information about the data generating process.

Applying this approach to study variable importance, we define *model class reliance* (MCR) as the highest and lowest degree to which any well-performing model within a given class may rely on a variable of interest for prediction accuracy. Roughly speaking, MCR captures the range of explanations, or mechanisms, associated with well-performing models. Because the resulting range summarizes many prediction models simultaneously, rather a single model, we expect this range to be less affected by the choices that an individual analyst makes during the model-fitting process. Instead of reflecting these choices, MCR aims to reflect the nature of the prediction problem itself.

We make several, specific technical contributions in deriving MCR. First, we review a core measure of how much an individual prediction model relies on covariates of interest for its accuracy, which we call *model reliance* (MR). This measure is based on permutation importance measures for Random Forests (Breiman et al., 2001; Breiman, 2001), and can be expanded to describe conditional importance (see Section 8, as well as Strobl et al. 2008). We draw a connection between permutation-based importance estimates (MR) and U-statistics, which facilitates later theoretical results. Additionally, we derive connections

©2019 Aaron Fisher, Cynthia Rudin, and Francesca Dominici.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/18-760.html.

between MR, conditional causal effects, and coefficients for additive models. Expanding on MR, we propose MCR, which generalizes the definition of MR for a *class of models.* We derive finite-sample bounds for MCR, which motivate an intuitive estimator of MCR. Finally, we propose computational procedures for this estimator.

The tools we develop to study Rashomon sets are quite general, and can be used to make finite-sample inferences for arbitrary characteristics of well-performing models. For example, beyond describing variable importance, these tools can describe the range of risk predictions that well-fitting models assign to a particular covariate profile, or the variance of predictions made by well-fitting models. In some cases, these novel techniques may provide finite-sample confidence intervals (CIs) where none have previously existed (see Section 5).

MCR and the Rashomon effect become especially relevant in the context of criminal recidivism prediction. Proprietary recidivism risk models trained from criminal records data are increasingly being used in U.S. courtrooms. One concern is that these models may be relying on information that would otherwise be considered unacceptable (for example, race, sex, or proxies for these variables), in order to estimate recidivism risk. The relevant models are often proprietary, and cannot be studied directly. Still, in cases where the predictions made by these models are publicly available, it may be possible to identify alternative prediction models that are sufficiently similar to the proprietary model of interest.

In this paper, we specifically consider the proprietary model COMPAS (Correctional Offender Management Profiling for Alternative Sanctions), developed by the company Northpointe Inc. (subsequently, in 2017, Northpointe Inc.,Courtview Justice Solutions Inc., and Constellation Justice Systems Inc. joined together under the name Equivant). Our goal is to estimate how much COMPAS relies on either race, sex, or proxies for these variables not measured in our data set. To this end, we apply a broad class of flexible, kernel-based prediction models to predict COMPAS score. In this setting, the MCR interval reflects the highest and lowest degree to which any prediction model in our class can rely on race and sex while still predicting COMPAS score relatively accurately. Equipped with MCR, we can relax the common assumption of being able to correctly specify the unknown model of interest (here, COMPAS) up to a parametric form. Instead, rather than assuming that the COMPAS model itself is contained in our class, we assume that our class contains at least one well-performing alternative model that relies on sensitive covariates to the same degree that COMPAS does. Under this assumption, the MCR interval will contain the VI value for COMPAS. Applying our approach, we find that race, sex, and their potential proxy variables, are likely not the dominant predictive factors in the COMPAS score (see analysis and discussion in Section 10).

The remainder of this paper is organized as follows. In Section 2 we introduce notation, and give a high level summary of our approach, illustrated with visualizations. In Sections 3 and 4 we formally present MR and MCR respectively, and derive theoretical properties of each. We also review related variable importance practices in the literature, such as retraining a model after removing one of the covariates. In Section 5, we discuss general applicability of our approach for determining finite-sample CIs for other problems. In Section 6, we present a general procedure for computing MCR. In Section 7, we give specific implementations of this procedure for (regularized) linear models, and linear models in a reproducing kernel Hilbert space. We also show that, for additive models, MR can be expressed in terms of the model's coefficients. In Section 8 we outline connections between

MR, causal inference, and conditional variable importance. In Section 9, we illustrate MR and MCR with a simulated toy example, to aid intuition. We also present simulation studies for the task of estimating MR for an unknown, underlying conditional expectation function, under misspecification. We analyze a well-known public data set on recidivism in Section 10, described above. All proofs are presented in the appendices.

2. Notation & Technical Summary

The label of "variable importance" measure has been broadly used to describe approaches for either inference (van der Laan, 2006; Díaz et al., 2015; Williamson et al., 2017) or prediction. While these two goals are highly related, we primarily focus on how much prediction models rely on covariates to achieve accuracy. We use terms such as "model reliance" rather than "importance" to clarify this context.

In order to evaluate how much prediction models rely on variables, we now introduce notation for random variables, data, classes of prediction models, and loss functions for evaluating predictions. Let $Z = (Y, X_1, X_2) \in \mathcal{Z}$ be a random variable with outcome $Y \in \mathcal{Y}$ and covariates $X = (X_1, X_2) \in \mathcal{X}$, where the covariate subsets $X_1 \in \mathcal{X}_1$ and $X_2 \in \mathcal{X}_2$ may each be multivariate. We assume that observations of Z are *iid*, that $n \geq 2$, and that solutions to arg min and arg max operations exist whenever optimizing over sets mentioned in this paper (for example, in Theorem 4, below). Our goal is to study how much different prediction models rely on X_1 to predict Y.

We refer to our data set as $\mathbf{Z} = \begin{bmatrix} \mathbf{y} & \mathbf{X} \end{bmatrix}$, a matrix composed of a n-length outcome vector $\mathbf{y}$ in the first column, and a $n \times p$ covariate matrix $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \end{bmatrix}$ in the remaining columns. In general, for a given vector $\mathbf{v}$, let $\mathbf{v}_{[j]}$ denote its jth element(s). For a given matrix $\mathbf{A}$, let $\mathbf{A}'$, $\mathbf{A}_{[i,\cdot]}$, $\mathbf{A}_{[\cdot,j]}$, and $\mathbf{A}_{[i,j]}$ respectively denote the transpose of $\mathbf{A}$, the ith row(s) of $\mathbf{A}$, the jth column(s) of $\mathbf{A}$, and the element(s) in the ith row(s) and j^{th} column(s) of $\mathbf{A}$.

We use the term *model class* to refer to a prespecified subset $\mathcal{F} \subset \{f \mid f : \mathcal{X} \to \mathcal{Y}\}$ of the measurable functions from $\mathcal{X}$ to $\mathcal{Y}$. We refer to member functions $f \in \mathcal{F}$ as *prediction models,* or simply as *models.* Given a model f, we evaluate its performance using a nonnegative *loss function* $L : (\mathcal{F} \times \mathcal{Z}) \to \mathbb{R}_{\geq 0}$. For example, L may be the squared error loss $L_{\text{se}}(f, (y, x_1, x_2)) = (y - f(x_1, x_2))^2$ for regression, or the hinge loss $L_{\text{h}}(f, (y, x_1, x_2)) = (1 - yf(x_1, x_2))_+$ for classification. We use the term *algorithm* to refer to any procedure $\mathcal{A} : \mathcal{Z}^n \to \mathcal{F}$ that takes a data set as input and returns a model $f \in \mathcal{F}$ as output.

2.1. Summary of Rashomon Sets & Model Class Reliance

Many traditional statistical estimates come from descriptions of a *single,* fitted prediction model. In contrast, in this section, we summarize our approach for studying a *set* of near-optimal models. To define this set, we require a prespecified "reference" model, denoted by f_{ref}, to serve as a benchmark for predictive performance. For example, f_{ref} may come from a flowchart used to predict injury severity in a hospital's emergency room, or from another quantitative decision rule that is currently implemented in practice. Given a reference model f_{ref}, we define a *population ϵ-Rashomon set* as the subset of models

with expected loss no more than ϵ above that of f_{ref}. We denote this set as $\mathcal{R}(\epsilon) := \{f \in \mathcal{F} : \mathbb{E}L(f, Z) \leq \mathbb{E}L(f_{\text{ref}}, Z) + \epsilon\}$, where $\mathbb{E}$ denotes expectations with respect to the population distribution. This set can be thought of as representing models that might be arrived at due to differences in data measurement, processing, filtering, model parameterization, covariate selection, or other analysis choices (see Section 4).

Illustrations of Rashomon Sets & Model Class Reliance

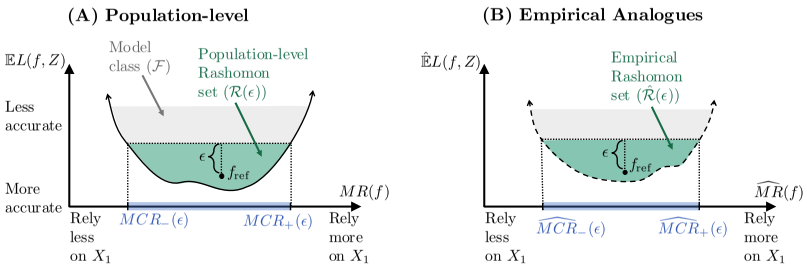


Figure 1: Rashomon sets and model class reliance – Panel (A) illustrates a hypothetical Rashomon set $\mathcal{R}(\epsilon)$, within a model class $\mathcal{F}$. The y-axis shows the expected loss of each model $f \in \mathcal{F}$, and the x-axis shows how much each model f relies on X_1 (defined formally in Section 3). Along the x-axis, the population-level MCR range is highlighted in blue, showing the values of MR corresponding to well-performing models (see Section 4). Panel (B) shows the in-sample analogue of Panel (A). Here, the y-axis denotes the in-sample loss, $\hat{\mathbb{E}}L(f, Z) := \frac{1}{n}\sum_{i=1}^{n} L(f, \mathbf{Z}_{[i,\cdot]})$; the x-axis shows the empirical model reliance of each model $f \in \mathcal{F}$ on X_1 (see Section 3); and the highlighted portion of the x-axis shows empirical MCR (see Section 4).

Figure 1-A illustrates a hypothetical example of a population ϵ-Rashomon set. Here, the y-axis shows the expected loss of each model $f \in \mathcal{F}$, and the x-axis shows how much each model relies on X_1 for its predictive accuracy. More specifically, given a prediction model f, the x-axis shows the percent increase in f's expected loss when noise is added to X_1. We refer to this measure as the *model reliance* (MR) of f on X_1, written informally as

$$MR(f) := \frac{\text{Expected loss of } f \text{ under noise}}{\text{Expected loss of } f \text{ without noise}}. \tag{2.1}$$

The added noise must satisfy certain properties, namely, it must render X_1 completely uninformative of the outcome Y, without altering the marginal distribution of X_1 (for details, see Section 3, as well as Breiman, 2001; Breiman et al., 2001).

Our central goal is to understand how much, or how little, models may rely on covariates of interest (X_1) while still predicting well. In Figure 1-A, this range of possible MR values

is shown by the highlighted interval along the x-axis. We refer to an interval of this type as a *population-level model class reliance* (MCR) range (see Section 4), formally defined as

$$[MCR_-(\epsilon),\ MCR_+(\epsilon)] := \left[\min_{f \in \mathcal{R}(\epsilon)} MR(f),\ \max_{f \in \mathcal{R}(\epsilon)} MR(f)\right]. \tag{2.2}$$

To estimate this range, we use empirical analogues of the population ϵ-Rashomon set, and of MR, based on observed data (Figure 1-B). We define an *empirical ϵ-Rashomon set* as the set of models with *in-sample* loss no more than ϵ above that of f_{ref}, and denote this set by $\hat{\mathcal{R}}(\epsilon)$. Informally, we define the *empirical* MR of a model f on X_1 as

$$\widehat{MR}(f) := \frac{\text{In-sample loss of } f \text{ under noise}}{\text{In-sample loss of } f \text{ without noise}}, \tag{2.3}$$

that is, the extent to which f appears to rely on X_1 in a given sample (see Section 3 for details). Finally, we define the *empirical model class reliance* as the range of empirical MR values corresponding to models with strong in-sample performance (see Section 4), formally written as

$$[\widehat{MCR}_-(\epsilon),\ \widehat{MCR}_+(\epsilon)] := \left[\min_{f \in \hat{\mathcal{R}}(\epsilon)} \widehat{MR}(f),\ \max_{f \in \hat{\mathcal{R}}(\epsilon)} \widehat{MR}(f)\right]. \tag{2.4}$$

In Figure 1-B, the above range is shown by the highlighted portion of the x-axis.

We make several technical contributions in the process of developing MCR.

- **Estimation of MR, and population-level MCR:** Given f, we show desirable properties of $\widehat{MR}(f)$ as an estimator of $MR(f)$, using results for U-statistics (Section 3.1 and Theorem 5). We also derive finite sample bounds for population-level MCR, some of which require a limit on the complexity of $\mathcal{F}$ in the form of a covering number. These bounds demonstrate that, under fairly weak conditions, empirical MCR provides a sensible estimate of population-level MCR (see Section 4 for details).

- **Computation of empirical MCR:** Although empirical MCR is fully determined given a sample, the minimization and maximization in Eq 2.4 require nontrivial computations. To address this, we outline a general optimization procedure for MCR (Section 6). We give detailed implementations of this procedure for cases when the model class $\mathcal{F}$ is a set of (regularized) linear regression models, or a set of regression models in a reproducing kernel Hilbert space (Section 7). The output of our proposed procedure is a closed-form, convex envelope containing $\mathcal{F}$, which can be used to approximate empirical MCR for any performance level ϵ (see Figure 2 for an illustration). Still, for complex model classes where standard empirical loss minimization is an open problem (for example, neural networks), computing empirical MCR remains an open problem as well.

- **Interpretation of MR in terms of model coefficients, and causal effects:** We show that MR for an additive model can be written as a function of the model's coefficients (Proposition 15), and that MR for a binary covariate X_1 can be written as a function of the conditional causal effects of X_1 on Y (Proposition 19).

- **Extensions to conditional importance:** We provide an extension of MR that is analogous to the notion of conditional importance (Strobl et al., 2008). This extension describes how much a model relies on the specific information in X_1 that cannot otherwise be gleaned from X_2 (Section 8.2).
- **Generalizations for Rashomon sets:** Beyond notions of variable importance, we also generalize our finite sample results for MCR to describe arbitrary characterizations of models in a population ϵ-Rashomon set. As we discuss in concurrent work (Coker et al., 2018), this generalization is analogous to the profile likelihood interval, and can, for example, be used to bound the range of risk predictions that well-performing prediction models may assign to a particular set of covariates (Section 5).

We begin in the next section by formally reviewing model reliance.

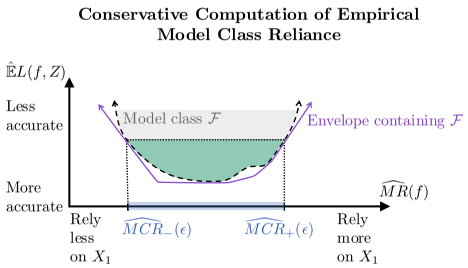


Figure 2: Illustration of output from our empirical MCR computational procedure – Our computation procedure produces a closed-form, convex envelope that contains $\mathcal{F}$ (shown above as the solid, purple line), which bounds empirical MCR for any value of ϵ (see Eq 2.4). The procedure works sequentially, tightening these bounds as much as possible near the ϵ value of interest (Section 6). The results from our data analysis (Figure 8) are presented in the same format as the above purple envelope.

3. Model Reliance

To formally describe how much the expected accuracy of a fixed prediction model f relies on the random variable X_1, we use the notion of a "switched" loss where X_1 is rendered uninformative. Throughout this section, we will treat f as a pre-specified prediction model of interest (as in Hooker, 2007). Let $Z^{(a)} = (Y^{(a)}, X_1^{(a)}, X_2^{(a)})$ and $Z^{(b)} = (Y^{(b)}, X_1^{(b)}, X_2^{(b)})$ be independent random variables, each following the same distribution as $Z = (Y, X_1, X_2)$. We define

$$e_{\text{switch}}(f) := \mathbb{E}L\{f, (Y^{(b)}, X_1^{(a)}, X_2^{(b)})\}$$

as representing the expected loss of model f across pairs of observations $(Z^{(a)}, Z^{(b)})$ in which the values of $X_1^{(a)}$ and $X_1^{(b)}$ have been switched. To see this interpretation of the above equation, note that we have used the variables $(Y^{(b)}, X_2^{(b)})$ from $Z^{(b)}$, but we have used the variable $X_1^{(b)}$ from an independent copy $Z^{(b)}$. This is why we say that $X_1^{(a)}$ and $X_1^{(b)}$ have been switched; the values of $(Y^{(b)}, X_1^{(a)}, X_2^{(b)})$ do not relate to each other as they would if they had been chosen together. An alternative interpretation of $e_{\text{switch}}(f)$ is as the expected loss of f when noise is added to X_1 in such a way that X_1 becomes completely uninformative of Y, but that the marginal distribution of X_1 is unchanged.

As a reference point, we compare $e_{\text{switch}}(f)$ against the standard expected loss when none of the variables are switched, $e_{\text{orig}}(f) := \mathbb{E}L(f, (Y, X_1, X_2))$. From these two quantities, we formally define *model reliance* (MR) as the ratio,

$$MR(f) := \frac{e_{\text{switch}}(f)}{e_{\text{orig}}(f)}, \tag{3.1}$$

as we alluded to in Eq 2.1. Higher values of $MR(f)$ signify greater reliance of f on X_1. For example, an $MR(f)$ value of 2 means that the model relies heavily on X_1, in the sense that its loss doubles when X_1 is scrambled. An $MR(f)$ value of 1 signifies no reliance on X_1, in the sense that the model's loss does not change when X_1 is scrambled. Models with reliance values strictly less than 1 are more difficult to interpret, as they rely less on the variable of interest than a random guess. Interestingly, it is possible to have models with reliance less than one. For instance, a model f' may satisfy $MR(f') < 1$ if it treats X_1 and Y as positively correlated when they are in fact negatively correlated. However, in many cases, the existence of a model $f' \in \mathcal{F}$ satisfying $MR(f') < 1$ implies the existence of another, better performing model $f'' \in \mathcal{F}$ satisfying $MR(f'') = 1$ and $e_{\text{orig}}(f'') \leq e_{\text{orig}}(f')$. That is, although models may exist with MR values less than 1, they will typically be suboptimal (see Appendix A.2).

Model reliance could alternatively be defined as a difference rather than a ratio, that is, as $MR_{\text{difference}}(f) := e_{\text{switch}}(f) - e_{\text{orig}}(f)$. In Appendix A.5, we discuss how many of our results remain similar under either definition.

3.1. Estimating Model Reliance with U-statistics, and Connections to Permutation-based Variable Importance

Given a model f and data set $\mathbf{Z} = \begin{bmatrix} \mathbf{y} & \mathbf{X} \end{bmatrix}$, we estimate $MR(f)$ by separately estimating the numerator and denominator of Eq 3.1. We estimate $e_{\text{orig}}(f)$ with the standard empirical loss,

$$\hat{e}_{\text{orig}}(f) := \frac{1}{n} \sum_{i=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[i,\cdot]})\}. \tag{3.2}$$

We estimate $e_{\text{switch}}(f)$ by performing a "switch" operation across all observed pairs, as in

$$\hat{e}_{\text{switch}}(f) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} L\{f, (\mathbf{y}_{[j]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]})\}. \tag{3.3}$$

Above, we have aggregated over all possible combinations of the observed values for (Y, X_2) and for X_1, excluding pairings that are actually observed in the original sample. If

the summation over all possible pairs (Eq 3.3) is computationally prohibitive due to sample size, another estimator of $e_{\text{switch}}(f)$ is

$$\hat{e}_{\text{divide}}(f) := \frac{1}{2\lfloor n/2 \rfloor} \sum_{i=1}^{\lfloor n/2 \rfloor} \left[L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[i+\lfloor n/2 \rfloor,\cdot]}, \mathbf{X}_{2[i,\cdot]})\} \right. \tag{3.4}$$

$$\left. + L\{f, (\mathbf{y}_{[i+\lfloor n/2 \rfloor]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[i+\lfloor n/2 \rfloor,\cdot]})\} \right]. \tag{3.5}$$

Here, rather than summing over all pairs, we divide the sample in half. We then match the first half's values for (Y, X_2) with the second half's values for X_1 (Line 3.4), and vice versa (Line 3.5). All three of the above estimators (Eqs 3.2, 3.3 & 3.5) are unbiased for their respective estimands, as we discuss in more detail shortly.

Finally, we can estimate $MR(f)$ with the plug-in estimator

$$\widehat{MR}(f) := \frac{\hat{e}_{\text{switch}}(f)}{\hat{e}_{\text{orig}}(f)}, \tag{3.6}$$

which we define as the *empirical model reliance* of f on X_1. In this way, we formalize the empirical MR definition in Eq 2.3.

Again, our definition of empirical MR is very similar to the permutation-based variable importance approach of Breiman (2001), where Breiman uses a single random permutation and we consider all possible pairs. To compare these two approaches more precisely, let $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_{n!}\}$ be a set of n-length vectors, each containing a different permutation of the set $\{1, \ldots, n\}$. The approach of Breiman (2001) is analogous to computing the loss $\sum_{i=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[\boldsymbol{\pi}_{l[i]},\cdot]}, \mathbf{X}_{2[i,\cdot]})\}$ for a randomly chosen permutation vector $\boldsymbol{\pi}_l \in \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_{n!}\}$. Similarly, our calculation in Eq 3.3 is proportional to the sum of losses over all possible $(n!)$ permutations, excluding the n unique combinations of the rows of $\mathbf{X}_1$ and the rows of $\begin{bmatrix} \mathbf{X}_2 & \mathbf{y} \end{bmatrix}$ that appear in the original sample (see Appendix A.3). Excluding these observations is necessary to preserve the (finite-sample) unbiasedness of $\hat{e}_{\text{switch}}(f)$.

The estimators $\hat{e}_{\text{orig}}(f)$, $\hat{e}_{\text{switch}}(f)$ and $\hat{e}_{\text{divide}}(f)$ all belong to the well-studied class of U-statistics. Thus, under fairly minor conditions, these estimators are unbiased, asymptotically normal, and have finite-sample probabilistic bounds (Hoeffding, 1948, 1963; Serfling, 1980; see also DeLong et al., 1988 for an early use of U-statistics in machine learning, as well as caveats in Demler et al., 2012). To our knowledge, connections between permutation-based importance and U-statistics have not been previously established.

While the above results from U-statistics depend on the model f being fixed a priori, we can also leverage these results to create *uniform* bounds on the MR estimation error for all models in a sufficiently regularized class $\mathcal{F}$. We formally present this bound in Section 4 (Theorem 5), after introducing required conditions on model class complexity. The existence of this uniform bound implies that it is feasible to train a model and to evaluate its importance using the *same data*. This differs from the classical VI approach of Random Forests (Breiman, 2001), which avoids in-sample importance estimation. There, each tree in the ensemble is fit on a random subset of data, and VI for the tree is estimated using the held-out data. The tree-specific VI estimates are then aggregated to obtain a VI estimate for the overall ensemble. Although sample-splitting approaches such as this are helpful in many cases, the uniform bound for MR suggests that they are not strictly necessary, depending on the sample size and the complexity of $\mathcal{F}$.

3.2. Limitations of Existing Variable Importance Methods

Several common approaches for variable selection, or for describing relationships between variables, do not necessarily capture a variable's importance. Null hypothesis testing methods may identify a relationship, but do not describe the relationship's strength. Similarly, checking whether a variable is included by a sparse model-fitting algorithm, such as the Lasso (Hastie et al., 2009), does not describe the extent to which the variable is relied on. Partial dependence plots (Breiman et al., 2001; Hastie et al., 2009) can be difficult to interpret if multiple variables are of interest, or if the prediction model contains interaction effects.

Another common VI procedure is to run a model-fitting algorithm twice, first on all of the data, and then again after removing X_1 from the data set. The losses for the two resulting models are then compared to determine the importance, or "necessity," of X_1 (Gevrey et al., 2003). Because this measure is a function of two prediction models rather than one, it does not measure how much either individual model relies on X_1. We refer to this approach as measuring empirical *Algorithm Reliance* (AR) on X_1, as the model-fitting algorithm is the common attribute between the two models. Related procedures were proposed by Breiman et al. (2001); Breiman (2001), which measure the sufficiency of X_1.

As we discuss in Section 3.1, the permutation-based VI measure from RFs (Breiman, 2001; Breiman et al., 2001) forms the inspiration for our definition of MR. This RF VI measure has been the topic of empirical studies (Archer and Kimes, 2008; Calle and Urrea, 2010; Wang et al., 2016), and several variations of the measure have been proposed (Strobl et al., 2007, 2008; Altmann et al., 2010; Hapfelmeier et al., 2014). Mentch and Hooker (2016) use U-statistics to study predictions of ensemble models fit to subsamples, similar to the bootstrap aggregation used in RFs. Procedures related to "Mean Difference Impurity," another VI measure derived for RFs, have been studied theoretically by Louppe et al. (2013); Kazemitabar et al. (2017). All of this literature focuses on VI measures for RFs, for ensembles, or for individual trees. Our estimator for model reliance differs from the traditional RF VI measure (Breiman, 2001) in that we permute inputs to the overall model, rather than permuting the inputs to each individual ensemble member. Thus, our approach can be used generally, and is not limited to trees or ensemble models.

Outside of the context of RF VI, Zhu et al. (2015) propose an estimand similar to our definition of model reliance, and Gregorutti et al. (2015, 2017) propose an estimand analogous to $e_{\text{switch}}(f) - e_{\text{orig}}(f)$. These recent works focus on the model reliance of f on X_1 specifically when f is equal to the conditional expectation function of Y (that is, $f(x_1, x_2) = \mathbb{E}[Y|X_1 = x_1, X_2 = x_2]$). In contrast, we consider model reliance for arbitrary prediction models f. Datta et al. (2016) study the extent to which a model's predictions are expected to change when a subset of variables is permuted, regardless of whether the permutation affects a loss function L. These VI approaches are specific to a single prediction model, as is MR. In the next section, we consider a more general conception of importance: how much *any* model in a particular set may rely on the variable of interest.

4. Model Class Reliance

Like many statistical procedures, our MR measure (Section 3) produces a description of a *single* predictive model. Given a model with high predictive accuracy, MR describes how much the model's performance hinges on covariates of interest (X_1). However, there will often be many other models that perform similarly well, and that rely on X_1 to different degrees. With this notion in mind, we now study how much *any* well-performing model from a prespecified class $\mathcal{F}$ may rely on covariates of interest.

Recall from Section 2.1 that, in order to define a population ϵ-Rashomon set of near-optimal models, we must choose a "reference" model f_{ref} to serve as a performance benchmark. In order to discuss this choice, we now introduce more explicit notation for the population ϵ-Rashomon set, written as

$$\mathcal{R}(\epsilon, f_{\text{ref}}, \mathcal{F}) := \{f \in \mathcal{F} : e_{\text{orig}}(f) \leq e_{\text{orig}}(f_{\text{ref}}) + \epsilon\}. \tag{4.1}$$

Note that we write $\mathcal{R}(\epsilon, f_{\text{ref}}, \mathcal{F})$ and $\mathcal{R}(\epsilon)$ interchangeably when f_{ref} and $\mathcal{F}$ are clear from context. Similarly, we occasionally write empirical ϵ-Rashomon sets using the more explicit notation $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F}) := \{f \in \mathcal{F} : \hat{e}_{\text{orig}}(f) \leq \hat{e}_{\text{orig}}(f_{\text{ref}}) + \epsilon\}$, but typically abbreviate these sets as $\hat{\mathcal{R}}(\epsilon)$.

While f_{ref} could be selected by minimizing the in-sample loss, the theoretical study of $\mathcal{R}(\epsilon, f_{\text{ref}}, \mathcal{F})$ is simplified under the assumption that f_{ref} is prespecified. For example, f_{ref} may come from a flowchart used to predict injury severity in a hospital's emergency room, or from another quantitative decision rule that is currently implemented in practice. The model f_{ref} can also be selected using sample splitting. In some cases it may be desirable to fix f_{ref} equal to the best-in-class model $f^\star := \arg\min_{f \in \mathcal{F}} e_{\text{orig}}(f)$, but this is generally infeasible because $f^\star$ is unknown. Still, for any $f_{\text{ref}} \in \mathcal{F}$, the Rashomon set $\mathcal{R}(\epsilon, f_{\text{ref}}, \mathcal{F})$ defined using f_{ref} will always be conservative in the sense that it contains the Rashomon set $\mathcal{R}(\epsilon, f^\star, \mathcal{F})$ defined using $f^\star$.

We can now formalize our definitions of population-level MCR and empirical MCR by simply plugging in our definitions for $MR(f)$ and $\widehat{MR}(f)$ (Section 3) into Eqs 2.2 & 2.4 respectively. Studying population-level MCR (Eq 2.2) is the main focus of this paper, as it provides a more comprehensive view of importance than measures from a single model. If $MCR_+(\epsilon)$ is low, then *no* well-performing model in $\mathcal{F}$ places high importance on X_1, and X_1 can be discarded at low cost regardless of future modeling decisions. If $MCR_-(\epsilon)$ is large, then *every* well-performing model in $\mathcal{F}$ must rely substantially on X_1, and X_1 should be given careful attention during the modeling process. Here, $\mathcal{F}$ may itself consist of several parametric model forms (for example, all linear models and all decision tree models with less than 6 single-split nodes). We stress that the range $[MCR_-(\epsilon), MCR_+(\epsilon)]$ does not depend on the fitting algorithm used to select a model $f \in \mathcal{F}$. The range is valid for any algorithm producing models in $\mathcal{F}$, and applies for any $f \in \mathcal{F}$.

In the remainder of this section, we derive finite sample bounds for population-level MCR, from which we argue that empirical MCR provides reasonable estimates of population-level MCR (Section 4.1). In Appendix B.7 we consider an alternate formulation of Rashomon sets and MCR where we replace the relative loss threshold in the definition of $\mathcal{R}(\epsilon)$ with an absolute loss threshold. This alternate formulation can be similar in practice, but still requires the specification of a reference function f_{ref} to ensure that $\mathcal{R}(\epsilon)$ and $\hat{\mathcal{R}}(\epsilon)$ are nonempty.

4.1. Motivating Empirical Estimators of MCR by Deriving Finite-sample Bounds

In this section we derive finite-sample, probabilistic bounds for $MCR_+(\epsilon)$ and $MCR_-(\epsilon)$. Our results imply that, under minimal assumptions, $\widehat{MCR}_+(\epsilon)$ and $\widehat{MCR}_-(\epsilon)$ are respectively within a neighborhood of $MCR_+(\epsilon)$ and $MCR_-(\epsilon)$ with high probability. However, the weakness of our assumptions (which are typical for statistical-learning-theoretic analysis) renders the width of our resulting CIs to be impractically large, and so we use these results only to show conditions under which $\widehat{MCR}_+(\epsilon)$ and $\widehat{MCR}_-(\epsilon)$ form sensible point estimates. In Sections 9.1 & 10, below, we apply a bootstrap procedure to account for sampling variability.

To derive these results we introduce three bounded loss assumptions, each of which can be assessed empirically. Let $b_{\text{orig}}, B_{\text{ind}}, B_{\text{ref}}, B_{\text{switch}} \in \mathbb{R}$ be known constants.

Assumption 1 *(Bounded individual loss) For a given model $f \in \mathcal{F}$, assume that $0 \leq L(f, (y, x_1, x_2)) \leq B_{ind}$ for any $(y, x_1, x_2) \in (\mathcal{Y} \times \mathcal{X}_1 \times \mathcal{X}_2)$.*

Assumption 2 *(Bounded relative loss) For a given model $f \in \mathcal{F}$, assume that $|L(f, (y, x_1, x_2)) - L(f_{ref}, (y, x_1, x_2))| \leq B_{ref}$ for any $(y, x_1, x_2) \in \mathcal{Z}$.*

Assumption 3 *(Bounded aggregate loss) For a given model $f \in \mathcal{F}$, assume that $\mathbb{P}\{0 < b_{orig} \leq \hat{e}_{orig}(f)\} = \mathbb{P}\{\hat{e}_{switch}(f) \leq B_{switch}\} = 1$.*

Each assumption is a property of a specific model $f \in \mathcal{F}$. The notation B_{ind} and B_{ref} refer to bounds for any individual observation, and the notation b_{orig} and B_{switch} refer to bounds on the aggregated loss L in a sample. These boundedness assumptions are central to our finite sample guarantees, shown below.

Crucially, loss functions L that are unbounded in general may be used so long as $L(f, (y, x_1, x_2))$ is bounded on a particular domain. For example, the squared-error loss can be used if $\mathcal{Y}$ is contained within a known range, and predictions $f(x_1, x_2)$ are contained within the same range for $(x_1, x_2) \in \mathcal{X} \times \mathcal{X}_2$. We give example methods of determining B_{ind} in Sections 7.3.2 & 7.4.2. For Assumption 3, we can approximate b_{orig} by training a highly flexible model to the data, and setting b_{orig} equal to half (or any positive fraction) of the resulting cross-validated loss. To determine B_{switch} we can simply set $B_{\text{switch}} = B_{\text{ind}}$, although this may be conservative. For example, in the case of binary classification models for non-separable groups (see Section 9.1), no linear classifier can misclassify all observations, particularly after a covariate is permuted. Thus, it must hold that $B_{\text{ind}} > B_{\text{switch}}$. Similarly, if f_{ref} satisfies Assumption 1, then B_{ref} may be conservatively set equal to B_{ind}. If model reliance is redefined as a difference rather than a ratio, then a similar form of the results in this section will apply without Assumption 3 (see Appendix A.5).

Based on these assumptions, we can create a finite-sample upper bound for $MCR_+(\epsilon)$ and lower bound for $MCR_-(\epsilon)$. In other words, we create an "outer" bound that contains the interval $[MCR_-(\epsilon), MCR_+(\epsilon)]$ with high probability.

Theorem 4 *("Outer" MCR Bounds) Given a constant $\epsilon \geq 0$, let $f_{+,\epsilon} \in \arg\max_{\mathcal{R}(\epsilon)} MR(f)$ and $f_{-,\epsilon} \in \arg\min_{\mathcal{R}(\epsilon)} MR(f)$ be prediction models that attain the highest and lowest model*

reliance among models in $\mathcal{R}(\epsilon)$. If $f_{+,\epsilon}$ and $f_{-,\epsilon}$ satisfy Assumptions 1, 2 & 3, then

$$\mathbb{P}\left(MCR_+(\epsilon) > \widehat{MCR}_+(\epsilon_{out}) + \mathcal{Q}_{out}\right) \leq \delta, \text{ and} \tag{4.2}$$

$$\mathbb{P}\left(MCR_-(\epsilon) < \widehat{MCR}_-(\epsilon_{out}) - \mathcal{Q}_{out}\right) \leq \delta, \tag{4.3}$$

where $\epsilon_{out} := \epsilon + 2B_{ref}\sqrt{\frac{\log(3\delta^{-1})}{2n}}$, *and* $\mathcal{Q}_{out} := \frac{B_{switch}}{b_{orig}} - \frac{B_{switch} - B_{ind}\sqrt{\frac{\log(6\delta^{-1})}{n}}}{b_{orig} + B_{ind}\sqrt{\frac{\log(6\delta^{-1})}{2n}}}$.

Eq 4.2 states that, with high probability, $MCR_+(\epsilon)$ is no higher than $\widehat{MCR}_+(\epsilon_{\text{out}})$ added to an error term $\mathcal{Q}_{\text{out}}$. As n increases, ϵ_{out} approaches ϵ and $\mathcal{Q}_{\text{out}}$ approaches zero. One practical implication is that, roughly speaking, if $\widehat{MCR}_+(\epsilon) \approx \widehat{MCR}_+(\epsilon_{\text{out}})$, then the empirical estimator $\widehat{MCR}_+(\epsilon)$ is unlikely to substantially underestimate $MCR_+(\epsilon)$. By similar reasoning, we can conclude from Eq 4.3 that if $\widehat{MCR}_-(\epsilon) \approx \widehat{MCR}_-(\epsilon_{\text{out}})$, then $\widehat{MCR}_-(\epsilon)$ is unlikely to substantially overestimate $MCR_-(\epsilon)$. By setting $\epsilon = 0$, Theorem 4 can also be used to create a finite-sample bound for the reliance of the unique (unknown) best-in-class model on X_1 (see Corollary 22 in Appendix A.4), although describing individual models is not the main focus of this paper.

We provide a visual illustration of Theorem 4 in Figure 3. A brief sketch of the proof is as follows. First, we enlarge the empirical ϵ-Rashomon set by increasing ϵ to ϵ_{out}, such that, by Hoeffding's inequality, $f_{+,\epsilon} \in \hat{\mathcal{R}}(\epsilon_{\text{out}})$ with high probability. When $f_{+,\epsilon} \in \hat{\mathcal{R}}(\epsilon_{\text{out}})$, we know that $\widehat{MR}(f_{+,\epsilon}) \leq \widehat{MCR}_+(\epsilon_{\text{out}})$ by the definition of $\widehat{MCR}_+(\epsilon_{\text{out}})$. Next, the term $\mathcal{Q}_{\text{out}}$ leverages finite-sample results for U-statistics to account for estimation error of $MR(f_{+,\epsilon}) = MCR_+(\epsilon)$ when using the estimator $\widehat{MR}(f_{+,\epsilon})$. Thus, we can relate $\widehat{MR}(f_{+,\epsilon})$ to both $\widehat{MCR}_+(\epsilon_{\text{out}})$ and $MCR_+(\epsilon)$ in order to obtain Eq 4.2. Similar steps can be applied to obtain Eq 4.3.

The bounds in Theorem 4 naturally account for potential overfitting without an explicit limit on model class complexity (such as a covering number, Rademacher complexity, or VC dimension). Instead, these bounds depend on being able to fully optimize MR across sets in the form of $\hat{\mathcal{R}}(\epsilon)$. If we allow our model class $\mathcal{F}$ to become more flexible, then the size of $\hat{\mathcal{R}}(\epsilon)$ will also increase. Because the bounds in Theorem 4 result from optimizing over $\hat{\mathcal{R}}(\epsilon)$, increasing the size of $\hat{\mathcal{R}}(\epsilon)$ results in wider, more conservative bounds. In this way, Eqs 4.2 and 4.3 implicitly capture model class complexity.

So far, Theorem 4 lets us bound the range of MR values corresponding to models that predict well, but it does not tell us whether these bounds are actually attained. Similarly, we can conclude from Theorem 4 that $[MCR_-(\epsilon), MCR_+(\epsilon)]$ is unlikely to exceed the estimated range $[\widehat{MCR}_-(\epsilon), \widehat{MCR}_+(\epsilon)]$ by a substantial margin, but we cannot determine whether this estimated range is unnecessarily wide. For example, consider the models that drive the $\widehat{MCR}_+(\epsilon)$ estimator: the models with strong in-sample accuracy, and high empirical reliance on X_1. These models' in-sample performance could merely be the result of overfitting, in which case they do not tell us direct information about $\mathcal{R}(\epsilon)$. Alternatively, even if all of these models truly do perform well on expectation (that is, even if they are contained in $\mathcal{R}(\epsilon)$), the model with the highest empirical reliance on X_1 may merely be the model for which our empirical MR estimate contains the most error. Either of these scenarios can cause $\widehat{MCR}_+(\epsilon)$ to be unnecessarily high, relative to $MCR_+(\epsilon)$.

Fortunately, both problematic scenarios are solved by requiring a limit on the complexity of $\mathcal{F}$. We propose a complexity measure in the form of a covering number, which allows us control a worst case scenario of either overfitting or MR estimation error. Specifically, we define the set of functions $\mathcal{G}_r$ as an *r-margin-expectation-cover* if for any $f \in \mathcal{F}$ and any distribution D, there exists $g \in \mathcal{G}_r$ such that

$$\mathbb{E}_{Z\sim D}\left|L\left(f, Z\right) - L\left(g, Z\right)\right| \leq r. \tag{4.4}$$

We define the *covering number* $\mathcal{N}(\mathcal{F}, r)$ to be the size of the smallest r-margin-expectation-cover for $\mathcal{F}$. In general, we use $\mathbb{P}_{V\sim D}$ and $\mathbb{E}_{V\sim D}$ to denote probabilities and expectations with respect to a random variable V following the distribution D. We abbreviate these quantities accordingly when V or D are clear from context, for example, as $\mathbb{P}_D$, $\mathbb{P}_V$, or simply $\mathbb{P}$. Unless otherwise stated, all expectations and probabilities are taken with respect to the (unknown) population distribution.

We first show that this complexity measure allows us to control the worst case MR estimation error, that is, the covering number $\mathcal{N}(\mathcal{F}, r)$ provides a uniform bound on the error of $\widehat{MR}(f)$ for all $f \in \mathcal{F}$.

Theorem 5 *(Uniform bound for $\widehat{MR}$) Given $r > 0$, if Assumptions 1 and 3 hold for all $f \in \mathcal{F}$, then*

$$\mathbb{P}\left[\sup_{f\in\mathcal{F}}\left|\widehat{MR}(f) - MR(f)\right| > q(\delta, r, n)\right] \leq \delta,$$

where

$$q(\delta, r, n) := \frac{B_{switch}}{b_{orig}} - \frac{B_{switch} - \left\{B_{ind}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r\sqrt{2}))}{n}} + 2r\sqrt{2}\right\}}{b_{orig} + \left\{B_{ind}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{2n}} + 2r\right\}}. \tag{4.5}$$

Theorem 5 states that, with high probability, the largest possible estimation error for $MR(f)$ across all models in $\mathcal{F}$ is bounded by $q(\delta, r, n)$, which can be made arbitrarily small by increasing n and decreasing r. As we noted in Section 3.1, this means that it is possible to train a model and estimate its reliance on variables without using sample-splitting.

The covering number $\mathcal{N}(\mathcal{F}, r)$ can also be used to limit the extent of overfitting (see Appendix B.5.1). As a result, it is possible to set an in-sample performance threshold low enough so that it will only be met by models with strong expected performance (that is, by models truly within $\mathcal{R}(\epsilon)$). To implement this idea of a stricter performance threshold, we contract the empirical ϵ-Rashomon set by subtracting a buffer term from ϵ. This requires that we generalize the definition of an empirical ϵ-Rashomon set to $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F}) := \{f_{\text{ref}}\} \cup \{f \in \mathcal{F} : \hat{e}_{\text{orig}}(f) \leq \hat{e}_{\text{orig}}(f_{\text{ref}}) + \epsilon\}$ for $\epsilon \in \mathbb{R}$, where the explicit inclusion of f_{ref} now ensures that $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F})$ is nonempty, even for $\epsilon < 0$. As before, we typically omit the notation f_{ref} and $\mathcal{F}$, writing $\hat{\mathcal{R}}(\epsilon)$ instead.

We are now prepared to answer the questions of whether the bounds from Theorem 4 are actually attained, and of whether the estimated range $[\widehat{MCR}_-(\epsilon), \widehat{MCR}_+(\epsilon)]$ is unnecessarily wide. Our answer comes in the form of an upper bound on $MCR_-(\epsilon)$, and a lower bound on $MCR_+(\epsilon)$.

Theorem 6 *("Inner" MCR Bounds) Given constants $\epsilon \geq 0$ and $r > 0$, if Assumptions 1, 2 and 3 hold for all $f \in \mathcal{F}$, and then*

$$\mathbb{P}\left(MCR_+(\epsilon) < \widehat{MCR}_+(\epsilon_{in}) - \mathcal{Q}_{in}\right) \leq \delta, \text{ and} \tag{4.6}$$

$$\mathbb{P}\left(MCR_-(\epsilon) > \widehat{MCR}_-(\epsilon_{in}) + \mathcal{Q}_{in}\right) \leq \delta, \tag{4.7}$$

where $\epsilon_{in} := \epsilon - 2B_{ref}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} - 2r$, and $\mathcal{Q}_{in} = q\left(\frac{\delta}{2}, r, n\right)$, as defined in Eq 4.5.

Theorem 6 can allow us to infer an "inner" bound that is contained within the interval $[MCR_-(\epsilon), MCR_+(\epsilon)]$ with high probability. In Figure 3, we illustrate the result of Theorem 6, and give a sketch of the proof. This proof follows a similar structure to that of Theorem 4, but incorporates Theorem 5's uniform bound on MR estimation error ($\mathcal{Q}_{\text{in}}$ term), as well as an additional uniform bound on the probability that any model has in-sample loss too far from its expected loss (ϵ_{in} term).

A practical implication of Theorem 6 is that, roughly speaking, if $\widehat{MCR}_+(\epsilon_{\text{in}}) \approx \widehat{MCR}_+(\epsilon)$, then it is unlikely for the empirical estimator $\widehat{MCR}_+(\epsilon)$ to substantially underestimate $MCR_+(\epsilon)$. Taken together with Theorem 4, we can conclude that, if $\widehat{MCR}_+(\epsilon_{\text{in}}) \approx \widehat{MCR}_+(\epsilon_{\text{out}})$, then the estimator $\widehat{MCR}_+(\epsilon)$ is unlikely either to overestimate or to underestimate $MCR_+(\epsilon)$ by very much. In large samples, it may be plausible to expect the condition $\widehat{MCR}_+(\epsilon_{\text{in}}) \approx \widehat{MCR}_+(\epsilon_{\text{out}})$ to hold, since ϵ_{in} and ϵ_{out} both approach ϵ as n increases. In the same way, if $\widehat{MCR}_-(\epsilon_{\text{in}}) \approx \widehat{MCR}_-(\epsilon_{\text{out}})$, we can conclude from Eqs 4.3 & 4.7 that the empirical estimator $\widehat{MCR}_-(\epsilon)$ is unlikely to either overestimate or underestimate $MCR_-(\epsilon)$ by very much. For this reason, we argue that $\widehat{MCR}_-(\epsilon)$ and $\widehat{MCR}_+(\epsilon)$ form sensible estimates of population-level MCR – each is contained within a neighborhood of its respective estimand, with high probability. The secondary x-axis of Figure 3 gives an illustration of this argument.

5. Extensions of Rashomon Sets Beyond Variable Importance

In this section we generalize the Rashomon set approach beyond the study of MR. In Section 5.1, we create finite-sample CIs for other summary characterizations of near-optimal, or best-in-class models. The generalization also helps to illustrate a core aspect of the argument underlying Theorem 4: models with near-optimal performance in the population tend to have relatively good performance in random samples.

In Section 5.2, we review existing literature on near-optimal models.

5.1. Finite-sample Confidence Intervals from Rashomon Sets

Rather than describing how much a model relies on X_1, here we assume the analyst is interested in an arbitrary characteristic of a model. We denote this characteristic of interest as $\phi : \mathcal{F} \to \mathbb{R}$. For example, if $f_{\boldsymbol{\beta}}$ is the linear model $f_{\boldsymbol{\beta}}(x) = x'\boldsymbol{\beta}$, then ϕ may be defined as the norm of the associated coefficient vector (that is, $\phi(f_{\boldsymbol{\beta}}) = \|\boldsymbol{\beta}\|_2^2$) or the prediction $f_{\boldsymbol{\beta}}$ would assign given a specific covariate profile x_{new} (that is, $\phi(f_{\boldsymbol{\beta}}) = f_{\boldsymbol{\beta}}(x_{\text{new}})$).

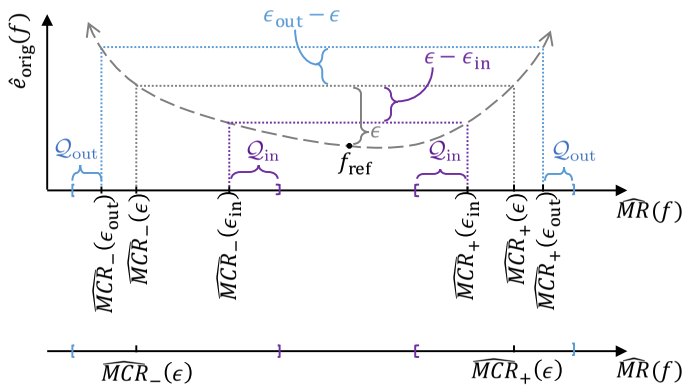


Figure 3: Illustration of terms in Theorems 4 and 6 – Above we show the relation between empirical MR (x-axis) and empirical loss (y-axis) for models f in a hypothetical model class $\mathcal{F}$. We mark f_{ref} by the black point. For each possible model reliance value $r \geq 0$, the curved, dashed line shows the lowest possible empirical loss for a function in $f \in \mathcal{F}$ satisfying $\widehat{MR}(f) = r$. The set $\hat{\mathcal{R}}(\epsilon)$ contains all models in $\mathcal{F}$ within the dotted gray lines. To create the bounds from Theorem 4, we expand the empirical ϵ-Rashomon set by increasing ϵ to ϵ_{out}, such that $f_{+,\epsilon}$ (or $f_{-,\epsilon}$) is contained in $\hat{\mathcal{R}}(\epsilon_{\text{out}})$ with high probability. We then add (or subtract) $\mathcal{Q}_{\text{out}}$ to account for estimation error of $\widehat{MR}(f_{+,\epsilon})$ (or $\widehat{MR}(f_{-,\epsilon})$). These steps are illustrated above in blue, with the final bounds shown by the blue bracket symbols along the x-axis. To create the bounds for $MCR_+(\epsilon)$ (and $MCR_-(\epsilon)$) in Theorem 6, we constrict the empirical ϵ-Rashomon set by decreasing ϵ to ϵ_{in}, such that all models with high expected loss are simultaneously excluded from $\hat{\mathcal{R}}(\epsilon_{\text{in}})$ with high probability. We then subtract (or add) $\mathcal{Q}_{\text{in}}$ to simultaneously account for MR estimation error for models in $\hat{\mathcal{R}}(\epsilon_{\text{in}})$. These steps are illustrated above in purple, with the final bounds shown by the purple bracket symbols along the x-axis. For emphasis, below this figure we show a copy of the x-axis with selected annotations, from which it is clear that $\widehat{MCR}_-(\epsilon)$ and $\widehat{MCR}_+(\epsilon)$ are always within the bounds produced by Theorems 4 and 6. With high probability, $\widehat{MCR}_-(\epsilon)$ and $\widehat{MCR}_+(\epsilon)$ are within a neighborhood of $MCR_-(\epsilon)$ and $MCR_+(\epsilon)$ respectively.

Given a descriptor ϕ, we now show a general result that allows creation of finite-sample CIs for the best performing models $\mathcal{R}(\epsilon)$. The resulting CIs are themselves based on empirical Rashomon sets.

Proposition 7 *(Finite sample CIs from Rashomon sets) Let* $\epsilon' := \epsilon + 2B_{ref}\sqrt{\frac{\log(2\delta^{-1})}{2n}}$*, let* $\hat{\phi}_-(\epsilon') := \min_{f\in\hat{\mathcal{R}}(\epsilon')} \phi(f)$ *and let* $\hat{\phi}_+(\epsilon') := \max_{f\in\hat{\mathcal{R}}(\epsilon')} \phi(f)$*.*

If Assumption 2 holds for all $f \in \mathcal{R}(\epsilon)$*, then*

$$\mathbb{P}\left[\{\phi(f) : f \in \mathcal{R}(\epsilon)\} \subseteq \left[\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')\right]\right] \geq 1 - \delta.$$

Proposition 7 generates a finite-sample CI for the range of values $\phi(f)$ corresponding to well-performing models, $\{\phi(f) : f \in \mathcal{R}(\epsilon)\}$. This CI, denoted by $\left[\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')\right]$, can itself be interpreted as the range of values $\phi(f)$ corresponding to models f with empirical loss not substantially above that of f_{ref}. Thus, the interval has both a rigorous coverage rate and a coherent in-sample interpretation. The proof of Proposition 7 uses Hoeffding's inequality to show that models in $\mathcal{F}$ are contained in $\hat{\mathcal{R}}(\epsilon')$ with high probability, that is, that models with good expected performance tend to perform well in random samples.

An immediate corollary of Proposition 7 is that we can generate finite-sample CIs for all best-in-class models $f^\star \in \arg\min_{f\in\mathcal{F}} \mathbb{E}L(f, Z)$ by setting $\epsilon = 0$. This corollary can be further strengthened if a single model $f^\star$ is assumed to uniquely minimize $\mathbb{E}L(f, Z)$ over $f \in \mathcal{F}$ (see Appendix B.6).

Note that Proposition 7 implicitly assumes that $\phi(f)$ can be determined exactly for any model $f \in \mathcal{F}$, in order for the interval $\left[\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')\right]$ to be precisely determined. This assumption does not hold, for example, if $\phi(f) = MR(f)$, or if $\phi(f) = \text{Var}\{f(X_1, X_2)\}$, as these quantities depend on both f and the (unknown) population distribution. In such cases, an additional correction factor must be incorporated to account for estimation error of $\phi(f)$, such as the $\mathcal{Q}_{\text{out}}$ term in Theorem 4.

In concurrent work, Coker et al. (2018) show that profile likelihood intervals take the same form as the interval $[\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')]$ in Proposition 7. This means that a profile likelihood interval can also be expressed by minimizing and maximizing over an empirical Rashomon set. More specifically, consider the case where the loss function L is the negative of the *known* log likelihood function, and where f_{ref} is the maximum likelihood estimate of the "true model," which in this case is $f^\star$. If additional minor assumptions are met (see Appendix A.6 for details), then the $(1-\delta)$-level profile likelihood interval for $\phi(f^\star)$ is equal to $[\hat{\phi}_-(\frac{\chi_{1,1-\delta}}{2n}), \hat{\phi}_+(\frac{\chi_{1,1-\delta}}{2n})]$, where $\hat{\phi}_-$ and $\hat{\phi}_+$ are defined as in Proposition 7, and $\chi_{1,1-\delta}$ is the $1-\delta$ percentile of a chi-square distribution with 1 degree of freedom.

Relative to a profile likelihood approach, the advantage of Proposition 7 is that it does not require asymptotics, it does not require that the likelihood be known up to a parametric form, and it can be extended to study the *set* of near-optimal prediction models $\mathcal{R}(\epsilon)$, rather than a single, potentially misspecified prediction model $f^\star$. This is especially useful when different near-optimal models accurately describe different aspects of the underlying data generating process, but none capture it completely. The disadvantage of Proposition 7 is that the required performance threshold of $\epsilon' = \epsilon + 2B_{\text{ref}}\sqrt{\frac{\log(2\delta^{-1})}{2n}}$ decreases more slowly than the performance threshold of $\frac{\chi_{1,1-\delta}}{2n}$ required in a profile likelihood interval. Because

our results from Section 4.1 carry a similar disadvantage, we use these results primarily to motivate point estimates describing the Rashomon set $\mathcal{R}(\epsilon)$.

Still, it is worth emphasizing the generality of Proposition 7. Through this result, Rashomon sets allow us to reframe a wide set of finite-sample inference problems as in-sample optimization problems. The implied CIs are not necessarily in closed form, but the approach still opens an exciting pathway for deriving non-asymptotic results. For example, they imply that existing methods for profile likelihood intervals might be able to be reapplied to achieve finite-sample results. For highly complex model classes where profile likelihoods are difficult to compute, such as neural networks or random forests, approximate inference is sometimes achieved via approximate optimization procedures (for example, Markov chain Monte Carlo for Bayesian additive regression trees, in Chipman et al., 2010). Proposition 7 shows that similar approximate optimization methods *could be repurposed to establish approximate, finite-sample inferences for the same model classes.*

5.2. Related Literature on the Rashomon Effect

Breiman et al. (2001) introduced the "Rashomon effect" of statistics as a problem of ambiguity: if many models fit the data well, it is unclear which model we should try to interpret. Breiman suggests that the ensembling many well-performing models together can resolve this ambiguity, as the new ensemble model may perform better than any of its individual members. However, this approach may only push the problem from the member level to the ensemble level, as there may also be many different ensemble models that fit the data well.

The Rashomon effect has also been considered in several subject areas outside of VI, including those in non-statistical academic disciplines (Heider, 1988; Roth and Mehta, 2002). Tulabandhula and Rudin (2014) optimize a decision rule to perform well under the predicted range of outcomes from any well-performing model. Statnikov et al. (2013) propose an algorithm to discover multiple Markov boundaries, that is, minimal sets of covariates such that conditioning on any one set induces independence between the outcome and the remaining covariates. Nevo and Ritov (2017) report interpretations corresponding to a set of well-fitting, sparse linear models. Meinshausen and Bühlmann (2010) estimate structural aspects of an underlying model (such as the variables included in that model) based on how stable those aspects are across a set of well-fitting models. This set of well-fitting models is identified by repeating an estimation procedure in a series of perturbed samples, using varying levels of regularization (see also Azen et al., 2001). Letham et al. (2016) search for a pair of well-fitting dynamical systems models that give maximally different predictions.

6. Calculating Empirical Estimates of Model Class Reliance

In this section, we propose a binary search procedure to bound the values of $\widehat{MCR}_-(\epsilon)$ and $\widehat{MCR}_+(\epsilon)$ (see Eq 2.4), which respectively serve as estimates of $MCR_-(\epsilon)$ and $MCR_+(\epsilon)$ (see Section 4.1). Each step of this search consists of minimizing a linear combination of $\hat{e}_{\text{orig}}(f)$ and $\hat{e}_{\text{switch}}(f)$ across $f \in \mathcal{F}$. Our approach is related to the fractional programming approach of Dinkelbach (1967), but accounts for the fact that the problem is constrained by the value of the denominator, $\hat{e}_{\text{orig}}(f)$. We additionally show that, for many model classes,

Model class and loss function ($\mathcal{F}$ & L)	**Computing $\widehat{MCR}_-$**	**Computing $\widehat{MCR}_+$**
(L2 Regularized) Linear models, with the squared error loss	Highly tractable (QP1QC, see Sections 7.2 & 7.3)	Highly tractable (QP1QC, see Sections 7.2 & 7.3)
Linear models in a reproducing kernel Hilbert space, with the squared error loss	Moderately tractable (QP1QC, see Section 7.4.1)	Moderately tractable (QP1QC, see Section 7.4.1)
Cases where irrelevant covariates do not improve predictions	Moderately tractable (Convex optimization problems, see Proposition 11)	Potentially intractable
Cases where minimizing the empirical loss is a convex optimization problem	Potentially intractable (DC programs, see Section 6.3)	Potentially intractable (DC programs, see Section 6.3)

Table 1: Tractability of empirical MCR computation for different model classes – For each case, we describe the tractability of computing $\widehat{MCR}_-$ and $\widehat{MCR}_+$ using our proposed approaches. Computing empirical MCR can be reduced to a sequence of optimization problems, the form of which are noted in parentheses within the above table.

computing $\widehat{MCR}_-(\epsilon)$ only requires that we minimize convex combinations of $\hat{e}_{\text{orig}}(f)$ and $\hat{e}_{\text{switch}}(f)$, which is no more difficult than minimizing the average loss over an expanded and reweighted sample (See Eq 6.2 & Proposition 11).

Computing $\widehat{MCR}_+(\epsilon)$ however will require that we are able to minimize arbitrary linear combinations of $\hat{e}_{\text{orig}}(f)$ and $\hat{e}_{\text{switch}}(f)$. In Section 6.3, we outline how this can be done for convex model classes – classes for which the loss function is convex in the model parameter. Later, in Section 7, we give more specific computational procedures for when $\mathcal{F}$ is the class of linear models, regularized linear models, or linear models in a reproducing kernel Hilbert space (RKHS). We summarize the tractability of computing empirical MCR for different model classes in Table 1.

To simplify notation associated with the reference model f_{ref}, we present our computational results in terms of bounds on empirical MR subject to performance thresholds on the *absolute* scale. More specifically, we present bound functions b_- and b_+ satisfying $b_-(\epsilon_{\text{abs}}) \leq \widehat{MR}(f) \leq b_+(\epsilon_{\text{abs}})$ simultaneously for all $\{f, \epsilon_{\text{abs}} : \hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}, f \in \mathcal{F}, \epsilon_{\text{abs}} > 0\}$ (Figures 2 & 8 show examples of these bounds). The binary search procedures we propose can be used to tighten these boundaries at a particular value ϵ_{abs} of interest.

We briefly note that as an alternative to the global optimization procedures we discuss below, heuristic optimization procedures such as simulated annealing can also prove useful in bounding empirical MCR. By definition, the empirical MR for any model in $\hat{\mathcal{R}}(\epsilon)$ forms a lower bound for $\widehat{MCR}_+(\epsilon)$, and an upper bound for $\widehat{MCR}_-(\epsilon)$. Heuristic maximization and minimization of empirical MR can be used to tighten these boundaries.

Throughout this section, we assume that $0 < \min_{f \in \mathcal{F}} \hat{e}_{\text{orig}}(f)$, to ensure that MR is finite.

6.1. Binary Search for Empirical MR Lower Bound

Before describing our binary search procedure, we introduce additional notation used in this section. Given a constant $\gamma \in \mathbb{R}$ and prediction model $f \in \mathcal{F}$, we define the linear combination $\hat{h}_{-,\gamma}$, and its minimizers (for example, $\hat{g}_{-,\gamma,\mathcal{F}}$), as

$$\hat{h}_{-,\gamma}(f) := \gamma \hat{e}_{\text{orig}}(f) + \hat{e}_{\text{switch}}(f), \quad \text{and} \quad \hat{g}_{-,\gamma,\mathcal{F}} \in \arg\min_{f \in \mathcal{F}} \hat{h}_{-,\gamma}(f).$$

We do not require that $\hat{h}_{-,\gamma}$ is uniquely minimized, and we frequently use the abbreviated notation $\hat{g}_{-,\gamma}$ when $\mathcal{F}$ is clear from context.

Our goal in this section is to derive a lower bound on $\widehat{MR}$ for subsets of $\mathcal{F}$ in the form of $\{f \in \mathcal{F} : \hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}\}$. We achieve this by minimizing a series of linear objective functions in the form of $\hat{h}_{-,\gamma}$, using a similar method to that of Dinkelbach (1967). Often, minimizing the linear combination $\hat{h}_{-,\gamma}(f)$ is more tractable than minimizing the MR ratio directly.

Almost all of the results shown in this section, and those in Section 6.2, also hold if we replace $\hat{e}_{\text{switch}}$ with $\hat{e}_{\text{divide}}$ throughout (see Eq 3.5), including in the definition of $\widehat{MR}$ and $\hat{h}_{-,\gamma}(f)$. The exception is Proposition 11, below, which we may still expect to approximately hold if we replace $\hat{e}_{\text{switch}}$ with $\hat{e}_{\text{divide}}$.

Given an observed sample, we define the following condition for a pair of values $\{\gamma, \epsilon_{\text{abs}}\} \in \mathbb{R} \times \mathbb{R}_{>0}$, and argmin function $\hat{g}_{-,\gamma}$:

Condition 8 (Criteria to continue search for $\widehat{MR}$ lower bound) $\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma}) \geq 0$ *and* $\hat{e}_{orig}(\hat{g}_{-,\gamma}) \leq \epsilon_{abs}$.

We are now equipped to determine conditions under which we can tractably create a lower bound for empirical MR.

Lemma 9 *(Lower bound for $\widehat{MR}$) If $\gamma \in \mathbb{R}$ satisfies $\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma}) \geq 0$, then*

$$\frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{abs}} - \gamma \leq \widehat{MR}(f) \tag{6.1}$$

for all $f \in \mathcal{F}$ satisfying $\hat{e}_{orig}(f) \leq \epsilon_{abs}$. It also follows that

$$-\gamma \leq \widehat{MR}(f) \qquad \text{for all } f \in \mathcal{F}.$$

Additionally, if $f = \hat{g}_{-,\gamma}$ and at least one of the inequalities in Condition 8 holds with equality, then Eq 6.1 holds with equality.

Lemma 9 reduces the challenge of lower-bounding $\widehat{MR}(f)$ to the task of minimizing the linear combination $\hat{h}_{-,\gamma}(f)$. The result of Lemma 9 is not only a single boundary for a particular value of ϵ_{abs}, but a boundary *function* that holds all values of $\epsilon_{\text{abs}} > 0$, with lower values of ϵ_{abs} leading to more restrictive lower bounds on $\widehat{MR}(f)$.

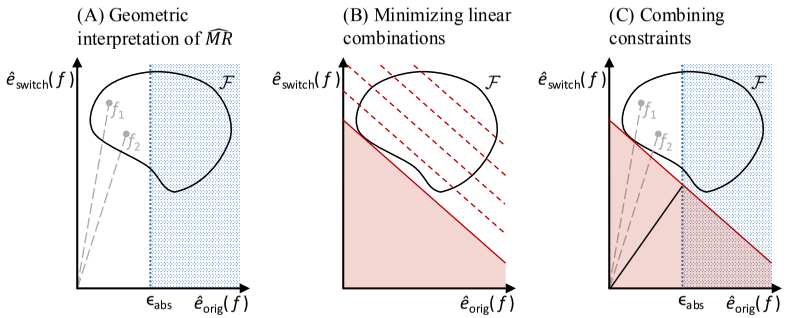


Figure 4: Above, we illustrate the geometric intuition for Lemma 9. In Panel (A), we show an example of a hypothetical model class $\mathcal{F}$, marked by the enclosed region. For each model $f \in \mathcal{F}$, the x-axis shows $\hat{e}_{\text{orig}}(f)$ and the y-axis shows $\hat{e}_{\text{switch}}(f)$. Here, we can see that the condition $\min_{f\in\mathcal{F}} \hat{e}_{\text{orig}}(f) > 0$ holds. The blue dotted region marks models with higher empirical loss. We mark two example models within $\mathcal{F}$ as f_1 and f_2. The slopes of the lines connecting the origin to f_1 and f_2 are equal to $\widehat{MR}(f_1)$ and $\widehat{MR}(f_2)$ respectively. Our goal is to lower-bound the slope corresponding to $\widehat{MR}$ for any model f satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$. In Panel (B), we consider the linear combination $\hat{h}_{-,\gamma}(f) = \gamma\hat{e}_{\text{orig}}(f) + \hat{e}_{\text{switch}}(f)$ for $\gamma = 1$. Above, contour lines of $\hat{h}_{-,\gamma}$ are shown in red. The solid red line indicates the smallest possible value of $\hat{h}_{-,\gamma}$ across $f \in \mathcal{F}$. Specifically, its y-intercept equals $\min_{f\in\mathcal{F}} \hat{h}_{-,\gamma}(f)$. If we can determine this minimum, we can determine a linear border constraint on $\mathcal{F}$, that is, we will know that no points corresponding to models $f \in \mathcal{F}$ may lie in the shaded region above. Additionally, if $\min_{f\in\mathcal{F}} \hat{h}_{-,\gamma}(f) \geq 0$ (see Lemma 9), then we know that the origin is either excluded by this linear constraint, or is on the boundary. In Panel (C), we combine the two constraints from Panels (A) & (B) to see that models $f \in \mathcal{F}$ satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$ must correspond to points in the white, unshaded region above. Thus, as long as the unshaded region does not contain the origin, any line connecting the origin to the a model f satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$ (for example, here, f_1,f_2) must have a slope at least as high as that of the solid black line above. It can be shown algebraically that the black line has slope equal to the left-hand side of Eq 6.1. Thus the left-hand side of Eq 6.1 is a lower bound for $\widehat{MR}(f)$ for all $\{f \in \mathcal{F} : \hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}\}$.

In addition to the formal proof for Lemma 9, we provide a heuristic illustration of the result in Figure 4, to aid intuition.

It remains to determine which value of γ should be used in Eq 6.1. The following lemma implies that this value can be determined by a binary search, given a particular value of interest for ϵ_{abs}.

Lemma 10 *(Monotonicity for $\widehat{MR}$ lower bound binary search) The following monotonicity results hold:*

1. *$\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})$ is monotonically increasing in γ.*
2. *$\hat{e}_{orig}(\hat{g}_{-,\gamma})$ is monotonically decreasing in γ.*
3. *Given ϵ_{abs}, the lower bound from Eq 6.1, $\left\{\frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{abs}} - \gamma\right\}$, is monotonically decreasing in γ in the range where $\hat{e}_{orig}(\hat{g}_{-,\gamma}) \leq \epsilon_{abs}$, and increasing otherwise.*

Given a particular performance level of interest, ϵ_{abs}, Point 3 of Lemma 10 tells us that the value of γ resulting in the tightest lower bound from Eq 6.1 occurs when γ is as low as possible while still satisfying Condition 8. Points 1 and 2 show that if γ_0 satisfies Condition 8, and one of the equalities in Condition 8 holds with equality, then Condition 8 holds for all $\gamma \geq \gamma_0$. Together, these results imply that we can use a binary search to determine the value of γ to be used in Lemma 9, reducing this value until Condition 8 is no longer met. In addition to the formal proof for Lemma 10, we provide an illustration of the result in Figure 5 to aid intuition.

Next we present simple conditions under which the binary search for values of γ can be restricted to the nonnegative real line. This result substantially extends the computational tractability of our approach, as minimizing $\hat{h}_{-,\gamma}$ for $\gamma \geq 0$ is equivalent to minimizing a reweighted empirical loss over an expanded sample of size n^2:

$$\hat{h}_{-,\gamma}(f) = \gamma\hat{e}_{\text{orig}}(f) + \hat{e}_{\text{switch}}(f) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_\gamma(i,j) L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[j,\cdot]}, \mathbf{X}_{2[i,\cdot]})\}, \tag{6.2}$$

where $w_\gamma(i,j) = \frac{\gamma 1(i=j)}{n} + \frac{1(i\neq j)}{n(n-1)} \geq 0$.

Proposition 11 *(Nonnegative weights for $\widehat{MR}$ lower bound binary search) Assume that L and $\mathcal{F}$ satisfy the following conditions.*

1. *(Predictions are sufficient for computing the loss) The loss $L\{f, (Y, X_1, X_2)\}$ depends on the covariates (X_1, X_2) only via the prediction function f, that is, $L\{f, (y, x_1^{(a)}, x_2^{(a)})\} = L\{f, (y, x_1^{(b)}, x_2^{(b)})\}$ whenever $f(x_1^{(a)}, x_2^{(a)}) = f(x_1^{(b)}, x_2^{(b)})$.*
2. *(Irrelevant information does not improve predictions) For any distribution D satisfying $X_1 \perp_D (X_2, Y)$, there exists a function f_D satisfying*

$$\mathbb{E}_D L\{f_D, (Y, X_1, X_2)\} = \min_{f\in\mathcal{F}} \mathbb{E}_D L\{f, (Y, X_1, X_2)\},$$

and

$$f_D(x_1^{(a)}, x_2) = f_D(x_1^{(b)}, x_2) \text{ for any } x_1^{(a)}, x_1^{(b)} \in \mathcal{X}_1 \text{ and } x_2 \in \mathcal{X}_2. \tag{6.3}$$

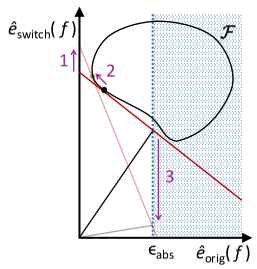


Figure 5: Monotonicity for binary search. Above we show a version of Figure 4-C for two alternative values of γ. This figure is meant to add intuition for the monotonicity results in Lemma 10, in addition to the formal proof. Increasing γ is equivalent to *decreasing* the slope of the red line in Figure 4-C. We define two values $\gamma_1 < \gamma_2$, where γ_1 corresponds to the solid red line, above, and γ_2 corresponds to the semi-transparent red line. The y-intercept values of these lines are equal to $\hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1})$ and $\hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2})$ respectively (see Figure 4-C caption). The solid and semi-transparent black dots mark $\hat{g}_{-,\gamma_1}$ and $\hat{g}_{-,\gamma_2}$ respectively. Plugging γ_1 and γ_2 into Eq 6.1 yields two lower bounds for $\widehat{MR}$, marked by the slopes of the solid and semi-transparent black lines respectively (see Figure 4-C caption). We see that (1) $\hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) \leq \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2})$, that (2) $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}) \geq \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2})$, and that (3) the left-hand side of Eq 6.1 is decreasing in γ when $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma}) \leq \epsilon_{\text{abs}}$. These three conclusions are marked by arrows in the above figure, with numbering matching the enumerated list in Lemma 10.

Let $\gamma = 0$. Under the above assumptions, it follows that either (i) there exists a function $\hat{g}_{-,0}$ minimizing $\hat{h}_{-,0}$ that does not satisfy Condition 8, or (ii) $\hat{e}_{orig}(\hat{g}_{-,0}) \leq \epsilon_{abs}$ and $\widehat{MR}(g_{-,0}) \leq 1$ for any function $\hat{g}_{-,0}$ minimizing $\hat{h}_{-,0}$.

The implication of Proposition 11 is that, when the conditions of Proposition 11 are met, the search region for γ can be limited to the nonnegative real line, and minimizing $\hat{h}_{-,\gamma}$ will be no harder than minimizing a reweighted empirical loss over an expanded sample (Eq 6.2). To see this, recall that for a fixed value of ϵ_{abs} we can tighten the boundary in Lemma 9 by conducting a binary search for the smallest value of γ that satisfies Condition 8. If setting γ equal to 0 does not satisfy Condition 8, and the search for γ can be restricted to the nonnegative real line, where minimizing $\hat{h}_{-,0}$ is more tractable (see Eq 6.2). Alternatively, if $\hat{e}_{\text{orig}}(g_{-,0}) \leq \epsilon_{\text{abs}}$ and $\widehat{MR}(g_{-,0}) \leq 1$, then we have identified a well-performing model $g_{-,0}$ with empirical MR no greater than 1. For $\epsilon_{\text{abs}} = \hat{e}_{\text{orig}}(f_{\text{ref}}) + \epsilon$, this implies that $\widehat{MCR}_{-}(\epsilon) \leq 1$, which is a sufficiently precise conclusion for most interpretational purposes (see Appendix A.2).

Because of the fixed pairing structure used in $\hat{e}_{\text{divide}}$, Proposition 11 will not necessarily hold if we replace $\hat{e}_{\text{switch}}$ with $\hat{e}_{\text{divide}}$ throughout (see Appendix C.3). However, since $\hat{e}_{\text{divide}}$ approximates $\hat{e}_{\text{switch}}$, we can expect Proposition 11 to hold approximately. The bound from Eq 6.1 still remains valid if we replace $\hat{e}_{\text{switch}}$ with $\hat{e}_{\text{divide}}$ and limit γ to the nonnegative reals, although in some cases it may not be as tight.

6.2. Binary Search for Empirical MR Upper Bound

We now briefly present a binary search procedure to upper bound $\widehat{MR}$, which mirrors the procedure from Section 6.1. Given a constant $\gamma \in \mathbb{R}$ and prediction model $f \in \mathcal{F}$, we define the linear combination $\hat{h}_{+,\gamma}$, and its minimizers (for example, $\hat{g}_{+,\gamma,\mathcal{F}}$), as

$$\hat{h}_{+,\gamma}(f) := \hat{e}_{\text{orig}}(f) + \gamma \hat{e}_{\text{switch}}(f), \quad \text{and} \quad \hat{g}_{+,\gamma,\mathcal{F}} \in \underset{f \in \mathcal{F}}{\arg\min}\, \hat{h}_{+,\gamma}(f).$$

As in Section 6.1, $\hat{h}_{+,\gamma}$ need not be uniquely minimized, and we generally abbreviate $\hat{g}_{+,\gamma,\mathcal{F}}$ as $\hat{g}_{+,\gamma}$ when $\mathcal{F}$ is clear from context.

Given an observed sample, we define the following condition for a pair of values $\{\gamma, \epsilon_{\text{abs}}\} \in \mathbb{R}_{\leq 0} \times \mathbb{R}_{>0}$, and argmin function $\hat{g}_{+,\gamma}$:

Condition 12 (Criteria to continue search for $\widehat{MR}$ upper bound) $\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma}) \geq 0$ *and* $\hat{e}_{orig}(\hat{g}_{+,\gamma}) \leq \epsilon_{abs}$.

We can now develop a procedure to upper bound $\widehat{MR}$, as shown in the next lemma.

Lemma 13 *(Upper bound for $\widehat{MR}$) If $\gamma \in \mathbb{R}$ satisfies $\gamma \leq 0$ and $\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma}) \geq 0$, then*

$$\widehat{MR}(f) \leq \left\{ \frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{abs}} - 1 \right\} \gamma^{-1} \tag{6.4}$$

for all $f \in \mathcal{F}$ satisfying $\hat{e}_{orig}(f) \leq \epsilon_{abs}$. It also follows that

$$\widehat{MR}(f) \leq |\gamma^{-1}| \qquad \text{for all } f \in \mathcal{F}. \tag{6.5}$$

Additionally, if $f = \hat{g}_{+,\gamma}$ and at least one of the inequalities in Condition 12 holds with equality, then Eq 6.4 holds with equality.

As in Section 6.1, it remains to determine the value of γ to use in Lemma 13, given a value of interest for $\epsilon_{\text{abs}} \geq \min_{f \in \mathcal{F}} \hat{e}_{\text{orig}}(f)$. The next lemma tells us that the boundary from Lemma 13 is tightest when γ is as low as possible while still satisfying Condition 12.

Lemma 14 *(Monotonicity for $\widehat{MR}$ upper bound binary search) The following monotonicity results hold:*

1. *$\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})$ is monotonically increasing in γ.*
2. *$\hat{e}_{orig}(\hat{g}_{+,\gamma})$ is monotonically decreasing in γ for $\gamma \leq 0$, and Condition 12 holds for $\gamma = 0$ and $\epsilon_{abs} \geq \min_{f \in \mathcal{F}} \hat{e}_{orig}(f)$.*

3. *Given ϵ_{abs}, the upper boundary $\left\{\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{abs}} - 1\right\}\gamma^{-1}$ is monotonically increasing in γ in the range where $\hat{e}_{orig}(\hat{g}_{+,\gamma}) \leq \epsilon_{abs}$ and $\gamma < 0$, and decreasing in the range where $\hat{e}_{orig}(\hat{g}_{+,\gamma}) > \epsilon_{abs}$ and $\gamma < 0$.*

Together, the results from Lemma 14 imply that we can use a binary search across $\gamma \in \mathbb{R}$ to tighten the boundary on $\widehat{MR}$ from Lemma 13.

6.3. Convex Models

In this section we show that empirical MCR can be conservatively computed when the loss function is convex in the model parameters – that is, when the models $f_\theta \in \mathcal{F}$ are indexed by a d-dimensional parameter $\theta \in \Theta \subseteq \mathbb{R}^d$, and when the loss function $L(f_\theta, (y, x_1, x_2))$ is convex in θ for all $(x_1, x_2, y) \in \mathcal{X}_1 \times (\mathcal{X}_2, \mathcal{Y})$.

Fortunately, neither Lemma 9 nor Lemma 13 require an exact minimum for $\hat{h}_{-,\gamma}$ or $\hat{h}_{+,\gamma}$. For Lemma 9, any lower bound on $\hat{h}_{-,\gamma}$ is sufficient to determine a lower bound on $MR(f)$. Likewise, for Lemma 13, any lower bound on $\hat{h}_{+,\gamma}$ is sufficient to determine an upper bound on $MR(f)$.

To find these lower bounds, we note that for "convex" model classes (defined above) the optimization problems in Sections 6.1 & 6.2 can be written either as convex optimization problems, or as difference convex function (DC) programs. A DC program is one that can be written as

$$\min_{\{\theta : c_{\text{DC}}(\theta) \leq k, \theta \in \Theta\}} g_{\text{DC}}(\theta) - h_{\text{DC}}(\theta),$$

where c_{DC} is a constraint function, $k \in \mathbb{R}^1$, and g_{DC}, h_{DC}, and c_{DC} are convex. Although precise solutions to DC problems are not always tractable, lower bounds can be attained by branch-and-bound (B&B) methods (Horst and Thoai, 1999). A simple B&B approach is to partition Θ into a set of simplexes. Within the jth simplex, a lower bound on $g_{\text{DC}}(\theta) - h_{\text{DC}}(\theta)$ can be determined by replacing h_{DC} with the hyperplane function h_j satisfying $h_j(v) = h_{\text{DC}}(v)$ at each vertex v of the jth simplex. Within this partition, $g_{\text{DC}}(\theta) - h_{\text{DC}}(\theta)$ is lower bounded by $l_j := \min_\theta g_{\text{DC}}(\theta) - h_j(\theta)$, which can be computed as the solution to a convex optimization problem. Any partition for which l_j is found to be too high is disregarded. Once a bound l_j is computed for each partition, the partition with the lowest value l_j is selected to be subdivided further, and additional lower bounds are recomputed for each new, resulting partition. This procedure continues until a sufficiently tight lower bound is attained (for more detailed procedures, see Horst and Thoai, 1999).

This approach allows us to conservatively approximate bounds on $\widehat{MR}(f)$ in the form of Eq 6.1 & 6.4 by replacing $\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})$ and $\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})$ with lower bounds from the B&B procedure. Although it will always yield valid bounds, the procedure may converge slowly when the dimension of Θ is large, giving highly conservative results. For some special cases of model classes however, even high dimensional DC problems simplify greatly. We discuss these cases in the next section.

7. MR & MCR for Linear Models, Additive Models, and Regression Models in a Reproducing Kernel Hilbert Space

For linear or additive models, many simplifications can be made to our approaches for MR and MCR. To simplify the interpretation of MR, we show below that population-level MR for a linear model can be expressed in terms of the model's coefficients (Section 7.1). To simplify computation, we show that the cost of computing empirical MR for a linear model grows only linearly in n (Section 7.1), even though the number of terms in the definition of empirical MR grows quadratically (see Eqs 3.3 & 3.6).

Moving on from MR, we show how empirical MCR can be computed for the class of linear models (Section 7.2), for regularized linear models (Section 7.3), and for regression models in a reproducing kernel Hilbert space (RKHS, Section 7.4). To do this, we build on the approach in Section 6 by giving approaches for minimizing arbitrary combinations of $\hat{e}_{\text{switch}}(f)$ and $\hat{e}_{\text{orig}}(f)$ across $f \in \mathcal{F}$. Even when the associated objective functions are non-convex, we can tractably obtain global minima for these model classes. We also discuss procedures to determine an upper bound B_{ind} on the loss for any observation when using these model classes (see Assumption 1).

Throughout this section, we assume that $\mathcal{X} \subset \mathbb{R}^p$ for $p \in \mathbb{Z}^+$, that $\mathcal{Y} \subset \mathbb{R}^1$, and that L is the squared error loss function $L(f, (y, x_1, x_2) = (y - f(x_1, x_2))^2$. As in Section 6, we also assume that $0 < \min_{f \in \mathcal{F}} \hat{e}_{\text{orig}}(f)$, to ensure that empirical MR is finite.

7.1. Interpreting and Computing MR for Linear or Additive Models

We begin by considering MR for linear models evaluated with the squared error loss. For this setting, we can show both an interpretable definition of MR, as well as a computationally efficient formula for $\hat{e}_{\text{switch}}(f)$.

Proposition 15 *(Interpreting MR, and computing empirical MR for linear models) For any prediction model f, let $e_{orig}(f)$, $e_{switch}(f)$, $\hat{e}_{\text{orig}}(f)$, and $\hat{e}_{switch}(f)$ be defined based on the squared error loss $L(f, (y, x_1, x_2)) := (y - f(x_1, x_2))^2$ for $y \in \mathbb{R}$, $x_1 \in \mathbb{R}^{p_1}$, and $x_2 \in \mathbb{R}^{p_2}$, where p_1 and p_2 are positive integers. Let $\beta = (\beta_1, \beta_2)$ and f_β satisfy $\beta_1 \in \mathbb{R}^{p_1}$, $\beta_2 \in \mathbb{R}^{p_2}$, and $f_\beta(x) = x'\beta = x_1'\beta_1 + x_2'\beta_2$. Then*

$$MR(f_\beta) = 1 + \frac{2}{e_{orig}(f_\beta)} \left\{ Cov(Y, X_1)\beta_1 - \beta_2' Cov(X_2, X_1)\beta_1 \right\}, \tag{7.1}$$

and, for finite samples,

$$\hat{e}_{switch}(f_\beta) = \frac{1}{n} \left\{ \mathbf{y}'\mathbf{y} - 2 \begin{bmatrix} \mathbf{X}_1'\mathbf{W}\mathbf{y} \\ \mathbf{X}_2'\mathbf{y} \end{bmatrix}' \beta + \beta' \begin{bmatrix} \mathbf{X}_1'\mathbf{X}_1 & \mathbf{X}_1'\mathbf{W}\mathbf{X}_2 \\ \mathbf{X}_2'\mathbf{W}\mathbf{X}_1 & \mathbf{X}_2'\mathbf{X}_2 \end{bmatrix} \beta \right\}, \tag{7.2}$$

where $\mathbf{W} := \frac{1}{n-1}(\mathbf{1}_n\mathbf{1}_n' - \mathbf{I}_n)$, $\mathbf{1}_n$ is the n-length vector of ones, and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

Eq 7.1 shows that model reliance for linear models can be interpreted in terms of the population covariances, the model coefficients, and the model's accuracy. Gregorutti et al. (2017) show an equivalent formulation of Eq 7.1 under the stronger assumptions that f_β is

equal to the conditional expectation function of Y (that is, $f_\beta(x) = \mathbb{E}(Y|X = x)$), and the covariates X_1 and X_2 are centered.

Eq 7.2 shows that, although the number of terms in the definition of $\hat{e}_{\text{switch}}$ grows quadratically in n (see Eq 3.3), the computational complexity of $\hat{e}_{\text{switch}}(f_\beta)$ for a linear model f_β grows *only linearly* in n. Specifically, the terms $\mathbf{X}_1'\mathbf{W}\mathbf{y}$ and $\mathbf{X}_1'\mathbf{W}\mathbf{X}_2$ in Eq 7.2 can be computed as $\frac{1}{n-1}\{(\mathbf{X}_1'\mathbf{1}_n)(\mathbf{1}_n'\mathbf{y}) - (\mathbf{X}_1'\mathbf{y})\}$ and $\frac{1}{n-1}\{(\mathbf{X}_1'\mathbf{1}_n)(\mathbf{1}_n'\mathbf{X}_2) - (\mathbf{X}_1'\mathbf{X}_2)\}$ respectively, where the computational complexity of each term in parentheses grows linearly in n.

As in Gregorutti et al. (2017), both results in Proposition 15 readily generalize to additive models of the form $f_{g_1,g_2}(X_1, X_2) := g_1(X_1) + g_2(X_2)$, since permuting X_1 is equivalent to permuting $g_1(X_1)$.

7.2. Computing Empirical MCR for Linear Models

Building on the computational result from the previous section, we now consider empirical MCR computation for linear model classes of the form

$$\mathcal{F}_{\text{lm}} := \left\{ f_\beta : f_\beta(x) = x'\beta, \quad \beta \in \mathbb{R}^p \right\}.$$

In order to implement the computational procedure from Sections 6.1 and 6.2, we must be able to minimize arbitrary linear combinations of $\hat{e}_{\text{orig}}(f_\beta)$ and $\hat{e}_{\text{switch}}(f_\beta)$. Fortunately, for linear models, this minimization reduces to a quadratic program, as we show in the next remark.

Remark 16 *(Tractability of empirical MCR for linear model classes) For any $f_\beta \in \mathcal{F}_{lm}$ and any fixed coefficients $\xi_{orig}, \xi_{switch} \in \mathbb{R}$, the linear combination*

$$\xi_{orig}\hat{e}_{orig}(f_\beta) + \xi_{switch}\hat{e}_{switch}(f_\beta) \tag{7.3}$$

is proportional in β to the quadratic function $-2\mathbf{q}'\beta + \beta'\mathbf{Q}\beta$, where

$$\mathbf{Q} := \xi_{orig}\mathbf{X}'\mathbf{X} + \xi_{switch}\begin{bmatrix} \mathbf{X}_1'\mathbf{X}_1 & \mathbf{X}_1'\mathbf{W}\mathbf{X}_2 \\ \mathbf{X}_2'\mathbf{W}\mathbf{X}_1 & \mathbf{X}_2'\mathbf{X}_2 \end{bmatrix}, \qquad \mathbf{q} := \left(\xi_{orig}\mathbf{y}'\mathbf{X} + \xi_{switch}\begin{bmatrix} \mathbf{X}_1'\mathbf{W}\mathbf{y} \\ \mathbf{X}_2'\mathbf{y} \end{bmatrix}' \right)',$$

and $\mathbf{W} := \frac{1}{n-1}(\mathbf{1}_n\mathbf{1}_n' - \mathbf{I}_n)$. Thus, minimizing $\xi_{orig}\hat{e}_{orig}(f_\beta) + \xi_{switch}\hat{e}_{switch}(f_\beta)$ is equivalent to an unconstrained (possibly non-convex) quadratic program.

Because our empirical MCR computation procedure from Sections 6.1 and 6.2 consists of minimizing a sequence of objective functions in the form of Eq 7.3, Remark 16 shows us that this procedure is tractable for the class of unconstrained linear models.

7.3. Regularized Linear Models

Next, we continue to build on the results from Section 7.2 to calculate boundaries on $\widehat{MR}$ for *regularized* linear models. We consider model classes formed by quadratically constrained subsets of $\mathcal{F}_{\text{lm}}$, defined as

$$\mathcal{F}_{\text{lm},r_{\text{lm}}} := \left\{ f_\beta : f_\beta(x) = x'\beta, \quad \beta \in \mathbb{R}^p, \quad \beta'\mathbf{M}_{\text{lm}}\beta \le r_{\text{lm}} \right\}, \tag{7.4}$$

where $\mathbf{M}_{\text{lm}}$ and r_{lm} are pre-specified. Again, this class describes linear models with a quadratic constraint on the coefficient vector.

7.3.1. Calculating MCR

As in Section 7.2, calculating bounds on $\widehat{MR}$ via Lemmas 9 & 13 requires that are able to minimizing linear combinations $\xi_{\text{orig}}\hat{e}_{\text{orig}}(f_\beta) + \xi_{\text{switch}}\hat{e}_{\text{switch}}(f_\beta)$ across $f_\beta \in \mathcal{F}_{\text{lm},r_{\text{lm}}}$ for arbitrary $\xi_{\text{orig}}, \xi_{\text{switch}} \in \mathbb{R}$. Applying Remark 16, we can again equivalently minimize $-2\mathbf{q}'\beta + \beta'\mathbf{Q}\beta$ subject to the constraint in Eq 7.4:

$$\begin{aligned} \text{minimize} \quad & -2\mathbf{q}'\beta + \beta'\mathbf{Q}\beta \\ \text{subject to} \quad & \beta'\mathbf{M}_{\text{lm}}\beta \le r_{\text{lm}}. \end{aligned} \tag{7.5}$$

The resulting optimization problem is a (possibly non-convex) quadratic program with one quadratic constraint (QP1QC). This problem is well-studied, and is related to the trust region problem (Boyd and Vandenberghe, 2004; Pólik and Terlaky, 2007; Park and Boyd, 2017). Thus, the bounds on MCR presented in Sections 6.1 and 6.2 again become computationally tractable for the class of quadratically constrained linear models.

7.3.2. Upper Bounding the Loss

One benefit of constraining the coefficient vector ($\beta'\mathbf{M}_{\text{lm}}\beta \le r_{\text{lm}}$) is that it facilitates determining an upper bound B_{ind} on the loss function $L(f_\beta, (y, x)) = (y - x'\beta)^2$, which automatically satisfies Assumption 1 for all $f \in \mathcal{F}_{\text{lm},r_{\text{lm}}}$. The following lemma gives sufficient conditions to determine B_{ind}.

Lemma 17 *(Loss upper bound for linear models) If $\mathbf{M}_{lm}$ is positive definite, Y is bounded within a known range, and there exists a known constant $r_{\mathcal{X}}$ such that $x'\mathbf{M}_{lm}^{-1}x \le r_{\mathcal{X}}$ for all $x \in (\mathcal{X}_1 \times \mathcal{X}_2)$, then Assumption 1 holds for the model class $\mathcal{F}_{lm,r_{lm}}$, the squared error loss function, and the constant*

$$B_{ind} = \max\left[\left\{ \min_{y\in\mathcal{Y}}(y) - \sqrt{r_{\mathcal{X}} r_{lm}} \right\}^2, \left\{ \max_{y\in\mathcal{Y}}(y) + \sqrt{r_{\mathcal{X}} r_{lm}} \right\}^2 \right].$$

In practice, the constant $r_{\mathcal{X}}$ can be approximated by the empirical distribution of X and Y. The motivation behind the restriction $x'\mathbf{M}_{\text{lm}}^{-1}x \le r_{\mathcal{X}}$ in Lemma 17 is to create complementary constraints on X and β. For example, if $\mathbf{M}_{\text{lm}}$ is diagonal, then the smallest elements of $\mathbf{M}_{\text{lm}}$ correspond to directions along which β is least restricted by $\beta'\mathbf{M}_{\text{lm}}\beta \le r_{\text{lm}}$ (Eq 7.5), as well as the directions along which x is most restricted by $x'\mathbf{M}_{\text{lm}}^{-1}x \le r_{\mathcal{X}}$ (Lemma 17).

7.4. Regression Models in a Reproducing Kernel Hilbert Space (RKHS)

We now expand our scope of model classes by considering regression models in a reproducing kernel Hilbert space (RKHS), which allow for nonlinear and nonadditive features of the covariates. We show that, as in Section 7.3, minimizing a linear combination of $\hat{e}_{\text{orig}}(f)$ and $\hat{e}_{\text{switch}}(f)$ across models f in this class can be expressed as a QP1QC, which allows us to implement the binary search procedure of Sections 6.1 & 6.2.

First we introduce notation required to describe regression in a RKHS. Let $\mathbf{D}$ be a $(R \times p)$ matrix representing a pre-specified dictionary of R reference points, such that each row of $\mathbf{D}$ is contained in $\mathcal{X} = \mathbb{R}^p$. Let k be a pre-specified positive definite kernel function, and let μ

be a prespecified estimate of $\mathbb{E}Y$. Let $\mathbf{K_D}$ be the $R \times R$ matrix with $\mathbf{K_D}_{[i,j]} = k(\mathbf{D}_{[i,\cdot]}, \mathbf{D}_{[j,\cdot]})$. We consider prediction models of the following form, where the distance to each reference point is used as a regression feature:

$$\mathcal{F}_{\mathbf{D},r_k} = \left\{ f_\alpha : f_\alpha(x) = \mu + \sum_{i=1}^{R} k\left(x, \mathbf{D}_{[i,\cdot]}\right) \alpha_{[i]}, \quad \|f_\alpha\|_k \leq r_k, \quad \alpha \in \mathbb{R}^R \right\}. \tag{7.6}$$

Above, the norm $\|f_\alpha\|_k$ is defined as

$$\|f_\alpha\|_k := \sum_{i=1}^{R} \sum_{j=1}^{R} \alpha_{[i]} \alpha_{[j]} k(\mathbf{D}_{[i,\cdot]}, \mathbf{D}_{[j,\cdot]}) = \alpha' \mathbf{K_D} \alpha. \tag{7.7}$$

In the next two sections, we show that bounds on empirical MCR can again be tractably computed for this class, and that the loss for models in this class can be feasibly upper bounded.

7.4.1. Calculating MCR

Again, calculating bounds on $\widehat{MR}$ from Lemmas 9 & 13 requires us to be able to minimize arbitrary linear combinations of $\hat{e}_{\text{orig}}(f_\alpha)$ and $\hat{e}_{\text{switch}}(f_\alpha)$.

Given a size-n sample of test observations $\mathbf{Z} = \begin{bmatrix} \mathbf{y} & \mathbf{X} \end{bmatrix}$, let $\mathbf{K}_{\text{orig}}$ be the $n \times R$ matrix with elements $\mathbf{K}_{\text{orig}[i,j]} = k\left(\mathbf{X}_{[i,\cdot]}, \mathbf{D}_{[j,\cdot]}\right)$. Let $\mathbf{Z}_{\text{switch}} = \begin{bmatrix} \mathbf{y}_{\text{switch}} & \mathbf{X}_{\text{switch}} \end{bmatrix}$ be the $(n(n-1)) \times (1+p)$ matrix with rows that contain the set $\{(\mathbf{y}_{[i]}, \mathbf{X}_{1[j,\cdot]}, \mathbf{X}_{2[i,\cdot]}) : i, j \in \{1, \ldots, n\} \text{ and } i \neq j\}$. Finally, let $\mathbf{K}_{\text{switch}}$ be the $n(n-1) \times R$ matrix with $\mathbf{K}_{\text{switch}[i,j]} = k\left(\mathbf{X}_{\text{switch}[i,\cdot]}, \mathbf{D}_{[j,\cdot]}\right)$.

For any two constants $\xi_{\text{orig}}, \xi_{\text{switch}} \in \mathbb{R}$, we can show that minimizing the linear combination $\xi_{\text{orig}} \hat{e}_{\text{orig}}(f_\alpha) + \xi_{\text{switch}} \hat{e}_{\text{switch}}(f_\alpha)$ over $\mathcal{F}_{\mathbf{D},r_k}$ is equivalent to the minimization problem

$$\text{minimize} \quad \frac{\xi_{\text{orig}}}{n} \|\mathbf{y} - \mu - \mathbf{K}_{\text{orig}} \alpha\|_2^2 + \frac{\xi_{\text{switch}}}{n(n-1)} \|\mathbf{y}_{\text{switch}} - \mu - \mathbf{K}_{\text{switch}} \alpha\|_2^2 \tag{7.8}$$

$$\text{subject to} \quad \alpha' \mathbf{K_D} \alpha < r_k. \tag{7.9}$$

Like Problem 7.5, Problem 7.8-7.9 is a QP1QC. To show Eqs 7.8-7.9, we first write $\hat{e}_{\text{orig}}(f_\alpha)$ as

$$\begin{aligned} \hat{e}_{\text{orig}}(f_\alpha) &= \frac{1}{n} \sum_{i=1}^{n} \left\{ \mathbf{y}_{[i]} - f_\alpha(\mathbf{X}_{[i,\cdot]}) \right\}^2 && (7.10) \\ &= \frac{1}{n} \sum_{i=1}^{n} \left\{ \mathbf{y}_{[i]} - \mu - \sum_{j=1}^{R} k(\mathbf{X}_{[i,\cdot]}, \mathbf{D}_{[j,\cdot]}) \alpha_{[j]} \right\}^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} \left\{ \mathbf{y}_{[i]} - \mu - \mathbf{K}'_{\text{orig}[i,\cdot]} \alpha \right\}^2 \\ &= \frac{1}{n} \|\mathbf{y} - \mu - \mathbf{K}_{\text{orig}} \alpha\|_2^2. && (7.11) \end{aligned}$$

Following similar steps, we can obtain

$$\hat{e}_{\text{switch}}(f_\alpha) = \frac{1}{n(n-1)} \|\mathbf{y}_{\text{switch}} - \mu - \mathbf{K}_{\text{switch}} \alpha\|_2^2.$$

Thus, for any two constants $\xi_{\text{orig}}, \xi_{\text{switch}} \in \mathbb{R}$, we can see that $\xi_{\text{orig}} \hat{e}_{\text{orig}}(f_\alpha) + \xi_{\text{switch}} \hat{e}_{\text{switch}}(f_\alpha)$ is quadratic in α. This means that we can tractably compute bounds on empirical MCR for this class as well.

7.4.2. Upper Bounding the Loss

Using similar steps as in Section 7.3.2, the following lemma gives sufficient conditions to determine B_{ind} for the case of regression in a RKHS.

Lemma 18 *(Loss upper bound for regression in a RKHS) Assume that Y is bounded within a known range, and there exists a known constant $r_{\mathbf{D}}$ such that $v(x)' \mathbf{K_D}^{-1} v(x) \leq r_{\mathbf{D}}$ for all $x \in (\mathcal{X}_1 \times \mathcal{X}_2)$, where $v : \mathbb{R}^p \to \mathbb{R}^R$ is the function satisfying $v(x)_{[i]} = k(x, \mathbf{D}_{[i,\cdot]})$. Under these conditions, Assumption 1 holds for the model class $\mathcal{F}_{\mathbf{D},r_k}$, the squared error loss function, and the constant*

$$B_{ind} = \max \left[\left\{ \min_{y \in \mathcal{Y}} (y) - (\mu + \sqrt{r_{\mathbf{D}} r_k}) \right\}^2, \left\{ \max_{y \in \mathcal{Y}} (y) + (\mu + \sqrt{r_{\mathbf{D}} r_k}) \right\}^2 \right].$$

Thus, for regression models in a RKHS, we can satisfy Assumption 1 for all models in the class.

8. Connections Between MR and Causality

Our MR approach can be fundamentally described as studying how a model's behavior changes under an intervention on the underlying data. We aim to study the causal effect of this intervention on the *model's* performance. This goal mirror's the conventional causal inference goal of studying how an intervention on variables will change outcomes generated by a process in *nature*.

This section explores this connection to causal inference further. Section 8.1 shows that when the prediction model in question is the conditional expectation function from nature itself, MR reduces to commonly studied quantities in the causal literature. Section 8.2 proposes an alternative to MR that focuses on interventions, or data perturbations, that are likely to occur in the underlying data generating process.

8.1. Model Reliance and Causal Effects

In this section, we show a connection between population-level model reliance and the conditional average causal effect. For consistency with the causal inference literature, we temporarily rename the random variables (Y, X_1, X_2) as (Y, T, C), with realizations (y, t, c). Here, $T := X_1$ represents a binary treatment indicator, $C := X_2$ represents a set of baseline covariates ("C" is for "covariates"), and Y represents an outcome of interest. Under this notation, $e_{\text{orig}}(f)$ represents the expected loss of a prediction function f, and $e_{\text{switch}}(f)$ denotes the expected loss in a pair of observations in which the treatment has been switched.

Let $f_0(t,c) := \mathbb{E}(Y|C=c, T=t)$ be the (unknown) conditional expectation function for Y, where we place no restrictions on the functional form of f_0.

Let Y_1 and Y_0 be potential outcomes under treatment and control respectively, such that $Y = Y_0(1-T) + Y_1 T$. The treatment effect for an individual is defined as $Y_1 - Y_0$, and the average treatment effect is defined as $\mathbb{E}(Y_1 - Y_0)$. Let $\text{CATE}(c) := \mathbb{E}(Y_1 - Y_0|C=c)$ be the (unknown) conditional average treatment effect of T for all patients with $C = c$. Causal inference methods typically assume $(Y_1, Y_0) \perp T|C$ (conditional ignorability), and $0 < \mathbb{P}(T=1|C=c) < 1$ for all values of c (positivity), in order for f_0 and CATE to be well defined and identifiable.

The next proposition quantifies the relation between the conditional average treatment effect function (CATE) and the model reliance of f_0 on X_1.

Proposition 19 *(Causal interpretations of MR) For any prediction model f, let $e_{orig}(f)$ and $e_{switch}(f)$ be defined based on the squared error loss $L(f,(y,t,c)) := (y - f(t,c))^2$.*

If $(Y_1, Y_0) \perp T|C$ (conditional ignorability) and $0 < \mathbb{P}(T=1|C=c) < 1$ for all values of c (positivity), then $MR(f_0)$ is equal to

$$1 + \frac{Var(T)}{\mathbb{E}_{T,C}\, Var(Y|T,C)} \sum_{t\in\{0,1\}} \left\{ \mathbb{E}(Y_1 - Y_0|T=t)^2 + Var(CATE(C)|T=t) \right\}, \tag{8.1}$$

where $Var(T)$ is the marginal variance of the treatment assignment.

We see above that model reliance decomposes into several terms that are each individually important in causal inference: the treatment prevalence (via $\text{Var}(T)$); the variability in Y that is not explained by C or T; the magnitude of the average treatment effect, conditional on T; and the variance of the conditional average treatment effect across subgroups. For example, if all patients are treated, then scrambling the treatment in a random pair of observations has no effect on the loss. In this case we see that $\text{Var}(T) = 0$ and $MR(f_0) = 1$, indicating no reliance. When $\text{Var}(T) > 0$, a higher average treatment effect magnitude $(\mathbb{E}(Y_1 - Y_0|T=t)^2)$ corresponds to f_0 requiring T more heavily to predict Y, all else equal. Similarly, if there is a high degree of treatment effect heterogeneity across subgroups (that is, when $\text{Var}\,(\text{CATE}(C)|T=t)$ is large), the model f_0 will again use T more heavily when predicting Y. For example, a treatment may be important for predicting Y even if the average treatment effect is zero, so long as the treatment helps some subgroups more than others.

8.2. Conditional Importance: Adjusting for Dependence Between X_1 and X_2

One common scenario where multiple models achieve low loss is when the sets of predictors X_1 and X_2 are highly correlated, or contain redundant information. Models may predict well either through reliance on X_1, or through reliance on X_2, and so MCR will correctly identify a wide range of potential reliances on X_1. However, we may specifically be interested how much models rely on the information in X_1 that cannot alternatively be gleaned from X_2.

For example, age and accumulated wealth may be correlated, and both may be predictive of future promotion. We may wish to know the how much a model for predicting promotion relies on information that is uniquely available from wealth measurements.

To formalize this notion, we define an alternative to e_{switch} where noise is added to X_1 in a way that accounts for the dependence between X_1 and X_2. Given a fixed prediction model f, we ask: how well would the model f perform if the values of X_1 were scrambled *across observations with the same value for* X_2. Specifically, let $Z^{(a)} = (Y^{(a)}, X_1^{(a)}, X_2^{(a)})$ and $Z^{(b)} = (Y^{(b)}, X_1^{(b)}, X_2^{(b)})$ denote a pair of independent random vectors following the same distribution as $Z = (Y, X_1, X_2)$, as in Section 3, and let

$$e_{\text{cond}}(f) := \mathbb{E}_{X_2}\mathbb{E}_{(Y^{(b)},X_1^{(a)},X_2^{(b)})}\left[L\{f,(Y^{(b)},X_1^{(a)},X_2^{(b)})\} | X_2^{(a)} = X_2^{(b)} = X_2 \right]. \tag{8.2}$$

In words, $e_{\text{cond}}(f)$ is the expected loss of a given model f across pairs of observations $(Z^{(a)}, Z^{(b)})$ in which the values of $X_1^{(a)}$ and $X_1^{(b)}$ have been switched, given that these pairs match on X_2. This quantity can also be interpreted as the expected loss of f if noise were added to X_1 in such a way that X_1 was no longer informative of Y, given X_2, but that the joint distribution of the covariates (X_1, X_2) was maintained.

We then define conditional model reliance, or "core" model reliance (CMR) for a fixed function f as

$$CMR(f) = \frac{e_{\text{cond}}(f)}{e_{\text{orig}}(f)}.$$

That is, CMR is the factor by which the model's performance degrades when the information unique to X_1 is removed. If $X_1 \perp X_2$, then X_1 contains no redundant information, and CMR and MR are equivalent. Otherwise, all else equal, CMR will decrease as X_2 becomes more predictive of X_1. Analogous to MCR, we define conditional MCR (CMCR) in the same way as in Eq 2.2, but with MR replaced with CMR. In comparison with MCR, CMCR will generally result in a range that is closer to 1 (null reliance).

An advantage of CMR is that it restricts the "noise-corrupted" inputs to be within the domain $\mathcal{X}$, rather than the expanded domain $\mathcal{X}_1 \times \mathcal{X}_2$ considered by MR. This means that CMR will not be influenced by impossible combinations of x_1 and x_2, while MR may be influenced by them. Hooker (2007) discuss a similar issue, arguing that evaluations of a prediction model's behavior in different circumstances should be weighted by, for example, how likely those circumstances are to occur.

A challenge facing the CMR approach is that matched pairs such as those in Eq 8.2 may occur rarely, making it difficult to estimate CMR nonparametrically. We explore this estimation issue next.

8.2.1. Estimation of CMR by Weighting, Matching, or Imputation

If the covariate space is discrete and low dimensional, nonparametric methods based on weighting or matching can be effective means of estimating CMR. Specifically, we can weight each pair of sample points i, j according to how likely the covariate combination $(\mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]})$ is to occur, as in

$$\hat{e}_{\text{weight}}(f) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j\neq i} w(\mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]}) \times L\{f,(\mathbf{y}_{[j]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]})\},$$

where $w(x_1, x_2) := \frac{\mathbb{P}(X_1=x_1|X_2=x_2)}{\mathbb{P}(X_1=x_1)}$ is an importance weight (see also Hooker, 2007). Here, pairs of observations corresponding to unlikely or impossible combinations of covariates are

down-weighted or discarded, respectively. If the probabilities $\mathbb{P}(X_1 = x_1|X_2 = x_2)$ and $\mathbb{P}(X_1 = x_1)$ are known, then $\hat{e}_{\text{weight}}(f)$ is unbiased for $e_{\text{cond}}(f)$ (see Appendix A.7).

Alternatively, if X_2 is discrete and low dimensional, we can restrict estimates of $e_{\text{cond}}(f)$ to only consider pairs of sample observations in which X_2 is constant, or "matched," as in

$$\hat{e}_{\text{match}}(f) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} \frac{1(\mathbf{X}_{2[j,\cdot]} = \mathbf{X}_{2[i,\cdot]})}{\mathbb{P}(X_2 = \mathbf{X}_{2[i,\cdot]})} \times L\{f, (\mathbf{y}_{[j]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]})\}. \tag{8.3}$$

This approach allows estimation of CMR without knowledge of the conditional distribution $\mathbb{P}(X_1 = x_1|X_2 = x_2)$. If the inverse probability weight $\mathbb{P}(X_2 = \mathbf{X}_{2[i,\cdot]})^{-1}$ is known, then $\hat{e}_{\text{match}}(f)$ is unbiased for $e_{\text{cond}}(f)$ (see Appendix A.7). The weight $\mathbb{P}(X_2 = \mathbf{X}_{2[i,\cdot]})^{-1}$ accounts for the fact that, for any given value x_2, the proportion of observations of X_2 taking the value x_2 will generally not be the same as the proportion of matched pairs $(X_2^{(a)}, X_2^{(b)})$ taking value the x_2, and so simply summing over all matched pairs would lead to bias. In practice, the proportion $\mathbb{P}(X_2 = \mathbf{X}_{2[i,\cdot]})$ can be approximated as $\frac{1}{n-1} \sum_{j' \neq i} 1(\mathbf{X}_{2[i,\cdot]} = \mathbf{X}_{2[j',\cdot]})$, with minor adjustments to Eq 8.3 to avoid dividing by zero. The resulting estimate is analogous to exact matching procedures commonly used in causal inference, which are known to work best when the covariates are discrete and low dimensional, in order for exact matches to be common (Stuart, 2010).

However, when the covariate space is continuous or high dimensional, we typically cannot estimate CMR nonparametrically. For such cases, we propose to estimate CMR under an assumption of homogeneous residuals. Specifically, we define μ_1 to be the conditional expectation function $\mu_1(x_2) = \mathbb{E}(X_1|X_2 = x_2)$, and assume that the random residual $X - \mu_1(X_2)$ is independent of X_2. Under this assumption, it can be shown that

$$e_{\text{cond}}(f) = \mathbb{E}L\left[f, (Y^{(b)}, \left\{X_1^{(a)} - \mu_1(X_2^{(a)})\right\} + \mu_1(X_2^{(b)}), X_2^{(b)})\right].$$

That is, $e_{\text{cond}}(f)$ is equal to the expected loss of f across random pairs of observations $(Z^{(a)}, Z^{(b)})$ in which the value *of the residual terms* (in curly braces) have been switched. Because of the independence assumption, no matching or weighting is required. If μ_1 is known, then we can again produce an unbiased estimate using the U-statistic

$$\hat{e}_{\text{impute}}(f) := \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} L\left[f, (\mathbf{y}_{[j]}, \left\{\mathbf{X}_{1[i,\cdot]} - \mu_1(\mathbf{X}_{2[i,\cdot]})\right\} + \mu_1(\mathbf{X}_{2[j,\cdot]}), \mathbf{X}_{2[j,\cdot]})\right].$$

This estimator aggregates over all pairs in our sample, switching the values of the residual terms (in curly braces) within each pair. In practice, when μ_1 is not known, an estimate of μ_1 can be achieved via regression or related machine learning techniques, and plugged in to the above equation. In this way, the assumption that $X - \mu_1(X_2) \perp X_2$ allows us to estimate CMR without explicitly modeling the joint distribution of X_1 and X_2.

In the existing literature, Strobl et al. (2008) introduce a similar procedure for estimating conditional variable importance. However, a formal comparison to Strobl et al. is complicated by the fact that the authors do not define a specific estimand, and that their approach is limited to tree-based regression models. Other existing approaches conditional importance approaches include methods for redefining X_1 and X_2 to induce approximate

independence, before computing an importance measure analogous to MR. This can be done by reducing the total number of covariates used, and hence reducing how well any one variable can be predicted by the others (as in Gregorutti et al., 2017). Alternatively, variables in X_2 that are predictive of X_1 can be regrouped directly into X_1 (as in Tolosi and Lengauer, 2011; see also the discussion from Kirk, Lewin and Stumpf, in Meinshausen and Bühlmann 2010).

In summary, CMR allows us to see how much a model relies on the information uniquely available in X_1. While CMR is more difficult to estimate than MR, several tractable approaches exist when X_2 is discrete, or when a homogenous residual assumption can be applied. One may also consider extending CMR by conditioning only on a subset of X_2. For example, we may consider conditioning only on elements of X_2 that are believed to causally effect X_1, by changing the outer expectation in Eq 8.2. For simplicity, we focus on the base case of estimating MR in this paper. Similar results could potentially be carried over for CMR as well.

9. Simulations

In this section, we first present a toy example to illustrate the concepts of MR, MCR, and AR. We then present a Monte Carlo simulation studying the effectiveness of bootstrap CIs for MCR.

9.1. Illustrative Toy Example with Simulated Data

To illustrate the concepts of MR, MCR, and AR (see Section 3.2), we consider a toy example where $X = (X_1, X_2) \in \mathbb{R}^2$, and $Y \in \{-1, 1\}$ is a binary group label. Our primary goal in this section is to build intuition for the differences between these three importance measures, and so we demonstrate them here only in a single sample. We focus on the empirical versions of our importance metrics ($\widehat{MR}$, $\widehat{MCR}_-$ and $\widehat{MCR}_+$), and compare them against AR, which is typically interpreted as an in-sample measure (Breiman, 2001), or as an intermediate step to estimate an alternate importance measure in terms of variable rankings (Gevrey et al., 2003; Olden et al., 2004).

We simulate $X|Y = -1$ from an independent, bivariate normal distribution with means $\mathbb{E}(X_1|Y = -1) = \mathbb{E}(X_2|Y = -1) = 0$ and variances $\text{Var}(X_1|Y = -1) = \text{Var}(X_2|Y = -1) = \frac{1}{9}$. We simulate $X|Y = 1$ by drawing from the same bivariate normal distribution, and then adding the value of a random vector $(C_1, C_2) := (\cos(U), \sin(U))$, where U is a random variable uniformly distributed on the interval $[-\pi, \pi]$. Thus, (C_1, C_2) is uniformly distributed across the unit circle.

Given a prediction model $f : \mathcal{X} \to \mathbb{R}$, we use the sign of $f(X_1, X_2)$ as our prediction of Y. For our loss function, we use the hinge loss $L(f, (y, x_1, x_2)) = (1 - yf(x_1, x_2))_+$, where $(a)_+ = a$ if $a \geq 0$ and $(a)_+ = 0$ otherwise. The hinge loss function is commonly used as a convex approximation to the zero-one loss $L(f, (y, x_1, x_2)) = 1[y \neq \text{sign}\{f(x_1, x_2)\}]$.

We simulate two samples of size 300 from the data generating process described above, one to be used for training, and one to be used for testing. Then, for the class of models

used to predict Y, we consider the set of degree-3 polynomial classifiers

$$\mathcal{F}_{d3} = \big\{ f_{\boldsymbol{\theta}} : f_{\boldsymbol{\theta}}(x_1, x_2) = \boldsymbol{\theta}_{[1]} + \boldsymbol{\theta}_{[2]} x_1 + \boldsymbol{\theta}_{[3]} x_2 + \boldsymbol{\theta}_{[4]} x_1^2 + \boldsymbol{\theta}_{[5]} x_2^2 + \boldsymbol{\theta}_{[6]} x_1 x_2 + \boldsymbol{\theta}_{[7]} x_1^3 + + \boldsymbol{\theta}_{[8]} x_2^3 + \boldsymbol{\theta}_{[9]} x_1^2 x_2 + \boldsymbol{\theta}_{[10]} x_1 x_2^2;\ \ \|\boldsymbol{\theta}_{[-1]}\|_2^2 \le r_{d3} \big\},$$

where $\boldsymbol{\theta}_{[-1]}$ denotes all elements of $\boldsymbol{\theta}$ except $\boldsymbol{\theta}_{[1]}$, and where we set r_{d3} to the value that minimizes the 10-fold cross-validated loss in the training data. Let $\mathcal{A}_{d3}$ be the algorithm that minimizes the hinge loss over the (convex) feasible region $\{f_\theta : \|\boldsymbol{\theta}_{[-1]}\|_2^2 \le r_{d3}\}$. We apply $\mathcal{A}_{d3}$ to the training data to determine a reference model f_{ref}. Also using the training data, we set ϵ equal to 0.10 multiplied by the cross-validated loss of $\mathcal{A}_{d3}$, such that $\mathcal{R}(\epsilon, f_{\text{ref}}, \mathcal{F}_{d3})$ contains all models in $\mathcal{F}_{d3}$ that exceed the loss of f_{ref} by no more than approximately 10% (see Eq 4.1). We then calculate empirical AR, MR, and MCR using the test observations.

We begin by considering the AR of $\mathcal{A}_{d3}$ on X_1. Calculating AR requires us to fit two separate models, first using all of the variables to fit a model on the training data, and then again using only X_2. In this case, the first model is equivalent to f_{ref}. We denote the second model as $\hat{f}_2$. To compute AR, we evaluate f_{ref} and $\hat{f}_2$ in the test observations. We illustrate this AR computation in Figure 6-A, marking the classification boundaries for f_{ref} and $\hat{f}_2$ by the black dotted line and the blue dashed lines respectively, and marking the test observations by labelled points ("x" for $Y = 1$, and "o" for $Y = -1$). Comparing the loss associated with these two models gives one form of AR–an estimate of the necessity of X_1 for the algorithm $\mathcal{A}_{d3}$. Alternatively, to estimate the *sufficiency* of X_1, we can compare the reference model f_{ref} against the model resulting from retraining algorithm $\mathcal{A}_{d3}$ only using X_1. We refer to this third model as $\hat{f}_1$, and mark its classification boundary by the solid blue lines in Figure 6-A.

Each of the classifiers in Figure 6-A can also be evaluated for its reliance on X_1, as shown in Figure 6-C. Here, we use $\hat{e}_{\text{divide}}$ in our calculation of $\widehat{MR}$ (see Eq 3.5). Unsurprisingly, the classifier fit without using X_1 (blue dashed line) has a model reliance of $\widehat{MR}(\hat{f}_2) = 1$. The reference model f_{ref} (dotted black line) has a model reliance of $\widehat{MR}(f_{\text{ref}}) = 3.47$. Each $\widehat{MR}$ value has an interpretation contained to a single model. That is, $\widehat{MR}$ compares a *single model's* behavior under different data distributions, rather than the AR approach of comparing *different models'* behavior on marginal distributions from a single joint distribution.

We illustrate MCR in Figure 6-B. In contrast to AR, MCR is only ever a function of well-performing prediction models. Here, we consider the empirical ϵ-Rashomon set $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F}_{d3})$, the subset of models in $\mathcal{F}_{d3}$ with test loss no more than ϵ above that of f_{ref}. We show the classification boundary associated with 15 well-performing models contained in $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F}_{d3})$ by the gray solid lines. We also show two of the models in $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F}_{d3})$ that approximately maximize and minimize empirical reliance on X_1 among models in $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F}_{d3})$. We denote these models as $\hat{f}_{+,\epsilon}$ and $\hat{f}_{-,\epsilon}$, and mark them by the solid green and dashed green lines respectively. For every model shown in Figure 6-B, we also mark its model reliance in Figure 6-C. We can then see from Figure 6-C that $\widehat{MR}$ for each model in $\hat{\mathcal{R}}(\epsilon, f_{\text{ref}}, \mathcal{F}_{d3})$ is contained between $\widehat{MR}(\hat{f}_{-,\epsilon})$ and $\widehat{MR}(\hat{f}_{+,\epsilon})$, up to a small approximation error.

In summary, unlike AR, MCR is only a function of models that fit the data well.

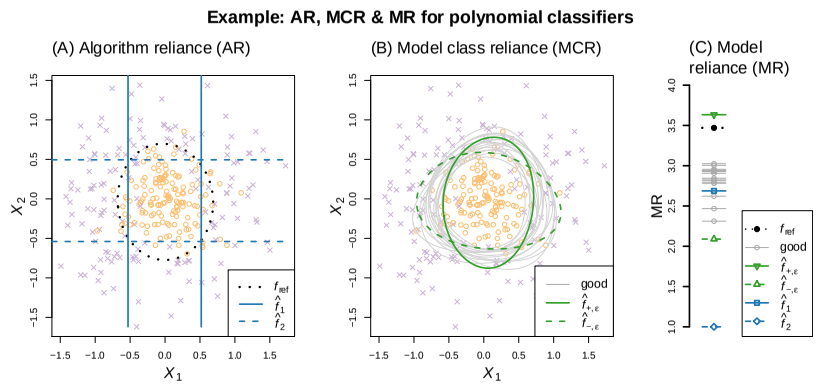


Figure 6: Example of AR, MCR & MR for polynomial classifiers. Panels (A) & (B) show the same 300 draws from a simulated data set, with the classification of each data point marked by "x" for $Y = 1$, and "o" for $Y = -1$. In Panel (A), for AR, we show single-feature models formed by dropping a covariate. Because these models take only a single input, we represent their classification boundaries as straight lines. In Panel (B), for MCR, we show the classification boundaries for several (two-feature) models with low in-sample loss. Of these models, the model with minimal dependence on X_1 is shown by the dashed green oval, and the model with maximal dependence on X_1 is shown by the solid green oval. Panel (C) shows the empirical model reliance on X_1 for each of the models in Panels (A) & (B). We see in Panel (C) that, as expected, no well-performing model relies (empirically) on X_1 more than $\hat{f}_{+,\epsilon}$ does, or relies (empirically) on X_1 less than $\hat{f}_{-,\epsilon}$ does. That is, no well-performing model has an empirical MR value greater than $\widehat{MCR}_+(\epsilon)$, or less than $\widehat{MCR}_-(\epsilon)$.

9.2. Simulations of Bootstrap Confidence Intervals

In this section we study the performance of MCR under model class misspecification. Our goal will be to estimate how much the conditional expectation function $f_0(x) = \mathbb{E}(Y|X = x)$ relies on subsets of covariates. Given a reference model f_{ref} and model class $\mathcal{F}$, our ability to describe $MR(f_0)$ will hinge on two conditions:

Condition 20 (Nearly correct model class) *The class $\mathcal{F}$ contains a well-performing model $\tilde{f} \in \mathcal{R}(\epsilon, f_{ref}, \mathcal{F})$ satisfying $MR(\tilde{f}) = MR(f_0)$ (see Eq 4.1).*

Condition 21 (Bootstrap coverage) *Bootstrap CIs for empirical MCR give appropriate coverage of population-level MCR.*

Condition 20 ensures that the interval $[MCR_-(\epsilon), MCR_+(\epsilon)]$ contains $MR(f_0)$, and Condition 21 ensures that this interval can be estimated in finite samples. Condition 20 can also be interpreted as saying that the model reliance value of $MR(f_c)$ is "well supported" by the class $\mathcal{F}$, even if $\mathcal{F}$ does not contain f_0. Our primary goal is to assess whether CIs derived from MCR can give appropriate coverage of $MR(f_0)$, which depends on both conditions. As a secondary goal, we also would like to be able to assess Conditions 20 & 21 individually.

Verifying the above conditions requires that we are able to calculate population-level MCR. To this end, we draw samples with replacement from a finite population of 20,000 observations, in which MCR can also be calculated directly. To derive a CI based on MCR, we divide each simulated sample $\mathcal{Z}_s$ into a training subset and analysis subset. We use the training subset to fit a reference model $f_{\text{ref},s}$, which is required for our definition of population-level MCR. We calculate a bootstrap CI by drawing 500 bootstrap samples from the analysis subset, and computing $\widehat{MCR}_-(\epsilon)$ and $\widehat{MCR}_+(\epsilon)$ in each bootstrap sample by optimizing over $\hat{\mathcal{R}}(\epsilon, f_{\text{ref},s}, \mathcal{F})$. We then take the 2.5% percentile of $\widehat{MCR}_-(\epsilon)$ values across bootstrap samples, and the 97.5% percentile of $\widehat{MCR}_+(\epsilon)$ values across bootstrap samples, as the lower and upper endpoints of our CI, respectively. We repeat this procedure for both X_1 and X_2.

We generate data according to a model with increasing amounts of nonlinearity. For $\gamma \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, we simulate continuous outcomes as $Y = f_0(X) + E$, where f_0 is the function $f_0(\mathbf{x}) = \sum_{j=1}^{p} j\mathbf{x}_{[j]} - \gamma\mathbf{x}_{[j]}^2$; the covariate dimension p is equal to 2, with X_1 and X_2 defined as the first and second elements of X; the covariates X are drawn from a multivariate normal distribution with $\mathbb{E}(X_1) = \mathbb{E}(X_2) = 0$, $\text{Var}(X_1) = \text{Var}(X_2) = 0$, and $\text{Cov}(X_1, X_2) = 1/4$; and E is a normally distributed noise variable with mean zero and variance equal to $\sigma_E^2 := \text{Var}(f_0(X))$. We consider sample sizes of $n = 400$ and 800, of which $n_{tr} = 200$ or 300 observations are assigned to the training subset respectively.

To implement our approach, we use the model class $\mathcal{F}_{\text{lm}} = \{f_{\boldsymbol{\beta}} : f_{\boldsymbol{\beta}}(\mathbf{x}) = \boldsymbol{\beta}_{[1]} + \sum_{j=1}^{2} \mathbf{x}_{[j]}\boldsymbol{\beta}_{[j+1]}, \boldsymbol{\beta} \in \mathbb{R}^3\}$. We set the performance threshold ϵ equal to $0.1 \times \sigma_E^2$. We refer to this MCR implementation with $\mathcal{F}_{\text{lm}}$ as "MCR-Linear."

As a comparator method, we consider a simpler bootstrap approach, which we refer to as "Standard-Linear." Here, we take 500 bootstrap samples from the simulated data $\mathcal{Z}_s$. In each bootstrap sample, indexed by b, we set aside n_{tr} training points to train a model $f_b \in \mathcal{F}_{\text{lm}}$, and calculate $\widehat{MR}(f_b)$ from the remaining data points. We then create a 95% bootstrap percentile CI for $MR(f_0)$ by taking the 2.5% and 97.5% percentiles of $\widehat{MR}(f_b)$ across $b = 1, \ldots, 500$.

9.2.1. Results

Overall, we find that MCR provides more robust and conservative intervals for the reliance of f_0 on X_1 and X_2, relative to standard bootstrap approaches. We also find that higher sample size generally exacerbates coverage errors due to misspecification, as methods become more certain of biased results.

MCR-Linear gave proper coverage for up to moderate levels of misspecification ($\gamma = 0.3$), where Standard-Linear began to break down (Figure 7). For larger levels of misspecification ($\gamma \geq 0.4$), both MCR-Linear and Standard-Linear failed to give appropriate coverage.

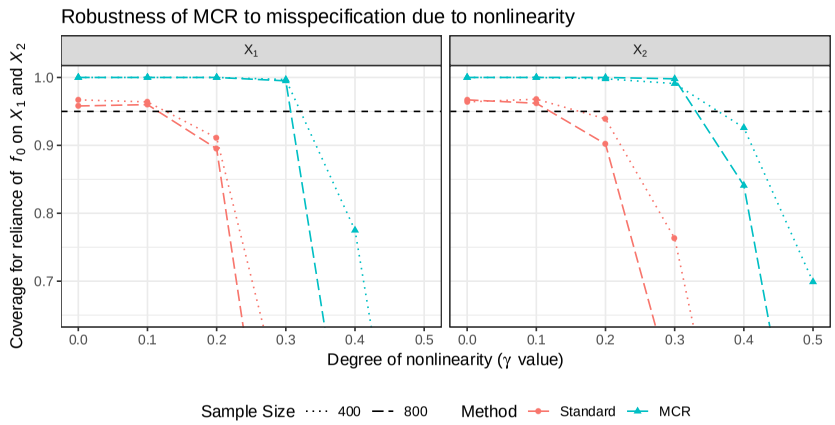


Figure 7: MR Coverage - The y-axis shows coverage rate for the reliance of f_0 on either X_1 (left column) or X_2 (right column), where X_2 is simulated to be more influential that X_1. The x-axis shows increasing levels of misspecification (γ). All methods aim to have at least 95% coverage for each scenario (dashed horizontal line).

The increased robustness of MCR comes at the cost of wider CIs. Intervals for MCR-Linear were typically larger than intervals for Standard-Linear by a factor of approximately 2-4. This is partly due to the fact that CIs for MCR are meant to cover the range of values $[MCR_-(\epsilon), MCR_+(\epsilon)]$ (defined using $f_{\text{ref},s}$), rather than to cover a single point.

When investigating Conditions 20 & 21 individually, we find that the coverage errors for MCR-Linear were largely attributable to violations of Condition 20. Condition 21 appears to hold conservatively for all scenarios studied–within each scenario, at least 95.9% of bootstrap CIs contained population-level MCR.

These simulation results highlight an aspect of MCR that is both a strength and a weakness: MCR is generic. MCR does not assume a particular means by which misspecification may occur, and is less powerful than sensitivity analyses which make that assumption correctly. Nonetheless, MCR still appears to add robustness. For sufficiently strong signals, an informative interval may still be returned. In our applied data analysis, below, we see that this is indeed the case.

10. Data Analysis: Reliance of Criminal Recidivism Prediction Models on Race and Sex

Evidence suggests that bias exists among judges and prosecutors in the criminal justice system (Spohn, 2000; Blair et al., 2004; Paternoster and Brame, 2008). In an aim to counter this bias, machine learning models trained to predict recidivism are increasingly

being used to inform judges' decisions on pretrial release, sentencing, and parole (Monahan and Skeem, 2016; Picard-Fritsche et al., 2017). Ideally, prediction models can avoid human bias and provide judges with empirically tested tools. But prediction models can also mirror the biases of the society that generates their training data, and perpetuate the same bias at scale. In the case of recidivism, if arrest rates across demographic groups are not representative of underlying crime rate (Beckett et al., 2006; Ramchand et al., 2006; U.S. Department of Justice - Civil Rights Devision, 2016), then bias can be created in both (1) the outcome variable, future crime, which is measured imperfectly via arrests or convictions, and (2) the covariates, which include the number of prior convictions on a defendant's record (Corbett-Davies et al., 2016; Lum and Isaac, 2016). Further, when a prediction model's behavior and mechanisms are an opaque black box, the model can evade scrutiny, and fail to offer recourse or explanations to individuals rated as "high risk."

We focus here on the issue of transparency, which takes an important role in the recent debate about the proprietary recidivism prediction tool COMPAS (Larson et al., 2016; Corbett-Davies et al., 2016). While COMPAS is known to not rely explicitly on race, there is concern that it may rely implicitly on race via proxies–variables statistically dependent with race (see further discussion in Section 11).

Our goal is to identify bounds for how much COMPAS relies on different covariate subsets, either implicitly or explicitly, under certain assumptions (defined below). We analyze a public data set of defendants from Broward County, Florida, in which COMPAS scores have been recorded (Larson et al., 2016). Within this data set, we only included defendants measured as African-American or Caucasian (3,373 in total) due to sparseness in the remaining categories. The outcome of interest (Y) is the COMPAS violent recidivism score. Of the available covariates, we consider three variables which we refer to as "admissible": an individual's age, their number of priors, and an indicator of whether the current charge is a felony. We also consider two variables which we refer to as "inadmissible": an individual's race and sex. Our labels of "admissible" and "inadmissible" are not intended to be legally precise–indeed, the boundary between these types of labels is not always clear (see Section 10.2). We compute empirical MCR and AR for each variable group, as well as bootstrap CIs for MCR (see Section 9.2).

To compute empirical MCR and AR, we consider a flexible class of linear models in a RKHS to predict the COMPAS score (described in more detail below). Given this class, the MCR range (See Eq 2.2) captures the highest and lowest degree to which any model in the class may rely on each covariate subset. We assume that our class contains at least one model that relies on "inadmissible variables" to the same extent that COMPAS relies either on "inadmissible variables" or on proxies that are unmeasured in our sample (analogous to Condition 20). We make the same assumption for "admissible variables." These assumptions can be interpreted as saying that the reliance values of COMPAS are relatively "well supported" by our chosen model class, and allows us to identify bounds on the MR values for COMPAS. We also consider the more conventional, but less robust approach of AR (Section 3.2), that is, how much would the accuracy suffer for a model-fitting algorithm trained on COMPAS score if a variable subset was removed?

These computations require that we predefine our loss function, model class, and performance threshold. We define MR, MCR, and AR in terms of the squared error loss $L(f,(y,x_1,x_2)) = \{y - f(x_1,x_2)\}^2$. We define our model class $\mathcal{F}_{\mathbf{D},r_k}$ in the form of Eq 7.6,

where we determine $\mathbf{D}$, μ, k, and r_k based on a subset $\mathcal{S}$ of 500 training observations. We set $\mathbf{D}$ equal to the matrix of covariates from $\mathcal{S}$; we set μ equal to the mean of Y in $\mathcal{S}$; we set k equal to the radial basis function $k_{\sigma_s}(\mathbf{x},\tilde{\mathbf{x}}) = \exp\left(-\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|^2}{2\sigma_s}\right)$, where we choose σ_s to minimize the cross-validated loss of a Nadaraya-Watson kernel regression (Hastie et al., 2009) fit to $\mathcal{S}$; and we select the parameters r_k by cross-validation on $\mathcal{S}$. We set ϵ equal to 0.1 times the cross-validated loss on $\mathcal{S}$. Also using $\mathcal{S}$, we train a reference model $f_{\text{ref}} \in \mathcal{F}_{\mathbf{D},r_k}$. Using the held-out 2,873 observations, we then estimate $MR(f_{\text{ref}})$ and MCR for $\mathcal{F}_{\mathbf{D},r_k}$. To calculate AR, we train models from $\mathcal{F}_{\mathbf{D},r_k}$ using $\mathcal{S}$, and evaluate their performance in the held-out observations.

10.1. Results

Our results imply that race and sex play somewhere between a null role and a modest role in determining COMPAS score, but that they are less important than "admissible" factors (Figure 8). As a benchmark for comparison, the empirical MR of f_{ref} is equal to 1.09 for "inadmissible variables," and 2.78 for "admissible variables." The AR is equal to 0.94 and 1.87 for "inadmissible" and "admissible" variables respectively, roughly in agreement with MR. The MCR range for "inadmissible variables" is equal to [1.00,1.56], indicating that *for any model in* $\mathcal{F}_{\mathbf{D},r_k}$ with empirical loss no more than ϵ above that of f_{ref}, the model's loss can increase by no more than 56% if race and sex are permuted. Such a statement cannot be made solely based on AR or MR methods, as these methods do not upper bound the reliance values of well-performing models. The bootstrap 95% CI for MCR on "inadmissible variables" is [1.00, 1.73]. Thus, *under our assumptions, if COMPAS relied on sex, race, or their unmeasured proxies by a factor greater than 1.73, then intervals as low as what we observe would occur with probability* < 0.05.

For "admissible variables" the MCR range is equal to [1.77,3.61], with a 95% bootstrap CI of [1.62, 3.96]. *Under our assumptions, this implies if COMPAS relied on age, number of priors, felony indication, or their unmeasured proxies by a factor lower than 1.77, then intervals as high as what we observe would occur with probability* < 0.05. This result is consistent with Rudin et al. (2019), who find age to be highly predictive of COMPAS score.

It is worth noting that the upper limit of 3.61 maximizes empirical MR on "admissible variables" not only among well-performing models, but globally across all models in the class (see Figure 8, and Eq 6.5). In other words, it is not possible to find models in $\mathcal{F}_{\mathbf{D},r_k}$ that perform arbitrarily poorly on perturbed data, but still perform well on unperturbed data, and so the ratio of $\hat{e}_{\text{switch}}(f)$ to $\hat{e}_{\text{orig}}(f)$ has a finite upper bound. Because the regularization constraints of $\mathcal{F}_{\mathbf{D},r_k}$ preclude MR values higher than 3.61, the MR of COMPAS on "admissible variables" may be underestimated by empirical MCR. Note also that both MCR intervals are left-truncated at 1, as it is often sufficiently precise to conclude that there exists a well-performing model with no reliance on the variables of interest (that is, MR equal to 1; see Appendix A.2).

10.2. Discussion & Limitations

Asking whether a proprietary model relies on sex and race, after adjusting for other covariates, is related to the fairness metric known as conditional statistical parity (CSP). A decision rule satisfies CSP if its decisions are independent of a sensitive variable, conditional

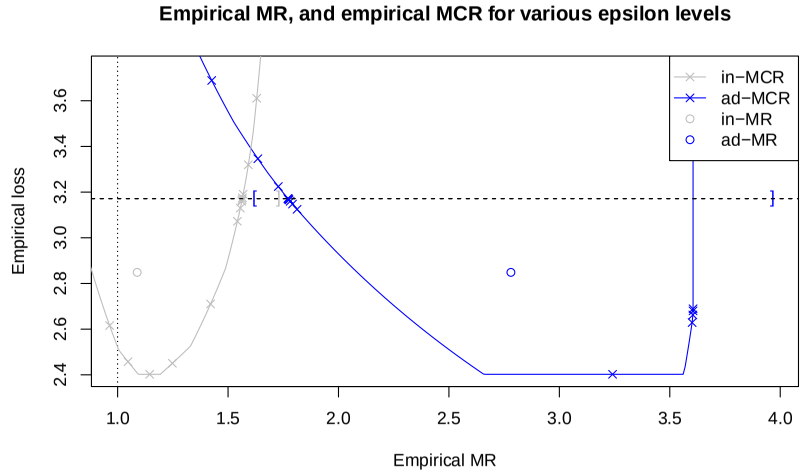


Figure 8: Empirical MR and MCR for Broward County criminal records data set - For any prediction model f, the y-axis shows empirical loss ($\hat{e}_{\text{stnd}}(f)$) and the x-axis shows empirical reliance ($\widehat{MR}(f)$) on each covariate subset. Null reliance (MR equal to 1.0) is marked by the vertical dotted line. Reliances on different covariate subsets are marked by color ("admissible" = blue; "inadmissible" = gray). For example, model reliance values for f_{ref} are shown by the two circular points, one for "admissible" variables and one for "inadmissible" variables. MCR for different values of ϵ can be represented as boundaries on this coordinate space. To this end, for each covariate subset, we compute conservative boundary functions (shown as solid lines, or "bowls") guaranteed to contain *all models in the class* (see Section 6). Specifically, all models in $f \in \mathcal{F}_{\mathbf{D},r_k}$ are guaranteed to have an empirical loss ($\hat{e}_{\text{stnd}}(f)$) and empirical MR value ($\widehat{MR}(f)$) for "inadmissible variables" corresponding to a point within the gray bowl. Likewise, all models in $\mathcal{F}_{\mathbf{D},r_k}$ are guaranteed to have an empirical loss and empirical MR value for "admissible variables" corresponding to a point within the blue bowl. Points shown as "×" represent additional models in $\mathcal{F}_{\mathbf{D},r_k}$ discovered during our computational procedure, and thus show where the "bowl" boundary is tight. The goal of our computation procedure (see Section 6) is to tighten the boundary as much as possible near the ϵ value of interest, shown by the dashed horizontal line above. This dashed line has a y-intercept equal to the loss of the reference model plus the ϵ value of interest. Bootstrap CIs for $MCR_-(\epsilon)$ and $MCR_+(\epsilon)$ are marked by brackets.

on a set of "legitimate" covariates C (Corbett-Davies et al., 2017; see also Kamiran et al., 2013). Roughly speaking, CSP reflects the idea that groups of people with similar covariates C are treated similarly (Dwork et al., 2012), regardless of the sensitive variable (for example, race or sex). However, the criteria becomes superficial if too many variables are included in C, and care should be taken to avoid including proxies for the sensitive variables. Several other fairness metrics have also been proposed, which often form competing objectives (Kleinberg et al., 2017; Chouldechova, 2017; Nabi and Shpitser, 2018; Corbett-Davies et al., 2017). Here, if COMPAS was not influenced by race, sex, or variables related to race or sex (conditional on a set of "legitimate" variables), it would satisfy CSP.

Unfortunately, it is often difficult to distinguish between "legitimate" (or "admissible") variables and "illegitimate" variables. Some variables function both as part of a reasonable predictor for risk, and, separately, as a proxy for race. Because of disproportional arrest rates, particularly for misdemeanors and drug-related offenses (U.S. Department of Justice - Civil Rights Devision, 2016; Lum and Isaac, 2016), prior misdemeanor convictions may act as such a proxy (Corbett-Davies et al., 2016; Lum and Isaac, 2016).

Proxy variables for race (defined as being statistically dependent with race) that are unmeasured in our sample are also not the only reason that race could be predictive of COMPAS score. Other inputs to the COMPAS algorithm might be associated with race *only conditionally* on variables we categorize as "admissible." However, our result from Section 10.1 that race has limited predictive utility for COMPAS score suggests that such conditional relationships are also limited.

11. Conclusion

In this article, we propose MCR as the upper and lower limit on how important a set of variables can be to any well-performing model in a class. In this way, MCR provides a more comprehensive and robust measure of importance than traditional importance measures for a single model. We derive bounds on MCR, which motivate our choice of point estimates. We also derive connections between permutation importance, U-statistics, conditional variable importance, and conditional causal effects. We apply MCR in a data set of criminal recidivism, in order to help inform the characteristics of the proprietary model COMPAS.

Several exciting areas remain open for future research. One research direction closely related to our current work is the development of exact or approximate MCR computation procedures for other model classes and loss functions. We have shown that, for model classes where minimizing the empirical loss is a convex optimization problem, MCR can be conservatively computed via a series of convex optimization problems. Further, we have shown that computing $\widehat{MCR}_-$ is often no more challenging that minimizing the empirical loss over a reweighted sample. General computation procedures for MCR are still an open research area.

Another direction is to consider MCR for variable selection. If MCR_+ is small for a variable, then no well-performing predictive model can heavily depend on that variable, indicating that it can be eliminated.

Our theoretical analysis of Rashomon sets depends on $\mathcal{F}$ and f_{ref} being prespecified. Above, we have actualized this by splitting our sample into subsets of size n_1 and n_2, using the first subset to determine $\mathcal{F}$ and f_{ref}, and conditioning on $\mathcal{F}$ and f_{ref} when estimating

MCR in the second subset. As a result, the boundedness constants in our assumptions (B_{ind}, $B_{\text{ref}}, B_{\text{switch}}$, and b_{orig}) depend on $\mathcal{F}$, and hence on n_1. However, because our results are non-asymptotic, we have not explored how Rashomon sets behave when n_1 and n_2 grow at different rates. An exciting future extension of this work is to study sequences of triples $\{\epsilon_{n_1}, f_{\text{ref},n_1}, \mathcal{F}_{n_1}\}$ that change as n_1 increases, and the corresponding Rashomon sets $\mathcal{R}(\epsilon_{n_1}, f_{\text{ref},n_1}, \mathcal{F}_{n_1})$, as this may more thoroughly capture how model classes are determined by analysts.

While we develop Rashomon sets with the goal of studying MR, Rashomon sets can also be useful for finite sample inferences about a wide variety of other attributes of best-in-class models (for example, Section 5). Characterizations of a Rashomon set itself may also be of interest. For example, in ongoing work, we are studying the size of a Rashomon set, and its connection to generalization of models and model classes (Semenova and Rudin, 2019). We are additionally developing methods for visualizing Rashomon sets (Dong and Rudin, 2019).

Acknowledgments

Support for this work was provided by the National Institutes of Health (grants P01CA134294, R01GM111339, R01ES024332, R35CA197449, R01ES026217, P50MD010428, DP2MD012722, R01MD012769, & R01ES028033), by the Environmental Protection Agency (grants 83615601 & 83587201-0), and by the Health Effects Institute (grant 4953-RFA14-3/16-4).

Appendix A. Miscellaneous Supplemental Sections

All labels for items in the following appendices begin with a letter (for example, Section A.2), while references to items in the main text contain only numbers (for example, Proposition 19).

A.1. Code

R code for our example in Section 9.1 and analysis in Section 10 is available at `https://github.com/aaronjfisher/mcr-supplement`.

A.2. Model Reliance Less than 1

While it is counterintuitive, it is possible for the expected loss of a prediction model to *decrease* when the information in X_1 is removed. Roughly speaking, a "pathological" model f_{silly} may use the information in X_1 to "intentionally" misclassify Y, such that $e_{\text{switch}}(f_{\text{silly}}) < e_{\text{orig}}(f_{\text{silly}})$ and $MR(f_{\text{silly}}) < 1$. The model f_{silly} may even be included in a population ϵ-Rashomon set (see Section 4) if it is still possible to predict Y sufficiently well from the information in X_2.

However, in these cases there will often exist another model that outperforms f_{silly}, and that has MR equal to 1 (i.e., no reliance on X_1). To see this, consider the case where $\mathcal{F} = \{f_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \mathbb{R}^d\}$ is indexed by a parameter $\boldsymbol{\theta}$. Let $\boldsymbol{\theta}_{\text{silly}}$ and $\boldsymbol{\theta}^\star$ be parameter values

such that $f_{\boldsymbol{\theta}_{\text{silly}}}$ is equivalent to f_{silly}, and $f_{\boldsymbol{\theta}^\star}$ is the best-in-class model. If $f_{\boldsymbol{\theta}^\star}$ satisfies $MR(f_{\boldsymbol{\theta}^\star}) > 1$ and if the model reliance function MR is continuous in $\boldsymbol{\theta}$, then there exists a parameter value $\boldsymbol{\theta}_1$ between $\boldsymbol{\theta}_{\text{silly}}$ and $\boldsymbol{\theta}^\star$ such that $MR(f_{\boldsymbol{\theta}_1}) = 1$. Further, if the loss function L is convex in $\boldsymbol{\theta}$, then $e_{\text{orig}}(f_{\boldsymbol{\theta}^\star}) \leq e_{\text{orig}}(f_{\boldsymbol{\theta}_1}) \leq e_{\text{orig}}(f_{\text{silly}})$, and any population ϵ-Rashomon set containing f_{silly} will also contain $f_{\boldsymbol{\theta}_1}$.

A.3. Relating $\hat{e}_{\text{switch}}(f)$ to All Possible Permutations of the Sample

Following the notation in Section 3, let $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_{n!}\}$ be a set of n-length vectors, each containing a different permutation of the set $\{1, \ldots, n\}$. We show in this section that $\hat{e}_{\text{switch}}(f)$ is equal to the product of

$$\sum_{l=1}^{n!} \sum_{i=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[\boldsymbol{\pi}_{l[i]},\cdot]}, \mathbf{X}_{2[i,\cdot]})\} 1(\boldsymbol{\pi}_{l[i]} \neq i), \tag{A.1}$$

and a proportionality constant that is only a function of n.

First, consider the sum

$$\sum_{l=1}^{n!} \sum_{i=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[\boldsymbol{\pi}_{l[i]},\cdot]}, \mathbf{X}_{2[i,\cdot]})\}, \tag{A.2}$$

which omits the indicator function found in Eq A.1.

The summation in Eq A.2 contains $n(n!)$ terms, each of which is a two-way combination of the form $L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[j,\cdot]}, \mathbf{X}_{2[i,\cdot]})\}$ for $i, j \in \{1, \ldots n\}$. There are only n^2 unique combinations of this form, and each must occur in at least $(n-1)!$ of the $n(n!)$ terms in Eq A.2. To see this, consider selecting two integer values $\tilde{i}, \tilde{j} \in \{1, \ldots n\}$, and enumerating all occurrences of the term $L\{f, (\mathbf{y}_{[\tilde{i}]}, \mathbf{X}_{1[\tilde{j},\cdot]}, \mathbf{X}_{2[\tilde{i},\cdot]})\}$ within the sum in Eq A.2. Of the permutation vectors $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_{n!}\}$, we know that $(n-1)!$ of them place $\tilde{i}$ in the $\tilde{j}^{th}$ position, i.e., that satisfy $\boldsymbol{\pi}_{l[\tilde{i}]} = \tilde{j}$. For each such permutation $\boldsymbol{\pi}_l$, the inner summation in Eq A.2 over all possible values of i must include the term $L\{f, (\mathbf{y}_{[\tilde{i}]}, \mathbf{X}_{1[\boldsymbol{\pi}_{l[\tilde{i}]},\cdot]}, \mathbf{X}_{2[\tilde{i},\cdot]})\} = L\{f, (\mathbf{y}_{[\tilde{i}]}, \mathbf{X}_{1[\tilde{j},\cdot]}, \mathbf{X}_{2[\tilde{i},\cdot]})\}$. Thus, Eq A.2 contains at least $(n-1)!$ occurrences of the term $L\{f, (\mathbf{y}_{[\tilde{i}]}, \mathbf{X}_{1[\tilde{j},\cdot]}, \mathbf{X}_{2[\tilde{i},\cdot]})\}$.

So far, we have shown that each unique combination occurs at least $(n-1)!$ times, but it also follows that each unique combination must occur precisely $(n-1)!$ times. This is because each of the n^2 unique combinations must occur at least $(n-1)!$ times, which accounts for $n^2((n-1)!) = n(n!)$ terms in total. As noted above, Eq has A.2 has only $n(n)!$ terms, so there can be no additional terms. We can then simplify Eq A.2 as

$$\sum_{l=1}^{n!} \sum_{i=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[\boldsymbol{\pi}_{l[i]},\cdot]}, \mathbf{X}_{2[i,\cdot]})\} = (n-1)! \sum_{i=1}^{n} \sum_{j=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[j,\cdot]}, \mathbf{X}_{2[i,\cdot]})\}.$$

By the same logic, we can simplify Eq A.1 as

$$
\begin{aligned}
&\sum_{l=1}^{n!}\sum_{i=1}^{n} L\{f,(\mathbf{y}_{[i]},\mathbf{X}_{1[\boldsymbol{\pi}_{l[i]},\cdot]},\mathbf{X}_{2[i,\cdot]})\}1(\boldsymbol{\pi}_{l[i]}\neq i)\\
=&(n-1)!\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} L\{f,(\mathbf{y}_{[i]},\mathbf{X}_{1[j,\cdot]},\mathbf{X}_{2[i,\cdot]})\}1(j\neq i)\right\}\\
=&(n-1)!\sum_{i=1}^{n}\sum_{j\neq i} L\{f,(\mathbf{y}_{[i]},\mathbf{X}_{1[j,\cdot]},\mathbf{X}_{2[i,\cdot]})\},
\end{aligned}
\tag{A.3}
$$

and Line A.3 is proportional to $\hat{e}_{\text{switch}}(f)$ up to a function of n.

A.4. Bound for MR of the Best-in-class Prediction Model

Although describing individual models is not the primary focus of this work, a corollary of Theorem 4 is that we can create a probabilistic bound for the reliance of the (unknown) best-in-class model $f^\star$ on X_1.

Corollary 22 *(Bound on Best-in-class MR) Let $f^\star \in \arg\min_{f\in\mathcal{F}} e_{orig}(f)$ be a prediction model that attains the lowest possible expected loss, and let $f_{+,\epsilon}$ and $f_{-,\epsilon}$ be defined as in Theorem 4. If $f_{+,\epsilon}$ and $f_{-,\epsilon}$ satisfy Assumptions 1, 2 and 3, then*

$$
\mathbb{P}\left(MR(f^\star)\in\left[\widehat{MCR}_-(\epsilon_{best})-\mathcal{Q}_{best},\quad \widehat{MCR}_+(\epsilon_{best})+\mathcal{Q}_{best}\right]\right)\geq 1-\delta,
$$

where $\epsilon_{best} := 2B_{ref}\sqrt{\frac{\log(6\delta^{-1})}{2n}}$ *, and* $\mathcal{Q}_{best} := \frac{B_{switch}}{b_{orig}} - \frac{B_{switch}-B_{ind}\sqrt{\frac{\log(12\delta^{-1})}{n}}}{b_{orig}+B_{ind}\sqrt{\frac{\log(12\delta^{-1})}{2n}}}$.

The above result does not require that $f^\star$ be unique. If several models achieve the minimum possible expected loss, the above boundaries apply simultaneously for each of them. In the special case when the true conditional expectation function $\mathbb{E}(Y|X_1,X_2)$ is equal to $f^\star$, then we have a boundary for the reliance of the function $\mathbb{E}(Y|X_1,X_2)$ on X_1. This reliance bound can also be translated into a causal statement using Proposition 19.

A.5. Ratios versus Differences in MR Definition

We choose our ratio-based definition of model reliance, $MR(f) = \frac{e_{\text{switch}}(f)}{e_{\text{orig}}(f)}$, so that the measure can be comparable across problems, regardless of the scale of Y. However, several existing works define VI measures in terms of differences (Strobl et al., 2008; Datta et al., 2016; Gregorutti et al., 2017), analogous to

$$
MR_{\text{difference}}(f) := e_{\text{switch}}(f) - e_{\text{orig}}(f). \tag{A.4}
$$

While this difference measure is less readily interpretable, it has several computational advantages. The mean, variance, and asymptotic distribution of the estimator $\widehat{MR}_{\text{difference}}(f) := \hat{e}_{\text{switch}}(f) - \hat{e}_{\text{orig}}(f)$ can be easily determined using results for U-statistics, without the use of the delta method (Dorfman, 1938; Lehmann and Casella, 2006; see also Ver Hoef, 2012).

Estimates in the form of $\widehat{MR}_{\text{difference}}(f)$ will also be more stable when $\min_{f\in\mathcal{F}} e_{\text{orig}}(f)$ is small, relative to estimates for the ratio-based definition of MR. To improve interpretability, we may also normalize $MR_{\text{difference}}(f)$ by dividing by the variance of Y, which can be easily estimated without the use of models, as in Williamson et al. (2017).

Under the difference-based definition for MR (Eq A.4), the results from Theorem 4, Theorem 6, and Corollary 22 will still hold under the following modified definitions of $\mathcal{Q}_{\text{out}}$, $\mathcal{Q}_{\text{in}}$, and $\mathcal{Q}_{\text{best}}$:

$$
\begin{aligned}
\mathcal{Q}_{\text{out,difference}} &:= \left(1+\frac{1}{\sqrt{2}}\right)B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}},\\
\mathcal{Q}_{\text{in,difference}} &:= B_{\text{ind}}\left\{\sqrt{\frac{\log(8\delta^{-1}\mathcal{N}(\mathcal{F},r\sqrt{2}))}{n}}+\sqrt{\frac{\log(8\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}}\right\}+2r(\sqrt{2}+1),\ \text{and}\\
\mathcal{Q}_{\text{best,difference}} &:= \left(1+\frac{1}{\sqrt{2}}\right)B_{\text{ind}}\sqrt{\frac{\log(12\delta^{-1})}{n}}.
\end{aligned}
$$

Respectively replacing $\mathcal{Q}_{\text{out}}$, $\mathcal{Q}_{\text{in}}$, $\mathcal{Q}_{\text{best}}$, MR, and $\widehat{MR}$ with $\mathcal{Q}_{\text{out,difference}}$, $\mathcal{Q}_{\text{in,difference}}$, $\mathcal{Q}_{\text{best,difference}}$, $MR_{\text{difference}}$ and $\widehat{MR}_{\text{difference}}$ entails only minor changes to the corresponding proofs (see Appendices B.3, B.5, and B.4). The results will also hold without Assumption 3, as is suggested by the fact that b_{orig} and B_{switch} do not appear in $\mathcal{Q}_{\text{out,difference}}$, $\mathcal{Q}_{\text{in,difference}}$, or $\mathcal{Q}_{\text{best,difference}}$.

We also prove an analogous version of Theorem 5, on uniform bounds for $\widehat{MR}_{\text{difference}}$, in Appendix B.5.1.

A.6. Rashomon Sets and Profile Likelihood Intervals

We note in Section 5.1 that, under certain conditions, the CIs returned from Proposition 7 take the same form as profile likelihood CIs (Coker et al., 2018). For completeness, we briefly review this connection. We assume here that models $f_\theta \in \mathcal{F}$ are indexed by a finite dimensional parameter vector $\theta\in\Theta$, where $\theta=(\gamma,\psi)$ contains a 1-dimensional parameter of interest $\gamma\in\mathbb{R}^1$, and a nuisance parameter $\psi\in\Psi$. We further assume and that $e_{\text{orig}}(f_\theta)$ is minimized by a unique parameter value $\theta^\star=(\gamma^\star,\psi^\star)\in\Theta$, and that our goal is to learn about $\gamma^\star$.

If $s_\theta := \int_{\mathcal{Z}} \exp\{-L(f_\theta,z)\}dz$ is finite for all $\theta\in\Theta$, we can convert L into the likelihood function $\mathcal{L}:(\mathcal{Z}\times\Theta)\to\mathbb{R}^1$ satisfying $\mathcal{L}(z;\theta)=\exp\{-L(f_\theta,z)\}/s_\theta$. As an abbreviation, let $\mathcal{L}(\mathbf{Z};\theta)$ denote $\prod_{i=1}^n \mathcal{L}(\mathbf{Z}_{[i,\cdot]};\theta)$. Additionally, let $\hat{\theta} := \arg\min_{\theta\in\Theta}\hat{e}_{\text{orig}}(f_\theta)$ be the empirical loss minimizer, and hence the maximum likelihood estimator of $\theta^\star$. If $\mathcal{L}$ is indeed the correct likelihood function, then $\theta^\star=(\gamma^\star,\psi^\star)$ corresponds to the true parameter vector. Further, if $\phi(f_\theta)=\phi(f_{(\gamma,\psi)})=\gamma$ returns the parameter element of interest (γ), then the $(1-\delta)$-level

profile likelihood interval for $\phi(f_{\theta^\star}) = \gamma^\star$ is

$$\begin{aligned}
\text{PLI}(\delta) :=& \left\{\gamma \,:\, \log \mathcal{L}(\mathbf{Z};\hat{\theta}) - \log \mathcal{L}(\mathbf{Z};\hat{\theta}_\gamma) \leq \frac{\chi_{1,1-\delta}}{2}, \text{ where } \hat{\theta}_\gamma = \underset{\{\theta\in\Theta\,:\,\phi(f_\theta)=\gamma\}}{\arg\max} \mathcal{L}(\mathbf{Z};\theta)\right\} \\
=& \left\{\gamma \,:\, \exists \hat{\theta}_\gamma \text{ satisfying } \phi\left(f_{\hat{\theta}_\gamma}\right) = \gamma \text{ and } \log \mathcal{L}(\mathbf{Z};\hat{\theta}) - \log \mathcal{L}(\mathbf{Z};\hat{\theta}_\gamma) \leq \frac{\chi_{1,1-\delta}}{2}\right\} \\
=& \left\{\gamma \,:\, \exists \hat{\theta}_\gamma \text{ satisfying } \phi\left(f_{\hat{\theta}_\gamma}\right) = \gamma \text{ and } \hat{e}_{\text{orig}}\left(f_{\hat{\theta}_\gamma}\right) \leq \hat{e}_{\text{orig}}\left(f_{\hat{\theta}}\right) + \frac{\chi_{1,1-\delta}}{2n}\right\} \\
=& \left\{\gamma \,:\, \exists f_{\hat{\theta}_\gamma} \text{ satisfying } \phi\left(f_{\hat{\theta}_\gamma}\right) = \gamma \text{ and } f_{\hat{\theta}_\gamma} \in \hat{\mathcal{R}}\left(\frac{\chi_{1,1-\delta}}{2n}, f_{\hat{\theta}}, \mathcal{F}\right)\right\} \qquad \text{(A.5)}
\end{aligned}$$

where $\chi_{1,1-\delta}$ is the $1-\delta$ percentile of a chi-square distribution with 1 degree of freedom. If $\text{PLI}(\alpha)$ is indeed a contiguous interval, then maximizing and minimizing $\phi(f_\theta)$ across models f_θ in the empirical Rashomon set in Eq A.5 yields the same interval.

A.7. Unbiased Estimates of CMR

We claim in Section 8.2 that both

$$\hat{e}_{\text{match}}(f) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j\neq i} \frac{1(\mathbf{X}_{2[j,\cdot]} = \mathbf{X}_{2[i,\cdot]})}{\mathbb{P}(X_2 = \mathbf{X}_{2[i,\cdot]})} \times L\{f, (\mathbf{y}_{[j]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]})\}.$$

and

$$\hat{e}_{\text{weight}}(f) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j\neq i} \frac{\mathbb{P}(X_1 = \mathbf{X}_{1[i,\cdot]} | X_2 = \mathbf{X}_{2[j,\cdot]})}{\mathbb{P}(X_1 = \mathbf{X}_{1[i,\cdot]})} \times L\{f, (\mathbf{y}_{[j]}, \mathbf{X}_{1[i,\cdot]}, \mathbf{X}_{2[j,\cdot]})\},$$

are unbiased for

$$e_{\text{cond}}(f) = \mathbb{E}_{X_2}\mathbb{E}\left[L\{f, (Y^{(b)}, X_1^{(a)}, X_2^{(b)})\} | X_2^{(a)} = X_2^{(b)}, X_2\right].$$

To show that $\hat{e}_{\text{match}}(f)$ is unbiased, we first note that each summation term in $\hat{e}_{\text{match}}(f)$ has the same expectation. Following the notation in Section 3, let $Z^{(a)} = (Y^{(a)}, X_1^{(a)}, X_2^{(a)})$ and $Z^{(b)} = (Y^{(b)}, X_1^{(b)}, X_2^{(b)})$ be independent random variables following the same distribution as $Z = (Y, X_1, X_2)$. The expectation of $\hat{e}_{\text{match}}(f)$ is

$$\begin{aligned}
\mathbb{E}\hat{e}_{\text{match}}(f) =& \mathbb{E}\left[\frac{1(X_2^{(a)} = X_2^{(b)})}{p_{x_2}(X_2^{(a)})} \times L\{f, (Y^{(b)}, X_1^{(a)}, X_2^{(b)})\}\right] \\
=& \mathbb{E}_{X_2^{(a)}}\mathbb{E}\left[\frac{1(X_2^{(a)} = X_2^{(b)})}{p_{x_2}(X_2^{(a)})} \times L\{f, (Y^{(b)}, X_1^{(a)}, X_2^{(b)})\} | X_2^{(a)}\right] \\
=& \mathbb{E}_{X_2^{(a)}}\left\{p_{x_2}(X_2^{(a)})\mathbb{E}\left[\frac{1}{p_{x_2}(X_2^{(a)})} \times L\{f, (Y^{(b)}, X_1^{(a)}, X_2^{(b)})\} | X_2^{(a)} = X_2^{(b)}, X_2^{(a)}\right] + 0\right\} \\
=& \mathbb{E}_{X_2^{(a)}}\mathbb{E}\left[L\{f, (Y^{(b)}, X_1^{(a)}, X_2^{(b)})\} | X_2^{(a)} = X_2^{(b)}, X_2^{(a)}\right] \\
=& e_{\text{cond}}(f).
\end{aligned}$$

To show that $\hat{e}_{\text{weight}}(f)$ is unbiased, we similarly note that each summation term in $\hat{e}_{\text{weight}}(f)$ has the same expectation. Without loss of generality, we show the result for discrete variables (Y, X_1, X_2). Let $\mathcal{Y}_{x_2}$ be the domain of Y conditional on the event that $X_2 = x_2$. The expectation of $\hat{e}_{\text{weight}}(f)$ is

$$\begin{aligned}
\mathbb{E}\hat{e}_{\text{weight}}(f) =& \sum_{x_2^{(b)}\in\mathcal{X}_2} \sum_{y^{(b)}\in\mathcal{Y}_{x_2^{(b)}}} \sum_{x_1^{(a)}\in\mathcal{X}_1} \left[L\{f, (y^{(b)}, x_1^{(a)}, x_2^{(b)})\}\left\{\frac{\mathbb{P}(X_1 = x_1^{(a)} | X_2 = x_2^{(b)})}{\mathbb{P}(X_1 = x_1^{(a)})}\right\}\right. \\
& \left. \times \mathbb{P}(X_1 = x_1^{(a)})\mathbb{P}(Y = y^{(b)}, X_2 = x_2^{(b)})\right] \\
=& \sum_{x_2^{(b)}\in\mathcal{X}_2} \mathbb{P}(X_2 = x_2^{(b)}) \sum_{y^{(b)}\in\mathcal{Y}_{x_2^{(b)}}} \sum_{x_1^{(a)}\in\mathcal{X}_1} \left[L\{f, (y^{(b)}, x_1^{(a)}, x_2^{(b)})\}\right. \\
& \left. \times \mathbb{P}(X_1 = x_1^{(a)} | X_2 = x_2^{(b)})\mathbb{P}(Y = y^{(b)} | X_2 = x_2^{(b)})\right] \\
=& \mathbb{E}_{X_2^{(b)}}\mathbb{E}\left[\int L\{f, (Y^{(b)}, X_1^{(a)}, X_2^{(b)})\} | X_2^{(a)} = X_2^{(b)}, X_2^{(b)})\right] \\
=& e_{\text{cond}}(f).
\end{aligned}$$

Appendix B. Proofs for Statistical Results

We present proofs for our statistical results in this section, and conclude by presenting proofs for our computational results in Appendix C.

B.1. Lemma Relating Empirical and Population Rashomon Sets

Throughout the remaining proofs, it will be useful to express the definition of population ϵ-Rashomon sets in terms of the expectation of a single loss function, rather than a comparison of two loss functions. To do this, we simply introduce the "standardized" loss function $\tilde{L}$, defined as

$$\tilde{L}(f, z) := L(f, z) - L(f_{\text{ref}}, z). \qquad \text{(B.1)}$$

Above, recall from Section 2 that $L(f, z)$ denotes $L(f, (y, x_1, x_2))$ for $z = (y, x_1, x_2)$. Because we assume f_{ref} is prespecified and fixed, we omit notation for f_{ref} in the definition of $\tilde{L}$. We can now write

$$\begin{aligned}
\mathcal{R}(\epsilon) &= \{f_{\text{ref}}\} \cup \{f \in \mathcal{F} : \mathbb{E}L(f, Z) \leq \mathbb{E}L(f_{\text{ref}}, Z) + \epsilon\} \\
&= \{f_{\text{ref}}\} \cup \left\{f \in \mathcal{F} : \mathbb{E}\tilde{L}(f, Z) \leq \epsilon\right\},
\end{aligned}$$

and, similarly,

$$\hat{\mathcal{R}}(\epsilon) = \{f_{\text{ref}}\} \cup \left\{f \in \mathcal{F} : \hat{\mathbb{E}}\tilde{L}(f, Z) \leq \epsilon\right\}.$$

With this definition, the following lemma allows us to limit the probability that a given model $f_1 \in \mathcal{R}(\epsilon)$ is excluded from an empirical Rashomon set.

Lemma 23 *For $\epsilon \in \mathbb{R}$ and $\delta \in (0,1)$, let $\epsilon_1' := \epsilon + 2B_{ref}\sqrt{\frac{\log(\delta^{-1})}{2n}}$, and let $f_1 \in \mathcal{R}(\epsilon)$ denote a specific, possibly unknown prediction model. If f_1 satisfies Assumption 2, then*

$$\mathbb{P}\{f_1 \in \hat{\mathcal{R}}(\epsilon_1')\} \geq 1 - \delta.$$

Proof If f_{ref} and f_1 are the same function, then the result holds trivially. Otherwise, the proof follows from Hoeffding's inequality (Theorem 2 of Hoeffding, 1963). First, note that if f_1 satisfies Assumption 2, then $\tilde{L}(f_1)$ is bounded within an interval of length $2B_{\text{ref}}$. Applying this in line B.3, below, we see that

$$\begin{aligned}
\mathbb{P}\{f_1 \notin \hat{\mathcal{R}}(\epsilon_1')\} &= \mathbb{P}\left[\hat{\mathbb{E}}\tilde{L}(f_1, Z) > \epsilon_1'\right] && \text{from } f_1 \notin \{f_{\text{ref}}\} \\
&= \mathbb{P}\left[\hat{\mathbb{E}}\tilde{L}(f_1, Z) - \epsilon > 2B_{\text{ref}}\sqrt{\frac{\log(\delta^{-1})}{2n}}\right] && \text{from definition of } \epsilon_1' \qquad \text{(B.2)} \\
&\leq \mathbb{P}\left[\hat{\mathbb{E}}\tilde{L}(f_1, Z) - \mathbb{E}\tilde{L}(f_1, Z) > 2B_{\text{ref}}\sqrt{\frac{\log(\delta^{-1})}{2n}}\right] && \text{from } \mathbb{E}\tilde{L}(f_1, Z) \leq \epsilon \\
&\leq \exp\left\{-\frac{2n}{(2B_{\text{ref}})^2}\left[2B_{\text{ref}}\sqrt{\frac{\log(\delta^{-1})}{2n}}\right]^2\right\} && \text{from Hoeffding's inequality} \qquad \text{(B.3)} \\
&= \delta. && \text{(B.4)}
\end{aligned}$$

For the inequality used in Line B.3, see Theorem 2 of Hoeffding, 1963. ■

B.2. Lemma to Transform Between Bounds

The following lemma will help us translate from bounds for variables to bounds for differences and ratios of those variables. We will apply this lemma to transform from bounds on empirical losses to bounds on empirical model reliance, defined either in terms of a ratio or in terms of a difference.

Lemma 24 *Let $X, Z, \mu_X, \mu_Z, k_X, k_Z \in \mathbb{R}$ be constants satisfying $|Z - \mu_Z| \leq k_Z$ and $|X - \mu_X| \leq k_X$, then*

$$|(Z - X) - (\mu_Z - \mu_X)| \leq q_{difference}(k_Z, k_X), \tag{B.5}$$

where $q_{difference}$ is the function

$$q_{difference}(k_Z, k_X) := k_Z + k_X. \tag{B.6}$$

Further, if there exists constants b_{orig} and B_{switch} such that $0 < b_{orig} \leq X, \mu_X$ and $Z, \mu_Z \leq B_{switch} < \infty$, then

$$\left|\frac{Z}{X} - \frac{\mu_Z}{\mu_X}\right| \leq q_{ratio}(k_Z, k_X), \tag{B.7}$$

where q_{ratio} is the function

$$q_{ratio}(k_Z, k_X) := \frac{B_{switch}}{b_{orig}} - \frac{B_{switch} - k_Z}{b_{orig} + k_X}. \tag{B.8}$$

Proof Showing Eq B.5,

$$\begin{aligned}
|(Z - X) - (\mu_Z - \mu_X)| &\leq |Z - \mu_Z| + |\mu_X - X| \\
&\leq k_Z + k_X.
\end{aligned}$$

Showing Eq B.7, let $A_Z = \max(Z, \mu_Z)$, $a_X = \min(X, \mu_X)$, $d_Z = |Z - \mu_Z|$, and $d_X = |X - \mu_X|$. This implies that $\max(X, \mu_X) = a_X + d_X$ and $\min(Z, \mu_Z) = A_Z - d_Z$. Thus, $\frac{Z}{X}$ and $\frac{\mu_Z}{\mu_X}$ are both bounded within the interval

$$\left[\frac{\min(Z, \mu_Z)}{\max(X, \mu_X)}, \frac{\max(Z, \mu_Z)}{\min(X, \mu_X)}\right] = \left[\frac{A_Z - d_Z}{a_X + d_X}, \frac{A_Z}{a_X}\right],$$

which implies

$$\left|\frac{Z}{X} - \frac{\mu_Z}{\mu_X}\right| \leq \frac{A_Z}{a_X} - \frac{A_Z - d_Z}{a_X + d_X}. \tag{B.9}$$

Taking partial derivatives of the right-hand side, we get

$$\begin{aligned}
\frac{\partial}{\partial a_X}\left(\frac{A_Z}{a_X} - \frac{A_Z - d_Z}{a_X + d_X}\right) &= \frac{-A_Z}{a_X^2} + \frac{A_Z - d_Z}{(a_X + d_X)^2} \leq 0, \\
\frac{\partial}{\partial A_Z}\left(\frac{A_Z}{a_X} - \frac{A_Z - d_Z}{a_X + d_X}\right) &= \frac{1}{a_X} - \frac{1}{a_X + d_X} \geq 0, \\
\frac{\partial}{\partial d_X}\left(\frac{A_Z}{a_X} - \frac{A_Z - d_Z}{a_X + d_X}\right) &= \frac{A_Z - d_Z}{(a_X + d_X)^2} > 0, \\
\text{and } \frac{\partial}{\partial d_Z}\left(\frac{A_Z}{a_X} - \frac{A_Z - d_Z}{a_X + d_X}\right) &= \frac{1}{a_X + d_X} > 0.
\end{aligned}$$

So the right-hand side of B.9 is maximized when d_Z, d_X, and A_Z are maximized, and when a_X is minimized. Thus, in the case where $|Z - \mu_Z| \leq k_Z$; $|X - \mu_X| \leq k_X$; $0 < b_{\text{orig}} \leq X, \mu_X$; and $Z, \mu_Z \leq B_{\text{switch}} < \infty$, we have

$$\left|\frac{Z}{X} - \frac{\mu_Z}{\mu_X}\right| \leq \frac{A_Z}{a_X} - \frac{A_Z - d_Z}{a_X + d_X}$$
$$\leq \frac{B_{\text{switch}}}{b_{\text{orig}}} - \frac{B_{\text{switch}} - k_Z}{b_{\text{orig}} + k_X}.$$

■

B.3. Proof of Theorem 4

Proof We proceed in 4 steps.

B.3.1. Step 1: Show that $\mathbb{P}\left[\widehat{MR}(f_{+,\epsilon}) \leq \widehat{MCR}_+(\epsilon_{\text{OUT}})\right] \geq 1 - \frac{\delta}{3}$.

Consider the event that

$$\widehat{MR}(f_{+,\epsilon}) \leq \widehat{MCR}_+(\epsilon_{\text{out}}). \tag{B.10}$$

Eq B.10 will always hold if $f_{+,\epsilon} \in \hat{\mathcal{R}}(\epsilon_{\text{out}})$, since $\widehat{MCR}_+(\epsilon_{\text{out}})$ upper bounds the empirical model reliance for models in $\hat{\mathcal{R}}(\epsilon_{\text{out}})$ by definition. Applying the above reasoning in Line B.11, below, we get

$$\mathbb{P}\left[\widehat{MR}(f_{+,\epsilon}) > \widehat{MCR}_+(\epsilon_{\text{out}})\right] \leq \mathbb{P}\left[f_{+,\epsilon} \notin \hat{\mathcal{R}}(\epsilon_{\text{out}})\right] \tag{B.11}$$
$$\leq \frac{\delta}{3} \qquad \text{from } \epsilon_{\text{out}} \text{ definition and Lemma 23.} \tag{B.12}$$

B.3.2. Step 2: Conditional on $\widehat{MR}(f_{+,\epsilon}) \leq \widehat{MCR}_+(\epsilon_{\text{OUT}})$, Upper Bound $MR(f_{+,\epsilon})$ by $\widehat{MCR}_+(\epsilon_{\text{OUT}})$ Added to an Error Term.

When Eq B.10 holds we have,

$$\begin{aligned}
\widehat{MR}(f_{+,\epsilon}) &\leq \widehat{MCR}_+(\epsilon_{\text{out}}) \\
\widehat{MR}(f_{+,\epsilon}) &\leq \widehat{MCR}_+(\epsilon_{\text{out}}) + \{MR(f_{+,\epsilon}) - MR(f_{+,\epsilon})\} \\
MR(f_{+,\epsilon}) &\leq \widehat{MCR}_+(\epsilon_{\text{out}}) + [MR(f_{+,\epsilon}) - \widehat{MR}(f_{+,\epsilon})].
\end{aligned} \tag{B.13}$$

B.3.3. Step 3: Probabilistically Bound the Error Term from Step 2.

Next we show that the bracketed term in Line B.13 is less than or equal to $\mathcal{Q}_{\text{out}}$ with high probability. For $k \in \mathbb{R}$, let $q_{\text{difference}}$ and q_{ratio} be defined as in Eqs B.6 and B.8. Let

$q : \mathbb{R} \to \mathbb{R}$ be the function such that $q(k) = q_{\text{ratio}}\left(k, \frac{k}{\sqrt{2}}\right)$. Then

$$\begin{aligned}
\mathcal{Q}_{\text{out}} &= \frac{B_{\text{switch}}}{b_{\text{orig}}} - \frac{B_{\text{switch}} - B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}}{b_{\text{orig}} + B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{2n}}} \\
&= q_{\text{ratio}}\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}, B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{2n}}\right) \\
&= q\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}\right).
\end{aligned}$$

Applying this relation below, we have

$$\mathbb{P}\left[MR(f_{+,\epsilon}) - \widehat{MR}(f_{+,\epsilon}) > \mathcal{Q}_{\text{out}}\right] \tag{B.14}$$
$$\leq \mathbb{P}\left[\left|MR(f_{+,\epsilon}) - \widehat{MR}(f_{+,\epsilon})\right| > q\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}\right)\right]$$
$$\leq \mathbb{P}\left[\left\{|\hat{e}_{\text{orig}}(f_{+,\epsilon}) - e_{\text{orig}}(f_{+,\epsilon})| > B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{2n}}\right\}\right.$$
$$\left.\bigcup\left\{|\hat{e}_{\text{switch}}(f_{+,\epsilon}) - e_{\text{switch}}(f_{+,\epsilon})| > B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}\right\}\right] \qquad \text{from Lemma 24}$$
$$\leq \mathbb{P}\left[|\hat{e}_{\text{orig}}(f_{+,\epsilon}) - e_{\text{orig}}(f_{+,\epsilon})| > B_{\text{ind}}\sqrt{\frac{\log\left(6\delta^{-1}\right)}{2n}}\right]$$
$$+ \mathbb{P}\left[|\hat{e}_{\text{switch}}(f_{+,\epsilon}) - e_{\text{switch}}(f_{+,\epsilon})| > B_{\text{ind}}\sqrt{\frac{\log\left(6\delta^{-1}\right)}{n}}\right] \qquad \text{from the Union bound}$$
$$\leq 2\exp\left\{-\frac{2n}{(B_{\text{ind}} - 0)^2}\left[B_{\text{ind}}\sqrt{\frac{\log\left(6\delta^{-1}\right)}{2n}}\right]^2\right\}$$
$$+ 2\exp\left\{-\frac{n}{(B_{\text{ind}} - 0)^2}\left[B_{\text{ind}}\sqrt{\frac{\log\left(6\delta^{-1}\right)}{n}}\right]^2\right\} \qquad \text{from Hoeffding's bound for U-statistics} \tag{B.15}$$
$$= \frac{2\delta}{6} + \frac{2\delta}{6} = \frac{2\delta}{3}. \tag{B.16}$$

In Line B.15, above, recall that $\hat{e}_{\text{orig}}(f_{+,\epsilon})$ and $\hat{e}_{\text{switch}}(f_{+,\epsilon})$ are both U-statistics. Note that $\mathbb{E}\left[\hat{e}_{\text{switch}}(f_{+,\epsilon})\right] = e_{\text{switch}}(f_{+,\epsilon})$ because $\hat{e}_{\text{switch}}(f_{+,\epsilon})$ is an average of terms, and each term has expectation equal to $e_{\text{switch}}(f_{+,\epsilon})$. For the same reason, $\mathbb{E}\left[\hat{e}_{\text{orig}}(f_{+,\epsilon})\right] = e_{\text{orig}}(f_{+,\epsilon})$. This allows us to apply Eq 5.7 of Hoeffding, 1963 (see also Eq 1 on page 201 of Serfling, 1980, in Theorem A) to obtain Line B.15.

Alternatively, if we instead define model reliance as $MR_{\text{difference}}(f) = e_{\text{switch}}(f) - e_{\text{orig}}(f)$ (see Appendix A.5), define empirical model reliance as $\widehat{MR}_{\text{difference}}(f) := \hat{e}_{\text{switch}}(f) - \hat{e}_{\text{orig}}(f)$, and define

$$\mathcal{Q}_{\text{out,difference}} := \left(1 + \frac{1}{\sqrt{2}}\right) B_{\text{ind}} \sqrt{\frac{\log(6\delta^{-1})}{n}} = q_{\text{difference}}\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}, B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{2n}}\right),$$

then the same proof holds without Assumption 3 if we replace MR, $\widehat{MR}$, $\mathcal{Q}_{\text{out}}$ respectively with $MR_{\text{difference}}$, $\widehat{MR}_{\text{difference}}$, $\mathcal{Q}_{\text{out,difference}}$, and redefine $q : \mathbb{R} \to \mathbb{R}$ as the function $q(k) = q_{\text{difference}}\left(k, \frac{k}{\sqrt{2}}\right)$.

Eqs B.14-B.16 also hold if we replace $\hat{e}_{\text{switch}}$ throughout with $\hat{e}_{\text{divide}}$, including in Assumption 3, since the same bound can be used for both $\hat{e}_{\text{switch}}$ and $\hat{e}_{\text{divide}}$ (Eq 5.7 of Hoeffding, 1963; see also Theorem A on page 201 of Serfling, 1980).

B.3.4. Step 4: Combine Results to Show Eq 4.2

Finally, we connect the above results to show Eq 4.2. We know from Eq B.12 that Eq B.10 holds with high probability. Eq B.10 implies Eq B.13, which bounds $MCR_+(\epsilon) = MR(f_{+,\epsilon})$ up to a bracketed residual term. We also know from Eq B.16 that, with high probability, the residual term in Eq B.13 is less than $\mathcal{Q}_{\text{out}} = q\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}\right)$. Putting this together, we can show Eq 4.2:

$$\begin{aligned}
&\mathbb{P}\left(MCR_+(\epsilon) > \widehat{MCR}_+(\epsilon_{\text{out}}) + \mathcal{Q}_{\text{out}}\right) \\
&\quad = \mathbb{P}\left(MR(f_{+,\epsilon}) > \widehat{MCR}_+(\epsilon_{\text{out}}) + \mathcal{Q}_{\text{out}}\right) \\
&\quad \le \mathbb{P}\left[\left(\widehat{MR}(f_{+,\epsilon}) > \widehat{MCR}_+(\epsilon_{\text{out}})\right) \bigcup \left(MR(f_{+,\epsilon}) - \widehat{MR}(f_{+,\epsilon}) > \mathcal{Q}_{\text{out}}\right)\right] \quad \text{from Step 2} \\
&\quad \le \mathbb{P}\left[\widehat{MR}(f_{+,\epsilon}) > \widehat{MCR}_+(\epsilon_{\text{out}})\right] + \mathbb{P}\left[MR(f_{+,\epsilon}) - \widehat{MR}(f_{+,\epsilon}) > \mathcal{Q}_{\text{out}}\right] \\
&\quad \le \frac{\delta}{3} + \frac{2\delta}{3} = \delta. \qquad \text{from Steps 1 \& 3}
\end{aligned} \tag{B.17}$$

This completes the proof for Eq 4.2. For Eq 4.3 we can use the same approach, shown below for completeness. Analogous to Eq B.12, we have

$$\mathbb{P}\left[\widehat{MR}(f_{-,\epsilon}) < \widehat{MCR}_-(\epsilon_{\text{out}})\right] \le \frac{\delta}{3}.$$

Analogous to Eq B.13, when $\widehat{MR}(f_{-,\epsilon}) \ge \widehat{MR}(\hat{f}_{-,\epsilon_{\text{out}}})$ we have

$$\begin{aligned}
\widehat{MR}(f_{-,\epsilon}) &\ge \widehat{MCR}_-(\epsilon_{\text{out}}) \\
\widehat{MR}(f_{-,\epsilon}) &\ge \widehat{MCR}_-(\epsilon_{\text{out}}) + \{MR(f_{-,\epsilon}) - MR(f_{-,\epsilon})\} \\
MR(f_{-,\epsilon}) &\ge \widehat{MCR}_-(\epsilon_{\text{out}}) - \left[\widehat{MR}(f_{-,\epsilon}) - MR(f_{-,\epsilon})\right].
\end{aligned}$$

Analogous to Eq B.16, we have

$$\mathbb{P}\left[\widehat{MR}(f_{-,\epsilon}) - MR(f_{-,\epsilon}) > q\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}\right)\right] \le \frac{2\delta}{3}. \tag{B.18}$$

Finally, analogous to Eq B.17, we have

$$\begin{aligned}
&\mathbb{P}\left(MCR_-(\epsilon) < \widehat{MCR}_-(\epsilon_{\text{out}}) - \mathcal{Q}_{\text{out}}\right) \\
&\quad \le \mathbb{P}\left[\left(\widehat{MR}(f_{-,\epsilon}) < \widehat{MCR}_-(\epsilon_{\text{out}})\right) \bigcup \left(\widehat{MR}(f_{-,\epsilon}) - MR(f_{-,\epsilon}) > q\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}\right)\right)\right] \qquad \text{(B.19)} \\
&\quad \le \mathbb{P}\left[\widehat{MR}(f_{-,\epsilon}) < \widehat{MCR}_-(\epsilon_{\text{out}})\right] + \mathbb{P}\left[\widehat{MR}(f_{-,\epsilon}) - MR(f_{-,\epsilon}) > q\left(B_{\text{ind}}\sqrt{\frac{\log(6\delta^{-1})}{n}}\right)\right] \\
&\quad \le \frac{\delta}{3} + \frac{2\delta}{3} = \delta.
\end{aligned}$$

Again, the same proof holds without Assumption 3 if we replace MR, $\widehat{MR}$, $\mathcal{Q}_{\text{out}}$ respectively with $MR_{\text{difference}}$, $\widehat{MR}_{\text{difference}}$, $\mathcal{Q}_{\text{out,difference}}$, and redefine q as the function satisfying $q(k) = q_{\text{difference}}\left(k, \frac{k}{\sqrt{2}}\right)$ in Eqs B.18 & B.19. ■

B.4. Proof of Corollary 22

Proof By definition, $MR(f_{-,\epsilon_{\text{best}}}) \le MR(f^\star) \le MR(f_{+,\epsilon_{\text{best}}})$. Applying this relation in Line B.20, below, we see

$$\begin{aligned}
&\mathbb{P}\left(MR(f^\star) \in \left[\widehat{MCR}_-(\epsilon_{\text{best}}) - \mathcal{Q}_{\text{best}}, \quad \widehat{MCR}_+(\epsilon_{\text{best}}) + \mathcal{Q}_{\text{best}}\right]\right) \\
&\quad = 1 - \mathbb{P}\left\{MR(f^\star) < \widehat{MCR}_-(\epsilon_{\text{best}}) - \mathcal{Q}_{\text{best}} \bigcup MR(f^\star) > \widehat{MCR}_+(\epsilon_{\text{best}}) + \mathcal{Q}_{\text{best}}\right\} \\
&\quad \ge 1 - \mathbb{P}\left\{MR(f^\star) < \widehat{MCR}_-(\epsilon_{\text{best}}) - \mathcal{Q}_{\text{best}}\right\} - \mathbb{P}\left\{MR(f^\star) > \widehat{MCR}_+(\epsilon_{\text{best}}) + \mathcal{Q}_{\text{best}}\right\} \\
&\quad \ge 1 - \mathbb{P}\left\{MR(f_{-,\epsilon}) < \widehat{MCR}_-(\epsilon_{\text{best}}) - \mathcal{Q}_{\text{best}}\right\} - \mathbb{P}\left\{MR(f_{+,\epsilon}) > \widehat{MCR}_+(\epsilon_{\text{best}}) + \mathcal{Q}_{\text{best}}\right\} \qquad \text{(B.20)} \\
&\quad \ge 1 - \frac{\delta}{2} - \frac{\delta}{2} \qquad \text{from Theorem 4.} \qquad \text{(B.21)}
\end{aligned}$$

To apply Theorem 4 in Line B.21, above, we note that $\mathcal{Q}_{\text{best}}$ and ϵ_{best} are equivalent to the definitions of $\mathcal{Q}_{\text{out}}$ and ϵ_{out} in Theorem 4, but with δ replaced by $\frac{\delta}{2}$.

Alternatively, if we define model reliance as $MR_{\text{difference}}(f) = e_{\text{switch}}(f) - e_{\text{orig}}(f)$ (see Appendix A.5), and define empirical model reliance as $\widehat{MR}_{\text{difference}}(f) = \hat{e}_{\text{switch}}(f) - \hat{e}_{\text{orig}}(f)$, then let

$$\mathcal{Q}_{\text{best,difference}} := \left(1 + \frac{1}{\sqrt{2}}\right) B_{\text{ind}} \sqrt{\frac{\log(12\delta^{-1})}{n}}.$$

The term $\mathcal{Q}_{\text{best,difference}}$ is equivalent to $\mathcal{Q}_{\text{out,difference}}$ but with δ replaced with $\frac{\delta}{2}$. Under this difference-based definition of model reliance, Theorem 4 holds without Assumption 3 if we replace $\mathcal{Q}_{\text{out}}$ with $\mathcal{Q}_{\text{out,difference}}$ (see Section B.3), and so we can apply this altered version of Theorem 4 in Line B.21. Thus, Theorem 22 also holds without Assumption 3 if we replace MR, $\widehat{MR}$, and $\mathcal{Q}_{\text{best}}$ respectively with $MR_{\text{difference}}$, $\widehat{MR}_{\text{difference}}$, and $\mathcal{Q}_{\text{best,difference}}$. ■

B.5. Proof of Theorems 5 & 6

We begin by proving Theorem 5, along with related results. We then apply these results to show Theorem 6.

B.5.1. Proof of Theorem 5, and Other Limits on Estimation Error, Based on Covering Number

The following theorem uses the covering number based on r-margin-expectation-covers to jointly bound empirical losses for any function $f \in \mathcal{F}$. Theorem 5 in the main text follows directly from Eq B.25, below.

Theorem 25 *If Assumptions 1, 2 and 3 hold for all $f \in \mathcal{F}$, then for any $r > 0$*

$$\mathbb{P}_D\left[\sup_{f\in\mathcal{F}} |\hat{e}_{orig}(f) - e_{orig}(f)| > B_{ind}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right] \leq \delta, \tag{B.22}$$

$$\mathbb{P}_D\left[\sup_{f\in\mathcal{F}} \left|\hat{\mathbb{E}}\tilde{L}(f,Z) - \mathbb{E}\tilde{L}(f,Z)\right| > 2B_{ref}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right] \leq \delta, \tag{B.23}$$

$$\mathbb{P}_D\left[\sup_{f\in\mathcal{F}} |\hat{e}_{switch}(f) - e_{switch}(f)| > B_{ind}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{n}} + 2r\right] \leq \delta, \tag{B.24}$$

$$\mathbb{P}\left[\sup_{f\in\mathcal{F}} \left|\frac{\hat{e}_{orig}(f)}{\hat{e}_{switch}(f)} - \frac{e_{orig}(f)}{e_{switch}(f)}\right| > \mathcal{Q}_4\right] \leq \delta, \tag{B.25}$$

$$\mathbb{P}_D\left[\sup_{f\in\mathcal{F}} |\{\hat{e}_{switch}(f) - \hat{e}_{orig}(f)\} - \{e_{switch}(f) - e_{orig}(f)\}| > \mathcal{Q}_{4,difference}\right] \leq \delta, \tag{B.26}$$

where

$$\mathcal{Q}_4 := q_{ratio}\left(B_{ind}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r\sqrt{2}))}{n}} + 2r\sqrt{2}, B_{ind}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right), \tag{B.27}$$

$$\mathcal{Q}_{4,difference} := q_{difference}\left(B_{ind}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r\sqrt{2}))}{n}} + 2r\sqrt{2}, B_{ind}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right), \tag{B.28}$$

and q_{ratio} and $q_{difference}$ are defined as in Lemma 24. For Eq B.26, the result is unaffected if we remove Assumption 3.

B.5.2. Proof of Eq B.22

Proof Let $\mathcal{G}_r$ be a r-margin-expectation-cover for $\mathcal{F}$ of size $\mathcal{N}(\mathcal{F},r)$. Let D_p denote the population distribution, let D_s be the sample distribution, and let $D^\star$ be the uniform mixture of D_p and D_s, i.e., for any $z \in \mathcal{Z}$,

$$\mathbb{P}_{D^\star}(Z \leq z) = \frac{1}{2}\mathbb{P}_{D_p}(Z \leq z) + \frac{1}{2}\mathbb{P}_{D_s}(Z \leq z). \tag{B.29}$$

Unless otherwise stated, we take expectations and probabilities with respect to D_p. Since $\mathcal{G}_r$ is a r-margin-expectation-cover, we know that for any $f \in \mathcal{F}$ we can find a function $g \in \mathcal{G}_r$ such that $\mathbb{E}_{D^\star}|L(g,Z) - L(f,Z)| = \mathbb{E}_{D^\star}\left|\tilde{L}(g,Z) - \tilde{L}(f,Z)\right| \leq r$, and

$$\begin{aligned}
\left|\hat{\mathbb{E}}L(f,Z) - \mathbb{E}L(f,Z)\right| &= \left|\hat{\mathbb{E}}L(f,Z) - \mathbb{E}L(f,Z) + \left\{\hat{\mathbb{E}}L(g,Z) - \hat{\mathbb{E}}L(g,Z)\right\} + \{\mathbb{E}L(g,Z) - \mathbb{E}L(g,Z)\}\right| \qquad \text{(B.30)}\\
&\leq \left|\hat{\mathbb{E}}L(g,Z) - \mathbb{E}L(g,Z)\right| + |\mathbb{E}L(g,Z) - \mathbb{E}L(f,Z)| + \left|\hat{\mathbb{E}}L(f,Z) - \hat{\mathbb{E}}L(g,Z)\right|\\
&\leq \left|\hat{\mathbb{E}}L(g,Z) - \mathbb{E}L(g,Z)\right| + \mathbb{E}_{D_p}|L(g,Z) - L(f,Z)| + \mathbb{E}_{D_s}|L(f,Z) - L(g,Z)|\\
&= \left|\hat{\mathbb{E}}L(g,Z) - \mathbb{E}L(g,Z)\right| + 2\mathbb{E}_{D^\star}|L(g,Z) - L(f,Z)|\\
&\leq \left|\hat{\mathbb{E}}L(g,Z) - \mathbb{E}L(g,Z)\right| + 2r.
\end{aligned}$$

Applying the above relation in Line B.31 below, we have

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f\in\mathcal{F}}\left|\hat{\mathbb{E}}L(f,Z) - \mathbb{E}L(f,Z)\right| > B_{\text{ind}}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right)\\
&= \mathbb{P}\left(\exists f \in \mathcal{F}:\ \left|\hat{\mathbb{E}}L(f,Z) - \mathbb{E}L(f,Z)\right| > B_{\text{ind}}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right)\\
&\leq \mathbb{P}\left(\exists g \in \mathcal{G}_r:\ \left|\hat{\mathbb{E}}L(g,Z) - \mathbb{E}L(g,Z)\right| + 2r > B_{\text{ind}}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right) \qquad \text{(B.31)}\\
&= \mathbb{P}\left(\bigcup_{g\in\mathcal{G}_r}\left|\hat{\mathbb{E}}L(g,Z) - \mathbb{E}L(g,Z)\right| > B_{\text{ind}}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}}\right)\\
&\leq \sum_{g\in\mathcal{G}_r}\mathbb{P}\left(\left|\hat{\mathbb{E}}L(g,Z) - \mathbb{E}L(g,Z)\right| > B_{\text{ind}}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}}\right) \qquad \text{from the Union bound}\\
&\leq \mathcal{N}(\mathcal{F},r)\,2\exp\left[-\frac{2n}{(B_{\text{ind}})^2}\left\{B_{\text{ind}}\sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}}\right\}^2\right] \qquad \text{from Hoeffding's inequality} \qquad \text{(B.32)}\\
&= \delta. \qquad \text{(B.33)}
\end{aligned}$$

To apply Hoeffding's inequality (Theorem 2 of Hoeffding, 1963) in Line B.32, above, we use the fact that $L(g,Z)$ is bounded within an interval of length B_{ind}. ■

B.5.3. Proof of Eq B.23

Proof The proof for Eq B.23 is nearly identical to the proof for Eq B.22. Simply replacing L and B_{ind} respectively with $\tilde{L}$ and $(2B_{\text{ref}})$ in Eqs B.30-B.33 yields a valid proof for Eq B.23. ■

B.5.4. Proof of Eq B.24

Proof Let F_D denote the cumulative distribution function for a distribution D. Let $\tilde{D}_p$ be the distribution such that

$$F_{\tilde{D}_p}(Y = y, X_1 = x_1, X_2 = x_2) = F_{D_p}(Y = y, X_2 = x_2) F_{D_p}(X_1 = x_1).$$

Let $\tilde{D}_s$ be the distribution satisfying

$$\mathbb{P}_{\tilde{D}_s}(Y = y, X_1 = x_1, X_2 = x_2) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} \mathbf{1}(\mathbf{y}_{[j]} = y, \mathbf{X}_{1[i,\cdot]} = x_1, \mathbf{X}_{2[j,\cdot]} = x_2).$$

Let $\tilde{D}^\star$ be the uniform mixture of $\tilde{D}_p$ and $\tilde{D}_s$, as in Eq B.29. Replacing e_{orig}, $\hat{e}_{\text{orig}}$, D_p, D_s, and $D^\star$ respectively with e_{switch}, $\hat{e}_{\text{switch}}$, $\tilde{D}_p$, $\tilde{D}_s$, and $\tilde{D}^\star$, we can follow the same steps as in the proof for Eq B.22. For any $f \in \mathcal{F}$, we know that there exists a function $g \in \mathcal{G}_r$ satisfying $\mathbb{E}_{\tilde{D}^\star} |L(g, Z) - L(f, Z)| \leq r$, which implies

$$\begin{aligned}
|\hat{e}_{\text{switch}}(f) - e_{\text{switch}}(f)| &= |\hat{e}_{\text{switch}}(f) - e_{\text{switch}}(f) + \{\hat{e}_{\text{switch}}(g) - \hat{e}_{\text{switch}}(g)\} + \{e_{\text{switch}}(g) - e_{\text{switch}}(g)\}| \\
&\leq |\hat{e}_{\text{switch}}(g) - e_{\text{switch}}(g)| + \mathbb{E}_{\tilde{D}_p} |L(g, Z) - L(f, Z)| + \mathbb{E}_{\tilde{D}_s} |L(f, Z) - L(g, Z)| \\
&= |\hat{e}_{\text{switch}}(g) - e_{\text{switch}}(g)| + 2\mathbb{E}_{\tilde{D}^\star} |L(g, Z) - L(f, Z)| \\
&\leq |\hat{e}_{\text{switch}}(g) - e_{\text{switch}}(g)| + 2r.
\end{aligned}$$

As a result,

$$\begin{aligned}
&\mathbb{P}\left(\sup_{f \in \mathcal{F}} |\hat{e}_{\text{switch}}(f) - e_{\text{switch}}(f)| > B_{\text{ind}} \sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{n}} + 2r \right) \\
&\quad \leq \mathbb{P}\left(\exists g \in \mathcal{G}_r : \ |\hat{e}_{\text{switch}}(g) - e_{\text{switch}}(g)| + 2r > B_{\text{ind}} \sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{n}} + 2r \right) \\
&\quad \leq \sum_{g \in \mathcal{G}_r} \mathbb{P}\left(|\hat{e}_{\text{switch}}(g) - e_{\text{switch}}(g)| > B_{\text{ind}} \sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{n}} \right) \\
&\quad \leq \mathcal{N}(\mathcal{F}, r)\, 2 \exp\left[-\frac{n}{(B_{\text{ind}} - 0)^2} \left\{ B_{\text{ind}} \sqrt{\frac{\log(2\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{n}} \right\}^2 \right] \qquad \text{(B.34)} \\
&\quad = \delta.
\end{aligned}$$

In Line B.34, above, we apply Eq 5.7 of Hoeffding, 1963 (see also Eq 1 on page 201 of Serfling, 1980, in Theorem A), in the same way as in Eq B.15. ■

B.5.5. Proof for Eq B.25

Proof We apply Lemma 24 and Eq B.27 in Line B.36, below, to obtain

$$\begin{aligned}
&\mathbb{P}\left[\sup_{f \in \mathcal{F}} \left| \frac{\hat{e}_{\text{orig}}(f)}{\hat{e}_{\text{switch}}(f)} - \frac{e_{\text{orig}}(f)}{e_{\text{switch}}(f)} \right| > \mathcal{Q}_4 \right] \qquad \text{(B.35)} \\
&= \mathbb{P}\left(\exists f \in \mathcal{F} : \ \left| \frac{\hat{e}_{\text{orig}}(f)}{\hat{e}_{\text{switch}}(f)} - \frac{e_{\text{orig}}(f)}{e_{\text{switch}}(f)} \right| > \mathcal{Q}_4 \right) \\
&\leq \mathbb{P}\left(\left\{ \exists f \in \mathcal{F} : \ |\hat{e}_{\text{orig}}(f) - e_{\text{orig}}(f)| > B_{\text{ind}} \sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{2n}} + 2r \right\} \right. \qquad \text{(B.36)} \\
&\qquad \left. \bigcup \left\{ \exists f \in \mathcal{F} : \ |\hat{e}_{\text{switch}}(f) - e_{\text{switch}}(f)| > B_{\text{ind}} \sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r\sqrt{2}))}{n}} + 2r\sqrt{2} \right\} \right) \\
&= \mathbb{P}\left(\sup_{f \in \mathcal{F}} |\hat{e}_{\text{orig}}(f) - e_{\text{orig}}(f)| > B_{\text{ind}} \sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{2n}} + 2r \right) \\
&\qquad + \mathbb{P}\left(\sup_{f \in \mathcal{F}} |\hat{e}_{\text{switch}}(f) - e_{\text{switch}}(f)| > B_{\text{ind}} \sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r\sqrt{2}))}{n}} + 2r\sqrt{2} \right) \\
&\leq \frac{\delta}{2} + \frac{\delta}{2}. \qquad \text{from Eqs B.22 and B.24} \qquad \text{(B.37)}
\end{aligned}$$

■

B.5.6. Proof for Eq B.26

Proof Finally, to show B.26, we apply the same steps as in Eqs B.35 through B.37. We apply Eq B.28 & Lemma 24 to obtain

$$\begin{aligned}
&\mathbb{P}\left[\sup_{f \in \mathcal{F}} |\{\hat{e}_{\text{switch}}(f) - \hat{e}_{\text{orig}}(f)\} - \{e_{\text{switch}}(f) - e_{\text{orig}}(f)\}| > \mathcal{Q}_{4,\text{difference}} \right] \\
&\leq \mathbb{P}\left(\left\{ \exists f \in \mathcal{F} : \ |\hat{e}_{\text{orig}}(f) - e_{\text{orig}}(f)| > B_{\text{ind}} \sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r))}{2n}} + 2r \right\} \right. \\
&\qquad \left. \bigcup \left\{ \exists f \in \mathcal{F} : \ |\hat{e}_{\text{switch}}(f) - e_{\text{switch}}(f)| > B_{\text{ind}} \sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F}, r\sqrt{2}))}{n}} + 2r\sqrt{2} \right\} \right) \\
&\leq \frac{\delta}{2} + \frac{\delta}{2}.
\end{aligned}$$

■

B.5.7. Implementing Theorem 25 to Show Theorem 6

Proof Consider the event that

$$\exists \hat{f}_{+,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\max}\, \widehat{MR}(f) \text{ such that } MCR_+(\epsilon) < MR\left(\hat{f}_{+,\epsilon_{\text{in}}}\right). \tag{B.38}$$

A brief outline of our proof for Eq 4.6 is as follows. We expect Eq B.38 to be unlikely due to the fact that $\epsilon_{\text{in}} < \epsilon$. If Eq B.38 does not hold, then the only way that $MCR_+(\epsilon) < \widehat{MCR}_+(\epsilon_{\text{in}}) - \mathcal{Q}_{\text{in}}$ holds is if there exists $\hat{f}_{+,\epsilon_{\text{in}}} \in \arg\max_{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})} \widehat{MR}(f)$ which has an empirical MR that differs from its population-level MR by at least $\mathcal{Q}_{\text{in}}$.

To show that Eq B.38 is unlikely, we apply Theorem 25:

$$\begin{aligned}
&\mathbb{P}\left(\exists \hat{f}_{+,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\max}\, \widehat{MR}(f) : MCR_+(\epsilon) < MR\left(\hat{f}_{+,\epsilon_{\text{in}}}\right)\right) \\
&\leq \mathbb{P}\left(\exists f \in \hat{\mathcal{R}}(\epsilon_{\text{in}}) : MCR_+(\epsilon) < MR(f)\right) \\
&= \mathbb{P}\left(\exists f \in \hat{\mathcal{R}}(\epsilon_{\text{in}}) \backslash f_{\text{ref}} : MCR_+(\epsilon) < MR(f)\right) && \text{by } MCR_+(\epsilon) \geq MR(f_{\text{ref}}) \\
&\leq \mathbb{P}\left(\exists f \in \hat{\mathcal{R}}(\epsilon_{\text{in}}) \backslash f_{\text{ref}} : \mathbb{E}\tilde{L}(f,Z) > \epsilon\right) && \text{by } MCR_+(\epsilon) \text{ Def} \\
&= \mathbb{P}\left(\exists f \in \mathcal{F}, \mathbb{E}\tilde{L}(f,Z) > \epsilon : \hat{\mathbb{E}}\tilde{L}(f,Z) \leq \epsilon_{\text{in}}\right) && \text{by } \hat{\mathcal{R}}(\epsilon) \text{ Def} \\
&= \mathbb{P}\left(\exists f \in \mathcal{F}, \mathbb{E}\tilde{L}(f,Z) > \epsilon : \right. \\
&\qquad \left. \hat{\mathbb{E}}\tilde{L}(f,Z) - \epsilon \leq -2B_{\text{ref}}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} - 2r\right) && \text{by } \epsilon_{\text{in}} \text{ Def}
\end{aligned} \tag{B.39}$$

$$\begin{aligned}
&\leq \mathbb{P}\left(\exists f \in \mathcal{F}, \mathbb{E}\tilde{L}(f,Z) > \epsilon : \right. \\
&\qquad \left. \hat{\mathbb{E}}\tilde{L}(f,Z) - \mathbb{E}\tilde{L}(f,Z) \leq -2B_{\text{ref}}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} - 2r\right) \\
&\leq \mathbb{P}\left(\sup_{f \in \mathcal{F}} |\hat{\mathbb{E}}\tilde{L}(f,Z) - \mathbb{E}\tilde{L}(f,Z)| \geq 2B_{\text{ref}}\sqrt{\frac{\log(4\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right) \\
&= \frac{\delta}{2} && \text{by Thm 25.}
\end{aligned} \tag{B.40}$$

If Eq B.38 does not hold, we have

$$\begin{aligned}
MCR_+(\epsilon) &\geq MR\left(\hat{f}_{+,\epsilon_{\text{in}}}\right) && \text{for all } \hat{f}_{+,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\max}\, \widehat{MR}(f) \\
&= \widehat{MR}\left(\hat{f}_{+,\epsilon_{\text{in}}}\right) - \left\{\widehat{MR}\left(\hat{f}_{+,\epsilon_{\text{in}}}\right) - MR\left(\hat{f}_{+,\epsilon_{\text{in}}}\right)\right\} && \text{for all } \hat{f}_{+,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\max}\, \widehat{MR}(f) \\
&= \widehat{MCR}_+(\epsilon_{\text{in}}) - \left\{\widehat{MR}\left(\hat{f}_{+,\epsilon_{\text{in}}}\right) - MR\left(\hat{f}_{+,\epsilon_{\text{in}}}\right)\right\} && \text{for all } \hat{f}_{+,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\max}\, \widehat{MR}(f) \\
&\geq \widehat{MCR}_+(\epsilon_{\text{in}}) - \sup_{f \in \mathcal{F}} |\widehat{MR}(f) - MR(f)|.
\end{aligned} \tag{B.41}$$

Let q_{ratio} and $q_{\text{difference}}$ be defined as in Lemma 24. Then

$$\begin{aligned}
\mathcal{Q}_{\text{in}} &= \frac{B_{\text{switch}}}{b_{\text{orig}}} - \frac{B_{\text{switch}} - \left\{B_{\text{ind}}\sqrt{\frac{\log(8\delta^{-1}\mathcal{N}(\mathcal{F},r\sqrt{2}))}{n}} + 2r\sqrt{2}\right\}}{b_{\text{orig}} + \left\{B_{\text{ind}}\sqrt{\frac{\log(8\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right\}} \\
&= q_{\text{ratio}}\left(B_{\text{ind}}\sqrt{\frac{\log(8\delta^{-1}\mathcal{N}(\mathcal{F},r\sqrt{2}))}{n}} + 2r\sqrt{2}, B_{\text{ind}}\sqrt{\frac{\log(8\delta^{-1}\mathcal{N}(\mathcal{F},r))}{2n}} + 2r\right)
\end{aligned} \tag{B.42}$$

Theorem 25 implies that the sup term in Eq B.41 is less than $\mathcal{Q}_{\text{in}}$ with probability at least $1 - \frac{\delta}{2}$. Now, examining the left-hand side of Eq 4.6, we see

$$\begin{aligned}
&\mathbb{P}\left(MCR_+(\epsilon) < \widehat{MCR}_+(\epsilon_{\text{in}}) - \mathcal{Q}_{\text{in}}\right) \\
&\leq \mathbb{P}\left[\left\{\exists \hat{f}_{+,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\max}\, \widehat{MR}(f) : MCR_+(\epsilon) < MR\left(\hat{f}_{+,\epsilon_{\text{in}}}\right)\right\}\right. \\
&\qquad \left. \bigcup \left\{\sup_{f \in \mathcal{F}} |\widehat{MR}(f) - MR(f)| > \mathcal{Q}_{\text{in}}\right\}\right] && \text{from Eq B.41} \\
&\leq \mathbb{P}\left[\exists \hat{f}_{+,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\max}\, \widehat{MR}(f) : MCR_+(\epsilon) < MR\left(\hat{f}_{+,\epsilon_{\text{in}}}\right)\right] \\
&\qquad + \mathbb{P}\left[\sup_{f \in \mathcal{F}} |\widehat{MR}(f) - MR(f)| > \mathcal{Q}_{\text{in}}\right] && \text{from the Union bound} \\
&= \frac{\delta}{2} + \frac{\delta}{2} && \text{from Eq B.40, Eq B.42, \& Theorem 25.}
\end{aligned} \tag{B.43}$$

This completes the proof for Eq 4.6.

Alternatively, if we have defined model reliance as $MR(f) = e_{\text{switch}}(f) - e_{\text{orig}}(f)$ (see Appendix A.5), with $\widehat{MR}(f) = \hat{e}_{\text{switch}}(f) - \hat{e}_{\text{orig}}(f)$, and

$$\mathcal{Q}_{\text{in,difference}} = B_{\text{ind}} \left\{ \sqrt{\frac{\log(8\delta^{-1}\mathcal{N}\left(\mathcal{F}, r\sqrt{2}\right))}{n}} + \sqrt{\frac{\log(8\delta^{-1}\mathcal{N}\left(\mathcal{F}, r\right))}{2n}} \right\} + 2r(\sqrt{2}+1)$$
$$= q_{\text{difference}} \left(B_{\text{ind}} \sqrt{\frac{\log(8\delta^{-1}\mathcal{N}\left(\mathcal{F}, r\sqrt{2}\right))}{n}} + 2r\sqrt{2}, B_{\text{ind}} \sqrt{\frac{\log(8\delta^{-1}\mathcal{N}\left(\mathcal{F}, r\right))}{2n}} + 2r \right),$$

then same proof of Eq 4.6 holds without Assumption 3 if we replace $\mathcal{Q}_{\text{in}}$ with $\mathcal{Q}_{\text{in,difference}}$, and apply Eq B.26 in Eq B.43.

For Eq 4.7 we can use the same approach. Consider the event that

$$\exists \hat{f}_{-,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\min} \widehat{MR}(f) : MCR_{-}(\epsilon) > MR\left(\hat{f}_{-,\epsilon_{\text{in}}}\right). \tag{B.44}$$

Applying steps analogous to those used to derive Eq B.40, we have

$$\mathbb{P}\left(\exists \hat{f}_{-,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\min} \widehat{MR}(f) : MCR_{-}(\epsilon) > MR\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) \right)$$
$$\leq \mathbb{P}\left(\exists f \in \mathcal{F}, \mathbb{E}\tilde{L}(f,Z) > \epsilon : \hat{\mathbb{E}}\tilde{L}(f,Z) \leq \epsilon_{\text{in}} \right) \leq \frac{\delta}{2}.$$

Analogous to B.41, when Eq B.44 does not hold, we have have

$$\begin{aligned}
MCR_{-}(\epsilon) &\leq MR\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) && \text{for all } \hat{f}_{-,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\min} \widehat{MR}(f) \\
&= \widehat{MR}\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) + \left\{ MR\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) - \widehat{MR}\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) \right\} && \text{for all } \hat{f}_{-,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\min} \widehat{MR}(f) \\
&= \widehat{MCR}_{-}\left(\epsilon_{\text{in}}\right) + \left\{ MR\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) - \widehat{MR}\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) \right\} && \text{for all } \hat{f}_{-,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\min} \widehat{MR}(f) \\
&\leq \widehat{MCR}_{-}\left(\epsilon_{\text{in}}\right) + \sup_{f \in \mathcal{F}} |MR(f) - \widehat{MR}(f)|
\end{aligned}$$

Finally, analogous to Eq B.43,

$$\begin{aligned}
&\mathbb{P}\left(MCR_{-}(\epsilon) > \widehat{MCR}\left(\epsilon_{\text{in}}\right) + \mathcal{Q}_{\text{in}} \right) \\
&\leq \mathbb{P}\left[\left\{ \exists \hat{f}_{-,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\min} \widehat{MR}(f) : MCR_{-}(\epsilon) > MR\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) \right\} \right. \\
&\qquad \left. \bigcup \left\{ \sup_{f \in \mathcal{F}} |\widehat{MR}(f) - MR(f)| > \mathcal{Q}_{\text{in}} \right\} \right] \\
&\leq \mathbb{P}\left[\exists \hat{f}_{-,\epsilon_{\text{in}}} \in \underset{f \in \hat{\mathcal{R}}(\epsilon_{\text{in}})}{\arg\min} \widehat{MR}(f) : MCR_{-}(\epsilon) > MR\left(\hat{f}_{-,\epsilon_{\text{in}}}\right) \right] \\
&\qquad + \mathbb{P}\left[\sup_{f \in \mathcal{F}} |\widehat{MR}(f) - MR(f)| > \mathcal{Q}_{\text{in}} \right] \\
&= \frac{\delta}{2} + \frac{\delta}{2}.
\end{aligned} \tag{B.45}$$

Under the difference-based definition of model reliance (see Appendix A.5), the same proof for Eq 4.7 holds without Assumption 3 if we replace MR, $\widehat{MR}$, & $\mathcal{Q}_{\text{in}}$ respectively with $MR_{\text{difference}}$, $\widehat{MR}_{\text{difference}}$, & $\mathcal{Q}_{\text{in,difference}}$, and apply Eq B.26 in Eq B.45. ■

B.6. Proof of Proposition 7, and Corollary for a Unique Best-in-class Model.

We first introduce a lemma to describe the performance of any individual model in the population ϵ-Rashomon set.

Lemma 26 *Let $\epsilon_1' := 2B_{ref}\sqrt{\frac{\log(\delta^{-1})}{2n}}$, and let the functions $\hat{\phi}_{-}$ and $\hat{\phi}_{+}$ be defined as in Proposition 7. Given a function $f_1 \in \mathcal{R}(\epsilon)$, if Assumption 2 holds for f_1, then*

$$\mathbb{P}\left\{ \phi(f_1) \in \left[\hat{\phi}_{-}(\epsilon_1'), \hat{\phi}_{+}(\epsilon_1') \right] \right\} \geq 1 - \delta.$$

Proof Consider the event that

$$\phi(f_1) \in \left[\hat{\phi}_{-}(\epsilon_1'), \hat{\phi}_{+}(\epsilon_1') \right]. \tag{B.46}$$

Eq B.46 will always hold if $f_1 \in \hat{\mathcal{R}}(\epsilon_1')$, since the interval $\left[\hat{\phi}_{-}(\epsilon_1'), \hat{\phi}_{+}(\epsilon_1') \right]$ contains $\phi(f)$ for any $f \in \hat{\mathcal{R}}(\epsilon_1')$ by definition. Thus,

$$\begin{aligned}
\mathbb{P}\left\{ \phi(f_1) \notin [\left[\hat{\phi}_{-}(\epsilon_1'), \hat{\phi}_{+}(\epsilon_1') \right] \right\} &\leq \mathbb{P}\left\{ f_1 \notin \hat{\mathcal{R}}(\epsilon_1') \right\} \\
&\leq \delta && \text{from Lemma 23.}
\end{aligned}$$

■

B.6.1. Proof of Proposition 7

Proof Let $f_{-,\epsilon,\phi} \in \arg\min_{f\in\mathcal{R}(\epsilon)} \phi(f)$ and $f_{+,\epsilon,\phi} \in \arg\max_{f\in\mathcal{R}(\epsilon)} \phi(f)$ respectively denote functions that attain the lowest and highest values of $\phi(f)$ among models $f \in \mathcal{R}(\epsilon)$. Applying the definitions of $f_{-,\epsilon,\phi}$ and $f_{+,\epsilon,\phi}$ in Line B.47, below, we have

$$\begin{aligned}
&\mathbb{P}\Big(\{\phi(f) : f \in \mathcal{R}(\epsilon)\} \not\subset [\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')]\Big) \\
&\quad = \mathbb{P}\Big([\phi(f_{-,\epsilon,\phi}), \phi(f_{+,\epsilon,\phi})] \not\subset [\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')]\Big) \qquad \text{(B.47)}\\
&\quad = \mathbb{P}\Big(\phi(f_{-,\epsilon,\phi}) \notin [\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')] \quad \bigcup \quad \phi(f_{+,\epsilon,\phi}) \notin [\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')]\Big) \\
&\quad \le \mathbb{P}\Big(\phi(f_{-,\epsilon,\phi}) \notin [\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')]\Big) + \mathbb{P}\Big(\phi(f_{+,\epsilon,\phi}) \notin [\hat{\phi}_-(\epsilon'), \hat{\phi}_+(\epsilon')]\Big) \\
&\quad \le \frac{\delta}{2} + \frac{\delta}{2} = \delta \qquad \text{from Lemma 26, and the definition of } \epsilon' = \epsilon + 2B_{\text{ref}}\sqrt{\frac{\log(2\delta^{-1})}{2n}}.
\end{aligned}$$

■

B.6.2. Corollary for a Unique Best-in-Class Model

When the best-in-class model is unique, it can be described by the corollary below.

Corollary 27 *Let $\hat{\phi}_-(\epsilon_0') := \min_{f\in\hat{\mathcal{R}}(\epsilon_1')} \phi(f)$ and $\hat{\phi}_+(\epsilon_1') := \max_{f\in\hat{\mathcal{R}}(\epsilon_1')} \phi(f)$, where $\epsilon_0' := 2B_{ref}\sqrt{\frac{\log(\delta^{-1})}{2n}}$. Let $f^\star \in \arg\min_{f\in\mathcal{F}} e_{orig}(f)$ be the prediction model that uniquely attains the lowest possible expected loss. If $f^\star$ satisfies Assumption 2, then*

$$\mathbb{P}\{\phi(f^\star) \in [\hat{\phi}_-(\epsilon_1'), \hat{\phi}_+(\epsilon_1')]\} \ge 1 - \delta.$$

Proof Since $f^\star \in \mathcal{R}(0)$, Corollary 27 follows immediately from Lemma 26.

Notice that by assuming $f^\star$ is unique, we can use the threshold $\epsilon_0' := 2B_{\text{ref}}\sqrt{\frac{\log(\delta^{-1})}{2n}}$, which is lower than the threshold of $\epsilon' = \epsilon + 2B_{\text{ref}}\sqrt{\frac{\log(2\delta^{-1})}{2n}}$ with $\epsilon = 0$, as in Proposition 7. In this way, assuming uniqueness allows a stronger statement than the one in Proposition 7. ■

B.7. Absolute Losses versus Relative Losses in the Definition of the Rashomon Set

In this paper we primarily define Rashomon sets as the models that perform well *relative* to a reference model f_{ref}. We can also study an alternate formulation of Rashomon sets by replacing the relative loss $\tilde{L}$ with the non-standardized loss L throughout. This results in a new interpretation of the Rashomon set $\mathcal{R}(\epsilon_{\text{abs}}, f_{\text{ref}}, \mathcal{F}) = \{f_{\text{ref}}\} \cup \{f \in \mathcal{F} : \mathbb{E}L(f, Z) \le \epsilon_{\text{abs}}\}$ as the union of f_{ref} and the subset of models with *absolute* loss L no higher than ϵ_{abs}, for $\epsilon_{\text{abs}} > 0$. The process of computing empirical MCR is largely unaffected by whether L or $\tilde{L}$ is used, as it is simple to transform from one optimization problem to the other.

We still require the explicit inclusion of f_{ref} in empirical and population Rashomon sets to ensure that they are nonempty. However, in many cases, this inclusion becomes redundant when interpreting a Rashomon set (e.g., when $\epsilon \ge 0$, and $\mathbb{E}L(f_{\text{ref}}, Z) \le \epsilon_{\text{abs}}$).

Under the replacement of $\tilde{L}$ with L, we also replace Assumption 2 with Assumption 1 (whenever this is not redundant), and replace $2B_{\text{ref}}$ with B_{ind} in the definitions of ϵ_{out}, ϵ_{best}, ϵ_{in}, ϵ' and ϵ_1' in Theorem 4, Corollary 22, Theorem 6, Proposition 7, and Corollary 27. This is because the motivation for the $2B_{\text{ref}}$ term is that $\tilde{L}(f_1)$ is bounded within an interval of length $2B_{\text{ref}}$ when f_1 satisfies Assumption 2. However, under Assumption 1, $L(f_1)$ is bounded within an interval of length B_{ind}.

B.8. Proof of Proposition 15

Proof To show Eq 7.1 we start with $e_{\text{orig}}(f_\beta)$,

$$\begin{aligned}
e_{\text{orig}}(f_\beta) &= \mathbb{E}[\{Y - X_1'\beta_1 - X_2'\beta_2\}^2] \\
&= \mathbb{E}[\{(Y - X_2'\beta_2) - X_1'\beta_1\}^2] \\
&= \mathbb{E}[(Y - X_2'\beta_2)^2] - 2\mathbb{E}[(Y - X_2'\beta_2)X_1']\beta_1 + \beta_1'\mathbb{E}[X_1X_1']\beta_1.
\end{aligned}$$

For $e_{\text{switch}}(f_\beta)$, we can follow the same steps as above:

$$\begin{aligned}
e_{\text{switch}}(f_\beta) &= \mathbb{E}_{Y^{(b)}, X_1^{(a)}, X_2^{(b)}}[\{Y^{(b)} - X_1^{(a)\prime}\beta_1 - X_2^{(b)\prime}\beta_2\}^2] \\
&= \mathbb{E}[(Y^{(b)} - X_2^{(b)\prime}\beta_2)^2] - 2\mathbb{E}[Y^{(b)} - X_2^{(b)\prime}\beta_2]\mathbb{E}[X_1^{(a)\prime}]\beta_1 + \beta_1'\mathbb{E}[X_1^{(a)}X_1^{(a)\prime}]\beta_1.
\end{aligned}$$

Since $(Y^{(b)}, X_1^{(b)}, X_2^{(b)})$ and $(Y^{(a)}, X_1^{(a)}, X_2^{(a)})$ each have the same distribution as (Y, X_1, X_2), we can omit the superscript notation to show Eq 7.1:

$$\begin{aligned}
e_{\text{switch}}(f_\beta) &= \mathbb{E}[(Y - X_2'\beta_2)^2] - 2\mathbb{E}[Y - X_2'\beta_2]\mathbb{E}[X_1']\beta_1 + \beta_1'\mathbb{E}[X_1X_1']\beta_1 \\
e_{\text{switch}}(f_\beta) &= e_{\text{orig}}(f_\beta) - 2\mathbb{E}[Y - X_2'\beta_2]\mathbb{E}[X_1']\beta_1 + 2\mathbb{E}[(Y - X_2'\beta_2)X_1']\beta_1 \\
e_{\text{switch}}(f_\beta) &= e_{\text{orig}}(f_\beta) + 2\text{Cov}(Y - X_2'\beta_2, X_1)\beta_1 \\
e_{\text{switch}}(f_\beta) &= e_{\text{orig}}(f_\beta) + 2\text{Cov}(Y, X_1)\beta_1 - 2\beta_2\text{Cov}(X_2, X_1)\beta_1.
\end{aligned}$$

Dividing both sides by $e_{\text{orig}}(f_\beta)$ gives the desired result.

Next, we can use a similar approach to show Eq 7.2:

$$\begin{aligned}
\hat{e}_{\text{switch}}(f_\beta) &= \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}(\mathbf{y}_{[j]} - \mathbf{X}_{2[j,\cdot]}\beta_2 - \mathbf{X}_{1[i,\cdot]}\beta_1)^2 \\
n(n-1)\hat{e}_{\text{switch}}(f_\beta) &= \sum_{i=1}^{n}\sum_{j\neq i}\{(\mathbf{y}_{[j]} - \mathbf{X}_{2[j,\cdot]}\beta_2)^2 - 2(\mathbf{y}_{[j]} - \mathbf{X}_{2[j,\cdot]}\beta_2)(\mathbf{X}_{1[i,\cdot]}\beta_1) + (\mathbf{X}_{1[i,\cdot]}\beta_1)^2\} \\
&= (n-1)\sum_{i=1}^{n}(\mathbf{y}_{[i]} - \mathbf{X}_{2[i,\cdot]}\beta_2)^2 \\
&\quad - 2\left\{\sum_{i=1}^{n}\sum_{j\neq i}(\mathbf{X}_{1[i,\cdot]}\beta_1)(\mathbf{y}_{[j]} - \mathbf{X}_{2[j,\cdot]}\beta_2)\right\} + (n-1)\sum_{i=1}^{n}(\mathbf{X}_{1[i,\cdot]}\beta_1)^2.
\end{aligned} \tag{B.48}$$

Focusing on the term in braces,

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j\neq i}(\mathbf{X}_{1[i,\cdot]}\beta_1)(\mathbf{y}_{[j]}-\mathbf{X}_{2[j,\cdot]}\beta_2)\\
&\quad=\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{X}_{1[i,\cdot]}\beta_1)(\mathbf{y}_{[j]}-\mathbf{X}_{2[j,\cdot]}\beta_2)-\sum_{i=1}^{n}(\mathbf{X}_{1[i,\cdot]}\beta_1)(\mathbf{y}_{[i]}-\mathbf{X}_{2[i,\cdot]}\beta_2)\\
&\quad=\sum_{i=1}^{n}(\mathbf{X}_{1[i,\cdot]}\beta_1)\sum_{j=1}^{n}(\mathbf{y}_{[j]}-\mathbf{X}_{2[j,\cdot]}\beta_2)-\sum_{i=1}^{n}(\mathbf{X}_{1[i,\cdot]}\beta_1)(\mathbf{y}_{[i]}-\mathbf{X}_{2[i,\cdot]}\beta_2)\\
&\quad=\left\{(\mathbf{X}_1\beta_1)'\mathbf{1}_n\right\}\left\{\mathbf{1}_n'(\mathbf{y}-\mathbf{X}_2\beta_2)\right\}-(\mathbf{X}_1\beta_1)'(\mathbf{y}-\mathbf{X}_2\beta_2) \qquad \text{(B.49)}\\
&\quad=(\mathbf{X}_1\beta_1)'(\mathbf{1}_n\mathbf{1}_n'-\mathbf{I}_n)(\mathbf{y}-\mathbf{X}_2\beta_2).
\end{aligned}
$$

Plugging this into Eq B.48, and applying the sample linear algebra representations as in Eq B.49, we get

$$
\begin{aligned}
n(n-1)\hat{e}_{\text{switch}}(f_\beta) &= (n-1)\|\mathbf{y}-\mathbf{X}_2\beta_2\|_2^2\\
&\quad-2(\mathbf{X}_1\beta_1)'(\mathbf{1}_n\mathbf{1}_n'-\mathbf{I}_n)(\mathbf{y}-\mathbf{X}_2\beta_2)\\
&\quad+(n-1)\|\mathbf{X}_1\beta_1\|_2^2\\
n\hat{e}_{\text{switch}}(f_\beta) &= \|\mathbf{y}-\mathbf{X}_2\beta_2\|_2^2\\
&\quad-2(\mathbf{X}_1\beta_1)'\mathbf{W}(\mathbf{y}-\mathbf{X}_2\beta_2)\\
&\quad+\|\mathbf{X}_1\beta_1\|_2^2\\
&=\mathbf{y}'\mathbf{y}-2\mathbf{y}'\mathbf{X}_2\beta_2+\beta_2'\mathbf{X}_2'\mathbf{X}_2\beta_2\\
&\quad-2\beta_1'\mathbf{X}_1'\mathbf{W}\mathbf{y}+2\beta_1'\mathbf{X}_1'\mathbf{W}\mathbf{X}_2\beta_2\\
&\quad+\beta_1'\mathbf{X}_1'\mathbf{X}_1\beta_1\\
&=\mathbf{y}'\mathbf{y}-2\begin{bmatrix}\mathbf{X}_1'\mathbf{W}\mathbf{y}\\ \mathbf{X}_2'\mathbf{y}\end{bmatrix}'\beta+\beta'\begin{bmatrix}\mathbf{X}_1'\mathbf{X}_1 & \mathbf{X}_1'\mathbf{W}\mathbf{X}_2\\ \mathbf{X}_2'\mathbf{W}\mathbf{X}_1 & \mathbf{X}_2'\mathbf{X}_2\end{bmatrix}\beta.
\end{aligned}
$$

■

B.9. Proof of Proposition 19

Proof First we consider $e_{\text{orig}}(f_0)$. We briefly recall that the notation $f_0(t,c)$ refers to the *true* conditional expectation function for *both* potential outcomes Y_1, Y_0, rather than the expectation for Y_0 alone.

Under the assumption that $(Y_1, Y_0) \perp T|C$, we have $f_0(t,c) = \mathbb{E}(Y|C=c, T=t) = \mathbb{E}(Y_t|C=c)$. Applying this, we see that

$$
\begin{aligned}
e_{\text{orig}}(f_0) &= \mathbb{E}L(f_0,(Y,T,C))\\
&= \mathbb{E}L(f_0,(Y_T,T,C))\\
&= \mathbb{E}_T\mathbb{E}_{C|T}\mathbb{E}_{Y_T|C}[\{Y_T-\mathbb{E}(Y_T|C)\}^2]\\
&= \mathbb{E}_T\mathbb{E}_{C|T}\text{Var}(Y_T|C)\\
&= q\mathbb{E}_{C|T=0}\text{Var}(Y_0|C)+p\mathbb{E}_{C|T=1}\text{Var}(Y_1|C), \qquad \text{(B.50)}
\end{aligned}
$$

where $p := \mathbb{P}(T=1)$ and $q := \mathbb{P}(T=0)$.

Now we consider $e_{\text{switch}}(f_0)$. Let $(Y_0^{(a)}, Y_1^{(a)}, T^{(a)}, C^{(a)})$ and $(Y_0^{(b)}, Y_1^{(b)}, T^{(b)}, C^{(b)})$ be a pair of independent random variable vectors, each with the same distribution as (Y_0, Y_1, T, C). Then

$$
\begin{aligned}
e_{\text{switch}}(f_0) &= \mathbb{E}_{T^{(b)},T^{(a)},C^{(b)},Y^{(b)}_{T^{(b)}}}[\{Y^{(b)}_{T^{(b)}}-f_0(T^{(a)},C^{(b)})\}^2]\\
&= \mathbb{E}_{T^{(b)},T^{(a)},C^{(b)},Y^{(b)}_{T^{(b)}}}[\{Y^{(b)}_{T^{(b)}}-\mathbb{E}(Y_{T^{(a)}}|C=C^{(b)})\}^2]\\
&= \mathbb{E}_{T^{(b)},T^{(a)}}\mathbb{E}_{C^{(b)}|T^{(b)}}\mathbb{E}_{Y^{(b)}_{T^{(b)}}|C^{(b)}}[\{Y^{(b)}_{T^{(b)}}-\mathbb{E}(Y_{T^{(a)}}|C=C^{(b)})\}^2].
\end{aligned}
$$

First we expand the outermost expectation, over $T^{(b)}, T^{(a)}$:

$$
\begin{aligned}
&e_{\text{switch}}(f_0)\\
&\quad=\sum_{i,j\in\{0,1\}}\mathbb{P}(T^{(b)}=i,T^{(a)}=j)\mathbb{E}_{C^{(b)}|T^{(b)}=i}\mathbb{E}_{Y^{(b)}_i|C^{(b)}}[\{Y^{(b)}_i-\mathbb{E}(Y_j|C=C^{(b)})\}^2]. \qquad \text{(B.51)}
\end{aligned}
$$

Since $T^{(b)} \perp T^{(a)}$, we can write

$$
\begin{aligned}
\mathbb{P}(T^{(b)}=i,T^{(a)}=j) &= \mathbb{P}(T^{(b)}=i)\mathbb{P}(T^{(a)}=j)\\
&= p^{i+j}q^{2-i-j}.
\end{aligned}
$$

Plugging this into Eq B.51 we get

$$
e_{\text{switch}}(f_0)=\sum_{i,j\in\{0,1\}}p^{i+j}q^{2-i-j}\mathbb{E}_{C^{(b)}|T^{(b)}=i}\mathbb{E}_{Y^{(b)}_i|C^{(b)}}[\{Y^{(b)}_i-\mathbb{E}(Y_j|C=C^{(b)})\}^2].
$$

Since $(Y_0^{(b)}, Y_1^{(b)}, C^{(b)}, T^{(b)})$ are the only random variables remaining, we can omit the superscript notation (e.g., replace $C^{(b)}$ with C) to get

$$
\begin{aligned}
e_{\text{switch}}(f_0) &= \sum_{i,j\in\{0,1\}}p^{i+j}q^{2-i-j}\mathbb{E}_{C|T=i}\mathbb{E}_{Y_i|C}[\{Y_i-\mathbb{E}(Y_j|C)\}^2]\\
&=: \sum_{i,j\in\{0,1\}}A_{ij},
\end{aligned}
$$

where $A_{ij} = p^{i+j}q^{2-i-j}\mathbb{E}_{C|T=i}\mathbb{E}_{Y_i|C}[\{Y_i - \mathbb{E}(Y_j|C)\}^2]$. First, we consider A_{00} and A_{11}:

$$\begin{aligned} A_{00} &= q^2\mathbb{E}_{C|T=0}\mathbb{E}_{Y_0|C}[\{Y_0 - \mathbb{E}(Y_0|C)\}^2] \\ &= q^2\mathbb{E}_{C|T=0}\text{Var}(Y_0|C), \end{aligned}$$

and, similarly, $A_{11} = p^2\mathbb{E}_{C|T=1}\text{Var}(Y_1|C)$.

Next we consider A_{01} and A_{10}:

$$\begin{aligned} A_{01}: &= pq\mathbb{E}_{C|T=0}\mathbb{E}_{Y_0|C}[\{Y_0 - \mathbb{E}(Y_1|C)\}^2] \\ &= pq\mathbb{E}_{C|T=0}\left(\mathbb{E}(Y_0^2|C) - 2\mathbb{E}(Y_0|C)\mathbb{E}(Y_1|C) + \mathbb{E}(Y_1|C)^2\right) \\ &= pq\mathbb{E}_{C|T=0}\left(\text{Var}(Y_0|C) + \mathbb{E}(Y_0|C)^2 - 2\mathbb{E}(Y_0|C)\mathbb{E}(Y_1|C) + \mathbb{E}(Y_1|C)^2\right) \\ &= pq\mathbb{E}_{C|T=0}\left(\text{Var}(Y_0|C) + [\mathbb{E}(Y_1|C) - \mathbb{E}(Y_0|C)]^2\right) \\ &= pq\mathbb{E}_{C|T=0}\left(\text{Var}(Y_0|C) + \text{CATE}(C)^2\right), \end{aligned}$$

and, following the same steps,

$$A_{10} = pq\mathbb{E}_{C|T=1}\left(\text{Var}(Y_1|C) + \text{CATE}(C)^2\right).$$

Plugging in A_{00}, A_{01}, A_{10}, and A_{11} we get

$$\begin{aligned} e_{\text{switch}}(f_0) &= \{A_{00} + A_{11}\} \\ &\quad + [A_{01} + A_{10}] \\ &= \{q^2\mathbb{E}_{C|T=0}\text{Var}(Y_0|C) + p^2\mathbb{E}_{C|T=1}\text{Var}(Y_1|C)\} \\ &\quad + \left[pq\mathbb{E}_{C|T=0}\left(\text{Var}(Y_0|C) + \text{CATE}(C)^2\right) + pq\mathbb{E}_{C|T=1}\left(\text{Var}(Y_1|C) + \text{CATE}(C)^2\right)\right] \\ &= \{q(q+p)\mathbb{E}_{C|T=0}\text{Var}(Y_0|C) + p(p+q)\mathbb{E}_{C|T=1}\text{Var}(Y_1|C)\} && \text{(B.52)} \\ &\quad + pq\left[\mathbb{E}_{C|T=0}\left(\text{CATE}(C)^2\right) + \mathbb{E}_{C|T=1}\left(\text{CATE}(C)^2\right)\right] && \text{(B.53)} \\ &= \{e_{\text{orig}}(f_0)\} && \text{(B.54)} \\ &\quad + \text{Var}(T)\left[\mathbb{E}_{C|T=0}\left(\text{CATE}(C)^2\right) + \mathbb{E}_{C|T=1}\left(\text{CATE}(C)^2\right)\right]. && \text{(B.55)} \end{aligned}$$

In Lines B.52 and B.53, we consolidate terms involving $\mathbb{E}_{C|T=0}\text{Var}(Y_0|C)$ and $\mathbb{E}_{C|T=1}\text{Var}(Y_1|C)$. In Line B.54, we use $p+q=1$ to reduce Line B.52 to the right-hand side of Eq B.50. Finally, in Line B.55, we use $qp = \text{Var}(T)$. Dividing both sides by $e_{\text{orig}}(f_0) = \mathbb{E}_{T,C}Var(Y|T,C)$ gives the desired result. ■

Appendix C. Proofs for Computational Results

Almost all of the proofs in this section are unchanged if we replace $\hat{e}_{\text{switch}}(f)$ with $\hat{e}_{\text{divide}}(f)$ in our definitions of $\hat{h}_{-,\gamma}$, $\hat{h}_{+,\gamma}$, $\hat{g}_{-,\gamma}$, $\hat{g}_{+,\gamma}$, and $\widehat{MR}$. The only exception is in Appendix C.3.

Throughout the following proofs, we will make use of the fact that, for constants $a, b, c, d \in \mathbb{R}$ satisfying $a \geq c$, the relation $a + b \leq c + d$ implies

$$\begin{aligned} a + b &\leq c + d \\ a - c &\leq d - b \\ 0 &\leq d - b && \text{since } 0 \leq a - c \\ b &\leq d. && \text{(C.1)} \end{aligned}$$

We also make use of the fact that for any $\gamma_1, \gamma_2 \in \mathbb{R}$, the definitions of $\hat{g}_{+,\gamma_1}$ and $\hat{g}_{-,\gamma_1}$ imply

$$\hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_1}) \leq \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_2}), \quad \text{and} \quad \hat{h}_{-,\gamma_1}(g_{-,\gamma_1}) \leq \hat{h}_{-,\gamma_1}(g_{-,\gamma_2}). \tag{C.2}$$

Finally, for any two values $\gamma_1, \gamma_2 \in \mathbb{R}$, we make use of the fact that

$$\begin{aligned} \hat{h}_{+,\gamma_1}(f) &= \hat{e}_{\text{orig}}(f) + \gamma_1\hat{e}_{\text{switch}}(f) \\ &= \hat{e}_{\text{orig}}(f) + \gamma_2\hat{e}_{\text{switch}}(f) + \{\gamma_1\hat{e}_{\text{switch}}(f) - \gamma_2\hat{e}_{\text{switch}}(f)\} \\ &= \hat{h}_{+,\gamma_2}(f) + (\gamma_1 - \gamma_2)\hat{e}_{\text{switch}}(f), \end{aligned} \tag{C.3}$$

and, by the same steps,

$$\hat{h}_{-,\gamma_1}(f) = \hat{h}_{-,\gamma_2}(f) + (\gamma_1 - \gamma_2)\hat{e}_{\text{orig}}(f). \tag{C.4}$$

C.1. Proof of Lemma 9 (Lower Bound for MR)

Proof We prove Lemma 9 in 2 parts.

C.1.1. Part 1: Showing Eq 6.1 Holds for All $f \in \mathcal{F}$ Satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$.

If $\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma}) \geq 0$, then for any function $f \in \mathcal{F}$ satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$ we know that

$$\begin{aligned} \frac{1}{\epsilon_{\text{abs}}} &\leq \frac{1}{\hat{e}_{\text{orig}}(f)} \\ \frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{\text{abs}}} &\leq \frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\hat{e}_{\text{orig}}(f)}. \end{aligned} \tag{C.5}$$

Now, for any $f \in \mathcal{F}$ satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$, the definition of $\hat{g}_{-,\gamma}$ implies that

$$\begin{aligned} \hat{h}_{-,\gamma}(f) &\geq \hat{h}_{-,\gamma}(\hat{g}_{-,\gamma}) \\ \gamma\hat{e}_{\text{orig}}(f) + \hat{e}_{\text{switch}}(f) &\geq \hat{h}_{-,\gamma}(\hat{g}_{-,\gamma}) \\ \gamma + \frac{\hat{e}_{\text{switch}}(f)}{\hat{e}_{\text{orig}}(f)} &\geq \frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\hat{e}_{\text{orig}}(f)} \\ \gamma + \frac{\hat{e}_{\text{switch}}(f)}{\hat{e}_{\text{orig}}(f)} &\geq \frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{\text{abs}}} && \text{from Eq C.5} \\ \widehat{MR}(f) &\geq \frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{\text{abs}}} - \gamma. \end{aligned}$$

C.1.2. Part 2: Showing that, if $f = \hat{g}_{-,\gamma}$, and at Least One of the Inequalities in Condition 8 Holds with Equality, then Eq 6.1 Holds with Equality.

We consider each of the two inequalities in Condition 8 separately. If $\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma}) = 0$, then

$$0 = \gamma \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma}) + \hat{e}_{\text{switch}}(\hat{g}_{-,\gamma})$$
$$\frac{-\hat{e}_{\text{switch}}(\hat{g}_{-,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma})} = \gamma.$$

As a result

$$\frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{\text{abs}}} - \gamma = \frac{0}{\epsilon_{\text{abs}}} - \left\{ \frac{-\hat{e}_{\text{switch}}(\hat{g}_{-,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma})} \right\} = \widehat{MR}(\hat{g}_{-,\gamma}).$$

Alternatively, if $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma}) = \epsilon_{\text{abs}}$, then

$$\frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{\text{abs}}} - \gamma = \frac{\gamma \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma}) + \hat{e}_{\text{switch}}(\hat{g}_{-,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma})} - \gamma = \gamma + \frac{\hat{e}_{\text{switch}}(\hat{g}_{-,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma})} - \gamma = \widehat{MR}(\hat{g}_{-,\gamma}).$$

■

C.2. Proof of Lemma 10 (Monotonicity for MR Lower Bound Binary Search)

Proof We prove Lemma 10 in 3 parts.

C.2.1. Part 1: $\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})$ is Monotonically Increasing in γ.

Let $\gamma_1, \gamma_2 \in \mathbb{R}$ satisfy $\gamma_1 < \gamma_2$. We have assumed that $0 < \hat{e}_{\text{orig}}(f)$ for any $f \in \mathcal{F}$. Thus, for any $f \in \mathcal{F}$ we have

$$\gamma_1 \hat{e}_{\text{orig}}(f) + \hat{e}_{\text{switch}}(f) < \gamma_2 \hat{e}_{\text{orig}}(f) + \hat{e}_{\text{switch}}(f)$$
$$\hat{h}_{-,\gamma_1}(f) < \hat{h}_{-,\gamma_2}(f). \tag{C.6}$$

Applying this, we have

$$\begin{aligned} \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) &\leq \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_2}) && \text{from Eq C.2} \\ &\leq \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) && \text{from Eq C.6.} \end{aligned}$$

This result is analogous to Lemma 3 from Dinkelbach (1967).

C.2.2. Part 2: $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma})$ is Monotonically Decreasing in γ.

Let $\gamma_1, \gamma_2 \in \mathbb{R}$ satisfy $\gamma_1 < \gamma_2$. Then

$$\begin{aligned} \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) &\leq \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_2}) && \text{from Eq C.2} \\ \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_1}) + (\gamma_1 - \gamma_2)\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}) &\leq \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) + (\gamma_1 - \gamma_2)\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}) && \text{from Eq C.4} \\ (\gamma_1 - \gamma_2)\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}) &\leq (\gamma_1 - \gamma_2)\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}) && \text{from Eqs C.1 \& C.2} \\ \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}) &\geq \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}). \end{aligned}$$

C.2.3. Part 3: $\left\{ \frac{\hat{h}_{-,\gamma}(\hat{g}_{-,\gamma})}{\epsilon_{\text{abs}}} - \gamma \right\}$ is Monotonically Decreasing in γ in the Range Where $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma}) \leq \epsilon_{\text{abs}}$, and Increasing Otherwise.

Suppose $\gamma_1 < \gamma_2$ and $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}), \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}) \leq \epsilon_{\text{abs}}$. Then, from Eq C.2,

$$\begin{aligned} \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) &\leq \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_1}) \\ \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) &\leq \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) + (\gamma_2 - \gamma_1)\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}) && \text{from Eq C.4} \\ \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) &\leq \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) + (\gamma_2 - \gamma_1)\epsilon_{\text{abs}} && \text{from } \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}) \leq \epsilon_{\text{abs}} \\ \frac{\hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2})}{\epsilon_{\text{abs}}} - \gamma_2 &\leq \frac{\hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1})}{\epsilon_{\text{abs}}} - \gamma_1. \end{aligned}$$

Similarly, if $\gamma_1 < \gamma_2$ and $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}), \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}) \geq \epsilon_{\text{abs}}$. Then, from Eq C.2

$$\begin{aligned} \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) &\leq \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_2}) \\ \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) &\leq \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) + (\gamma_1 - \gamma_2)\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}) && \text{from Eq C.4} \\ \hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1}) &\leq \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) + (\gamma_1 - \gamma_2)\epsilon_{\text{abs}} && \text{from } \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}) \geq \epsilon_{\text{abs}} \\ \frac{\hat{h}_{-,\gamma_1}(\hat{g}_{-,\gamma_1})}{\epsilon_{\text{abs}}} - \gamma_1 &\leq \frac{\hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2})}{\epsilon_{\text{abs}}} - \gamma_2. \end{aligned}$$

■

C.3. Proof of Proposition 11 (Nonnegative Weights for MR Lower Bound Binary Search)

Proof Let $\gamma_1 := \frac{1}{n-1}$. First we show that there exists a function $\hat{g}_{-,\gamma_1}$ minimizing $\hat{h}_{-,\gamma_1}$ such that $\widehat{MR}(\hat{g}_{-,\gamma_1}) = 1$. Let D_s denote the sample distribution of the data, and let D_m be the distribution satisfying

$$\begin{aligned} \mathbb{P}_{D_m}\{(Y, X_1, X_2) = (y, x_1, x_2)\} &= \mathbb{P}_{D_s}\{(Y, X_2) = (y, x_2)\} \times \mathbb{P}_{D_s}(X_1 = x_1) \\ &= \frac{1}{n^2} \sum_{i=1}^{n} 1(\mathbf{y}_{[i]} = y \text{ and } \mathbf{X}_{2[i]} = x_2) \sum_{j=1}^{n} 1(\mathbf{X}_{1[j]} = x_1). \end{aligned}$$

From $\gamma_1 = \frac{1}{n-1}$ and Eq 6.2, we see that

$$\begin{aligned} \hat{h}_{-,\gamma_1}(f) &= \sum_{i=1}^{n} \sum_{j=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[j]}, \mathbf{X}_{2[i]})\} \times \left\{ \frac{\gamma_1 1(i = j)}{n} + \frac{1(i \neq j)}{n(n-1)} \right\} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} L\{f, (\mathbf{y}_{[i]}, \mathbf{X}_{1[j]}, \mathbf{X}_{2[i]})\} \times \left\{ \frac{1}{n(n-1)} \right\}. \\ &\propto \mathbb{E}_{D_m} L\{f, (Y, X_1, X_2)\}. \end{aligned}$$

Thus, from Condition 2 of Proposition 11, we know there exists a function $\hat{g}_{-,\gamma_1}$ that minimizes $\hat{h}_{-,\gamma_1}$ with $\hat{g}_{-,\gamma_1}(x_1^{(a)}, x_2) = \hat{g}_{-,\gamma_1}(x_1^{(b)}, x_2)$ for any $x_1^{(a)}, x_1^{(b)} \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$. Condition 1 of Proposition 11 then implies that $L\{\hat{g}_{-,\gamma_1}, (y, x_1^{(a)}, x_2)\} = L\{\hat{g}_{-,\gamma_1}, (y, x_1^{(b)}, x_2)\}$

for any $x_1^{(a)}, x_1^{(b)} \in \mathcal{X}_1$, $x_2 \in \mathcal{X}_2$, and $y \in \mathcal{Y}$. We apply this result in Line C.7, below, to show that loss of model $\hat{g}_{-,\gamma_1}$ is unaffected by permuting X_1 within our sample:

$$\begin{aligned}
\hat{e}_{\text{switch}}(\hat{g}_{-,\gamma_1}) &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n-1}\sum_{j\neq i} L\{\hat{g}_{-,\gamma_1}, (\mathbf{y}_{[i]}, \mathbf{X}_{1[j]}, \mathbf{X}_{2[i]})\} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n-1}\sum_{j\neq i} L\{\hat{g}_{-,\gamma_1}, (\mathbf{y}_{[i]}, \mathbf{X}_{1[i]}, \mathbf{X}_{2[i]})\} \qquad \text{(C.7)} \\
&= \frac{1}{n}\sum_{i=1}^{n} L\{\hat{g}_{-,\gamma_1}, (\mathbf{y}_{[i]}, \mathbf{X}_{1[i]}, \mathbf{X}_{2[i]})\} \\
&= \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}).
\end{aligned}$$

It follows that $\widehat{MR}(\hat{g}_{-,\gamma_1}) = 1$. To show the result of Proposition 11, let $\gamma_2 = 0$. For any function $\hat{g}_{-,\gamma_2}$ minimizing $\hat{h}_{-,\gamma_2}$, we know that

$$\begin{aligned}
\hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) &\leq \hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_1}) && \text{from the definition of } \hat{g}_{-,\gamma_2} \\
0 + \hat{e}_{\text{switch}}(\hat{g}_{-,\gamma_2}) &\leq 0 + \hat{e}_{\text{switch}}(\hat{g}_{-,\gamma_1}) && \text{from } \gamma_2 = 0 \text{ and the definition of } \hat{h}_{-,\gamma_2}. \qquad \text{(C.8)}
\end{aligned}$$

From $\gamma_2 \leq \gamma_1$, and Part 2 of Lemma 10, we know that

$$\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}) \geq \hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1}). \qquad \text{(C.9)}$$

Combining Eqs C.8 and C.9, we have

$$\widehat{MR}(\hat{g}_{-,\gamma_2}) = \frac{\hat{e}_{\text{switch}}(\hat{g}_{-,\gamma_2})}{\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2})} \leq \frac{\hat{e}_{\text{switch}}(\hat{g}_{-,\gamma_1})}{\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_1})} = \widehat{MR}(\hat{g}_{-,\gamma_1}) = 1. \qquad \text{(C.10)}$$

Since $\hat{h}_{-,\gamma_2}(\hat{g}_{-,\gamma_2}) = \hat{e}_{\text{switch}}(\hat{g}_{-,\gamma_2}) \geq 0$ by definition, Condition 8 holds for γ_2, ϵ_{abs} and $\hat{g}_{-,\gamma_2}$ if and only if $\hat{e}_{\text{orig}}(\hat{g}_{-,\gamma_2}) \leq \epsilon_{\text{abs}}$. This, combined with Eq C.10, completes the proof.

The same result does not necessarily hold if we replace $\hat{e}_{\text{switch}}$ with $\hat{e}_{\text{divide}}$ in our definitions of $\hat{h}_{-,\gamma}$, $\widehat{MR}$, and $\widehat{MCR}_-$. This is because $\hat{e}_{\text{divide}}$ does not correspond to the expectation over a distribution in which X_1 is independent of X_2 and Y, due to the fixed pairing structure used in $\hat{e}_{\text{divide}}$. Thus, Condition 2 of Proposition 11 will not apply. ■

C.4. Proof of Lemma 13 (Upper Bound for MR)

Proof We prove Lemma 13 in 2 parts.

C.4.1. Part 1: Showing Eq 6.4 Holds for All $f \in \mathcal{F}$ Satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$.

If $\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma}) \geq 0$, then for any function $f \in \mathcal{F}$ satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$ we know that

$$\begin{aligned}
\frac{1}{\epsilon_{\text{abs}}} &\leq \frac{1}{\hat{e}_{\text{orig}}(f)} \\
\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} &\leq \frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\hat{e}_{\text{orig}}(f)}. \qquad \text{(C.11)}
\end{aligned}$$

Now, if $\gamma \leq 0$, then for any $f \in \mathcal{F}$ satisfying $\hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$, the definition of $\hat{g}_{+,\gamma}$ implies

$$\begin{aligned}
\hat{h}_{+,\gamma}(f) &\geq \hat{h}_{+,\gamma}(\hat{g}_{+,\gamma}) \\
\hat{e}_{\text{orig}}(f) + \gamma\hat{e}_{\text{switch}}(f) &\geq \hat{h}_{+,\gamma}(\hat{g}_{+,\gamma}) \\
1 + \gamma\frac{\hat{e}_{\text{switch}}(f)}{\hat{e}_{\text{orig}}(f)} &\geq \frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\hat{e}_{\text{orig}}(f)} \\
1 + \gamma\frac{\hat{e}_{\text{switch}}(f)}{\hat{e}_{\text{orig}}(f)} &\geq \frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} && \text{from Eq C.11} \\
1 + \gamma\widehat{MR}(f) &\geq \frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} \\
\widehat{MR}(f) &\leq \left\{\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} - 1\right\}\gamma^{-1}.
\end{aligned}$$

C.4.2. Part 2: Showing that if $f = \hat{g}_{+,\gamma}$, and at Least One of the Enequalities in Condition 12 Holds with Equality, then Eq 6.4 Holds with Equality.

We consider each of the two inequalities in Condition 12 separately. If $\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma}) = 0$, then

$$\begin{aligned}
0 &= \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) + \gamma\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma}) \\
-\gamma\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma}) &= \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) \\
-\frac{\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma})} &= \frac{1}{\gamma}.
\end{aligned}$$

As a result,

$$\left\{\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} - 1\right\}\gamma^{-1} = \left\{\frac{0}{\epsilon_{\text{abs}}} - 1\right\}\left\{-\frac{\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma})}\right\} = \widehat{MR}(\hat{g}_{+,\gamma}).$$

Alternatively, if $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) = \epsilon_{\text{abs}}$, then

$$\begin{aligned}
\left\{\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} - 1\right\}\gamma^{-1} = \left\{\frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) + \gamma\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma})} - 1\right\}\gamma^{-1} &= \left\{1 + \gamma\frac{\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma})}{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma})} - 1\right\}\gamma^{-1} \\
&= \widehat{MR}(\hat{g}_{+,\gamma}).
\end{aligned}$$

■

C.5. Proof of Lemma 14 (Monotonicity for MR Upper Bound Binary Search)

Proof We prove Lemma 14 in 3 parts.

C.5.1. Part 1: $\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})$ is Monotonically Increasing in γ.

Let $\gamma_1, \gamma_2 \in \mathbb{R}$ satisfy $\gamma_1 < \gamma_2$. We have assumed that $0 \leq \hat{e}_{\text{switch}}(f)$ for any $f \in \mathcal{F}$. Thus, for any $f \in \mathcal{F}$ we have

$$\begin{aligned} \hat{e}_{\text{orig}}(f) + \gamma_1 \hat{e}_{\text{switch}}(f) &< \hat{e}_{\text{orig}}(f) + \gamma_2 \hat{e}_{\text{switch}}(f) \\ \hat{h}_{+,\gamma_1}(f) &< \hat{h}_{+,\gamma_2}(f). \end{aligned} \tag{C.12}$$

Applying this, we have

$$\begin{aligned} \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_1}) &\leq \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_2}) && \text{from Eq C.2} \\ &< \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_2}) && \text{from Eq C.12.} \end{aligned}$$

C.5.2. Part 2: $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma})$ is Monotonically Decreasing in γ for $\gamma \leq 0$, and Condition 12 Holds for $\gamma = 0$ and $\epsilon_{\text{abs}} \geq \min_{f \in \mathcal{F}} \hat{e}_{\text{orig}}(f)$.

Let $\gamma_1, \gamma_2 \in \mathbb{R}$ satisfy $\gamma_1 < \gamma_2 \leq 0$. Then

$$\begin{aligned} \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_1}) &\leq \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_2}) && \text{from Eq C.2} \\ \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_1}) + (\gamma_1 - \gamma_2)\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) &\leq \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_2}) + (\gamma_1 - \gamma_2)\hat{e}_{\text{switch}}(g_{\gamma_2}) && \text{from Eq C.3} \\ (\gamma_1 - \gamma_2)\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) &\leq (\gamma_1 - \gamma_2)\hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_2}) && \text{from Eqs C.1 \& C.2} \\ \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) &\geq \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_2}) \\ \gamma_2 \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) &\leq \gamma_2 \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_2}) && \text{from } \gamma_2 \leq 0. \end{aligned} \tag{C.13}$$

Now we are equipped to show the result that $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma})$ is monotonically decreasing in γ for $\gamma \leq 0$:

$$\begin{aligned} \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_2}) &\leq \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_1}) && \text{from Eq C.2} \\ \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) + \gamma_2 \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_2}) &\leq \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}) + \gamma_2 \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) \\ \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) &\leq \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}) && \text{from Eqs C.1 \& C.13.} \end{aligned} \tag{C.14}$$

To show that Condition 12 holds for $\gamma = 0$ and $\min_{f \in \mathcal{F}} \hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}$, we first note that $h_{0,+}(g_{0,+}) = \hat{e}_{\text{orig}}(g_{0,+})$, which is positive by assumption. Second, we note that

$$\hat{e}_{\text{orig}}(g_{0,+}) = h_{0,+}(g_{0,+}) = \min_{f \in \mathcal{F}} h_{0,+}(f) = \min_{f \in \mathcal{F}} \hat{e}_{\text{orig}}(f) \leq \epsilon_{\text{abs}}.$$

C.5.3. Part 3: $\left\{\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} - 1\right\}\gamma^{-1}$ is Monotonically Increasing in γ in the Range Where $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) \leq \epsilon_{\text{abs}}$ and $\gamma < 0$, and Decreasing in the Range Where $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) > \epsilon_{\text{abs}}$ and $\gamma < 0$.

To prove the first result, suppose that $\gamma_1 < \gamma_2 < 0$ and $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}), \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) \leq \epsilon_{\text{abs}}$. This implies

$$\begin{aligned} \frac{1}{\gamma_2} &< \frac{1}{\gamma_1} \\ \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}) - \epsilon_{\text{abs}}}{\gamma_2} &> \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}) - \epsilon_{\text{abs}}}{\gamma_1}. \end{aligned} \tag{C.15}$$

Then, starting with Eq C.2,

$$\begin{aligned} \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_2}) &\leq \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_1}) \\ \hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_2}) &\leq \gamma_2 \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) + \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}) \\ \frac{\hat{h}_{+,\gamma_2}(\hat{g}_{+,\gamma_2}) - \epsilon_{\text{abs}}}{\gamma_2} &\geq \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) + \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}) - \epsilon_{\text{abs}}}{\gamma_2} \\ &\geq \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_1}) + \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}) - \epsilon_{\text{abs}}}{\gamma_1} && \text{from Eq C.15} \\ &= \frac{\hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_1}) - \epsilon_{\text{abs}}}{\gamma_1}. \end{aligned}$$

Dividing both sides of the above equation by ϵ_{abs} proves that $\left\{\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} - 1\right\}\gamma^{-1}$ is monotonically increasing in γ in the range where $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) \leq \epsilon_{\text{abs}}$ and $\gamma < 0$.

To prove the second result we proceed in the same way. Suppose that $\gamma_1 < \gamma_2 < 0$ and $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_1}), \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) \geq \epsilon_{\text{abs}}$. This implies

$$\begin{aligned} \frac{1}{\gamma_2} &< \frac{1}{\gamma_1} \\ \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) - \epsilon_{\text{abs}}}{\gamma_2} &< \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) - \epsilon_{\text{abs}}}{\gamma_1}. \end{aligned} \tag{C.16}$$

Then, starting with Eq C.2,

$$\begin{aligned} \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_1}) &\leq \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_2}) \\ \hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_1}) &\leq \gamma_1 \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_2}) + \hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) \\ \frac{\hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_1}) - \epsilon_{\text{abs}}}{\gamma_1} &\geq \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_2}) + \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) - \epsilon_{\text{abs}}}{\gamma_1} \\ &\geq \hat{e}_{\text{switch}}(\hat{g}_{+,\gamma_2}) + \frac{\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma_2}) - \epsilon_{\text{abs}}}{\gamma_2} && \text{from Eq C.16} \\ &= \frac{\hat{h}_{+,\gamma_1}(\hat{g}_{+,\gamma_2}) - \epsilon_{\text{abs}}}{\gamma_2}. \end{aligned}$$

Diving both sides of the above equation by ϵ_{abs} proves that $\left[\left\{\frac{\hat{h}_{+,\gamma}(\hat{g}_{+,\gamma})}{\epsilon_{\text{abs}}} - 1\right\}\gamma^{-1}\right]$ is monotonically decreasing in γ in the range where $\hat{e}_{\text{orig}}(\hat{g}_{+,\gamma}) > \epsilon_{\text{abs}}$ and $\gamma < 0$. ■

C.6. Proof of Remark 16 (Tractability of Empirical MCR for Linear Model Classes)

Proof To show Remark 16, we apply Proposition 15 to see that

$$\begin{aligned}
&\xi_{\text{orig}}\hat{e}_{\text{orig}}(f_\beta) + \xi_{\text{switch}}\hat{e}_{\text{switch}}(f_\beta) \\
&= \frac{\xi_{\text{orig}}}{n}\,||\mathbf{y} - \mathbf{X}\beta||_2^2 + \xi_{\text{switch}}\hat{e}_{\text{switch}}(f_\beta) \\
&= \frac{\xi_{\text{orig}}}{n}\left(\mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\beta + \beta'\mathbf{X}'\mathbf{X}\beta\right) \\
&\qquad + \frac{\xi_{\text{switch}}}{n}\left\{\mathbf{y}'\mathbf{y} - 2\begin{bmatrix}\mathbf{X}_1'\mathbf{W}\mathbf{y} \\ \mathbf{X}_2'\mathbf{y}\end{bmatrix}'\beta + \beta'\begin{bmatrix}\mathbf{X}_1'\mathbf{X}_1 & \mathbf{X}_1'\mathbf{W}\mathbf{X}_2 \\ \mathbf{X}_2'\mathbf{W}\mathbf{X}_1 & \mathbf{X}_2'\mathbf{X}_2\end{bmatrix}\beta\right\} \\
&\propto_\beta -2\mathbf{q}'\beta + \beta'\mathbf{Q}\beta.
\end{aligned}$$

■

C.7. Proof of Lemma 17 (Loss Upper Bound for Linear Models)

Proof Under the conditions in Lemma 17 and Eq 7.5, we can construct an upper bound on $L(f_\beta, (y, x)) = (y - x'\beta)^2$ by either maximizing or minimizing $x'\beta$. First, we consider the maximization problem

$$\max_{\beta, x \in \mathbb{R}^p} x'\beta \text{ subject to } x'\mathbf{M}_{\text{lm}}^{-1}x \le r_{\mathcal{X}} \text{ and } \beta'\mathbf{M}_{\text{lm}}\beta \le r_{\text{lm}}. \tag{C.17}$$

We can see that both constraints hold with equality at the solution to this problem. Next, we apply the change of variables $\tilde{x} = \frac{1}{\sqrt{r_{\mathcal{X}}}}\mathbf{D}^{\frac{-1}{2}}\mathbf{U}'x$ and $\tilde{\beta} = \frac{1}{\sqrt{r_{\text{lm}}}}\mathbf{D}^{\frac{1}{2}}\mathbf{U}'\beta$, where $\mathbf{UDU}' = \mathbf{M}_{\text{lm}}$ is the eigendecomposition of $\mathbf{M}_{\text{lm}}$. We obtain

$$\max_{\tilde{\beta}, \tilde{x} \in \mathbb{R}^p} \tilde{x}'\tilde{\beta}\sqrt{r_{\mathcal{X}} r_{\text{lm}}} \text{ subject to } \tilde{x}'\tilde{x} = 1 \text{ and } \tilde{\beta}'\tilde{\beta} = 1,$$

which has an optimal objective value equal to $\sqrt{r_{\mathcal{X}} r_{\text{lm}}}$. By negating the objective in Eq C.17, we see that the minimum possible value of $x'\beta$, subject to the constraints in Eq 7.5 and Lemma 17, is found at $-\sqrt{r_{\mathcal{X}} r_{\text{lm}}}$. Thus, we know that

$$L(f, (y, x_1, x_2)) \le \max\left[\left\{\left(\min_{y \in \mathcal{Y}} y\right) - \sqrt{r_{\mathcal{X}} r_{\text{lm}}}\right\}^2, \left\{\left(\max_{y \in \mathcal{Y}} y\right) + \sqrt{r_{\mathcal{X}} r_{\text{lm}}}\right\}^2\right],$$

for any $(y, x_1, x_2) \in (\mathcal{Y} \times \mathcal{X}_1 \times \mathcal{X}_2)$. ■

C.8. Proof of Lemma 18 (Loss Upper Bound for Regression in a RKHS)

This proofs follows a similar structure as the proof in Section C.7. From the assumptions of Lemma 18, we know from Eq 7.7 that the largest possible output from a model $f_\alpha \in \mathcal{F}_{\mathbf{D}, r_k}$

is

$$\begin{aligned}
&\mu + \max_{x \in \mathbb{R}^p, \alpha \in \mathbb{R}^R} \sum_{i=1}^{R} k(x, \mathbf{D}_{[i,\cdot]})\alpha_{[i]} && \text{subject to } v(x)'\mathbf{K}_{\mathbf{D}}^{-1}v(x) \le r_{\mathbf{D}} \text{ and } \alpha'\mathbf{K}_{\mathbf{D}}\alpha \le r_k \\
=&\mu + \max_{x \in \mathbb{R}^p, \alpha \in \mathbb{R}^R} v(x)'\alpha && \text{subject to } v(x)'\mathbf{K}_{\mathbf{D}}^{-1}v(x) \le r_{\mathbf{D}} \text{ and } \alpha'\mathbf{K}_{\mathbf{D}}\alpha \le r_k \\
\le&\mu + \max_{\mathbf{z}, \alpha \in \mathbb{R}^R} \mathbf{z}'\alpha && \text{subject to } \mathbf{z}'\mathbf{K}_{\mathbf{D}}^{-1}\mathbf{z} \le r_{\mathbf{D}} \text{ and } \alpha'\mathbf{K}_{\mathbf{D}}\alpha \le r_k.
\end{aligned}$$

The above problem can be solved in the same way as Eq C.17, and has a solution at $\left(\mu + \sqrt{r_{\mathbf{D}} r_k}\right)$. The smallest possible model output will similarly be lower bounded by $-\left(\mu + \sqrt{r_{\mathbf{D}} r_k}\right)$. Thus, B_{ind} is less than or equal to

$$\max\left[\left\{\min_{y \in \mathcal{Y}}(y) - (\mu + \sqrt{r_{\mathbf{D}} r_k})\right\}^2, \left\{\max_{y \in \mathcal{Y}}(y) + (\mu + \sqrt{r_{\mathbf{D}} r_k})\right\}^2\right].$$

References

André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

Kellie J Archer and Ryan V Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4):2249–2260, 2008.

Razia Azen, David V Budescu, and Benjamin Reiser. Criticality of predictors in multiple regression. *British Journal of Mathematical and Statistical Psychology*, 54(2):201–225, 2001.

Katherine Beckett, Kris Nyrop, and Lori Pfingst. Race, drugs, and policing: understanding disparities in drug delivery arrests. *Criminology*, 44(1):105–137, 2006.

Irene V Blair, Charles M Judd, and Kristine M Chapleau. The influence of afrocentric facial features in criminal sentencing. *Psychological science*, 15(10):674–679, 2004.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Leo Breiman et al. Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

M Luz Calle and Víctor Urrea. Letter to the editor: stability of random forest importance measures. *Briefings in bioinformatics*, 12(1):86–89, 2010.

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Alexandra Chouldechova. Fair prediction with disparate impact: a study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

Beau Coker, Cynthia Rudin, and Gary King. A theory of statistical inference for ensuring the robustness of scientific results. *arXiv preprint arXiv:1804.08646*, 2018.

Sam Corbett-Davies, Emma Pierson, Avi Feller, and Sharad Goel. A computer program used for bail and sentencing decisions was labeled biased against blacks. it's actually not that clear. *The Washington Post*, October 2016. URL `https://www.washingtonpost.com/news/monkey-cage/wp/2016/10/17/can-an-algorithm-be-racist-our-analysis-is-more-cautious-than-propublicas/?utm_term=.e896ff1e4107`.

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806. ACM, 2017.

Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: theory and experiments with learning systems. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 598–617. IEEE, 2016.

Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3):837–845, 1988.

Olga V Demler, Michael J Pencina, and Ralph B D'Agostino Sr. Misuse of delong test to compare aucs for nested models. *Statistics in medicine*, 31(23):2577–2587, 2012.

Iván Díaz, Alan Hubbard, Anna Decker, and Mitchell Cohen. Variable importance and prediction methods for longitudinal problems with missing variables. *PloS one*, 10(3): e0120031, 2015.

Werner Dinkelbach. On nonlinear fractional programming. *Management science*, 13(7): 492–498, 1967.

Jiayun Dong and Cynthia Rudin. Variable importance clouds: A way to explore variable importance for the set of good models. *arXiv preprint arXiv:1901.03209*, 2019.

R Dorfman. A note on the delta-method for finding variance formulae. *The Biometric Bulletin*, 1(129-137):92, 1938.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

Muriel Gevrey, Ioannis Dimopoulos, and Sovan Lek. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, 160(3):249–264, 2003.

Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis*, 90:15–35, 2015.

Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3):659–678, 2017.

Alexander Hapfelmeier, Torsten Hothorn, Kurt Ulm, and Carolin Strobl. A new variable importance measure for random forests with missing data. *Statistics and Computing*, 24 (1):21–34, 2014.

T Hastie, R Tibshirani, and J Friedman. *The elements of statistical learning 2nd edition*. New York: Springer, 2009.

Karl G Heider. The Rashomon effect: when ethnographers disagree. *American Anthropologist*, 90(1):73–81, 1988.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The annals of mathematical statistics*, pages 293–325, 1948.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL `https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1963.10500830`.

Giles Hooker. Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732, 2007.

Reiner Horst and Nguyen V Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

Faisal Kamiran, Indrė Žliobaitė, and Toon Calders. Quantifying explainable discrimination and removing illegal discrimination in automated decision making. *Knowledge and information systems*, 35(3):613–644, 2013.

Jalil Kazemitabar, Arash Amini, Adam Bloniarz, and Ameet S Talwalkar. Variable importance using decision trees. In *Advances in Neural Information Processing Systems*, pages 425–434, 2017.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica*, May 2016. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

Guillaume Lecué. *Interplay between concentration, complexity and geometry in learning theory with applications to high dimensional data analysis*. PhD thesis, Université Paris-Est, 2011.

Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

Benjamin Letham, Portia A Letham, Cynthia Rudin, and Edward P Browne. Prediction uncertainty and optimal experimental design for learning dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(6):063110, 2016.

Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pages 431–439, 2013.

Kristian Lum and William Isaac. To predict and serve? *Significance*, 13(5):14–19, 2016.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Lucas Mentch and Giles Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*, 17(1):841–881, 2016.

John Monahan and Jennifer L Skeem. Risk assessment in criminal sentencing. *Annual review of clinical psychology*, 12:489–513, 2016.

Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2018, page 1931. NIH Public Access, 2018.

Daniel Nevo and Ya'acov Ritov. Identifying a minimal class of models for high-dimensional data. *The Journal of Machine Learning Research*, 18(1):797–825, 2017.

Julian D Olden, Michael K Joy, and Russell G Death. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, 178(3):389–397, 2004.

Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv preprint arXiv:1703.07870*, 2017.

Raymond Paternoster and Robert Brame. Reassessing race disparities in maryland capital cases. *Criminology*, 46(4):971–1008, 2008.

Sarah Picard-Fritsche, Michael Rempel, Jennifer A. Tallon, Julian Adler, and Natalie Reyes. Demystifying risk assessment, key principles and controversies. Technical report, 2017. Available at `https://www.courtinnovation.org/publications/demystifying-risk-assessment-key-principles-and-controversies`.

Imre Pólik and Tamás Terlaky. A survey of the s-lemma. *SIAM review*, 49(3):371–418, 2007.

Rajeev Ramchand, Rosalie Liccardo Pacula, and Martin Y Iguchi. Racial differences in marijuana-users' risk of arrest in the united states. *Drug and alcohol dependence*, 84(3): 264–272, 2006.

Friedrich Recknagel, Mark French, Pia Harkonen, and Ken-Ichi Yabunaka. Artificial neural network approach for modelling and prediction of algal blooms. *Ecological Modelling*, 96 (1-3):11–28, 1997.

Wendy D Roth and Jal D Mehta. The Rashomon effect: combining positivist and interpretivist approaches in the analysis of contested events. *Sociological Methods & Research*, 31(2):131–173, 2002.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, May 2019.

Cynthia Rudin, Caroline Wang, and Beau Coker. The age of secrecy and unfairness in recidivism prediction. *Harvard Data Science Review*, 2019. accepted.

Michele Scardi and Lawrence W Harding. Developing an empirical model of phytoplankton primary production: a neural network case study. *Ecological modelling*, 120(2):213–223, 1999.

Lesia Semenova and Cynthia Rudin. A study in rashomon curves and volumes: a new perspective on generalization and model simplicity in machine learning. *arXiv preprint arXiv:1908.01755*, 2019.

Robert J Serfling. *Approximation theorems of mathematical statistics*. John Wiley & Sons, 1980.

Cassia Spohn. Thirty years of sentencing reform: the quest for a racially neutral sentencing process. *Criminal justice*, 3:427–501, 2000.

Alexander Statnikov, Nikita I Lytkin, Jan Lemeire, and Constantin F Aliferis. Algorithms for discovery of multiple markov boundaries. *Journal of Machine Learning Research*, 14 (Feb):499–566, 2013.

Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC bioinformatics*, 8(1):25, 2007.

Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9 (1):307, 2008.

Elizabeth A Stuart. Matching methods for causal inference: a review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

Laura Toloşi and Thomas Lengauer. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27(14):1986–1994, 2011.

Theja Tulabandhula and Cynthia Rudin. Robust optimization using machine learning for uncertainty sets. *arXiv preprint arXiv:1407.1097*, 2014.

U.S. Department of Justice - Civil Rights Devision. Investigation of the Baltimore City Police Department, August 2016. Available at `https://www.justice.gov/crt/file/883296/download`.

Mark J van der Laan. Statistical inference for variable importance. *The International Journal of Biostatistics*, 2(1), 2006.

Jay M Ver Hoef. Who invented the delta method? *The American Statistician*, 66(2): 124–127, 2012.

Huazhen Wang, Fan Yang, and Zhiyuan Luo. An experimental study of the intrinsic stability of random forest variable importance measures. *BMC bioinformatics*, 17(1):60, 2016.

Brian D Williamson, Peter B Gilbert, Noah Simon, and Marco Carone. Nonparametric variable importance assessment using machine learning techniques. *bepress (unpublished preprint)*, 2017.

Jingtao Yao, Nicholas Teng, Hean-Lee Poh, and Chew Lim Tan. Forecasting and analysis of marketing data using neural networks. *J. Inf. Sci. Eng.*, 14(4):843–862, 1998.

Ruoqing Zhu, Donglin Zeng, and Michael R Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.

Journal of Machine Learning Research 20 (2019) 1-29 Submitted 1/19; Revised 6/19; Published 10/19

Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning

Daniel C. Castro dc315@imperial.ac.uk

Jeremy Tan jht16@imperial.ac.uk

Bernhard Kainz b.kainz@imperial.ac.uk
Biomedical Image Analysis Group
Imperial College London
London SW7 2AZ, United Kingdom

Ender Konukoglu ender.konukoglu@vision.ee.ethz.ch
Computer Vision Laboratory
ETH Zürich
8092 Zürich, Switzerland

Ben Glocker b.glocker@imperial.ac.uk
Biomedical Image Analysis Group
Imperial College London
London SW7 2AZ, United Kingdom

Editor: Isabelle Guyon

Abstract

Revealing latent structure in data is an active field of research, having introduced exciting technologies such as variational autoencoders and adversarial networks, and is essential to push machine learning towards unsupervised knowledge discovery. However, a major challenge is the lack of suitable benchmarks for an objective and quantitative evaluation of learned representations. To address this issue we introduce Morpho-MNIST, a framework that aims to answer: "to what extent has my model learned to represent specific factors of variation in the data?" We extend the popular MNIST dataset by adding a morphometric analysis enabling quantitative comparison of trained models, identification of the roles of latent variables, and characterisation of sample diversity. We further propose a set of quantifiable perturbations to assess the performance of unsupervised and supervised methods on challenging tasks such as outlier detection and domain adaptation. Data and code are available at `https://github.com/dccastro/Morpho-MNIST`.

Keywords: representation learning, generative models, empirical evaluation, disentanglement, morphometrics

1. Introduction

A key factor for progress in machine learning has been the availability of well curated, easy-to-use, standardised and sufficiently large annotated datasets for benchmarking different algorithms and models. This has led to major advances in speech recognition, computer vision, and natural language processing. A commonality between these tasks is their natural

©2019 Daniel Coelho de Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v20/19-033.html`.

formulation as supervised learning tasks, wherein performance can be measured in terms of accuracy on a test set.

The general problem of representation learning (i.e. to reveal latent structure in data) is more difficult to assess due the lack of suitable benchmarks. Although the field is very active, with many recently proposed techniques such as probabilistic autoencoders and adversarial learning, it is less clear where the field stands in terms of progress or which approaches are more expressive for specific tasks. The lack of reproducible ways to quantify performance has led to subjective means of evaluation: visualisation techniques have been used to show low-dimensional projections of the latent space and visual inspection of generated or reconstructed samples are popular to provide subjective measures of descriptiveness. On the other hand, the quality of sampled images generally tells us little about how well the learned representations capture known factors of variation in the training distribution. In order to advance progress, the availability of tools for objective assessment of representation learning methods seems essential yet lacking.

This paper introduces Morpho-MNIST, a collection of shape metrics and perturbations, in a step towards quantitative assessment of representation learning. We build upon one of the most popular machine learning benchmarks, MNIST, which despite its shortcomings remains widely used. While MNIST was originally constructed to facilitate research in image classification, in the form of recognising handwritten digits (LeCun et al., 1998), it has found its use in representation learning, for example, to demonstrate that the learned latent space yields clusters consistent with digit labels. Methods aiming to disentangle the latent space claim success if individual latent variables capture specific style variations (e.g. stroke thickness, sidewards leaning digits and other visual characteristics).

It is important to emphasise that MNIST is a prime representative for datasets of rasterised two-dimensional shapes. In this type of data, while the underlying shapes may have relatively simple and well understood factors of variation, they manifest in the raster images as high-order correlations between large numbers of pixels. The challenge for a learning agent is to abstract away the low-level pixel intensity variations and recover a meaningful shape representation. MNIST is one of the interesting cases wherein it is possible to explicitly measure morphological attributes from the images themselves, enabling us to directly evaluate to what extent a trained model has learned to represent them.

More generally, MNIST remains popular in practice due to several factors: it allows reproducible comparisons with previous results reported in the literature; the dataset is sufficiently large for its complexity and consists of small, two-dimensional greyscale images defining a tractable ten-class classification problem; computation and memory requirements are low; most popular deep learning frameworks and libraries offer tutorials using MNIST, which makes it straightforward for new researchers to enter the field and to experiment with new ideas and explore latest developments. Furthermore, there is now renewed interest in MNIST due to the recent replication of the original generation pipeline and the rediscovery of additional 50,000 test images, beckoning researchers to re-evaluate over 20 years of research on MNIST (Yadav and Bottou, 2019). We leverage these qualities and extend MNIST in multiple ways, as summarised in the following.

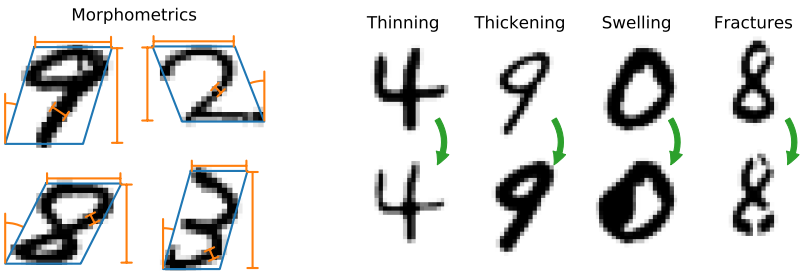


Figure 1: *Left:* MNIST morphometrics—stroke thickness and length (not shown), width, height and slant of digits. *Right:* MNIST perturbations (many more examples of each type in Appendix B).

1.1. Contributions

Our aim is to provide a useful resource to the machine learning community, opening a variety of new avenues for experimentation and analysis. As we preserve the general properties of MNIST—such as image size, file format, numbers of training and test images, and the original ten-class classification problem—we hope this new quantitative framework for assessing representation learning will experience widespread adoption and may inspire further extensions facilitated by a publicly available Morpho-MNIST code base.

1.1.1. Morphometrics

We propose to describe real and generated digit images in terms of measurable shape attributes. These include stroke thickness and length, and the width, height, and slant of digits (see Fig. 1, left). Whereas some of these MNIST properties have been analysed only visually in previous work, we demonstrate that quantifying each of them allows us to objectively characterise the roles of inferred representations.

More generally, the exclusive reliance on synthetic annotated datasets (Section 1.2.4) has been a crucial limitation for research into evaluation of learned representations (Eastwood and Williams, 2018), as this restricts us to the analysis of *inferential* behaviour (data → representation), even when studying generative models. The tools we introduce here can be used to measure model samples, effectively enabling direct assessment of *generative* performance (representation → data).

These measurements can be directly employed to re-evaluate existing models and may be added retrospectively to previous experiments involving the original MNIST dataset. Adoption of our morphometric analysis may provide new insights into the effectiveness of representation learning methods in terms of revealing meaningful latent structures. Furthermore, for other datasets it suffices to design the relevant scalar metrics and include them in the very same evaluation framework.

1.1.2. Perturbations

We introduce a set of parametrisable global and local perturbations, inspired by natural and pathological variability in medical images. Global changes involve overall thinning and thickening of digits, while local changes include both swelling and fractures (see examples on the right in Fig. 1 and many more in Appendix B). The perturbation framework we present in Section 3—illustrating various ways to exploit the outputs of the proposed processing pipeline (e.g. skeleton, distance map, and even morphometrics themselves; see Section 2.1)—is sufficiently general that additional transformations can easily be designed as necessary. Injecting these perturbations into the dataset adds a new type of complexity to the data manifold and opens up a variety of interesting applications.

The proposed perturbations are designed to enable a wide range of new studies and applications for both supervised and unsupervised tasks. Detection of 'abnormalities' (i.e. local perturbations) is an evident application, although more challenging tasks can also be defined, such as classification from noisy/corrupted data, domain adaptation, localisation of perturbations, characterising semantics of learned latent representations, and more. We explore a few supplementary examples of supervised tasks in Section 4.3.

1.2. Related Work: Datasets

In this section, we provide an overview of some datasets that are related to MNIST, by either sharing its original source content, containing transformations of the original MNIST images or being distributed in the same format for easy replacement. We also mention a few prevalent datasets of synthetic images with generative factor annotations, analogous to the morphometrics proposed in this paper.

1.2.1. NIST Datasets

The MNIST (modified NIST) dataset (LeCun et al., 1998) was constructed from handwritten digits in NIST Special Databases 1 and 3, now released as Special Database 19 (Grother and Hanaoka, 2016). Cohen et al. (2017) generated a much larger dataset based on the same NIST database, containing additional upper- and lower-case letters, called EMNIST (extended MNIST). LeCun et al.'s original processing pipeline was recently replicated, enabling the rediscovery of 50,000 test images that had been excluded from MNIST—a dataset now released as QMNIST (Yadav and Bottou, 2019).

1.2.2. MNIST Perturbations

The seminal paper by LeCun et al. (1998) employed data augmentation using planar affine transformations including translation, scaling, squeezing, and shearing. Loosli et al. (2007) employed random elastic deformations to construct the Infinite MNIST dataset. Other MNIST variations include rotations and insertion of random and structured background (Larochelle et al., 2007), and Tieleman (2013) applied spatial affine transformations and provided ground-truth transformation parameters.

1.2.3. MNIST Format

Due to the ubiquity of the MNIST dataset in machine learning research and the resulting multitude of compatible model architectures available, it is appealing to release new datasets in the same format (28×28, 8-bit grayscale images). One such effort is Fashion-MNIST (Xiao et al., 2017), containing images of clothing articles from ten distinct classes, adapted from an online shopping catalogue. Another example is notMNIST (Bulatov, 2011), a dataset of character glyphs for letters 'A'–'J' (also ten classes), in a challengingly diverse collection of typefaces.

1.2.4. Annotated Datasets

Synthetic computer vision datasets that are popular for evaluating disentanglement of learned latent factors of variation include those from Paysan et al. (2009) and Aubry et al. (2014). They contain 2D renderings of 3D faces and chairs, respectively, with ground-truth pose parameters (azimuth, elevation) and lighting conditions (faces only). A further initiative in that direction is the dSprites dataset (Matthey et al., 2017), which consists of binary images containing three types of shapes with varying location, orientation and size. The availability of the ground-truth values of such attributes has motivated the accelerated adoption of these datasets in the evaluation of representation learning algorithms.

1.3. Related Work: Quantitative Evaluation

Evaluation of representation learning is a notoriously challenging task and remains an open research problem. Numerous solutions have been proposed, with many of the earlier ones focusing on the test log-likelihood under the model (Kingma and Welling, 2014) or, for likelihood-free models, under a kernel density estimate (KDE) of generated samples (Goodfellow et al., 2014; Makhzani et al., 2016)—shown not to be reliable proxies for the true model likelihood (Theis et al., 2016).

Another perspective for evaluation of generative models of images is the visual fidelity of its samples to the training data, which would normally require manual inspection. To address this issue, a successful family of metrics have been proposed, based on visual features extracted by the Inception network (Szegedy et al., 2016). The original Inception score (Salimans et al., 2016) relies on the 'crispness' of class predictions, whereas the Fréchet Inception distance (FID, Heusel et al., 2017) and the kernel Inception distance (KID, Bińkowski et al., 2018) statistically compare high-level representations instead of the final network outputs.

Although the approaches above can reveal vague signs of mode collapse, it may be useful to diagnose this phenomenon on its own. With this objective, Arora et al. (2018) proposed to estimate the support of the learned distribution (assumed discrete) using the birthday paradox test, by counting pairs of visual duplicates among model samples. Unfortunately, the adoption of this technique is hindered by its reliance on manual visual inspection to flag identical images.

There have been several attempts at quantifying representation disentanglement performance. For example, Higgins et al. (2017) proposed to use the accuracy in predicting which factor of variation was held fixed in a simulated dataset. There exist further information-theoretic approaches, involving the KL divergence contribution from each latent dimension (Dupont, 2018) or their mutual information with each known generative factor (Chen et al.,

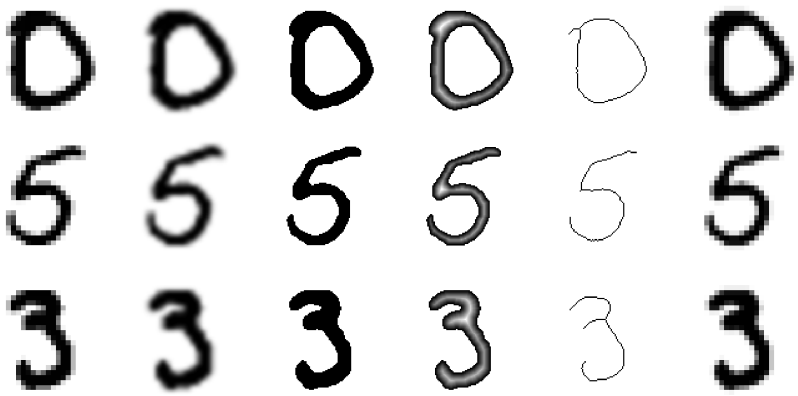


Figure 2: Stages of the image processing pipeline. *Left to right:* original image, upscaled image, binarised image, distance transform, skeleton, downscaled image.

2018). Yet another method, explored in Kumar et al. (2018), is based on the predictive accuracy of individual latent variables to each generative factor. The concurrent work of Eastwood and Williams (2018) introduced a comprehensive methodology for characterising different aspects of representation disentanglement.

2. Morphometry

Meaningful morphometrics are instrumental in characterising distributions of rasterised shapes, such as MNIST digits, and can be useful as additional data for downstream learning tasks. We regard the current choice of morphometrics as a minimal useful collection of attributes that are largely independent and can be estimated robustly. Although further more intricate metrics can be defined, they are likely to be entangled with the ones defined here and to be noisy due to the limited resolution of the original images. We begin this section by describing the image processing pipeline employed for extracting the metrics and for applying perturbations (Section 3), followed by details on the computation of each measurement.

2.1. Processing Pipeline

The original 28×28 resolution of the MNIST images is generally not high enough to enable satisfactory morphological processing: stroke properties (e.g. length, thickness) measured directly on the binarised images would likely be inaccurate and heavily quantised. To mitigate this issue and enable sub-pixel accuracy in the measurements, we propose to use the following processing steps:

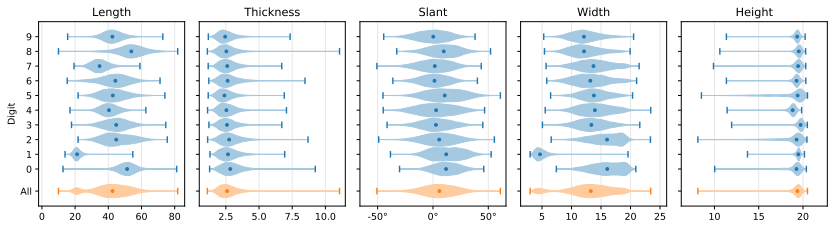


Figure 3: Distribution of morphological attributes in the plain MNIST training dataset

1. *Upscale* (e.g. ×4, to 112×112);[1]
2. *Binarise:* the blurry upscaled image is thresholded at half its intensity range (e.g. 128 for most images), which ensures thin or faint digits are not erased;
3. Compute *Euclidean distance transform* (EDT): each pixel within the digit boundaries contains the distance to its nearest boundary point;
4. *Skeletonise:* detect the ridges of the EDT, i.e. the locus of points equidistant to two or more boundary points (also known as the *medial axis*) (Blum, 1967);
5. Optionally apply *perturbation* to binarised image (cf. Section 3);
6. *Downscale* binarised or perturbed image to original resolution.

We illustrate the pipeline in Fig. 2. The binary high-resolution digits have smooth boundaries and faithfully capture subtle variations in contour shape and stroke thickness that are only vaguely discernible in the low-resolution images. Additionally, note how the final downscaled image is almost indistinguishable from the original.

All morphometric attributes described below are calculated for each digit after applying steps 1–4 of this pipeline. The distributions for plain MNIST are plotted in Fig. 3, and the distributions after applying each type of perturbation can be found in Appendix A.

2.2. Stroke Length

Here we approximate the trace of the pen tip, as a digit was being written, by the computed morphological skeleton. In this light, the total length of the skeleton is an estimate of the length of the pen stroke, which in turn is a measure of shape complexity.

It can be computed in a single pass by accumulating the Euclidean distance of each skeleton pixel to its immediate neighbours, taking care to only count the individual contributions once. This approach is more accurate and more robust against rotations than a naïve estimate by simply counting the pixels.

1. Up- and downscaling by a factor of f are done with bicubic interpolation and Gaussian smoothing (bandwidth $\sigma = 2f/6$), following `scikit-image` defaults (van der Walt et al., 2014).

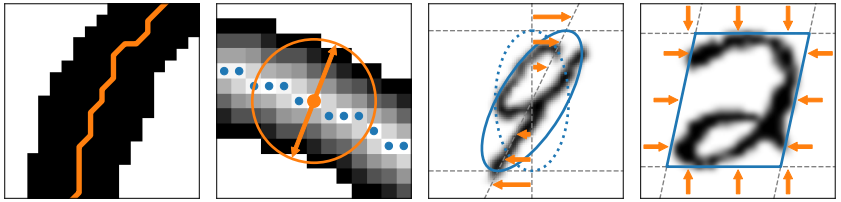


Figure 4: Measuring shape attributes. *Left to right:* stroke length (length of binarised image's skeleton); stroke thickness (average distance transform over skeleton pixels ×2); slant (horizontal shearing angle); width and height (bounding parallelogram).

2.3. Stroke Thickness

A prominent factor of style variation in the MNIST digits is the overall thickness of the strokes, due to both legitimate differences in pen thickness and force applied, and also to the rescaling of the original NIST images by different factors.

We estimate it by exploiting both the computed distance transform and the skeleton. Since skeleton pixels are equidistant to the nearest boundaries and the value of the EDT at those locations corresponds to the local half-width (Blum, 1967), we take twice the mean of the EDT over all skeleton pixels as our global estimate for stroke thickness.

2.4. Slant

The extent by which handwritten symbols lean right or left (forward and backward slant, respectively) is a further notorious and quantifiable dimension of handwriting style. It introduces so much variation in the appearance of characters in images that it is common practice in optical character recognition (OCR) systems to 'deslant' them, in an attempt to reduce within-class variance (LeCun et al., 1998; Teow and Loe, 2002).

We adapt the referred deslanting methodology to describe the slant *angle* of the handwritten digits. After estimating the second-order image moments,

$$\mathbf{S} = \begin{pmatrix} \sum_{i,j} x_{ij}(i-\bar{\imath})^2 & \sum_{i,j} x_{ij}(i-\bar{\imath})(j-\bar{\jmath}) \\ \sum_{i,j} x_{ij}(i-\bar{\imath})(j-\bar{\jmath}) & \sum_{i,j} x_{ij}(j-\bar{\jmath})^2 \end{pmatrix}, \tag{1}$$

we define the slant based on the horizontal shear:

$$\alpha = \arctan\left(-\frac{S_{12}}{S_{22}}\right), \tag{2}$$

where x_{ij} is the intensity of the pixel at row j and column i, and $(\bar{\imath}, \bar{\jmath})$ are the centroid coordinates. The minus sign ensures that positive and negative values of α correspond to forward and backward slant, respectively.

Note that $-\alpha$ is the angle by which the image would need to be horizontally sheared toward the vertical centre line such that the digit is upright (i.e. principal axes aligned with the coordinate axes). Note how this differs from a *rotation* angle, which would instead be defined in terms of $\frac{1}{2}\arctan(2S_{12}/(S_{22} - S_{11}))$.

2.5. Width and Height

It is useful to measure other general shape attributes, such as width, height, and aspect ratio, which also present substantial variation related to personal handwriting style.[2] To this end, we propose to fit a *bounding parallelogram* to each digit, with horizontal and slanted sides (see Fig. 1). We highlight that it does not suffice to simply fit a bounding box with perpendicular sides, since the width would clearly be confounded with slant. These metrics would therefore contain redundant information that obfuscates the true digit width.

The image is swept top-to-bottom with a horizontal boundary to compute a vertical marginal cumulative distribution function (CDF), and likewise left-to-right with a *slanted* boundary for a horizontal marginal CDF, with angle α as computed above. The bounds are then chosen based on equal-tailed intervals containing a given proportion of the image mass—98% in both directions (1% from each side) proved accurate and robust in our experiments.

3. Perturbations

As discussed in Section 1, we bring forward a number of morphological perturbations for MNIST digits, to enable diverse applications and experimentation. In this section, we detail these parametrisable transformations, categorised as global or local.

3.1. Global: Thinning and Thickening

The first pair of transformations we present is based on simple morphological operations: the binarised image of a digit is dilated or eroded with a circular structuring element. Its radius is set proportionally to the estimated stroke thickness (Section 2.3), so that the overall thickness of each digit will decrease or increase by an approximately fixed factor (here, -70% and +100%; see Figs. B.1 and B.2).

Since there is substantial thickness variability in the original MNIST data (cf. Fig. 3) and most thinned and thickened digits look very plausible, we believe that these perturbations can constitute a powerful form of data augmentation for training. For the same reason, we have not included these perturbations in the abnormality detection experiments (Section 4.3.3).

3.2. Local: Swelling

In addition to the global transformations above, we introduce *local* perturbations with variable location and extent, which are harder to detect automatically. Given a radius R, a centre location $\mathbf{r}_0$ and a strength parameter $\gamma > 1$, the coordinates $\mathbf{r}$ of pixels within distance R of $\mathbf{r}_0$ are nonlinearly warped according to a radial power transform:

$$\mathbf{r} \mapsto \mathbf{r}_0 + (\mathbf{r} - \mathbf{r}_0)\Big(\frac{\|\mathbf{r} - \mathbf{r}_0\|}{R}\Big)^{\gamma-1}, \tag{3}$$

leaving the remaining portions of the image untouched and resampling with bicubic interpolation.

2. Little variation in height is expected, since the original handwritten digits were scaled to fit a 20×20 box (LeCun et al., 1998). Nevertheless, a minority of digits were originally wider than they were tall, which explains the long tails in the distribution of heights (Fig. 3).

In the experiments and released dataset, we set $\gamma = 7$ and $R = 3\sqrt{\theta}/2$, where θ is thickness. Unlike simple linear scaling with θ, this choice for R produces noticeable but not exaggerated effects across the thickness range observed in the dataset (cf. Fig. B.3). The centre location, $\mathbf{r}_0$, is picked uniformly at random from the pixels along the estimated skeleton.

3.3. Local: Fractures

Finally, we describe the proposed procedure for adding fractures to an MNIST digit, where we define a fracture as a break in the continuity of a pen stroke. Because single fractures can in many cases be easily mistaken for true gaps between strokes, we add multiple fractures to each affected digit.

When selecting where to place a fracture, we attempt to avoid locations too close to stroke tips (points on the skeleton with a single neighbour) or fork points (more than two neighbours). This is achieved by sampling only among those skeleton pixels above a certain distance to these detected points. In addition, we desire fractures to be transversal to the pen strokes. Local orientation is determined based on second-order moments of the skeleton inside a window centred at the chosen location, and the length of the fracture is estimated from the boundary EDT. Finally, the fracture is drawn onto the high-resolution binary image with a circular brush along the estimated normal.

In practice, we found that adding three fractures with 1.5 px thickness, 2 px minimum distance to tips and forks and angle window of $5\times5\,\text{px}^2$ ('px' as measured in the low resolution image) produces detectable but not too obvious perturbations (see Fig. B.4). We also extend the lines on both ends by 0.5 px to ensure some tolerance.

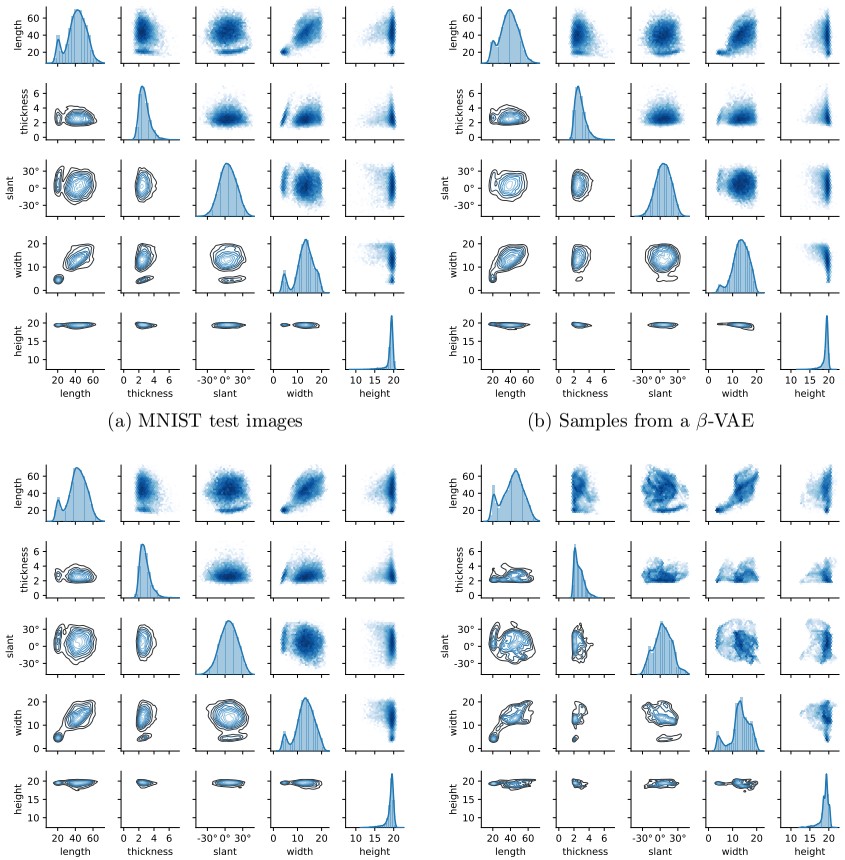
4. Evaluation Case Studies

In this section, we demonstrate potential uses of the proposed framework: using morphometrics to characterise the distribution of samples from generative models and finding associations between learned latent representations and morphometric attributes. In addition, we exemplify in Section 4.3 a variety of novel ways to make use of morphometry and perturbations for evaluating predictive models on MNIST and beyond.

4.1. Sample Diversity

Here we aim to illustrate ways in which the proposed MNIST morphometrics may be used to visualise distributions learned by generative models and to quantify their agreement with the true data distribution in terms of these concrete attributes. Moreover, extracting such measurements from model samples could be a step toward diagnosing mode collapse.

We exemplify this scenario with a generative adversarial network (GAN, Goodfellow et al., 2014) and a β-variational autoencoder (β-VAE, Higgins et al., 2017), both with generator (resp. decoder) and discriminator architecture as used in the MNIST experiments in Chen et al. (2016), and encoder mirroring the decoder. We train a β-VAE with $\beta = 4$, as in Higgins et al. (2017)'s experiments on small binary images, and a vanilla GAN with non-saturating loss, both with 64-dimensional latent space. To explore the behaviour of a much less expressive model, we additionally train a GAN with only two latent dimensions.



(a) MNIST test images

(b) Samples from a β-VAE

(c) Samples from a GAN

(d) Samples from a low-dimensional GAN

Figure 5: Distribution of morphometric attributes for MNIST test dataset and samples from some generative models. Diagonals show marginal histograms and KDEs, upper-triangular plots show pairwise log-histograms and lower-triangular plots show pairwise KDEs.

Test data vs.	Dims.	$\mathrm{MMD}_l^2 \pm$ std. error ($\times 10^{-3}$)	p
β-VAE	64	0.792 ± 1.569	.3068
GAN	64	1.458 ± 1.650	.1885
GAN	2	$\mathbf{8.876 \pm 1.807}$	**.0000**

Table 1: Kernel two-sample tests between model samples and true test data. The bold row indicates a significant divergence, thus a failure of the model in faithfully reproducing the data distribution.

4.1.1. Visualisation

Figure 5 illustrates the morphometric distributions of the plain MNIST test images and of 10,000 samples from each of these three models. As can be seen, morphometrics provide *interpretable* low-dimensional statistics which allow comparing distributions learned by generative models between each other and with true datasets. While Figs. 5b and 5c show model samples roughly as diverse as the true images, the samples from the low-dimensional GAN in Fig. 5d seem concentrated on certain regions, covering a distribution that is less faithful to the true one in Fig. 5a.

4.1.2. Statistical Comparison

We argue that in this lower-dimensional space of morphometrics it is possible to statistically compare the distributions, since this was shown not to be effective directly in image space (e.g. Theis et al., 2016). To this end, we propose to use kernel two-sample tests based on maximum mean discrepancy (MMD) between morphometrics of the test data and of each of the sample distributions (Gretton et al., 2012). The MMD statistic is a measure of dissimilarity between distributions, such that a value significantly greater than zero (small p-value) suggests a detectable difference. Here, we performed the linear-time asymptotic test described in Gretton et al. (2012, §6) (details and further considerations in Appendix C).

The results of these tests are displayed in Table 1, where we observe that samples from the high-dimensional β-VAE and GAN did not exhibit a significantly large MMD, thus there is no evidence of distributional mismatch. On the other hand, the low-dimensional GAN's samples show a significant departure from the data distribution, confirming the qualitative judgement based on comparing Figs. 5a and 5d.

4.2. Disentanglement

In this experiment, we demonstrate that: (a) standard MNIST can be augmented with morphometric attributes to quantitatively study representations computed by an *inference* model (as already possible with annotated datasets, e.g. dSprites and 3D faces); (b) we can measure shape attributes of samples to assess disentanglement of a *generative* model, which is unprecedented to the best of our knowledge; and (c) this analysis can also diagnose when a model unexpectedly *fails* to disentangle a known aspect of the data.

As for the sample diversity experiments in Section 4.1, the purpose of this disentanglement analysis is to study the behaviour of specific trained model instances, and results may not

generalise to the respective model classes or architectures, to different datasets, or even to distinct runs of the same set-up. Similarly to any other evaluation methodology, findings concerning disentanglement patterns are reproducible only to the extent that the model under study can be reliably retrained to a similar state.

4.2.1. Methodology

Using an approach related to the disentanglement measure introduced by Kumar et al. (2018), we study the correlation structures between known generative factors and learned latent codes. Specifically, we compute the *partial correlation* ($r_{c_i y_j \cdot \mathbf{c}_{-i}}$) between each latent code variable (c_i) and each morphometric attribute (y_j), controlling for the variation in the remaining latent variables ($\mathbf{c}_{-i}$).[3] As opposed to the simple correlation, this technique allows us to study the *net* first-order effect of each latent code, all else being equal.

This partial correlation is calculated as follows, ranging over codes i and attributes j:

$$r_{c_i y_j \cdot \mathbf{c}_{-i}} = -\frac{P^{(j)}_{1,i+1}}{\sqrt{P^{(j)}_{11} P^{(j)}_{i+1,i+1}}} \quad \in [-1, 1]\,, \tag{4}$$

where $\mathbf{P}^{(j)}$ is the sample precision matrix (i.e. inverse of sample covariance) of $(y_j, c_1, \ldots, c_D)$ over all data points. Note that Eq. (4) is closely related to the least-squares coefficients for multiple linear regression of y_j on $(c_1, \ldots, c_D)$, and its magnitude can be directly interpreted as a relative importance score within the framework of Eastwood and Williams (2018).

We plot the calculated partial correlations in the form of tables, where each row corresponds to a shape attribute, and columns represent the latent variables. These tables should be read row-wise, as the scores are normalised separately for each shape factor. Entries close to zero do not indicate that the latent dimension is unrelated to the measured attribute, but rather that it is entangled with others and has little marginal effect on its own.

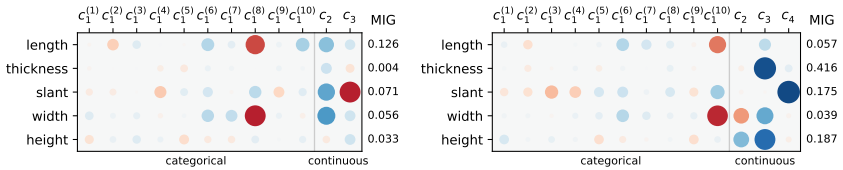
In addition, we demonstrate how our morphometry framework is fully compatible with contemporary disentanglement metrics by also computing the mutual information gap (MIG score; Chen et al., 2018) in each scenario. This score is based on the mutual information of each pair of generative factor (here, shape attribute) and latent code dimension, and reports the gap between the two leading codes for each factor, normalised to $[0, 1]$. It therefore quantifies how prominently the attribute is represented by a single latent dimension, i.e. its *completeness* in the terminology of Eastwood and Williams (2018). The overall MIG for each model is given by the average over all attributes.

For these experiments, we trained InfoGAN (Chen et al., 2016) instances with various latent space configurations, then took MAP estimates of latent codes for each image (i.e. maximal logit for categorical codes and mean for continuous codes) as predicted by the variational recognition network. Models were trained for 20 epochs using 64 images per batch, with no hyperparameter tuning. We emphasise that our goal was to illustrate how the proposed morphometrics can serve as tools to better understand whether models behave as intended, and not to optimally train them in each scenario.

3. For the categorical code, c_1, we take a single binary dummy variable for each category, $c_1^{(k)}$, while controlling only for the remaining codes (c_2, c_3 etc.) to avoid multicollinearity.

	# Cat.	# Cont.	# Bin.	Dataset
InfoGAN-A	10	2	0	PLAIN: plain only
InfoGAN-B	10	3	0	GLOBAL: plain + thinning + thickening
InfoGAN-C	10	2	2	LOCAL: plain + swelling + fractures

Table 2: Settings for InfoGAN disentanglement experiments



(a) InfoGAN-A (overall MIG = 0.0582) (b) InfoGAN-B (overall MIG = 0.1748)

Figure 6: Partial correlations between inferred latent codes and morphometrics of test images. Circle area and colour strength are proportional to correlation magnitude, blue is positive and red is negative. On the right, we indicate the mutual information gap score (MIG) for each attribute.

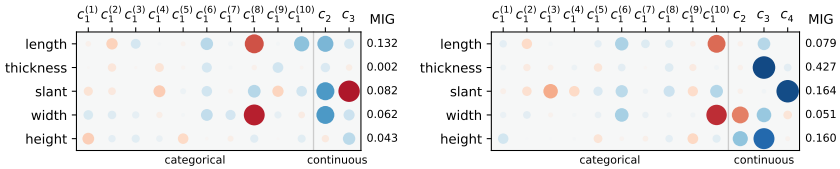
4.2.2. Inferential Disentanglement

To illustrate how this methodology can be applied in practice to assess disentanglement, we consider two settings. The first is the same as in the MNIST experiment from Chen et al. (2016), with a 10-way categorical and two continuous latent codes, trained and evaluated on the plain MNIST digits, to which we will refer as InfoGAN-A.

The second setting was designed to investigate whether the model could disentangle the concept of *thickness*, by including an additional continuous latent code and training on a dataset with exaggerated thickness variations. We constructed this dataset by randomly interleaving plain, thinned and thickened digit images in equal proportions. Since the perturbations were applied completely at random, we expect a trained generative model to identify that thickness should be largely independent of the other morphological attributes. We refer to this set-up as InfoGAN-B. Table 2 summarises the different experimental settings, for reference.

In Fig. 6a, we see that InfoGAN-A learned to encode slant mostly in c_3, while $c_1^{(8)}$ clearly relates to the '1' class (much narrower digit shape and shorter pen stroke; cf. Fig. 3). Figure 6b quantitatively confirms the hypothesis that InfoGAN-B's recognition network would learn to separate slant and thickness (in c_4 and c_3, resp.), the most prominent factors of style variation in this dataset. Interestingly, it shows that c_3 also associates with height, as thicker digits also tend to be taller.

Moreover, we observe that the MIG for each attribute correlates with the qualitative interpretation of these partial correlation tables. For example, thickness is represented considerably more compactly in InfoGAN-B (mainly by c_3) than in InfoGAN-A (highly entangled), exhibiting a correspondingly much higher MIG score. For InfoGAN-B, a



(a) InfoGAN-A (overall MIG = 0.0641) (b) InfoGAN-B (overall MIG = 0.1762)

Figure 7: Partial correlations between 1000 sampled latent codes and morphometrics of the corresponding generated images

noticeable portion of the slant variation was additionally captured by c_1, which is reflected in its lower MIG than that for thickness. Lastly, the difference in overall MIG between models validates our qualitative judgement about global disentanglement.

4.2.3. Generative Disentanglement

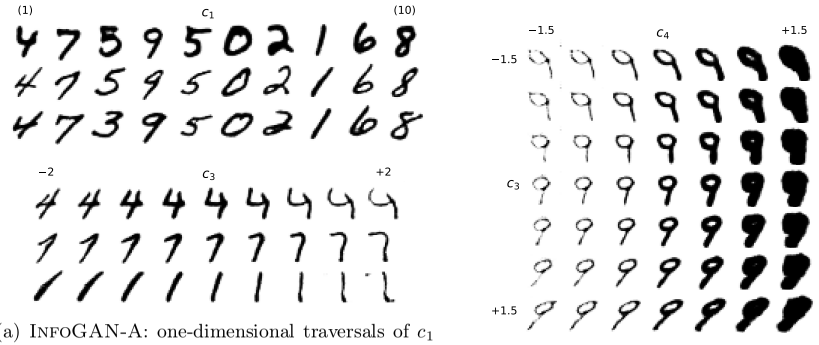
The evaluation methodology described above is useful for investigating the behaviour of the *inference* direction of a model, and can readily be used with datasets which include ground-truth generative factor annotations. On the other hand, unless we trust that the inference approximation is highly accurate, this tells us little about the *generative* expressiveness of the model. This is where computed metrics truly show their potential: one can measure generated samples, and investigate how their attributes relate to the latent variables used to create them.

Figure 7 shows results for a similar analysis to Fig. 6, but now evaluated on *samples* from that model. As the tables are mostly indistinguishable, we may argue that in this case the inference and generator networks have learned to consistently encode and decode the digit shape attributes. Further, the per-attribute and overall MIG scores agree once more with the observed disentanglement patterns.

As further illustration, Fig. 8 displays traversals of the latent space, obtained by varying a subset of the latent variables while holding the remaining ones (including noise) constant. With these examples, we can qualitatively verify the quantitative results in Fig. 7. Note that, until now, visual inspection was typically the only means of evaluating disentanglement and expressiveness of the generative direction of image models (Kim and Mnih, 2018).

4.2.4. Diagnosing Failure

We also attempted to detect whether an InfoGAN had learned to discover that local perturbations (swelling and fractures) correspond to independent generative factors. To this end, we extended the model formulation with additional Bernoulli latent codes, which may learn to encode presence/absence of each type of local perturbation. The model investigated here, dubbed InfoGAN-C (cf. Table 2), had a 10-way categorical, two continuous and two binary codes, and was trained with a dataset of plain, swollen and fractured digits (randomly mixed as above).



(a) InfoGAN-A: one-dimensional traversals of c_1 (*top*, 'digit type') and c_3 (*bottom*, 'slant'). Samples in each row share the values of remaining latent variables and noise.

(b) InfoGAN-B: two-dimensional traversal of $c_4 \times c_3$ ('thickness' × 'slant')

Figure 8: InfoGAN latent space traversals

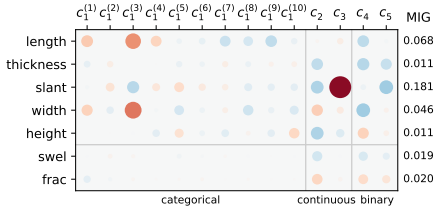


Figure 9: Partial correlations between inferred latent codes and morphometrics of test images for InfoGAN-C (overall MIG = 0.0507). 'swel' and 'frac' refer to the binary perturbation labels.

Again via inferential partial correlation analysis—now including ground-truth perturbation labels—we can quantitatively verify that this particular model instance was unable to meaningfully disentangle the perturbations (Fig. 9, bottom-right block). This is also confirmed by the low MIG for the corresponding rows. In fact, it appears that the addition of the binary variables did not lead to more expressive representations in this case, even impairing the disentanglement of the categorical variables, if compared to Figs. 6a and 6b, for example.

4.3. Predictive Model Evaluation

Although the driving motivation for introducing Morpho-MNIST has been the lack of means for quantitative evaluation of generative models, the proposed framework may also be a valuable resource in the context of predictive models. In this section we present several case studies to demonstrate how the proposed perturbations can be useful for systematically

Test data	kNN	SVM	MLP	CNN
PLAIN	96.25	95.71	97.97	98.95
LOCAL	95.22	92.47	93.15	95.33

Table 3: Example results on digit recognition (accuracy, %) under synthetic domain shift using the proposed data perturbations

evaluating predictive robustness under controlled domain shift and how morphometry enables an entirely new variety of experimental settings and nuanced analyses based on shape stratification. Further we demonstrate how the perturbation labels and morphometrics can serve as prediction targets in their own right for new supervised tasks. For simplicity and reproducibility, we use the same datasets as in the disentanglement experiments (Section 4.2): plain digits (PLAIN), plain mixed with thinned and thickened digits (GLOBAL), and plain mixed with swollen and fractured digits (LOCAL).

4.3.1. Robustness

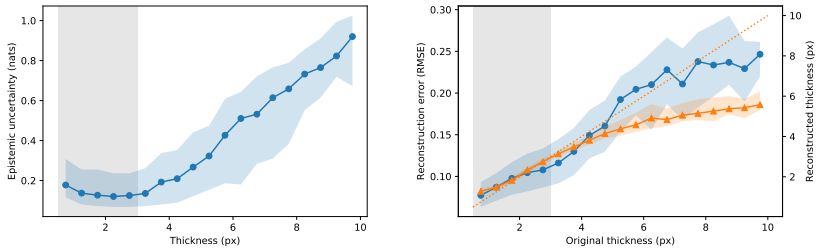
Synthetic perturbations enable fabrication of domain shift scenarios, aiming to systematically evaluate the robustness of predictive models to specific types of variation. For example, digits with local perturbations (swelling and fractures) are still clearly identifiable by humans (see Figs. B.3 and B.4). However, a practitioner might be interested in investigating how different types of classifiers react to these changes.

Here we compare simple representative methods from four classification paradigms: neighbourhood-based, kernel-based, fully connected, and convolutional. Specifically, we evaluated k-nearest-neighbours (kNN) using $k = 5$ neighbours and ℓ_1 distance weighting, a support vector machine (SVM) with polynomial kernel and penalty parameter $C = 100$, a multi-layer perceptron (MLP) with 784–200–200–L architecture (L: number of outputs), and a LeNet-5 (LeCun et al., 1998) convolutional neural network (CNN).

Each model is trained for digit recognition once on PLAIN, then tested separately on both PLAIN and LOCAL test datasets, to investigate the effect of *domain shift*. All methods suffer a drop in test accuracy on LOCAL (Table 3), and, considering that one third of the images in LOCAL are identical to ones in PLAIN, the absolute performance drops for the perturbed digits alone are in fact in the order of 1.5%, 4.9%, 7.2%, and 5.4%, respectively. kNN appears to be the most robust to local perturbations, conceivably because they affect only a few pixels, leaving the image distance between neighbours largely unchanged. Contrarily, both the global patterns that SVM and MLP rely on and the CNN's local filter responses may have changed considerably. Such analyses can provide new insights into the behaviour, strengths, and weaknesses of different models, and can be useful to inform new strategies for developing more robust approaches.

4.3.2. Data Stratification

Originally, MNIST could be stratified exclusively by digit class, as this was the only meta-information available. However, with morphometrics, data can be sliced and grouped



(a) Uncertainty of MC-dropout model

(b) Auto-encoder reconstruction error (left, ●) and thickness of reconstructed digits (right, ▲)

Figure 10: Results for toy outlier detection experiments showcasing data stratification by stroke thickness: training digits were restricted to those thinner than 3 px (shaded range), and test digits were grouped into 0.5 px-wide bins for analysis. Lines and shaded regions indicate medians and interquartile ranges.

according to entirely new criteria: we are able to select the shape properties we desire in the training set and also perform finer analyses at test time. It is even possible to combine multiple attributes for multifaceted studies.

To illustrate how these capabilities can be put to use, we present a simplified case study on a hypothetical unsupervised outlier detection scenario. We take the GLOBAL dataset, restrict the training data to digits whose thickness is below 3 px, and evaluate the model's response to test digits across the entire thickness range, especially out-of-distribution samples. Here we assume the true selection criterion is unknown at learning time. Two example frameworks were considered: uncertainty-based and reconstruction-based.

For the first setting, we assume a research question about outlier detection with respect to the digit classification task. We take a LeNet-5 classifier (LeCun et al., 1998) and employ Monte Carlo dropout (Kendall and Gal, 2017) to estimate the model's epistemic uncertainty via mutual information (Smith and Gal, 2018). This metric, which quantifies the model's uncertainty about its parameters when presented with a test input, has been shown to be effective at detecting out-of-distribution samples (Smith and Gal, 2018). Figure 10a shows that the model seems markedly more uncertain on average regarding thicker digits (e.g. above 5 px), although there is no abrupt jump in uncertainty at the inlier boundary. Since we have 'ground-truth' outlier labels for the test images based on their thickness, we are also able to evaluate the discriminative power of this score: with an area under ROC curve (AUC) of only 0.67, this approach did not result in a particularly strong outlier detector.

In the second setting, we explored outlier detection with a deep convolutional auto-encoder.[4] It was likewise trained on digits thinner than 3 px, and we plot in Fig. 10b its reconstruction error for test digits of varying thicknesses. We observe a steeper trend than

4. The encoder has three convolutional and one fully connected layers, and the mirrored decoder employs transposed convolutions. The convolutional layers have 3×3 filters with stride 2 and 32–64–128 feature channels, and the bottleneck layer is 10 units wide.

Prediction task	kNN	SVM	MLP	CNN
Abnormality detection (%)	65.10	77.59	88.25	97.53
Thickness regression (RMSE)	0.467	0.365	0.348	0.279

Table 4: Example results on some of the new MNIST prediction tasks

for the previous model's uncertainty (Fig. 10a), and in fact this auto-encoder's reconstruction error is far more discriminative (AUC = 0.85). In addition, it is also possible to collect morphometrics of the reconstructions and visualise to what extent the original shape attributes are preserved. Figure 10b suggests the interesting finding that the reconstructions of thick digits are consistently thinner than the originals, as if the auto-encoder 'projected' these digits towards the training subspace to some extent.

4.3.3. New Prediction Tasks

In the previous two sections, we demonstrated how perturbations may be leveraged to systematically evaluate robustness to domain shift and how morphometry can prove valuable for selecting training data and for performing stratified test-time analysis. Here we illustrate how perturbation labels and morphometric attributes can be used for interesting new prediction tasks on MNIST, evaluating the same models from Section 4.3.1.

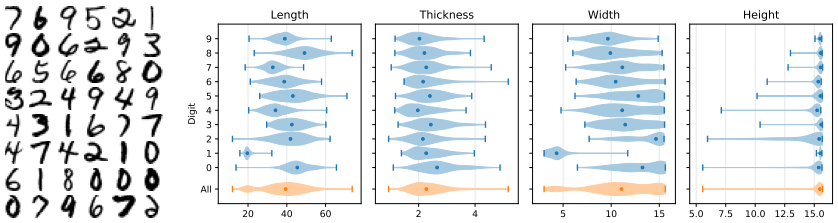
As our first example, we define a supervised abnormality detection task, using the LOCAL dataset, to predict whether a digit is normal or perturbed (swollen or fractured)—compare with lesion detection in medical scans. The top row of Table 4 indicates that the CNN was able to detect abnormalities with high accuracy, likely thanks to local invariances of its architecture, whereas the shallow classifiers, kNN and SVM, performed remarkably worse. Note how the performance is generally much lower than for digit classification, revealing the higher difficulty of this binary problem compared to the ten-class recognition task.

Finally, we also constructed a regression task for digit thickness using the GLOBAL dataset, mimicking medical imaging tasks such as estimating brain age from cortical grey matter maps. Since this is a non-trivial task requiring some awareness of local geometry, it is unsurprising that the convolutional model outperformed the others, which rely on holistic features (Table 4, bottom row).

5. Discussion and Perspectives

Through concrete examples, we demonstrated in Section 4 how morphometry and perturbations can be exploited 1) for qualitative and, more importantly, *quantitative* evaluation of generative models in terms of sample diversity and disentanglement, and 2) in developing novel experimental protocols for analysing predictive models. The presented methodologies are broadly applicable to datasets of rasterised shapes beyond MNIST.

For example, we apply our exact same morphometry pipeline to handwritten digits from the USPS dataset (LeCun et al., 1990), also composed of small grayscale images (16×16). Example images from this dataset and distributions of length, thickness, width, and height can be seen in Fig. 11. Slant was excluded because the USPS digits were already deslanted, with residual variation in the order of ±1°. Another popular dataset to which our morphometrics



(a) USPS digits (b) Distribution of morphological attributes for USPS test images

Figure 11: Application of the proposed morphometrics to the USPS dataset

can be directly applied is Omniglot (Lake et al., 2015), consisting of handwritten symbols from 50 different alphabets.

Besides, the general perspective of measuring attributes directly from images is evidently not limited to datasets of handwritten characters. Consider for example the dSprites dataset (Matthey et al., 2017), often used in disentanglement research, which contains 64×64 synthetic binary images of small shapes with ground-truth annotations. Attributes such as location and scale can easily be measured based on image moments (see Section 2.4). They are expected to be robust even for blurry generated samples. As discussed previously, while the annotations provided with such data facilitate evaluation of inferential disentanglement, these measurements enable the unprecedented analysis of generative behaviour.

An obvious limitation of morphometry is the tacit assumption that generated samples resemble true data closely enough that the shape measurements are indeed meaningful. What happens if the generative model under study is underfitted and produces nonsensical images? In such case, one expects the joint distribution of sample morphometrics to be strikingly different from that of true data, even if some individual metrics may have been extracted successfully (e.g. width and height of a reasonably sized blur). Like mode collapse, this issue can be diagnosed straightforwardly—for example with an analysis similar to that from Section 4.1—and may constitute an interesting finding by itself.

6. Conclusion

With Morpho-MNIST we provide a number of mechanisms to quantitatively assess representation learning with respect to measurable factors of variation in the data. We believe that this may prove an important asset for future research on generative models, and will hopefully also be useful in evaluating predictive models. We emphasise that the proposed morphometrics can be used *post hoc* to evaluate already trained models, potentially revealing novel insights about their behaviour. In addition, we have shown that our morphometry and perturbation framework easily extends to datasets other than MNIST. We hope that the case studies presented here may also invite research into developing metrics for other kinds of images beyond rasterised 2D shapes or even to non-imaging data, paving the way to more widespread analysis of true generative performance.

Acknowledgments

This work was supported by CAPES, Brazil (BEX 1500/2015-05) and received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 757173, project MIRA, ERC-2017-STG).

Appendix A. Morphometrics of Plain and Perturbed Datasets

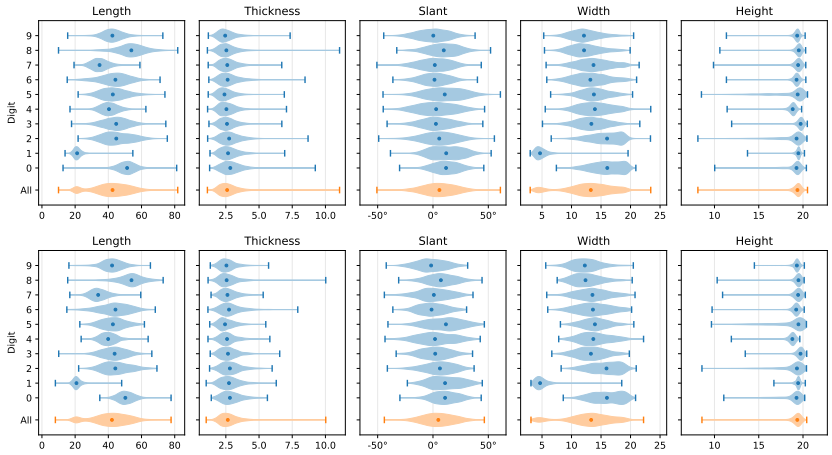


Figure A.1: Distribution of morphological attributes for plain MNIST digits. *Top:* training set; *bottom:* test set.

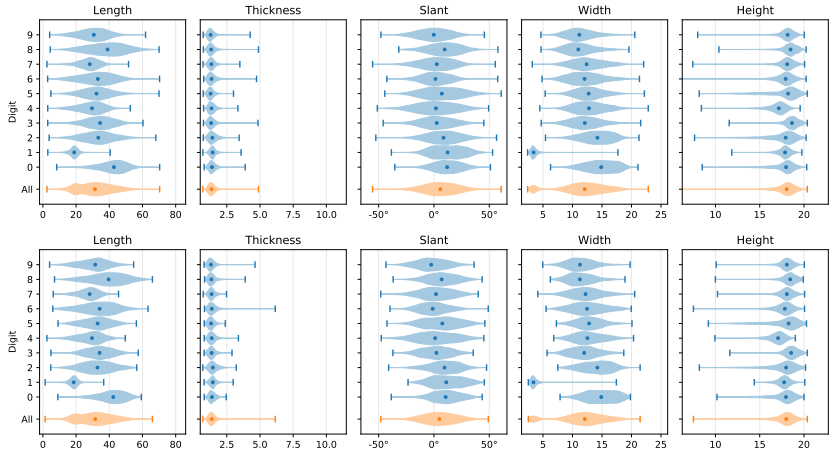


Figure A.2: Distribution of morphological attributes for thinned MNIST digits. *Top:* training set; *bottom:* test set.

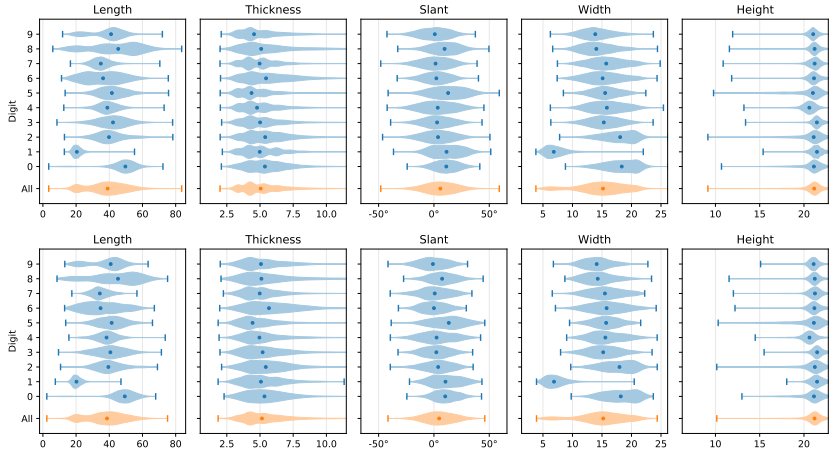


Figure A.3: Distribution of morphological attributes for thickened MNIST digits. *Top:* training set; *bottom:* test set.

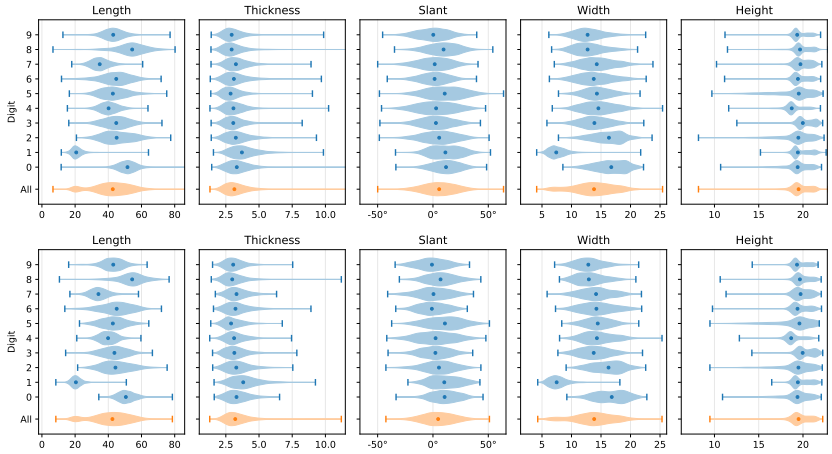


Figure A.4: Distribution of morphological attributes for swollen MNIST digits. *Top:* training set; *bottom:* test set.

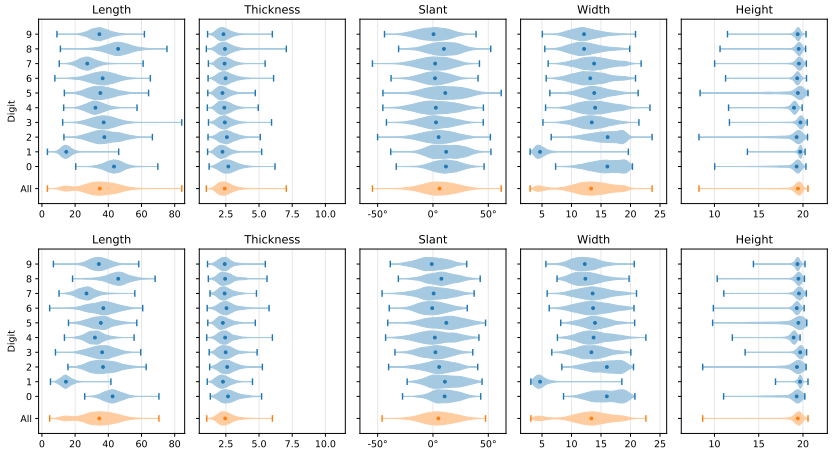


Figure A.5: Distribution of morphological attributes for fractured MNIST digits. *Top:* training set; *bottom:* test set.

Appendix B. Perturbation Examples

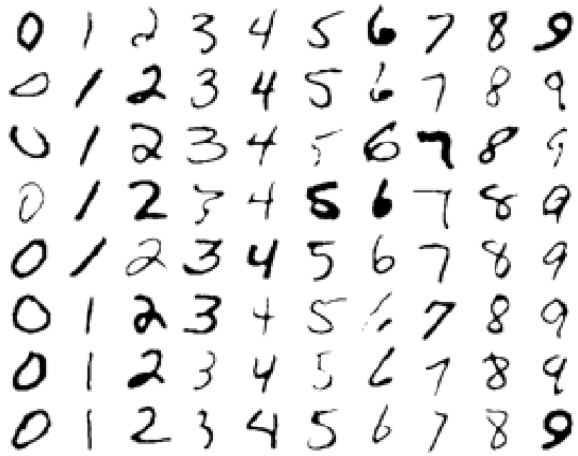


Figure B.1: Examples of globally thinned digits

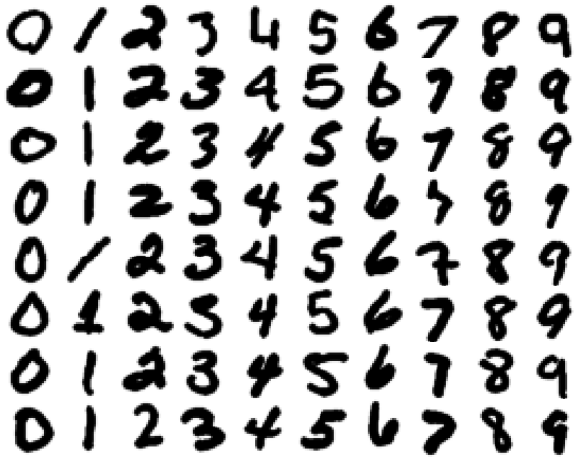


Figure B.2: Examples of globally thickened digits

Figure B.3: Examples of digits with local swelling

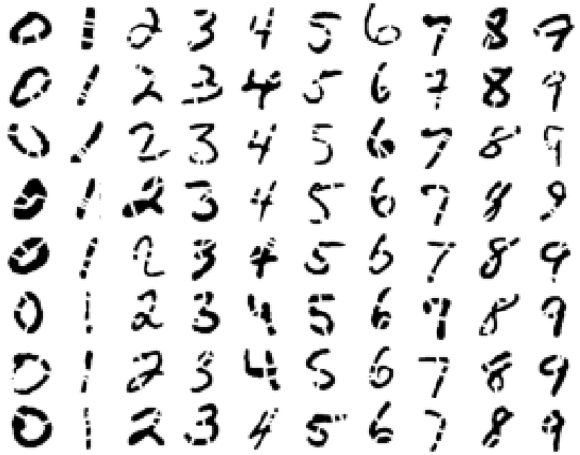


Figure B.4: Examples of digits with local fractures

Appendix C. MMD Details

We employed a Gaussian product kernel with bandwidths derived from Scott's rule, analogously to the KDE plots in Fig. 5. Scott's rule of thumb defines the bandwidth for a density estimation kernel as $N^{-1/(D+4)}$ times the standard deviation in each dimension, where N and D denote sample size and number of dimensions (Scott, 1992, Eq. 6.42). We determine the KDE bandwidths separately for real and sample data, then add their squares to obtain the squared bandwidth of the MMD's Gaussian kernel, as it corresponds to the *convolution* of the density estimation kernels chosen for each set of data. See Gretton et al. (2012, §3.3.1) for further details on the relation between MMD and L_2 distance of kernel density estimates.

Whereas the bandwidth heuristic used here is fairly crude, much more sophisticated kernel selection procedures exist, e.g. by explicitly optimising the test power (Sutherland et al., 2017). A further analysis tool in a similar vein would be to apply a *relative* MMD similarity test (Bounliphone et al., 2016), to rank trained models based on sample fidelity. It would also be possible to adopt a model criticism methodology based on the MMD witness function (Lloyd and Ghahramani, 2015), to identify over- and under-represented regions in morphometric space (and corresponding generated image exemplars could also be inspected).

References

Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? Some theory and empirics. In *International Conference on Learning Representations (ICLR 2018)*, 2018.

Mathieu Aubry, Daniel Maturana, Alexei A. Efros, Bryan C. Russell, and Josef Sivic. Seeing 3D chairs: exemplar part-based 2D–3D alignment using a large dataset of CAD models. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, pages 3762–3769. IEEE, 2014. Dataset URL `https://www.di.ens.fr/willow/research/seeing3Dchairs/`.

Mikołaj Bińkowski, Dougal Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR 2018)*, 2018.

Harry Blum. A transformation for extracting new descriptors of shape. In *Models for the perception of speech and visual form*, pages 362–380. MIT Press, 1967.

Wacha Bounliphone, Eugene Belilovsky, Matthew B. Blaschko, Ioannis Antonoglou, and Arthur Gretton. A test of relative similarity for model selection in generative models. In *International Conference on Learning Representations (ICLR 2016)*, 2016.

Yaroslav Bulatov. notMNIST dataset, sep 2011. URL `https://yaroslavvb.blogspot.co.uk/2011/09/notmnist-dataset.html`.

Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pages 2610–2620, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pages 2172–2180, 2016.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters, 2017. arXiv:1702.05373. Dataset URL `https://www.nist.gov/itl/iad/image-group/emnist-dataset`.

Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pages 708–718, 2018.

Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations (ICLR 2018)*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 2672–2680, 2014.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Patrick J. Grother and Kayee K. Hanaoka. NIST Special Database 19. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2016. Dataset URL `https://www.nist.gov/srd/nist-special-database-19`.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 6626–6637, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR 2017)*, 2017.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5574–5584, 2017.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, volume 80 of *PMLR*, pages 2649–2658. PMLR, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR 2014)*, 2014.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations (ICLR 2018)*, 2018.

Brenden M. Lake, Ruslan R. Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050. Dataset URL `https://github.com/brendenlake/omniglot`.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pages 473–480, 2007. doi: 10.1145/1273496.1273556. Dataset URL `https://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/MnistVariations`.

Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2 (NIPS 1989)*, pages 396–404, 1990. Dataset URL `https://web.stanford.edu/~hastie/ElemStatLearn/data.html`.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. Dataset URL `http://yann.lecun.com/exdb/mnist/`.

James R. Lloyd and Zoubin Ghahramani. Statistical model criticism using kernel two sample tests. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 829–837, 2015.

Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA, USA, 2007. Dataset URL `http://leon.bottou.org/projects/infimnist`.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. In *International Conference on Learning Representations Workshop (ICLRW 2016)*, 2016.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing sprites dataset, 2017. URL `https://github.com/deepmind/dsprites-dataset/`.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *Proceedings of the Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2009)*, pages 296–301. IEEE, 2009. doi: 10.1109/AVSS.2009.58. Dataset URL `https://faces.dmi.unibas.ch/bfm/index.php?nav=1-1-1&id=scans`.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, pages 2234–2242, 2016.

David W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. John Wiley & Sons, Inc., New York, NY, USA, 1992.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI 2018)*, pages 560–569. AUAI Press, 2018.

Dougal J. Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alexander J. Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *International Conference on Learning Representations (ICLR 2017)*, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 2818–2826. IEEE, 2016. doi: 10.1109/CVPR.2016.308.

Loo-Nin Teow and Kia-Fock Loe. Robust vision-based features and classification schemes for off-line handwritten digit recognition. *Pattern Recognition*, 35(11):2355–2364, 2002. doi: 10.1016/S0031-3203(01)00228-X.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR 2016)*, 2016.

Tijmen Tieleman. affNIST, aug 2013. URL `https://www.cs.toronto.edu/~tijmen/affNIST/`.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014. doi: 10.7717/peerj.453.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017. arXiv:1708.07747. Dataset URL `https://github.com/zalandoresearch/fashion-mnist`.

Chhavi Yadav and Léon Bottou. Cold case: The lost MNIST digits, 2019. arXiv:1905.10498. Dataset URL `https://github.com/facebookresearch/qmnist`.

Journal of Machine Learning Research 20 (2019) 1-62 Submitted 2/19; Revised 8/19; Published 11/19

Differentiable reservoir computing

Lyudmila Grigoryeva LYUDMILA.GRIGORYEVA@UNI-KONSTANZ.DE
Department of Mathematics and Statistics
Graduate School of Decision Sciences
Universität Konstanz
Germany

Juan-Pablo Ortega JUAN-PABLO.ORTEGA@UNISG.CH
Faculty of Mathematics and Statistics
Universität Sankt Gallen
Switzerland
Centre National de la Recherche Scientifique (CNRS)
France

Editor: Sayan Mukherjee

Abstract

Numerous results in learning and approximation theory have evidenced the importance of differentiability at the time of countering the curse of dimensionality. In the context of reservoir computing, much effort has been devoted in the last two decades to characterize the situations in which systems of this type exhibit the so-called echo state (ESP) and fading memory (FMP) properties. These important features amount, in mathematical terms, to the existence and continuity of global reservoir system solutions. That research is complemented in this paper with the characterization of the differentiability of reservoir filters for very general classes of discrete-time deterministic inputs. This constitutes a novel strong contribution to the long line of research on the ESP and the FMP and, in particular, links to existing research on the input-dependence of the ESP. Differentiability has been shown in the literature to be a key feature in the learning of attractors of chaotic dynamical systems. A Volterra-type series representation for reservoir filters with semi-infinite discrete-time inputs is constructed in the analytic case using Taylor's theorem and corresponding approximation bounds are provided. Finally, it is shown as a corollary of these results that any fading memory filter can be uniformly approximated by a finite Volterra series with finite memory.

Keywords: reservoir computing, fading memory property, finite memory, echo state property, differentiable reservoir filter, Volterra series representation, state-space systems, system identification, machine learning.

1. Introduction

Context and preliminary discussion. Reservoir computing (RC) is a neural approach to the learning of dynamic processes which advocates the use of paradigms in which the supervised estimation of all available interconnection weights is not necessary and only the training of a static memoryless readout suffices to obtain good performances. This computational strategy has been simultaneously inspired by ideas coming from three different fields, namely, recurrent neural networks, dynamical systems, and biologically inspired neural microcircuits. The common thread to these analyses is the use of rich dynamics to process information and to create memory traces. This explains why RC it can be found in the literature under other denominations like ***Liquid State Machines*** Maass and Sontag (2000); Maass et al. (2002); Natschläger et al. (2002); Maass et al. (2004, 2007) and is represented by various

©2019 Lyudmila Grigoryeva and Juan-Pablo Ortega.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-150.html.

learning paradigms, being the ***Echo State Networks*** introduced in Jaeger (2010); Jaeger and Haas (2004) a particularly important example.

RC has shown superior performance in many forecasting and classification engineering tasks (see Lukoševičius and Jaeger (2009) and references therein) and has shown unprecedented abilities in the learning of the attractors of complex nonlinear infinite dimensional dynamical systems Jaeger and Haas (2004); Pathak et al. (2017, 2018); Lu et al. (2018). Additionally, RC implementations with dedicated hardware have been designed and built (see, for instance, Appeltant et al. (2011); Rodan and Tino (2011); Vandoorne et al. (2011); Larger et al. (2012); Paquot et al. (2012); Brunner et al. (2013); Vandoorne et al. (2014); Vinckier et al. (2015); Laporte et al. (2018); Tanaka et al. (2019)) that exhibit information processing speeds that largely outperform standard Turing-type computers.

Ever since the inception of this methodology, much effort has been devoted to identify the features that make a RC system capable of retaining relevant memory traces of the inputs and computationally powerful. The first question has given rise to various notions and computational schemes for the memory capacity of RC systems Jaeger (2002); White et al. (2004); Ganguli et al. (2008); Hermans and Schrauwen (2010); Dambre et al. (2012); Grigoryeva et al. (2015); Couillet et al. (2016); Grigoryeva et al. (2016); Tio (2018). Another strand of interesting literature that we will not explore in this work has to do with the Turing computability capabilities of the systems of the type that we just introduced; recent relevant works in this direction are Kilian and Siegelmann (1996); Siegelmann et al. (1997); Cabessa and Villa (2015, 2016), and references therein.

Regarding computational power, there are three properties that pervade the literature and that are usually declared as necessary to obtain an adequate functioning in a RC system (see, for instance, Legenstein and Maass (2007); Lukoševičius and Jaeger (2009); Maass (2011) and references therein), namely, the ***fading memory property (FMP)***, the ***echo state property (ESP)***, and the pairwise ***separation property (SP)***. The FMP is a notion observed in many modeling situations in which the influence of the input gradually fades out in time. This property is repeatedly invoked in systems theory Volterra (1930); Wiener (1958), computational neurosciences Maass et al. (2004), physics Coleman and Mizel (1968), or mechanics (see Fabrizio et al. (2010) and references therein). The ESP Jaeger (2010); Yildiz et al. (2012); Manjunath and Jaeger (2013) is an existence and uniqueness property for the solutions of a state-space system that guarantees that the past history of the input fully determines the state of the system at any given point in time. Finally, the SP is satisfied by an input/output system if for any two input time series which differed in the past, the network assumes at subsequent time points different states.

Even though these three properties are an essential part of the "RC jargon", it is not always clear in the literature why they are important. A partial answer to this question has been given in the development of universality theorems for RC machine learning paradigms. Indeed, it has been shown in Maass and Sontag (2000); Maass et al. (2002, 2004, 2007); Grigoryeva and Ortega (2018a,b) that various families of RC systems that have these three properties are uniform universal approximants in a dynamical context in the presence of uniformly bounded (respectively, almost surely uniformly bounded) deterministic (respectively, stochastic) inputs. Moreover, these properties are exactly what is needed to prove universality statements using the Stone-Weierstrass theorem. Nevertheless, it has also been shown Gonon and Ortega (2018) that when the uniform approximation criterion is replaced by a L^p norm defined with the measure induced by the input stochastic process, then the FMP does not play any role anymore.

Additionally, when these properties are invoked, it is not always clear what the actual definition that is being used is and they are even used exchangeably sometimes. The reason for this confusion is that, *in the presence of various compactness and contractivity hypotheses, the ESP and the FMP are automatically simultaneously satisfied.* Moreover, the same entanglement occurs when it comes to the actual dynamical implications that these properties entail like the input and state forgetting properties (see later on in the text for detailed definitions).

From a learning theoretical perspective, the connections that we just brought up between these dynamical properties (FMP, ESP, and SP) and universality can be rephrased by saying that families that exhibit them are capable of making the approximation error in a learning task as small as desired. Also in the approximation error context, classical results in static setups (see, for instance, Girosi and Anzellotti (1992); Girosi (1995)) show that the differentiability of the objects that need to be approximated is as beneficial for convergence rates as the dimensionality of the input is detrimental. This feature is sometimes referred to as the *blessing of smoothness*, as opposed to the *curse of dimensionality*. Differentiability is hence a crucial element in the understanding of the learning theoretical properties of most machine learning paradigms and, as far as we know, has never been tackled in the reservoir computing context and that is at the core of this paper.

Important existing results. In order to make these remarks explicit, we recall here some results that will help us later on to introduce the contributions in this paper. Consider the discrete-time nonlinear state-space transformation

$$\begin{cases} \mathbf{x}_t = F(\mathbf{x}_{t-1}, \mathbf{z}_t), & (1.1) \\ \mathbf{y}_t = h(\mathbf{x}_t). & (1.2) \end{cases}$$

In the context of supervised machine learning we will refer to these transformations as ***reservoir systems*** and we will think of them as special types of recurrent neural networks. In that setup, the map $F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$, $n, N \in \mathbb{N}^+$, is called the ***reservoir***, it is usually randomly generated and $h : \mathbb{R}^N \to \mathbb{R}^d$ is the ***readout***, which is estimated via a supervised learning procedure. The ***input*** in this system is given by the elements of the infinite sequence $\mathbf{z} = (\ldots, \mathbf{z}_{-1}, \mathbf{z}_0, \mathbf{z}_1, \ldots) \in (\mathbb{R}^n)^{\mathbb{Z}}$ and the ***output*** by the components of $\mathbf{y} \in (\mathbb{R}^d)^{\mathbb{Z}}$. Given that the state-space may need to be high-dimensional in order to exhibit adequate approximation properties, it is desirable that the readout is as simple as possible (linear or polynomial, for instance). In this direction, various families of reservoir systems with linear readouts like Echo State Networks Jaeger and Haas (2004) or State Affine Systems (see later on in the text) have been shown to have universal approximation properties Grigoryeva and Ortega (2018a,b); Gonon and Ortega (2018). Training for these systems reduces to the solution of a linear regression problem (eventually regularized) when the mean square error is used as loss function.

We say that the reservoir system (1.1)-(1.2) satisfies the ***echo state property (ESP)*** when for any $\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}}$ there exists a unique $\mathbf{y} \in (\mathbb{R}^d)^{\mathbb{Z}}$ that satisfies (1.1). When this existence and uniqueness feature is available one can associate well-defined filters $U^F : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^N)^{\mathbb{Z}}$ and $U_h^F : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}_-}$ to the reservoir map F and the reservoir system (1.1)-(1.2), respectively.

Very general situations have been characterized in which the ESP holds. For example, suppose that we restrict ourselves to inputs that are uniformly bounded by a constant $M > 0$, that is, consider the space K_M of semi-infinite sequences given by

$$K_M := \left\{ \mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-} \mid \|\mathbf{z}_t\| \leq M \quad \text{for all} \quad t \in \mathbb{Z}_- \right\}, \quad M > 0, \tag{1.3}$$

and assume that the reservoir map F is continuous and a contraction on the first entry that maps $F : \overline{B_{\|\cdot\|}(\mathbf{0}, L)} \times \overline{B_{\|\cdot\|}(\mathbf{0}, M)} \longrightarrow \overline{B_{\|\cdot\|}(\mathbf{0}, L)}$, with $L > 0$ (the symbol $\overline{B_{\|\cdot\|}(\mathbf{v}, r)}$ denotes the closure of the open ball $B_{\|\cdot\|}(\mathbf{v}, r)$ with respect to a given norm $\|\cdot\|$, center $\mathbf{v}$, and radius $r > 0$). In that case, it can be shown (see, for instance, (Grigoryeva and Ortega, 2018b, Theorem 3.1)) that for any $\mathbf{z} \in K_M$ there exists a unique $\mathbf{x} \in K_L := \{\mathbf{x} \in \left(\mathbb{R}^N\right)^{\mathbb{Z}_-} \mid \|\mathbf{x}_t\| \leq L \quad \text{for all} \quad t \in \mathbb{Z}_-\}$ that satisfies (1.1), that is, the ESP holds. This fact allows us to associate a unique filter $U^F : K_M \longrightarrow K_L$ to the reservoir map F and $U_h^F : K_M \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}_-}$ to the reservoir system (1.1)-(1.2), respectively, with $U_h^F := h \circ U^F$.

Moreover, in this situation (see again (Grigoryeva and Ortega, 2018b, Theorem 3.1)) the continuity of F and h implies that both U^F and U_h^F are continuous when we consider either the uniform or the product topologies in the domain and target spaces. The continuity with respect to the product topology is called in this setup the ***fading memory property (FMP)*** and, as we shall see below, can

be characterized using weighted norms in the spaces of input and output sequences, which shows that recent inputs are more represented in the outputs of FMP filters than older ones. Equivalently, the outputs produced by FMP filters associated to inputs that are close in the recent past are close, even when those inputs may be very different in the distant past.

The restriction to uniformly bounded inputs of the type (1.3) when using contracting reservoir maps does not only make the ESP and the FMP to simultaneously hold but it also simplifies enormously the characterization of the FMP. Indeed, it has been shown in Sandberg (2003); Grigoryeva and Ortega (2018b) that in that case the fading memory property is not a metric but an exclusively topological property that does not depend on the weighted norm used to define it. Therefore, the FMP does not contain in that situation any information about the rate at which the dependence on the past inputs in the system output declines. This is not the case anymore when we consider unbounded input sets since, as we show later on in Theorem 7, *reservoir systems have the FMP only with respect to weighting sequences that converge to zero sufficiently fast and at a rate that is related to the contracting properties of the reservoir map.*

There are important connections between the notions and the results that we just reviewed and fundamental concepts in the theories of non-autonomous and of random dynamical systems. Even though we shall not pursuit that line of thought, the reader is encouraged to check with Arnold (1998); Kloeden (2003); Kloeden and Rasmussen (2010); Manjunath and Jaeger (2014); Newman (2018) and references therein for in-depth presentations.

Main contributions of the paper. The *core contributions of this paper are, first, the analysis of the ESP and the FMP in the absence of boundedness hypotheses and, second, the extension of the FMP-related continuity statements in the literature to the study of the differentiability properties of reservoir computers.* In particular, we aim at characterizing the situations in which one can obtain the differentiability of reservoir filters out of the differentiability properties of the maps that define the corresponding reservoir system.

Regarding the first objective, there are several reasons to study reservoir computing systems with unbounded inputs. First, even though we only deal in this paper with the deterministic setup, any random component in the data generating process of the inputs, like a Gaussian perturbation, would imply unboundedness. Second, when dealing with reservoir systems associated to physical systems, it is certainly reasonable to assume boundedness in the input due to the saturation effects that most of those systems present. Nevertheless, the value of the bounding constant is in general unknown beforehand, which makes uniform boundedness hypotheses unrealistic. Finally, in the study of the differentiability properties of reservoir computers, the differentiability of Fréchet type is only defined on open subsets of normed spaces. We shall see that any open set in the Banach space of inputs with a weighted norm contains unbounded sequences, which forces us to deal with that situation.

As to the analysis of the differentiability properties of reservoir systems, this is an important question for several reasons:

- It has been shown (see, for instance, Girosi and Anzellotti (1992); Girosi (1995)) that differentiability is a key element in decreasing the complexity that is needed at the time of approximating a function with a prescribed accuracy level. The influence of this feature is comparable to that of the dimensionality of the input. Even though the development of bounds for the approximation error in the RC context is the subject of a forthcoming paper, it is reasonable to presume that differentiability is a crucial element in the understanding of the learning theoretical properties of this type of machine learning paradigms.

- RC applications to the learning of the attractors of chaotic deterministic dynamical systems have been shown (see Lu et al. (2018)) to be much related with the notion of *Generalized Synchronization* Kocarev and Parlitz (1995, 1996) for which differentiability is a relevant feature Hunt et al. (1997).

Indeed, in the absence of differentiability, the synchronization mapping may be "wild" enough (in the terminology of Hunt et al. (1997)) to create a gap between the information dimensions of the attractors of the input system and the system used to learn it. Additionally, one of the standard techniques to assess the quality of the result of this learning task is the comparison of the Lyapunov spectra of the problem system and the learnt proxy. These spectra are only available in the presence of differentiability.

- Also in the context of the learning of dynamical systems, differential topological arguments have been used Hart et al. (2019) to establish Takens-type embedding results for Echo State Networks. This is an important result that justifies the forecasting abilities of RC that are empirically observed in this framework.

- When filters are analytic, they obviously admit a Taylor series expansion which coincides with the so-called discrete-time Volterra series representation Volterra (1930); Schetzen (1980); Rugh (1981); Priestley (1988) and, moreover, different Taylor remainders can be used to provide bounds on the approximation errors that are committed when those series are truncated. This path has been explicitly explored in Sandberg (1998a, 1999) for analytic filters with respect to the supremum norm and with inputs with a finite past. We extend this work and we characterize the inputs for which an analytic fading memory reservoir filter with respect to a weighted norm admits a Volterra series representation with semi-infinite inputs. Additionally, we can use the causality and time-invariance hypotheses to show that the corresponding Volterra series representations have time-independent coefficients (this feature is not available in the case studied in Sandberg (1999)) that automatically satisfy the convergence conditions spelled out in Sandberg (1998b,c). The availability of this series representation has important learning theoretical consequences since, as we shall show in a forthcoming publication, implies that any analytic filter can be represented as a reservoir filter with linear readouts and where the reservoir map has been randomly generated using a well-specified distribution. This result is a corollary of the Volterra series representation presented later on in the paper combined with an adequately chosen version of the Johnson-Lindenstrauss Lemma. In a continuous-time setup the construction involves the use of the so-called *signature process* designed in Rough Path Theory.

- These statements can be combined with the results in Grigoryeva and Ortega (2018a) to provide an alternative proof of the following Volterra series universality theorem that was stated for the first time in (Boyd and Chua, 1985, Theorems 3 and 4): *any time-invariant and causal fading memory filter can be uniformly approximated by a finite Volterra series with finite memory.*

- The local nature of the differential allows the formulation of conditions that ensure both the local and global existence of differentiable and, in passing, fading memory solutions. These conditions are a novel strong contribution to the long line of research on the ESP and the FMP and, in particular, link to existing research Manjunath and Jaeger (2013) on the input-dependence of the echo state property.

- The metric nature of the differential allows us to measure the speed at which fading memory filters forget inputs. As we see later on in Theorem 26, we are able to characterize this important piece of information with the differentiability property.

Organization of the paper. The paper is organized as follows:

- The introductory Section 2 presents the causal and time-invariant filters and functionals that are at the center of this paper. In Section 2.1 the Banach sequence spaces where the semi-infinite inputs and outputs of the reservoir systems that we study are defined. Various elementary facts about weighted and supremum norms are stated. In Section 2.2 the notions of fading memory, continuity,

and differentiability of maps between sequence spaces are carefully introduced. Section 2.3 focuses on causal and time-invariant filters defined on the sequence spaces introduced in Section 2.1. Those results are put to work in Section 2.4 to easily show well-known results that link the continuity of a filter with input and output spaces endowed with weighted norms with its asymptotic independence on the remote past input.

- Starting from Section 3 the paper focuses on reservoir filters. The main result in this section is Theorem 7 that provides a sufficient (but not necessary) condition for the ESP and FMP to hold in the presence of inputs that are not necessarily bounded. This is a significant generalization with respect to the "standard compactness conditions" imposed in Jaeger (2010) or the uniform boundedness in the inputs that was required in similar results in, for instance, Grigoryeva and Ortega (2018b). An important observation in Theorem 7 is that for general inputs, the FMP depends on the weighting sequence that is used to define it and establishes that, roughly speaking, *reservoir systems have the FMP only with respect to weighting sequences that converge to zero sufficiently fast and at a rate that is related to the contracting properties of the reservoir map.* This newly introduced FMP condition is spelled out for several widely used families of reservoir systems. The above mentioned results involving uniform boundedness hypotheses can be obtained as a corollary (see Corollary 10) of the results in this section. Another statement that we prove (see Theorem 12) is that when the target of the reservoir map is a compact set then the echo state property is in that situation guaranteed *for no matter what input*, even though the FMP may obviously not hold in that case.

- Section 4 is the core of the paper and studies the differentiability properties of reservoir filters determined by differentiable reservoir maps. The main results are contained in Theorems 14 and 19. The first theorem provides an explicit and easy-to-verify sufficient condition for the ESP and the FMP to hold around a given input for which we know that the reservoir system associated to a differentiable reservoir map has a solution. Theorem 19 is a global extension of the previous result that, unlike Theorems 7 and 14, fully characterizes the ESP and the differentiability (and hence the FMP) of the reservoir filter associated to a differentiable reservoir map. In Section 4.2 we show that the global conditions in Theorem 19 are much stronger than the local ones in Theorem 14 by introducing an example that shows how *the ESP and the FMP are structural features of a reservoir system when considered globally but are mostly input dependent when considered only locally.* This important observation has already been noticed in Manjunath and Jaeger (2013) where, using tools coming from the theory of non-autonomous dynamical systems, sufficient conditions have been formulated (see, for instance, (Manjunath and Jaeger, 2013, Theorem 2)) that ensure the ESP in connection to a given specific input. The differentiability conditions that we impose to our reservoir systems allow us to draw similar conclusions and, additionally, to automatically establish the FMP of the resulting locally defined reservoir filters. In Section 4.3 we show how for globally differentiable reservoir filters we can formulate a non-uniform version of the well-known input forgetting property for FMP filters that we recovered in Section 2.3 for inputs that are not necessarily bounded. Moreover, a novel uniform differential version of that result is provided in Theorem 26.

- Section 5 contains two main results. First, Theorem 29 shows the availability of discrete-time Volterra series representations for analytic, causal, time-invariant, and FMP filters. This result extends a similar statement formulated in Sandberg (1998a, 1999) to inputs with a semi-infinite past that are not necessarily bounded. Second, in Theorem 31, we combine the previous result with a universality statement in Grigoryeva and Ortega (2018a) to provide an alternative proof of the Volterra series universality theorem stated for the first time in (Boyd and Chua, 1985, Theorems 3 and 4).

The proofs of most results are provided in the appendices at the end of the paper.

2. Causal and time-invariant input/output systems

2.1. The input and output spaces

This paper studies input/output systems that are causal, that is, the output depends only on the past history of the input and that, in general, have infinite memory. This makes us consider the spaces of left-infinite sequences with values in $\mathbb{R}^n$, that is, $(\mathbb{R}^n)^{\mathbb{Z}_-} = \{\mathbf{z} = (\ldots, \mathbf{z}_{-2}, \mathbf{z}_{-1}, \mathbf{z}_0) \mid \mathbf{z}_i \in \mathbb{R}^n, i \in \mathbb{Z}_-\}$. Analogously, $(D_n)^{\mathbb{Z}_-}$ stands for the space of semi-infinite sequences with elements in the subset $D_n \subset \mathbb{R}^n$. The space $\mathbb{R}^n$ will be considered as a normed space with a norm denoted by $\|\cdot\|$ which is not necessarily the Euclidean one (even though they are all equivalent), unless it is explicitly mentioned.

We endow these infinite product spaces with the Banach space structures associated to one of the following two norms. First, the ***supremum norm*** $\|\mathbf{z}\|_\infty := \sup_{t\in\mathbb{Z}_-}\{\|\mathbf{z}_t\|\}$. The symbol $\ell^\infty_-(\mathbb{R}^n)$ is used to denote the Banach space formed by the elements that have a finite supremum norm. Second, given a strictly decreasing sequence with zero limit $w : \mathbb{N} \longrightarrow (0, 1]$ and that $w_0 = 1$, we define the ***weighted norm*** $\|\cdot\|_w$ on $(\mathbb{R}^n)^{\mathbb{Z}_-}$ associated to w by $\|\mathbf{z}\|_w := \sup_{t\in\mathbb{Z}_-}\{\|\mathbf{z}_t w_{-t}\|\}$. It can be shown (see Grigoryeva and Ortega (2018b)) that the set $\ell^w_-(\mathbb{R}^n)$ formed by the elements that have a finite w-weighted norm is a Banach space. Moreover, it is easy to show that $\|\mathbf{z}\|_w \leq \|\mathbf{z}\|_\infty$, for all $\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}$. This implies that $\ell^\infty_-(\mathbb{R}^n) \subset \ell^w_-(\mathbb{R}^n)$ and that the inclusion map $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty) \hookrightarrow (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is continuous.

The Banach spaces $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$ and $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ are particular cases of weighted Banach sequence spaces $(\ell^{p,w}_-(\mathbb{R}^n), \|\cdot\|_{p,w})$ where

$$\|\mathbf{z}\|_{p,w} := \left(\sum_{t\in\mathbb{Z}_-} \|\mathbf{z}_t\|^p w_{-t}\right)^{\frac{1}{p}}, \quad \text{with } 1 \leq p < +\infty,\ \mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}, \text{ and } w \text{ a sequence.} \tag{2.1}$$

When $p = +\infty$ we set $\|\cdot\|_{p,w} := \|\cdot\|_w$. We then define

$$\ell^{p,w}_-(\mathbb{R}^n) := \left\{\mathbf{z} \in \mathbb{R}^n \mid \|\mathbf{z}\|_{p,w} < +\infty\right\}. \tag{2.2}$$

These spaces are defined in the literature (see, for instance, Rekic-Vukovic et al. (2015); Gunawan et al. (2015)) without the requirement that w is a weighting sequence in the sense of the definition above. Indeed, the standard Banach spaces $(\ell^p_-(\mathbb{R}^n), \|\cdot\|_p)$, with $1 \leq p \leq +\infty$, are particular cases of $(\ell^{p,w}_-(\mathbb{R}^n), \|\cdot\|_{p,w})$ that are obtained by taking as sequence w the constant sequence w^ι given by $w^\iota_t := 1$, for all $t \in \mathbb{N}$. This observation is used in the paper to obtain many results for the spaces $\ell^\infty_-(\mathbb{R}^n)$ as a particular case of those proved for $\ell^w_-(\mathbb{R}^n)$.

We emphasize that w^ι is not a weighting sequence and that the spaces $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ considered in this paper are all based on sequences w of weighting type. It can be proved (see (Rekic-Vukovic et al., 2015, Theorems 3.3 and 4.1 and Corollary 4.1)) that, in that case:

$$\ell^{p,w}_-(\mathbb{R}^n) \subset \ell^w_-(\mathbb{R}^n), \quad \text{for any } 1 \leq p < +\infty, \tag{2.3}$$

and that,

$$\ell^p_-(\mathbb{R}^n) \subsetneq \ell^{p,w}_-(\mathbb{R}^n), \quad \text{for any } 1 \leq p \leq +\infty. \tag{2.4}$$

All the results in this paper are formulated for the weighted spaces $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ even though many of the statements that we provide are also valid for $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$ and $(\ell^{p,w}_-(\mathbb{R}^n), \|\cdot\|_{p,w})$. That will be explicitly pointed out in the statements or in remarks when it is the case. The Appendix 6.1 contains a collection of results regarding the topologies induced by weighted and supremum norms.

2.2. FMP, continuity, and differentiability of maps on infinite sequence spaces

Much of this paper is related to the continuity and the differentiability of maps of the type $f : W \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow V \subset \ell^{w^2}_-(\mathbb{R}^N)$, with w^1, w^2 weighting sequences and W and V subsets of $\ell^{w^1}_-(\mathbb{R}^n)$ and $\ell^{w^2}_-(\mathbb{R}^N)$, respectively, that in the case of differentiable maps are necessarily open. Maps that are continuous with respect to topologies generated by weighted norms will be generically referred to as ***fading memory maps*** (or we say that they have the ***fading memory property (FMP)***) while when the topology considered is generated by the supremum norm, we just say that the map is ***continuous***. Most of the definitions that we provide in what follows for the weighted norms case can be adapted to the supremum norm case by replacing the weighting sequences by the constant sequence w^ι given by $w^\iota_t := 1$, for all $t \in \mathbb{N}$.

Suppose now that W and V are open subsets. The map $f : W \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow V \subset \ell^{w^2}_-(\mathbb{R}^N)$ is ***(Fréchet) differentiable*** at $\mathbf{u}_0 \in W$ when there exists a bounded linear map $Df(\mathbf{u}_0) : \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow \ell^{w^2}_-(\mathbb{R}^N)$ that satisfies

$$\lim_{\mathbf{u}\to\mathbf{u}_0} \frac{f(\mathbf{u}) - f(\mathbf{u}_0) - Df(\mathbf{u}_0)\cdot(\mathbf{u} - \mathbf{u}_0)}{\|\mathbf{u} - \mathbf{u}_0\|_{w^1}} = \mathbf{0}. \tag{2.5}$$

We say that $f : W \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow V \subset \ell^{w^2}_-(\mathbb{R}^N)$ is of class $C^1(W)$ when it is differentiable at any point in W and the induced map $Df : W \longrightarrow L\left(\ell^{w^1}_-(\mathbb{R}^n), \ell^{w^2}_-(\mathbb{R}^N)\right)$ is continuous, where the space of linear maps $L\left(\ell^{w^1}_-(\mathbb{R}^n), \ell^{w^2}_-(\mathbb{R}^N)\right)$ is endowed with the operator norm $|||\cdot|||_{w^1,w^2}$ defined by

$$|||A|||_{w^1,w^2} := \sup_{\mathbf{u}\in\ell^{w^1}_-(\mathbb{R}^n)} \left\{\frac{\|A(\mathbf{u})\|_{w^2}}{\|\mathbf{u}\|_{w^1}} \middle| \mathbf{u} \neq \mathbf{0}\right\}, \quad A \in L\left(\ell^{w^1}_-(\mathbb{R}^n), \ell^{w^2}_-(\mathbb{R}^N)\right). \tag{2.6}$$

When in the domain and the range we use the same weighting sequence w, we will write $|||A|||_w$ instead of $|||A|||_{w^1,w^2}$. The higher order derivatives

$$D^r f(\mathbf{u}_0) : \underbrace{\ell^{w^1}_-(\mathbb{R}^n) \times \cdots \times \ell^{w^1}_-(\mathbb{R}^n)}_{r \text{ times}} \longrightarrow \ell^{w^2}_-(\mathbb{R}^N), \quad r \in \mathbb{N}^+,$$

are inductively defined and the map f is said to be of class $C^r(W)$ when it is r-times differentiable at any point in W and the induced map $D^r f : W \longrightarrow L^r\left(\ell^{w^1}_-(\mathbb{R}^n), \ell^{w^2}_-(\mathbb{R}^N)\right)$ into the normed space of r-multilinear maps is continuous. We recall that the operator norm $|||\cdot|||_{w^1,w^2}$ in $L^r\left(\ell^{w^1}_-(\mathbb{R}^n), \ell^{w^2}_-(\mathbb{R}^N)\right)$ is given by

$$|||A|||_{w^1,w^2} := \sup_{\mathbf{u}_1,\ldots,\mathbf{u}_r\in\ell^{w^1}_-(\mathbb{R}^n)} \left\{\frac{\|A(\mathbf{u}_1,\ldots,\mathbf{u}_r)\|_{w^2}}{\|\mathbf{u}_1\|_{w^1}\cdots\|\mathbf{u}_r\|_{w^1}} \middle| \mathbf{u}_1,\ldots,\mathbf{u}_r \neq \mathbf{0}\right\}, \quad A \in L^r\left(\ell^{w^1}_-(\mathbb{R}^n), \ell^{w^2}_-(\mathbb{R}^N)\right). \tag{2.7}$$

We recall that differentiable functions are automatically continuous and we denote the class of continuous functions by $C^0(W)$. When f is of class $C^r(W)$ in W for any $r \in \mathbb{N}^+$, we say that f is ***smooth*** in W and we denote this class by $C^\infty(W)$. When f is smooth in W we can construct for it a Taylor power series expansion. We say that f is ***analytic*** in W when the convergence domain of that power series includes W. The analytic class is denoted by $C^\omega(W)$.

It can be shown (see Lemma 32 in the appendices) that for any weighting sequence w, any open set in $\left(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w\right)$ contains unbounded sequences. For instance, let $B_{\|\cdot\|_w}(\mathbf{0}, \epsilon)$ be the ball of radius $\epsilon > 0$ around the zero sequence and let $\mathbf{v} \in \mathbb{R}^n$ be a vector such that $\|\mathbf{v}\| = 1$. The divergent sequence $\mathbf{z}$ defined by $\mathbf{z}_t := \epsilon\mathbf{v}/2w_{-t}$ is such that $\|\mathbf{z}\|_w = \epsilon/2$ and hence $\mathbf{z} \in B_{\|\cdot\|_w}(\mathbf{0}, \epsilon) \subset \ell^w_-(\mathbb{R}^n)$.

2.3. Causal and time-invariant filters and functionals

Let $D_n \subset \mathbb{R}^n$ and $D_N \subset \mathbb{R}^N$. We refer to the maps of the type $U : (D_n)^{\mathbb{Z}} \longrightarrow (D_N)^{\mathbb{Z}}$ as ***filters*** or ***operators*** and to those like $H : (D_n)^{\mathbb{Z}} \longrightarrow D_N$ (or $H : (D_n)^{\mathbb{Z}_\pm} \longrightarrow D_N$) as $\mathbb{R}^N$-valued ***functionals***. These definitions can be easily extended to accommodate situations where the domains and the targets of the filters are not necessarily product spaces but just arbitrary subsets V_n and V_N of $(\mathbb{R}^n)^{\mathbb{Z}}$ and $\left(\mathbb{R}^N\right)^{\mathbb{Z}}$ like, for instance, $\ell^\infty(\mathbb{R}^n)$ and $\ell^\infty(\mathbb{R}^N)$, or $\ell^w_-(\mathbb{R}^n)$ and $\ell^w_-(\mathbb{R}^N)$, for some weighting sequence w. A filter $U : (D_n)^{\mathbb{Z}} \longrightarrow (D_N)^{\mathbb{Z}}$ is called ***causal*** when for any two elements $\mathbf{z}, \mathbf{w} \in (D_n)^{\mathbb{Z}}$ that satisfy that $\mathbf{z}_\tau = \mathbf{w}_\tau$ for any $\tau \leq t$, for a given $t \in \mathbb{Z}$, we have that $U(\mathbf{z})_t = U(\mathbf{w})_t$. Let $T^{\mathbb{Z}}_\tau : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}}$ be the ***time delay*** operator defined by $T^{\mathbb{Z}}_\tau(\mathbf{z})_t := \mathbf{z}_{t-\tau}$, $\tau \in \mathbb{Z}$. A subset $V_n \subset (\mathbb{R}^n)^{\mathbb{Z}}$ is called time-invariant when $T^{\mathbb{Z}}_\tau(V_n) = V_n$, for all $\tau \in \mathbb{Z}$. The filter U is called ***time-invariant*** when it is defined on a time-invariant set and commutes with the time delay operator, that is, $T^{\mathbb{Z}}_\tau \circ U = U \circ T^{\mathbb{Z}}_\tau$, for any $\tau \in \mathbb{Z}$ (in this expression, the two operators $T^{\mathbb{Z}}_\tau$ have to be understood as defined in the appropriate sequence spaces).

We recall that there is a bijection between causal time-invariant filters and functionals on $(D_n)^{\mathbb{Z}_-}$. Indeed, given a causal and time-invariant filter $U : (D_n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^N)^{\mathbb{Z}}$, we can associate to it a functional $H_U : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^N$ via the assignment $H_U(\mathbf{z}) := U(\mathbf{z}^e)_0$, where $\mathbf{z}^e \in (\mathbb{R}^n)^{\mathbb{Z}}$ is an arbitrary extension of $\mathbf{z} \in (D_n)^{\mathbb{Z}_-}$ to $(D_n)^{\mathbb{Z}}$. Conversely, for any functional $H : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^N$, we can define a time-invariant causal filter $U_H : (D_n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^N)^{\mathbb{Z}}$ by $U_H(\mathbf{z})_t := H((\mathbb{P}_{\mathbb{Z}_-} \circ T^{\mathbb{Z}}_{-t})(\mathbf{z}))$, where $T^{\mathbb{Z}}_{-t}$ is the $(-t)$-time delay operator and $\mathbb{P}_{\mathbb{Z}_-} : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}$ is the natural projection. Moreover, when considering causal and time-invariant filters $U : (D_n)^{\mathbb{Z}} \longrightarrow (D_N)^{\mathbb{Z}}$ it suffices to work just with the restriction $U : (D_n)^{\mathbb{Z}_-} \longrightarrow (D_N)^{\mathbb{Z}_-}$, that we denote with the same symbol, since the latter uniquely determines the former. Indeed, by definition, for any $\mathbf{z} \in (D_n)^{\mathbb{Z}}$ and $t \in \mathbb{N}^+$:

$$U(\mathbf{z})_t = \left(T^{\mathbb{Z}}_{-t}\left(U(\mathbf{z})\right)\right)_0 = U\left(T^{\mathbb{Z}}_{-t}(\mathbf{z})\right)_0, \tag{2.8}$$

where the second equality holds by the time-invariance of U and the value in the right-hand side depends only on $\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\mathbf{z})\right) \in (D_n)^{\mathbb{Z}_-}$, by causality.

In view of this observation, we restrict our study to filters with domain and target in the spaces of left semi-infinite sequences. In particular, we say that a causal and time-invariant filter U has the fading memory property or that it is continuous when the corresponding restricted filter defined on left semi-infinite inputs has those properties, as we defined them in Section 2.2.

Additionally, from now on we consider most of the time delay operators with domain and target in $(\mathbb{R}^n)^{\mathbb{Z}_-}$ and that we simply denote as $T_{-\tau} : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}$. The definition of these restricted time delay operators $T_{-\tau}$ requires considering two cases:

- $T_{-\tau} : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}$ with τ negative: as before, $T_{-\tau}(\mathbf{z})_t := \mathbf{z}_{t+\tau}$, for any $\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}$ and $t \in \mathbb{Z}_-$. This implies that, in this case,

$$T_{-\tau}(\mathbf{z}) = \mathbb{P}_{\mathbb{Z}_-} \circ T^{\mathbb{Z}}_{-\tau}(\mathbf{z}^e), \quad \mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}, \quad \tau < 0,$$

  where $\mathbf{z}^e \in (\mathbb{R}^n)^{\mathbb{Z}}$ is an arbitrary extension of $\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}$ to $(\mathbb{R}^n)^{\mathbb{Z}}$. The map $T_{-\tau}$, $\tau \in \mathbb{Z}_-$, is surjective, that is, $T_\tau((\mathbb{R}^n)^{\mathbb{Z}_-}) = (\mathbb{R}^n)^{\mathbb{Z}_-}$, but it is not injective. The same applies to the restriction of $T_{-\tau}$ to any time-invariant set $V_n \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$ which satisfies $T_{-\tau}(V_n) = V_n$.

- $T_{-\tau} : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}$ with τ positive: there is in principle not a unique way to define the restricted operators $T_{-\tau}$ since that involves the choice of vectors $\mathbf{v}_\tau \in (\mathbb{R}^n)^\tau$ such that $T_{-\tau}(\mathbf{z}) := (\mathbf{z}, \mathbf{v}_\tau)$, for any $\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}$. The choice $\mathbf{v}_\tau = \mathbf{0}$ for all $\tau > 0$ is canonical since it is the only one that makes the resulting maps linear and additionally satisfy

$$T_{-\tau} = \underbrace{T_{-1} \circ \cdots \circ T_{-1}}_{\tau \text{ times}}.$$

We hence adopt the definition

$$T_{-\tau}(\mathbf{z}) := (\mathbf{z}, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{\tau \text{ times}}), \quad \mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}, \quad \tau > 0,$$

for the rest of the paper. In this case $T_{-\tau}$ it is injective but not surjective.

The following lemma gathers some differentiability properties of projections and time delay operators when restricted to normed sequence spaces and that will be used later on. A key element in this result is what we call, for each weighting sequence w, their ***decay ratio*** D_w and ***inverse decay ratio*** L_w, that are defined as:

$$D_w := \sup_{t\in\mathbb{N}} \left\{ \frac{w_{t+1}}{w_t} \right\} \quad \text{and} \quad L_w := \sup_{t\in\mathbb{N}} \left\{ \frac{w_t}{w_{t+1}} \right\}. \tag{2.9}$$

As w is by definition strictly decreasing we necessarily have that $0 < w_{t+1}/w_t < 1$, for all $t \in \mathbb{N}$, and $1 < w_0/w_1 \leq \sup_{t\in\mathbb{N}} \{w_t/w_{t+1}\} = L_w$. Consequently:

$$0 < D_w \leq 1 \quad \text{and} \quad 1 < L_w \leq +\infty.$$

The decay ratios provide a geometric bound for the convergence speed of w and the divergence rate of w^{-1}. Indeed, it is easy to see that

$$w_t \leq D_w^t \quad \text{and} \quad 1/w_t \leq L_w^t, \quad \text{for any } t \in \mathbb{N}. \tag{2.10}$$

Additionally, the fact that for all $t \in \mathbb{N}$ we have that $1 < w_t/w_{t+1}$ and that $0 < w_{t+1}/w_t < 1$ implies that

$$1/\sup_{t\in\mathbb{N}} \left\{ \frac{w_t}{w_{t+1}} \right\} = \inf_{t\in\mathbb{N}} \left\{ \frac{w_{t+1}}{w_t} \right\} \leq \sup_{t\in\mathbb{N}} \left\{ \frac{w_{t+1}}{w_t} \right\} \quad \text{and} \quad 1/\sup_{t\in\mathbb{N}} \left\{ \frac{w_{t+1}}{w_t} \right\} = \inf_{t\in\mathbb{N}} \left\{ \frac{w_t}{w_{t+1}} \right\} \leq \sup_{t\in\mathbb{N}} \left\{ \frac{w_t}{w_{t+1}} \right\},$$

which, in both cases, implies that

$$L_w D_w \geq 1. \tag{2.11}$$

More generally, in relation with the power weighting sequences that we discussed in Lemma 35, we have that:

$$0 < D_{w^n} \leq D_w \leq D_{w^{1/m}} \leq 1 \quad \text{and} \quad 1 < L_{w^{1/m}} \leq L_w \leq L_{w^n} \leq +\infty, \quad \text{for any } m, n \in \mathbb{N}^+. \tag{2.12}$$

Lemma 1 *Let w be a weighting sequence and $n \in \mathbb{N}^+$. Then:*

(i) *The projections $p_t : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\mathbb{R}^n, \|\cdot\|)$, $t \in \mathbb{Z}_-$, given by $p_t(\mathbf{z}) := \mathbf{z}_t$, $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$, are linear, smooth, and hence continuous. Moreover, $|||p_t|||_w = 1/w_{-t}$.*

(ii) *Consider the restriction of the time delay operator T_{-t} to $\ell^w_-(\mathbb{R}^n)$ for any $t \in \mathbb{Z}$. We consider two cases. First, if $t < 0$ and the inverse decay ratio L_w of w is finite, then T_{-t} maps into $\ell^w_-(\mathbb{R}^n)$, that is, $\ell^w_-(\mathbb{R}^n)$ is T_{-t}-invariant and $T_{-t} : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is surjective, open, and a submersion, that is, $\ker T_{-t}$ is a split subspace of $\ell^w_-(\mathbb{R}^n)$. If $t > 0$, then $\ell^w_-(\mathbb{R}^n)$ is always T_{-t}-invariant. $T_{-t} : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is in that case an immersion, that is, it is injective and its image $\operatorname{Im} T_{-t}$ is split. Moreover, for any $t > 0$, $T_t \circ T_{-t} = \mathbb{I}_{\ell^w_-(\mathbb{R}^n)}$, and in both cases the maps T_{-t} are linear, smooth, and hence continuous. Additionally,*

$$|||T_1|||_w = L_w, \quad |||T_{-1}|||_w = D_w, \quad |||T_{-t}|||_w \leq L_w^{-t}, \quad \text{and} \quad |||T_t|||_w \leq D_w^{-t}, \quad \text{for all } t \in \mathbb{Z}_-. \tag{2.13}$$

(iii) *For any $t_1, t_2 \in \mathbb{Z}_-$ we have*

$$p_{t_1+t_2} = p_{t_1} \circ T_{-t_2} = p_{t_2} \circ T_{-t_1}. \tag{2.14}$$

These statements also hold true when $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is replaced by $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$. In that case one has to take as sequence w the constant sequence w^t given by $w^t_t := 1$, for all $t \in \mathbb{N}$, and L_w and D_w are replaced by the constant 1.

Remark 2 The decay ratios are easy to compute for many families of weighting sequences. Two cases that we frequently encounter are:

(i) Geometric sequence: $w_t := \lambda^t$, $t \in \mathbb{N}$, with $0 < \lambda < 1$. In this case:

$$L_w := \sup_{t\in\mathbb{N}} \left\{ \frac{\lambda^t}{\lambda^{t+1}} \right\} = \frac{1}{\lambda} > 1 \quad \text{and} \quad D_w := \sup_{t\in\mathbb{N}} \left\{ \frac{\lambda^{t+1}}{\lambda^t} \right\} = \lambda < 1.$$

(ii) Harmonic sequence: $w_t := 1/(1+td)$, $t \in \mathbb{N}$, with $d > 0$. In this case $D_w = 1$ and $L_w = 1 + d$.

We emphasize that the finiteness of the inverse decay ratio is not guaranteed for all weighting sequences. An example that illustrates this fact is the sequence $w_t := \exp(-t^2)$. It is easy to verify that in that case $L_w = +\infty$ and $D_w = 1/e$.

Remark 3 The inequalities (2.13) can be combined with Gelfand's formula (Lax, 2002, page 195) to provide bounds for the spectral radii $\rho(T_{-t})$ and $\rho(T_t)$ for all $t \in \mathbb{Z}_-$. Indeed,

$$\rho(T_{-t}) = \lim_{n\to\infty} |||T^n_{-t}|||^{1/n}_w \leq \lim_{n\to\infty} (L^{-tn}_w)^{1/n} = L^{-t}_w, \quad \text{with } t \in \mathbb{Z}_-.$$

Analogously, one shows that $\rho(T_t) \leq D^{-t}_w$.

Remark 4 Lemma 1 remains valid when instead of the spaces $\ell^w_-(\mathbb{R}^n)$ we use the spaces $\ell^{p,w}_-(\mathbb{R}^n)$ that we introduced in Section 2.1, for any $1 \leq p < +\infty$. In that case, and for any $t \in \mathbb{Z}_-$,

$$|||p_t|||_{p,w} = \frac{1}{w^{1/p}_{-t}}, \tag{2.15}$$

$$|||T_1|||_{p,w} = L^{1/p}_w, \ |||T_{-1}|||_{p,w} = D^{1/p}_w, \ |||T_{-t}|||_{p,w} \leq L^{-t/p}_w, \text{ and } |||T_t|||_{p,w} \leq D^{-t/p}_w, \text{ for all } t \in \mathbb{Z}_-. \tag{2.16}$$

Remark 5 Some of the properties of time delays operators that we just studied have interesting interpretations in a Hilbert space context. See Lindquist and Picci (2015) for a detailed study.

2.4. The fading memory property and remote past input independence

The properties of time delay operators that we enunciated in Lemma 1 allow us to show how the fading memory property, defined as the continuity of a filter linking input and output spaces endowed with weighted norms, (see Section 2.1) can be interpreted as its asymptotic independence on the remote past input (Wiener, 1958, page 89). Analogously, we can see that the FMP amounts to the attribute that, in the words of Volterra (Volterra, 1930, page 188), the influence of the input a long time before the given moment fades out. This property has also been characterized as a ***unique steady-state property*** in Boyd and Chua (1985) and referred to as the ***input forgetting property*** in Jaeger (2010). All these characterizations were proved under various compactness and/or uniformly boundedness hypotheses on the inputs. The next result shows that property as a straightforward corollary of Lemma 1 that, later on in Section 4.3, will be generalized to situations where the inputs are eventually unbounded.

In the following statement we will be using the following notation: given the sequences $\mathbf{u} \in (\mathbb{R}^n)^{\mathbb{Z}_-}$ and $\mathbf{v} \in (\mathbb{R}^n)^t$, $t \in \mathbb{N}$, the symbol $\overline{\mathbf{uv}} \in (\mathbb{R}^n)^{\mathbb{Z}_-} \times (\mathbb{R}^n)^t$ denotes the ***concatenation*** of $\mathbf{u}$ and $\mathbf{v}$.

Theorem 6 (FMP and the uniform input forgetting property) *Let $M, L > 0$, $n, N \in \mathbb{N}^+$ and let $K_M \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$, $K_L \subset (\mathbb{R}^N)^{\mathbb{Z}_-}$ (respectively, $K^+_M \subset (\mathbb{R}^n)^{\mathbb{N}^+}$, $K^+_L \subset (\mathbb{R}^N)^{\mathbb{N}^+}$) be the sets of uniformly bounded left (respectively, right) semi-infinite sequences defined in (1.3). Let $U : K_M \longrightarrow K_L$ be a causal and time-invariant fading memory filter. Then, for any $\mathbf{u}, \mathbf{v} \in K_M$ and $\mathbf{z} \in K^+_M$ we have that*

$$\lim_{t\to+\infty} \|U(\overline{\mathbf{uz}})_t - U(\overline{\mathbf{vz}})_t\| = 0, \tag{2.17}$$

where in this expression the filter U is defined by time-invariance on positive times using (2.8). The convergence in (2.17) is uniform on $\mathbf{u}, \mathbf{v}$, and $\mathbf{z}$ in the sense that there exists a monotonously decreasing sequence w^U with zero limit such that for all $\mathbf{u}, \mathbf{v} \in K_M$, $\mathbf{z} \in K^+_M$, and $t \in \mathbb{N}$,

$$\|U(\overline{\mathbf{uz}})_t - U(\overline{\mathbf{vz}})_t\| \leq w^U_t. \tag{2.18}$$

*Filters that satisfy condition (2.17) for any $\mathbf{u}, \mathbf{v} \in K_M$ and $\mathbf{z} \in K^+_M$ are said to have the **input forgetting property** and we refer to (2.18) as the **uniform input forgetting property**.*

Proof. We start by recalling that in the presence of uniformly bounded inputs, the FMP can be characterized as the continuity of the map $U : K_M \longrightarrow K_L$ with the sets K_M and K_L endowed with the relative topology induced either by the product topology on $(\mathbb{R}^n)^{\mathbb{Z}_-}$ and $(\mathbb{R}^N)^{\mathbb{Z}_-}$, respectively, or by the weighted norms in the spaces $\ell^w_-(\mathbb{R}^n)$ and $\ell^w_-(\mathbb{R}^N)$, with w any weighting sequence (see (Grigoryeva and Ortega, 2018b, Corollary 2.7 and Proposition 2.11)). Moreover, the sets K_M and K_L are compact in this topology (Grigoryeva and Ortega, 2018b, Corollay 2.8) and hence the FMP filter $U : K_M \longrightarrow K_L$ is not only continuous but also uniformly continuous. Consequently, once we have fixed a weighting sequence w, an increasing modulus of continuity $\omega_U : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ can be associated to the map $U : (K_M, \|\cdot\|_w) \longrightarrow (K_L, \|\cdot\|_w)$. We emphasize that ω_U depends on w since it is a metric and not a purely topological notion. Now, using (2.8) and an arbitrary weighting sequence w that we choose with $D_w < 1$, we can write for any $t \in \mathbb{N}$

$$\begin{aligned} \|U(\overline{\mathbf{uz}})_t - U(\overline{\mathbf{vz}})_t\| &= \left\| U\left(\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{uz}})\right)\right)_0 - U\left(\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{vz}})\right)\right)_0 \right\| \\ &= \left\| p_0 \circ U\left(\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{uz}})\right)\right) - p_0 \circ U\left(\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{vz}})\right)\right) \right\| \\ &\leq \left\| U\left(\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{uz}})\right)\right) - U\left(\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{vz}})\right)\right) \right\|_w, \end{aligned} \tag{2.19}$$

where we used that $\|p_0\|_w = 1$ by the first part of Lemma 1. We now notice that

$$\mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{uz}})\right) = T_{-t}(\mathbf{u}) + (\ldots, \mathbf{0}, \mathbf{z}_1, \ldots, \mathbf{z}_t), \quad \text{and} \quad \mathbb{P}_{\mathbb{Z}_-}\left(T^{\mathbb{Z}}_{-t}(\overline{\mathbf{vz}})\right) = T_{-t}(\mathbf{v}) + (\ldots, \mathbf{0}, \mathbf{z}_1, \ldots, \mathbf{z}_t),$$

which substituted in (2.19) and using the second part of Lemma 1 yields

$$\begin{aligned} \|U(\overline{\mathbf{uz}})_t - U(\overline{\mathbf{vz}})_t\| &\leq \omega_U\left(\|T_{-t}(\mathbf{u}-\mathbf{v})\|_w\right) \\ &\leq \omega_U\left(|||T_{-t}|||_w \|\mathbf{u}-\mathbf{v}\|_w\right) \leq \omega_U\left(D^t_w \|\mathbf{u}-\mathbf{v}\|_w\right) \leq \omega_U\left(2MD^t_w\right). \end{aligned} \tag{2.20}$$

Now, as w has been chosen so that $D_w < 1$ and $\lim_{t\to 0} \omega_U(t) = 0$, we set $w^U_t := \omega_U\left(2MD^t_w\right)$, and we have that

$$\lim_{t\to+\infty} w^U_t = \lim_{t\to+\infty} \omega_U\left(2MD^t_w\right) = 0, \tag{2.21}$$

which using the inequality (2.20) proves the claim. ■

3. The fading memory property in reservoir filters with unbounded inputs

Starting in this section we focus on filters defined by reservoir systems of the type introduced in (1.1)–(1.2), but this time we consider reservoir maps $F : D_N \times D_n \longrightarrow D_N$ where the input variable takes values on a set $D_n \subset \mathbb{R}^N$ that is not necessarily bounded. All along this section, the reservoir map F will be assumed to be continuous and a contraction on the first entry with constant $0 < c < 1$, that is,

$$\left\|F(\mathbf{x}^1, \mathbf{z}) - F(\mathbf{x}^2, \mathbf{z})\right\| \leq c \left\|\mathbf{x}^1 - \mathbf{x}^2\right\|, \quad \text{for all } \mathbf{x}^1, \mathbf{x}^2 \in D_N \text{ and } \mathbf{z} \in D_n.$$

When the inputs are assumed to be uniformly bounded by a constant $M > 0$ and F maps into a ball $\overline{B_{\|\cdot\|}(\mathbf{0}, L)} \subset \mathbb{R}^N$, $L > 0$, it has been proved (see (Grigoryeva and Ortega, 2018b, Proposition 2.1 and Theorem 3.1)) that we can associate to this system unique filters $U^F : K_M \longrightarrow K_L$ and $U^F_h : K_M \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}_-}$ (the sets K_M and K_L are introduced in (1.3)) that are causal, time-invariant, continuous and, moreover, satisfy the fading memory property with respect to any weighting sequence w. We recall that U^F is the filter associated to the solutions of the reservoir equation (1.1) and assigns to any input sequence $\mathbf{z} \in K_M$ the output $U^F(\mathbf{z})$ that satisfies

$$U^F(\mathbf{z})_t = F(U^F(\mathbf{z})_{t-1}, \mathbf{z}_t), \quad \text{for any} \quad t \in \mathbb{Z}_-. \tag{3.1}$$

Recall also that $U^F_h : K_M \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}_-}$ is the filter associated to the full system (1.1)–(1.2) and is given by $U^F_h := h \circ U^F$. We denote by $H^F : K_M \longrightarrow \overline{B_{\|\cdot\|}(\mathbf{0}, L)}$ and $H^F_h : K_M \longrightarrow \mathbb{R}^d$ the corresponding reservoir functionals. The reservoir functionals are related to the corresponding reservoir filters via the identities:

$$H^F(\mathbf{z}) = U^F(\mathbf{z})_0 = F(U^F(\mathbf{z})_{-1}, \mathbf{z}_0) \quad \text{and} \quad H^F_h(\mathbf{z}) = h\left(U^F(\mathbf{z})\right), \tag{3.2}$$

for all $\mathbf{z} \in K_M$.

The next theorem is the most important result in this section and shows that the results that we just recalled about the ESP and the FMP for reservoir filters with uniformly bounded inputs remain valid in the presence of unbounded inputs. However, in that case, the fading memory property depends on the weighting sequence that is used to define it. The sufficient condition for the FMP spelled out in the next theorem asserts, roughly speaking, that reservoir systems have the FMP only with respect to weighting sequences that converge to zero sufficiently fast and at a rate that is related to the contracting properties of the reservoir map.

Theorem 7 (ESP and FMP with continuous reservoir maps) *Let $F : D_N \times D_n \longrightarrow D_N$ be a continuous reservoir map where $D_n \subset \mathbb{R}^n$, $D_N \subset \mathbb{R}^N$, $n, N \in \mathbb{N}^+$. Assume, additionally, that it is a contraction on the first entry with constant $0 < c < 1$. Let w be a weighting sequence with finite inverse decay ratio L_w and let $V_n \subset (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)$ be a time-invariant set. We consider two situations regarding the target D_N of the reservoir map:*

(i) *D_N is a compact subset of $\mathbb{R}^N$.*

(ii) *$(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ is a complete subset of the Banach space $\left(\ell^w_-(\mathbb{R}^N), \|\cdot\|_w\right)$, F is Lipschitz continuous, and the reservoir system (1.1) associated to F has a solution $(\mathbf{x}^0, \mathbf{z}^0) \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n$, that is, $\mathbf{x}^0_t = F(\mathbf{x}^0_{t-1}, \mathbf{z}^0_t)$, for all $t \in \mathbb{Z}_-$*

In both cases, if

$$cL_w < 1 \tag{3.3}$$

then the reservoir system associated to F with inputs in V_n has the echo state property and hence determines a unique continuous, causal, and time-invariant reservoir filter $U^F : (V_n, \|\cdot\|_w) \longrightarrow ((D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N), \|\cdot\|_w)$ that has the fading memory property with respect to w. Moreover, if F is Lipschitz on the second component (which is always the case under the hypotheses in **(ii)***) with constant L_z, that is,*

$$\left\|F(\mathbf{x}, \mathbf{z}^1) - F(\mathbf{x}, \mathbf{z}^2)\right\| \leq L_z \left\|\mathbf{z}^1 - \mathbf{z}^2\right\|, \quad \text{for any} \quad \mathbf{x} \in D_N,\ \mathbf{z}^1, \mathbf{z}^2 \in D_n,$$

then U^F is also Lipschitz with constant

$$L_{U^F} := \frac{L_z}{1 - cL_w}. \tag{3.4}$$

This statement also holds true under the hypotheses in part **(ii)** *when $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is replaced by $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$. In that case L_w is replaced by the constant 1 and hence condition (3.3) is automatically satisfied. The resulting reservoir filter $U^F : (V_n, \|\cdot\|_\infty) \longrightarrow ((D_N)^{\mathbb{Z}_-} \cap \ell^\infty_-(\mathbb{R}^N), \|\cdot\|_\infty)$ is continuous.*

Remark 8 A very common situation that provides the solution $(\mathbf{x}^0, \mathbf{z}^0) \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n$ for the reservoir system needed in part **(ii)**, is the existence of a fixed point $(\overline{\mathbf{x}}^0, \overline{\mathbf{z}}^0) \in D_N \times D_n$ of F that satisfies $F(\overline{\mathbf{x}}^0, \overline{\mathbf{z}}^0) = \overline{\mathbf{x}}^0$. In that case the required solution is given by the constant sequences $\mathbf{x}^0_t = \overline{\mathbf{x}}^0$, $\mathbf{z}^0_t = \overline{\mathbf{z}}^0$, for all $t \in \mathbb{Z}_-$.

Remark 9 If the target D_N of the reservoir map is a closed subset of $\mathbb{R}^N$, that is $\overline{D_N} = D_N$, then by part **(iii)** of the Corollary 33, the set $(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ is a closed subset of $\left(\ell^w_-(\mathbb{R}^N), \|\cdot\|_w\right)$ and it is hence necessarily complete, as required in case **(ii)** of the theorem. Moreover, if D_N is closed and V_n contains a constant sequence $\mathbf{z}_0$ then the condition on the existence of a solution $(\mathbf{x}^0, \mathbf{z}^0) \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n$ is automatically satisfied. Indeed, let $\mathbf{z} \in \mathbb{R}^n$ be such that $(\mathbf{z}_0)_t := \mathbf{z}$ for all $t \in \mathbb{Z}_-$ and let $\overline{\mathbf{x}} \in D_N$ arbitrary. Consider the sequence $\{\overline{\mathbf{x}}, F(\overline{\mathbf{x}}, \mathbf{z}), F(F(\overline{\mathbf{x}}, \mathbf{z}), \mathbf{z}), F(F(F(\overline{\mathbf{x}}, \mathbf{z}), \mathbf{z}), \mathbf{z}), \ldots\}$. The Banach Contraction-Mapping Principle (see (Shapiro, 2016, Theorem 3.2)) guarantees that this sequence converges to the unique fixed point, we call it $\mathbf{x} \in D_N$, of the map $F(\cdot, \mathbf{z})$. The pair $(\mathbf{x}^0, \mathbf{z}^0) \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n$, with $(\mathbf{x}_0)_t := \mathbf{x}$ for all $t \in \mathbb{Z}_-$, is the solution needed in case **(ii)** of the theorem.

As a corollary of Theorem 7 it can be shown that reservoir systems that have by construction uniformly bounded inputs and outputs always have the ESP and FMP properties and that, for any weighting sequence w. This result was already shown in (Grigoryeva and Ortega, 2018b, Theorem 3.1).

Corollary 10 *Let $M, L > 0$, let $K_M \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$ and $K_L \subset \left(\mathbb{R}^N\right)^{\mathbb{Z}_-}$ be subsets of uniformly bounded sequences defined as in (1.3), and let $F : \overline{B_{\|\cdot\|}(\mathbf{0}, L)} \times \overline{B_{\|\cdot\|}(\mathbf{0}, M)} \longrightarrow \overline{B_{\|\cdot\|}(\mathbf{0}, L)}$ be a continuous reservoir map. Assume, additionally, that F is a contraction on the first entry with constant $0 < c < 1$. Then, the reservoir system associated to F has the echo state property. Moreover, this system has a unique associated causal and time-invariant filter $U^F : K_M \longrightarrow K_L$ that has the fading memory property with respect to any weighting sequence w.*

Proof. Given that $\overline{B_{\|\cdot\|}(\mathbf{0}, L)}$ is a compact subset of $\mathbb{R}^N$, the hypothesis in part **(i)** of Theorem 7 and condition (3.3) guarantee that there exists a reservoir filter $U^F : K_M \longrightarrow K_L$ associated to F that has the fading memory property with respect to any weighting sequence that satisfies (3.3). Such a sequence always exists as it suffices to take any geometric sequence $w_t := \lambda^t$, $t \in \mathbb{N}$, with $c < \lambda < 1$. However, as it has been shown in (Grigoryeva and Ortega, 2018b, Corollary 2.7), all the weighted norms induce in the sets K_M and K_L the same topology, namely, the product topology and hence if U^F is continuous with respect to the topology induced by the weighted norm $\|\cdot\|_w$ then so it is with respect to the norm associated to any other weighting sequence. ■

Remark 11 This corollary shows that, in general, the condition (3.3) is sufficient but not necessary. Indeed, if the hypotheses in the corollary are satisfied, the resulting filter U^F has the fading memory

property with respect to any geometric sequence $w_t := \lambda^t$, with $0 < \lambda < 1$, $t \in \mathbb{N}$ for which (see Remark 2) $L_w = 1/\lambda$. In particular, this holds true when λ is chosen so that $0 < \lambda < c$ and hence when (3.3) is not satisfied since in that case $cL_w > 1$. Additional concrete examples that show that the condition (3.3) is sufficient but not necessary are provided in Section 3.1. We emphasize that the FMP condition (3.3) is sufficient but not necessary even in the absence of boundedness conditions like in Corollary 10

Another important statement that can be proved when the target of the reservoir map is a compact subset of $\mathbb{R}^N$ is that the echo state property is in that situation guaranteed *for no matter what input*[1] *in* $(\mathbb{R}^n)^{\mathbb{Z}_-}$ even though the FMP may obviously not hold in that case.

Theorem 12 (ESP for reservoir maps with compact target) *Let $F : D_N \times D_n \longrightarrow D_N$ be a continuous reservoir map, $D_n \subset \mathbb{R}^n$, $D_N \subset \mathbb{R}^N$, $n, N \in \mathbb{N}^+$, such that D_N is a compact subset of $\mathbb{R}^N$ and F is a contraction on the first entry with constant $0 < c < 1$. Then, the reservoir system associated to F has the echo state property for any input in $(D_n)^{\mathbb{Z}_-}$. Let $U^F : (D_n)^{\mathbb{Z}_-} \longrightarrow (D_N)^{\mathbb{Z}_-}$ be the associated reservoir filter. For any weighting sequence w such that $cL_w < 1$ the map $U^F : (D_n)^{\mathbb{Z}_-} \longrightarrow ((D_N)^{\mathbb{Z}_-}, \|\cdot\|_w)$ is continuous when in $(D_n)^{\mathbb{Z}_-}$ we consider the relative topology induced by the product topology in $(\mathbb{R}^N)^{\mathbb{Z}_-}$. Moreover, if $(D_n)^{\mathbb{Z}_-} \subset \ell^w_-(\mathbb{R}^n)$ then U^F has the fading memory property.*

The following result shows how the FMP of the filter associated to a reservoir map established in Theorem 7 propagates to the FMP of the filter of the full reservoir system if the readout map is continuous.

Corollary 13 *In the conditions of Theorem 7, let $h : D_N \longrightarrow \mathbb{R}^d$ be a continuous readout map. Consider the following two cases that correspond to the two sets of hypotheses studied in Theorem 7:*

(i) *If D_N is a compact subset of $\mathbb{R}^N$ then there is a constant $R > 0$ such that the filter U^F_h defined by $U^F_h(\mathbf{z})_t := h\left(U^F(\mathbf{z})_t\right)$, $t \in \mathbb{Z}_-$, $\mathbf{z} \in V_n$ maps $U^F_h : (V_n, \|\cdot\|_w) \longrightarrow (K_R, \|\cdot\|_w)$ and has the fading memory property.*

(ii) *If $(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ is a complete subset of $\left(\ell^w_-(\mathbb{R}^N, \|\cdot\|_w\right)$ and h is Lipschitz continuous on D_N such that $U^F_h(\mathbf{z}^0) \in \ell^w_-(\mathbb{R}^d)$, then the reservoir filter $U^F_h : (V_n, \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^d), \|\cdot\|_w)$ has the fading memory property.*

This statement also holds true under the hypotheses in part **(ii)** *when $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is replaced by $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$. The resulting reservoir filter $U^F_h : (V_n, \|\cdot\|_\infty) \longrightarrow (\ell^\infty_-(\mathbb{R}^d), \|\cdot\|_\infty)$ is continuous.*

3.1. Examples

In the following paragraphs we show how the sufficient condition (3.3) explicitly looks like for reservoir systems that are widely used and that have been shown to have universality properties in the fading memory category both with deterministic and stochastic inputs Grigoryeva and Ortega (2018a,b); Gonon and Ortega (2018).

Linear reservoir maps. Consider the reservoir map $F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$ given by

$$F(\mathbf{x}, \mathbf{z}) = A\mathbf{x} + \mathbf{c}\mathbf{z}, \quad \text{with} \quad A \in \mathbb{M}_N, \mathbf{c} \in \mathbb{M}_{N,n}. \tag{3.5}$$

It is easy to see that F is a contraction on the first entry whenever the matrix A satisfies that $|||A||| < 1$. In that case, using the notation in Theorem 7, $c = |||A|||$. Indeed, for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^N$, $\mathbf{z} \in \mathbb{R}^n$:

$$\|F(\mathbf{x}_1, \mathbf{z}) - F(\mathbf{x}_2, \mathbf{z})\| = \|A(\mathbf{x}_1 - \mathbf{x}_2)\| \leq |||A||| \, \|\mathbf{x}_1 - \mathbf{x}_2\| .$$

1. We thank Lukas Gonon for pointing this out.

We now assume that $|||A||| < 1$. The following two statements are proved in the Appendix 6.8:

(i) The reservoir system associated to (3.5) has the echo state property and defines a unique reservoir filter $U^F : \ell^w_-(\mathbb{R}^n) \longrightarrow \ell^w_-(\mathbb{R}^N)$ that has the fading memory property with respect to any weighting sequence w that satisfies the condition

$$\sum_{j=0}^{\infty} \frac{|||A^j|||}{w_j} < +\infty. \tag{3.6}$$

The FMP condition (3.3) reads in this case as

$$|||A|||L_w < 1, \tag{3.7}$$

and implies (3.6) but not vice versa.

(ii) If the inputs presented to the reservoir system associated to (3.5) are uniformly bounded then it has the fading memory property with respect to any weighting sequence. This result was already known as it can be easily obtained by combining (Grigoryeva and Ortega, 2018a, Corollary 11) with (Grigoryeva and Ortega, 2018b, Corollary 2.7). We obtain it here directly out of Corollary 10 by noting that for any $M > 0$,

$$F(\overline{B_{\|\cdot\|}(\mathbf{0}, L)}, \overline{B_{\|\cdot\|}(\mathbf{0}, M)} \subset \overline{B_{\|\cdot\|}(\mathbf{0}, L)}, \quad \text{with} \quad L := \frac{|||\mathbf{c}|||M}{1 - |||A|||}. \tag{3.8}$$

Echo state networks (ESN). Let $\sigma : \mathbb{R} \longrightarrow [-1, 1]$ be a squashing function, that is, σ is non-decreasing, $\lim_{x\to-\infty} \sigma(x) = -1$, and $\lim_{x\to\infty} \sigma(x) = 1$. Moreover, assume that $L_\sigma := \sup_{x\in\mathbb{R}}\{|\sigma'(x)|\} < +\infty$. Let $\boldsymbol{\sigma} : \mathbb{R}^N \longrightarrow [-1, 1]^N$ be the map obtained by componentwise application of the the squashing function σ. An echo state network is a reservoir system with linear readout and reservoir map given by

$$F(\mathbf{x}, \mathbf{z}) = \boldsymbol{\sigma}(A\mathbf{x} + \mathbf{c}\mathbf{z} + \boldsymbol{\zeta}), \quad \text{with} \quad A \in \mathbb{M}_N, \mathbf{c} \in \mathbb{M}_{N,n}, \boldsymbol{\zeta} \in \mathbb{R}^N. \tag{3.9}$$

We notice first that if $|||A|||L_\sigma < 1$ then F is a contraction on the first component with constant $|||A|||L_\sigma$ (see the second part in (Grigoryeva and Ortega, 2018b, Corollary 3.2)). By construction, F maps into the compact space $[-1, 1]^N \subset \mathbb{R}^N$ and hence satisfies the hypotheses in the first part of Theorem 7. Consequently, for any weighting sequence w that satisfies

$$|||A|||L_\sigma L_w < 1 \tag{3.10}$$

there exists a unique reservoir filter $U^F : \ell^w_-(\mathbb{R}^n) \longrightarrow \ell^w_-(\mathbb{R}^N)$ associated to F that has the fading memory property with respect to w. By Corollary 10 this statement holds true for any w when one considers uniformly bounded inputs.

Non-homogeneous state-affine systems (SAS). These systems are determined by reservoir maps $F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$ of the form

$$F(\mathbf{x}, \mathbf{z}) := p(\mathbf{z})\mathbf{x} + q(\mathbf{z}), \tag{3.11}$$

where p and q are polynomials with matrix and vector coefficients, respectively, that depending on their nature determine the following two families of SAS systems:

(i) Regular SAS. p and q are polynomials of degree r and s of the form:

$$\begin{aligned} p(\mathbf{z}) &= \sum_{\substack{i_1,\ldots,i_n\in\{0,\ldots,r\} \\ i_1+\cdots+i_n\leq r}} z_1^{i_1}\cdots z_n^{i_n} A_{i_1,\ldots,i_n}, \quad A_{i_1,\ldots,i_n} \in \mathbb{M}_N, \quad \mathbf{z} \in D_n \subset \mathbb{R}^n, \\ q(\mathbf{z}) &= \sum_{\substack{i_1,\ldots,i_n\in\{0,\ldots,s\} \\ i_1+\cdots+i_n\leq s}} z_1^{i_1}\cdots z_n^{i_n} B_{i_1,\ldots,i_n}, \quad B_{i_1,\ldots,i_n} \in \mathbb{M}_{N,1}, \quad \mathbf{z} \in D_n \subset \mathbb{R}^n. \end{aligned}$$

(ii) Trigonometric SAS. We use trigonometric polynomials instead:

$$p(\mathbf{z}) = \sum_{k=1}^{r} A_k^p \cos(\mathbf{u}_k^p \cdot \mathbf{z}) + B_k^p \sin(\mathbf{v}_k^p \cdot \mathbf{z}), \quad A_k^p, B_k^p \in \mathbb{M}_N, \quad \mathbf{u}_k^p, \mathbf{v}_k^p \in \mathbb{R}^N, \quad \mathbf{z} \in D_n \subset \mathbb{R}^n,$$

$$q(\mathbf{z}) = \sum_{k=1}^{s} A_k^q \cos(\mathbf{u}_k^q \cdot \mathbf{z}) + B_k^q \sin(\mathbf{v}_k^q \cdot \mathbf{z}), \quad A_k^q, B_k^q \in \mathbb{M}_{N,1}, \quad \mathbf{u}_k^q, \mathbf{v}_k^q \in \mathbb{R}^N, \quad \mathbf{z} \in D_n \subset \mathbb{R}^n.$$

In both cases, define

$$M_p := \sup_{\mathbf{z}\in D_n} \{|||p(\mathbf{z})|||\} \quad \text{and} \quad M_q := \sup_{\mathbf{z}\in D_n} \{|||q(\mathbf{z})|||\}.$$

Note that for regular SAS defined by nontrivial polynomials, the set D_n needs to be bounded in order for M_p and M_q to be finite. Additionally, it is easy to see that F is a contraction on the first entry with constant M_p whenever $M_p < 1$, which is a condition that we will assume holds true in the rest of this example. Additionally, we assume that $M_q < +\infty$. Regular SAS are a generalization of the linear case that we considered in the first part of this section and hence two statements can be proved (see Appendix 6.8) that are analogous to the ones in that part, namely:

(i) The reservoir system associated to (3.11) has the echo state property and defines a unique reservoir filter $U^F : \ell_-^w(\mathbb{R}^n) \cap (D_n)^{\mathbb{Z}_-} \longrightarrow \ell_-^w(\mathbb{R}^N)$ that has the fading memory property with respect to any weighting sequence w that satisfies the condition

$$\sum_{j=0}^{\infty} \frac{M_p^j}{w_j} < +\infty. \tag{3.12}$$

The FMP condition (3.3) that in this case reads as $M_p L_w < 1$ implies (3.12) but not vice versa.

(ii) If the inputs presented to the reservoir system associated to (3.11) are uniformly bounded then it has the fading memory property with respect to any weighting sequence. We obtain this result out of Corollary 10 by noting that for any $M > 0$,

$$F(\overline{B_{\|\cdot\|}(\mathbf{0}, L)}, \overline{B_{\|\cdot\|}(\mathbf{0}, M)} \subset \overline{B_{\|\cdot\|}(\mathbf{0}, L)}, \quad \text{with} \quad L := \frac{M_q}{1 - M_p}.$$

We emphasize that in the case of regular SAS, this is the only situation for which one can have $M_p < 1$ and $M_q < +\infty$.

4. Differentiability in reservoir filters with unbounded inputs

We now extend the results in the previous section from continuity to differentiability. More specifically, we characterize the situations in which one can prove the existence and obtain the differentiability of reservoir filters out of the differentiability properties of the maps that define the reservoir system. This approach gives us in passing new techniques to establish the echo state and the fading memory properties of reservoir systems. In particular, differentiability being a local property, we show how systems that do not globally have any of these properties may still have them in a neighborhood of certain types of inputs. A phenomenon of this type has also been explored in Manjunath and Jaeger (2013).

It is worth emphasizing that the study of the differentiability properties of fading memory reservoir filters calls naturally for the handling of unbounded inputs since the definition of the Fréchet derivative requires them to be defined on open subsets of the Banach space $\ell_-^w(\mathbb{R}^n)$ that always contain unbounded sequences (see the first part of Lemma 32 in the appendices).

4.1. Differentiable reservoir filters associated to differentiable reservoir maps

The first result in this section shows that under certain conditions, the echo state and the fading memory properties associated to differentiable reservoir systems locally persist, that is, if a reservoir system has a unique filter associated to a specific input and it is continuous and differentiable at it, then the same property holds for neighboring inputs.

Theorem 14 (Local persistence of the ESP and FMP properties) *Let $F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$ be a reservoir map and let w be a weighting sequence with finite inverse decay ratio L_w. Suppose that F is of class $C^1(\mathbb{R}^N \times \mathbb{R}^n)$ and that the corresponding reservoir system (1.1) has a solution $(\mathbf{x}^0, \mathbf{z}^0) \in \ell_-^w(\mathbb{R}^N) \times \ell_-^w(\mathbb{R}^n)$, that is, $\mathbf{x}_t^0 = F(\mathbf{x}_{t-1}^0, \mathbf{z}_t^0)$, for all $t \in \mathbb{Z}_-$. Suppose, additionally, that*

$$L_F := \sup_{(\mathbf{x},\mathbf{z})\in\mathbb{R}^N\times\mathbb{R}^n} \{|||DF(\mathbf{x}, \mathbf{z})|||\} < +\infty. \tag{4.1}$$

Define

$$L_{F_x}(\mathbf{x}^0, \mathbf{z}^0) := \sup_{t\in\mathbb{Z}_-} \{|||D_x F(\mathbf{x}_{t-1}^0, \mathbf{z}_t^0)|||\}$$

and suppose that

$$L_{F_x}(\mathbf{x}^0, \mathbf{z}^0) L_w < 1. \tag{4.2}$$

Then there exist open time-invariant neighborhoods $V_{\mathbf{x}^0}$ and $V_{\mathbf{z}^0}$ of $\mathbf{x}^0$ and $\mathbf{z}^0$ in $\ell_-^w(\mathbb{R}^N)$ and $\ell_-^w(\mathbb{R}^n)$, respectively, such that the reservoir system associated to F with inputs in $V_{\mathbf{z}^0}$ has the echo state property and hence determines a unique causal and time-invariant reservoir filter $U^F : (V_{\mathbf{z}^0}, \|\cdot\|_w) \longrightarrow (V_{\mathbf{x}^0}, \|\cdot\|_w)$. Moreover, U^F is differentiable at all the points of the form $T_{-t}(\mathbf{z}^0)$, $t \in \mathbb{Z}_-$, it is locally Lipschitz continuous on $V_{\mathbf{z}^0}$, and it hence has the fading memory property.

Remark 15 We refer to (4.2) as the ***persistence condition***. We emphasize that this inequality puts into relation the solution $(\mathbf{x}^0, \mathbf{z}^0)$ whose persistence we are studying with the weighting sequence w. In particular, that relation tells us that solutions are more likely to persist with respect to weighting sequences that decay more slowly (that is, L_w is smaller).

Remark 16 There is a situation where the persistence condition is particularly easy to verify, namely, when the solution of the reservoir system is constructed as a constant sequence coming from a fixed point of the reservoir map, that is, $(\mathbf{x}^0, \mathbf{z}^0) \in \mathbb{R}^N \times \mathbb{R}^n$ such that $F(\mathbf{x}^0, \mathbf{z}^0) = \mathbf{x}^0$. In that case $L_{F_x}(\mathbf{x}^0, \mathbf{z}^0) := |||D_x F(\mathbf{x}^0, \mathbf{z}^0)|||$.

Remark 17 The persistence condition (4.2) can be interpreted as a stability condition for the reservoir system determined by F at the solution $(\mathbf{x}^0, \mathbf{z}^0)$ with respect to perturbations in $\ell_-^w(\mathbb{R}^n)$. The persistence of solutions under stability conditions of that type has been thoroughly studied for many types of dynamical systems (see, for instance, Montaldi (1997b,a); Ortega and Ratiu (1997); Chossat et al. (2003)).

Remark 18 The derivative $DU^F(\mathbf{z}^0)$ at $\mathbf{z}^0$ of the locally defined reservoir filter U^F is determined by the differentiation of the relation (3.1). Indeed, for any $\mathbf{u} \in \ell_-^w(\mathbb{R}^n)$, and $t \in \mathbb{Z}_-$, the directional derivative $DU^F(\mathbf{z}^0) \cdot \mathbf{u}$ is determined by the recursions

$$(DU^F(\mathbf{z}^0) \cdot \mathbf{u})_t = DF\left(U^F(\mathbf{z}^0)_{t-1}, \mathbf{z}_t^0\right) \cdot \left(\left(DU^F(\mathbf{z}^0) \cdot \mathbf{u}\right)_{t-1}, \mathbf{u}_t\right) \tag{4.3}$$

$$= D_x F\left(U^F(\mathbf{z}^0)_{t-1}, \mathbf{z}_t^0\right) \cdot \left(DU^F(\mathbf{z}^0) \cdot \mathbf{u}\right)_{t-1} + D_z F(U(\mathbf{z}^0)_{t-1}, \mathbf{z}_t^0) \cdot \mathbf{u}_t. \tag{4.4}$$

This relation implies, in particular, that $DU^F(\mathbf{z}^0) : \ell_-^w(\mathbb{R}^n) \longrightarrow \ell_-^w(\mathbb{R}^N)$ is a bounded linear operator and that

$$|||DU^F(\mathbf{z}^0)|||_w \leq \frac{L_{F_z}(\mathbf{x}^0, \mathbf{z}^0)}{1 - L_{F_x}(\mathbf{x}^0, \mathbf{z}^0) L_w}, \tag{4.5}$$

where

$$L_{F_z}(\mathbf{x}^0,\mathbf{z}^0) := \sup_{t\in\mathbb{Z}_-}\left\{\left|\!\left|\!\left| D_zF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\right|\!\right|\!\right|\right\}.$$

Indeed, notice first that for any $t\in\mathbb{Z}_-$,

$$\left|\!\left|\!\left| D_xF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\right|\!\right|\!\right| \leq \left|\!\left|\!\left| DF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\right|\!\right|\!\right| \quad\text{and}\quad \left|\!\left|\!\left| D_zF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\right|\!\right|\!\right| \leq \left|\!\left|\!\left| DF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\right|\!\right|\!\right|, \tag{4.6}$$

which, using hypothesis (4.1) implies that

$$L_{F_x}(\mathbf{x}^0,\mathbf{z}^0) \leq L_F < +\infty \quad\text{and}\quad L_{F_z}(\mathbf{x}^0,\mathbf{z}^0) \leq L_F < +\infty. \tag{4.7}$$

Now, for any $\mathbf{u}\in\ell^w_-(\mathbb{R}^n)$, and $t\in\mathbb{Z}_-$, the relation (4.3) and the inequalities (4.7) imply that

$$\begin{aligned}
\left\|DU^F(\mathbf{z}^0)\cdot\mathbf{u}\right\|_w &= \sup_{t\in\mathbb{Z}_-}\left\{\left\|\left(DU^F(\mathbf{z}^0)\cdot\mathbf{u}\right)_t\right\| w_{-t}\right\}\\
&= \sup_{t\in\mathbb{Z}_-}\left\{\left\|DF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\cdot\left(\left(DU^F(\mathbf{z}^0)\cdot\mathbf{u}\right)_{t-1},\mathbf{u}_t\right)\right\| w_{-t}\right\}\\
&= \sup_{t\in\mathbb{Z}_-}\left\{\left\|D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\cdot\left(DU^F(\mathbf{z}^0)\cdot\mathbf{u}\right)_{t-1} + D_zF(U(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t)\cdot\mathbf{u}_t\right\| w_{-t}\right\}\\
&\leq L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)\sup_{t\in\mathbb{Z}_-}\left\{\left\|\left(DU^F(\mathbf{z}^0)\cdot\mathbf{u}\right)_{t-1}\right\| w_{-t}\right\} + L_{F_z}(\mathbf{x}^0,\mathbf{z}^0)\sup_{t\in\mathbb{Z}_-}\left\{\|\mathbf{u}_t\|\, w_{-t}\right\}\\
&\leq L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)\sup_{t\in\mathbb{Z}_-}\left\{\left\|\left(DU^F(\mathbf{z}^0)\cdot\mathbf{u}\right)_{t-1}\right\| w_{-(t-1)}\frac{w_{-t}}{w_{-(t-1)}}\right\}\\
&\qquad + L_{F_z}(\mathbf{x}^0,\mathbf{z}^0)\sup_{t\in\mathbb{Z}_-}\left\{\|\mathbf{u}_t\|\, w_{-t}\right\}\\
&\leq L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w\left\|DU^F(\mathbf{z}^0)\cdot\mathbf{u}\right\|_w + L_{F_z}(\mathbf{x}^0,\mathbf{z}^0)\left\|\mathbf{u}\right\|_w,
\end{aligned}$$

which implies (4.5).

The previous theorem proves that when the persistence condition (4.2) is satisfied at a preexisting solution of a reservoir system then this system has a unique fading memory (and differentiable) filter associated for neighboring inputs. In the next results we show that a global version of that condition ensures first, that globally defined reservoir filters exist, and second, that those filters are differentiable and hence have the fading memory property.

Theorem 19 (Characterization of global reservoir filter differentiability) *Let $F:\mathbb{R}^N\times\mathbb{R}^n\longrightarrow\mathbb{R}^N$ be a reservoir map of class $C^1(\mathbb{R}^N\times\mathbb{R}^n)$ and let w be a weighting sequence with finite inverse decay ratio L_w.*

(i) *Suppose that F satisfies (4.1) and define*

$$L_{F_x} := \sup_{(\mathbf{x},\mathbf{z})\in\mathbb{R}^N\times\mathbb{R}^n}\left\{\left|\!\left|\!\left| D_xF(\mathbf{x},\mathbf{z})\right|\!\right|\!\right|\right\} \quad\text{and}\quad L_{F_z} := \sup_{(\mathbf{x},\mathbf{z})\in\mathbb{R}^N\times\mathbb{R}^n}\left\{\left|\!\left|\!\left| D_zF(\mathbf{x},\mathbf{z})\right|\!\right|\!\right|\right\}.$$

If the reservoir system (1.1) associated to F has a solution $(\mathbf{x}^0,\mathbf{z}^0)\in\ell^w_-(\mathbb{R}^N)\times\ell^w_-(\mathbb{R}^n)$, that is, $\mathbf{x}^0_t = F(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)$, for all $t\in\mathbb{Z}_-$, and

$$L_{F_x}L_w < 1 \tag{4.8}$$

then it has the echo state property and hence determines a unique causal and time-invariant reservoir filter $U^F:(\ell^w_-(\mathbb{R}^n),\|\cdot\|_w)\longrightarrow(\ell^w_-(\mathbb{R}^N),\|\cdot\|_w)$. Moreover, U^F is differentiable and Lipschitz continuous on $\ell^w_-(\mathbb{R}^n)$ with Lipschitz constant L_{U^F} given by

$$L_{U^F} := \frac{L_{F_z}}{1-L_{F_x}L_w} \quad\text{and}\quad \left|\!\left|\!\left| DU^F(\mathbf{z})\right|\!\right|\!\right|_w \leq \frac{L_{F_z}}{1-L_{F_x}L_w}, \quad\text{for any } \mathbf{z}\in\ell^w_-(\mathbb{R}^n). \tag{4.9}$$

The filter U^F has hence the fading memory property.

(ii) *Conversely, let $V_n\subset\ell^w_-(\mathbb{R}^n)$ be an open and time-invariant subset of $\ell^w_-(\mathbb{R}^n)$ and assume that the reservoir system (1.1) associated to F has a unique causal and time-invariant reservoir filter $U^F:V_n\longrightarrow\ell^w_-(\mathbb{R}^N)$ that is differentiable at $\mathbf{z}^0\in\ell^w_-(\mathbb{R}^n)$. Then,*

$$\rho\left(\left(\prod_{t\in\mathbb{Z}_-}D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\right)\circ T_1\right) < 1, \tag{4.10}$$

where ρ stands for the spectral radius. This in turn implies that

$$\lim_{k\to+\infty}\left(\left|\!\left|\!\left| D_xF\left(U^F(\mathbf{z}^0)_{-1},\mathbf{z}^0_0\right)\circ\cdots\circ D_xF\left(U^F(\mathbf{z}^0)_{-k},\mathbf{z}^0_{-k+1}\right)\right|\!\right|\!\right|\frac{1}{w_k}\right) = 0. \tag{4.11}$$

Examples 20 We briefly examine the form that the hypotheses of Theorem 19 take for the three families of reservoir systems that we analyzed in Section 3.1:

(i) Linear reservoir maps. In this case, for any $\mathbf{x}\in\mathbb{R}^N$ and $\mathbf{z}\in\mathbb{R}^n$,

$$DF(\mathbf{x},\mathbf{z}) = (A\,|\,\mathbf{c}), \quad D_xF(\mathbf{x},\mathbf{z}) = A, \quad\text{and}\quad D_zF(\mathbf{x},\mathbf{z}) = \mathbf{c}.$$

Consequently $L_F = \left|\!\left|\!\left|(A\,|\,\mathbf{c})\right|\!\right|\!\right|$, $L_{F_x} = \left|\!\left|\!\left|A\right|\!\right|\!\right|$, $L_{F_z} = \|\mathbf{c}\|$. The condition (4.1) is always satisfied and in this case the sufficient differentiability condition (4.8) amounts to $\left|\!\left|\!\left|A\right|\!\right|\!\right|L_w < 1$ that, as we saw in (3.7), is the same as the sufficient condition for the FMP to hold.

(ii) Echo state networks (ESN). Consider an ESN constructed using a squashing function σ that satisfies that $L_\sigma := \sup_{x\in\mathbb{R}}\{|\sigma'(x)|\} < +\infty$. In this case, for any $\mathbf{x}\in\mathbb{R}^N$ and $\mathbf{z}\in\mathbb{R}^n$,

$$\begin{aligned}
DF(\mathbf{x},\mathbf{z}) &= D\boldsymbol{\sigma}(A\mathbf{x}+\mathbf{c}\mathbf{z}+\boldsymbol{\zeta})\circ(A\,|\,\mathbf{c}),\\
D_xF(\mathbf{x},\mathbf{z}) &= D\boldsymbol{\sigma}(A\mathbf{x}+\mathbf{c}\mathbf{z}+\boldsymbol{\zeta})\circ A,\\
D_zF(\mathbf{x},\mathbf{z}) &= D\boldsymbol{\sigma}(A\mathbf{x}+\mathbf{c}\mathbf{z}+\boldsymbol{\zeta})\circ\mathbf{c}.
\end{aligned}$$

Notice that $\left|\!\left|\!\left|D\boldsymbol{\sigma}(\mathbf{x})\right|\!\right|\!\right| < L_\sigma < +\infty$, for any $\mathbf{x}\in\mathbb{R}^N$, and hence

$$\begin{aligned}
\left|\!\left|\!\left|DF(\mathbf{x},\mathbf{z})\right|\!\right|\!\right| &\leq L_\sigma\left|\!\left|\!\left|(A\,|\,\mathbf{c})\right|\!\right|\!\right| < +\infty,\\
\left|\!\left|\!\left|D_xF(\mathbf{x},\mathbf{z})\right|\!\right|\!\right| &\leq L_\sigma\left|\!\left|\!\left|A\right|\!\right|\!\right| < +\infty,\\
\left|\!\left|\!\left|D_zF(\mathbf{x},\mathbf{z})\right|\!\right|\!\right| &\leq L_\sigma\|\mathbf{c}\| < +\infty,
\end{aligned}$$

for any $\mathbf{x}\in\mathbb{R}^N$ and $\mathbf{z}\in\mathbb{R}^n$. This implies, in particular, that in this case

$$L_F < +\infty, \quad L_{F_x} < +\infty, \quad L_{F_z} < +\infty,$$

and the sufficient differentiability condition (4.8) is implied by the inequality

$$\left|\!\left|\!\left|A\right|\!\right|\!\right|L_\sigma L_w < 1. \tag{4.12}$$

(iii) Non-homogeneous state-affine systems (SAS). A straightforward computations shows that for any $\mathbf{x}\in\mathbb{R}^N$ and $\mathbf{z}\in\mathbb{R}^n$,

$$\begin{aligned}
DF(\mathbf{x},\mathbf{z}) &= (p(\mathbf{z}), Dp(\mathbf{z})(\cdot)\mathbf{x}+Dq(\mathbf{z})(\cdot)),\\
D_xF(\mathbf{x},\mathbf{z}) &= p(\mathbf{z}),\\
D_zF(\mathbf{x},\mathbf{z}) &= Dp(\mathbf{z})(\cdot)\mathbf{x}+Dq(\mathbf{z})(\cdot).
\end{aligned} \tag{4.13}$$

As we already pointed out, for regular SAS defined by nontrivial polynomials the norm $|||p(z)|||$ is not bounded in $\mathbb{R}^n$ and hence $L_{F_x} = \sup_{(\mathbf{x},\mathbf{z})\in\mathbb{R}^N\times\mathbb{R}^n}\{|||D_xF(\mathbf{x},\mathbf{z})|||\} = \sup_{\mathbf{z}\in\mathbb{R}^n}\{|||p(\mathbf{z})|||\} = M_p$ is not finite; the same applies to L_F, which implies that in this case neither (4.1) nor (4.8) can be satisfied.

This is not the case for trigonometric SAS for which the norms of the derivatives in (4.13) are bounded on their domains which, in particular, implies that $L_F < +\infty$, $L_{F_x} < +\infty$, and $L_{F_z} < +\infty$. Moreover, the sufficient differentiability condition (4.8) in this case reads

$$M_pL_w < 1.$$

Remark 21 We recall here an example that we introduced in Section 3.1 to show that, as it was already the case with the FMP condition (3.3) in Theorem 7, the differentiability condition (4.8) is sufficient but not necessary. Indeed, consider a linear system with matrix A given by

$$A = \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}, \quad \text{with} \quad a > 0.$$

Given that $|||A||| = a$, the reservoir map determined by A is not necessarily a contraction on the first entry. Nevertheless, the nilpotency of A implies that the reservoir system associated to (3.5) always has a solution for any input $\mathbf{z} \in (\mathbb{R}^2)^{\mathbb{Z}_-}$ and hence has the ESP and induces a filter $U : (\mathbb{R}^2)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^2)^{\mathbb{Z}_-}$ given by $U(\mathbf{z})_t := \mathbf{z}_t + A\mathbf{z}_{t-1}$, $t \in \mathbb{Z}_-$ or, equivalently, $U = \mathbb{I}_{(\mathbb{R}^2)^{\mathbb{Z}_-}} + (\prod_{t\in\mathbb{Z}_-} A)\circ T_1$. Consider now any weighting sequence w with finite inverse decay ratio L_w. Then the restriction of U to $\ell^w_-(\mathbb{R}^2)$ always maps into $\ell^w_-(\mathbb{R}^2)$, has the FMP, and it is differentiable. Indeed, it is easy to show using the linearity of the filter that $U = DU(\mathbf{z})$ for any $\mathbf{z} \in \ell^w_-(\mathbb{R}^2)$ and that

$$|||U|||_w = |||DU(\mathbf{z})|||_w \leq (1 + aL_w). \tag{4.14}$$

Note that in this case $L_{F_x} = |||A||| = a$ and as (4.14) shows the differentiability of U with respect to any weighting sequence with finite L_w, we can conclude that the condition (4.8) is not necessary for filter differentiability.

The following corollary puts together the previous theorem and a condition on the readout map that guarantees that the filter associated to the resulting reservoir system is differentiable.

Corollary 22 *Consider a reservoir system determined by a reservoir map $F : \mathbb{R}^N\times\mathbb{R}^n \longrightarrow \mathbb{R}^N$ of class $C^1(\mathbb{R}^N\times\mathbb{R}^n)$ and by a readout map $h : \mathbb{R}^N \longrightarrow \mathbb{R}^d$ that is also of class $C^1(\mathbb{R}^N)$. Assume, additionally that F satisfies the hypotheses in part* **(i)** *of Theorem 19 and that h is such that*

$$c_h := \sup_{\mathbf{x}\in\mathbb{R}^N}\{|||Dh(\mathbf{x})|||\} < +\infty, \tag{4.15}$$

and the sequence $\mathbf{y}^0 := \left(h\left(\mathbf{x}^0\right)\right)_{t\in\mathbb{Z}_-} = \left(h\left(U^F(\mathbf{z}^0)\right)\right)_{t\in\mathbb{Z}_-} \in \ell^w_-(\mathbb{R}^d)$. Then, the reservoir filter $U^F_h : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^d), \|\cdot\|_w)$ is differentiable at each point in its domain and it hence has the fading memory property.

Proof. Define first the map

$$\mathcal{H} := \prod_{t\in\mathbb{Z}_-} h\circ p_t : \ell^w_-(\mathbb{R}^N) \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}_-}. \tag{4.16}$$

Given that $U^F_h = \mathcal{H}\circ U^F$ and by Theorem 19 the filter U^F is differentiable then it suffices to prove that $\mathcal{H}$ is differentiable. This is a consequence of part **(iii)** in Lemma 36 and the hypothesis (4.15). Indeed, let $H_t := h\circ p_t$, $t \in \mathbb{Z}_-$, and notice that by the first part of Lemma 1

$$\sup_{\mathbf{x}\in\ell^w_-(\mathbb{R}^N)}\{|||DH_t(\mathbf{x})|||\} \leq \sup_{\mathbf{x}_t\in\mathbb{R}^N}\{|||Dh(\mathbf{x}_t)|||\}\cdot\sup_{\mathbf{x}\in\ell^w_-(\mathbb{R}^N)}\{|||p_t(\mathbf{x})|||\} \leq \frac{c_h}{w_{-t}}.$$

Now, as $\left\|(c_h/w_{-t})_{t\in\mathbb{Z}_-}\right\|_w = c_h < +\infty$ and by hypothesis $\mathcal{H}(\mathbf{x}^0) \in \ell^w_-(\mathbb{R}^d)$ it follows from Lemma 36 that $\mathcal{H}$ maps into $\ell^w_-(\mathbb{R}^d)$ and that it is differentiable, as required. ■

In some occasions it is important to determine if a given filter is invertible. The differentiability of reservoir filters associated to reservoir systems associated to differentiable reservoir and readout maps that we established in the previous result allows us to use the inverse function theorem to formulate a sufficient invertibility condition. As we see in the next statement, this criterion can be written down entirely in terms of the derivatives of the reservoir and the readout maps.

Corollary 23 *Consider a reservoir system determined by a reservoir map $F : \mathbb{R}^N\times\mathbb{R}^n \longrightarrow \mathbb{R}^N$ and a readout map $h : \mathbb{R}^N \longrightarrow \mathbb{R}^d$ that are of class $C^1(\mathbb{R}^N\times\mathbb{R}^n)$ and $C^1(\mathbb{R}^N)$, respectively, and additionally satisfy the conditions spelled out in the statement of Corollary 22. Let $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$, $\mathbf{x} := U^F(\mathbf{z}) \in \ell^w_-(\mathbb{R}^N)$, and $\mathbf{y} := U^F_h(\mathbf{z}) \in \ell^w_-(\mathbb{R}^d)$, and suppose that the map*

$$D\mathcal{H}(\mathbf{x})\circ\left(\mathbb{I}_{\ell^w_-(\mathbb{R}^N)} - \left(\prod_{t\in\mathbb{Z}_-} D_xF\left(\mathbf{x}_{t-1},\mathbf{z}_t\right)\right)\circ T_1\right)^{-1}\circ\left(\prod_{t\in\mathbb{Z}_-} D_zF\left(\mathbf{x}_{t-1},\mathbf{z}_t\right)\right) : \ell^w_-(\mathbb{R}^n) \longrightarrow \ell^w_-(\mathbb{R}^d) \tag{4.17}$$

is a linear homeomorphism (continuous linear bijection with continuous inverse) with $\mathcal{H}$ as defined in (4.16). Then there exist open neighborhoods $V_{\mathbf{z}} \subset \ell^w_-(\mathbb{R}^n)$ and $V_{\mathbf{y}} \subset \ell^w_-(\mathbb{R}^N)$ of $\mathbf{z}$ and $\mathbf{y}$, respectively, such that the restriction of the filter $U^F_h|_{V_{\mathbf{z}}} : V_{\mathbf{z}} \longrightarrow V_{\mathbf{y}}$ has an inverse $\left(U^F_h|_{V_{\mathbf{z}}}\right)^{-1}$. When the condition (4.17) is satisfied for all the solutions $(\mathbf{z}, U^F(\mathbf{z}))$ of the reservoir system determined by F then the reservoir filter U^F_h admits a global inverse $\left(U^F_h\right)^{-1} : U^F_h\left(\ell^w_-(\mathbb{R}^n)\right) \longrightarrow \ell^w_-(\mathbb{R}^n)$.

Proof. It is a straightforward consequence of the inverse function theorem as formulated in (Schechter, 1997, page 670) (see also Ver Eecke (1974)) applied to the Fréchet derivative of $U^F_h = \mathcal{H}\circ U^F$ at the point $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$. It is easy to see using the chain rule and (6.69) (which is in turn a consequence of (4.4)) that this derivative coincides with the operator in (4.17) whose invertibility we require. ■

4.2. The local versus the global echo state property

Theorem 14 emphasizes the local nature of both the echo state and the fading memory properties by providing a sufficient condition that ensures the existence of a locally defined causal and time-invariant filter around a given solution that is shown to have the FMP. In contrast with this local approach, Theorem 19 characterizes the existence of a globally defined differentiable filter associated to a given reservoir system, that hence satisfies the FMP and the ESP for any input.

Even though the conditions in Theorems 14 and 19 are very alike, the latter is much stronger than the former. In the following paragraphs we illustrate with a family of ESNs of the type introduced in Section 3.1 how it is possible to be in violation of the global condition of Theorem 19 and nevertheless to find solutions of such reservoir systems around which one can locally define FMP reservoir filters. This example illustrates how *the ESP and the FMP are structural features of a reservoir system when considered globally but are mostly input dependent when considered only locally.* This important observation has already been noticed in Manjunath and Jaeger (2013) where, using tools coming from the theory of non-autonomous dynamical systems, sufficient conditions have been formulated (see, for instance, (Manjunath and Jaeger, 2013, Theorem 2)) that ensure the ESP in connection to a given specific input. The differentiability conditions that we impose to our reservoir systems allow us to draw similar conclusions and, additionally, to automatically conclude the FMP of the resulting locally defined reservoir filters.

Consider the one-dimensional echo state map $F : \mathbb{R}\times\mathbb{R} \longrightarrow \mathbb{R}$, where

$$F(x,z) := \sigma(ax+z), \quad \text{with} \quad a\in\mathbb{R} \quad \text{and} \quad \sigma(x) := \frac{x}{\sqrt{1+x^2}}. \tag{4.18}$$

The sigmoid function σ in this expression has been chosen so that we can provide algebraic expressions in the following developments. Similar conclusions could nevertheless be drawn using other popular squashing functions.

The function σ maps the real line into the interval $[-1,1]$ and it is easy to see, using the notation introduced in the examples 20, that $L_\sigma := \sup_{x\in\mathbb{R}}\{|\sigma'(x)|\} = 1$. Moreover, the one-dimensional character of the system makes that, in this case,

$$L_{F_x} = |a|. \tag{4.19}$$

Consequently, by Lemma 43, the reservoir map F is a contraction on the first entry if and only if $|a| < 1$, in which case, by Theorem 7, the associated ESN has the ESP and the FMP with respect to any input in $\ell^w_-(\mathbb{R}^n)$, where w is a weighting sequence that satisfies

$$|a| L_w < 1. \tag{4.20}$$

The FMP holds with respect to any sequence w if we consider uniformly bounded inputs by Corollary 10. Moreover, a well-known result for ESNs due to H. Jaeger (see (Jaeger, 2010, Proposition 3)) shows that the ESP cannot be satisfied whenever

$$|a| > 1. \tag{4.21}$$

Additionally, the global sufficient differentiability condition (4.8) in Theorem 19 states that the condition (4.20) also ensures that the ESP filter is also differentiable.

We now prove using Theorem 14 *the existence of locally defined FMP filters associated to this ESN in a neighborhood of certain inputs, even when condition* (4.21) *is satisfied which, as we already mentioned, prevents the global existence of such objects.*

Notice first that the solutions of the equation $\sigma(ax) = x$, $x \in \mathbb{R}$, are characterized by the relation

$$a^2x^2 = x^2(a^2x^2+1) \tag{4.22}$$

that has as solutions

$$\begin{cases} x_0 &= 0, \\ x_a^\pm &= \pm\frac{\sqrt{a^2-1}}{a}, \end{cases}$$

where the solutions in the second line obviously exist and are different from the first one only when $|a| > 1$, a condition that we assume holds true in the rest of the section. The condition (4.22) implies that the constant sequences $(\mathbf{x}_0, \mathbf{z}_0)$ and $(\mathbf{x}_a^\pm, \mathbf{z}_0)$ defined by

$$(\mathbf{x}_0, \mathbf{z}_0)_t := (x_0, 0) \quad \text{and} \quad (\mathbf{x}_a^\pm, \mathbf{z}_0)_t := (x_a^\pm, 0), \quad \text{for any} \quad t \in \mathbb{Z}_-,$$

are solutions of the reservoir system determined by F. Moreover, in the notation of Theorem 14, it is easy to see that

$$L_{F_x}(\mathbf{x}_0, \mathbf{z}_0) = |a| > 1 \quad \text{and} \quad L_{F_x}(\mathbf{x}_a^\pm, \mathbf{z}_0) = \frac{1}{a^2} < 1.$$

The persistence condition (4.2) in that result implies that for any weighting sequence that satisfies

$$\frac{L_w}{a^2} < 1,$$

there exist open time-invariant neighborhoods $V_{\mathbf{x}_a^\pm}$ and $V_{\mathbf{z}^0}$ of $\mathbf{x}_a^\pm$ and $\mathbf{z}^0$ in $\ell^w_-(\mathbb{R}^N)$ and $\ell^w_-(\mathbb{R}^n)$, respectively, such that the reservoir system associated to F with inputs in $V_{\mathbf{z}^0}$ has the echo state property and hence determines a unique causal, time-invariant, and FMP reservoir filter $U^F : (V_{\mathbf{z}^0}, \|\cdot\|_w) \longrightarrow (V_{\mathbf{x}^0}, \|\cdot\|_w)$.

4.3. Remote past input independence and the state forgetting property for unbounded inputs

In Section 2.3 we saw how fading memory filters presented with uniformly bounded inputs exhibit what we called the uniform input forgetting property. An analysis of the proof of the main result in that section, namely Theorem 6, shows that the compactness of the space of inputs guaranteed the existence of a modulus of continuity for the filter, which ensured the validity of the input forgetting property and, moreover, it made it uniform. In the context of reservoir systems, we saw in Theorems 7 and 19 that there are very weak hypotheses that, even when the inputs are not uniformly bounded, guarantee that the associated reservoir filters are Lipschitz and hence have a modulus of continuity. This allows us to prove an input forgetting property in that more general context.

Theorem 24 (Input forgetting property for FMP reservoir filters) *Let $F : D_N \times D_n \longrightarrow D_N$ be a reservoir map where $D_n \subset \mathbb{R}^n$, $D_N \subset \mathbb{R}^N$, $n, N \in \mathbb{N}^+$. Assume that the hypotheses of Theorem 7 part* **(ii)** *(plus F Lipschitz on the second component) or 19 part* **(i)** *are satisfied with respect to a weighting sequence w such that $D_w < 1$. Let $U^F : (V_n, \|\cdot\|_w) \longrightarrow ((D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N), \|\cdot\|_w)$ be the associated causal and time-invariant reservoir filter ($V_n \subset (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)$ under the hypotheses of Theorem 7; $V_n = \ell^w_-(\mathbb{R}^n)$ and $D_N = \mathbb{R}^N$ under the hypotheses of Theorem 19). Then, for any $\mathbf{u}, \mathbf{v} \in \ell^w_-(\mathbb{R}^n)$ and $\mathbf{z} \in (D_N)^{\mathbb{Z}_-}$ we have that*

$$\lim_{t\to+\infty} \left\| U^F(\overline{\mathbf{uz}})_t - U^F(\overline{\mathbf{vz}})_t \right\| = 0. \tag{4.23}$$

If D_N is compact then the convergence in (4.23) *is uniform on $\mathbf{u}, \mathbf{v}$, and $\mathbf{z}$ in the sense that there exists a monotonously decreasing sequence w^{U^F} with zero limit such that for all $\mathbf{u}, \mathbf{v}$, $\mathbf{z}$, and $t \in \mathbb{N}$,*

$$\left\| U^F(\overline{\mathbf{uz}})_t - U^F(\overline{\mathbf{vz}})_t \right\| \leq w_t^{U^F}. \tag{4.24}$$

Proof. It mimics the proof of Theorem 6 using as modulus of continuity the map $\omega_{U^F}(t) := L_{U^F} t$, $t \geq 0$, where L_{U^F} is the Lipschitz constant whose existence is ensured by the hypotheses of Theorem 7 or 19 and given by (3.4) or by (4.9), respectively. ■

Remark 25 In Remark 41 in the appendices it is shown how Theorem 7 can be extended to continuous reservoir systems with inputs and outputs in $\ell^{p,w}_-(\mathbb{R}^n)$ and $\ell^{p,w}_-(\mathbb{R}^N)$, respectively. In particular, it is shown that the resulting filters are Lipschitz and hence have a non-trivial modulus of continuity. This implies that a result analogous to Theorem 24 can be proved for such systems that hence could also be referred to as fading memory from a dynamical point of view.

When filters are differentiable, there is one more way to measure how they forget inputs simply by looking at their partial derivatives with respect to past input components. The result is a differential input forgetting property that, unlike Theorem 24, can be formulated in a uniform way even when the inputs are not uniformly bounded.

Theorem 26 (Differential uniform input forgetting property) *Assume that the hypotheses of Theorem 19* **(i)** *are satisfied. Let $D_{z_t^i} H^F(\mathbf{z}) \in \mathbb{R}^N$ be the partial derivative of the reservoir functional $H^F : \ell^w_-(\mathbb{R}^n) \longrightarrow \mathbb{R}^N$ with respect to the i-th component of the t-th entry of $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$. Then, there exists a monotonously decreasing sequence w^F with zero limit such that, for any $t \in \mathbb{Z}_-$,*

$$\left\| D_{z_t^i} H^F(\mathbf{z}) \right\| \leq w^F_{-t}, \quad \text{for any} \quad \mathbf{z} \in \ell^w_-(\mathbb{R}^n) \quad \text{and} \quad i \in \{1, \ldots, n\}. \tag{4.25}$$

Proof. Let $\mathbf{e}^{i,t} := \left(\ldots, \mathbf{0}, \mathbf{e}^i, \mathbf{0}, \ldots, \mathbf{0}\right) \in \ell^w_-(\mathbb{R}^n)$, where the vector $\mathbf{e}^i$ is the canonical vector in $\mathbb{R}^n$ and it is placed in the t-th position. Then, since $\left\|\mathbf{e}^{i,t}\right\|_w = w_{-t}$, we have by (4.9) and for any $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$ that

$$\left\|D_{z_t^i} H^F(\mathbf{z})\right\| = \left\|DH^F(\mathbf{z}) \cdot \mathbf{e}^{i,t}\right\| \leq \left\|\left\|DH^F(\mathbf{z})\right\|\right\|_w \left\|\mathbf{e}^{i,t}\right\|_w \leq \left\|\left\|p_0 \circ DU^F(\mathbf{z})\right\|\right\|_w w_{-t} \leq \frac{L_{F_z}}{1 - L_{F_x} L_w} w_{-t},$$

which proves (4.25) by setting

$$w_t^F := \frac{L_{F_z}}{1 - L_{F_x} L_w} w_t, \quad t \in \mathbb{N}. \quad \blacksquare$$

Apart from the filters that reservoir maps define when they have the echo state property, we can also use this object to define controlled forward-looking dynamical systems and flows. Indeed, given $F : D_N \times D_n \longrightarrow D_N$ a reservoir map, we denote by $\mathcal{F}^F : (D_n)^{\mathbb{N}^+} \times D_N \longrightarrow (D_N)^{\mathbb{N}^+}$ the ***reservoir flow*** associated to F that is uniquely determined by the recurrence relations:

$$\left\{ \begin{array}{rcl} \mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_1 & = & F(\mathbf{x}_0, \mathbf{z}_1) \quad \text{with} \quad \mathbf{z} \in (D_n)^{\mathbb{N}^+},\ \mathbf{x}_0 \in D_N, \\ \mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_t & = & F(\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_{t-1}, \mathbf{z}_t), \quad t > 1. \end{array} \right. \tag{4.26}$$

The value $\mathbf{x}_0 \in D_N$ is called the ***initial condition*** of the ***path*** $\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0) \in (D_N)^{\mathbb{N}^+}$ associated to the ***input*** or ***control sequence*** $\mathbf{z} \in (D_n)^{\mathbb{N}^+}$.

As we saw in Theorems 7 and 19, the contracting property on the first component in a reservoir map is much related to the ESP and the FMP of the resulting reservoir filter and, in passing, (see Theorem 24) to the input forgetting property. The next result shows that something similar happens with reservoir flows associated to contracting reservoir maps as they *forget* the influence of initial conditions that are used to create the paths. This feature is referred to as the ***state forgetting property*** in Jaeger (2010).

Theorem 27 (State forgetting property for contracting reservoir flows) *Let $F : D_N \times D_n \longrightarrow D_N$ be a reservoir map where $D_n \subset \mathbb{R}^n$, $D_N \subset \mathbb{R}^N$, $n, N \in \mathbb{N}^+$, and suppose that F is a contraction on the first component. Given an input sequence $\mathbf{z} \in (D_n)^{\mathbb{N}^+}$, the reservoir flow $\mathcal{F}^F : (D_n)^{\mathbb{N}^+} \times D_N \longrightarrow (D_N)^{\mathbb{N}^+}$ associated to F satisfies that:*

$$\lim_{t \to +\infty} \left\|\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_t - \mathcal{F}^F(\mathbf{z}, \overline{\mathbf{x}}_0)_t\right\| = 0, \quad \textit{for any} \quad \mathbf{x}_0, \overline{\mathbf{x}}_0 \in D_N. \tag{4.27}$$

If D_N is compact then the convergence in (4.27) is uniform on $\mathbf{z}, \mathbf{x}_0$, and $\overline{\mathbf{x}}_0$ in the sense that there exists a monotonously decreasing sequence $w^{\mathcal{F}}$ with zero limit such that for all $\mathbf{x}_0, \overline{\mathbf{x}}_0 \in D_N$, $\mathbf{z} \in (D_n)^{\mathbb{N}^+}$, and $t \in \mathbb{N}$,

$$\left\|\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_t - \mathcal{F}^F(\mathbf{z}, \overline{\mathbf{x}}_0)_t\right\| \leq w_t^{\mathcal{F}}. \tag{4.28}$$

*Reservoir filters that satisfy condition (4.27) are said to have the **state forgetting property** and we refer to (4.28) as the **uniform state forgetting property**.*

Proof. Let $c < 1$ be the contraction constant of F. Using the recursions (4.26) that define the reservoir flow we can write that for any $t > 1$:

$$\begin{aligned} \left\|\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_t - \mathcal{F}^F(\mathbf{z}, \overline{\mathbf{x}}_0)_t\right\| &= \left\|F(\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_{t-1}, \mathbf{z}_t) - F(\mathcal{F}^F(\mathbf{z}, \overline{\mathbf{x}}_0)_{t-1}, \mathbf{z}_t)\right\| \\ &\leq c \left\|\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_{t-1} - \mathcal{F}^F(\mathbf{z}, \overline{\mathbf{x}}_0)_{t-1}\right\| \leq \cdots \leq c^{t-1} \left\|\mathcal{F}^F(\mathbf{z}, \mathbf{x}_0)_1 - \mathcal{F}^F(\mathbf{z}, \overline{\mathbf{x}}_0)_1\right\| \\ &\leq c^{t-1} \left\|F(\mathbf{x}_0, \mathbf{z}_1) - F(\overline{\mathbf{x}}_0, \mathbf{z}_1)\right\|. \end{aligned}$$

Taking limits $t \to +\infty$ on both sides of this inequality yields (4.27). Now, if D_N is compact then there exists a constant $D > 0$ such that $\left\|F(\mathbf{x}_0, \mathbf{z}_1) - F(\overline{\mathbf{x}}_0, \mathbf{z}_1)\right\| < D$ for all $\mathbf{x}_0, \overline{\mathbf{x}}_0 \in D_N$, and $\mathbf{z}_1 \in D_n$, and hence (4.28) holds if we set $w_t^{\mathcal{F}} := c^{t-1} D$, $t \in \mathbb{N}$. $\blacksquare$

4.4. Analytic reservoir filters associated to analytic reservoir maps

The results in Section 4.1 characterized the conditions under which reservoir maps of class C^1 yield differentiable reservoir filters with respect to inputs and outputs in weighted sequence spaces. This setup is convenient because it is able to accommodate unbounded signals and allows for an elegant encoding of the fading memory property. However, due to the infinite dimensional character of our setup one cannot immediately obtain higher order differentiable reservoir filters out of higher order differentiable reservoir maps (see Remark 40) because one needs, roughly speaking, to modify the weighted norm in the target of the map that defines the filter (see Proposition 39). This makes impossible the application in a higher order differentiability context of the Implicit Function Theorem, which is the main tool used in the results in the previous section. That is why in the following paragraphs we deal with analytic reservoir maps (as real valued functions) and we study the analyticity of the associated reservoir filters with respect to the supremum norm, as opposed to the weighted norms that we considered in the previous section.

Using the supremum norm implies that filter differentiability in that context, when one manages to establish it, ensures filter continuity and not the fading memory property. In exchange, analyticity allows us to construct Taylor series expansions that, as we see later on, are discrete-time Volterra series representations.

The next result is the analytic analog of the Local Persistence Theorem 14 formulated using the supremum norm that proves that analytic reservoir maps have locally defined analytic reservoir filters associated around constant solutions.

Theorem 28 (Local persistence of the ESP, continuity, and analyticity) *Let $F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$ be a reservoir map. Suppose that F is analytic and that the corresponding reservoir system (1.1) has a constant solution $(\mathbf{x}^0, \mathbf{z}^0) \in \mathbb{R}^N \times \mathbb{R}^n$, that is, $\mathbf{x}^0 = F(\mathbf{x}^0, \mathbf{z}^0)$. Suppose, additionally, that for all $r \geq 1$,*

$$L_{F,r} := \sup_{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^N \times \mathbb{R}^n} \left\{\left\|\left\|D^r F(\mathbf{x}, \mathbf{z})\right\|\right\|\right\} < +\infty. \tag{4.29}$$

Suppose that

$$L_{F_x}(\mathbf{x}^0, \mathbf{z}^0) := \left\|\left\|D_x F(\mathbf{x}^0, \mathbf{z}^0)\right\|\right\| < 1. \tag{4.30}$$

Then, there exist open time-invariant neighborhoods $V_{\mathbf{x}^0}$ and $V_{\mathbf{z}^0}$ of $\mathbf{x}^0$ and $\mathbf{z}^0$ in $\ell^\infty_-(\mathbb{R}^N)$ and $\ell^\infty_-(\mathbb{R}^n)$, respectively, such that the reservoir system associated to F with inputs in $V_{\mathbf{z}^0}$ has the echo state property and hence determines a unique causal, time-invariant, and analytic (and hence continuous) reservoir filter $U^F : (V_{\mathbf{z}^0}, \|\cdot\|_\infty) \longrightarrow (V_{\mathbf{x}^0}, \|\cdot\|_\infty)$.

5. The Volterra series representation of analytic filters and a universality theorem

In this section we study the Taylor series expansions of analytic causal and time-invariant filters that, as we prove in the next result, coincide with the so-called discrete-time Volterra series representations. A very similar result has been formulated in Sandberg (1998a, 1999) for analytic filters with respect to the supremum norm and with inputs with a finite past. The next result extends that statement and characterizes the inputs for which an analytic time-invariant fading memory filter with respect to a weighted norm admits a Volterra series representation with semi-infinite past inputs. This generalized result allows this series representation for inputs that are not necessarily bounded. Additionally, we use the causality and time-invariance hypotheses to show that the corresponding Volterra series representations have time-independent coefficients.

Theorem 29 *Let w be a weighting sequence and let $U : B_{\|\cdot\|_w}(\mathbf{z}^0, M) \subset \ell^w_-(\mathbb{R}) \longrightarrow B_{\|\cdot\|_w}(U(\mathbf{z}^0), L) \subset \ell^w_-(\mathbb{R}^N)$ be a causal and time-invariant analytic filter, for some time-invariant $\mathbf{z}^0 \in \ell^{1,w}_-(\mathbb{R})$ (that is, $T_{-t}(\mathbf{z}^0) = \mathbf{z}^0$, for all $t \in \mathbb{Z}_-$) and $M, L > 0$. Then, for any element in the domain that satisfies*

$$\mathbf{z} \in B_{\|\cdot\|_w}(\mathbf{z}^0, M) \cap \ell^{1,w}_-(\mathbb{R}), \quad \text{that is} \quad \sum_{t\in\mathbb{Z}_-} |z_t| w_{-t} < +\infty, \tag{5.1}$$

there exists a unique expansion

$$U(\mathbf{z})_t = U(\mathbf{z}^0)_t + \sum_{j=1}^{\infty} \sum_{m_1=-\infty}^{0} \cdots \sum_{m_j=-\infty}^{0} g_j(m_1,\ldots,m_j)(z_{m_1+t} - z^0_{m_1+t}) \cdots (z_{m_j+t} - z^0_{m_j+t}), \quad t \in \mathbb{Z}_-, \tag{5.2}$$

where the maps $g_j : \mathbb{Z}^j_- \longrightarrow \mathbb{R}^N$, $j \geq 1$, are uniquely determined by the derivatives of the functional $H_U : B_{\|\cdot\|_w}(\mathbf{z}^0, M) \subset \ell^w_-(\mathbb{R}) \longrightarrow \mathbb{R}^N$ associated to U (that by Proposition 39 in the Appendix is analytic) via the relation

$$g_j(m_1,\ldots,m_j) := \frac{1}{j!} D^j H(\mathbf{z}^0)\left(e_{m_1},\ldots,e_{m_j}\right) \quad \text{with} \quad (e_n)_t := \begin{cases} 1 & \text{if } t = n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

Moreover, for any $p \in \mathbb{N}^+$, we have that

$$\left\| U(\mathbf{z})_t - U(\mathbf{z}^0)_t - \sum_{j=1}^{p} \sum_{m_1=-\infty}^{0} \cdots \sum_{m_j=-\infty}^{0} g_j(m_1,\ldots,m_j)(z_{m_1+t} - z^0_{m_1+t}) \cdots (z_{m_j+t} - z^0_{m_j+t}) \right\| \leq \frac{L}{w_{-t}} \left(1 - \frac{\|\mathbf{z}\|_w}{M}\right)^{-1} \left(\frac{\|\mathbf{z}\|_w}{M}\right)^{p+1}. \tag{5.4}$$

These statements also hold true when $\ell^w_-(\mathbb{R})$ and $\ell^w_-(\mathbb{R}^N)$ are replaced by $\ell^\infty_-(\mathbb{R})$ and $\ell^\infty_-(\mathbb{R}^N)$, respectively. In that case, the relation (5.2) holds whenever $\mathbf{z} \in B_{\|\cdot\|_\infty}(\mathbf{z}^0, M) \cap \ell^1_-(\mathbb{R}^n)$ and the inequality (5.4) is obtained by taking as the sequence w the constant sequence w^ι given by $w^\iota_t := 1$, for all $t \in \mathbb{N}$.

Remark 30 The error estimate (5.4) can be reformulated in terms of the weighted norm of the sequence

$$\mathbf{R}_p(\mathbf{z}) := \left(U(\mathbf{z})_t - U(\mathbf{z}^0)_t - \sum_{j=1}^{p} \sum_{m_1=-\infty}^{0} \cdots \sum_{m_j=-\infty}^{0} g_j(m_1,\ldots,m_j)(z_{m_1+t} - z^0_{m_1+t}) \right)_{t\in\mathbb{Z}_-},$$

as

$$\|\mathbf{R}_p(\mathbf{z})\|_w \leq L \left(1 - \frac{\|\mathbf{z}\|_w}{M}\right)^{-1} \left(\frac{\|\mathbf{z}\|_w}{M}\right)^{p+1}. \tag{5.5}$$

5.1. Finite discrete-time Volterra series are universal in the fading memory category

In this section we combine the Volterra series representation Theorem 29 with previous universality results in Grigoryeva and Ortega (2018a) to show that any fading memory filter with uniformly bounded inputs can be arbitrarily well approximated with a Volterra series with finite terms of the type in (5.2). This result provides an alternative proof of a Volterra series universality theorem that was stated for the first time in (Boyd and Chua, 1985, Theorems 3 and 4). In particular, this result shows that any time-invariant and causal fading memory filter can be uniformly approximated by a finite memory filter.

Theorem 31 (Universality of finite discrete-time Volterra series) *Let $M, L > 0$ and let $K_M \subset (\mathbb{R})^{\mathbb{Z}_-}, K_L \subset \left(\mathbb{R}^d\right)^{\mathbb{Z}_-}$ be as in (1.3). Let $U : K_M \longrightarrow K_L$ be a causal and time-invariant fading memory filter. Then, for any $\epsilon > 0$ there exist $\mathbf{x}^0 \in K_L$ and $J \in \mathbb{N}^+$ such that for any $j \in \{1,\ldots,J\}$ there exist j numbers $M^j_1,\ldots,M^j_j \in \mathbb{N}^+$ and maps $g_j : \mathbb{Z}^j_- \longrightarrow \mathbb{R}$ such that the filter determined by the finite Volterra series given by*

$$V(\mathbf{z})_t = \mathbf{x}^0_t + \sum_{j=1}^{J} \sum_{m_1=-M^j_1}^{0} \cdots \sum_{m_j=-M^j_j}^{0} g_j(m_1,\ldots,m_j) z_{m_1+t} \cdots z_{m_j+t} \tag{5.6}$$

is such that

$$\|U - V\|_\infty = \sup_{\mathbf{z}\in K_M} \{\|U(\mathbf{z}) - V(\mathbf{z})\|\} < \epsilon.$$

Proof. The Corollary 11 in Grigoryeva and Ortega (2018a) guarantees that for any $\epsilon > 0$ there exists a linear reservoir system with polynomial readout $h \in \mathbb{R}[\mathbf{x}]$ and nilpotent connectivity matrix $A \in \mathbb{M}_N$, determined by the expressions

$$\begin{cases} \mathbf{x}_t = A\mathbf{x}_{t-1} + \mathbf{c}\mathbf{z}_t, \quad A \in \mathbb{M}_N, \mathbf{c} \in \mathbb{M}_{N,n}, & (5.7) \\ y_t = h(\mathbf{x}_t), \quad h \in \mathbb{R}[\mathbf{x}], & (5.8) \end{cases}$$

such that it has an associated reservoir filter $U^{A,\mathbf{c}}_h : K_M \longrightarrow K_L$ that satisfies

$$\left\| U - U^{A,\mathbf{c}}_h \right\|_\infty < \epsilon. \tag{5.9}$$

Let $J = \deg(h) + 1$ and assume that A is nilpotent of index p. In order to prove the theorem it suffices to show that the Volterra series expansion in (5.2) corresponding to $U^{A,\mathbf{c}}_h$ has an expression of the type (5.6). If that is the case, the statement in (5.9) proves the theorem.

Indeed, recall (see, for instance, (Grigoryeva and Ortega, 2018a, Corollary 11)) that the functional $H^{A,\mathbf{c}}_h$ associated to the filter $U^{A,\mathbf{c}}_h$ is given by

$$H^{A,\mathbf{c}}_h(\mathbf{z}) = h\left(\sum_{j=0}^{p-1} A^j(\mathbf{c}z_{-j})\right),$$

which is a composition of the polynomial h with the functional $H^{A,\mathbf{c}}$ associated to the reservoir equation (5.7) given by the linear operator

$$H^{A,\mathbf{c}}(\mathbf{z}) := \sum_{j=0}^{p-1} A^j(\mathbf{c}z_{-j}). \tag{5.10}$$

It is easy to see that $H^{A,\mathbf{c}} : (\ell^\infty_-(\mathbb{R}), \|\cdot\|_\infty) \longrightarrow \mathbb{R}^N$ has a finite operator norm $\left\|\left|H^{A,\mathbf{c}}\right|\right\|_\infty$ and that $\left\|\left|H^{A,\mathbf{c}}\right|\right\|_\infty \leq \|\mathbf{c}\|/(1 - \|A\|)$, with $\|\mathbf{c}\|$ and $\|A\|$ the top singular values of $\mathbf{c}$ and A, respectively. Moreover, it is easy to see that for any $j \in \mathbb{N}^+$, $\mathbf{z} \in K_M$, and $\mathbf{v}_1,\ldots,\mathbf{v}_j \in \ell^\infty_-(\mathbb{R})$, we have

$$D^j H^{A,\mathbf{c}}_h(\mathbf{z})(\mathbf{v}_1,\ldots,\mathbf{v}_j) = D^j h(H^{A,\mathbf{c}}(\mathbf{z}))\left(H^{A,\mathbf{c}}(\mathbf{v}_1),\ldots,H^{A,\mathbf{c}}(\mathbf{v}_j)\right)),$$

which shows that $H^{A,\mathbf{c}}_h : (\ell^\infty_-(\mathbb{R}), \|\cdot\|_\infty) \longrightarrow \mathbb{R}^d$ is everywhere analytic. Using this expression and (5.3) we define

$$g_j(m_1,\ldots,m_j) := \frac{1}{j!} D^j h(\mathbf{0})\left(H^{A,\mathbf{c}}(e_{m_1}),\ldots,H^{A,\mathbf{c}}(e_{m_j})\right). \tag{5.11}$$

As h has finite degree then $D^j h(\mathbf{0}) = 0$ for any $j > \deg(h) + 1 = J$. Moreover, since the sum in (5.10) is finite by the nilpotency of A it is clear that $g_j(m_1, \ldots, m_j)$ in (5.11) is nonzero as long as $1 \leq j \leq \deg(h) + 1 = J$ and $-(p-1) \leq m_1, \ldots, m_j \leq 0$. If we define $M_1^j, \ldots, M_j^j := p - 1$ then the Taylor series expansion of $U^{A,\mathbf{c}}(\mathbf{z})$ coincides with (5.6).

We emphasize that in this case this expansion is valid for any $\mathbf{z} \in K_M$ by the finiteness of the number of terms in the sum and that the condition (5.1) is hence not necessary. ■

6. Appendices

6.1. The topologies induced by weighted and supremum norms

An important feature of the topology generated by weighted norms is that they coincide with the product topology on subsets made of uniformly bounded sequences like the space K_M in (1.3). This fact holds true for any weighting sequence w and has important consequences (see Grigoryeva and Ortega (2018b) for the details). First, the fading memory property is independent of the weighting sequence used to define it. Second, the subsets $K_M \subset \ell_-^w(\mathbb{R}^n)$ are compact in the topology induced by the weighted norms $\|\cdot\|_w$. We emphasize that these statements are valid exclusively in the context of uniformly bounded subsets which, as we see in the next result, are never open in the weighted topology.

We adopt in the sequel the following notation for product sets and functions: for any family $\{A_t\}_{t\in\mathbb{Z}_-}$, of subsets $A_t \subset \mathbb{R}^n$ the symbol

$$\prod_{t\in\mathbb{Z}_-} A_t := \left\{ \mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-} \mid \mathbf{z}_t \in A_t, \quad \text{for all } t \in \mathbb{Z}_- \right\}, \tag{6.1}$$

denotes the Cartesian product of the sets in the family. When all the elements in the family are identical to a given subset A, we will exchangeably use the symbols $\prod_{t\in\mathbb{Z}_-} A$ and $(A)^{\mathbb{Z}_-}$. A similar notation is adopted for the Cartesian product of maps: let V be a set and let $f_t : V \longrightarrow A_t$ be a map, $t \in \mathbb{Z}_-$. The symbol $\prod_{t\in\mathbb{Z}_-} f_t$ denotes the map

$$\begin{array}{rccc} \prod_{t\in\mathbb{Z}_-} f_t : & V & \longrightarrow & \prod_{t\in\mathbb{Z}_-} A_t \\ & v & \longmapsto & (\ldots, f_{-2}(v), f_{-1}(v), f_0(v)) \,. \end{array} \tag{6.2}$$

Lemma 32 *Let w be a weighting sequence and $n \in \mathbb{N}^+$. Then:*

(i) *For any $\mathbf{z} \in \ell_-^w(\mathbb{R}^n)$ and $r > 0$,*

$$B_{\|\cdot\|_w}(\mathbf{z}, r) = \bigcup_{\delta<r} \left(\prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}\left(\mathbf{z}_t, \frac{\delta}{w_{-t}}\right) \right). \tag{6.3}$$

In particular, this implies that

$$B_{\|\cdot\|_w}(\mathbf{z}, r) \subset \prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}\left(\mathbf{z}_t, \frac{r}{w_{-t}}\right) \subset \overline{B_{\|\cdot\|_w}(\mathbf{z}, r)}. \tag{6.4}$$

The identity (6.3) implies that any open ball $B_{\|\cdot\|_w}(\mathbf{z}, r)$ in $\ell_-^w(\mathbb{R}^n)$ contains unbounded sequences.

(ii) *Let $\{A_t\}_{t\in\mathbb{Z}_-}$ be a family of subsets $A_t \subset \mathbb{R}^n$ such that there exists a sequence $\{c_t\}_{t\in\mathbb{Z}_-}$ that satisfies*

$$\sup_{\mathbf{z}_t\in A_t} \{\|\mathbf{z}_t\| \, w_{-t}\} < c_t, \text{ for each } t \in \mathbb{Z}_- \text{ and } \sup_{t\in\mathbb{Z}_-} \{c_t\} < +\infty, \tag{6.5}$$

then the product set

$$\prod_{t\in\mathbb{Z}_-} A_t \subset \ell_-^w(\mathbb{R}^n).$$

(iii) *For every family $\{A_t\}_{t\in\mathbb{Z}_-}$ of subsets $A_t \subset \mathbb{R}^n$ such that the product set satisfies $\prod_{t\in\mathbb{Z}_-} A_t \subset \ell_-^w(\mathbb{R}^n)$, we have*

$$\overline{\prod_{t\in\mathbb{Z}_-} A_t} = \prod_{t\in\mathbb{Z}_-} \overline{A_t}. \tag{6.6}$$

These statements, except for the last sentence in part **(i)**, *are also valid for the space $\ell_-^\infty(\mathbb{R}^n)$ and are obtained by taking as sequence w the constant sequence w^ι given by $w_t^\iota := 1$, for all $t \in \mathbb{N}$.*

Proof. **(i)** We prove (6.3) by double inclusion. First, let $\mathbf{x} \in B_{\|\cdot\|_w}(\mathbf{z}, r)$. By definition $\|\mathbf{x} - \mathbf{z}\|_w = \sup_{t\in\mathbb{Z}_-} \{\|\mathbf{x}_t - \mathbf{z}_t\| \, w_{-t}\} < r$ and hence for any $\delta_x > 0$ such that $\|\mathbf{x} - \mathbf{z}\|_w < \delta_x < r$ we have that $\|\mathbf{x}_t - \mathbf{z}_t\| < \delta_x / w_{-t}$, for all $t \in \mathbb{Z}_-$. This implies that

$$\mathbf{x} \in \prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}\left(\mathbf{z}_t, \frac{\delta_x}{w_{-t}}\right) \subset \bigcup_{\delta<r} \left(\prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}\left(\mathbf{z}_t, \frac{\delta}{w_{-t}}\right) \right).$$

Conversely, given an element $\mathbf{x} \in \ell_-^w(\mathbb{R}^n)$ in the right hand side of (6.3), there exists $\delta_x < r$ such that $\mathbf{x} \in \prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}(\mathbf{z}_t, \delta_x / w_{-t})$. This implies that $\|\mathbf{x}_t - \mathbf{z}_t\| \, w_{-t} < \delta_x$, for all $t \in \mathbb{Z}_-$, and hence $\sup_{t\in\mathbb{Z}_-} \{\|\mathbf{x}_t - \mathbf{z}_t\| \, w_{-t}\} = \|\mathbf{x} - \mathbf{z}\|_w \leq \delta_x < r$, which proves the inclusion.

As to (6.4), the first inclusion is a straightforward consequence of (6.3). Let now $\mathbf{x} \in \prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}\left(\mathbf{z}_t, \frac{r}{w_{-t}}\right)$. By definition this implies that $\|\mathbf{x}_t - \mathbf{z}_t\| \, w_{-t} < r$, for all $t \in \mathbb{Z}_-$, and consequently $\sup_{t\in\mathbb{Z}_-} \{\|\mathbf{x}_t - \mathbf{z}_t\| \, w_{-t}\} \leq r$ or, equivalently, $\|\mathbf{x} - \mathbf{z}\|_w \leq r$. This implies that $\mathbf{x} \in \overline{B_{\|\cdot\|_w}(\mathbf{z}, r)}$ and proves the second inclusion.

(ii) Let $\mathbf{x} \in \prod_{t\in\mathbb{Z}_-} A_t$. Then,

$$\|\mathbf{x}\|_w = \sup_{t\in\mathbb{Z}_-} \{\|\mathbf{x}_t\| \, w_{-t}\} \leq \sup_{t\in\mathbb{Z}_-} \{c_t\} < +\infty,$$

as required.

(iii) We first prove that $\prod_{t\in\mathbb{Z}_-} \overline{A_t} \subset \overline{\prod_{t\in\mathbb{Z}_-} A_t}$. If $\mathbf{z} \in \prod_{t\in\mathbb{Z}_-} \overline{A_t}$, then for any $\epsilon > 0$ and each $t \in \mathbb{Z}_-$ there exists an element $\mathbf{x}_t \in A_t \cap B_{\|\cdot\|}\left(\mathbf{z}_t, \frac{\epsilon}{2w_{-t}}\right)$. Let $\mathbf{x} := (\mathbf{x}_t)_{t\in\mathbb{Z}_-}$. By construction:

$$\|\mathbf{x} - \mathbf{z}\|_w = \sup_{t\in\mathbb{Z}_-} \{\|\mathbf{x}_t - \mathbf{z}_t\| \, w_{-t}\} \leq \frac{\epsilon}{2} < \epsilon,$$

which implies that $\mathbf{x} \in B_{\|\cdot\|_w}(\mathbf{z}, \epsilon) \cap \prod_{t\in\mathbb{Z}_-} A_t$ and, as $\mathbf{z} \in \prod_{t\in\mathbb{Z}_-} \overline{A_t}$ is arbitrary, it guarantees that $\mathbf{z} \in \overline{\prod_{t\in\mathbb{Z}_-} A_t}$.

In order to show the reverse inclusion first note that, as it is proved later on in Lemma 1, the projections $p_t : \ell_-^w(\mathbb{R}^n) \longrightarrow \mathbb{R}^n$, $t \in \mathbb{Z}_-$, defined by $p_t(\mathbf{z}) := \mathbf{z}_t$, are continuous. Let $\mathbf{z} \in \overline{\prod_{t\in\mathbb{Z}_-} A_t}$ arbitrary, let $t \in \mathbb{Z}_-$ be arbitrary but fixed, and let V_t be an open set in $\mathbb{R}^n$ that contains $\mathbf{z}_t$. The continuity of p_t implies that $p_t^{-1}(V_t)$ is an open set in $\ell_-^w(\mathbb{R}^n)$ that contains $\mathbf{z}$ and therefore there exists $\mathbf{x} \in \left(\prod_{t\in\mathbb{Z}_-} A_t\right) \cap p_t^{-1}(V_t)$. We consequently have that $\mathbf{x}_t \in A_t$, which guarantees that $\mathbf{z}_t \in \overline{A_t}$, as required. ■

Corollary 33 *Let D_n be a subset of $\mathbb{R}^n$ and let w be a weighting sequence. Then:*

(i) *If $(D_n)^{\mathbb{Z}_-} \cap \ell_-^w(\mathbb{R}^n)$ is an open subset of $\ell_-^w(\mathbb{R}^n)$ then $D_n = \mathbb{R}^n$, necessarily.*

(ii) *If $(D_n)^{\mathbb{Z}_-} \subset \ell_-^w(\mathbb{R}^n)$ is a closed subset of $\ell_-^w(\mathbb{R}^n)$ then D_n is necessarily closed in $\mathbb{R}^n$, that is, $D_n = \overline{D_n}$.*

(iii) *The following inclusion always holds*

$$\overline{(D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)} \subset \left(\overline{D_n}\right)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n). \tag{6.7}$$

In particular, if D_n is closed in $\mathbb{R}^n$ then so is $(D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)$ in $\ell^w_-(\mathbb{R}^n)$.

These statements in parts **(ii)** *and* **(iii)** *are also valid when the space $\ell^w_-(\mathbb{R}^n)$ is replaced by $\ell^\infty_-(\mathbb{R}^n)$.*

Proof. **(i)** We proceed by contradiction. Suppose that $D_n \neq \mathbb{R}^n$. Let $\mathbf{x}_0 \in \mathbb{R}^n \setminus D_n$ and let $\mathbf{z}_0 \in D_n$. Define the constant sequences $\mathbf{x} := (\mathbf{x}_0)_{t\in\mathbb{Z}_-} \in \ell^w_-(\mathbb{R}^n) \setminus \left((D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)\right)$ and $\mathbf{z} := (\mathbf{z}_0)_{t\in\mathbb{Z}_-} \in (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)$. Since by hypothesis $(D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)$ is an open subset of $\ell^w_-(\mathbb{R}^n)$ there exists $\epsilon > 0$ such that $B_{\|\cdot\|_w}(\mathbf{z}, 2\epsilon) \subset (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)$. By the relation (6.4) in Lemma 32 we also have

$$B_{\|\cdot\|_w}(\mathbf{z}, \epsilon) \subset \prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}\left(\mathbf{z}_0, \frac{\epsilon}{w_{-t}}\right) \subset \overline{B_{\|\cdot\|_w}(\mathbf{z}, \epsilon)} \subset B_{\|\cdot\|_w}(\mathbf{z}, 2\epsilon) \subset (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n),$$

and, in particular,

$$\prod_{t\in\mathbb{Z}_-} B_{\|\cdot\|}\left(\mathbf{z}_0, \frac{\epsilon}{w_{-t}}\right) \subset (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n), \text{ which implies } B_{\|\cdot\|}\left(\mathbf{z}_0, \frac{\epsilon}{w_{-t}}\right) \subset D_n, \text{ for all } t \in \mathbb{Z}_-. \tag{6.8}$$

Let $r_0 := \|\mathbf{x}_0 - \mathbf{z}_0\|$ and let $t_0 \in \mathbb{Z}_-$ be such that for all $t < t_0$ we have that $\epsilon/w_{-t_0} > r_0$. By (6.8) we have that $\mathbf{x}_0 \in B_{\|\cdot\|}\left(\mathbf{z}_0, \frac{\epsilon}{w_{-t}}\right) \subset D_n$, which contradicts the assumption on the choice of $\mathbf{x}_0$.

(ii) By Lemma 32 **(iii)** we have that

$$\overline{(D_n)^{\mathbb{Z}_-}} = \left(\overline{D_n}\right)^{\mathbb{Z}_-}. \tag{6.9}$$

Since by hypothesis $(D_n)^{\mathbb{Z}_-}$ is closed and hence it holds true that

$$\overline{(D_n)^{\mathbb{Z}_-}} = (D_n)^{\mathbb{Z}_-}. \tag{6.10}$$

Consequently, by (6.9) and (6.10) we have that $\left(\overline{D_n}\right)^{\mathbb{Z}_-} = (D_n)^{\mathbb{Z}_-}$ which implies that $\overline{D_n} = D_n$ as required.

(iii) Let $\mathbf{x} \in \overline{(D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)} \subset \ell^w_-(\mathbb{R}^n)$ and consider a sequence $\{\mathbf{x}^m\}_{m\in\mathbb{N}^+} \subset (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)$ with $\lim_{m\to\infty} \mathbf{x}^m = \mathbf{x}$, that is for each $\epsilon > 0$ there exists such $N(\epsilon) \in \mathbb{N}^+$ such that for all $m > N(\epsilon)$ it holds that $\|\mathbf{x}^m - \mathbf{x}\|_w < \epsilon$. Hence for all $s \in \mathbb{Z}_-$ one has that

$$w_{-s}\|\mathbf{x}^m_s - \mathbf{x}_s\| \leq \sup_{t\in\mathbb{Z}_-} \{\|\mathbf{x}^m_t - \mathbf{x}_t\| \, w_{-t}\} = \|\mathbf{x}^m - \mathbf{x}\|_w \leq \epsilon,$$

which immediately implies that

$$\|\mathbf{x}^m_s - \mathbf{x}_s\| < \frac{\epsilon}{w_{-s}}$$

and hence one gets that $\mathbf{x}_s \in \overline{D_n}$ and therefore (6.7) holds as required.
The last claim in part **(iii)** follows from (6.7). Indeed, if $D_n = \overline{D_n}$ then by (6.7) we have that

$$\overline{(D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)} \subset \left(\overline{D_n}\right)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n) = (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n).$$

Since the reverse inclusion obviously always holds, we finally have that

$$\overline{(D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n)} = (D_n)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^n). \quad \blacksquare$$

We also recall (see (Grigoryeva and Ortega, 2018b, Proposition 2.9)) that the norm topology in $\ell^w_-(\mathbb{R}^n)$ is strictly finer than the subspace topology induced by the product topology in $(\mathbb{R}^n)^{\mathbb{Z}_-}$ on $\ell^w_-(\mathbb{R}^n) \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$. We complement this fact by comparing the norm topology on $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$ with the relative topology induced by $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ on it.

Corollary 34 *The relative topology $\tau_{w,\infty}$ induced by the norm topology τ_w of $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ on $\ell^\infty_-(\mathbb{R}^n)$ is strictly coarser than the norm topology τ_∞ on $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$, that is, $\tau_{w,\infty} \subsetneq \tau_\infty$.*

Proof. Since, as we already saw, $\|\mathbf{z}\|_w \leq \|\mathbf{z}\|_\infty$, for all $\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-}$, we have that $\ell^\infty_-(\mathbb{R}^n) \subset \ell^w_-(\mathbb{R}^n)$ (see (2.4)) and the inclusion $\iota : \ell^\infty_-(\mathbb{R}^n) \hookrightarrow \ell^w_-(\mathbb{R}^n)$ is continuous. Consequently, for any open $W \in \tau_w$ the set $\iota^{-1}(W) = W \cap \ell^\infty_-(\mathbb{R}^n) \in \tau_{w,\infty}$ is also open in τ_∞. This immediately implies that

$$\tau_{w,\infty} \subset \tau_\infty.$$

In order to establish that this inclusion is strict, one needs to notice that, given an arbitrary open ball $B_{\|\cdot\|_\infty}(\mathbf{z}, r)$, $r > 0$, around $\mathbf{z} \in \ell^\infty_-(\mathbb{R}^n)$, all the open balls $B_{\|\cdot\|_w}(\mathbf{z}, \epsilon)$ for all $\epsilon > 0$ contain elements that are not included in $B_{\|\cdot\|_\infty}(\mathbf{z}, r)$ by Lemma 32 **(i)**. $\blacksquare$

Lemma 35 *Let w be a weighting sequence and $n \in \mathbb{N}^+$. We denote by w^a, $a \in \mathbb{R}$, the sequence with terms w^a_t, $t \in \mathbb{N}$. Then, the following inclusions are continuous:*

$$\left(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty\right) \hookrightarrow \cdots \hookrightarrow \left(\ell^{w^{\frac{1}{k+1}}}_-(\mathbb{R}^n), \|\cdot\|_{w^{\frac{1}{k+1}}}\right) \hookrightarrow \left(\ell^{w^{\frac{1}{k}}}_-(\mathbb{R}^n), \|\cdot\|_{w^{\frac{1}{k}}}\right) \hookrightarrow \cdots \hookrightarrow \left(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w\right), \tag{6.11}$$

$$\left(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w\right) \hookrightarrow \cdots \hookrightarrow \left(\ell^{w^k}_-(\mathbb{R}^n), \|\cdot\|_{w^k}\right) \hookrightarrow \left(\ell^{w^{k+1}}_-(\mathbb{R}^n), \|\cdot\|_{w^{k+1}}\right) \hookrightarrow \cdots \hookrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}, \tag{6.12}$$

where $k \in \mathbb{N}^+$ and in $(\mathbb{R}^n)^{\mathbb{Z}_-}$ we consider the trivial topology. Define

$$S_w := \bigcap_{k\in\mathbb{N}^+} \ell^{w^{\frac{1}{k}}}_-(\mathbb{R}^n) \quad \textit{and} \quad S^w := \bigcup_{k\in\mathbb{N}^+} \ell^{w^k}_-(\mathbb{R}^n). \tag{6.13}$$

Then, in general,

$$\ell^\infty_-(\mathbb{R}^n) \subsetneq S_w \quad \textit{and} \quad S^w \subsetneq (\mathbb{R}^n)^{\mathbb{Z}_-}. \tag{6.14}$$

Proof. The continuity of the inclusions (6.11) and (6.12) is a consequence of the fact that:

$$\|\mathbf{z}\|_{w^{\frac{1}{k}}} \leq \|\mathbf{z}\|_{w^{\frac{1}{k+1}}}, \quad \text{for all } k \in \mathbb{N}^+ \text{ and } \mathbf{z} \in \ell^{w^{\frac{1}{k}}}_-(\mathbb{R}^n), \tag{6.15}$$

$$\|\mathbf{z}\|_{w^{k+1}} \leq \|\mathbf{z}\|_{w^k}, \quad \text{for all } k \in \mathbb{N}^+ \text{ and } \mathbf{z} \in \ell^{w^k}_-(\mathbb{R}^n). \tag{6.16}$$

Regarding (6.14), the first inclusion follows from the fact that $\ell^\infty_-(\mathbb{R}^n) \subset \ell^w_-(\mathbb{R}^n)$ for any weighting sequence. In order to show that this inclusion is in general not an equality it suffices to consider the following example: let $\mathbf{z} \in (\mathbb{R})^{\mathbb{Z}_-}$ given by $z_t := -t$, $t \in \mathbb{Z}_-$, and let w be the weighting sequence defined by $w_t := \lambda^t$, with $t \in \mathbb{N}$ and $0 < \lambda < 1$. A simple application of the L'Hôpital rule shows that, for any $k \in \mathbb{N}^+$,

$$\lim_{t\to-\infty} z_t w_{-t}^{1/k} = 0,$$

which proves, in particular, that $\|\mathbf{z}\|_{w^{1/k}} < \infty$ and hence that $\mathbf{z} \in \ell^{w^{1/k}}_-(\mathbb{R})$, for any $k \in \mathbb{N}^+$. This implies that $\mathbf{z} \in S_w$. However, $\mathbf{z}$ is an unbounded sequence and hence it does not belong to $\ell^\infty_-(\mathbb{R})$. In order to show that the second inclusion in (6.14) is also strict, take $\mathbf{z} \in (\mathbb{R})^{\mathbb{Z}_-}$ given by $z_t := \lambda^{-t}$ with

$\lambda > 1$ and $t \in \mathbb{Z}_-$ and let w be the weighting sequence defined by $w_0 := 1$ and $w_t := \frac{1}{t}$, for any $t \in \mathbb{N}^+$. The L'Hôpital rule shows that, for any $k \in \mathbb{N}^+$,

$$\lim_{t \to -\infty} |z_t w_{-t}^k| = +\infty,$$

and consequently $\mathbf{z}$ does not belong to any of the spaces $\ell_-^{w^k}(\mathbb{R})$ and hence $\mathbf{z} \notin S^w$. ■

6.2. Products of continuous and differentiable functions using weighted norms

The following lemma spells out conditions under which infinite Cartesian products of continuous and differentiable functions are continuous and differentiable when we use weighted and supremum norms.

Lemma 36 *Let $W \subset V$ with $(V, \|\cdot\|)$ a normed space and let $D_N \subset \mathbb{R}^N$ be a subset of $\mathbb{R}^N$. Let $H_t : W \longrightarrow D_N$, $t \in \mathbb{Z}_-$, be a family of maps. Consider the corresponding product map $\mathcal{H} : W \longrightarrow (D_N)^{\mathbb{Z}_-}$, defined as in (6.2):*

$$\mathcal{H} := \prod_{t \in \mathbb{Z}_-} H_t := (\ldots, H_{-2}, H_{-1}, H_0), \text{ or equivalently, } (\mathcal{H}(\mathbf{z}))_t := H_t(\mathbf{z}), \ \mathbf{z} \in W, \ t \in \mathbb{Z}_-. \tag{6.17}$$

(i) *Endow $W \subset V$ with the subspace topology. If D_N is a compact subset of $\mathbb{R}^N$ then $(D_N)^{\mathbb{Z}_-} \subset \ell_-^w(\mathbb{R}^N)$ for any weighting sequence w. If each of the functions H_t is continuous then $\mathcal{H} : W \longrightarrow (D_N)^{\mathbb{Z}_-} \subset \ell_-^w(\mathbb{R}^N)$ is also continuous.*

(ii) *Let w be a weighting sequence and suppose that W contains a point $\mathbf{z}^0$ such that $\mathcal{H}(\mathbf{z}^0) \in \ell_-^w(\mathbb{R}^N)$. If each of the functions H_t is Lipschitz continuous with Lipschitz constant c_t^0 and the sequence $c^0 := (c_t^0)_{t \in \mathbb{Z}_-}$ formed by these Lipschitz constants satisfies that $c^0 \in \ell_-^w(\mathbb{R})$, then $\mathcal{H} : W \longrightarrow (D_N)^{\mathbb{Z}_-} \cap \ell_-^w(\mathbb{R}^N)$ is Lipschitz continuous with Lipschitz constant $c_{\mathcal{H}}^0 \leq \left\|c^0\right\|_w$.*

(iii) *Suppose that W is an open convex subset of $(V, \|\cdot\|)$ and that it contains a point $\mathbf{z}^0$ such that $\mathcal{H}(\mathbf{z}^0) \in \ell_-^w(\mathbb{R}^N)$. Suppose also that the maps H_t are of class $C^r(W)$, $r \geq 1$, and let c_t^r be finite constants such that $\sup_{\mathbf{z} \in W} \{|||D^r H_t(\mathbf{z})|||\} \leq c_t^r < +\infty$. If $c^r := (c_t^r)_{t \in \mathbb{Z}_-} \in \ell_-^w(\mathbb{R})$ then $\mathcal{H}$ is differentiable of order r when considered as a map $\mathcal{H} : W \subset (V, \|\cdot\|) \longrightarrow \left(\ell_-^w(\mathbb{R}^N), \|\cdot\|_w\right)$ and*

$$|||D^r \mathcal{H}(\mathbf{z})||| \leq \|c^r\|_w, \text{ for any } \mathbf{z} \in W. \tag{6.18}$$

Additionally, if $c^j \in \ell_-^w(\mathbb{R})$ for all $j \in \{1, \ldots, r\}$, then $\mathcal{H}$ is of class $C^{r-1}(W)$ and the map $D^{r-1}\mathcal{H} : (W, \|\cdot\|) \longrightarrow \left(L^{r-1}(V, \ell_-^w(\mathbb{R}^N)), |||\cdot|||\right)$ is Lipschitz continuous with Lipschitz constant $c_{\mathcal{H}}^r \leq \|c^r\|_w$.

(iv) *Suppose that W is an open convex subset of $(V, \|\cdot\|)$ and that it contains a point $\mathbf{z}^0$ such that $\mathcal{H}(\mathbf{z}^0) \in \ell_-^w(\mathbb{R}^N)$. If the maps H_t are smooth and $\|c^r\|_w < +\infty$, for each $r \in \mathbb{N}^+$, then so is $\mathcal{H} : W \subset (V, \|\cdot\|) \longrightarrow (\ell_-^w(\mathbb{R}^N), \|\cdot\|_w)$. Suppose, additionally, that the maps H_t are analytic and that $\rho_t > 0$ is the radius of convergence of the series expansion of H_t. If $\rho := \inf_{t \in \mathbb{Z}_-} \{\rho_t\} > 0$ then $\mathcal{H}$ is analytic when considered as a map $\mathcal{H} : W \subset (V, \|\cdot\|) \longrightarrow (\ell_-^w(\mathbb{R}^N), \|\cdot\|_w)$ and the radius of convergence $\rho_{\mathcal{H}}$ of its series expansion satisfies that $\rho_{\mathcal{H}} \geq \rho > 0$.*

Parts **(ii)**, **(iii)**, *and* **(iv)** *also hold true when the Banach space $(\ell_-^w(\mathbb{R}^N), \|\cdot\|_w)$ is replaced by $(\ell_-^\infty(\mathbb{R}^N), \|\cdot\|_\infty)$. Part* **(i)** *is in general false in that situation.*

Proof. **(i)** The compactness of D_N guarantees (Munkres, 2014, Theorem 27.3) that there exists $L > 0$ such that $D_N \subset \overline{B_{\|\cdot\|}(\mathbf{0}, L)}$ and hence $(D_N)^{\mathbb{Z}_-} \subset \ell_-^w(\mathbb{R}^N)$ necessarily. It can also be shown (see

(Grigoryeva and Ortega, 2018b, Corollary 2.7)) that when D_N is compact, the relative topology $(D_N)^{\mathbb{Z}_-}$ induced by the weighted norm $\|\cdot\|_w$ in $\ell_-^w(\mathbb{R}^N)$ coincides with the product topology. This implies (see (Munkres, 2014, Theorem 19.6)) that if the functions H_t are continuous then so is $\mathcal{H}$.

(ii) Let $\mathbf{z}^1, \mathbf{z}^2 \in W$. Then,

$$\left\|\mathcal{H}(\mathbf{z}^1) - \mathcal{H}(\mathbf{z}^2)\right\|_w = \sup_{t \in \mathbb{Z}_-} \left\{\left\|H_t(\mathbf{z}^1) - H_t(\mathbf{z}^2)\right\| w_{-t}\right\} \leq \sup_{t \in \mathbb{Z}_-} \left\{c_t^0 \left\|\mathbf{z}^1 - \mathbf{z}^2\right\| w_{-t}\right\} \leq \|c^0\|_w \left\|\mathbf{z}^1 - \mathbf{z}^2\right\|, \tag{6.19}$$

which proves simultaneously that $\mathcal{H}$ is Lipschitz continuous and that it maps into $\ell_-^w(\mathbb{R}^N)$. Regarding the last point, recall that by hypothesis there exists a point $\mathbf{z}^0$ such that $\mathcal{H}(\mathbf{z}^0) \in \ell_-^w(\mathbb{R}^N)$ and hence by (6.19) we have, for any $\mathbf{z} \in W$,

$$\|\mathcal{H}(\mathbf{z})\|_w \leq \|c^0\|_w \left\|\mathbf{z} - \mathbf{z}^0\right\| + \left\|\mathcal{H}(\mathbf{z}^0)\right\|_w < +\infty. \tag{6.20}$$

(iii) First, it is easy to prove recursively that for any $\mathbf{z} \in W$, the map $D^r\mathcal{H}(\mathbf{z}) := \prod_{t \in \mathbb{Z}_-} D^r H_t(\mathbf{z})$ satisfies the condition (2.5). In order to prove the first statement of the lemma, it suffices to show that the multilinear map

$$D^r\mathcal{H}(\mathbf{z}) : \underbrace{(V, \|\cdot\|) \times \cdots \times (V, \|\cdot\|)}_{r \text{ times}} \longrightarrow (\ell_-^w(\mathbb{R}^n), \|\cdot\|_w),$$

is bounded for any $\mathbf{z} \in W$. Let $\left(\mathbf{v}^1, \ldots, \mathbf{v}^r\right) \in V^r$. Using the r-order differentiability of H_t we can write

$$\begin{aligned}
\left\|D^r\mathcal{H}(\mathbf{z}) \cdot \left(\mathbf{v}^1, \ldots, \mathbf{v}^r\right)\right\|_w &= \left\|\prod_{t \in \mathbb{Z}_-} D^r H_t(\mathbf{z}) \cdot \left(\mathbf{v}^1, \ldots, \mathbf{v}^r\right)\right\|_w = \sup_{t \in \mathbb{Z}_-} \left\{\left\|D^r H_t(\mathbf{z}) \cdot \left(\mathbf{v}^1, \ldots, \mathbf{v}^r\right)\right\| w_{-t}\right\} \\
&\leq \sup_{t \in \mathbb{Z}_-} \left\{|||D^r H_t(\mathbf{z})||| \left\|\mathbf{v}^1\right\| \cdots \left\|\mathbf{v}^r\right\| w_{-t}\right\} \\
&\leq \left\|\mathbf{v}^1\right\| \cdots \left\|\mathbf{v}^r\right\| \sup_{t \in \mathbb{Z}_-} \left\{c_t^r w_{-t}\right\} \leq \|c^r\|_w \left\|\mathbf{v}_1\right\| \cdots \left\|\mathbf{v}_r\right\|,
\end{aligned} \tag{6.21}$$

which proves the boundedness of $D^r\mathcal{H}(\mathbf{z})$ and the inequality in (6.18).

We now assume that $c^j \in \ell_-^w(\mathbb{R})$ for all $j \in \{1, \ldots, r\}$ and show that $\mathcal{H}$ maps into $\ell_-^w(\mathbb{R}^N)$ and that it is of class $C^{r-1}(W)$. Notice, first of all, that for any $t \in \mathbb{Z}_-$ and any $\mathbf{z}^1, \mathbf{z}^2 \in V_n$, we have by the convexity of W, the mean value theorem Abraham et al. (1988), and the hypothesis $H_t \in C^r(W)$, that for all $j \in \{1, \ldots, r\}$,

$$\left|\left|\left|D^{j-1}H_t(\mathbf{z}^1) - D^{j-1}H_t(\mathbf{z}^2)\right|\right|\right|_w \leq \sup_{\mathbf{z} \in W} \left\{|||D^j H_t(\mathbf{z})|||\right\} \left\|\mathbf{z}^1 - \mathbf{z}^2\right\| = c_t^j \left\|\mathbf{z}^1 - \mathbf{z}^2\right\|. \tag{6.22}$$

Taking $j = 1$ in the previous inequality, we see that the functions H_t are Lipschitz continuous with constants c_t^1 that form a sequence that by hypothesis belongs to $\ell_-^w(\mathbb{R})$. This guarantees by part **(ii)** that $\mathcal{H}$ maps into $\ell_-^w(\mathbb{R}^N)$ necessarily. Now, using the inequality (6.22), we have that for any $\mathbf{z}^1, \mathbf{z}^2 \in W$,

$$\begin{aligned}
&\left|\left|\left|D^{r-1}\mathcal{H}(\mathbf{z}^1) - D^{r-1}\mathcal{H}(\mathbf{z}^2)\right|\right|\right|_w \\
&\quad = \sup_{\substack{\mathbf{v}^1, \ldots, \mathbf{v}^{r-1} \in V \\ \mathbf{v}^1, \ldots, \mathbf{v}^{r-1} \neq \mathbf{0}}} \left\{\frac{\left\|\left(D^{r-1}\mathcal{H}(\mathbf{z}^1) - D^{r-1}\mathcal{H}(\mathbf{z}^2)\right) \cdot \left(\mathbf{v}^1, \ldots, \mathbf{v}^{r-1}\right)\right\|_w}{\left\|\mathbf{v}^1\right\| \cdots \left\|\mathbf{v}^{r-1}\right\|}\right\} \\
&\quad = \sup_{\substack{\mathbf{v}^1, \ldots, \mathbf{v}^{r-1} \in V \\ \mathbf{v}^1, \ldots, \mathbf{v}^{r-1} \neq \mathbf{0}}} \left\{\frac{\sup_{t \in \mathbb{Z}_-} \left\{\left\|\left(D^{r-1}H_t(\mathbf{z}^1) - D^{r-1}H_t(\mathbf{z}^2)\right) \cdot \left(\mathbf{v}^1, \ldots, \mathbf{v}^{r-1}\right)\right\| w_{-t}\right\}}{\left\|\mathbf{v}^1\right\| \cdots \left\|\mathbf{v}^{r-1}\right\|}\right\} \\
&\quad \leq \sup_{\substack{\mathbf{v}^1, \ldots, \mathbf{v}^{r-1} \in V \\ \mathbf{v}^1, \ldots, \mathbf{v}^{r-1} \neq \mathbf{0}}} \left\{\frac{\sup_{t \in \mathbb{Z}_-} \left\{c_t^r w_{-t} \left\|\mathbf{z}^1 - \mathbf{z}^2\right\| \cdot \left\|\mathbf{v}^1\right\| \cdots \left\|\mathbf{v}^{r-1}\right\|\right\}}{\left\|\mathbf{v}^1\right\| \cdots \left\|\mathbf{v}^{r-1}\right\|}\right\} = \|c^r\|_w \left\|\mathbf{z}^1 - \mathbf{z}^2\right\|,
\end{aligned} \tag{6.23}$$

which shows that the map $D^{r-1}\mathcal{H} : (W, \|\cdot\|_w) \longrightarrow \left(L^{r-1}(\ell^w_-(\mathbb{R}^n), \ell^w_-(\mathbb{R}^N)), \|\cdot\|_w\right)$ is Lipschitz continuous with Lipschitz constant $c^r_{\mathcal{H}} \leq \|c^r\|_w$.

(iv) The previous part of the lemma together with the hypothesis $c := \sup_{r\in\mathbb{N}^+}\{\|c^r\|_w\} < +\infty$ guarantees that the differentiability of any order in the functions H_t gets translated into the differentiability of any order of the map $\mathcal{H} : W \subset (V, \|\cdot\|) \longrightarrow (\ell^w_-(\mathbb{R}^N), \|\cdot\|_w)$. Moreover, let $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ and let $\mathbf{u}^r := (\mathbf{u}, \ldots, \mathbf{u}) \in \left(\ell^w_-(\mathbb{R}^n)\right)^r$, $r \in \mathbb{N}^+$. The Taylor series expansion of $\mathcal{H}$ around $\mathbf{0} \in$ is

$$\mathcal{H}(\mathbf{0}) + \sum_{r=1}^{\infty}\frac{1}{r!}D^r\mathcal{H}(\mathbf{0})\cdot\mathbf{u}^r = \prod_{t\in\mathbb{Z}_-}\left(H_t(\mathbf{0}) + \sum_{r=1}^{\infty}\frac{1}{r!}D^rH_t(\mathbf{0})\cdot\mathbf{u}^r\right). \tag{6.24}$$

The expansion in the left hand side of this equality is convergent if and only if each of the series in the product in the right hand side is convergent. This is the case when $\|\mathbf{u}\|_w < \rho_t$, for all $t \in \mathbb{Z}_-$, which guarantees the convergence of the Taylor series expansion in (6.24) for all the elements $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ that satisfy $\|\mathbf{u}\|_w < \inf_{t\in\mathbb{Z}_-}\{\rho_t\} = \rho$. Since by hypothesis $\rho > 0$, we have proved that $\mathcal{H}$ is analytic with radius of convergence $\rho_{\mathcal{H}} \geq \rho$.

The proof of the statements in **(ii)**, **(iii)**, and **(iv)** for the space $\ell^\infty_-(\mathbb{R}^N)$ is obtained by mimicking the proofs that we just provided, replacing the weighting sequence w by the constant sequence w^t that is equal to 1 for each $t \in \mathbb{Z}_-$. In order to show that part **(i)** is in general false in that situation take $W = (-1,1)$, $D_N = [-1,1]$, and define $H_t(z) := \tanh(-tz)$, with $t \in \mathbb{Z}$ and $z \in (-1,1)$. Given that $H_t^{-1}\left(-\frac{1}{2}, \frac{1}{2}\right) = \left(-\frac{1}{t}\tanh^{-1}(-\frac{1}{2}), -\frac{1}{t}\tanh^{-1}(\frac{1}{2})\right)$ it is clear that

$$\mathcal{H}^{-1}\left(B_{\|\cdot\|_\infty}\left(\mathbf{0}, \frac{1}{2}\right)\right) = \bigcap_{t\in\mathbb{Z}_-}\left(-\frac{1}{t}\tanh^{-1}\left(-\frac{1}{2}\right), -\frac{1}{t}\tanh^{-1}\left(\frac{1}{2}\right)\right) = \{0\}.$$

This equality shows that the preimage by the product map $\mathcal{H}$ of an open set is not open and hence $\mathcal{H}$ is not continuous. ■

6.3. Proof of Lemma 1

(i) The linearity of p_t is obvious. Let $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ arbitrary. Since $\|p_t(\mathbf{u})\| = \|\mathbf{u}_t\| = \|\mathbf{u}_t\|\, w_{-t}/w_{-t} \leq \sup_{j\in\mathbb{Z}_-}\{\|\mathbf{u}_j\|\, w_{-j}\}/w_{-t} = \|\mathbf{u}\|_w/w_{-t}$, we can conclude that $\|p_t\|_w \leq 1/w_{-t}$. Let now $\mathbf{v} \in \mathbb{R}^n$ such that $\|\mathbf{v}\| = 1$ and define the element $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$ by $\mathbf{z}_t := \mathbf{v}/w_{-t}$, for all $t \in \mathbb{Z}_-$. It is clear that $\|\mathbf{z}\|_w = 1$ and that $\|p_t(\mathbf{z})\| / \|\mathbf{z}\|_w = 1/w_{-t}$, which shows that $\|p_t\|_w = 1/w_{-t}$, as required.

(ii) We first prove the statements in this part in the case $t < 0$. Suppose that the inverse decay ratio L_w is finite and let $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ arbitrary. Then

$$\begin{aligned}\|T_1(\mathbf{u})\|_w &= \sup_{t\in\mathbb{Z}_-}\{\|\mathbf{u}_{t-1}\|\, w_{-t}\} = \sup_{t\in\mathbb{Z}_-}\left\{\|\mathbf{u}_{t-1}\|\, w_{-(t-1)}\frac{w_{-t}}{w_{-(t-1)}}\right\} \\ &\leq \sup_{t\in\mathbb{Z}_-}\left\{\|\mathbf{u}_{t-1}\|\, w_{-(t-1)}\right\}\sup_{t\in\mathbb{Z}_-}\left\{\frac{w_{-t}}{w_{-(t-1)}}\right\} \leq \|\mathbf{u}\|_w\, L_w. \end{aligned}\tag{6.25}$$

This inequality shows that T_1 maps $\ell^w_-(\mathbb{R}^n)$ into $\ell^w_-(\mathbb{R}^n)$ and that $\|T_1\|_w \leq L_w$. Given that for any $t \in \mathbb{Z}_-$ we can write

$$T_{-t} = \underbrace{T_1 \circ \cdots \circ T_1}_{-t \text{ times}},$$

the previous conclusion also proves that T_{-t} maps $\ell^w_-(\mathbb{R}^n)$ into $\ell^w_-(\mathbb{R}^n)$ and that

$$\|T_{-t}\|_w = \|T_1 \circ \cdots \circ T_1\|_w \leq \|T_1\|_w \cdots \|T_1\|_w \leq L_w^{-t}.$$

It remains to be shown that $\|T_1\|_w = L_w$. In order to do so, take an element $\mathbf{v} \in \mathbb{R}^n$ such that $\|\mathbf{v}\| = 1$ and define the element $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ by $\mathbf{u}_t := \mathbf{v}/w_{-t}$, for all $t \in \mathbb{Z}_-$. Notice that by construction $\|\mathbf{u}\|_w = 1$ and, moreover,

$$\frac{\|T_1(\mathbf{u})\|_w}{\|\mathbf{u}\|_w} = \sup_{t\in\mathbb{Z}_-}\{\|\mathbf{u}_{t-1}\|\, w_{-t}\} = \sup_{t\in\mathbb{Z}_-}\left\{\frac{\|\mathbf{v}\|}{w_{-(t-1)}}w_{-t}\right\} = L_w,$$

which proves the required identity.

We now show that $T_{-t} : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is surjective. Indeed, it is clear that for any $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$, the element

$$\widetilde{\mathbf{u}} := (\mathbf{u}, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{-t \text{ times}}) \quad \text{is such that} \quad T_{-t}(\widetilde{\mathbf{u}}) = \mathbf{u}.$$

We hence just need to show that $\widetilde{\mathbf{u}} \in \ell^w_-(\mathbb{R}^n)$. This is the case because

$$\begin{aligned}\|\widetilde{\mathbf{u}}\|_w &= \sup_{s\in\mathbb{Z}_-}\{\|\widetilde{\mathbf{u}_s}\|\, w_{-s}\} = \sup_{s\in\mathbb{Z}_-}\{\|\mathbf{u}_{s+t}\|\, w_{-s}\} = \sup_{s\in\mathbb{Z}_-}\left\{\|\mathbf{u}_{s+t}\|\, w_{-(s+t)}\frac{w_{-s}}{w_{-(s+t)}}\right\} \\ &= \sup_{s\in\mathbb{Z}_-}\left\{\|\mathbf{u}_{s+t}\|\, w_{-(s+t)}\frac{w_{-s}}{w_{-(s+1)}}\frac{w_{-(s+1)}}{w_{-(s+2)}}\cdots\frac{w_{-(s+t-1)}}{w_{-(s+t)}}\right\} \leq \|\mathbf{u}\|_w\, L_w^{-t} < +\infty, \end{aligned}\tag{6.26}$$

because $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ and by hypothesis $L_w < +\infty$. Now, since we already showed that $T_{-t} : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is continuous, then the Banach-Schauder Open Mapping Theorem (Abraham et al., 1988, Theorem 2.2.15) implies that T_{-t} is necessarily an open map.

It remains to be shown that $T_{-t} : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$ is a submersion (see (Abraham et al., 1988, Section 3.5) for context and definitions). First, it is obvious that

$$\ker T_{-t} = \left\{(\ldots, \mathbf{0}, \mathbf{0}, \mathbf{v}) \mid \mathbf{v} \in (\mathbb{R}^n)^{-t}\right\}.$$

Since T_{-t} is linear and bounded, in order to show that it is a submersion it suffices to show that $\ker T_{-t}$ is split, that is, it has a closed complement in $(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w)$. We now prove that such a complement is given by the subspace

$$C_{-t} := \left\{(\mathbf{u}, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{-t \text{ times}}) \mid \mathbf{u} \in \ell^w_-(\mathbb{R}^n)\right\}. \tag{6.27}$$

The inequality (6.26) implies that

$$C_{-t} \subset \ell^w_-(\mathbb{R}^n). \tag{6.28}$$

Additionally, C_{-t} is clearly closed in $\ell^w_-(\mathbb{R}^n)$. We conclude by showing by double inclusion that

$$\ell^w_-(\mathbb{R}^n) = \ker T_{-t} \oplus C_{-t}. \tag{6.29}$$

Let first $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ and define

$$\mathbf{u}_1 := (\ldots, \mathbf{u}_{t-2}, \mathbf{u}_{t-1}, \mathbf{u}_t, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{-t \text{ times}}) \quad \text{and} \quad \mathbf{u}_2 := (\ldots, \mathbf{0}, \mathbf{0}, \mathbf{u}_{t+1}, \mathbf{u}_{t+2}, \ldots, \mathbf{u}_0).$$

It is clear that $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$. Additionally, the sequence $\mathbf{u}_2$ is obviously in $\ker T_{-t}$ and using an argument similar to the one in (6.26) it is easy to show that $\mathbf{u}_1 \in C_{-t}$, which proves the inclusion $\ell^w_-(\mathbb{R}^n) \subseteq \ker T_{-t} \oplus C_{-t}$.

Conversely, let $\mathbf{u}_1 \in C_{-t}$ and $\mathbf{u}_2 \in \ker T_{-t}$. By (6.28) we have that $\|\mathbf{u}_1\|_w < +\infty$ and it is also clear that $\|\mathbf{u}_2\|_w < +\infty$. Therefore $\|\mathbf{u}_1 + \mathbf{u}_2\|_w \leq \|\mathbf{u}_1\|_w + \|\mathbf{u}_2\|_w < +\infty$ and hence $\mathbf{u}_1 + \mathbf{u}_2 \in \ell^w_-(\mathbb{R}^n)$, which shows that T_{-t} is a submersion.

Finally, the statements in the case $t > 0$ are proved in a similar fashion. In particular, it is easy to see that

$$T_{-t} \circ T_t = \mathbb{P}_{C_t}, \quad \text{for any } t > 0,$$

where $\mathbb{P}_{C_t}$ is the projection onto the subspace C_t defined in (6.27) according to the splitting (6.29). Moreover, it is easy to see that T_{-t} is injective and that its image $\operatorname{Im} T_{-t}$ is split because $\operatorname{Im} T_{-t} = C_t$ and by (6.29)

$$\ell^w_-(\mathbb{R}^n) = \ker T_t \oplus \operatorname{Im} T_{-t}, \quad t > 0,$$

which proves that T_{-t} is an immersion.

(iii) Straightforward consequence of the definitions.

The proofs for the space $(\ell^\infty_-(\mathbb{R}^n), \|\cdot\|_\infty)$ can be obtained by replacing in the previous arguments the weighting sequence w by the constant sequence w^ι. ■

6.4. Equivalence of FMP and differentiability in filters and functionals

The facts established in Lemma 1 can be used to show the equivalence between the continuity and the differentiability of causal and time-invariant filters and that of their associated functionals. The following result focuses on continuity and the fading memory property and generalizes to the context of eventually unbounded inputs the equivalence between fading memory filters and functionals established in (Grigoryeva and Ortega, 2018b, Propositions 2.11 and 2.12) for uniformly bounded inputs. In the results that follow we work in a setup slightly more general than the one that is customary in the literature as we will allow for the weighting sequences considered in the domain and the target of the filters to be different. This degree of generality is needed later on in the text.

Proposition 37 *Let $V_n \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$ and $V_N \subset \left(\mathbb{R}^N\right)^{\mathbb{Z}_-}$ be time-invariant subsets and let $D_N \subset \mathbb{R}^N$. Let w^1, w^2 be weighting sequences with inverse decay ratios L_{w^1} and L_{w^2}, respectively.*

(i) *Let $U : V_n \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow V_N \subset \ell^{w^2}_-(\mathbb{R}^N)$ be a causal and time-invariant filter. If U has the fading memory property then so does its associated functional $H_U : V_n \longrightarrow p_0(V_N)$. The same conclusion holds for continuous filters $U : V_n \subset \ell^\infty_-(\mathbb{R}^n) \longrightarrow V_N \subset \ell^\infty_-(\mathbb{R}^N)$.*

(ii) *Let $H : V_n \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow D_N$ be a fading memory functional. If L_{w^1} is finite and D_N is compact then the associated causal and time-invariant filter $U_H : V_n \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow (D_N)^{\mathbb{Z}_-} \subset \ell^{w^2}_-(\mathbb{R}^N)$ has also the fading memory property.*

(iii) *Let $H : V_n \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow D_N$ be a fading memory functional and suppose that V_n contains a point $\mathbf{z}^0$ such that $U_H(\mathbf{z}^0) \in \ell^{w^2}_-(\mathbb{R})$, where U_H is the causal and time-invariant filter associated to H. If H is Lipschitz, c_H is a Lipschitz constant, and the weighting sequences satisfy one of the following two conditions*

$$\text{either } R_{w^1,w^2} := \sup_{s,t\in\mathbb{N}} \left\{ \frac{w^1_t w^2_s}{w^1_{t+s}} \right\} < +\infty \text{ or the sequence } \mathcal{L}_{w^1} := \left(L^{-t}_{w^1}\right)_{t\in\mathbb{Z}_-} \in \ell^{w^2}_-(\mathbb{R}), \tag{6.30}$$

then $U_H : V_n \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow (D_N)^{\mathbb{Z}_-} \cap \ell^{w^2}_-(\mathbb{R}^N)$ has also the fading memory property, it is Lipschitz, and $R_{w^1,w^2} c_H$ or $\|\mathcal{L}_{w^1}\|_{w^2} c_H$, respectively, is a Lipschitz constant of U_H. The same conclusion holds for continuous functionals $H : V_n \subset \ell^\infty_-(\mathbb{R}^n) \longrightarrow D_N$ where the condition (6.30) is not needed.

Proof. **(i)** As H_U is given by $H_U = p_0 \circ U$, the FMP (respectively, continuity) of U and the first part of Lemma 1 prove the statement.

(ii) Notice first that as

$$U_H = \prod_{t\in\mathbb{Z}_-} H \circ T_{-t} \tag{6.31}$$

then, as L_{w^1} is finite, U_H is by the second part of Lemma 1 the Cartesian product of continuous functions $H_t := H \circ T_{-t} : V_n \subset \ell^{w^1}_-(\mathbb{R}^n) \longrightarrow D_N$. Since D_N is by hypothesis compact, the result follows from the first part of Lemma 36.

(iii) Let $\mathbf{z}^1, \mathbf{z}^2 \in V_n$ arbitrary. Then by (6.31) and the Lipschitz hypothesis on H, we have that

$$\begin{aligned}\left\|U_H(\mathbf{z}^1) - U_H(\mathbf{z}^2)\right\|_{w^2} &= \sup_{t\in\mathbb{Z}_-} \left\{ \left\|H(T_{-t}(\mathbf{z}^1)) - H(T_{-t}(\mathbf{z}^2))\right\| w^2_{-t} \right\} \\ &\leq c_H \sup_{t\in\mathbb{Z}_-} \left\{ \left\|T_{-t}(\mathbf{z}^1) - T_{-t}(\mathbf{z}^2)\right\|_{w^1} w^2_{-t} \right\}. \end{aligned} \tag{6.32}$$

If the first condition in (6.30) is satisfied, this expression is bounded above by

$$\begin{aligned} c_H \sup_{t,s\in\mathbb{Z}_-} \left\{ \left\|T_{-t}(\mathbf{z}^1)_s - T_{-t}(\mathbf{z}^2)_s\right\| w^2_{-t} w^1_{-s} \right\} &= c_H \sup_{t,s\in\mathbb{Z}_-} \left\{ \left\|\mathbf{z}^1_{t+s} - \mathbf{z}^2_{t+s}\right\| w^1_{-(t+s)} \frac{w^2_{-t} w^1_{-s}}{w^1_{-(t+s)}} \right\} \\ &\leq R_{w^1,w^2} c_H \left\|\mathbf{z}^1 - \mathbf{z}^2\right\|_{w^1} \end{aligned}$$

which proves that in that case U_H has the fading memory property, it is Lipschitz, and $R_{w^1,w^2} c_H$ is a Lipschitz constant. If the second condition in (6.30) is satisfied then the inverse decay ratio L_{w^1} is necessarily finite and hence (6.32) can be bounded using the second part of Lemma 1 as

$$c_H \sup_{t\in\mathbb{Z}_-} \left\{ \left\|T_{-t}(\mathbf{z}^1) - T_{-t}(\mathbf{z}^2)\right\|_{w^1} w^2_{-t} \right\} \leq c_H \left\|\mathbf{z}^1 - \mathbf{z}^2\right\|_{w^1} \sup_{t\in\mathbb{Z}_-} \left\{ L^{-t}_{w^1} w^2_{-t} \right\} = \|\mathcal{L}_{w^1}\|_{w^2} c_H \left\|\mathbf{z}^1 - \mathbf{z}^2\right\|_{w^1},$$

which proves that in that case $\|\mathcal{L}_{w^1}\|_{w^2} c_H$ is a Lipschitz constant of U_H. The Lipschitz continuity of U_H together with the hypothesis on the existence of a point $\mathbf{z}^0$ such that $U_H(\mathbf{z}^0) \in \ell^{w^2}_-(\mathbb{R})$ guarantee that U_H maps into $\ell^{w^2}_-(\mathbb{R}^N)$ using a strategy similar to the one followed in (6.20).

The proof for the spaces $\ell^\infty_-(\mathbb{R}^n)$ and $\ell^\infty_-(\mathbb{R}^N)$ is obtained by taking as weighting sequences the constant sequence w^ι given by $w^t_t := 1$, for all $t \in \mathbb{N}$, that automatically satisfies any of the two conditions in (6.30). ■

Remark 38 When in part **(iii)** we consider the same weighting sequence w for the domain and the target, it is easy to see that

$$R_w := \sup_{s,t\in\mathbb{N}} \left\{ \frac{w_t w_s}{w_{t+s}} \right\}$$

satisfies that $R_w \leq \|\mathcal{L}_w\|_w$ and therefore the second condition in (6.30) implies the first one. Indeed,

$$R_w = \sup_{s,t\in\mathbb{N}} \left\{ \frac{w_t w_s}{w_{t+s}} \right\} = \sup_{s,t\in\mathbb{N}} \left\{ \frac{w_t}{w_{t+1}} \frac{w_{t+1}}{w_{t+2}} \cdots \frac{w_{t+s-1}}{w_{t+s}} w_s \right\} \leq \sup_{s\in\mathbb{N}} \left\{ L^s_w w_s \right\} = \|\mathcal{L}_w\|_w, \quad \text{as required.}$$

In this setup, the condition (6.30) is satisfied by many families of commonly used weighting sequences. In the two examples considered in Remark 2 we have that $R_w = \|\mathcal{L}_w\|_w = 1$ for the geometric sequence; for the harmonic sequence $\|\mathcal{L}_w\|_w = +\infty$ but $R_w = 1$ and hence (6.30) is still satisfied.

We emphasize that condition (6.30) is not automatically satisfied by all weighting sequences. For example, as we saw in Remark 2, the sequence $w_t := \exp(-t^2)$ is such that $L_w = +\infty$ and, additionally, it is easy to see that $R_w := \sup_{s,t\in\mathbb{N}} \{\exp(2st)\} = +\infty$.

Proposition 39 *Let w^1 and w^2 be two weighting sequences with inverse decay ratios L_{w^1} and L_{w^1}, respectively. Let $V_n \subset \ell_-^{w^1}(\mathbb{R}^n)$ and $V_N \subset \ell_-^{w^2}(\mathbb{R}^N)$ be time-invariant open subsets, and let D_N be an open subset of $\mathbb{R}^N$.*

(i) *Let $U : V_n \subset \ell_-^{w^1}(\mathbb{R}^n) \longrightarrow V_N \subset \ell_-^{w^2}(\mathbb{R}^N)$ be a causal and time-invariant filter. If U is of class $C^r(V_n)$ (respectively, smooth or analytic) when considered as a map $U : V_n \subset \left(\ell_-^{w^1}(\mathbb{R}^n), \|\cdot\|_w\right) \longrightarrow V_N \subset \left(\ell_-^{w^2}(\mathbb{R}^N), \|\cdot\|_w\right)$, then so is the associated functional $H_U : V_n \subset \left(\ell_-^{w^1}(\mathbb{R}^n), \|\cdot\|_w\right) \longrightarrow p_0(V_N) \subset \mathbb{R}^N$. Moreover,*

$$\left\|\!\left|D^r H_U(\mathbf{z})\right|\!\right\|_{w^1} \leq \left\|\!\left|D^r U(\mathbf{z})\right|\!\right\|_{w^1,w^2}, \quad \text{for any } \mathbf{z} \in V_n. \tag{6.33}$$

The same conclusion holds when the weighted sequence spaces are replaced by $\left(\ell_-^\infty(\mathbb{R}^n), \|\cdot\|_\infty\right)$ and $\left(\ell_-^\infty(\mathbb{R}^N), \|\cdot\|_\infty\right)$.

(ii) *Let $H : V_n \subset \ell_-^{w^1}(\mathbb{R}^n) \longrightarrow D_N$ be a functional and suppose that V_n is convex and contains a point $\mathbf{z}^0$ such that $U_H(\mathbf{z}^0) \in \ell_-^{w^2}(\mathbb{R})$, where U_H is the causal and time-invariant filter associated to H. If the functional H is of class $C^r(V_n)$ and for any $j \in \{1, \ldots, r\}$ we have that $c^j := \sup_{\mathbf{z}\in V_n} \left\{\left\|\!\left|D^j H(\mathbf{z})\right|\!\right\|_{w^1}\right\} < +\infty$ and the weighting sequences satisfy that*

$$\mathcal{L}_{w^1,j} := (L_{w^1}^{-jt})_{t\in\mathbb{Z}_-} \in \ell_-^{w^2}(\mathbb{R}), \tag{6.34}$$

then the associated causal and time-invariant filter U_H is differentiable of order r when considered as a map $U_H : V_n \subset \left(\ell_-^{w^1}(\mathbb{R}^n), \|\cdot\|_{w^1}\right) \longrightarrow (D_N)^{\mathbb{Z}_-} \cap \left(\ell_-^{w^2}(\mathbb{R}^N), \|\cdot\|_{w^2}\right)$. Moreover, for any $\mathbf{z} \in V_n$,

$$\left\|\!\left|D^r U_H(\mathbf{z})\right|\!\right\|_{w^1,w^2} \leq c^r \|\mathcal{L}_{w^1,r}\|_{w^2}. \tag{6.35}$$

Additionally, U_H is of class $C^{r-1}(V_n)$ and the map

$$D^{r-1}U_H : (V_n, \|\cdot\|_{w^1}) \longrightarrow \left(L^{r-1}\left(\ell_-^{w^1}(\mathbb{R}^n), \ell_-^{w^2}(\mathbb{R}^N)\right), \|\!|\cdot|\!\|_{w^1,w^2}\right)$$

is Lipschitz continuous with Lipschitz constant $c^r\|\mathcal{L}_{w^1,r}\|_{w^2}$. The same conclusion holds when the weighted sequence spaces are replaced by $\left(\ell_-^\infty(\mathbb{R}^n), \|\cdot\|_\infty\right)$ and $\left(\ell_-^\infty(\mathbb{R}^N), \|\cdot\|_\infty\right)$. In that case the inequality (6.35) holds with $\|\mathcal{L}_{w^1,r}\|_{w^2} = 1$.

(iii) *Let $H : V_n \subset \ell_-^{w^1}(\mathbb{R}^n) \longrightarrow D_N$ be a functional and suppose that V_n is convex and contains a point $\mathbf{z}^0$ such that $U_H(\mathbf{z}^0) \in \ell_-^{w^2}(\mathbb{R})$, where U_H is the causal and time-invariant filter associated to H. If the functional H is smooth and $c^r < +\infty$ for all $r \in \mathbb{N}^+$, then so is the associated causal and time-invariant filter $U_H : V_n \subset \left(\ell_-^{w^1}(\mathbb{R}^n), \|\cdot\|_{w^1}\right) \longrightarrow (D_N)^{\mathbb{Z}_-} \cap \left(\ell_-^{w^2}(\mathbb{R}^N), \|\cdot\|_{w^2}\right)$. The same conclusion holds when the weighted spaces are replaced by $\left(\ell_-^\infty(\mathbb{R}^n), \|\cdot\|_\infty\right)$ and $\left(\ell_-^\infty(\mathbb{R}^N), \|\cdot\|_\infty\right)$. In that case, if H is analytic then so is U_H and the radius of convergence of the series expansion of U_H is bigger or equal than that of H.*

Proof. **(i)** Recall first that H_U can be written as $H_U = p_0 \circ U$. The chain rule and the linearity of the projection p_0 imply that $D^r H_U(\mathbf{z}) = p_0 \circ D^r U(\mathbf{z})$ for any $\mathbf{z} \in V_n$. The first part of Lemma 1 guarantees then that H_U is of class $C^r(V_n)$ and that

$$\left\|\!\left|D^r H_U(\mathbf{z})\right|\!\right\|_{w^1} = \left\|\!\left|p_0 \circ D^r U(\mathbf{z})\right|\!\right\|_{w^1} \leq \|p_0\|_{w^2} \cdot \left\|\!\left|D^r U(\mathbf{z})\right|\!\right\|_{w^1,w^2} = \left\|\!\left|D^r U(\mathbf{z})\right|\!\right\|_{w^1,w^2}, \text{ for any } \mathbf{z} \in V_n,$$

as required. The proof for the spaces $\ell_-^\infty(\mathbb{R}^n)$ and $\ell_-^\infty(\mathbb{R}^N)$ is obtained by taking as sequence w the constant sequence w^t given by $w_t^t := 1$, for all $t \in \mathbb{N}$.

(ii) First of all, notice that the hypothesis on c^1 and the convexity of V_n imply via the mean value theorem Abraham et al. (1988) that H is Lipschitz. Moreover, the hypothesis on $\mathcal{L}_{w,1}$ in the statement implies that condition (6.30) is satisfied and hence the third part in Proposition 37 guarantees that U_H maps into $\ell_-^{w^2}(\mathbb{R}^N)$.

Now, the expression (6.31) implies that for any $\mathbf{z} \in V_n$,

$$D^r U_H(\mathbf{z}) = \prod_{t\in\mathbb{Z}_-} D^r H(T_{-t}(\mathbf{z})) \circ \underbrace{(T_{-t}, \ldots, T_{-t})}_{r \text{ times}}, \quad r \geq 1. \tag{6.36}$$

In order to prove (6.33) consider $\mathbf{u}^1, \ldots, \mathbf{u}^r \in \ell_-^{w^1}(\mathbb{R}^n)$ arbitrary and notice that by the second part of Lemma 1 we have

$$\begin{aligned}
\left\|D^r U_H(\mathbf{z})\left(\mathbf{u}^1, \ldots, \mathbf{u}^r\right)\right\|_{w^2} &= \sup_{t\in\mathbb{Z}_-} \left\{\left\|D^r H(T_{-t}(\mathbf{z})) \cdot \left(T_{-t}(\mathbf{u}^1), \ldots, T_{-t}(\mathbf{u}^r)\right)\right\| w_{-t}^2\right\} \\
&\leq c^r \sup_{t\in\mathbb{Z}_-} \left\{\left\|T_{-t}(\mathbf{u}^1)\right\|_{w^1} \cdots \left\|T_{-t}(\mathbf{u}^r)\right\|_{w^1} w_{-t}^2\right\} \\
&\leq c^r \sup_{t\in\mathbb{Z}_-} \left\{\left\|\mathbf{u}^1\right\|_{w^1} \cdots \left\|\mathbf{u}^r\right\|_{w^1} L_{w^1}^{-rt} w_{-t}^2\right\} \leq c^r \left\|\mathcal{L}_{w^1,r}\right\|_{w^2} \left\|\mathbf{u}^1\right\|_{w^1} \cdots \left\|\mathbf{u}^r\right\|_{w^1},
\end{aligned}$$

as required. We now show that U_H is of class $C^{r-1}(V_n)$. Let $\mathbf{z}^1, \mathbf{z}^2 \in V_n$ arbitrary. Then, using a strategy similar to that one in the last inequality in the previous expression, we have

$$\begin{aligned}
&\left\|\!\left|D^{r-1}U_H(\mathbf{z}^1) - D^{r-1}U_H(\mathbf{z}^2)\right|\!\right\|_{w^1,w^2} \\
&\qquad = \sup_{\substack{\mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\in\ell_-^{w^1}(\mathbb{R}^n) \\ \mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\neq\mathbf{0}}} \left\{\frac{\left\|\left(D^{r-1}U_H(\mathbf{z}^1) - D^{r-1}U_H(\mathbf{z}^2)\right)\cdot\left(\mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\right)\right\|_{w^2}}{\left\|\mathbf{u}^1\right\|_{w^1}\cdots\left\|\mathbf{u}^{r-1}\right\|_{w^1}}\right\} \\
&= \sup_{\substack{\mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\in\ell_-^{w^1}(\mathbb{R}^n) \\ \mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\neq\mathbf{0}}} \left\{\frac{\sup_{t\in\mathbb{Z}_-}\left\{\left\|\left(D^{r-1}H(T_{-t}(\mathbf{z}^1)) - D^{r-1}H(T_{-t}(\mathbf{z}^2))\right)\cdot\left(T_{-t}(\mathbf{u}^1),\ldots,T_{-t}(\mathbf{u}^{r-1})\right)\right\| w_{-t}^2\right\}}{\left\|\mathbf{u}^1\right\|_{w^1}\cdots\left\|\mathbf{u}^{r-1}\right\|_{w^1}}\right\} \\
&\leq \sup_{\substack{\mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\in\ell_-^{w^1}(\mathbb{R}^n) \\ \mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\neq\mathbf{0}}} \left\{\frac{\sup_{t\in\mathbb{Z}_-}\left\{c^r\left\|T_{-t}(\mathbf{z}^1) - T_{-t}(\mathbf{z}^2)\right\|_{w^1}\cdot\left\|T_{-t}(\mathbf{u}^1)\right\|_{w^1}\cdots\left\|T_{-t}(\mathbf{u}^{r-1})\right\|_{w^1}\cdot w_{-t}^2\right\}}{\left\|\mathbf{u}^1\right\|_{w^1}\cdots\left\|\mathbf{u}^{r-1}\right\|_{w^1}}\right\} \\
&\qquad = \sup_{\substack{\mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\in\ell_-^{w^1}(\mathbb{R}^n) \\ \mathbf{u}^1,\ldots,\mathbf{u}^{r-1}\neq\mathbf{0}}} \left\{\frac{\sup_{t\in\mathbb{Z}_-}\left\{c^r\left\|\mathbf{z}^1 - \mathbf{z}^2\right\|_{w^1}\cdot\left\|\mathbf{u}^1\right\|_{w^1}\cdots\left\|\mathbf{u}^{r-1}\right\|_{w^1} L_{w^1}^{-rt} w_{-t}^2\right\}}{\left\|\mathbf{u}^1\right\|_{w^1}\cdots\left\|\mathbf{u}^{r-1}\right\|_{w^1}}\right\} \\
&\qquad\qquad \leq c^r\left\|\mathcal{L}_{w^1,r}\right\|_{w^2}\left\|\mathbf{z}^1 - \mathbf{z}^2\right\|_{w^1},
\end{aligned}$$

which shows that the map $D^{r-1}U_H : (V_n, \|\cdot\|_{w^1}) \longrightarrow \left(L^{r-1}(\ell_-^{w^1}(\mathbb{R}^n), \ell_-^{w^2}(\mathbb{R}^N)), \|\!|\cdot|\!\|_{w^1,w^2}\right)$ is Lipschitz continuous with Lipschitz constant $c^r\left\|\mathcal{L}_{w^1,r}\right\|_{w^2}$.

(iii) First, the condition $c^r < +\infty$ for all $r \in \mathbb{N}^+$ implies by part **(ii)** that U_H is smooth if H is. Suppose now that we work with the supremum norm. The expression (6.36) shows that the point $\mathbf{z} \in V_n$ belongs to the domain of convergence of the series expansion of U_H if and only if all the points $T_{-t}(\mathbf{z})$ belong to the domain of convergence of the series expansion of H. Finally, suppose that $\mathbf{z} \in V_n$ belongs to the domain of convergence of the series expansion of H. Since $\|T_{-t}\|_\infty \leq 1$ for all $t \in \mathbb{Z}_-$ by Lemma 1, we have that $\|T_{-t}(\mathbf{z})\|_\infty \leq \|\mathbf{z}\|_\infty$, which guarantees that all the points $T_{-t}(\mathbf{z})$ belong to the domain of convergence of the series expansion of H and hence, by the argument above, $\mathbf{z} \in V_n$ belongs to the domain of convergence of the series expansion of U_H, which proves the statement. ■

Remark 40 An important consequence of part **(ii)** in this proposition and, in particular, of the condition (6.34) is that, in general, one cannot obtain (higher order) differentiable filters out of differentiable functionals using the same weighted norm in the domain and the target of the filter. The weighted norm in the target needs to be chosen so that it satisfies the nonautomatic condition (6.34) that, additionally, depends on the differentiability degree that we want to preserve. Weighted norms that satisfy that property are relatively easy to find in most cases. For example, if we take as w^1 the geometric sequence in Remark 2, then $\mathcal{L}_{w^1,j} = \left(\lambda^{-jt}\right)_{t\in\mathbb{N}}$ and hence condition (6.34) is satisfied if we take as w^2 any sequence of the type $\left(w^1\right)^r$ (using the notation in Lemma 35) with $r \geq j$.

6.5. Proof of Theorem 7

Consider the map

$$\begin{array}{rccc} \mathcal{F}: & (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n & \longrightarrow & (D_N)^{\mathbb{Z}_-} \\ & (\mathbf{x},\mathbf{z}) & \longmapsto & (\mathcal{F}(\mathbf{x},\mathbf{z}))_t := F(\mathbf{x}_{t-1},\mathbf{z}_t). \end{array} \tag{6.37}$$

We now show that first, under the two sets of hypotheses in the statement, $\mathcal{F}$ actually maps into $(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ and second, that $\mathcal{F}$ is continuous. Suppose first that we are in the hypotheses in **(i)**. Since D_N is compact then $(D_N)^{\mathbb{Z}_-} \subset \ell^w_-(\mathbb{R}^N)$ and hence $\mathcal{F}$ obviously maps into $(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$. Regarding the continuity, notice that $\mathcal{F}$ can be written as

$$\mathcal{F} = \prod_{t\in\mathbb{Z}_-} F_t \quad \text{with} \quad F_t := F \circ p_t \circ (T_1 \times \mathrm{id}_{V_n}) : (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n \longrightarrow D_N. \tag{6.38}$$

The continuity of F, the fact that L_w is by hypothesis finite, and Lemma 1 imply that all the functions $F_t : (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n \subset \ell^w_-(\mathbb{R}^N) \oplus \ell^w_-(\mathbb{R}^n) \longrightarrow D_N \subset \mathbb{R}^N$ are continuous and moreover, they map into a compact subset of $\mathbb{R}^N$. An argument mimicking the proof of the first part of Lemma 36 allows us to conclude that $\mathcal{F} : (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n \longrightarrow (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ is a continuous map.

Suppose now that we are in the hypotheses in part **(ii)**. We now show that since F is Lipschitz then so are all the functions $F_t := F \circ p_t \circ (T_1 \times \mathrm{id}_{V_n})$, $t \in \mathbb{Z}_-$, by Lemma 1, where we consider the direct sum of weighted spaces $\ell^w_-(\mathbb{R}^N) \oplus \ell^w_-(\mathbb{R}^n)$ as a Banach space with the sum norm $\|\cdot\|_{w\oplus w}$ defined by $\|(\mathbf{x},\mathbf{z})\|_{w\oplus w} := \|\mathbf{x}\|_w + \|\mathbf{z}\|_w$, for any $(\mathbf{x},\mathbf{z}) \in \ell^w_-(\mathbb{R}^N) \oplus \ell^w_-(\mathbb{R}^n)$. Indeed, let c_F be the Lipschitz constant of F and let $(\mathbf{x}^1,\mathbf{z}^1), (\mathbf{x}^2,\mathbf{z}^2) \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n$, then:

$$\begin{aligned} \left\|F \circ p_t \circ (T_1 \times \mathrm{id}_{V_n})(\mathbf{x}^1,\mathbf{z}^1) - F \circ p_t \circ (T_1 \times \mathrm{id}_{V_n})(\mathbf{x}^2,\mathbf{z}^2)\right\| &\leq c_F \left\|p_t \circ (T_1 \times \mathrm{id}_{V_n})(\mathbf{x}^1-\mathbf{x}^2,\mathbf{z}^1-\mathbf{z}^2)\right\| \\ &\leq \frac{c_F}{w_{-t}} \left\|(T_1 \times \mathrm{id}_{V_n})(\mathbf{x}^1-\mathbf{x}^2,\mathbf{z}^1-\mathbf{z}^2)\right\|_w \leq \frac{c_F}{w_{-t}}(L_w \left\|\mathbf{x}^1-\mathbf{x}^2\right\|_w + \left\|\mathbf{z}^1-\mathbf{z}^2\right\|_w) \\ &\leq \frac{c_F}{w_{-t}} L_w(\left\|\mathbf{x}^1-\mathbf{x}^2\right\|_w + \left\|\mathbf{z}^1-\mathbf{z}^2\right\|_w) = \frac{c_F}{w_{-t}} L_w \left\|(\mathbf{x}^1,\mathbf{z}^1)-(\mathbf{x}^2,\mathbf{z}^2)\right\|_{w\oplus w}. \end{aligned} \tag{6.39}$$

This chain of inequalities show that F_t is a Lipschitz continuous function and that $c_F L_w / w_{-t}$ is a Lipschitz constant. Given that the sequence $c_{\mathcal{F}} := (c_F L_w / w_{-t})_{t\in\mathbb{Z}_-}$ is such that $\|c_{\mathcal{F}}\|_w = c_F L_w < +\infty$, the part **(ii)** of Lemma 36 in the Appendices guarantees that $\mathcal{F}$ is Lipschitz continuous and that $c_F L_w$ is a Lipschitz constant, that is,

$$\left\|\mathcal{F}(\mathbf{x}^1,\mathbf{z}^1) - \mathcal{F}(\mathbf{x}^2,\mathbf{z}^2)\right\|_w \leq \; c_F L_w \left\|(\mathbf{x}^1,\mathbf{z}^1)-(\mathbf{x}^2,\mathbf{z}^2)\right\|_{w\oplus w}. \tag{6.40}$$

Moreover, let $\mathbf{u}^0 := (\mathbf{x}^0,\mathbf{z}^0) \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N) \times V_n$. The fact that $\mathbf{u}^0$ is a solution of the reservoir system implies that $\mathcal{F}(\mathbf{u}^0) = \mathbf{x}^0 \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$. An argument mimicking (6.20) in the proof of part **(ii)** in Lemma 36 proves that in those conditions $\mathcal{F}$ maps into $(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$.

We now show that in the presence of hypothesis (3.3) $\mathcal{F}$ is a contraction on the first entry with constant $cL_w < 1$. Indeed, for any $\mathbf{x}^1,\mathbf{x}^2 \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ and any $\mathbf{z} \in V_n$, we have

$$\left\|\mathcal{F}(\mathbf{x}^1,\mathbf{z}) - \mathcal{F}(\mathbf{x}^2,\mathbf{z})\right\|_w = \sup_{t\in\mathbb{Z}_-} \left\{\left\|F(\mathbf{x}^1_{t-1},\mathbf{z}_t) - F(\mathbf{x}^2_{t-1},\mathbf{z}_t)\right\| w_{-t}\right\} \leq \sup_{t\in\mathbb{Z}_-} \left\{\left\|\mathbf{x}^1_{t-1}-\mathbf{x}^2_{t-1}\right\| c w_{-t}\right\}, \tag{6.41}$$

where we used that F is a contraction on the first entry. Now,

$$\sup_{t\in\mathbb{Z}_-} \left\{\left\|\mathbf{x}^1_{t-1}-\mathbf{x}^2_{t-1}\right\| c w_{-t}\right\} = c \sup_{t\in\mathbb{Z}_-} \left\{\left\|\mathbf{x}^1_{t-1}-\mathbf{x}^2_{t-1}\right\| w_{-(t-1)} \frac{w_{-t}}{w_{-(t-1)}}\right\} \leq cL_w \left\|\mathbf{x}^1-\mathbf{x}^2\right\|_w. \tag{6.42}$$

This shows that $\mathcal{F}$ is a family of contractions with constant $cL_w < 1$ that is continuously parametrized by the elements in V_n. Since by hypothesis, the domain $(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ is complete, Theorem 6.4.1 in Sternberg (2010) implies the existence of a continuous map $U^F : (V_n, \|\cdot\|_w) \longrightarrow \left((D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N), \|\cdot\|_w\right)$ that is uniquely determined by the identity

$$\mathcal{F}\left(U^F(\mathbf{z}),\mathbf{z}\right) = U^F(\mathbf{z}), \quad \text{for all } \mathbf{z} \in V_n. \tag{6.43}$$

The causality and the time-invariance of U^F are a consequence of the time invariance of V_n and of Proposition 2.1 in Grigoryeva and Ortega (2018a).

We now assume that F is Lipschitz on the second component and prove (3.4). The relation (6.43) that defines U^F is equivalent to

$$U^F(\mathbf{z})_t = F(U^F(\mathbf{z})_{t-1},\mathbf{z}_t), \quad \text{for all} \quad \mathbf{z} \in V_n, \; t \in \mathbb{Z}_-.$$

Consequently, for any $\mathbf{z}^1,\mathbf{z}^2 \in V_n$, we have,

$$\begin{aligned} \left\|U^F(\mathbf{z}^1) - U^F(\mathbf{z}^2)\right\|_w &= \sup_{t\in\mathbb{Z}_-} \left\{\left\|U^F(\mathbf{z}^1)_t - U^F(\mathbf{z}^2)_t\right\| w_{-t}\right\} \\ &= \sup_{t\in\mathbb{Z}_-} \left\{\left\|F(U^F(\mathbf{z}^1)_{t-1},\mathbf{z}^1_t) - F(U^F(\mathbf{z}^2)_{t-1},\mathbf{z}^2_t)\right\| w_{-t}\right\} \\ \leq \sup_{t\in\mathbb{Z}_-} &\left\{\left(\left\|F(U^F(\mathbf{z}^1)_{t-1},\mathbf{z}^1_t) - F(U^F(\mathbf{z}^1)_{t-1},\mathbf{z}^2_t)\right\| + \left\|F(U^F(\mathbf{z}^1)_{t-1},\mathbf{z}^2_t) - F(U^F(\mathbf{z}^2)_{t-1},\mathbf{z}^2_t)\right\|\right) w_{-t}\right\} \\ &\leq \sup_{t\in\mathbb{Z}_-} \left\{L_z \left\|\mathbf{z}^1_t - \mathbf{z}^2_t\right\| w_{-t} + c \left\|U^F(\mathbf{z}^1)_{t-1} - U^F(\mathbf{z}^2)_{t-1}\right\| w_{-t}\right\}. \end{aligned}$$

If we repeat this procedure i times, it is easy to see that

$$\begin{aligned} &\left\|U^F(\mathbf{z}^1) - U^F(\mathbf{z}^2)\right\|_w \\ &\leq L_z \sup_{t\in\mathbb{Z}_-} \left\{\sum_{j=0}^{i} c^j \left\|\mathbf{z}^1_{t-j} - \mathbf{z}^2_{t-j}\right\| w_{-t}\right\} + c^{i+1} \sup_{t\in\mathbb{Z}_-} \left\{\left\|U^F(\mathbf{z}^1)_{t-(i+1)} - U^F(\mathbf{z}^2)_{t-(i+1)}\right\| w_{-t}\right\}. \end{aligned} \tag{6.44}$$

We now study separately the two summands in the right hand side of the previous inequality. First, by Lemma 1,

$$\begin{aligned}
L_z \sup_{t\in\mathbb{Z}_-}\left\{\sum_{j=0}^{i} c^j \left\|\mathbf{z}^1_{t-j}-\mathbf{z}^2_{t-j}\right\| w_{-t}\right\} &= L_z \sup_{t\in\mathbb{Z}_-}\left\{\sum_{j=0}^{i} c^j \left\|\left(T_j(\mathbf{z}^1)\right)_t-\left(T_j(\mathbf{z}^2)\right)_t\right\| w_{-t}\right\}\\
&= L_z \sup_{t\in\mathbb{Z}_-}\left\{\sum_{j=0}^{i} c^j \left\|T_j(\mathbf{z}^1-\mathbf{z}^2)_t\right\| w_{-t}\right\} \leq L_z \sum_{j=0}^{i} c^j \sup_{t\in\mathbb{Z}_-}\left\{\left\|T_j(\mathbf{z}^1-\mathbf{z}^2)_t\right\| w_{-t}\right\}\\
&= L_z \sum_{j=0}^{i} c^j \left\|T_j(\mathbf{z}^1-\mathbf{z}^2)\right\|_w \leq L_z \left\|\mathbf{z}^1-\mathbf{z}^2\right\|_w \sum_{j=0}^{i} c^j |||T_j|||_w\\
&\leq L_z \left\|\mathbf{z}^1-\mathbf{z}^2\right\|_w \sum_{j=0}^{i} (cL_w)^j = L_z \left\|\mathbf{z}^1-\mathbf{z}^2\right\|_w \frac{1-(cL_w)^{i+1}}{1-cL_w}, \qquad (6.45)
\end{aligned}$$

while the second summand can be bounded as follows

$$\begin{aligned}
c^{i+1}\sup_{t\in\mathbb{Z}_-}\left\{\left\|U^F(\mathbf{z}^1)_{t-(i+1)}-U^F(\mathbf{z}^2)_{t-(i+1)}\right\| w_{-t}\right\} &= c^{i+1}\sup_{t\in\mathbb{Z}_-}\left\{\left\|T_{i+1}\left(U^F(\mathbf{z}^1)\right)_t - T_{i+1}\left(U^F(\mathbf{z}^2)\right)_t\right\| w_{-t}\right\}\\
&= c^{i+1}\left\|T_{i+1}(U^F(\mathbf{z}^1)-(U^F(\mathbf{z}^2))\right\|_w \leq c^{i+1}|||T_{i+1}|||_w \left\|U^F(\mathbf{z}^1)-U^F(\mathbf{z}^2)\right\|_w\\
&\leq (cL_w)^{i+1}\left\|U^F(\mathbf{z}^1)-U^F(\mathbf{z}^2)\right\|_w. \qquad (6.46)
\end{aligned}$$

If we now chain the inequalities (6.45) and (6.46) with (6.44) we can conclude that

$$(1-(cL_w)^{i+1})\left\|U^F(\mathbf{z}^1)-U^F(\mathbf{z}^2)\right\|_w \leq L_z\left\|\mathbf{z}^1-\mathbf{z}^2\right\|_w \frac{1-(cL_w)^{i+1}}{1-cL_w}, \qquad (6.47)$$

which after simplification using the condition (3.3) results in (3.4). ■

Remark 41 A slight modification of the proof of Theorem 7 **(ii)** can be used to extend this statement to reservoir systems with inputs and outputs in $\ell^{p,w}_-(\mathbb{R}^n)$ and $\ell^{p,w}_-(\mathbb{R}^N)$, respectively. Indeed, assume that we are under the hypotheses of Theorem 7 **(ii)** with those spaces instead of $\ell^w_-(\mathbb{R}^n)$ and $\ell^w_-(\mathbb{R}^N)$. Suppose, additionally, that

$$cL_w^{1/p} < 1. \qquad (6.48)$$

Then, there exists a unique causal and time-invariant continuous reservoir filter $U^F : (V_n, \|\cdot\|_{p,w}) \longrightarrow ((D_N)^{\mathbb{Z}_-} \cap \ell^{p,w}_-(\mathbb{R}^N), \|\cdot\|_{p,w})$. Additionally, U^F is also Lipschitz with constant

$$L_{U^F} := \frac{L_z}{1-cL_w^{1/p}}.$$

The proof of this fact is carried out by showing that the map $\mathcal{F}$ in (6.38) is Lipschitz continuous when $\ell^{p,w}_-(\mathbb{R}^n)$ and $\ell^{p,w}_-(\mathbb{R}^N)$ spaces are considered in its domain and target, respectively, with Lipschitz constant $c_F L_w^{1/p}$ and hence (6.40) holds in that situation. Indeed, for any $(\mathbf{x}^1,\mathbf{z}^1),(\mathbf{x}^2,\mathbf{z}^2) \in (D_N)^{\mathbb{Z}_-} \cap$

$\ell^{p,w}_-(\mathbb{R}^N) \times V_n$ we can show using the statements in Remark 4 that

$$\begin{aligned}
\left\|\mathcal{F}(\mathbf{x}^1,\mathbf{z}^1)-\mathcal{F}(\mathbf{x}^2,\mathbf{z}^2)\right\|^p_{p,w} &= \sum_{t\in\mathbb{Z}_-}\left\|F_t(\mathbf{x}^1_{t-1},\mathbf{z}^1_t)-F_t(\mathbf{x}^2_{t-1},\mathbf{z}^2_1)\right\|^p w_{-t}\\
&= \sum_{t\in\mathbb{Z}_-}\left\|F(\mathbf{x}^1_{t-1},\mathbf{z}^1_t)-F_t(\mathbf{x}^2_{t-1},\mathbf{z}^2_t)\right\|^p w_{-t} \leq c_F^p \sum_{t\in\mathbb{Z}_-}\left\|\mathbf{x}^1_{t-1}-\mathbf{x}^2_{t-1}\right\|^p w_{-t} + c_F^p \sum_{t\in\mathbb{Z}_-}\left\|\mathbf{z}^1_t-\mathbf{z}^2_t\right\|^p w_{-t}\\
&\leq c_F^p\left\|T_1(\mathbf{x}^1-\mathbf{x}^2)\right\|^p_{p,w} + c_F^p\left\|\mathbf{z}^1-\mathbf{z}^2\right\|^p_{p,w} \leq c_F^p (L_w^{1/p})^p\left\|\mathbf{x}^1-\mathbf{x}^2\right\|^p_{p,w} + c_F^p\left\|\mathbf{z}^1-\mathbf{z}^2\right\|^p_{p,w}\\
&\leq c_F^p (L_w^{1/p})^p\left\|(\mathbf{x}^1,\mathbf{z}^1)-(\mathbf{x}^2,\mathbf{z}^2)\right\|^p_{p,w\oplus w},
\end{aligned}$$

where in the last inequality we used that $L_w^{1/p} > 1$. We now show that $\mathcal{F}$ is a contraction on the first entry whenever condition (6.48) is satisfied. Indeed,

$$\begin{aligned}
\left\|\mathcal{F}(\mathbf{x}^1,\mathbf{z})-\mathcal{F}(\mathbf{x}^2,\mathbf{z})\right\|^p_{p,w} &= \sum_{t\in\mathbb{Z}_-}\left\|F(\mathbf{x}^1_{t-1},\mathbf{z}_t)-F_t(\mathbf{x}^2_{t-1},\mathbf{z}_t)\right\|^p w_{-t}\\
&\leq c^p \sum_{t\in\mathbb{Z}_-}\left\|\mathbf{x}^1_{t-1}-\mathbf{x}^2_{t-1}\right\|^p w_{-t} = c^p\left\|T_1(\mathbf{x}^1-\mathbf{x}^2)\right\|^p_{p,w} \leq c^p (L_w^{1/p})^p\left\|\mathbf{x}^1-\mathbf{x}^2\right\|^p_{p,w}.
\end{aligned}$$

The rest of the proof can be obtained by mimicking the developments after (6.42).

6.6. Proof of Theorem 12

Consider the map $\mathcal{F} : (D_N)^{\mathbb{Z}_-} \times (D_n)^{\mathbb{Z}_-} \longrightarrow (D_N)^{\mathbb{Z}_-}$ defined in (6.37) and endow $(D_n)^{\mathbb{Z}_-}$ and $(D_N)^{\mathbb{Z}_-}$ with the relative topologies induced by the product topologies in $(\mathbb{R}^n)^{\mathbb{Z}_-}$ and $(\mathbb{R}^N)^{\mathbb{Z}_-}$, respectively. It is easy to see that the maps p_t and T_1 are continuous with respect to those product topologies and hence $\mathcal{F}$ can be written using (6.38) as a Cartesian product of continuous functions, which is always continuous in the product topology.

Consider now any weighting sequence w such that $cL_w < 1$. Using an argument similar to the proof of Lemma 36 **(i)**, we can conclude that $(D_N)^{\mathbb{Z}_-} \subset \ell^w_-(\mathbb{R}^N)$ and that the product topology on $(D_N)^{\mathbb{Z}_-}$ coincides with the norm topology induced by $\|\cdot\|_w$. Now, following the expressions (6.41) and (6.42) it can be shown that $\mathcal{F}$ is a contraction on the first entry and with respect to $\|\cdot\|_w$. In view of these facts and given that the product topology in $(D_n)^{\mathbb{Z}_-} \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$ is metrizable (see (Munkres, 2014, Theorem 20.5)) and that $(D_N)^{\mathbb{Z}_-} \subset (\mathbb{R}^N)^{\mathbb{Z}_-}$ is compact by Tychonoff's Theorem (see (Munkres, 2014, Theorem 37.3)) in the product topology and hence complete, Theorem 6.4.1 in Sternberg (2010) implies the existence of a unique fixed point of $\mathcal{F}$ for each $\mathbf{z} \in (D_n)^{\mathbb{Z}_-}$, which establishes the ESP. Moreover, that result also shows the continuity of the associated filter $U^F : (D_n)^{\mathbb{Z}_-} \longrightarrow ((D_N)^{\mathbb{Z}_-}, \|\cdot\|_w)$.

Finally, if $(D_n)^{\mathbb{Z}_-} \subset \ell^w_-(\mathbb{R}^n)$, we know from (Grigoryeva and Ortega, 2018b, Proposition 2.9) that the inclusion $\ell^w_-(\mathbb{R}^n) \hookrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}$ is continuous and hence so is U^F when in $(D_n)^{\mathbb{Z}_-}$ we consider the topology generated by the norm $\|\cdot\|_w$, which establishes the FMP in that situation. ■

6.7. Proof of Corollary 13

Under the hypothesis in part **(i)**, the continuity of h implies that $h(D_N)$ is compact and hence there exists a constant $R > 0$ such that $h(D_N) \subset \overline{B_{\|\cdot\|}(\mathbf{0}, R)}$. The first part of Lemma 36 guarantees that the map $\mathcal{H} := \prod_{t\in\mathbb{Z}_-} h : ((D_N)^{\mathbb{Z}_-}, \|\cdot\|_w) \longrightarrow (K_R, \|\cdot\|_w)$ is continuous and as $U^F_h = \mathcal{H} \circ U^F$ and we proved that under the hypotheses **(i)** in the theorem that $U^F : (V_n, \|\cdot\|_w) \longrightarrow (K_L, \|\cdot\|_w)$ is continuous, the claim follows.

We now prove the statement under the hypotheses in part **(ii)**. First, we show that if h is Lipschitz continuous in D^N with constant c_h then so is the map $\mathcal{H}$ in $(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$. Indeed, let $\mathbf{x}^1, \mathbf{x}^2 \in$

$(D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$, then

$$\left\|\mathcal{H}(\mathbf{x}^1) - \mathcal{H}(\mathbf{x}^2)\right\|_w = \sup_{t\in\mathbb{Z}_-} \left\{\left\|h(\mathbf{x}^1_t) - h(\mathbf{x}^2_t)\right\| w_{-t}\right\} \leq c_h \left\|\mathbf{x}^1 - \mathbf{x}^2\right\|_w .$$

The hypothesis $U^F_h(\mathbf{z}^0) \in \ell^w_-(\mathbb{R}^d)$ amounts to the fact that the point $U^F(\mathbf{z}^0) \in (D_N)^{\mathbb{Z}_-} \cap \ell^w_-(\mathbb{R}^N)$ is such that $\mathcal{H}(U^F(\mathbf{z}^0)) \in \ell^w_-(\mathbb{R}^d)$. An argument mimicking (6.20) in the proof of part **(ii)** in Lemma 36 proves that in those conditions $\mathcal{H}$ maps into $\ell^w_-(\mathbb{R}^d)$. ■

6.8. Proof of the statements in Section 3.1

Proof of statement (i) for linear reservoir maps. One can show by mimicking the proof of (Grigoryeva and Ortega, 2018a, Corollary 11) that whenever condition (3.6) is satisfied for a given weighting sequence w, the reservoir system determined by (3.5) has a unique reservoir filter $U^F : \ell^w_-(\mathbb{R}^n) \longrightarrow \ell^w_-(\mathbb{R}^N)$ associated that is determined by the linear functional $H^F : \ell^w_-(\mathbb{R}^n) \longrightarrow \mathbb{R}^N$ given by

$$H^F(\mathbf{z}) := \sum_{j=0}^{\infty} A^j \mathbf{c}\mathbf{z}_{-j}.$$

This linear functional is bounded because for any $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$, the hypothesis (3.6) implies that:

$$\left\|H^F(\mathbf{z})\right\| \leq \sum_{j=0}^{\infty} \left|\left|\left|A^j\right|\right|\right| \|\mathbf{c}\| \, \|\mathbf{z}_{-j}\| = \|\mathbf{c}\| \sum_{j=0}^{\infty} \left|\left|\left|A^j\right|\right|\right| \|\mathbf{z}_{-j}\| \frac{w_j}{w_j} \leq \|\mathbf{c}\| \, \|\mathbf{z}\|_w \sum_{j=0}^{\infty} \frac{\left|\left|\left|A^j\right|\right|\right|}{w_j} < +\infty$$

We now show that for any weighting sequence w that satisfies $|||A|||L_w < 1$, the condition (3.6) always holds. Indeed, using (2.10) we obtain

$$\sum_{j=0}^{\infty} \frac{\left|\left|\left|A^j\right|\right|\right|}{w_j} \leq \sum_{j=0}^{\infty} \frac{|||A|||^j}{w_j} \leq \sum_{j=0}^{\infty} |||A|||^j L_w^j = \frac{1}{1 - |||A|||L_w} < +\infty, \quad \text{as required.}$$

We finally show that there exist sequences w that satisfy (3.6) but not $|||A|||L_w < 1$, which is one more example of the fact, that we already indicated in Remark 11, that the FMP condition (3.3) is sufficient but not necessary. Let w be a harmonic weighting sequence as in Remark 2 given by $w_j := 1/(1 + jd)$, $j \in \mathbb{N}$, with $d > 0$. In this case $L_w = 1 + d$ so we can choose a value d such that $|||A|||(1 + d) > 1$. However, at the same time, the condition (3.6) holds in this case because

$$\begin{aligned}\sum_{j=0}^{\infty} \frac{\left|\left|\left|A^j\right|\right|\right|}{w_j} &\leq \sum_{j=0}^{\infty} |||A|||^j (1 + jd) = \sum_{j=0}^{\infty} |||A|||^j + d(j+1)|||A|||^j - d|||A|||^j \\ &= \sum_{j=0}^{\infty} (1-d)|||A|||^j + d(j+1)|||A|||^j = \frac{1-d}{1 - |||A|||} + \frac{d}{(1 - |||A|||)^2} = \frac{1 + |||A|||(d-1)}{(1 - |||A|||)^2} < +\infty.\end{aligned}$$

Another example in this direction can be obtained by using a nilpotent matrices. If A is nilpotent then (3.6) is always satisfied for any weighting sequence w. At the same time, there are nilpotent matrices with arbitrarily large norm $|||A|||$ which, once more, shows that (3.6) can hold, and hence the FMP, without (3.7) being necessarily true. We notice too that reservoir systems determined by nilpotent matrices always satisfy the echo state property even though they are not necessarily contractions.

Proof of statement (ii) for linear reservoir maps. We first prove the statement (3.8). For any $\mathbf{x} \in \overline{B_{\|\cdot\|}(\mathbf{0}, L)}$ and $\mathbf{z} \in \overline{B_{\|\cdot\|}(\mathbf{0}, M)}$,

$$\|F(\mathbf{x}, \mathbf{z})\| = \|A\mathbf{x} + \mathbf{c}\mathbf{z}\| \leq |||A|||L + \|\mathbf{c}\|M = L, \quad \text{as required.}$$

This implies that the reservoir map F in (3.5) restricts to a map $F_{L,M} : \overline{B_{\|\cdot\|}(\mathbf{0}, L)} \times \overline{B_{\|\cdot\|}(\mathbf{0}, M)} \longrightarrow \overline{B_{\|\cdot\|}(\mathbf{0}, L)}$ that is a contraction on the first entry with constant $|||A||| < 1$ and hence satisfies the hypotheses of Corollary 10. This guarantees the existence of a unique associated causal and time-invariant filter $U^F : K_M \longrightarrow K_L$ that has the fading memory property with respect to any weighting sequence w.

Proof of the statements for state-affine systems. We prove only the statement **(i)** since statement **(ii)** can be easily obtained by mimicking the similar statement for the linear case. Indeed, a straightforward generalization of (Grigoryeva and Ortega, 2018a, Proposition 14) shows that whenever $M_p < 1$ and $M_q < +\infty$, the reservoir system determined by (3.11) has a unique reservoir filter $U^F : \ell^w_-(\mathbb{R}^n) \cap (D_n)^{\mathbb{Z}_-} \longrightarrow \ell^w_-(\mathbb{R}^N)$ associated that is determined by the linear functional $H^F : \ell^w_-(\mathbb{R}^n) \cap (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^N$ given by

$$H^F(\mathbf{z}) := \sum_{j=0}^{\infty} \left(\prod_{k=0}^{j-1} p(\mathbf{z}_{-k}) \right) q(\mathbf{z}_{-j}).$$

Mimicking the proof of (Grigoryeva and Ortega, 2018a, Proposition 16) it can be shown that there exists a constant $C_{p,q} > 0$ that depends exclusively of p and q such that for any $\mathbf{z}, \mathbf{s} \in \ell^w_-(\mathbb{R}^n)$

$$\left\|H^F(\mathbf{z}) - H^F(\mathbf{s})\right\| \leq C_{p,q} \sum_{j=0}^{\infty} M_p^j \|\mathbf{z}_{-j} - \mathbf{s}_{-j}\| = C_{p,q} \sum_{j=0}^{\infty} M_p^j \|\mathbf{z}_{-j} - \mathbf{s}_{-j}\| \frac{w_j}{w_j} \leq C_{p,q} \|\mathbf{z} - \mathbf{s}\|_w \sum_{j=0}^{\infty} \frac{M_p^j}{w_j},$$

which shows that $H^F : \ell^w_-(\mathbb{R}^n) \cap (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^N$ is Lipschitz continuous whenever the condition (3.12) holds. The last claim regarding the relation between (3.12) and the FMP condition (3.3) is proved by mimicking the similar statement for the linear case.

6.9. Proof of Theorem 14

We start with a preliminary result whose proof mimics that of Lemma 36 and is also a consequence of Lemma 1. As we already did in the proof of Theorem 7, in the statement we consider the direct sum of weighted spaces $\ell^w_-(\mathbb{R}^N) \oplus \ell^w_-(\mathbb{R}^n)$ as a Banach space with the sum norm $\|\cdot\|_{w\oplus w}$ defined by $\|(\mathbf{u}, \mathbf{v})\|_{w\oplus w} := \|\mathbf{u}\|_w + \|\mathbf{v}\|_w$, for any $(\mathbf{u}, \mathbf{v}) \in \ell^w_-(\mathbb{R}^N) \oplus \ell^w_-(\mathbb{R}^n)$. Additionally, in all that follows V_n stands for any open convex subset of the Banach space $\left(\ell^w_-(\mathbb{R}^n), \|\cdot\|_w\right)$.

Lemma 42 *In the hypotheses of the theorem, consider the map*

$$\begin{array}{rccc} \mathcal{F}: & \ell^w_-(\mathbb{R}^N) \times V_n & \longrightarrow & (\mathbb{R}^N)^{\mathbb{Z}_-} \\ & (\mathbf{x}, \mathbf{z}) & \longmapsto & (\mathcal{F}(\mathbf{x}, \mathbf{z}))_t := F(\mathbf{x}_{t-1}, \mathbf{z}_t), \end{array} \tag{6.49}$$

where V_n is an open convex subset of $\ell^w_-(\mathbb{R}^n)$. Then,

(i) *$\mathcal{F}$ is Lipschitz continuous with constant $L_F L_w$ and maps into $\ell^w_-(\mathbb{R}^N)$.*

(ii) *If F is of class $C^r(\mathbb{R}^N \times \mathbb{R}^n)$, $r \geq 1$, suppose that*

$$L_{F,r} := \sup_{(\mathbf{x},\mathbf{z})\in\mathbb{R}^N\times\mathbb{R}^n} \{|||D^r F(\mathbf{x}, \mathbf{z})|||\} < +\infty. \tag{6.50}$$

and let w' be any weighting sequence such that

$$c_{w',w^r} = \sup_{t\in\mathbb{Z}_-} \left\{ \frac{w'_{-t}}{w^r_{-t}} \right\} < +\infty. \tag{6.51}$$

Then the map $\mathcal{F}$ is a functor between the sets

$$\mathcal{F}:\ell_-^w(\mathbb{R}^N)\times V_n\subset \ell_-^w(\mathbb{R}^N)\oplus \ell_-^w(\mathbb{R}^n)\longrightarrow \ell_-^{w'}(\mathbb{R}^N)$$

and is differentiable of order r and of class $C^{r-1}(\ell_-^w(\mathbb{R}^N)\times V_n)$. Moreover,

$$\left\|\!\left|D^r\mathcal{F}(\mathbf{x},\mathbf{z})\right|\!\right\|_{w,w'}\leq L_{F,r}L_w^r c_{w',w^r},\ \textit{for all } (\mathbf{x},\mathbf{z})\in \ell_-^w(\mathbb{R}^N)\times V_n \tag{6.52}$$

and the map $D^{r-1}\mathcal{F}:\ell_-^w(\mathbb{R}^N)\times V_n\longrightarrow L^{r-1}(\ell_-^w(\mathbb{R}^n)\oplus \ell_-^w(\mathbb{R}^N),\ell_-^{w'}(\mathbb{R}^N))$ is Lipschitz continuous with Lipschitz constant $L_{F,r}L_w^r c_{w',w^r}$.

(iii) *The linear map $D_x\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0):(\ell_-^w(\mathbb{R}^N),\|\cdot\|_w)\longrightarrow(\ell_-^w(\mathbb{R}^N),\|\cdot\|_w)$ is a contraction with constant $L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w<1$.*

These results also hold when the spaces $(\ell_-^w(\mathbb{R}^n),\|\cdot\|_w)$ and $(\ell_-^w(\mathbb{R}^N),\|\cdot\|_w)$ are replaced by $(\ell_-^\infty(\mathbb{R}^n),\|\cdot\|_\infty)$ and $(\ell_-^\infty(\mathbb{R}^N),\|\cdot\|_\infty)$, respectively. In that case, the statement is obtained by taking as the sequences w and w' the constant sequence w^ι given by $w_t^\iota:=1$, for all $t\in\mathbb{N}$. The inequality (6.52) holds true with $L_w=c_{w',w^r}=1$.

Proof of the lemma. **(i)** Notice first that, as we pointed out in (6.38), and using the notation in Lemma 36,

$$\mathcal{F}=\prod_{t\in\mathbb{Z}_-}F_t,\quad \text{where}\quad F_t:=F\circ p_t\circ(T_1\times \mathrm{id}_{V_n}):\ell_-^w(\mathbb{R}^N)\times V_n\longrightarrow\mathbb{R}^N. \tag{6.53}$$

Also, the hypothesis (4.1), the mean value theorem, and the convexity of the set

$$p_t\circ(T_1\times\mathrm{id}_{V_n})\left(\ell_-^w(\mathbb{R}^N)\times V_n\right)$$

imply that F is a Lipschitz function with constant L_F. A development identical to (6.39) guarantees that the maps F_t are Lipschitz and that L_FL_w/w_{-t} is a Lipschitz constant of F_t, $t\in\mathbb{Z}_-$. Given that the sequence $c_{\mathcal{F}}:=(L_FL_w/w_{-t})_{t\in\mathbb{Z}_-}$ is such that $\|c_{\mathcal{F}}\|_w=L_FL_w<+\infty$ and $\mathcal{F}=\prod_{t\in\mathbb{Z}_-}F_t$, the part **(ii)** of Lemma 36 guarantees that $\mathcal{F}$ is Lipschitz continuous and that L_FL_w is a Lipschitz constant of $\mathcal{F}$.

Since by hypothesis the reservoir system has a solution $(\mathbf{x}^0,\mathbf{z}^0)\in\ell_-^w(\mathbb{R}^N)\times V_n$, we have that $\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0)=\mathbf{x}^0\in\ell_-^w(\mathbb{R}^N)$. This implies that $\mathcal{F}$ maps into $\ell_-^w(\mathbb{R}^N)$ since the Lipschitz condition that we just proved shows that for any $(\mathbf{x},\mathbf{z})\in\ell_-^w(\mathbb{R}^N)\times V_n$

$$\|\mathcal{F}(\mathbf{x},\mathbf{z})\|_w\leq L_FL_w\left\|(\mathbf{x},\mathbf{z})-(\mathbf{x}^0,\mathbf{z}^0)\right\|_{w\oplus w}+\|\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0)\|_w,$$

which shows that $\|\mathcal{F}(\mathbf{x},\mathbf{z})\|_w<+\infty$ and hence that $\mathcal{F}(\mathbf{x},\mathbf{z})\in\ell_-^w(\mathbb{R}^N)$.

(ii) The expression (6.53), the chain rule, the finiteness of L_w, and the linearity of p_t and T_1 imply that for any $(\mathbf{x},\mathbf{z})\in\ell_-^w(\mathbb{R}^N)\times V_n$:

$$D^rF_t(\mathbf{x},\mathbf{z})=D^rF(\mathbf{x}_{t-1},\mathbf{z}_t)\circ(p_t\circ(T_1\times\mathrm{id}_{V_n}),\ldots,p_t\circ(T_1\times\mathrm{id}_{V_n})):\left(\ell_-^w(\mathbb{R}^N)\oplus\ell_-^w(\mathbb{R}^n)\right)^r\longrightarrow\mathbb{R}^N. \tag{6.54}$$

We now prove (6.52). Notice first that for $\mathbf{u}=\left(\mathbf{u}^1,\ldots,\mathbf{u}^r\right)=\left((\mathbf{u}_\mathbf{x}^1,\mathbf{u}_\mathbf{z}^1),\ldots,(\mathbf{u}_\mathbf{x}^r,\mathbf{u}_\mathbf{z}^r)\right)\in\left(\ell_-^w(\mathbb{R}^N)\oplus\ell_-^w(\mathbb{R}^n)\right)^r$ we can write using (4.1) and Lemma 1:

$$\begin{aligned}
\|D^rF_t(\mathbf{x},\mathbf{z})\cdot\mathbf{u}\| &= \left\|D^rF(\mathbf{x}_{t-1},\mathbf{z}_t)\circ\left(p_t\circ(T_1\times\mathrm{id}_{V_n})(\mathbf{u}^1),\ldots,p_t\circ(T_1\times\mathrm{id}_{V_n})(\mathbf{u}^r)\right)\right\|\\
&\leq \left\|\!\left|D^rF(\mathbf{x}_{t-1},\mathbf{z}_t)\right|\!\right\|\left\|p_t\circ(T_1\times\mathrm{id}_{V_n})(\mathbf{u}^1)\right\|\cdots\|p_t\circ(T_1\times\mathrm{id}_{V_n})(\mathbf{u}^r)\|\\
&\leq \frac{L_{F,r}}{w_{-t}^r}\left\|\left(T_1(\mathbf{u}_\mathbf{x}^1),\mathbf{u}_\mathbf{z}^1\right)\right\|_{w\oplus w}\cdots\|(T_1(\mathbf{u}_\mathbf{x}^r),\mathbf{u}_\mathbf{z}^r)\|_{w\oplus w}\\
&\leq \frac{L_{F,r}}{w_{-t}^r}\left(\left\|T_1(\mathbf{u}_\mathbf{x}^1)\right\|_w+\left\|\mathbf{u}_\mathbf{z}^1\right\|_w\right)\cdots\left(\|T_1(\mathbf{u}_\mathbf{x}^r)\|_w+\|\mathbf{u}_\mathbf{z}^r\|_w\right)\\
&\leq \frac{L_{F,r}}{w_{-t}^r}\left(L_w\left\|\mathbf{u}_\mathbf{x}^1\right\|_w+\left\|\mathbf{u}_\mathbf{z}^1\right\|_w\right)\cdots\left(L_w\|\mathbf{u}_\mathbf{x}^r\|_w+\|\mathbf{u}_\mathbf{z}^r\|_w\right)\\
&\leq \frac{L_{F,r}L_w^r}{w_{-t}^r}\left(\left\|\mathbf{u}_\mathbf{x}^1\right\|_w+\left\|\mathbf{u}_\mathbf{z}^1\right\|_w\right)\cdots\left(\|\mathbf{u}_\mathbf{x}^r\|_w+\|\mathbf{u}_\mathbf{z}^r\|_w\right)=\frac{L_{F,r}L_w^r}{w_{-t}^r}\left\|\mathbf{u}^1\right\|_{w\oplus w}\cdots\|\mathbf{u}^r\|_{w\oplus w},
\end{aligned}$$

which shows that

$$\left\|\!\left|D^rF_t(\mathbf{x},\mathbf{z})\right|\!\right\|_w\leq\frac{L_{F,r}L_w^r}{w_{-t}^r}. \tag{6.55}$$

Since, as we saw in part **(i)** $\mathcal{F}$ maps into $\ell_-^w(\mathbb{R}^N)$, and by Lemma 35 $\ell_-^w(\mathbb{R}^N)\subset\ell_-^{w^r}(\mathbb{R}^N)$, then $\mathcal{F}$ also maps into $\ell_-^{w^r}(\mathbb{R}^N)$. Additionally, since the sequence $c^r:=\left(L_{F,r}L_w^r/w_{-t}^r\right)_{t\in\mathbb{Z}_-}$ is such that $\|c^r\|_{w^r}=L_{F,r}L_w^r<+\infty$, the part **(iii)** of Lemma 36 guarantees that the map

$$\mathcal{F}:\ell_-^w(\mathbb{R}^N)\times V_n\longrightarrow\ell_-^{w^r}(\mathbb{R}^N)$$

is differentiable of order r and that

$$\left\|\!\left|D^r\mathcal{F}(\mathbf{x},\mathbf{z})\right|\!\right\|_{w,w^r}\leq L_{F,r}L_w^r<+\infty. \tag{6.56}$$

This argument can be reproduced with the power sequence w^r replaced by any other sequence w' that satisfies (6.51), in which case, it is easy to see that $\ell_-^{w^r}(\mathbb{R}^N)\subset\ell_-^{w'}(\mathbb{R}^N)$, and we can conclude the differentiability of the map $\mathcal{F}:\ell_-^w(\mathbb{R}^N)\times V_n\longrightarrow\ell_-^{w'}(\mathbb{R}^N)$ for which the relation (6.56) is replaced by

$$\left\|\!\left|D^r\mathcal{F}(\mathbf{x},\mathbf{z})\right|\!\right\|_{w,w'}\leq L_{F,r}L_w^rc_{w',w^r}<+\infty. \tag{6.57}$$

The rest of the statement is a consequence of part **(iii)** of Lemma 36 applied in this setup.

(iii) A computation similar to the one that was used to establish (6.54) leads to the following expression for the partial derivatives $D_x\mathcal{F}$ of $\mathcal{F}$:

$$D_x^r\mathcal{F}(\mathbf{x},\mathbf{z})=\prod_{t\in\mathbb{Z}_-}D_x^rF_t(\mathbf{x},\mathbf{z})=\prod_{t\in\mathbb{Z}_-}D_x^rF(\mathbf{x}_{t-1},\mathbf{z}_t)\circ(p_t\circ T_1,\ldots,p_t\circ T_1). \tag{6.58}$$

Using this expression for $r=1$ and Lemma 1 we can write, for any $\mathbf{u}\in\ell_-^w(\mathbb{R}^N)$,

$$\begin{aligned}
\left\|D_x\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0)\cdot\mathbf{u}\right\|_w &= \sup_{t\in\mathbb{Z}_-}\left\{\|D_xF(\mathbf{x}_{t-1}^0,\mathbf{z}_t^0)\circ(p_t\circ T_1)(\mathbf{u})\|w_{-t}\right\}\\
&\leq L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)\sup_{t\in\mathbb{Z}_-}\left\{\|p_t\|_w\|T_1(\mathbf{u})\|_ww_{-t}\right\}\\
&\leq L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)\sup_{t\in\mathbb{Z}_-}\left\{\frac{1}{w_{-t}}\|T_1(\mathbf{u})\|_ww_{-t}\right\}\leq L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w\|\mathbf{u}\|_w,
\end{aligned}$$

as required. ▼

We now proceed with the proof of the theorem in which we obtain the persistence result as a consequence of the Implicit Function Theorem and of the Lemma 42 that we just proved. Using the same notation as in that result we define the map

$$\begin{array}{rccc} \mathcal{G}: & \ell^w_-(\mathbb{R}^N)\times \ell^w_-(\mathbb{R}^n) & \longrightarrow & \ell^w_-(\mathbb{R}^N) \\ & (\mathbf{x},\mathbf{z}) & \longmapsto & \mathcal{F}(\mathbf{x},\mathbf{z})-\mathbf{x}, \end{array}$$

or equivalently, $\mathcal{G}=\mathcal{F}-\pi_N$, where $\pi_N:\ell^w_-(\mathbb{R}^N)\times \ell^w_-(\mathbb{R}^n)\longrightarrow \ell^w_-(\mathbb{R}^N)$ is just the projection onto the first factor.

Notice that by construction and the hypothesis on the point $(\mathbf{x}^0,\mathbf{z}^0)$ we have that

$$\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0)=\mathbf{0}. \tag{6.59}$$

Since the projection π_N is linear and by Lemma 42 $\mathcal{F}$ is Lipschitz continuous and differentiable of order 1, then so is $\mathcal{G}=\mathcal{F}-\pi_N$. This implies in particular that the partial derivative $D_x\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0):\ell^w_-(\mathbb{R}^N)\longrightarrow \ell^w_-(\mathbb{R}^N)$ is a bounded operator that we now set to prove that it is an isomorphism. We proceed in two stages that show how the hypotheses in the statement of the theorem imply that this linear map is both injective and surjective.

The partial derivative $D_x\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0):\ell^w_-(\mathbb{R}^N)\longrightarrow \ell^w_-(\mathbb{R}^N)$ is injective. Notice first that,

$$D_x\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0)\cdot\mathbf{u}=D_x\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0)\cdot\mathbf{u}-\mathbf{u},\quad \text{for any } \mathbf{u}\in\ell^w_-(\mathbb{R}^N).$$

Consequently, the points $\mathbf{u}\in\ell^w_-(\mathbb{R}^N)$ such that $D_x\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0)\cdot\mathbf{u}=\mathbf{0}$ coincide with the fixed points of the map $D_x\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0):\ell^w_-(\mathbb{R}^N)\longrightarrow \ell^w_-(\mathbb{R}^N)$. Since by part **(iii)** of Lemma 42 $D_x\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0)$ is a contracting linear map in $\ell^w_-(\mathbb{R}^N)$ it has hence only zero as unique fixed point and the claim follows.

The partial derivative $D_x\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0):\ell^w_-(\mathbb{R}^N)\longrightarrow \ell^w_-(\mathbb{R}^N)$ is surjective. We prove that for any $\mathbf{v}\in\ell^w_-(\mathbb{R}^N)$ there exists $\mathbf{u}\in\ell^w_-(\mathbb{R}^N)$ such that $D_x\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0)\cdot\mathbf{u}=\mathbf{v}$. By the definition of $\mathcal{F}$ in (6.49) and the expression of its partial derivative in (6.58), this equation is equivalent to the recursions,

$$\mathbf{v}_t=D_xF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\cdot\mathbf{u}_{t-1}-\mathbf{u}_t,\quad \text{for all } t\in\mathbb{Z}_-. \tag{6.60}$$

This equation has a unique solution given by the series

$$\mathbf{u}_t=-\mathbf{v}_t+\sum_{j=1}^{\infty}D_xF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\cdot D_xF(\mathbf{x}^0_{t-2},\mathbf{z}^0_{t-1})\cdots D_xF(\mathbf{x}^0_{t-j},\mathbf{z}^0_{t-j+1})(-\mathbf{v}_{t-j}),\quad t\in\mathbb{Z}_-. \tag{6.61}$$

Indeed, it is straightforward to show that (6.61) satisfies (6.60). It remains then to be shown that the sequence $\mathbf{u}$ determined by (6.61) belongs to $\ell^w_-(\mathbb{R}^N)$. In order to do so we first show that the series in (6.61) is convergent by proving that for any $t\in\mathbb{Z}_-$, the sequence $\{S_n\}_{n\in\mathbb{N}^+}$ defined by

$$S_n:=\sum_{j=1}^{n}D_xF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\cdot D_xF(\mathbf{x}^0_{t-2},\mathbf{z}^0_{t-1})\cdots D_xF(\mathbf{x}^0_{t-j},\mathbf{z}^0_{t-j+1})(-\mathbf{v}_{t-j})w_{-t}, \tag{6.62}$$

is a Cauchy sequence. This is so because for any $m,n\in\mathbb{N}^+$, $m\geq n$,

$$\begin{aligned}
&\|S_m-S_n\|= \\
&\left\|\sum_{j=n+1}^{m}D_xF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)\cdot D_xF(\mathbf{x}^0_{t-2},\mathbf{z}^0_{t-1})\cdots D_xF(\mathbf{x}^0_{t-j},\mathbf{z}^0_{t-j+1})(-\mathbf{v}_{t-j})\right\| w_{-(t-j)}\frac{w_{-(t-j+1)}}{w_{-(t-j)}}\cdots\frac{w_{-t}}{w_{-(t-1)}} \\
&\quad\leq\sum_{j=n+1}^{m}|||D_xF(\mathbf{x}^0_{t-1},\mathbf{z}^0_t)|||\cdot|||D_xF(\mathbf{x}^0_{t-2},\mathbf{z}^0_{t-1})|||\cdots|||D_xF(\mathbf{x}^0_{t-j},\mathbf{z}^0_{t-j+1})|||\,\|-\mathbf{v}_{t-j}\|\,w_{-(t-j)}L^j_w \\
&\quad\leq\sum_{j=n+1}^{m}L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)^jL^j_w\|\mathbf{v}\|_w=\frac{\left(L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w\right)^{n+1}-\left(L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w\right)^{m+1}}{1-L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w}\|\mathbf{v}\|_w,
\end{aligned} \tag{6.63}$$

which can be made as small as we want because the sequence $\{\left(L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w\right)^j\}_{j\in\mathbb{N}^+}$ is convergent and hence Cauchy due to the hypothesis $L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w<1$. This implies that $\{S_n\}_{n\in\mathbb{N}^+}$ is convergent and hence so is the series that defines $\mathbf{u}_t$ in (6.61).

It remains to be shown that the sequence $\mathbf{u}:=(\mathbf{u}_t)_{t\in\mathbb{Z}_-}$ defined by (6.61) is an element of $\ell^w_-(\mathbb{R}^N)$. Following the same strategy that we used to construct the inequalities (6.63) it is easy to see that

$$\|\mathbf{u}_t\|\,w_{-t}\leq\frac{1}{1-L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w}\|\mathbf{v}\|_w,\quad \text{for all } t\in\mathbb{Z}_-.$$

Consequently,

$$\|\mathbf{u}\|_w=\sup_{t\in\mathbb{Z}_-}\{\|\mathbf{u}_t\|\,w_{-t}\}\leq\frac{1}{1-L_{F_x}(\mathbf{x}^0,\mathbf{z}^0)L_w}\|\mathbf{v}\|_w<+\infty,$$

as required.

The partial derivative $D_x\mathcal{G}(\mathbf{x}^0,\mathbf{z}^0):\ell^w_-(\mathbb{R}^N)\longrightarrow \ell^w_-(\mathbb{R}^N)$ is a linear homeomorphism. This fact is a consequence of the Banach Isomorphism Theorem (see for instance Abraham et al. (1988)) that states that any continuous linear isomorphism of Banach spaces has necessarily a continuous inverse.

Using all the facts that we just proved, we can invoke the the Implicit Function Theorem as formulated in (Schechter, 1997, page 671) (see also Ver Eecke (1974)) to show the existence of two open neighborhoods $\widetilde{V}_{\mathbf{x}^0}$ and $\widetilde{V}_{\mathbf{z}^0}$ of $\mathbf{x}^0$ and $\mathbf{z}^0$ in $\ell^w_-(\mathbb{R}^N)$ and $\ell^w_-(\mathbb{R}^n)$, respectively, and a unique Lipschitz continuous map $\widetilde{U^F}:(\widetilde{V}_{\mathbf{z}^0},\|\cdot\|_w)\longrightarrow(\widetilde{V}_{\mathbf{x}^0},\|\cdot\|_w)$ that is differentiable at $\mathbf{z}^0$ and satisfies

$$\mathcal{G}(\widetilde{U^F}(\mathbf{z}),\mathbf{z})=\mathbf{0},\ \text{for all } \mathbf{z}\in\widetilde{V}_{\mathbf{z}^0},$$

which is equivalent to $\mathcal{F}(\widetilde{U^F}(\mathbf{z}),\mathbf{z})=U^F(\mathbf{z})$. In view of the identities (3.1) this means, in other words, that $\widetilde{U^F}$ is the unique reservoir filter with inputs in $\widetilde{V}_{\mathbf{z}^0}$ associated to the reservoir system determined by F. This filter is clearly causal and its Lipschitz continuity implies that it has the fading memory property.

We conclude the proof by showing that the filter $\widetilde{U^F}$ can be extended to a time-invariant filter U^F defined on the time-invariant saturations $V_{\mathbf{x}^0}$ and $V_{\mathbf{z}^0}$ of the sets $\widetilde{V}_{\mathbf{x}^0}$ and $\widetilde{V}_{\mathbf{z}^0}$, respectively, and that has the properties listed in the statement. Indeed, define

$$V_{\mathbf{x}^0}:=\bigcup_{t\in\mathbb{Z}_-}T_{-t}\left(\widetilde{V}_{\mathbf{x}^0}\right)\quad\text{and}\quad V_{\mathbf{z}^0}:=\bigcup_{t\in\mathbb{Z}_-}T_{-t}\left(\widetilde{V}_{\mathbf{z}^0}\right).$$

The sets $V_{\mathbf{x}^0}$ and $V_{\mathbf{z}^0}$ are by construction time-invariant and open by the openness of the maps T_{-t} that we established in part **(ii)** of Lemma 1. Define now the map $U^F : V_{\mathbf{z}^0} \longrightarrow V_{\mathbf{x}^0}$ as

$$U^F\left(T_{-t}(\mathbf{z})\right) := T_{-t}\left(\widetilde{U^F}(\mathbf{z})\right), \quad \text{for some} \quad t \in \mathbb{Z}_- \text{ and } \mathbf{z} \in \widetilde{V}_{\mathbf{z}^0}. \tag{6.64}$$

We first show that U^F is well-defined and time-invariant. Let $t_1, t_2 \in \mathbb{Z}_-$ and $\mathbf{z}_1, \mathbf{z}_2 \in \widetilde{V}_{\mathbf{z}^0}$ be such that $T_{-t_1}(\mathbf{z}_1) = T_{-t_2}(\mathbf{z}_2)$. Let us now show that

$$U^F\left(T_{-t_1}(\mathbf{z}_1)\right) = U^F\left(T_{-t_2}(\mathbf{z}_2)\right). \tag{6.65}$$

Indeed, for any $t \in \mathbb{Z}_-$, the definition (6.64) and the causality of $\widetilde{U^F}$ imply that

$$\begin{aligned}\left(U^F\left(T_{-t_1}(\mathbf{z}_1)\right)\right)_t &= \left(T_{-t_1}\left(\widetilde{U^F}(\mathbf{z}_1)\right)\right)_t \\ &= \widetilde{U^F}(\mathbf{z}_1)_{t+t_1} = \widetilde{U^F}(\mathbf{z}_2)_{t+t_2} = \left(T_{-t_2}\left(\widetilde{U^F}(\mathbf{z}_2)\right)\right)_t = \left(U^F\left(T_{-t_2}(\mathbf{z}_2)\right)\right)_t,\end{aligned}$$

which proves (6.65). The time-invariance of U^F, as defined in (6.64), is straightforward.

We conclude by showing that U^F is differentiable at all the points of the form $T_{-t}(\mathbf{z}^0)$, $t \in \mathbb{Z}_-$ and that it is locally Lipschitz continuous on $V_{\mathbf{z}^0}$. Since differentiability is a local property, it suffices to prove this property for the restriction of U^F to open sets. Before we do that, we note that since by part **(ii)** of Lemma 1 the map $T_{-t} : \widetilde{V}_{\mathbf{z}^0} \longrightarrow T_{-t}\left(\widetilde{V}_{\mathbf{z}^0}\right)$ is a submersion, the Local Onto Theorem (see (Abraham et al., 1988, Theorem 3.5.2)) guarantees that for $\mathbf{z}' := T_{-t}(\mathbf{z}^0) \in T_{-t}\left(\widetilde{V}_{\mathbf{z}^0}\right)$ there exists an open neighborhood $V_{\mathbf{z}'} \subset T_{-t}\left(\widetilde{V}_{\mathbf{z}^0}\right)$ and a smooth section $\sigma_{\mathbf{z}'} : V_{\mathbf{z}'} \longrightarrow \widetilde{V}_{\mathbf{z}^0}$ of T_{-t} that satisfies that

$$\sigma_{\mathbf{z}'}(\mathbf{z}') = \mathbf{z}^0 \quad \text{and} \quad T_{-t} \circ \sigma_{\mathbf{z}'} = \mathrm{id}_{V_{\mathbf{z}'}}. \tag{6.66}$$

The section $\sigma_{\mathbf{z}'}$ allows us to write down the restriction $U^F|_{V_{\mathbf{z}'}}$ of U^F to the open subset $V_{\mathbf{z}'}$ as

$$U^F|_{V_{\mathbf{z}'}}(\mathbf{z}) = T_{-t} \circ \widetilde{U^F}\left(\sigma_{\mathbf{z}'}(\mathbf{z})\right), \quad \text{for all} \quad \mathbf{z} \in V_{\mathbf{z}'}. \tag{6.67}$$

This is so because by (6.66) we have that $\mathbf{z} = T_{-t}\left(\sigma_{\mathbf{z}'}(\mathbf{z})\right)$, with $\sigma_{\mathbf{z}'}(\mathbf{z}) \in \widetilde{V}_{\mathbf{z}^0}$, as well as by (6.64). Consequently, since by (6.67) the restriction $U^F|_{V_{\mathbf{z}'}}$ is a composition of Lipschitz continuous functions then so is $U^F|_{V_{\mathbf{z}'}}$. The differentiability of U^F at the point $\mathbf{z}' = T_{-t}(\mathbf{z}^0)$ can also be concluded using (6.67) by invoking the differentiability of T_{-t} and $\sigma_{\mathbf{z}'}$ on their domains and the differentiability of $\widetilde{U^F}$ at $\sigma_{\mathbf{z}'}(\mathbf{z}') = \mathbf{z}^0$. ■

6.10. Proof of Theorem 19

(i) We start with a lemma that shows how condition (4.8) guarantees the existence of a globally defined filter $U^F : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^N), \|\cdot\|_w)$.

Lemma 43 *Let $F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$ be a reservoir map of class $C^1(\mathbb{R}^N \times \mathbb{R}^n)$ and let w be a weighting sequence with finite inverse decay ratio L_w. The reservoir map F is a contraction on the first entry if and only if*

$$L_{F_x} < 1. \tag{6.68}$$

Moreover, whenever conditions (4.1) and (4.8) are satisfied and $(\mathbf{x}^0, \mathbf{z}^0) \in (\mathbb{R}^N)^{\mathbb{Z}_-} \times (\mathbb{R}^n)^{\mathbb{Z}_-}$ is a solution of the reservoir system determined by F, then there exists a unique causal, time-invariant, and fading memory filter $U^F : (\ell^w_-(\mathbb{R}^n), \|\cdot\|_w) \longrightarrow (\ell^w_-(\mathbb{R}^N), \|\cdot\|_w)$.

Proof of the lemma. We first show that F is a contraction on the first entry if and only if $L_{F_x} < 1$. Suppose first that F is a contraction with contraction rate $0 < c < 1$. Then for any $(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^N \times \mathbb{R}^n$ and any $\mathbf{u} \in \mathbb{R}^N$, the partial derivative $D_xF(\mathbf{x}, \mathbf{z}) : \mathbb{R}^N \longrightarrow \mathbb{R}^N$ satisfies that

$$\|D_xF(\mathbf{x}, \mathbf{z}) \cdot \mathbf{u}\| = \lim_{t \to 0} \frac{\|F(\mathbf{x} + t\mathbf{u}, \mathbf{z}) - F(\mathbf{x}, \mathbf{z})\|}{t} \leq \lim_{t \to 0} \frac{ct\|\mathbf{u}\|}{t} = c\|\mathbf{u}\|,$$

which implies that $|||D_xF(\mathbf{x}, \mathbf{z})||| \leq c$ and hence

$$L_{F_x} := \sup_{(\mathbf{x}, \mathbf{z}) \in D_N \times D_n} \{|||D_xF(\mathbf{x}, \mathbf{z})|||\} \leq c < 1.$$

Conversely, suppose that $L_{F_x} < 1$. Since F is of class $C^1(\mathbb{R}^N \times \mathbb{R}^n)$, the mean value theorem guarantees that for any $(\mathbf{x}^1, \mathbf{z}), (\mathbf{x}^2, \mathbf{z}) \in \mathbb{R}^N \times \mathbb{R}^n$:

$$\left\|F(\mathbf{x}^1, \mathbf{z}) - F(\mathbf{x}^2, \mathbf{z})\right\| \leq \sup_{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^N \times \mathbb{R}^n} \{|||D_xF(\mathbf{x}, \mathbf{z})|||\} \left\|\mathbf{x}^1 - \mathbf{x}^2\right\| = L_{F_x} \left\|\mathbf{x}^1 - \mathbf{x}^2\right\|,$$

and F is hence is a contraction on the first entry.

Suppose now that conditions (4.1) and (4.8) are satisfied and that $(\mathbf{x}^0, \mathbf{z}^0) \in (\mathbb{R}^N)^{\mathbb{Z}_-} \times (\mathbb{R}^n)^{\mathbb{Z}_-}$ is a solution of the reservoir system determined by F. Notice first that since $L_w > 1$ then the condition (4.8) implies that $L_{F_x} < 1$, necessarily, and hence, as we just proved, F is a contraction on the first entry with constant L_{F_x}. Additionally, as (4.1) is satisfied, the mean value theorem implies that F is Lipschitz continuous with constant L_F. All these facts allow us to invoke part **(ii)** of Theorem 7 to conclude the existence of the filter U^F in the statement, since in this situation, the condition (3.3) coincides with (4.8). ▼

The proof of the first part of the theorem can now be obtained by applying Theorem 14 to each point of the form

$$(U^F(\mathbf{z}), \mathbf{z}) \in \ell^w_-(\mathbb{R}^N) \times \ell^w_-(\mathbb{R}^n)$$

for which, according to its statement, there exist open neighborhoods $V_{U^F(\mathbf{z})}$ and $V_{\mathbf{z}}$ of $U^F(\mathbf{z})$ and $\mathbf{z}$ in $\ell^w_-(\mathbb{R}^N)$ and $\ell^w_-(\mathbb{R}^n)$, as well as a unique locally defined causal reservoir filter $\widetilde{U}_F : V_{\mathbf{z}} \longrightarrow V_{U^F(\mathbf{z})}$ associated to F. The uniqueness feature implies that $\widetilde{U}_F = U^F|_{V_{\mathbf{z}}}$. Moreover, since $\widetilde{U}_F$ is differentiable at $\mathbf{z}$ and we can repeat this construction for any point $\mathbf{z} \in \ell^w_-(\mathbb{R}^n)$ we can conclude that U^F is differentiable at any point in $\ell^w_-(\mathbb{R}^n)$.

Finally, the Lipschitz continuity on $\ell^w_-(\mathbb{R}^n)$ of U^F is a consequence of the mean value theorem, the inequality (4.5), and the fact that

$$\sup_{\mathbf{z} \in \ell^w_-(\mathbb{R}^n)} \left\{\left|\left|\left|DU^F(\mathbf{z})\right|\right|\right|_w\right\} \leq \sup_{\mathbf{z} \in \ell^w_-(\mathbb{R}^n)} \left\{\frac{L_{F_z}(U^F(\mathbf{z}), \mathbf{z})}{1 - L_{F_x}(U^F(\mathbf{z}), \mathbf{z}))L_w}\right\} \leq \frac{L_{F_z}}{1 - L_{F_x}L_w},$$

which proves (4.9).

(ii) First of all, the existence of the filter $U^F : V_n \longrightarrow \ell^w_-(\mathbb{R}^N)$ and its differentiability at $\mathbf{z}^0 \in V_n$ imply that for any $\mathbf{u} \in \ell^w_-(\mathbb{R}^n)$ and $t \in \mathbb{Z}_-$ it satisfies (3.1) as well as (4.4), that is,

$$(DU^F(\mathbf{z}^0) \cdot \mathbf{u})_t = D_xF\left(U^F(\mathbf{z}^0)_{t-1}, \mathbf{z}^0_t\right) \cdot \left(DU^F(\mathbf{z}^0) \cdot \mathbf{u}\right)_{t-1} + D_zF(U(\mathbf{z}^0)_{t-1}, \mathbf{z}^0_t) \cdot \mathbf{u}_t.$$

This identity can be rewritten in terms of operators on sequences as

$$DU^F(\mathbf{z}^0) = \left(\prod_{t \in \mathbb{Z}_-} D_xF\left(U^F(\mathbf{z}^0)_{t-1}, \mathbf{z}^0_t\right)\right) \circ T_1 \circ DU^F(\mathbf{z}^0) + \prod_{t \in \mathbb{Z}_-} D_zF\left(U^F(\mathbf{z}^0)_{t-1}, \mathbf{z}^0_t\right),$$

or equivalently as

$$\left(\mathbb{I}_{\ell^w_-(\mathbb{R}^N)} - \left(\prod_{t\in\mathbb{Z}_-} D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\right)\circ T_1\right) DU^F(\mathbf{z}^0) = \prod_{t\in\mathbb{Z}_-} D_zF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right). \tag{6.69}$$

This identity determines $DU^F(\mathbf{z}^0)$ that by hypothesis exists if and only if the operator on the left hand side is invertible, which is in turn equivalent to the condition (4.10). We finally show that (4.10) implies (4.11).

We first notice that by Gelfand's formula (Lax, 2002, page 195) the condition (4.10) is equivalent to

$$\lim_{k\to+\infty}\left\|\left(\left(\prod_{t\in\mathbb{Z}_-} D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\right)\circ T_1\right)^k\right\|_w = 0.$$

This in turn implies that for any $\mathbf{u}\in\ell^w_-(\mathbb{R}^n)$, we have that

$$\lim_{k\to+\infty}\left\|\left(\left(\prod_{t\in\mathbb{Z}_-} D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\right)\circ T_1\right)^k(\mathbf{u})\right\|_w = 0,$$

or, equivalently, that

$$\lim_{k\to+\infty}\left(\sup_{t\in\mathbb{Z}_-}\left\{\left\|\left(D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\circ\cdots\circ D_xF\left(U^F(\mathbf{z}^0)_{t-k},\mathbf{z}^0_{t-k+1}\right)\right)(\mathbf{u}_{t-k})\right\|\right\}\right) = 0. \tag{6.70}$$

If we now take vectors $\mathbf{u}\in\ell^w_-(\mathbb{R}^n)$ in (6.70) of the form $\mathbf{u}_t := \widetilde{\mathbf{u}}/w_{-t}$, $t\in\mathbb{Z}_-$, with $\widetilde{\mathbf{u}}\in\mathbb{R}^n$ such that $\|\widetilde{\mathbf{u}}\|=1$, and we take the supremum in (6.70) with respect to all those vectors $\widetilde{\mathbf{u}}$, we obtain that

$$\lim_{k\to+\infty}\left(\sup_{t\in\mathbb{Z}_-}\left\{\left|\left|\left|D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\circ\cdots\circ D_xF\left(U^F(\mathbf{z}^0)_{t-k},\mathbf{z}^0_{t-k+1}\right)\right|\right|\right|\frac{w_{-t}}{w_{-(t-k)}}\right\}\right) = 0. \tag{6.71}$$

Given that

$$\begin{aligned}&\left|\left|\left|D_xF\left(U^F(\mathbf{z}^0)_{-1},\mathbf{z}^0_0\right)\circ\cdots\circ D_xF\left(U^F(\mathbf{z}^0)_{-k},\mathbf{z}^0_{-k+1}\right)\right|\right|\right|\frac{1}{w_k}\\ &\qquad\le \sup_{t\in\mathbb{Z}_-}\left\{\left|\left|\left|D_xF\left(U^F(\mathbf{z}^0)_{t-1},\mathbf{z}^0_t\right)\circ\cdots\circ D_xF\left(U^F(\mathbf{z}^0)_{t-k},\mathbf{z}^0_{t-k+1}\right)\right|\right|\right|\frac{w_{-t}}{w_{-(t-k)}}\right\},\end{aligned}$$

the condition (4.11) follows. ■

6.11. Proof of Theorem 28

It follows the same scheme as that of Theorem 14. In the following paragraphs we just hint the additional facts that need to be taken into account in order to adapt that proof to this setup.

The first complementary fact has to do with the second part of Lemma 42 which, using the hypothesis (4.29) allows us to conclude that the map $\mathcal{F}:\ell^\infty_-(\mathbb{R}^N)\times\ell^\infty_-(\mathbb{R}^n)\longrightarrow\ell^\infty_-(\mathbb{R}^N)$ defined in (6.38) is smooth. Additionally, it can be easily seen that it is also analytic and that the radii of convergence ρ_F and $\rho_{\mathcal{F}}$ of the Taylor series expansions of F and $\mathcal{F}$ around $(\mathbf{x}^0,\mathbf{z}^0)$ and the associated constant sequence (that we denote with the same symbol) satisfy

$$\rho_F \le \rho_{\mathcal{F}}. \tag{6.72}$$

Indeed, (6.54) implies that the Taylor series expansion of $\mathcal{F}$ around the constant sequence $(\mathbf{x}^0,\mathbf{z}^0)$ can be written, for any $\mathbf{u}^r := (\mathbf{u},\ldots,\mathbf{u}) = ((\mathbf{u_x},\mathbf{u_z}),\ldots,(\mathbf{u_x},\mathbf{u_z}))\in\left(\ell^\infty_-(\mathbb{R}^N)\oplus\ell^\infty_-(\mathbb{R}^n)\right)^r$, as

$$\begin{aligned}\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0)+\sum_{r=1}^{\infty}\frac{1}{r!}D^r\mathcal{F}(\mathbf{x}^0,\mathbf{z}^0)\cdot(\mathbf{u}-(\mathbf{x}^0,\mathbf{z}^0))^r &= \prod_{t\in\mathbb{Z}_-}\left(F_t(\mathbf{x}^0,\mathbf{z}^0)+\sum_{r=1}^{\infty}\frac{1}{r!}D^rF_t(\mathbf{x}^0,\mathbf{z}^0)\cdot(\mathbf{u}-(\mathbf{x}^0,\mathbf{z}^0))^r\right)\\ &= \prod_{t\in\mathbb{Z}_-}\left(F(\mathbf{x}^0,\mathbf{z}^0)+\sum_{r=1}^{\infty}\frac{1}{r!}D^rF(\mathbf{x}_{t-1},\mathbf{z}_t)\circ(p_t\circ(T_1\times\mathrm{id}),\ldots,p_t\circ(T_1\times\mathrm{id}))\cdot(\mathbf{u}-(\mathbf{x}^0,\mathbf{z}^0))^r\right).\end{aligned} \tag{6.73}$$

Suppose now that $\mathbf{u}=(\mathbf{u_x},\mathbf{u_z})\in\ell^\infty_-(\mathbb{R}^N)\oplus\ell^\infty_-(\mathbb{R}^n)$ is chosen such that

$$\|\mathbf{u}\|_\infty = \|\mathbf{u_x}\|_\infty + \|\mathbf{u_z}\|_\infty < \rho_F. \tag{6.74}$$

Lemma 1 implies that for any $t\in\mathbb{Z}_-$, we have in that case that

$$\|p_t\circ(T_1\times\mathrm{id})(\mathbf{u})\| \le \|\mathbf{u}\|_\infty < \rho_F$$

and hence we can conclude that all the series labeled by $t\in\mathbb{Z}_-$ in each of the factors that make up the last term of (6.73) converge for all the elements $\mathbf{u}\in\ell^\infty_-(\mathbb{R}^N)\oplus\ell^\infty_-(\mathbb{R}^n)$ that satisfy (6.74). This implies that such elements are inside the radius of convergence of the Taylor series expansion of $\mathcal{F}$ around the constant sequence $(\mathbf{x}^0,\mathbf{z}^0)$ and hence (6.72) holds which, as ρ_F is nontrivial by hypothesis, proves that $\mathcal{F}$ is analytic.

The rest of the proof can be obtained by mimicking that of Theorem 14 where, as it is customary, we replace the weighting sequence w by the constant sequence w^ι given by $w^\iota_t := 1$, for all $t\in\mathbb{N}$, and L_w is replaced by the constant 1.

A technical modification is needed at the time of invoking the Implicit Function Theorem. In Theorem 14 we used a version that requires only first order differentiability as hypothesis and produces Lipschitz continuous implicitly defined functions. In this case we can prove that the function $\mathcal{G}$ is analytic and hence it can be shown that the implicitly defined local filter $\widehat{U^F}:(\widetilde{V}_{\mathbf{z}^0},\|\cdot\|_w)\longrightarrow(\widetilde{V}_{\mathbf{x}^0},\|\cdot\|_w)$ is analytic by invoking, for instance, (Valent, 1988, page 175), and references therein. ■

6.12. Proof of Theorem 29

Since by hypothesis U is analytic in $B_{\|\cdot\|_w}(\mathbf{z}^0,M)$ then

$$U(\mathbf{z}) = U(\mathbf{z}^0)+\sum_{j=1}^{\infty}\frac{1}{j!}D^jU(\mathbf{z}^0)(\underbrace{\mathbf{z}-\mathbf{z}^0,\ldots,\mathbf{z}-\mathbf{z}^0}_{j\text{ times}}),\quad\text{for any}\quad \mathbf{z}\in B_{\|\cdot\|_w}(\mathbf{z}^0,M). \tag{6.75}$$

We now show that for the elements that satisfy (5.1) the series expansion (6.75) amounts to the discrete-time Volterra series expansion (5.2). Let $m\in\mathbb{Z}_-$ and let $\delta_m\in\ell^w_-(\mathbb{R})$ be the sequence defined by

$$(\delta_m)_t := \begin{cases}\frac{1}{w_{-m}} & \text{if } t=m,\\ 0 & \text{otherwise.}\end{cases} \tag{6.76}$$

Note that $\|\delta_m\|_w = 1$ for all $m\in\mathbb{Z}_-$. Moreover, for any $\mathbf{z}\in\ell^w_-(\mathbb{R})$ we can write

$$\mathbf{z}-\mathbf{z}^0 = \sum_{t\in\mathbb{Z}_-}\widetilde{z}_t\delta_t,\quad\text{with}\quad \widetilde{z}_t = (z_t-z^0_t)w_{-t},$$

and hence by the multilinearity of the derivatives $D^jU(\mathbf{z}^0)(\mathbf{z}-\mathbf{z}^0,\ldots,\mathbf{z}-\mathbf{z}^0)$ and the causality of the filter U we have that

$$D^jU(\mathbf{z}^0)(\mathbf{z}-\mathbf{z}^0,\ldots,\mathbf{z}-\mathbf{z}^0)_t=\sum_{m_1=-\infty}^{t}\cdots\sum_{m_j=-\infty}^{t}\widetilde{z}_{m_1}\cdots\widetilde{z}_{m_j}D^jU(\mathbf{z}^0)(\delta_{m_1},\ldots,\delta_{m_j})_t,\ \text{for all } t\in\mathbb{Z}_-. \tag{6.77}$$

We first show that for the elements that satisfy (5.1) the sum in the right hand side of (6.77) is finite. Indeed, for any $t\in\mathbb{Z}_-$:

$$\begin{aligned}
&\left|\sum_{m_1=-\infty}^{t}\cdots\sum_{m_j=-\infty}^{t}\widetilde{z}_{m_1}\cdots\widetilde{z}_{m_j}D^jU(\mathbf{z}^0)(\delta_{m_1},\ldots,\delta_{m_j})_t\right|\\
&\quad=\left|\sum_{m_1=-\infty}^{t}\cdots\sum_{m_j=-\infty}^{t}\widetilde{z}_{m_1}\cdots\widetilde{z}_{m_j}\frac{1}{w_{-t}}w_{-t}D^jU(\mathbf{z}^0)(\delta_{m_1},\ldots,\delta_{m_j})_t\right|\\
&\quad\le\sum_{m_1=-\infty}^{t}\cdots\sum_{m_j=-\infty}^{t}\left|\widetilde{z}_{m_1}\cdots\widetilde{z}_{m_j}\right|\frac{1}{w_{-t}}\|D^jU(\mathbf{z}^0)(\delta_{m_1},\ldots,\delta_{m_j})\|_w\\
&\quad\le\sum_{m_1=-\infty}^{t}\cdots\sum_{m_j=-\infty}^{t}\frac{\left|\widetilde{z}_{m_1}\cdots\widetilde{z}_{m_j}\right|}{w_{-t}}|||D^jU(\mathbf{z}^0)|||_w\|(\delta_{m_1},\ldots,\delta_{m_j})\|_w\\
&\quad=j\frac{|||D^jU(\mathbf{z}^0)|||_w}{w_{-t}}\sum_{m_1=-\infty}^{t}\cdots\sum_{m_j=-\infty}^{t}|\widetilde{z}_{m_1}|\cdots|\widetilde{z}_{m_j}|\\
&\quad=j\frac{|||D^jU(\mathbf{z}^0)|||_w}{w_{-t}}\left(\sum_{m=-\infty}^{t}\left|z_m-z_m^0\right|w_{-m}\right)^j<+\infty,
\end{aligned}\tag{6.78}$$

where the last equality is a consequence of, for example, (Apostol, 1974, Theorem 8.44), and the last inequality follows from two facts. First, as U is analytic, it is in particular smooth and hence $|||D^jU(\mathbf{z}^0)|||_w<+\infty$ for all $j\in\mathbb{Z}_-$. Second, since by hypothesis $\mathbf{z},\mathbf{z}^0\in\ell^{1,w}_-(\mathbb{R}^n)$ then $\mathbf{z}-\mathbf{z}^0\in\ell^{1,w}_-(\mathbb{R}^n)$ and hence $\sum_{m=-\infty}^{t}\left|z_m-z_m^0\right|w_{-m}<+\infty$.

We now show that (6.75) can be rewritten as (5.2). Notice first that for any $t,m\in\mathbb{Z}_-$ such that $m\le t$, the sequences (6.76) satisfy

$$T_{-t}\left(\delta_m\right)=\frac{w_{-(m-t)}}{w_{-m}}\delta_{m-t}. \tag{6.79}$$

Second, the time-invariance of U and of the sequence $\mathbf{z}^0$, imply that for any $j\in\mathbb{N}^+$, $t\in\mathbb{Z}_-$, and $\mathbf{z}^1,\ldots,\mathbf{z}^j\in\ell^w_-(\mathbb{R})$, we have that

$$T_{-t}\left(D^jU(\mathbf{z}^0)\left(\mathbf{z}^1,\ldots,\mathbf{z}^j\right)\right)=D^jU(T_{-t}\left(\mathbf{z}^0\right))\left(T_{-t}\left(\mathbf{z}^1\right),\ldots,T_{-t}\left(\mathbf{z}^j\right)\right)=D^jU(\mathbf{z}^0)\left(T_{-t}\left(\mathbf{z}^1\right),\ldots,T_{-t}\left(\mathbf{z}^j\right)\right).$$

These two relations imply that for any $t\in\mathbb{Z}_-$

$$\begin{aligned}
D^jU(\mathbf{z}^0)(\delta_{m_1},\ldots,\delta_{m_j})_t&=\left(T_{-t}\left(D^jU(\mathbf{z}^0)(\delta_{m_1},\ldots,\delta_{m_j})\right)\right)_0=D^jU(\mathbf{z}^0)(T_{-t}(\delta_{m_1}),\ldots,T_{-t}(\delta_{m_j}))_0\\
&=D^jU(\mathbf{z}^0)(\delta_{m_1-t},\ldots,\delta_{m_j-t})_0\frac{w_{-(m_1-t)}}{w_{-m_1}}\cdots\frac{w_{-(m_j-t)}}{w_{-m_j}}.
\end{aligned}$$

If we substitute this relation in the summands of (6.77), we obtain that

$$\begin{aligned}
&\widetilde{z}_{m_1}\cdots\widetilde{z}_{m_j}D^jU(\mathbf{z}^0)(\delta_{m_1},\ldots,\delta_{m_j})_t\\
&\qquad=(z_{m_1}-z^0_{m_1})\cdots(z_{m_j}-z^0_{m_j})\cdot w_{-(m_1-t)}\cdots w_{-(m_j-t)}\cdot D^jU(\mathbf{z}^0)(\delta_{m_1-t},\ldots,\delta_{m_j-t})_0.
\end{aligned}\tag{6.80}$$

Define now

$$\begin{aligned}
g_j(n_1,\ldots,n_j)&:=w_{-n_1}\cdots w_{-n_j}\frac{1}{j!}D^jU(\mathbf{z}^0)(\delta_{n_1},\ldots,\delta_{n_j})_0\\
&=w_{-n_1}\cdots w_{-n_j}\frac{1}{j!}D^jH(\mathbf{z}^0)(\delta_{n_1},\ldots,\delta_{n_j})_0=\frac{1}{j!}D^jH(\mathbf{z}^0)(e_{n_1},\ldots,e_{n_j})_0,
\end{aligned}\tag{6.81}$$

where $e_m\in\ell^w_-(\mathbb{R})$ is the sequence defined in (5.3). If we make the change of variables $n_i:=m_i-t$ in (6.80), we use (6.81), and we insert the resulting expression in (6.77) and subsequently in (6.75) we obtain (5.2). The uniqueness of this series expansion follows from the same argument as in (Sandberg, 1999, Theorem 1).

We now prove the error estimates (5.4) with the same strategy as in Sandberg (1999). Using the Cauchy bounds for analytic functions (see, for instance, the last expression in (Hille and Phillips, 1957, page 112)) and the analyticity hypothesis on $U:B_{\|\cdot\|_w}(\mathbf{z}^0,M)\subset\ell^w_-(\mathbb{R})\longrightarrow B_{\|\cdot\|_w}(U(\mathbf{z}^0),L)\subset\ell^w_-(\mathbb{R}^N)$, we have that for any $j\in\mathbb{N}^+$ and $t\in\mathbb{Z}_-$

$$\|D^jU(\mathbf{z}^0)(\mathbf{z},\ldots,\mathbf{z})_t\|=\|p_t\circ D^jU(\mathbf{z}^0)(\mathbf{z},\ldots,\mathbf{z})\|\le|||p_t|||_w\|D^jU(\mathbf{z}^0)(\mathbf{z},\ldots,\mathbf{z})\|_w\le\frac{j!L}{w_{-t}}\left(\frac{\|\mathbf{z}\|_w}{M}\right)^j, \tag{6.82}$$

where we also used the first part of Lemma 1. Now, as we saw in the previous paragraphs,

$$\begin{aligned}
&\left\|U(\mathbf{z})_t-U(\mathbf{z}^0)_t-\sum_{j=1}^{p}\sum_{m_1=-\infty}^{0}\cdots\sum_{m_j=-\infty}^{0}g_j(m_1,\ldots,m_j)(z_{m_1+t}-z^0_{m_1+t})\cdots(z_{m_j+t}-z^0_{m_j+t})\right\|\\
&\le\left\|U(\mathbf{z})_t-U(\mathbf{z}^0)_t-\sum_{j=1}^{p}\frac{1}{j!}D^jU(\mathbf{z}^0)(\mathbf{z}-\mathbf{z}^0,\ldots,\mathbf{z}-\mathbf{z}^0)\right\|=\sum_{j=p+1}^{\infty}\frac{1}{j!}D^jU(\mathbf{z}^0)(\mathbf{z}-\mathbf{z}^0,\ldots,\mathbf{z}-\mathbf{z}^0)\\
&\qquad\le\frac{L}{w_{-t}}\sum_{j=p+1}^{\infty}\left(\frac{\|\mathbf{z}\|_w}{M}\right)^j\le\frac{L}{w_{-t}}\left(1-\frac{\|\mathbf{z}\|_w}{M}\right)^{-1}\left(\frac{\|\mathbf{z}\|_w}{M}\right)^{p+1},
\end{aligned}\tag{6.83}$$

where the inequalities in the last line follow from (6.82). ■

6.13. Time invariance of the solutions of a reservoir system

The filters studied in this paper are those determined by reservoir systems of the type introduced in (1.1)–(1.2). As we already pointed out, in that case we can associate unique reservoir filters U^F and U^F_h to the reservoir map F and the reservoir system, respectively, whenever (1.1) satisfies the echo state property. In that case, it has been shown in (Grigoryeva and Ortega, 2018b, Proposition 2.1) that both U^F and U^F_h are necessarily causal and time-invariant. We complement this fact with a similar elementary statement that does not require the echo state property or the existence reservoir filters.

Lemma 44 *Let $(\mathbf{x}^0,\mathbf{z}^0)\in\left(\mathbb{R}^N\right)^{\mathbb{Z}_-}\times\left(\mathbb{R}^n\right)^{\mathbb{Z}_-}$ be a solution of the reservoir system determined by the map $F:\mathbb{R}^N\times\mathbb{R}^n\longrightarrow\mathbb{R}^N$. Then, for any $\tau\in\mathbb{Z}_-$, the pair $(T_\tau(\mathbf{x}^0),T_\tau(\mathbf{z}^0))\in\left(\mathbb{R}^N\right)^{\mathbb{Z}_-}\times\left(\mathbb{R}^n\right)^{\mathbb{Z}_-}$ is also a solution.*

Proof. By hypothesis, for any $t \in \mathbb{Z}_-$ we have that

$$F(\mathbf{x}^0_{t-1}, \mathbf{z}^0_t) = \mathbf{x}^0_t,$$

and hence

$$F\left(T_\tau(\mathbf{x}^0)_{t-1}, T_\tau(\mathbf{z}^0)_t\right) = F(\mathbf{x}^0_{t-\tau-1}, \mathbf{z}^0_{t-\tau}) = \mathbf{x}^0_{t-\tau} = T_\tau(\mathbf{x}^0)_t, \quad \text{as required.} \quad \blacksquare$$

Acknowledgments

We thank Lukas Gonon and Herbert Jaeger for fruitful discussions as well as the input from the two anonymous referees that helped in improving the paper. The authors acknowledge partial financial support of the French ANR "BIPHOPROC" project (ANR-14-OHRI-0002-02). LG acknowledges partial financial support of the Graduate School of Decision Sciences of the Universität Konstanz. JPO acknowledges partial financial support coming from the Research Commission of the Universität Sankt Gallen and the Swiss National Science Foundation (grant number 200021_175801/1).

Glossary of Symbols

$\ell^\infty_-(\mathbb{R}^n)$	Banach space formed by the semi-infinite sequences that have a finite supremum norm
$\ell^{p,w}_-(\mathbb{R}^n)$	Banach space formed by the semi-infinite sequences that have a finite (p,w)-norm
$\ell^p_-(\mathbb{R}^n)$	Banach space formed by the semi-infinite sequences that have a finite p-norm
$\ell^w_-(\mathbb{R}^n)$	Banach space formed by the semi-infinite sequences that have a finite weighted supremum norm
$\mathbb{M}_N$	Space of real square matrices of size N
$\mathcal{F}^F$	Reservoir flow associated to the reservoir map F
$\prod_{t\in\mathbb{Z}_-} A_t$	Cartesian product of the sets A_t
$\prod_{t\in\mathbb{Z}_-} f_t$	Cartesian product of the functions f_t
$\rho(A)$	Spectral radius of the matrix A
σ	Activation function (in ESN, for example)
c	Contraction constant on the first entry of the reservoir map
d	Dimension of the elements of the output signal
$D^r f(\mathbf{z})$	r-order Fréchet differential of the map f at the point $\mathbf{z}$
D_w	Decay ratio of the weighting sequence w
$D_{\mathbf{x}} f(\mathbf{x},\mathbf{z})$	Partial derivative of the map f with respect to the first entry at the point $(\mathbf{x},\mathbf{z})$
$Df(\mathbf{z})$	Fréchet differential of the map f at the point $\mathbf{z}$
$F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$	Reservoir map
$H_U : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^d$	Functional associated to the causal and time-invariant filter $U : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}}$
$h : \mathbb{R}^N \to \mathbb{R}^d$	Generic readout map
K_M	Space of semi-infinite sequences that are uniformly bounded by M
L_σ	Lipschitz constant of the activation function σ
L_F	Lipschitz constant of the reservoir map F

L_w	Inverse decay ratio of the weighting sequence w
L_{F_x}	Lipschitz constant on the first entry of the reservoir map F
L_{U^F}	Lipschitz constant of the reservoir filter U^F
N	Number of virtual neurons. Dimension of the reservoir state vectors
n	Dimension of the elements of the input signal
$p_t : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^n$	Projection onto the tth-entry
$T_\tau : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}$	Time delay operator defined on semi-infinite sequences
$T^{\mathbb{Z}}_\tau : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}}$	Time delay operator defined on two-sided infinite sequences
$U^F_h : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}}$	Reservoir filter determined by the reservoir map F and the readout h
$U^F : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^N)^{\mathbb{Z}}$	Filter determined by the reservoir map F
$U : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}}$	Filter with inputs in $\mathbb{R}^n$ and outputs in $\mathbb{R}^d$
$U^{A,\mathbf{c}}_h : K_M \longrightarrow \mathbb{R}^{\mathbb{Z}}$	Linear reservoir filter determined by $A, \mathbf{c}$, and the polynomial h
$U_H : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}}$	Causal and time-invariant filter associated to the functional $H : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}^d$
$w : \mathbb{N} \longrightarrow (0,1]$	Weighting sequence
$\mathbf{x}$	(Semi)-infinite sequence containing the reservoir states. The elements of this sequence are denoted by $\mathbf{x}_t \in \mathbb{R}^N$
$\mathbf{y}$	(Semi)-infinite output signal. The elements of this sequence are denoted by $y_t \in \mathbb{R}^d$
$\mathbf{z}$	(Semi)-infinite input signal. The elements of this sequence are denoted by $\mathbf{z}_t \in \mathbb{R}^n$

References

R. Abraham, J. E. Marsden, and T. S. Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75. Applied Mathematical Sciences. Springer-Verlag, 1988.

T. Apostol. *Mathematical Analysis*. Addison Wesley, second edition, 1974.

L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer. Information processing using a single dynamical node as complex system. *Nature Communications*, 2:468, jan 2011.

L. Arnold. *Random Dynamical Systems*. Springer, 1998.

S. Boyd and L. Chua. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, nov 1985.

D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Communications*, 4(1364), 2013.

J. Cabessa and A. E. Villa. Computational capabilities of recurrent neural networks based on their attractor dynamics. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, jul 2015.

J. Cabessa and A. E. Villa. Expressive power of first-order recurrent neural networks determined by their attractor dynamics. *Journal of Computer and System Sciences*, 82(8):1232–1250, 2016.

P. Chossat, D. Lewis, J.-P. Ortega, and T. S. Ratiu. Bifurcation of relative equilibria in mechanical systems with symmetry. *Advances in Applied Mathematics*, 31:10–45, 2003.

B. D. Coleman and V. J. Mizel. On the general theory of fading memory. *Archive for Rational Mechanics and Analysis*, 29(1):18–31, jan 1968.

R. Couillet, G. Wainrib, H. Sevi, and H. T. Ali. The asymptotic performance of linear echo state neural networks. *Journal of Machine Learning Research*, 17(178):1–35, 2016.

J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar. Information processing capacity of dynamical systems. *Scientific reports*, 2(514), 2012.

M. Fabrizio, C. Giorgi, and V. Pata. *A new approach to equations with memory*, volume 198. 2010.

S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 105(48):18970–5, dec 2008.

F. Girosi. Approximation error bounds that use VC-bounds. In F. Fogelman-Soulie and P. Gallinari, editors, *Proc. International Conference on Artificial Neural Networks*, volume 1, pages 295–302, 1995.

F. Girosi and G. Anzellotti. Convergence rates of approximation by translates. Technical report, Defense Technical Information Center, 1992.

L. Gonon and J.-P. Ortega. Reservoir computing universality with stochastic inputs. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

L. Grigoryeva and J.-P. Ortega. Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. *Journal of Machine Learning Research*, 19(24):1–40, 2018a.

L. Grigoryeva and J.-P. Ortega. Echo state networks are universal. *Neural Networks*, 108:495–508, 2018b.

L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. Optimal nonlinear information processing capacity in delay-based reservoir computers. *Scientific Reports*, 5(12858):1–11, 2015.

L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. Nonlinear memory capacity of parallel time-delay reservoir computers in the processing of multidimensional signals. *Neural Computation*, 28: 1411–1451, 2016.

H. Gunawan, S. Konca, and M. Idris. p-summable sequence spaces with inner products. *Bitlis Eren University Journal of Science and Technology*, 5(1):1–9, 2015.

A. G. Hart, J. L. Hook, and J. H. P. Dawes. Embedding and approximation theorems for echo state networks. *Preprint*, 2019.

M. Hermans and B. Schrauwen. Memory in linear recurrent neural networks in continuous time. *Neural networks : the official journal of the International Neural Network Society*, 23(3):341–55, apr 2010.

E. Hille and R. S. Phillips. *Functional Analysis and Semi-Groups*. American Mathematical Society, 1957.

B. R. Hunt, E. Ott, and J. A. Yorke. Differentiable generalized synchronization of chaos. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 55(4):4029–4034, 1997.

H. Jaeger. Short term memory in echo state networks. *Fraunhofer Institute for Autonomous Intelligent Systems. Technical Report.*, 152, 2002.

H. Jaeger. The 'echo state' approach to analysing and training recurrent neural networks with an erratum note. Technical report, German National Research Center for Information Technology, 2010.

H. Jaeger and H. Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, 2004.

J. Kilian and H. T. Siegelmann. The dynamic universality of sigmoidal neural networks. *Information and Computation*, 128(1):48–56, 1996.

P. E. Kloeden. Synchronization of nonautonomous dynamical systems. *Electronic Journal of Differential Equations*, 2003(39):1–10, 2003.

P. E. Kloeden and M. Rasmussen. *Nonautonomous Dynamical Systems*. American Mathematical Society, 2010.

L. Kocarev and U. Parlitz. General approach for chaotic synchronization with applications to communication. *Physical Review Letters*, 74(25):5028–5031, 1995.

L. Kocarev and U. Parlitz. Generalized synchronization, predictability, and equivalence of unidirectionally coupled dynamical systems. *Physical Review Letters*, 76(11):1816–1819, 1996.

F. Laporte, A. Katumba, J. Dambre, and P. Bienstman. Numerical demonstration of neuromorphic computing with photonic crystal cavities. *Optics Express*, 26(7):7955, apr 2018.

L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. *Optics Express*, 20(3):3241, jan 2012.

P. Lax. *Functional Analysis*. Wiley-Interscience, 2002.

R. Legenstein and W. Maass. What makes a dynamical system computationally powerful? In S. Haykin, editor, *New directions in statistical signal processing: from systems to brain*. MIT Press, Cambridge, MA, 2007.

A. Lindquist and G. Picci. *Linear Stochastic Systems*. Springer-Verlag, 2015.

Z. Lu, B. R. Hunt, and E. Ott. Attractor reconstruction by machine learning. *Chaos*, 28(6), 2018.

M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14:2531–2560, 2002.

W. Maass. Liquid state machines: motivation, theory, and applications. In S. S. Barry Cooper and A. Sorbi, editors, *Computability In Context: Computation and Logic in the Real World*, chapter 8, pages 275–296. 2011.

W. Maass and E. D. Sontag. Neural Systems as Nonlinear Filters. *Neural Computation*, 12(8):1743–1772, aug 2000.

W. Maass, T. Natschläger, and H. Markram. Fading memory and kernel properties of generic cortical microcircuit models. *Journal of Physiology Paris*, 98(4-6 SPEC. ISS.):315–330, 2004.

W. Maass, P. Joshi, and E. D. Sontag. Computational aspects of feedback in neural circuits. *PLoS Computational Biology*, 3(1):e165, 2007.

G. Manjunath and H. Jaeger. Echo state property linked to an input: exploring a fundamental characteristic of recurrent neural networks. *Neural Computation*, 25(3):671–696, 2013.

G. Manjunath and H. Jaeger. The dynamics of random difference equations Is remodeled by closed relations. *SIAM Journal on Mathematical Analysis*, 46(1):459–483, jan 2014.

J. A. Montaldi. Persistence and stability of relative equilibria. *Nonlinearity*, 10:449–466, 1997a.

J. A. Montaldi. Persistance d'orbites p{é}riodiques relatives dans les syst{è}mes hamiltoniens sym{é}triques. *C. R. Acad. Sci. Paris S{é}r. I Math.*, 324:553–558, 1997b.

J. Munkres. *Topology*. Pearson, second edition, 2014.

T. Natschläger, W. Maass, and H. Markram. The "Liquid Computer": a novel strategy for real-time computing on time series. *Special Issue on Foundations of Information Processing of TELEMATIK*, 8(1):39–43, 2002.

J. Newman. Necessary and sufficient conditions for stable synchronization in random dynamical systems. *Ergodic Theory and Dynamical Systems*, 38(5):1857–1875, 2018.

J.-P. Ortega and T. S. Ratiu. Persistence and smoothness of critical relative elements in Hamiltonian systems with symmetry. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 325 (10):1107–1111, nov 1997.

Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar. Optoelectronic reservoir computing. *Scientific reports*, 2:287, jan 2012.

J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos*, 27(12), 2017.

J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Physical Review Letters*, 120(2): 24102, 2018.

M. B. Priestley. *Non-linear and Non-stationary Time Series Analysis*. Academic Press, 1988.

A. Rekic-Vukovic, N. Okicic, and E. Dunjakovic. On weighted Banach sequence spaces. *Advances in Mathematics: Scientific Journal*, 4(2):127–138, 2015.

A. Rodan and P. Tino. Minimum complexity echo state network. *IEEE Transactions on Neural Networks*, 22(1):131–44, jan 2011.

W. J. Rugh. *Nonlinear System Theory. The Volterra/Wiener Approach*. The Johns Hopkins University Press, 1981.

I. W. Sandberg. Time-delay neural networks, Volterra series, and rates of approximation. *Circuits, Systems, and Signal Processing*, 17(5):653–655, 1998a.

I. W. Sandberg. A note on representation theorems for linear discrete-space systems. *Circuits, Systems, and Signal Processing*, 17(6):703–708, 1998b.

I. W. Sandberg. A representation theorem for linear discrete-space systems. *Mathematical Problems in Engineering*, 4:369–375, 1998c.

I. W. Sandberg. Bounds for discrete-time Volterra series representations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 46(1):135–139, 1999.

I. W. Sandberg. Notes of fading-memory conditions. *Circuits, Systems, and Signal Processing*, 22(1): 43–55, 2003.

E. Schechter. *Handbook of Analysis and its Foundations*, volume 91. Academic Press, 1997.

M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. Wiley, 1980.

J. H. Shapiro. *A Fixed-Point Farrago*. Springer International Publishing Switzerland, 2016.

H. Siegelmann, B. Horne, and C. Giles. Computational capabilities of recurrent NARX neural networks. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 27(2):208–215, apr 1997.

S. Sternberg. *Dynamical Systems*. Dover, 2010.

G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115: 100–123, 2019.

P. Tio. Asymptotic Fisher memory of randomized linear symmetric Echo State Networks. *Neurocomputing*, 298:4–8, 2018.

T. Valent. *Boundary Value Problems of Finite Elasticity*. Springer Verlag, 1988.

K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman. Parallel reservoir computing using optical amplifiers. *IEEE Transactions on Neural Networks*, 22(9):1469–1481, sep 2011.

K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature Communications*, 5:78–80, mar 2014.

P. Ver Eecke. Sur le calcul différentiel dans les espaces vectoriels topologiques. *Cahiers de topologie et géométrie différentielle catégoriques*, 15(3):293–339, 1974.

Q. Vinckier, F. Duport, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar. High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica*, 2(5):438–446, 2015.

V. Volterra. *Theory of Functionals and of Integral and Integro-Differential Equations*. Blackie & Son Limited, Glasgow, 1930.

O. White, D. Lee, and H. Sompolinsky. Short-Term Memory in Orthogonal Neural Networks. *Physical Review Letters*, 92(14):148102, apr 2004.

N. Wiener. *Nonlinear Problems in Random Theory*. The Technology Press of MIT, 1958.

I. B. Yildiz, H. Jaeger, and S. J. Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, nov 2012.

Journal of Machine Learning Research 20 (2019) 1-7 Submitted 3/19; Revised 8/19; Published 10/19

DPPy: DPP Sampling with Python

Guillaume Gautier[†*] G.GAUTIER@INRIA.FR
Guillermo Polito[†*] GUILLERMO.POLITO@UNIV-LILLE.FR
Rémi Bardenet[†] REMI.BARDENET@GMAIL.COM
Michal Valko[‡*†] VALKOM@DEEPMIND.COM
[†] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL, 59651 Villeneuve d'Ascq, France*
[*] *Inria Lille-Nord Europe, 40 avenue Halley 59650 Villeneuve d'Ascq, France*
[‡] *DeepMind Paris, 14 Rue de Londres, 75009 Paris, France*

Editor: Andreas Mueller

Abstract

Determinantal point processes (DPPs) are specific probability distributions over clouds of points that are used as models and computational tools across physics, probability, statistics, and more recently machine learning. Sampling from DPPs is a challenge and therefore we present DPPy, a Python toolbox that gathers known exact and approximate sampling algorithms for both finite and continuous DPPs. The project is hosted on GitHub and equipped with an extensive documentation.

Keywords: determinantal point processes, sampling, MCMC, random matrices, Python

1. Introduction

Determinantal point processes (DPPs) are distributions over configurations of points that encode diversity through a kernel function K. They were introduced by Macchi (1975) as models for beams of fermions, and they have since found applications in fields as diverse as probability (Soshnikov, 2000; König, 2004; Hough et al., 2006), statistical physics (Pathria & Beale, 2011), Monte Carlo methods (Bardenet & Hardy, 2019), spatial statistics (Lavancier et al., 2012), and machine learning (ML, Kulesza & Taskar, 2012).

In ML, DPPs are mainly used as models for diverse sets of items, as in recommendation (Kathuria et al., 2016; Gartrell et al., 2016) or text summarization (Dupuy & Bach, 2018). Consequently, MLers mostly use finite DPPs, which are distributions over subsets of a finite *ground set* of cardinality M, parametrized by an $M \times M$ kernel matrix $\mathbf{K}$. Routine inference tasks such as normalization, marginalization, or sampling have complexity $\mathcal{O}(M^3)$ (Gillenwater, 2014). Like other kernel methods, when M is large, $\mathcal{O}(M^3)$ is a bottleneck.

In terms of software, the R library spatstat of Baddeley & Turner (2005), a general-purpose toolbox on spatial point processes, includes sampling and learning of continuous DPPs with stationary kernels, as described by Lavancier et al. (2012). Complementarily, we propose DPPy, a turnkey Python implementation of known general algorithms to sample *finite* DPPs. We also include algorithms for non-stationary continuous DPPs, e.g., related to random covariance matrices or Monte Carlo methods, which are also of interest for MLers.

The DPPy project, hosted on GitHub, is already being used by the cross-disciplinary DPP community (Burt et al., 2019; Kammoun, 2018; Poulson, 2019; Dereziński et al., 2019; Gautier et al., 2019). We use Travis for continuous integration and Coveralls for test coverage. Through ReadTheDocs we provide an extensive documentation, which covers the essential mathematical background and showcases the key properties of DPPs through DPPy objects and associated methods. DPPy thus also serves as a tutorial on DPPs.

github.com/guilgautier/DPPy travis-ci.com/guilgautier/DPPy
dppy.readthedocs.io coveralls.io/github/guilgautier/DPPy

©2019 Guillaume Gautier, Guillermo Polito, Rémi Bardenet, and Michal Valko.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-179.html.

2. Definitions

A point process is a random subset of points $\mathcal{X} = \{X_1, \dots, X_N\} \subset \mathbb{X}$, where the number of points N is itself random. We further add to the definition that N should be almost surely finite and that all points in a sample are distinct. Given a reference measure μ on $\mathbb{X}$, a point process is usually characterized by giving its k-correlation functions ρ_k for all k, where

$$\mathbb{P}\left[\begin{array}{c}\exists \text{ one point of the process in} \\ \text{each ball } B(x_i, \mathrm{d}x_i), \forall i = 1, \dots, k\end{array}\right] = \rho_k(x_1, \dots, x_k) \prod_{i=1}^{k} \mu(\mathrm{d}x_i),$$

see Møller & Waagepetersen (2004, Section 4). The functions ρ_k describe the interaction among points in $\mathcal{X}$ by quantifying co-occurrence of points at a set of locations.

A point process $\mathcal{X}$ on $(\mathbb{X}, \mu)$ parametrized by a kernel $K : \mathbb{X} \times \mathbb{X} \to \mathbb{C}$ is said to be *determinantal*, denoted as $\mathcal{X} \sim \mathrm{DPP}(K)$, if its k-correlation functions satisfy

$$\rho_k(x_1, \dots, x_k) = \det\left[K(x_i, x_j)\right]_{i,j=1}^{k}, \quad \forall k \geq 1.$$

In ML, most DPPs are in the finite setting where $\mathbb{X} = \{1, \dots, M\}$ and $\mu = \sum_{i=1}^{M} \delta_i$. In this context, the kernel function becomes an $M \times M$ matrix $\mathbf{K}$, and the correlation functions refer to inclusion probabilities. DPPs are thus often defined as $\mathcal{X} \sim \mathrm{DPP}(\mathbf{K})$ if

$$\mathbb{P}[S \subset \mathcal{X}] = \det \mathbf{K}_S, \quad \forall S \subset \mathbb{X}, \tag{1}$$

where $\mathbf{K}_S$ denotes the submatrix of $\mathbf{K}$ formed by the rows and columns indexed by S. The kernel matrix $\mathbf{K}$ is commonly assumed to be real-symmetric, in which case the existence and uniqueness of the DPP in Equation 1 is equivalent to the condition that the eigenvalues of $\mathbf{K}$ lie in $[0, 1]$. The result also holds for general Hermitian kernel functions K with additional assumptions (Soshnikov, 2000, Theorem 3). We note that there are also DPPs with nonsymmetric kernels (Borodin et al., 2010; Gartrell et al., 2019).

Oftentimes, ML practitioners favor a more flexible definition of a DPP in terms of a *likelihood kernel* $\mathbf{L}$, which only requires $\mathbf{L} \succeq 0$,

$$\mathbb{P}[\mathcal{X} = S] = \frac{\det \mathbf{L}_S}{\det\left[I + \mathbf{L}\right]}. \tag{2}$$

This avoids defining a *correlation kernel* $0 \preceq \mathbf{K} \preceq I$. Yet, the $\mathbf{L}$ parametrization makes Equation 1 less interpretable and does not cover important cases such as fixed-size DPPs, which correspond to projection kernels $\mathbf{K}$. Kulesza & Taskar (2012, Section 5) countered that with k-DPPs, which can be understood as DPPs parametrized by a likelihood kernel, further conditioned to have exactly k elements. However, in general, k-DPPs are not DPPs.

The main interest in DPPs in ML is that they model diversity while being tractable. Compared to independent sampling with the same marginals, Equation 1 entails

$$\mathbb{P}[\{i, j\} \subset \mathcal{X}] = \mathbf{K}_{ii}\mathbf{K}_{jj} - \mathbf{K}_{ij}\mathbf{K}_{ji} = \mathbb{P}[\{i\} \subset \mathcal{X}] \times \mathbb{P}[\{j\} \subset \mathcal{X}] - \mathbf{K}_{ij}^2. \tag{3}$$

In particular, the larger $\mathbf{K}_{ij}$, the less likely items i and j co-occur. If $\mathbf{K}_{ij}$ models the similarity between items i and j, DPPs are thus random *diverse* sets of elements.

Most point processes that encode diversity are not tractable, in the sense that efficient algorithms to sample, marginalize, or compute normalization constants are not available. However, DPPs are amenable to these tasks with polynomial complexity (Gillenwater, 2014). Next, we focus on the sampling task, which is the core of DPPy.

3. Sampling determinantal point processes

We henceforth assume that K is real-symmetric and satisfies suitable conditions (Soshnikov, 2000, Theorem 3), so that its spectral decomposition is available

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi_i(y), \quad \text{with} \int_{\mathbb{X}} \phi_i(x)\phi_j(x)\mu(\mathrm{d}x) = \delta_{ij}.$$

Note that, in the finite case, the spectral theorem is enough to eigendecompose $\mathbf{K}$. Hough et al. (2006, Theorem 7) proved that sampling $\mathrm{DPP}(K)$ can be done in two steps:

1. draw $B_i \sim \mathcal{B}\mathrm{er}(\lambda_i)$ independently and denote $\{i_1, \dots, i_N\} = \{i : B_i = 1\}$,
2. sample from the DPP with kernel $\widetilde{K}(x,y) = \sum_{n=1}^{N} \phi_{i_n}(x)\phi_{i_n}(y)$.

In other words, all DPPs are mixtures of *projection* DPPs, i.e., DPPs parametrized by an orthogonal projection kernel. In a nutshell, Step 1 selects a component of the mixture and Step 2 generates a sample of the projection $\mathrm{DPP}(\widetilde{K})$. Hough et al. (2006, Algorithm 18) provide a generic projection DPP sampler that we briefly describe. First, the projection DPP with kernel $\widetilde{K}$ has exactly $N = \operatorname{rank} \widetilde{K}$ points, μ-almost surely. Then, the sequential aspect of the chain rule applied to sample $(X_1, \dots, X_N)$ with probability distribution

$$\frac{\det\left[\widetilde{K}(x_p, x_n)\right]_{p,n=1}^{N}}{N!} \prod_{n=1}^{N} \mu(\mathrm{d}x_n) = \frac{\|\Phi(x_1)\|^2}{N}\mu(\mathrm{d}x_1) \prod_{n=2}^{N} \frac{\mathrm{distance}^2\left(\Phi(x_n), \mathrm{span}\left\{\Phi(x_p)\right\}_{p=1}^{n-1}\right)}{N-(n-1)}\mu(\mathrm{d}x_n), \tag{4}$$

can be discarded to get a valid sample $\{X_1, \dots, X_N\} \sim \mathrm{DPP}(\widetilde{K})$. To each $x \in \mathbb{X}$ we associate a *feature vector* $\Phi(x) \triangleq (\phi_{i_1}(x), \dots, \phi_{i_N}(x))$, so that $\widetilde{K}(x,y) = \Phi(x)^{\intercal}\Phi(y)$.

A few remarks are in order. First, the LHS of Equation 4 defines an exchangeable probability distribution. Second, the successive ratios that appear in the RHS are the normalized conditional densities (w.r.t. μ) that drive the chain rule. The associated normalizing constants are independent of the previous points. The numerators can be written as the ratio of two determinants and further expanded with Woodbury's formula. They can be identified as the incremental posterior variances in Gaussian process regression with kernel $\widetilde{K}$ (Rasmussen & Williams, 2006, Equation 2.26). Third, the chain rule expressed in Equation 4 has a strong Gram-Schmidt flavor since it actually comes from a recursive application of the base×height formula. In the end, DPPs favor configuration of points whose feature vectors $\Phi(x_1), \dots, \Phi(x_N)$ span a large volume, which is another way of understanding repulsiveness. The previous sampling scheme is exact and generic but, except for projection kernels, it requires the eigendecomposition of the underlying kernel.

In the finite setting, this corresponds to an initial $\mathcal{O}(M^3)$ eigendecomposition cost, followed by an average complexity of order $\mathcal{O}(M\,\mathbb{E}[|\mathcal{X}|]^2)$ (see, e.g., Gillenwater, 2014; Tremblay et al., 2018). Besides, there exist some alternative exact samplers. Poulson (2019) and Launay et al. (2018) use a $\mathcal{O}(M^3)$ Cholesky-based chain rule on sets; each item in turn is decided to be excluded or included in the sample. Dereziński et al. (2019) first sample an intermediate distribution and correct the bias by thinning the intermediate sample (with size smaller than M) using a carefully designed DPP. In certain regimes, this procedure may be more practical with an overall $\mathcal{O}(M\,\mathrm{poly}(\mathbb{E}[|\mathcal{X}|])\,\mathrm{polylog}(M))$ cost. In the continuous case, sampling exactly each conditional that appears in the right-hand side of Equation 4 can be done using rejection sampling with a tailored proposal, see, e.g., Gautier et al. (2019).

In applications where the cost of exact sampling becomes a bottleneck, users rely on approximate sampling. Research has focused mainly on kernel approximation (Affandi et al., 2013) and MCMC samplers (Anari et al., 2016; Li et al., 2016; Gautier et al., 2017).

However, specific DPPs admit efficient exact samplers that do not rely on Equation 4, e.g., uniform spanning trees (UST, Propp & Wilson, 1998, Figure 1(c)) or eigenvalues of random matrices. For instance, a β-ensemble is a set of N points of $\mathbb{R}$ with joint distribution

$$\frac{1}{Z_{N,\beta}} \prod_{p<n} |x_p - x_n|^{\beta} \prod_{n=1}^{N} \omega(x_n)\,\mathrm{d}x_n, \quad \text{where } \beta > 0.$$

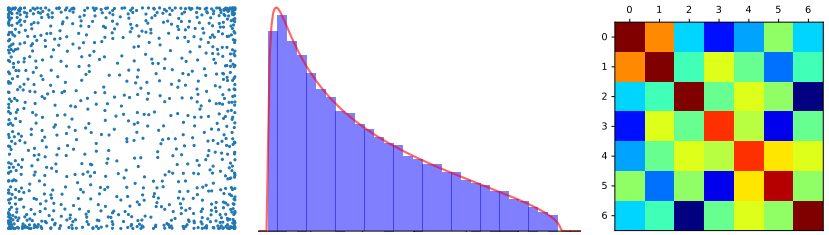
For some choices of the weight function ω, the β-ensemble can be sampled by computing the eigenvalues of simple tridiagonal (Dumitriu & Edelman, 2002) or quindiagonal random matrices (Killip & Nenciu, 2004). In particular, ($\beta = 2$)-ensembles correspond to projection DPPs (König, 2004). They are therefore examples of continuous DPPs that can be sampled exactly in $\mathcal{O}(N^2)$ time, without rejection. Some of these ensembles are of direct interest to MLers. The *Laguerre* ensemble, for instance, has ω be a Gamma pdf, and corresponds to the eigenvalues of the empirical covariance matrix of i.i.d. Gaussian vectors, see Figure 1(b). Finally, we mention that DPPy also features an exact sampler of the multivariate extension of the *Jacobi* ensemble, which has been central in recent results on faster-than-Monte Carlo numerical integration (Bardenet & Hardy, 2019; Gautier et al., 2019; Mazoyer et al., 2019).

4. The DPPy toolbox

DPPy handles Python objects that fit the natural definition of the corresponding DPPs; see also the documentation and the corresponding Jupyter notebooks, which showcase DPPy objects. For example, `FiniteDPP(kernel_type="correlation", **{"K":K})` instantiates a finite $\mathrm{DPP}(\mathbf{K})$. Its two main methods, `.sample_exact()` and `.sample_mcmc()` implement the different exact samplers and current state-of-the-art MCMC samplers. To sample k-DPPs, the additional `.sample_exact_k_dpp()` and `.sample_mcmc_k_dpp()` methods are available.

A Laguerre β-ensemble is instantiated as `LaguerreEnsemble(beta=2)`. When $\beta \in \{1, 2, 4\}$, it can be sampled either with `.sample_full_model()`, as the eigenvalues of a random covariance matrix, or, more efficiently and for any $\beta > 0$, using a tridiagonal matrix with `.sample_banded_model()`. Samples can be displayed via `.plot()` or `.hist()` to construct the empirical distribution that converges to the Marčenko-Pastur distribution, see Figure 1(b).

DPPy can readily serve as research and teaching support. DPPy is also ready for other contributors to add content and enlarge its scope, e.g., with procedures for learning kernels.



(a) 2D Jacobi ensemble (b) $\beta = 2$-Laguerre ensemble (c) $\mathbf{K}$ kernel of UST

Figure 1: Some displays available in DPPy

Acknowledgments

We acknowledge funding by European CHIST-ERA project DELTA, the French Ministry of Higher Education and Research, the Nord-Pas-de-Calais Regional Council, Inria and Otto-von-Guericke-Universität Magdeburg associated-team north-European project Allocate, and French National Research Agency project BoB (n.ANR-16-CE23-0003).

References

Affandi, R. H., Kulesza, A., Fox, E. B., and Taskar, B. Nyström Approximation for Large-Scale Determinantal Processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

Anari, N., Gharan, S. O., and Rezaei, A. Monte Carlo Markov Chain Algorithms for Sampling Strongly Rayleigh Distributions and Determinantal Point Processes. In *Conference on Learning Theory (COLT)*, 2016. arXiv:1602.05242.

Baddeley, A. and Turner, R. spatstat : An R Package for Analyzing Spatial Point Patterns. *Journal of Statistical Software*, 2005.

Bardenet, R. and Hardy, A. Monte Carlo with Determinantal Point Processes. *Annals of Applied Probability, in press*, 2019. arXiv:1605.00361.

Borodin, A., Diaconis, P., and Fulman, J. On adding a list of numbers (and other one-dependent determinantal processes). *Bulletin of the American Mathematical Society*, 2010. arXiv:0904.3740.

Burt, D., Rasmussen, C. E., and Wilk, M. V. D. Rates of Convergence for Sparse Variational Gaussian Process Regression. In *International Conference on Machine Learning (ICML)*, 2019. arXiv:1903.03571.

Dereziński, M., Calandriello, D., and Valko, M. Exact sampling of determinantal point processes with sublinear time preprocessing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. arXiv:1905.13476.

Dumitriu, I. and Edelman, A. Matrix Models for Beta Ensembles. *Journal of Mathematical Physics*, 2002. arXiv:math-ph/0206043.

Dupuy, C. and Bach, F. Learning Determinantal Point Processes in Sublinear Time. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018. arXiv:1610.05925.

Gartrell, M., Paquet, U., and Koenigstein, N. Low-Rank Factorization of Determinantal Point Processes for Recommendation. In *AAAI Conference on Artificial Intelligence*, 2016. arXiv:1602.05436.

Gartrell, M., Brunel, V.-E., Dohmatob, E., and Krichene, S. Learning Nonsymmetric Determinantal Point Processes. *ArXiv e-prints*, 2019. arXiv:1905.12962.

Gautier, G., Bardenet, R., and Valko, M. Zonotope hit-and-run for efficient sampling from projection DPPs. *International Conference on Machine Learning (ICML)*, 2017. arXiv:1705.10498.

Gautier, G., Bardenet, R., and Valko, M. On two ways to use determinantal point processes for Monte Carlo integration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Gillenwater, J. Approximate inference for determinantal point processes. PhD thesis, University of Pennsylvania, 2014.

Hough, J. B., Krishnapur, M., Peres, Y., and Virág, B. Determinantal Processes and Independence. In *Probability Surveys*, 2006. arXiv:math/0503110.

Kammoun, M. S. Monotonous subsequences and the descent process of invariant random permutations. *Electronic Journal of Probability*, 2018. arXiv:1805.05253.

Kathuria, T., Deshpande, A., and Kohli, P. Batched Gaussian Process Bandit Optimization via Determinantal Point Processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. arXiv:1611.04088.

Killip, R. and Nenciu, I. Matrix models for circular ensembles. *International Mathematics Research Notices*, 2004. arXiv:math/0410034.

König, W. Orthogonal polynomial ensembles in probability theory. *Probability Surveys*, 2004. arXiv:math/0403090.

Kulesza, A. and Taskar, B. Determinantal Point Processes for Machine Learning. *Foundations and Trends in Machine Learning*, 2012. arXiv:1207.6083.

Launay, C., Galerne, B., and Desolneux, A. Exact Sampling of Determinantal Point Processes without Eigendecomposition. *ArXiv e-prints*, 2018. arXiv:1802.08429.

Lavancier, F., Møller, J., and Rubak, E. Determinantal point process models and statistical inference : Extended version. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 2012. arXiv:1205.4818.

Li, C., Jegelka, S., and Sra, S. Fast Mixing Markov Chains for Strongly Rayleigh Measures, DPPs, and Constrained Sampling. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. arXiv:1608.01008.

Macchi, O. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 1975.

Mazoyer, A., Coeurjolly, J.-F., and Amblard, P.-O. Projections of determinantal point processes. *ArXiv e-prints*, 2019. arXiv:1901.02099.

Møller, J. and Waagepetersen, R. P. Statistical inference and simulation for spatial point processes. Chapman & Hall/CRC. 2004.

Pathria, R. K. and Beale, P. D. Statistical Mechanics. Academic Press. 2011.

Poulson, J. High-performance sampling of generic Determinantal Point Processes. *ArXiv e-prints*, 2019. arXiv:1905.00165.

Propp, J. G. and Wilson, D. B. How to Get a Perfectly Random Sample from a Generic Markov Chain and Generate a Random Spanning Tree of a Directed Graph. *Journal of Algorithms*, 1998.

Rasmussen, C. E. and Williams, C. K. I. Gaussian processes for machine learning. MIT Press. 2006.

Soshnikov, A. Determinantal random point fields. *Russian Mathematical Surveys*, 2000. arXiv:math/0002099.

Tremblay, N., Barthelme, S., and Amblard, P.-O. Optimized Algorithms to Sample Determinantal Point Processes. *ArXiv e-prints*, 2018. arXiv:1802.08471.

Journal of Machine Learning Research 20 (2019) 1-23 Submitted 3/19; Revised 9/19; Published 11/19

Neural Empirical Bayes

Saeed Saremi SAEED@BERKELEY.EDU
Redwood Center for Theoretical Neuroscience
University of California
Berkeley, CA 94720-3198, USA
NNAISENSE Inc., Austin, TX

Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
University College London, UK
Université Paris-Saclay, Inria, France
University of Helsinki, Finland

Editor: Yoshua Bengio

Abstract

We unify *kernel density estimation* and *empirical Bayes* and address a set of problems in unsupervised machine learning with a geometric interpretation of those methods, rooted in the *concentration of measure* phenomenon. Kernel density is viewed symbolically as $X \rightharpoonup Y$ where the random variable X is smoothed to $Y = X + N(0, \sigma^2 I_d)$, and empirical Bayes is the machinery to denoise in a *least-squares* sense, which we express as $X \leftharpoonup Y$. A learning objective is derived by combining these two, symbolically captured by $X \rightleftharpoons Y$. Crucially, instead of using the original nonparametric estimators, we parametrize *the energy function* with a neural network denoted by ϕ; at optimality, $\nabla\phi \approx -\nabla \log f$ where f is the density of Y. The optimization problem is abstracted as interactions of high-dimensional spheres which emerge due to the concentration of isotropic Gaussians. We introduce two algorithmic frameworks based on this machinery: (i) a "walk-jump" sampling scheme that combines Langevin MCMC (walks) and empirical Bayes (jumps), and (ii) a probabilistic framework for *associative memory*, called NEBULA, defined à la Hopfield by the *gradient flow* of the learned energy to a set of attractors. We finish the paper by reporting the *emergence* of very rich "creative memories" as attractors of NEBULA for highly-overlapping spheres.

Keywords: empirical Bayes, unnormalized densities, concentration of measure, Langevin MCMC, associative memory

1. Introduction

Let $X_1, \ldots, X_n$ be an *i.i.d.* sequence in $\mathbb{R}^d$ from a probability density function f_X. A well-known *nonparametric estimator* of f_X is the *kernel density estimator* $\widehat{f}$ that spreads the *Dirac masses* of the empirical distribution with a (gaussian) kernel f_N (Parzen, 1962). Consider $\widehat{f}$ and its standard definition, but take the kernel bandwidth to be a *hyperparameter*. Now, we can consider $\widehat{f}$ to be an estimator of the smoothed density $f_Y = f_X * f_N$, which is the probability density function of the noisy random variable

$$Y = X + N(0, \sigma^2 I_d).$$

©2019 Saeed Saremi and Aapo Hyvärinen.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-216.html.

This setup is denoted symbolically as $X \rightharpoonup Y$.

Remark 1 (X and Y) *The distinction between X and Y is crucial, and in this work we constantly go "back and forth" between them. Regarding their probability density functions, we adopt a "clean notation": $f(y) = f_Y(y)$ and $f(x) = f_X(x)$, where the subscripts in f_X and f_Y are understood from their arguments x and y. In the absence of arguments, the subscripts are brought back, e.g.*

$$f_Y = f_X * f_N.$$

In addition, as it will become clear shortly, we are only concerned with approximating $\nabla \log f_Y$ in this paper, therefore the subscript Y becomes the default; it is understood that $f = f_Y$.

Next, we discuss the random variable Y in $\mathbb{R}^d$ from the perspective of *concentration of measure*. Assume that the random variable X in high dimensions is concentrated on a low-dimensional manifold $\mathscr{M}$, and denote the manifold where the random variable $Y = X + N(0, \sigma^2 I_d)$ is concentrated as $\mathscr{N}$.[1] We are interested in formalizing the hypothesis that as $d \to \infty$, the convolution of f_X and f_N (for any σ) would "disintegrate" $\mathscr{M}$ such that $\dim(\mathscr{N}) \gg \dim(\mathscr{M})$. We study an example of a "gaussian manifold", defined by $X \sim N(0, \Sigma_\sharp)$ where

$$\Sigma_\sharp = \begin{bmatrix} I_{d_\sharp} & 0 \\ 0 & \epsilon^2 I_{d-d_\sharp} \end{bmatrix},$$

with $\epsilon \ll 1$ and $d_\sharp \ll d$. We revisit textbook calculations on concentration of measure and show that in high dimensions, the smoothed manifold $\mathscr{N}$ is approximately a sphere of dimension $d-1$:

$$\mathscr{N} \approx \sqrt{(\epsilon^2 + \sigma^2)(d - d_\sharp)}\, S^{d-1}.$$

This analysis paints a picture of "manifold disintegration-expansion" as a general phenomenon.

(C1) The first conceptual contribution of this paper is to expose a notion of manifold disintegration-expansion, with a general takeaway that in (very) high dimensions, gaussian smoothing pushes away (all) the probability masses near $\mathscr{M}$. This is due to the fact that in high dimensions, the Gaussian random variable

$$N(0, \sigma^2 I_d) \approx \mathrm{Unif}(\sigma\sqrt{d} S^{d-1})$$

is not necessarily concentrated on $\mathscr{M}$, and there are many directions (asymptotically infinite) where samples from Y "escape" the manifold. The thesis is that, in convolving f_X and f_N, $\mathscr{M}$ would be mapped to a (much) higher dimensional manifold.

A seemingly unrelated theory is *empirical Bayes*, as formulated by Robbins (1956), which we use for pulling the probability mass back towards $\mathscr{M}$. In 1956, Robbins considered a scenario of a random variable Y that depends "in a known way" on an "unknown" random variable

1. Unfortunately, the concept of "manifold" for a probability distribution, let alone its dimension, is quite a fuzzy concept. But it should also be stated that this is a very important problem; the *assumption* on the *existence* of $\mathscr{M}$ is a pillar of machine learning (Saul and Roweis, 2003; Bengio et al., 2013a).

X.[2] He found that, given an observation $Y = y$, the *least squares estimator* of X is the *Bayes estimator*, and quite remarkably, can be expressed *purely* in terms of the distribution of Y (he showed this for Poisson, geometric, and Laplacian kernels). Robbins' results were later extended to *gaussian kernels* by (Miyasawa, 1961) who derived the estimator

$$\widehat{x}(y) = y + \sigma^2 \nabla \log f(y).$$

The least-squares estimator of X above is derived for *any* σ, not necessarily infinitesimal. To signify this important fact, we also refer to this estimation as "Robbins jump" or *jump* for short. The empirical Bayes machinery is denoted symbolically as

$$X \leftharpoonup Y.$$

The smoothing of kernel density estimation, $X \rightharpoonup Y$, and the denoising mechanism of empirical Bayes, $X \leftharpoonup Y$, come together to define the learning objective

$$\mathcal{L} = \mathbb{E}\, \|X - \widehat{x}(Y)\|^2.$$

(The squared ℓ_2 norm $\|\cdot\|^2$ in the definition of $\mathcal{L}$ is due to the fact that the estimator of X is a *least-squares estimator*.) In this setup, we take the two *nonparametric estimators* that defined $\mathcal{L}$ and parametrize the *energy function* of the random variable Y with a neural network $\phi : \mathbb{R}^d \to \mathbb{R}$ with parameters θ. It should be stated that ϕ is defined modulo an additive constant. The learning objective, expressed in terms of θ, is therefore given by

$$\mathcal{L}(\theta) = \mathbb{E}\, \|X - Y + \sigma^2 \nabla \phi(Y, \theta)\|^2. \tag{1}$$

(Throughout the paper, ∇ is the gradient taken with respect to the inputs y, not parameters.)

(C2) The second conceptual contribution of this paper is this unification of kernel density estimation and empirical Bayes. The unification, denoted symbolically as $X \rightleftharpoons Y$, is encapsulated in the expression for the learning objective above, $\mathcal{L} = \mathbb{E}\, \|X - \widehat{x}(Y)\|^2$. We thus combine two principles of nonparametric estimation into a single learning objective. In optimizing the objective, we choose to parametrize the energy function with a (overparametrized) neural network. The growing understanding on the representational power of deep (and wide) neural networks and the effectiveness of SGD for the "problem of learning" (Vapnik, 1995) is behind this choice.

Remark 2 (DEEN) *For gaussian noise, the objective $\mathcal{L} = \mathbb{E}\, \|X - \widehat{x}(Y)\|^2$ is the same as the learning objective in "deep energy estimator networks" (DEEN) (Saremi et al., 2018) which itself was based on denoising score matching (Hyvärinen, 2005; Vincent, 2011). However, this equivalence breaks down beyond gaussian kernels—see (Raphan and Simoncelli, 2011) for a comprehensive survey of empirical Bayes least squares estimators. From this angle, empirical Bayes appears as a more fundamental framework to formulate the problem of unnormalized density estimation for noisy random variables.*

2. The starting point in (Robbins, 1956) was the existence of a noisy random variable that was denoted by X. In that setup, one can *only* observe values of X. Here, we started with the *"clean"* i.i.d. sequence $X_1, \ldots, X_n$ and *artifically created* $Y = X + N(0, \sigma^2 I_d)$. This departure in starting points is related to our new take on empirical Bayes which will become clear in Section 3.

When analyzing a finite i.i.d. sequence, $X_1, \ldots, X_n$, i.i.d. samples from $Y = X + N(0, \sigma^2 I_d)$ are generated as

$$Y_{ij} = X_i + \varepsilon_j,\ \varepsilon_j \sim N(0, \sigma^2 I_d),$$

and the learning objective $\mathcal{L}(\theta)$ is then approximated as

$$\mathcal{L}(\theta) \approx \sum \left\| X_i - Y_{ij} + \sigma^2 \nabla \phi(Y_{ij}, \theta) \right\|^2,$$

where $\sum$ is a shorthand for $\frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m$. In high dimensions, the samples Y_{ij} are (approximately) uniformly distributed on a "thin spherical shell" around the sphere $\sigma\sqrt{d} S_i^{d-1}$ with its center at X_i which leads to the following definition.

Definition 3 (i-sphere) *The sphere $\sigma\sqrt{d} S_i^{d-1}$ centered at X_i is a useful abstraction and it is referred to as "i-sphere". The i in i-sphere refers to the index i in X_i, and the index j is reserved for the gaussian noise ε_j. Therefore, Y_{ij} is the jth sample on the i-sphere. Fixing the index i, the samples Y_{ij} are approximately uniformly distributed on a thin spherical shell around the i-sphere (see Figure 1).*

In fact, the learning based on $\mathcal{L}$ can be interpreted as shaping the global energy function such that locally, for samples Y_{ij} from a single i-sphere, the denoised versions $\widehat{x}(Y_{ij})$ come as close as possible (in squared ℓ_2 norm) to X_i at the center (see Figure 1c). This "shaping of the energy function" is programmed in a simple algorithm,

DEEN: minimize $\mathcal{L}(\theta)$ with stochastic gradient descent and return θ^*.

Remark 4 *In most of the paper, we are interested in studying ϕ which is already at optimality with parameteres $\theta^* = \operatorname{argmin} \mathcal{L}(\theta)$. In these contexts, θ^* is dropped, and its presence is understood, e.g. at optimality, $\phi \approx -\log f$ (modulo a constant).*

Next, we define a notion of interactions between i-spheres that we find useful in thinking about the goal of approximating the *score function* through optimizing $\mathcal{L}(\theta)$. However, this can be skipped until our discussions of the associative memory in Sections 6 and 7.

Definition 5 (i-sphere interactions) *The interaction between i-spheres is defined as the competition that emerge due to the presence of $\nabla\phi$ in the problem of minimizing $\mathcal{L}$. In other words, it refers to "the set of constraints" that $\nabla\phi$, the gradient of the globally defined energy function, has to satisfy locally on i-spheres to arrive at the optimality such that $\nabla\phi \approx -\nabla \log f$. It is indeed an abstract notion in this work, but it is a useful abstraction to have which we come back to in thinking about the gradient flow $y' = -\nabla\phi(y)$. It should be stated that there is also a notion of interaction that already exists for the kernel density in the sense of "mass aggregation" (see Section 7 for references), but in this framework, i-spheres interact even when they do not overlap in that they "communicate" via $\nabla\phi$.*

(C3) The third conceptual contribution of this paper is introducing the physical picture of interactions between *i.i.d.* samples. This is a useful abstraction in this work in thinking about the problem of approximating the *score function* with the gradient of an *energy function*. The interaction is a code for "the set of constraints" that $\nabla\phi$ (evaluated at i.i.d. samples Y_{ij}) has to satisfy to arrive at optimality, $\nabla\phi \approx -\nabla \log f$. These interactions could also give rise to some *collective phenomena* (Anderson, 1972).

Next, we outline the technical contributions of this paper. Approximating $\nabla \log f$ in the empirical Bayes setup leads to a novel sampling algorithm, a new notion of associative memory and the emergence of "creative memories". First, we define the matrix χ that was introduced to quantify i-sphere overlaps and was used for designing experiments.

(i) Concentration of measure leads to a geometric picture for the kernel density as a *"mixture of i-spheres"*, and we define the matrix χ to quantify the extent to which the spheres in the mixture overlap; the entries in χ are essentially pairwise distances, scaled by $2\sqrt{d}$ (where the scaling is related to the concentration of isotropic Gaussians in high dimensions):

$$\chi_{ii'} = \frac{\|X_i - X_{i'}\|}{2\sqrt{d}}. \tag{2}$$

The i-sphere and the i'-sphere do not overlap if $\sigma < \chi_{ii'}$, and they overlap if $\sigma > \chi_{ii'}$ (see Figure 2). Of special interest is σ_c, defined as

$$\sigma_c = \max_{ii'} \chi_{ii'}.$$

We define "extreme noise" as the regime $\sigma > \sigma_c$; it is the regime where all i-spheres have some degree of overlap.

(ii) "Walk-jump sampling" is an approximative sampling algorithm that first draws exact samples from the density $\exp(-\phi)/Z$ (Z is the unknown normalizing constant) by *Langevin MCMC*:

$$y_{t+1} = y_t - \delta^2 \nabla \phi(y_t) + \sqrt{2}\delta\varepsilon, \quad \varepsilon \sim N(0, I_d),$$

where $\delta \ll 1$ is the step size and here t is discrete time. At an arbitrary time τ, approximative samples "close" to $\mathscr{M}$ are generated with the least-squares estimator of X—*the jumps*:

$$\widehat{x}(y_\tau) = y_\tau - \sigma^2 \nabla \phi(y_\tau).$$

A signature of walk-jump sampling is that the walks and the jumps are decoupled in that the jumps can be made at arbitrary times.

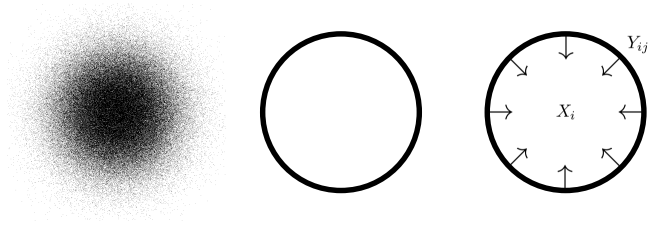
(iii) "Neural empirical Bayes associative memory"—named NEBULA, where the "LA" refers to *à la* Hopfield (1982)—is defined as the flow to strict local minima of the energy function ϕ. In continuous time, the memory dynamics is governed by the *gradient flow*:

$$y'(t) = -\nabla \phi(y(t)).$$

In retrieving a "memory", one flows deterministically to an *attractor*. NEBULA is an intriguing construct as the deterministic dynamics is governed by a *probability density function*, since $\nabla \phi \approx -\nabla \log f$—this is very far from true for Hopfield networks.

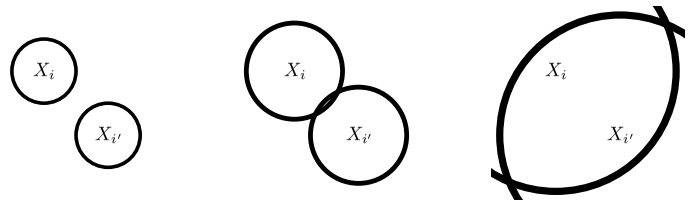
(iv) Memories in NEBULA are believed to be formed due to *i-sphere interactions* (Definition 3). We provide evidence for the presence of this abstract notion of interaction in this problem by reporting the emergence of very structured memories in a regime of highly overlapping i-spheres. They are named "creative memories" because they have intuitively appealing structures, while clearly being new instances, in fact quite different from the training data.

This paper is organized as follows. *Manifold disintegration-expansion* is discussed in Section 2. In Section 3, we review empirical Bayes and give a derivation of the least squares estimator of X for gaussian kernels. We then talk about a *Gedankenexperiment*—in the school of Robbins (1956) and Robbins and Monro (1951)—to bring home the *unification scheme* $X \rightleftharpoons Y$ and shed some light on the inner-working of the learning objective. In Section 4, we define the notion of *extreme noise* and demonstrate two extreme denoising experiments. In Section 5, we present the *walk-jump sampling* algorithm. *NEBULA* is defined in Section 6, where the abstract notion of *i-sphere interactions* is grounded to some extent; we present two sets of experiments with qualitatively different behaviors. In Section 7, we report the emergence of *creative memories* for two values of σ close to σ_c. We finish with a summary.



(a) $d = 2$ gaussian (b) $d \gg 1$ gaussian (c) $X \rightleftharpoons Y$

Figure 1: (a) Samples from a 2D isotropic gaussian, obtained and rendered in the programming language Processing. (b) Schematic of an isotropic gaussian in high dimensions, where the concentration of norm is illustarted. (c) Schematic of the i-sphere, with samples $Y_{ij} = X_i + \varepsilon_j$, $\varepsilon_j \sim N(0, \sigma^2 I_d)$. The arrows represent $-\nabla\phi$, evaluated on the sphere. The learning objective is encapsulated by $X \rightleftharpoons Y$, where the squared ℓ_2 norm $\|X_i - \widehat{x}(Y_{ij})\|^2$ is the learning signal and minimized in expectation. Ignoring the other spheres, the learning objective is constructed such that $-\nabla\phi$ evaluated at Y_{ij} points to X_i.



(a) $\sigma < \chi_{ii'}$ (b) $\sigma \approx \chi_{ii'}$ (c) $\sigma > 2\chi_{ii'}$

Figure 2: (*overlapping i-spheres*) The extent of the overlap between i-sphere and i'-sphere is tuned by σ in relation to $\chi_{ii'} = \|X_i - X_{i'}\|/(2\sqrt{d})$. The scaling $(2\sqrt{d})$ is due to the fact that $N(0, \sigma^2 I_d) \approx \mathrm{Unif}(\sigma\sqrt{d}S^{d-1})$ in high dimensions.

2. Manifold disintegration-expansion in high dimensions

Our low-dimensional intuitions break down spectacularly in high dimensions. This is in large part due to the exponential growth of space in high dimensions (this abundance of space also underlies the *curse of dimensionality*, well known in statistics and machine learning). Our focus in this section is to develop some intuitions on the effect of gaussian smoothing, $X \rightharpoonup Y$, on the data manifold $\mathscr{M}$. We start by a summary of results on *concentration of measure*. The textbooks (Ledoux, 2001; Tao, 2012; Vershynin, 2018) should be consulted to fill in details. We then extend the textbook calculations for a "gaussian manifold" and make a case for *manifold disintegration-expansion*.

Start with the isotropic gaussian. In 2D, samples from the gaussian form a "cloud of points", centered around its mode as illustrated in Figure 1a. Next, take the isotropic Gaussian in d dimensions, $(X_1, \ldots, X_d) = N(0, I_d)$, and ask where the random variable is likely to be located. Before stating a well-known result on the concentration of measure, we can build intuition by observing the following identities for the expectation and the variance of the squared norm:

$$\begin{aligned} \mathbb{E}\,\|X\|^2 &= \sum_{i=1}^{d} \mathbb{E}\,X_i^2 = d, \\ \mathbb{V}\|X\|^2 &= \sum_{i=1}^{d} \mathbb{E}(X_i^2 - 1)^2 = 2d. \end{aligned}$$

Since the components of $X = (X_1, \ldots, X_d)$ are jointly independent, and since $d \gg 1$, the following holds with high probability,

$$\begin{aligned} \|X\|^2 &= d \pm O(\sqrt{d}), \\ \|X\| &= \sqrt{d} \pm O(1). \end{aligned}$$

The concentration of norm is visualized in Figure 1b; also see Figure 3.2 in (Vershynin, 2018). In contradiction to our low-dimensional intuition, in high dimensions, the Gaussian random variable $N(0, I_d)$ is not concentrated close to the mode of its density f_N (at the origin here), but in a "thin spherical shell" of width $O(1)$ around the sphere of radius $\sqrt{d}$. The concentration of norm suggests an even deeper phenomenon, the concentration of measure, captured by a non-asymptotic result for the deviation of the norm $\|X\|$ from $\sqrt{d}$, where the deviation has a *sub-gaussian* tail (Tao, 2012; Vershynin, 2018):

$$\mathbb{P}\left(\left|\|X\| - \sqrt{d}\right| \geq t\right) \leq 2\exp(-ct^2), \quad \text{for all } t \geq 0.$$

A related result states that

$$\mathbb{P}(\|X\| \leq \epsilon\sqrt{d}) \leq (C\epsilon)^d, \quad \text{for all } \epsilon \geq 0.$$

In above expressions, C and c are absolute constants. In high dimensions, the Gaussian random variable is thus approximated with the uniform distribution on the sphere $\sigma\sqrt{d}S^{d-1}$,

$$N(0, \sigma^2 I_d) \approx \mathrm{Unif}(\sigma\sqrt{d}S^{d-1}).$$

The approximation becomes equality in distribution, as $d \to \infty$.

Remark 6 *In high dimensions, our intuitions for the uniform distribution itself breaks down, where the probability mass no longer has a "uniform geometry". For example,* $\mathrm{Unif}([0,1]^d)$ *is concentrated near the hyperplane* $X_1 + \cdots + X_d = d/2$*, which is easy to see, starting from* $\mathbb{E}\,X = 1/2$ *for the univariate* X *(Ledoux, 2001). The same goes for* $\mathrm{Unif}(\sqrt{d}S^{d-1})$*, which is a prime example of "concentration without independence", with counterintuitive results like the "blow-up" phenomenon (Vershynin, 2018).*

We finish this section with a new analysis. Consider $X \sim N(0, \Sigma_\sharp)$ in $\mathbb{R}^d$. A simple low-dimensional "manifold structure" with the dimension $d_\sharp \ll d$ is imposed by considering

$$\Sigma_\sharp = \begin{bmatrix} I_{d_\sharp} & 0 \\ 0 & \epsilon^2 I_{d-d_\sharp} \end{bmatrix},$$

where $\epsilon \ll 1$. The manifold $\mathscr{M}$ is viewed as a "gaussian manifold" where, in an abuse of notation, $\dim(\mathscr{M}) \approx d_\sharp$. Now consider $Y = X + N(0, \sigma^2 I_d)$ which means

$$Y = N(0, \Sigma_\sharp + \sigma^2 I_d).$$

We repeat the concentration of norm calculations:

$$\begin{aligned} \mathbb{E}\,\|Y\|^2 &= d_\sharp(1+\sigma^2) + (d - d_\sharp)(\epsilon^2 + \sigma^2), \\ \mathbb{V}\|Y\|^2 &= 2d_\sharp(1+\sigma^2)^2 + 2(d - d_\sharp)(\epsilon^2 + \sigma^2)^2. \end{aligned}$$

In high dimensions, assuming

$$(d/d_\sharp - 1) \gg \max(\Delta, \Delta^2), \;\; \Delta = (1+\sigma^2)/(\epsilon^2 + \sigma^2),$$

the "manifold terms", $d_\sharp(1+\sigma^2)$ and $d_\sharp(1+\sigma^2)^2$, become negligible, and the following holds with high probability:

$$\begin{aligned} \|Y\|^2 &= (\epsilon^2 + \sigma^2)(d - d_\sharp \pm O(\sqrt{d - d_\sharp})), \\ \|Y\| &= \sqrt{\epsilon^2 + \sigma^2}(\sqrt{d - d_\sharp} \pm O(1)). \end{aligned}$$

The calculations in the previous page are reproduced by setting $\sigma = 1$, $\epsilon = 0$, $d_\sharp = 0$.

In summary, under the convolution $f_Y = f_X * f_N$, the "gaussian manifold" $\mathscr{M}$ is transformed/mapped to

$$\mathscr{N} \approx \sqrt{(\epsilon^2 + \sigma^2)(d - d_\sharp)}S^{d-1},$$

therefore $\dim(\mathscr{N}) \approx d-1$. In our terminology, $\mathscr{M}$ has been *disintegrated-expanded* to $\mathscr{N}$. We expect this phenomenon to hold for any $\mathscr{M}$ asymptotically, i.e. as $d \to \infty$, $\mathscr{N} = \sigma\sqrt{d}S^{d-1}$, but (unfortunately) the limit $d \to \infty$ is not practical. However, the picture should stay: in high dimensions, the convolution of f_X and f_N has severe side effects on the manifold $\mathscr{M}$ itself. Smoothing is indeed desired, but also a mechanism to restore $\mathscr{M}$. In the next section, we discuss such a mechanism using *empirical Bayes*, where the "restoration" is in *least squares*.

3. Neural empirical Bayes

In a seminal work from 1956, titled *an empirical Bayes approach to statistics*, Robbins considered an intriguing scenario of a random variable Y having a probability distribution that depends "in a known way" on an "unknown" random variable X (Robbins, 1956). Observing $Y = y$, the *least-squares estimator* of X is the *Bayes estimator* (see Remark 1):

$$\widehat{x}(y) = \frac{\int x f(y|x) f(x) dx}{\int f(y|x) f(x) dx} = \int x f(x|y) dx = \mathbb{E}(X|Y = y).$$

If f_X is known to "the experimenter", $\widehat{x}$ is a computable function, but what if the prior f_X is unknown? It is quite remarkable that for a large class of kernels, the least-squares estimator can be derived in closed form purely in terms of the distribution of Y. Informally speaking, there is an "abstraction barrier" (Abelson and Sussman, 1985) between X and Y where the knowledge of f_X is not needed to estimate X. This *functional* dependence of $\widehat{x}$ on f_Y was achieved for the Poisson, geometric, and Laplacian kernels in (Robbins, 1956), and it was later extended to the gaussian kernel by (Miyasawa, 1961).

This work builds on Miyasawa's result, and we repeat the calculation here in our notations. Take X to be a random variable in $\mathbb{R}^d$ and Y a noisy observation of X with a known gaussian kernel with symmetric positive-definite $\Sigma \succ 0$:

$$f(y|x) = \frac{1}{(2\pi)^{d/2} |\det(\Sigma)|^{1/2}} \exp\left(-(y-x)^\top \Sigma^{-1} (y-x)/2\right).$$

It follows,

$$\Sigma \ \nabla_y f(y|x) = f(y|x)(x - y).$$

Multiply the expression above by $f(x)$ and integrate,

$$\Sigma \ \nabla f(y) = \int (x - y) f(y|x) f(x) dx = f(y)(\widehat{x}(y) - y),$$

which then leads to the expression

$$\widehat{x}(y) = y + \Sigma \ \nabla \log f(y).$$

For the isotropic case $\Sigma = \sigma^2 I_d$, the estimator takes the form

$$\widehat{x}(y) = y + \sigma^2 \nabla \log f(y). \tag{3}$$

To sum up, the estimator above is obtained in a setup where only the corrupted data (the random variable Y) are observed, with the knowledge of the measurement (gaussian) kernel *alone* and without *any knowledge* of the prior f_X. The remarkable result is the fact the least-squares estimator of X is written in closed form purely as a functional of $\nabla \log f$, also known as the *score function* (Hyvärinen, 2005). The expression above for the *least squares* estimator of X is the basis for an algorithm to approximate $\nabla \log f$ which is discussed next.

In *empirical Bayes*, the random variable X is estimated in least squares from the noisy observations Y, without *any* knowledge of the *prior* f_X. But how did we get here? We started the paper with the i.i.d. sequence $X_1, \ldots, X_n$, where Y did not even exist!

The idea is a Gedankenexperiment of sort, where we first construct $Y = X + N(0, \sigma^2 I_d)$ by taking samples $Y_{ij} \sim f_X * f_N$; in turn, the experimenter (in the school of empirical Bayes) "observes" Y_{ij} and estimates X. But in this artificially supervised setup (the "target" is X_i), the error signal $\|X_i - \widehat{x}(Y_{ij})\|^2$ can also be measured—this is not the case in (Robbins, 1956)—and it drives the learning. The learning is achieved with stochastic gradient descent as the experimenter has parameterized the energy function with a neural network and can compute the stochastic gradients (Robbins and Monro, 1951),

$$\nabla_\theta \|X_i - Y_{ij} + \nabla_y \phi(Y_{ij}, \theta)\|^2.$$

Approximating $\nabla \log f$ in this manner is the essence of neural empirical Bayes.

For *finite* samples, the best we can do is to *approximate* $\nabla \log f$. Given $X_1, \ldots, X_n$, the i.i.d. samples $Y_{ij} \sim f_X * f_N$ are given by $Y_{ij} = X_i + \varepsilon_j$, $\varepsilon_j \sim N(0, \sigma^2 I_d)$. These are indeed i.i.d. samples from density estimated by the kernel density estimator

$$\widehat{f_Y}(y) = \frac{1}{n} \sum_{i=1}^{n} f_N(y - X_i),$$

where the subscript in f_N is a short for $N(0, \sigma^2 I_d)$. Here, σ is a *hyperparameter* and the kernel estimator above is considered an estimator of f_Y. Note that the *optimal bandwidth* in estimating f_X depends on n (van der Vaart, 2000).

The problem is that, in high dimensions, the kernel density estimator $\widehat{f_Y}$ (or any other *nonparametric estimator*) suffers from a severe *curse of dimensionality* (Wainwright, 2019). This leads naturally to *deep neural architectures* which has been viewed as an alternative for "breaking the curse of dimensionality" (Bengio, 2009). In this work, the *energy function* is parametrized with a deep neural network $\phi : \mathbb{R}^d \to \mathbb{R}$ with parameters θ. The objective $\mathcal{L}(\theta)$ is then approximated,

$$\mathcal{L}(\theta) \approx \sum \left\| X_i - Y_{ij} + \sigma^2 \nabla \phi(Y_{ij}, \theta) \right\|^2,$$

At optimality (achieved with *stochastic gradient descent*), $\theta^* = \operatorname{argmin} \mathcal{L}(\theta)$,

$$-\nabla \phi(\cdot, \theta^*) \approx \nabla \log f(\cdot).$$

Remark 7 (implicit vs. explicit parameterization) *As it is made clear, our goal is to approximate $\nabla \log f : \mathbb{R}^d \to \mathbb{R}^d$. In doing so, another choice could have been the explicit parametrization of the score function as $\psi : \mathbb{R}^d \to \mathbb{R}^d$, where one avoids the computation $\nabla \phi$ in the optimization, but there, $\partial_j \psi_i = \partial_i \psi_j$ must hold pointwise. The parametrization of the energy function as $\phi : \mathbb{R}^d \to \mathbb{R}$ is essentially an implicit parametrization of $\nabla \log f$, since the computation $\nabla \phi$ is not a symbolic one, but achieved with automatic differentiation (Baydin et al., 2018). Also, the equivalence between the two strategies for one-hidden-layer networks (Vincent, 2011) breaks down for two hidden layers and beyond. See (Saremi, 2019) for a proof and further discussions.*

	min	median	mean	max $\approx \sigma_c$
χ	0.0186	0.1822	0.1812	0.2884

Table 1: Some statistics of the matrix χ (see Equation 2) estimated from 10^7 pairs from the *handwritten digit database* (LeCun et al., 1998). The σ in each experiment must be viewed in relation to these numbers and our review of the *concentration of measure* in Section 2 (see Figure 2).

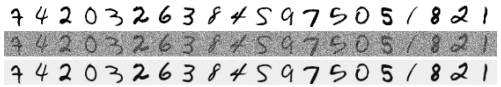


Figure 3: ($\sigma \approx \sigma_c$) Denoising performance of DEEN with a single jump for $\sigma = 0.3$. The noisy pixel values are in the range $[-1.200, 1.995]$, and the denoised ones are in $[-0.0749, 1.0539]$.

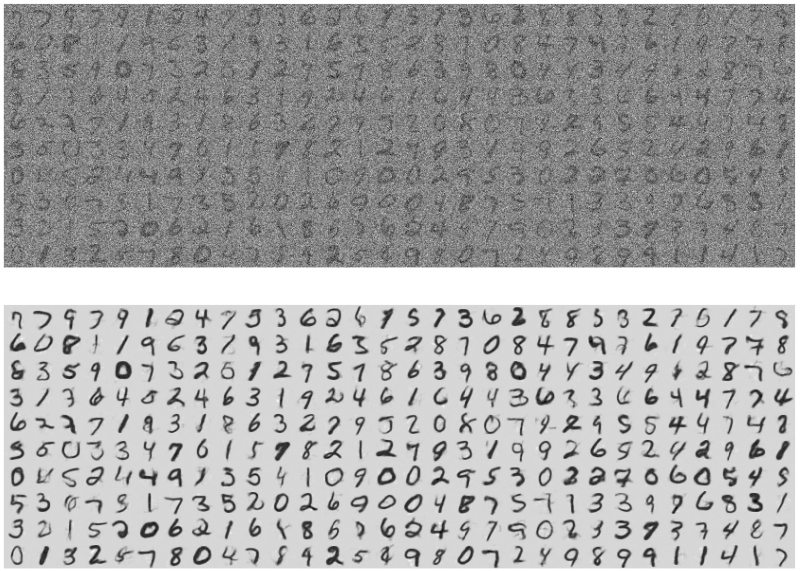


Figure 4: ($\sigma > 2\sigma_c$) Here, $\sigma = 0.7$. The noisy pixel values are in the range $[-3.314, 3.683]$, and the denoised ones are in $[-0.2405, 1.2686]$. In this regime, the whole database is inside each i-sphere (see Figure 2c).

4. Extreme noise/denoising

In empirical Bayes, the least-squares estimator of X,

$$\widehat{x}(y) = y + \sigma^2 \nabla \log f(y),$$

is derived for any σ, which inspires us to call this a "jump" to capture the fact σ may in fact be "large". Denoising is not our actual goal, but the expression above can be viewed as "denoising y to $\widehat{x}(y)$". The denoising performance is important since in the machinery of neural empirical Bayes, the goal of approximating the score function is formulated by the "denoising objective"

$$\mathcal{L}(\theta) = \mathbb{E}\, \|X - Y + \sigma^2 \nabla \phi(Y, \theta)\|^2.$$

As the least-squares estimator of X is derived for any σ, the denoising performance of DEEN may also be tested to its limits. Here, we report such experiments for MNIST, where $d = 784$ and X is in the hypercube $[0, 1]^d$ (see Remark 8). Next, we elaborate on the geometric meaning of the noise levels, where in addition we define a notion of "extreme noise".

- ($\sigma > \sigma_c$). The value $\sigma_c = \max_{ii'} \chi_{ii'}$ is defined as the onset of *extreme noise*. Called "extreme", because for $\sigma > \sigma_c$, all i-spheres in the dataset overlap (see Figure 2b). For MNIST, $\sigma_c \approx 0.2884$ was estimated from 10^7 handwritten digit pairs (see Table 1). The results for $\sigma = 0.3$ on the *test set* is presented in Figure 3. It should be stated that, this value of "extreme noise", just above σ_c, is not visually *perceived* as extreme.
- ($\sigma > 2\sigma_c$). In this regime, due to geometry, the whole dataset is in *inside* of each i-sphere (see Figure 2c). This is very extreme! For MNIST, we experimented with $\sigma = 0.7$ which is in this regime. The results on the *test set* are presented in Figure 4.

Remark 8 *Note that the least-squares estimator of X can take values anywhere in $\mathbb{R}^d$, not restricted to be in $[0, 1]^d$. This is the reason for the "gray background" in the experiments, in relation to which, in the figure captions, we report the min/max of all the pixel values. This also holds for the jump samples presented in the next section.*

4.1. The network architecture

The following comments are in order regarding the *network architecture* we used in our experiments.

- (*activation function*) The denoising results are a significant improvement over (Saremi et al., 2018). This was due to the use of ConvNets, instead of a fully connected network (more on that below) and the use of a *"smooth ReLU"* activation function

  $$\sigma(z, \beta) = z/(1 + \exp(-\beta z)),$$

  where the default was $\beta = 1$. The activation function above converges uniformly to ReLU in the limit $\beta \to \infty$. It has been studied independently from different angles (Elfwing et al., 2018; Ramachandran et al., 2017). ReLU consistently gave a higher loss, which we believe is due to the term $\nabla_y \phi$ in the learning objective, which must be computed first before computing $\nabla_\theta \mathcal{L}$ to update the parameters $\theta \leftarrow \theta - \epsilon \nabla_\theta \mathcal{L}$.

- *(wide convnets)* All experiments reported in the paper were performed in a *fixed* wide ConvNet architecture with the expanding channels $= (256, 512, 1024)$, *without pooling*, and with a *bottleneck layer* of size 10. All hidden layers were activated with the activation function $\sigma(\cdot, \beta = 1)$, and the readout layer was linear.
- *(readout overparametrization)* We observed a slightly faster convergence to lower loss (especially, very early in the training) by *overparameterizing* the linear output layer. These experiments were inspired by the "acceleration effects" studied for *linear neural networks* (Arora et al., 2018), but we did not study it thoroughly.
- We used the *Adam optimizer* (Kingma and Ba, 2014). The optimization was stable over a wide range of *mini-batch sizes* and *learning rates*, and as expected, we did not observe the validation/test loss going up in the experiments. This stability is due to the fact that the inputs to ϕ are noisy samples. The *automatic differentiation* (Baydin et al., 2018) was implemented in PyTorch (Paszke et al., 2017). DEEN is "memory hungry" but the computational costs are not addressed here in detail.

5. Walk-jump sampling: Langevin walk, Robbins jump

Sampling "complex distributions" in high dimensions is an intractable problem due to the fact that their probability mass is concentrated in *sharp ridges* that are separated by *large regions of low probability*, therefore making MCMC methods ineffective *(the "low-dimensional manifold" $\mathscr{M}$ is in fact quite complex when viewed in the ambient space $\mathbb{R}^d$)*. These problems are well known, but empirically they have mostly been studied for sampling f_X. Regarding the random variable Y, it is clear that the problem of sampling $f = f_X * f_N$ is in fact easier. The idea is that MCMC mixes faster as $\mathscr{N}$ is less complex. The "sharp ridges" are smoothed out and (as we argued in Section 2) $\mathscr{M}$ itself is expanded in dimensions, $\dim(\mathscr{N}) \gg \dim(\mathscr{M})$, therefore the "large regions of low probability" themselves are smaller in size. In summary, *Langevin MCMC* is believed to be more effective in sampling

$$\exp(-\phi)/Z \approx f$$

than f_X (putting aside the harder problem of learning $\nabla \log f_X$). At any time, the estimator $\widehat{x}$ can be used to *jump* near $\mathscr{M}$. "Near" is intuitive, and it should be stated that neither $f_{X|Y}$ nor f_X is exactly sampled during jumps; the jump samples must be seen as heuristic.

In what follows, it is understood that ϕ is at optimality. We first give a description of the algorithm and then express the equations.

- *(walks)* The Langevin MCMC is used to draw statistical samples $y_t \sim \exp(-\phi)/Z$. Langevin MCMC is based on discretizing the *Langevin diffusion* (van Kampen, 1992) and the updates are based solely on $\nabla\phi$ (the equations are coming next), therefore not knowing the *normalizing constant* Z is of no concern.
- *(jumps)* Given y_τ, ideally an *exact sample* (MacKay, 2003) from $\exp(-\phi)/Z$, the jumps are made with the *Bayes estimator* of X. The *spirit of empirical Bayes* is fully present here: the jump samples are generated without knowing ∇f_X (or its approximation), and without running Langevin MCMC on its "rougher terrain".

What emerges is a sampling algorithm in two parts. The Langevin MCMC is the *engine* that should run continously. By contrast, the jumps can be made at any time. That is:

(i) Sample from $\exp(-\phi)/Z$ with Langevin MCMC,

$$y_{t+1} = y_t - \delta^2 \nabla\phi(y_t) + \sqrt{2}\delta\varepsilon, \ \varepsilon \sim N(0, I_d),$$

where $\delta \ll 1$ is the step size and here t is discrete time.

(ii) At any time τ, use the Bayes estimator of X to jump from y_τ to $\widehat{x}(y_\tau)$,

$$\widehat{x}(y_\tau) = y_\tau - \sigma^2 \nabla\phi(y_\tau).$$

Two types of approximations have been made in walk-jump sampling. The first is that the Langevin MCMC does not sample f but the density $\exp(-\phi)/Z \approx f$; this approximation is unavoidable. The approximation involved in jumps are less understood. Stepping back, $\widehat{x}(y_\tau)$ *can* be thought of as approximating the *posterior* $f_{X|y_\tau}$ with a single Dirac mass at $\mathbb{E}(X|Y = y_\tau)$ but this is a very poor approximation to the posterior. Regarding f_X itself, the intuitive idea behind Robbins jump is that the *Bayes estimator* of X "lands" near $\mathscr{M}$. But this intuition must be validated by computing the covariance of the estimator, and this is indeed the next step to better understand the walk-jump sampling.

We tested the algorithm on the *handwritten digit database* for $\sigma = 0.3$ and $\sigma = 0.15$. The results are shown in Figure 5 and 6 respectively. For $\sigma = 0.3$ (in the regime of highly overlapping spheres), there was mixing between styles/classes. For $\sigma = 0.15$ (in the regime of mostly non-overlapping spheres), samples stayed within a class while the styles changed.

Some side-by-side comparisons with *denoising autoencoders* are in order.

- In walk-jump sampling, the key step is sampling from the density $\exp(-\phi)/Z$. The least-squares estimation of X—the jumps—are decoupled from Langevin MCMC. This is in contrast to the chain $X_t \to Y_t \to X_{t+1}$ constructed in *generalized denoising autoencoders* (Bengio et al., 2013b), which does not suffer from the problems we mentioned for the jumps.
- Here, the learning objective is derived for *any* σ. By contrast, *denoising autoencoders* approximate the score function only in the limit $\sigma \to 0$ (Alain and Bengio, 2014). *Generalized denoising autoencoders* (Bengio et al., 2013b) and *generative stochastic networks* (Alain et al., 2016) were devised as a remedy for those $\sigma \to 0$ limitations. This is avoided altogether here.
- In this work, there is a clear *geometrical* notion for large and small σ, which is based on the concentration of measure phenomenon. This picture is lacking in the references cited, and it is not clear which ranges of noise values should be considered there.
- Most importantly, denoising autoencoders suffer from *"limited parameterization"* as expressed by Alain and Bengio (2014), and independently observed in (Saremi et al., 2018). To summarize, in denoising autoencoders, one must learn a curl-free encoder-decoder to be able to properly approximate the *score function* and this is problematic beyond *one-hidden-layer* architectures. In our work, this problem is avoided due to the *energy parameterization* (see Remark 7).

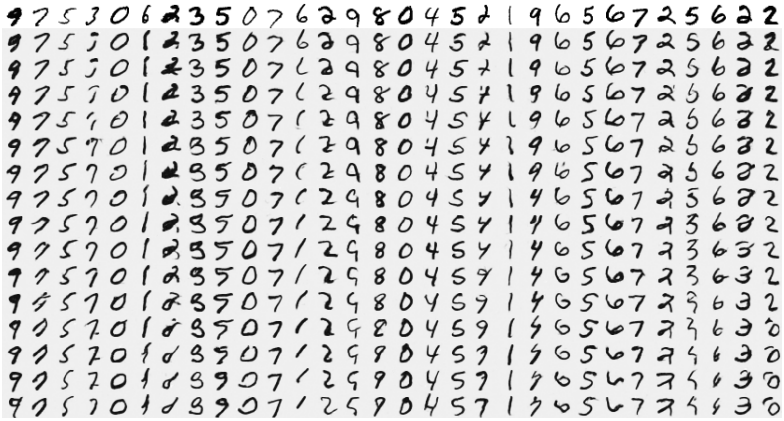


Figure 5: *($\sigma \approx \sigma_c$)* Top row is $x_0 \sim f_X$, sampled from the handwritten digit database which DEEN was trained on. The *Langevin MCMC* was intialized at $y_0 = x_0 + \varepsilon$, $\varepsilon \sim N(0, \sigma^2 I_d)$, where σ was the same value of σ which DEEN had been trained on. The samples y_t are are not shown. The *jumps* are shown in multiples of $\tau_0 = 10^4$, and the step size was $\delta = \sigma/100$. Here, $\sigma = 0.3$. The pixel values are in the range `[-0.07, 1.10]`.

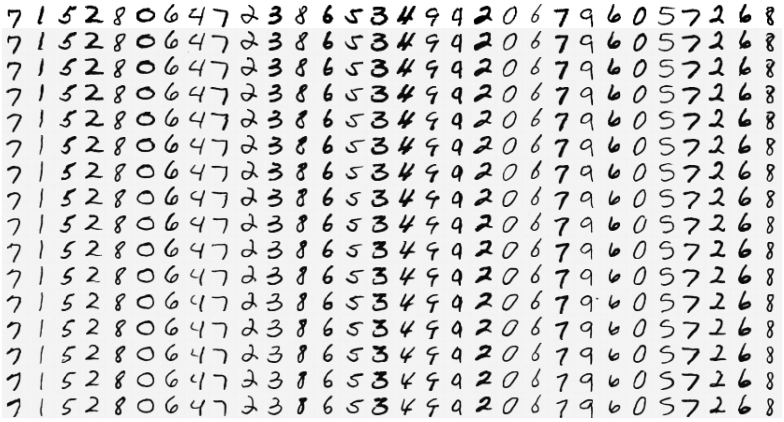


Figure 6: ($\sigma < \text{median}(\chi)$) Here, $\sigma = 0.15$, which is in the regime of mostly non-overlapping spheres. The Langevin MCMC parameters are set as above. The pixel values are in the range `[-0.05, 1.04]`.

6. Neural empirical Bayes associative memory (NEBULA)

In this section, the neural empirical Bayes machinery is used to define a new notion of associative memory. *Associative memory* (also known as *content-addressable memory*) is a deep subject in neural networks with a rich history and with roots in psychology and neuroscience (long-term potentiation and Hebb's rule). The depiction of this computation is even present in the arts and literature, championed by Marcel Proust, in the form of "stream of consciousness", and the constant back and forth between *perception* and *memory*.

In 1982, Hopfield brought together earlier attempts to formulate *associative memory*, and showed that the collective computation of a system of neurons with symmetric weights, in the form of asynchronous updates of *McCulloch-Pitts neurons* (McCulloch and Pitts, 1943), minimize an energy function (Hopfield, 1982). The energy function was constructed in a Hebbian fashion. The associative memory was then formulated as the "flow" (the neurons were binary) to the local minima of the energy function.

However, Hopfield's energy function is not *the* energy function[3]: it does not approximate (learn) the negative log probability density function of its stored memories. The *Boltzmann machine* (Hinton and Sejnowski, 1986) was developed in fact inspired by that observation, in which learning *the* energy function was achieved by introducing *hidden units*. Regarding the associative memory and the *phase space flow*, the problem is that in Boltzmann machines, hidden units need to be inferred first before having any notion of flow for the *visible units*. And inference is indeed computationally very expensive— *"the curse of inference"* (but not nearly as fundamental as the *curse of dimensionality*)—in *probabilistic graphical models* (Wainwright and Jordan, 2008; Koller and Friedman, 2009).

What we have achieved so far is to learn a function ϕ which approximates *the energy function* of Y, $\phi \approx -\log f$ (modulo a constant). The key here is that ϕ is a *function* (a "computer") that *computes* $f(y)$ for any y; the "hidden units" in ϕ are not inferred. In other words, the hidden units (and the parameters) in ϕ are there solely for *universal approximation*, $-\nabla\phi \approx \nabla \log f$. In *Boltzmann machines*, the hidden units are there to think! And this is the big difference. The definition below should be viewed against this backdrop.

Definition 9 *We define the neural empirical Bayes associative memory (NEBULA), à la Hopfield (1982), as the flow to strict local minima of the energy function ϕ. In continuous time, the memory dynamics is governed by the gradient flow:*

$$y'(t) = -\nabla\phi(y(t)). \tag{4}$$

NEBULA is identified by its attractors, the set of all the strict local minima of ϕ:

$$\mathcal{X}^* = \{X_1^*, \ldots, X_A^*\}.$$

The mapping $\mathscr{A} : \mathbb{R}^d \to \mathcal{X}^*$ denotes the convergence to an attractor under the gradient flow. The basin of attraction for the attractor X_a^* is denoted by $\mathcal{A}(X_a^*)$ and defined (intuitively) as the largest subset of $\mathbb{R}^d$ such that (in a slight abuse of notation) $\mathscr{A}(\mathcal{A}(X_a^*)) = \{X_a^*\}$.

3. In addition, Hopfield networks have severe limitations in *memory capacity*, addressed most recently in (Hillar and Tran, 2014; Krotov and Hopfield, 2016; Chaudhuri and Fiete, 2017).

What is so special about the local minima of ϕ? Start with a *single sample* X_1,

$$\phi(y) = \|y - X_1\|^2/(2\sigma^2),\ \mathcal{X}^* = \{X_1^*\},\quad X_1^* = X_1,\ \mathscr{A}(\cdot) = X_1,\ \mathcal{A}(X_1) = \mathbb{R}^d.$$

For *many samples*, the learning objective is *optimized* such that $-\nabla\phi$ evaluated on i-spheres point to X_i (see Figure 1c), but there are *conflicts of interests*, and these "conflicts" are the essence of i-sphere interactions, stated in Definition 3; in its practical summary, larger σ means more i-sphere overlaps, and therefore larger "interaction couplings". *For NEBULA, the presence of i-sphere interactions implies that X_a^* are not statistical samples from f_X.* In other words, given $X_i \sim f_X$, $\mathscr{A}(X_i) \not\sim f_X$. But what is the "right metric" $d(X_i, \mathscr{A}(X_i))$, to measure the distance between $X_i \sim f_X$ and $\mathscr{A}(X_i)$? The problem is that the flow is the gradient flow of $\phi \approx -\log f$ but the attractors are roughly speaking "close to $\mathscr{M}$". The total length of the path taken by the gradient flow is a natural choice but we leave that for furture studies. Our expectation (based on i-sphere interactions) is that $d(X_i, \mathscr{A}(X_i))$ will be larger for larger σ, visualized in Figures 7 and 8, with pronounced qualitative differences.

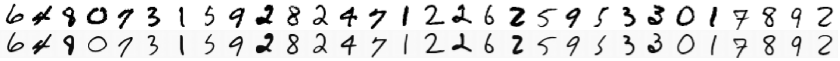


Figure 7: *(σ=0.15)* The top row are X_i from the MNIST test set. The bottom row are the attractors $\mathscr{A}(X_i)$; *they are not statistical samples from f_X: $\mathscr{A}(X_i) \not\sim f_X$.*

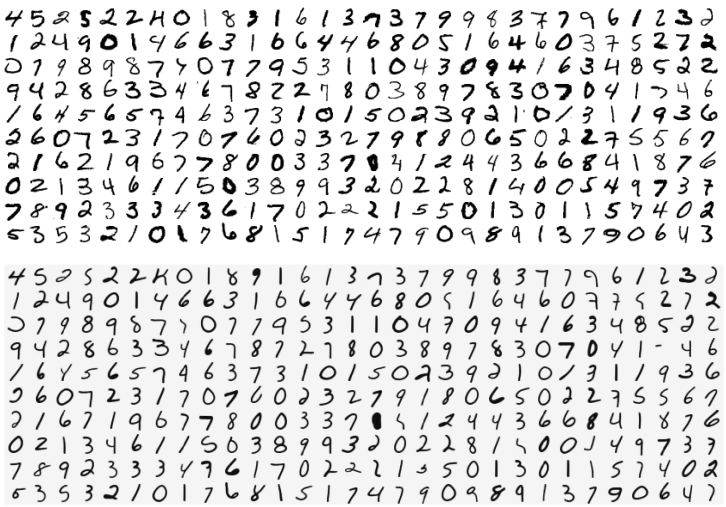


Figure 8: *(σ=0.3)* As expected, i-sphere interactions are stronger compared to $\sigma = 0.15$. Note the bottom right digit, flowing from a "3" to a "7".

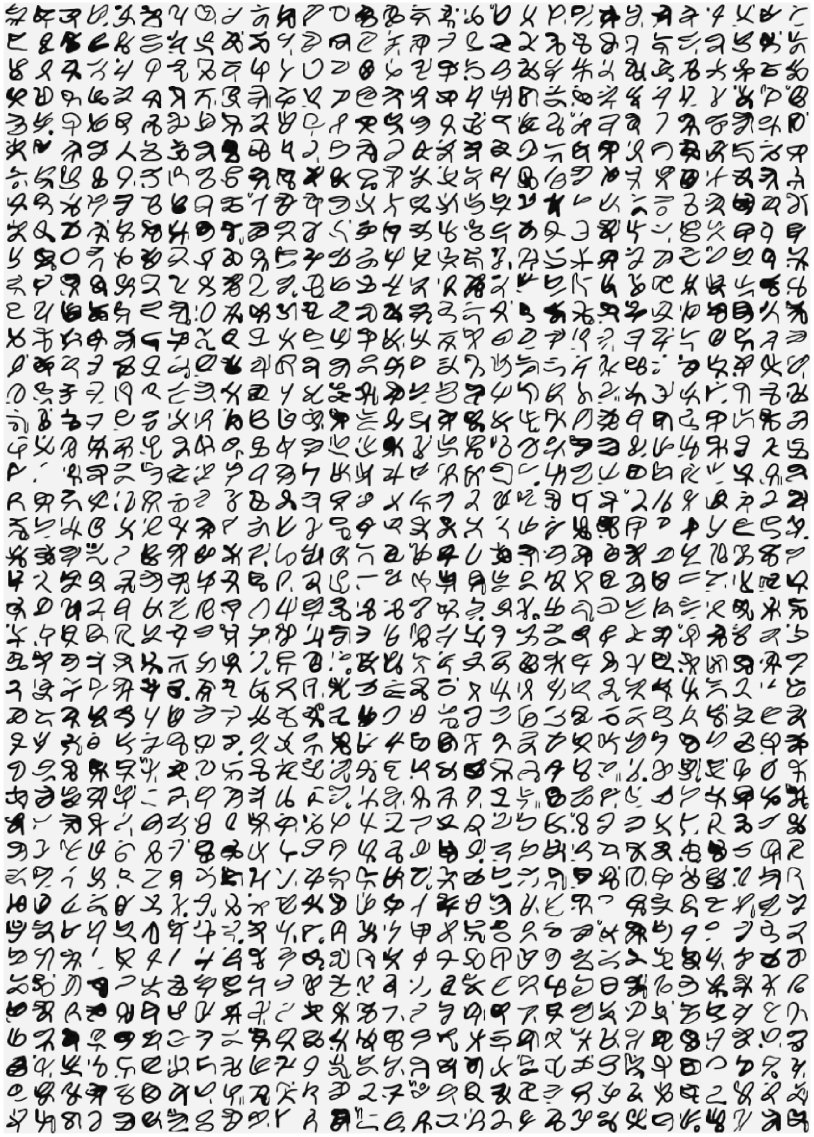


Figure 9: Attractors of NEBULA for $\sigma = 0.3$ ($\sigma_c \approx 0.29$).

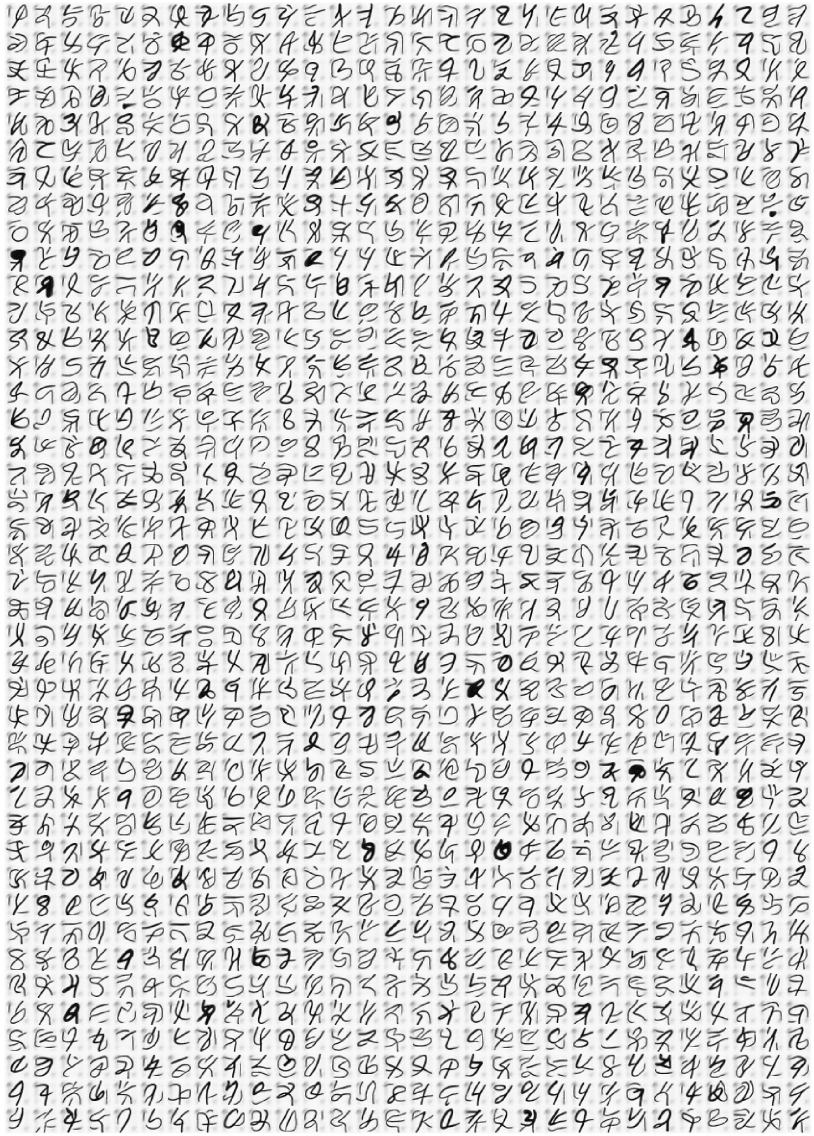


Figure 10: Attractors of NEBULA for $\sigma = 0.25$ ($\sigma_c \approx 0.29$).

7. Emergence of "creative memories"

We finish the paper with some surprising results on the emergence of highly-structured non-digit memories, named *"creative memories"*. The experiments designed in this section are a natural continuation of the experiments from the previous section. In its summary, NEBULA is "well behaved" when it is initialized at $X_i \sim f_X$. *But what if it is not initialized as such?* Here, we explore *i-sphere interactions* (see Definition 5) from a different angle by initializing NEBULA at a random point,

$$y_0 \sim \mathrm{Unif}([0,1]^d).$$

The results in Figures 9 and 10 are the strongest evidence for *i-sphere interactions* as being a good *abstraction*. In viewing Figures 9 and 10, consider Figure 2, but with *60000* highly overlapping i-spheres, and $-\nabla\phi$ on each i-sphere that should "point towards" X_i *in expectation*, where in addition, there are other complexities (touched upon in Remark 6) regarding the i-spheres themselves.

Several remarks are in order.

- The images shown in Figures 9 and 10 are *not* statistical samples from f_X nor f_Y.
- They are not the result of mass aggregation of the *kernel density estimator* $\widehat{f}_Y$. *Mode seeking* is a deep topic in the kernel density literature around the *mean shift algorithm* (Fukunaga and Hostetler, 1975; Cheng, 1995; Comaniciu and Meer, 2002), also around the intriguing topic of "ghost modes" (Carreira-Perpiñán and Williams, 2003). Of course, the *attractors* reported here has not been reported in the kernel density literature (see (Carreira-Perpiñán, 2015) and the examples therein for MNIST).
- The results presented are from all the random initializations in the *unit hypercube* from which we ran the algorithm, without any hand-picking!

8. Summary

In *neural empirical Bayes*, the smoothing of a random variable X to Y ($f = f_X * f_N$), denoising, and the (unnormalized) density estimation of the smoothed density f were unified in a single machinery. Its inner workings were captured symbolically by $X \rightleftharpoons Y$, as well as by a *Gedankenexperiment* with an "experimenter" in the school of Robbins (1956) and Robbins and Monro (1951).[4] In this machinery, the energy function is parametrized with a neural network and SGD becomes the engine for learning, whose end result is captured by $\nabla\phi(\cdot,\theta^*) \approx -\nabla \log f(\cdot)$. We proposed an approximative walk-jump sampling scheme which produces samples which are particularly appealing visually due to the denoising jump used. The energy function can further be used as an associative memory (NEBULA), which even has some "creative" capacities.

4. In our presentation, the smoothing part was *viewed* from the angle of *kernel density estimation*, with an important distinction that the kernel bandwidth was a *hyperparameter*. But on reflection, this is not necessary. Fundamentally, setting aside the "*neural* energy function", one can go on with our program by just knowing the (deep) machineries of *empirical Bayes* (Robbins, 1956) and *stochastic gradients* (Robbins and Monro, 1951).

Acknowledgement

This work was supported by the Canadian Institute for Advanced Research, the Gatsby Charitable Foundation, and the National Science Foundation through grant IIS-1718991. We especially thank Bruno Olshausen, Eero Simoncelli, and Francis Bach for discussions.

References

Harold Abelson and Gerald Jay Sussman. Structure and interpretation of computer programs. *MIT Press*, 1985.

Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

Guillaume Alain, Yoshua Bengio, Li Yao, Jason Yosinski, Eric Thibodeau-Laufer, Saizheng Zhang, and Pascal Vincent. GSNs: generative stochastic networks. *Information and Inference: A Journal of the IMA*, 5(2):210–249, 2016.

Philip W Anderson. More is different. *Science*, 177(4047):393–396, 1972.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153), 2018.

Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013a.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013b.

Miguel Á Carreira-Perpiñán. A review of mean-shift algorithms for clustering. *arXiv preprint arXiv:1503.00687*, 2015.

Miguel Á Carreira-Perpiñán and Christopher KI Williams. On the number of modes of a gaussian mixture. In *International Conference on Scale-Space Theories in Computer Vision*, pages 625–640. Springer, 2003.

Rishidev Chaudhuri and Ila Fiete. Associative content-addressable networks with exponentially many robust stable states. *arXiv preprint arXiv:1704.02019*, 2017.

Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 2018.

Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.

Christopher Hillar and Ngoc M Tran. Robust exponential memory in Hopfield networks. *arXiv preprint arXiv:1411.4625*, 2014.

Geoffrey E Hinton and Terrence J Sejnowski. Learning and relearning in Boltzmann machines. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1:282–317, 1986.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems*, pages 1172–1180, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Society, 2001.

David MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

Koichi Miyasawa. An empirical Bayes estimator of the mean of a normal population. *Bulletin of the International Statistical Institute*, 38(4):181–188, 1961.

Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7, 2017.

Martin Raphan and Eero P Simoncelli. Least squares estimation without priors or supervision. *Neural Computation*, 23(2):374–420, 2011.

Herbert Robbins. An empirical Bayes approach to statistics. In *Proc. Third Berkeley Symp.*, volume 1, pages 157–163, 1956.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

Saeed Saremi. On approximating ∇f with neural networks. *arXiv preprint arXiv:1910.12744*, 2019.

Saeed Saremi, Arash Mehrjou, Bernhard Schölkopf, and Aapo Hyvärinen. Deep energy estimator networks. *arXiv preprint arXiv:1805.08306*, 2018.

Lawrence K Saul and Sam T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4(Jun):119–155, 2003.

Terence Tao. *Topics in random matrix theory*. American Mathematical Society, 2012.

Aad van der Vaart. *Asymptotic statistics*. Cambridge University Press, 2000.

Nicolaas Godfried van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer, 1995.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press, 2018.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

Journal of Machine Learning Research 20 (2019) 1-46 Submitted 3/19; Revised 10/19; Published 10/19

Model Selection in Bayesian Neural Networks via Horseshoe Priors

Soumya Ghosh GHOSHSO@US.IBM.COM
MIT-IBM Watson AI Lab and Center for Computational Health, IBM Research, Cambridge, MA 02142.

Jiayu Yao JIY328@G.HARVARD.EDU
Finale Doshi-Velez FINALE@SEAS.HARVARD.EDU
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138.

Editor: Mohammad Emtiyaz Khan

Abstract

The promise of augmenting accurate predictions provided by modern neural networks with well-calibrated predictive uncertainties has reinvigorated interest in Bayesian neural networks. However, model selection—even choosing the number of nodes—remains an open question. Poor choices can severely affect the quality of the produced uncertainties. In this paper, we explore continuous shrinkage priors, the horseshoe, and the regularized horseshoe distributions, for model selection in Bayesian neural networks. When placed over node pre-activations and coupled with appropriate variational approximations, we find that the strong shrinkage provided by the horseshoe is effective at turning off nodes that do not help explain the data. We demonstrate that our approach finds compact network structures even when the number of nodes required is grossly over-estimated. Moreover, the model selection over the number of nodes does not come at the expense of predictive or computational performance; in fact, we learn smaller networks with comparable predictive performance to current approaches. These effects are particularly apparent in sample-limited settings, such as small data sets and reinforcement learning.

Keywords: Bayesian Neural Networks, Model Selection, Horseshoe Priors, Variational Inference, Structured approximations

1. Introduction

Bayesian neural networks (BNNs) treat the weights in a neural network as random variables. By performing posterior inference on these weights, BNNs can avoid overfitting in the regime of small data, provide posterior uncertainty estimates, and model a large class of stochastic functions with heteroskedastic and multi-modal noise. These properties have resulted in BNNs being adopted in applications ranging from Bayesian optimization (Springenberg et al., 2016) and active learning (Gal et al., 2016a) to reinforcement learning (Blundell et al., 2015; Depeweg et al., 2017).

While there have been many recent advances in training BNNs (Hernández-Lobato and Adams, 2015; Blundell et al., 2015; Gal and Ghahramani, 2016; Louizos and Welling, 2016; Hernandez-Lobato et al., 2016), model-selection has received relatively less attention. Unfortunately, the consequences for a poor choice of architecture are severe: too few nodes,

©2019 Soumya Ghosh, Jiayu Yao, Finale Doshi-Velez.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-236.html.

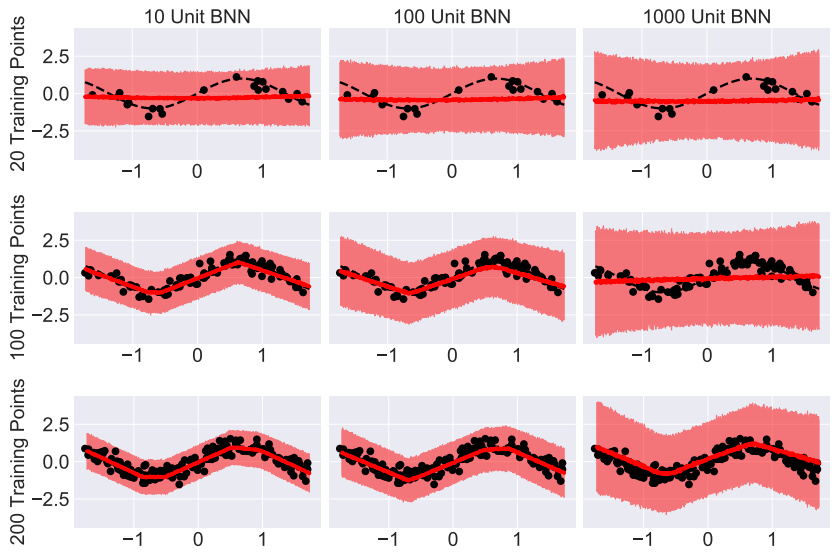


Figure 1: **Predictive distributions** from a single layer BNN with $\mathcal{N}(0,1)$ priors over weights, ReLU activations, containing ten, hundred, and thousand units, trained on noisy samples (in black) from a smooth one dimensional function shown in black, $y = \mathcal{N}(sin(x), 0.1)$. With fixed data increasing BNN capacity leads to large predictive uncertainties.

and the BNN will not be flexible enough to model the function of interest; too many nodes, and the BNN predictions will have large variance. We note that these Bayesian model selection concerns are subtly different from overfitting and underfitting concerns that arise from maximum likelihood training: here, more expressive models (e.g. those with more nodes) require more data to concentrate the posterior. When there is insufficient data, the posterior uncertainty over the BNN weights will remain large, resulting in large variances in the BNN's predictions. We illustrate this issue in Figure 1, where we see a BNN trained with a large number parameters has higher variance around its predictions than one with fewer. Thus, the core concern of Bayesian model selection is to identify a model class expressive enough that it can explain the observed data set, but not so expressive that it can explain everything (Rasmussen and Ghahramani, 2001; Murray and Ghahramani, 2005).

Model selection in BNNs is challenging because the number of nodes in a layer is a discrete quantity, forcing practitioners to perform onerous searches over different layer sizes. In this work, we demonstrate that we can perform computationally-efficient and statistically-

effective model selection in Bayesian neural networks by placing horseshoe (Carvalho et al., 2009) and related priors over the variance of weights incident to each node in the network. These priors can be interpreted as a continuous relaxation of a spike-and-slab approach that would assign a discrete on-off variable to each node. The prior has heavy tails and supports both zero values and large values. Thus, if we fix the mean of the incident weights to be zero, nodes with small variance parameters are effectively turned off—all incident weights will be close to zero—while nodes with large variance parameters can be interpreted as active. In this way, we can perform model selection over the number of nodes required in a Bayesian neural network. The continuous relaxation provided by these priors keep the model differentiable; with appropriate parameterization, we can take advantage of recent advances in variational inference (e.g., Kingma and Welling, 2014) for training. Especially in sample-limited settings, we demonstrate that our approach returns compact network structures even when the required number of nodes in the network is grossly over-estimated; these compact structures sacrifice little—and sometimes even improve—predictive performance. We demonstrate the value of these prediction gains through a reinforcement learning application.

Finally, we note that aspects of this work have appeared in (Ghosh and Doshi-Velez, 2017a,b; Ghosh et al., 2018); here we organize these results and provide a cohesive picture on priors for model selection in Bayesian neural networks.

2. Bayesian Neural Networks

A deep neural network with $L-1$ hidden layers is parameterized by a set of weight matrices $\mathcal{W} = \{W_l\}_1^L$, with each weight matrix W_l being of size $\mathbb{R}^{(K_{l-1}+1)\times K_l}$ where K_l is the number of units (excluding the bias) in layer l. The neural network maps an input $x \in \mathbb{R}^{D\times 1}$ to a response $f(\mathcal{W}, x)$ by recursively applying the transformation $h(W_l^T[z_{l-1}; 1])$, where the vector z_l is the input with the initial input $z_0 = x$, and h is a point-wise non-linearity, for instance the rectified-linear function, $h(a) = \max(0, a)$.

A Bayesian neural network captures uncertainty in the weight parameters $\mathcal{W}$ by endowing them with distributions $\mathcal{W} \sim p(\mathcal{W})$. Given a data set of N observation-response pairs $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$, we are interested in estimating the posterior distribution,

$$p(\mathcal{W} \mid \mathcal{D}) = \frac{p(\mathcal{W}) \prod_{n=1}^N p(y_n \mid f(\mathcal{W}, x_n))}{p(\mathcal{D})},$$

and leveraging the learned posterior for predicting responses to unseen data x_*, $p(y_* \mid x_*) = \int p(y_* \mid f(\mathcal{W}, x_*))p(\mathcal{W} \mid \mathcal{D})d\mathcal{W}$. The prior $p(\mathcal{W})$ ostensibly allows one to encode problem-specific beliefs as well as general properties about weights. Specifying meaningful priors for Bayesian neural networks, however, remains challenging.

2.1. Priors over neural network weights

Zero-mean Gaussian distributions are perhaps the most commonly used prior on weights. They have been explored both in the classic works of (MacKay, 1992; Neal, 1993, 1997) as well as in more recent approaches (Louizos and Welling, 2016; Hernández-Lobato and Adams, 2015; Louizos and Welling, 2017; Pawlowski et al., 2017). Log uniform (Kingma et al., 2015), two component scale mixture of Gaussians (Blundell et al., 2015), layer wise structured Gaussians (Louizos and Welling, 2016; Sun et al., 2017), as well as data space priors (Hafner et al., 2019) have been proposed. Buntine and Weigend (1991) explored entropic priors on BNN weights. While zero mean fixed variance Gaussian priors on weights are a convenient choice they can lead to undesirable pathologies (Figures 1, 4) — inflated predictive uncertainties, especially when used in conjunction with fully factorized variational inference. To see why such pathologies arise, note that the prior function, induced by a network with standard Gaussian weight priors, when evaluated at two data points exhibits a covariance that scales with the number of units in the penultimate layer (Williams, 1997). Further, since neural networks are over-parameterized, the prior can overwhelm the data, especially in the small data, wide network limit leading to high predictive uncertainties in these regimes. A work around is to scale the prior variances by the number of inputs as prescribed by Neal (1997). Alternatively, one could exploit the fact that weights of a network are effectively lower dimensional and attempt to parametrize the network in this lower dimensional space (Pradier et al., 2018; Izmailov et al., 2019). Such a parameterization may prevent the prior from overwhelming small amounts of data. Another option for avoiding such issues is to use hierarchical priors over weights (MacKay, 1995). This is the direction explored in this paper. In particular, we employ the horseshoe distribution (Carvalho et al., 2009), an example of a global-local shrinkage prior (Polson and Scott, 2010), that in addition to alleviating the inflated predictive uncertainty issue allows us to recover smaller networks.

3. Bayesian Neural Networks with Structured Sparsity

Let the node weight vector $w_{kl} \in \mathbb{R}^{(K_{l-1}+1)}$ denote the set of weights incident into unit k of hidden layer l. There exist several ways to induce group sparsity, that is, turn all the weights incident to a node small. In this work, we focus on the use of horseshoe priors to induce group sparsity.

3.1. Horseshoe Priors

We assume that each node weight vector w_{kl} is conditionally independent and distributed according to a Gaussian scale mixture,

$$w_{kl} \mid \tau_{kl}, \upsilon_l \sim \mathcal{N}(0, (\tau_{kl}^2 \upsilon_l^2)\mathbb{I}), \quad \tau_{kl} \sim C^+(0, b_0), \quad \upsilon_l \sim C^+(0, b_g). \tag{1}$$

Here, $\mathbb{I}$ is an identity matrix, $a \sim C^+(0, b)$ is the Half-Cauchy distribution with density $p(a|b) = 2/\pi b(1 + (a^2/b^2))$ for $a > 0$, τ_{kl} is a unit specific scale parameter, while the scale parameter υ_l is shared across the layer.

The distribution over weights in Equation 1 defines the horseshoe prior (Carvalho et al., 2009). It exhibits Cauchy-like flat, heavy tails while maintaining an infinitely tall spike at zero. Consequently, it has the desirable property of allowing sufficiently large node weight vectors w_{kl} to escape un-shrunk—by having a large scale parameter—while providing severe shrinkage to smaller weights. This is in contrast to Lasso style regularizers and their Bayesian counterparts that provide uniform shrinkage to all weights. By forcing the weights incident on a unit to share scale parameters, the prior in Equation 1 induces sparsity at the unit level, turning off units that are unnecessary for explaining the data well. Intuitively,

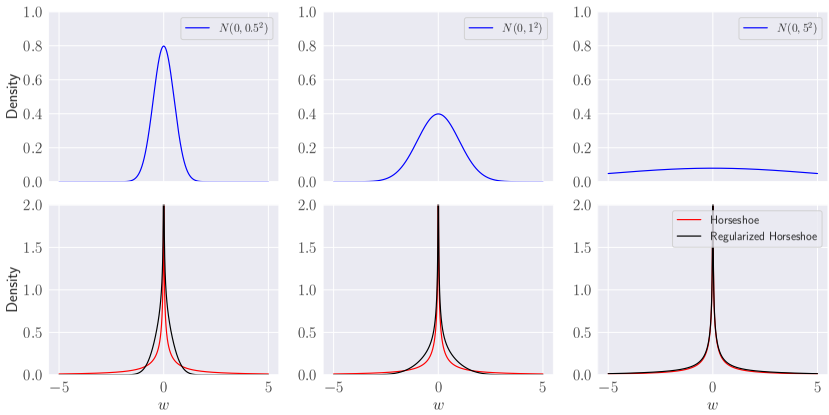


Figure 2: **Horseshoe and regularized horseshoe densities**. Both densities exhibit a spike at zero. While the horseshoe exhibits heavy Cauchy-like tails, the tails of the regularized Horseshoe are (soft) truncated. The soft truncation is governed by a zero-mean Gaussian density $\mathcal{N}(0, c^2)$, shown in the top row. Smaller values of c lead to sharper truncation. To model uncertainty in our beliefs about the truncation strength c, we place an inverse Gamma prior, $c^2 \sim \text{Inv-Gamma}(c_a, c_b)$. In the above plots, the horseshoe parameters b_0 and b_g are both set to one.

the shared layer wide scale υ_l pulls all units in layer l to zero, while the heavy tailed unit specific τ_{kl} scales allow some of the units to escape the shrinkage.

3.2. Regularized Horseshoe Priors

While the horseshoe prior has some good properties, when the amount of training data is limited, units with essentially no shrinkage can produce large weights and adversely affect generalization performance of HS-BNNs, with minor perturbations of the data leading to vastly different predictions. To deal with this issue, we consider a related alternative, the regularized horseshoe prior (Piironen and Vehtari, 2017). Under this prior w_{kl} is drawn from,

$$w_{kl} \mid \tau_{kl}, \upsilon_l, c \sim \mathcal{N}(0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}), \quad \tilde{\tau}_{kl}^2 = \frac{c^2 \tau_{kl}^2}{c^2 + \tau_{kl}^2 \upsilon_l^2}. \tag{2}$$

Note that for the weight node vectors that are strongly shrunk to zero, we will have tiny $\tau_{kl}^2 \upsilon_l^2$. When, $\tau_{kl}^2 \upsilon_l^2 \ll c^2$, $\tilde{\tau}_{kl}^2 \to \tau_{kl}^2 \upsilon_l^2$, recovering the original horseshoe prior. On the other hand, for the un-shrunk weights $\tau_{kl}^2 \upsilon_l^2$ will be large, and when $\tau_{kl}^2 \upsilon_l^2 \gg c^2$, $\tilde{\tau}_{kl}^2 \to c^2$. Thus, these weights under the regularized horseshoe prior follow $w_{kl} \sim \mathcal{N}(0, c^2\mathbb{I})$ and c acts as a weight decay hyper-parameter. We place a Inv-Gamma(c_a, c_b) prior on c^2. Another way to

interpret the regularized horseshoe prior is to observe that,

$$w_{kl} \mid \tau_{kl}, \upsilon_l, c \propto \mathcal{N}(0, (\tau_{kl}^2 \upsilon_l^2)\mathbb{I})\mathcal{N}(0, c^2\mathbb{I}).$$

This view makes it clear that the regularized variant soft thresholds the horseshoe tails by penalizing large values of w_{kl} through $\mathcal{N}(0, c^2\mathbb{I})$. See Figure 2 for a visual comparison of the densities of the horseshoe and the regularized horseshoe distributions for different values of c and Figure 24 for a comparison with the Laplace distribution. In the experimental section, we find that the regularizedHS-BNN does indeed improve generalization over HS-BNN.

3.3. Half-Cauchy re-parameterization for variational learning.

Both the standard and regularized horseshoe priors involve the Half-Cauchy distribution. While a direct parameterization of the Half-Cauchy distribution in Equation 1 is possible, it leads to challenges during variational learning: standard exponential family variational approximations struggle to capture the thick Cauchy tails, and using Cauchy approximating family leads to high variance gradients.

Therefore, we use a more convenient auxiliary variable parameterization (Wand et al., 2011),

$$a \sim C^+(0, b) \iff a^2 \mid \lambda \sim \text{Inv-Gamma}(\frac{1}{2}, \frac{1}{\lambda}); \lambda \sim \text{Inv-Gamma}(\frac{1}{2}, \frac{1}{b^2}),$$

where $v \sim \text{Inv-Gamma}(a, b)$ is the Inverse Gamma distribution with density $p(v) \propto v^{-a-1}\exp\{-b/v\}$ for $v > 0$. Since the number of output units is fixed by the problem at hand, there is no need for sparsity inducing prior for the output layer. We place isotropic Gaussian priors, $w_{kL} \sim \mathcal{N}(0, \kappa^2\mathbb{I})$ with vague hyper-priors $\kappa \sim C^+(0, b_\kappa = 5)$ on the output layer weights.

The joint distribution of the horseshoe Bayesian neural network is then given by,

$$\begin{aligned} p_{\text{HS}}(\mathcal{D}, \theta) =& r(\kappa, \rho_\kappa \mid b_\kappa) \prod_{k=1}^{K_L} \mathcal{N}(w_{kL} \mid 0, \kappa^2\mathbb{I}) \prod_{l=1}^{L} \Big\{ r(\upsilon_l, \vartheta_l \mid b_g) \\ & \prod_{k=1}^{K_l} r(\tau_{kl}, \lambda_{kl} \mid b_0) \mathcal{N}(w_{kl} \mid 0, (\tau_{kl}^2 \upsilon_l^2)\mathbb{I}) \Big\} \prod_{n=1}^{N} p(y_n \mid f(\mathcal{W}, x_n)), \end{aligned}$$

where $p(y_n | f(\mathcal{W}, x_n))$ is an appropriate likelihood function, and,

$$r(a, \lambda | b) = \text{Inv-Gamma}(a^2 | 1/2, 1/\lambda)\text{Inv-Gamma}(\lambda | 1/2, 1/b^2),$$

with $\theta = \{\mathcal{W}, \mathcal{T}, \kappa^2, \rho_\kappa\}$, $\mathcal{T} = \{\{\tau_{kl}\}_{k=1,l=1}^{K,L}, \{\upsilon_l\}_{l=1}^{L}, \{\lambda_{kl}\}_{k=1,l=1}^{K,L}, \{\vartheta_l\}_{l=1}^{L}\}$. An analogous reparameterization can be applied to the Bayesian neural network with regularized horseshoe priors:

$$\begin{aligned} p_{\text{regHS}}(\mathcal{D}, \theta) =& p(c \mid c_a, c_b) r(\kappa, \rho_\kappa \mid b_\kappa) \prod_{k=1}^{K_L} \mathcal{N}(w_{kL} \mid 0, \kappa^2\mathbb{I}) \prod_{l=1}^{L} \Big\{ r(\upsilon_l, \vartheta_l \mid b_g) \\ & \prod_{k=1}^{K_l} r(\tau_{kl}, \lambda_{kl} \mid b_0) \mathcal{N}(w_{kl} \mid 0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}) \Big\} \prod_{n=1}^{N} p(y_n \mid f(\mathcal{W}, x_n)), \end{aligned}$$

with $\theta = \{\mathcal{W}, \mathcal{T}, \kappa^2, \rho_\kappa, c^2\}$, $\mathcal{T} = \{\{\tau_{kl}^2\}_{k=1,l=1}^{K,L}, \{\upsilon_l^2\}_{l=1}^{L}, \{\lambda_{kl}\}_{k=1,l=1}^{K,L}, \{\vartheta_l\}_{l=1}^{L}\}$.

3.4. Non-Centered Parameterization

Both the standard and regularized horseshoe priors exhibit strong correlations between the weights w_{kl} and the scales $\tau_{kl}\upsilon_l$. While their favorable sparsity inducing properties stem from this coupling, it also gives rise to coupled posteriors that exhibit pathological funnel shaped geometries (Betancourt and Girolami, 2015; Ingraham and Marks, 2017) that are difficult to reliably sample or approximate. Fully factorized approximations are particularly problematic and can lead to non-sparse solutions erasing the benefits of using the horseshoe prior.

Adopting non-centered parameterizations (Ingraham and Marks, 2017), helps alleviate the issue. Consider a reformulation of Equation 2,

$$\beta_{kl} \sim \mathcal{N}(0, \mathbb{I}), \quad w_{kl} = \tilde{\tau}_{kl}\upsilon_l\beta_{kl}, \tag{3}$$

where the distribution on the scales are left unchanged. Since the scales and weights are sampled from independent prior distributions and are *marginally* uncorrelated, such a parameterization is referred to as non-centered. The likelihood is now responsible for introducing the coupling between the two, when conditioning on observed data. Non-centered parameterizations are known to lead to simpler posterior geometries (Betancourt and Girolami, 2015). Figure 3 summarizes the conditional dependencies assumed by the centered and the non-centered horseshoe BNN models. As with the Half-Cauchy reparameterization, this parameterization does not change the functional form of the standard or regularized horseshoe prior, but it will improve our ability to perform inference.

4. Variational Inference for Model Learning

We approximate the intractable posterior $p(\theta \mid \mathcal{D})$ with a computationally convenient alternative. The first step involves selecting a tractable family of distributions $q(\theta \mid \phi)$, with free variational parameters ϕ. Next, learning involves optimizing ϕ such that the Kullback-Liebler divergence between the approximation and the true posterior, $\mathrm{KL}(q(\theta \mid \phi)||p(\theta \mid \mathcal{D}))$ is minimized. This is equivalent to maximizing the lower bound to the marginal likelihood (or evidence) $p(\mathcal{D})$, $p(\mathcal{D}) \geq \mathcal{L}(\phi) = \mathbb{E}_{q(\theta|\phi)}[\ln p(\mathcal{D}, \theta)] + \mathbb{H}[q(\theta \mid \phi)]$.

4.1. Variational Approximation Choices

The choice of the approximating family governs the quality of inference. The more flexible the approximating family the better it approximates the true posterior. Below, we first describe a straight-forward fully-factored approximation and then more sophisticated structured approximations that we demonstrate to have better statistical properties.

4.1.1. Fully Factorized Approximation

The simplest possibility is to use a fully factorized variational family,

$$q(\theta \mid \phi) = \prod_{a \in \{c, \kappa, \rho_\kappa\}} q(a \mid \phi_a) \prod_{i,j,l} q(\beta_{ij,l} \mid \phi_{\beta_{ij,l}}) \prod_{k,l} q(\tau_{kl}^2 \mid \phi_{\tau_{kl}}) q(\lambda_{kl} \mid \phi_{\lambda_{kl}}) \prod_l q(\upsilon_l^2 \mid \phi_{\upsilon_l}) q(\vartheta_l \mid \phi_{\vartheta_l}). \tag{4}$$

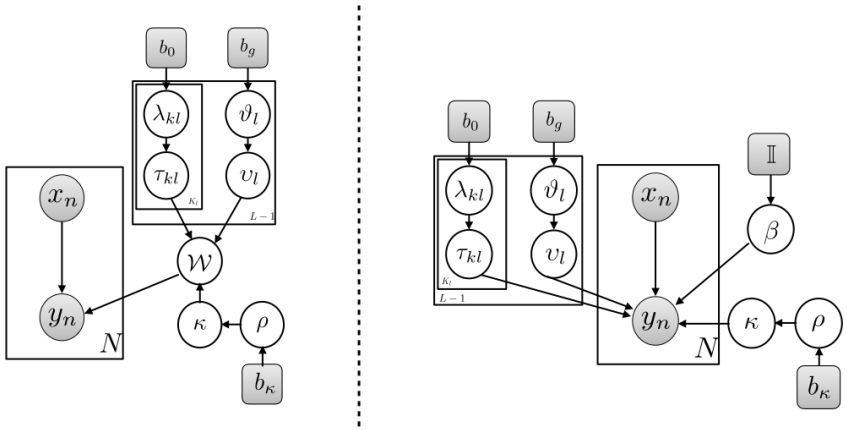


Figure 3: **Graphical models** capturing the conditional dependencies assumed by Bayesian Neural Networks with horseshoe priors. Left: Centered parameterization, Right: Non-centered parameterization necessary for robust inference.

Restricting the variational distribution for the non-centered weight $\beta_{ij,l}$ between units i in layer $l-1$ and j in layer l, $q(\beta_{ij,l} \mid \phi_{\beta_{ijl}})$ to the Gaussian family $\mathcal{N}(\beta_{ij,l} \mid \mu_{ij,l}, \sigma_{ij,l}^2)$, and the non-negative scale parameters τ_{kl}^2 and υ_l^2 and the variance of the output layer weights to the log-Normal family, $q(\ln \tau_{kl}^2 \mid \phi_{\tau_{kl}}) = \mathcal{N}(\mu_{\tau_{kl}}, \sigma_{\tau_{kl}}^2)$, $q(\ln \upsilon_l^2 \mid \phi_{\upsilon_l}) = \mathcal{N}(\mu_{\upsilon_l}, \sigma_{\upsilon_l}^2)$, and $q(\ln \kappa^2 \mid \phi_\kappa) = \mathcal{N}(\mu_\kappa, \sigma_\kappa^2)$, allows for the development of straightforward inference algorithms (Ghosh and Doshi-Velez, 2017a; Louizos et al., 2017). While other variational approximations for the scale parameters are possible, the log-Normal family is particularly convenient. It provides a simple parameterization allowing for reparameterized gradients, it is closed under multiplication, the product of global and local scales is thus another log-Normal distribution, and as we will see later in this section, it allows for convenient modeling of correlations between weights and scales. It is not necessary to impose distributional constraints on the variational approximations of the auxiliary variables ϑ_l, λ_{kl}, or ρ_κ. Conditioned on the other variables, the optimal variational family for these latent variables follow inverse Gamma distributions. We refer to this approximation as the *factorized* approximation.

Parameter-tying in the fully-factorized approximation. The conditional variational distribution on w_{kl} implied by Equations 3, 4 is $q(w_{kl} \mid \tau_{kl}, \upsilon_l) = \mathcal{N}(w_{kl} \mid \tau_{kl}\upsilon_l\mu_{kl}, (\tau_{kl}\upsilon_l)^2\Psi)$, where Ψ is a diagonal matrix with elements populated by $\sigma_{ij,l}^2$ and μ_{kl} consists of the corresponding variational means $\mu_{ij,l}$. The distributions of weights incident into a unit are thus coupled through $\tau_{kl}\upsilon_l$ while all weights in a layer are coupled through the layer wise scale υ_l. This view suggests that using a simpler approximating family $q(\beta_{ij,l} \mid \phi_{\beta_{ijl}}) = \mathcal{N}(\beta_{ij,l} \mid \mu_{ij,l}, 1)$ results in an isotropic Gaussian approximation $q(w_{kl} \mid \tau_{kl}, \upsilon_l) = \mathcal{N}(w_{kl} \mid$

$\tau_{kl}\upsilon_l\mu_{kl}, (\tau_{kl}\upsilon_l)^2\mathbb{I})$. Crucially, the scale parameters $\tau_{kl}\upsilon_l$ still allow for pruning of units when the scales approach zero. Moreover, by tying the variances of the non-centered weights together this approximation effectively halves the number of variational parameters to be learned and stored. Variational parameters associated with the network weights grow quadratically with network width while the scale and auxiliary variable parameters grow linearly. Hence, storage and learning costs in large networks are dominated by the weight parameters.

We find in Section 6.5 that such parameter tying speeds up training. However, this computational benefit comes at a price. The tied approximation retains the model selection benefits, but the variational distributions for the units that escape shrinkage are forced to be isotropic. This can lead to poorer predictive performance on regression tasks.

4.1.2. Structured Variational Approximations

Although computationally convenient, the factorized approximations fail to capture posterior correlations among the network weights, and more pertinently for the horseshoe prior, between weights and their scales.

Capturing layer-wise weight correlations We take a step towards a more structured variational approximation by using a layer-wise matrix variate normal (Gupta and Nagar, 1999) variational distribution for the non-centered weights while retaining the form of the other factors from Equation 4. Let $\beta_l \in \mathbb{R}^{(K_{l-1}+1)\times K_l}$ denote the set of weights between layers $l-1$ and l, then under this variational approximation we have, $q(\beta_l \mid \phi_{\beta_l}) = \mathcal{MN}(\beta_l \mid M_l, U_l, V_l)$, where $M_l \in \mathbb{R}^{(K_{l-1}+1)\times K_l}$ is the mean, $V_l \in \mathbb{R}^{K_l\times K_l}$ and $U_l \in \mathbb{R}^{(K_{l-1}+1)\times(K_{l-1}+1)}$ capture the covariances among the columns and rows of β_l, thereby modeling dependencies among the variational approximation to the weights in a layer. Louizos and Welling (2016) demonstrated that even when U_l and V_l are restricted to be diagonal, the matrix Gaussian approximation can lead to significant improvements over fully factorized approximations for standard BNNs. We call this the *semi-structured* approximation because it only captures correlations among weights but not between weights and scales.

Capturing weight-scale correlations The horseshoe prior exhibits strong correlations between weights and their scales, which encourages strong posterior coupling between β_{kl} and τ_{kl}. However, the variational families considered to this point do not account for this dependency. For effective shrinkage towards zero, it is important that the variational approximations are able to capture this strong dependence. We begin by defining, $B_l = \begin{bmatrix} \beta_l \\ \nu_l^T \end{bmatrix}$, $\nu_l = [\nu_{1l}, \ldots, \nu_{K_l l}]^T$, and $\nu_{kl} = \ln \tau_{kl}^2$. Using the variational approximation $q(B_l \mid \phi_{B_l}) = \mathcal{MN}(B_l \mid M_l, U_l, V_l)$, allows us to retain the coupling between weights incident onto a unit and the corresponding unit specific scales, with appropriate parameterizations of U_l. In particular, we note that a diagonal U_l fails to capture the necessary correlations, and defeats the purpose of using a matrix Gaussian variational family to model the posterior of B_l. To retain computational efficiency while capturing dependencies among the rows of B_l we enforce a low-rank structure, $U_l = \Psi_l + h_l h_l^T$, where $\Psi_l \in \mathbb{R}^{(K_{l-1}+2)\times(K_{l-1}+2)}$ is a diagonal matrix and $h_l \in \mathbb{R}^{(K_{l-1}+2)\times 1}$ is a column vector. We retain a diagonal structure for $V_l \in \mathbb{R}^{K_l\times K_l}$. We refer to this as the *structured* approximation. In Section 6, we find that this structured approximation indeed leads to stronger shrinkage towards zero in

Table 1: Variational Approximation Families.

Approximation	Description
Factorized	$q(\nu_l \mid \phi_{\nu_l})q(\beta_l \mid \phi_{\beta_l}) = \prod_{i,j,l} \mathcal{N}(\mu_{ij,l}, \sigma^2_{ij,l}) \prod_{k,l} q(\nu_{kl} \mid \phi_{\nu_{kl}})$
Factorized (tied)	$q(\nu_l \mid \phi_{\nu_l})q(\beta_l \mid \phi_{\beta_l}) = \prod_{i,j,l} \mathcal{N}(\mu_{ij,l}, 1) \prod_{k,l} q(\nu_{kl} \mid \phi_{\nu_{kl}})$
Semi-structured	$q(\nu_l \mid \phi_{\nu_l})q(\beta_l \mid \phi_{\beta_l}) = \mathcal{MN}(\beta_l \mid M_l, U_l, V_l) \prod_{k,l} q(\nu_{kl} \mid \phi_{\nu_{kl}})$
Structured	$q(\beta_l, \nu_l \mid \phi_{B_l}) = \mathcal{MN}(B_l \mid M_l, U_l, V_l)$

the recovered solutions. When combined with an appropriate pruning rule, it significantly compresses networks with excess capacity.

Table 1 summarizes the variational approximations introduced in this section. All our variational approximations employ a, possibly correlated, Gaussian distribution for the weights and log-Normal distributions for the scale parameters. These choices were primarily governed by computational ease. The Gaussian distribution on weights allows for the use of the "local reparameterization trick" (Kingma et al., 2015), and consequently low variance gradients. The log-Normal distributions are similarly amenable to reparameterized gradients again allowing for low variance gradients. Moreover, it allows us to easily model correlations between the weights and the log scales through a multivariate Gaussian distribution. Other choices of variational distributions are certainly possible. Exploring variational distribution families that better trade off computational ease for accuracy is an interesting direction for future work.

4.2. Black Box Variational Inference

Irrespective of the chosen variational family, the resulting evidence lower bound (ELBO),

$$\mathcal{L}(\phi) = \sum_n \mathbb{E}_{q(\theta|\phi)}[\ln\, p(y_n \mid f(\theta, x_n))] + \mathbb{E}_{q(\theta|\phi)}[\ln\, p(\theta \mid b_0, b_g, b_\kappa, c_a, c_b)] + \mathbb{H}[q(\theta \mid \phi)], \quad (5)$$

is challenging to evaluate. Here we have used β to denote the set of all non-centered weights in the network. The non-linearities introduced by the neural network and the potential lack of conjugacy between the neural network parameterized likelihoods and the horseshoe priors render the first expectation in Equation 5 intractable. Consequently, the traditional prescription of optimizing the ELBO by cycling through a series of fixed point updates is no longer available.

Recent progress in black box variational inference (Kingma and Welling, 2014; Rezende et al., 2014; Ranganath et al., 2014; Titsias and Lázaro-gredilla, 2014) subverts this difficulty. These techniques compute noisy unbiased estimates of the gradient $\nabla_\phi \hat{\mathcal{L}}(\phi)$, by approximating the offending expectations with unbiased Monte-Carlo estimates and relying on either score function estimators (Williams, 1992; Ranganath et al., 2014) or reparameterization gradients (Kingma and Welling, 2014; Rezende et al., 2014; Titsias and Lázaro-gredilla, 2014) to differentiate through the sampling process. With the unbiased gradi-

ents in hand, stochastic gradient ascent can be used to optimize the ELBO. In practice, reparameterization gradients exhibit significantly lower variances than their score function counterparts and are typically favored for differentiable models. The reparameterization gradients rely on the existence of a parameterization that separates the source of randomness from the parameters with respect to which the gradients are sought. For our Gaussian variational approximations, the well known non-centered parameterization, $\zeta \sim \mathcal{N}(\mu, \sigma^2) \Leftrightarrow \epsilon \sim \mathcal{N}(0,1), \zeta = \mu + \sigma\epsilon$, allows us to compute Monte-Carlo gradients,

$$\nabla_{\mu,\sigma}\mathbb{E}_{q_w}[g(w)] \Leftrightarrow \nabla_{\mu,\sigma}\mathbb{E}_{\mathcal{N}(\epsilon|0,1)}[g(\mu+\sigma\epsilon)] \approx \frac{1}{S}\sum_s \nabla_{\mu,\sigma} g(\mu + \sigma\epsilon^{(s)}), \tag{6}$$

for any differentiable function g and $\epsilon^{(s)} \sim \mathcal{N}(0,1)$. Furthermore, practical implementations of variational Bayesian neural networks typically use a further re-parameterization to lower variance of the gradient estimator. They sample from the implied variational distribution over a layer's pre-activations instead of directly sampling the much higher dimensional weights (Kingma et al., 2015).

4.2.1. Variational distribution on pre-activations for structured approximating families

While the "local" re-parametrization is straightforward for the factorized and semi-structured approximations, it is more involved for the structured case. We observe that by factorizing $q(B_l \mid \phi_{B_l})$ as $q(\beta_l \mid \nu_l, \phi_{\beta_l})q(\nu_l \mid \phi_{\nu_l})$ and that conditioned on $\nu_l \sim q(\nu_l \mid \phi_{\nu_l})$, β_l follows another matrix Gaussian distribution. This conditional variational distribution is given by, $q(\beta_l \mid \nu_l, \phi_{\beta_l}) = \mathcal{MN}(M_{\beta_l|\nu_l}, U_{\beta_l|\nu_l}, V)$. The conditional mean $M_{\beta_l|\nu_l}$ and row variances $U_{\beta_l|\nu_l}$ expressions are derived in the supplement. It then follows that $b = \beta_l^T a$ for an input $a \in \mathbb{R}^{(K_{l-1}+1)\times 1}$ into layer l, is distributed as,

$$b \mid a, \nu_l, \phi_{\beta_l} \sim \mathcal{N}(b \mid \mu_b, \Sigma_b), \tag{7}$$

with $\mu_b = M_{\beta_l|\nu_l}^T a$, and $\Sigma_b = (a^T U_{\beta_l|\nu_l} a)V$. Since, $a^T U_{\beta_l|\nu_l} a$ is scalar and V is diagonal, Σ is diagonal as well. For regularized HS-BNN, recall that the pre-activation of node k in layer l, is $u_{kl} = \tilde{\tau}_{kl} \upsilon_l b$, and the corresponding variational posterior is,

$$q(u_{kl} \mid \mu_{u_{kl}}, \sigma^2_{u_{kl}}) = \mathcal{N}(u_{kl} \mid \mu_{u_{kl}}, \sigma^2_{u_{kl}}); \quad \mu_{u_{kl}} = \tilde{\tau}_{kl}^{(s)} \upsilon_l^{(s)} \mu_{bk}; \quad \sigma^2_{u_{kl}} = \tilde{\tau}_{kl}^{(s)^2} \upsilon_l^{(s)^2} \Sigma_{bk,k}, \tag{8}$$

where $\tau_{kl}^{(s)}$, $\upsilon_l^{(s)}, c^{(s)}$ are samples from the corresponding log-Normal posteriors and $\tilde{\tau}_{kl}^{(s)}$ is constructed as $c^{(s)^2}\tau_{kl}^{(s)^2}/(c^{(s)^2} + \tau_{kl}^{(s)^2}\upsilon_l^{(s)^2})$. Thus, to compute the pre-activation distributions under the structured variational approximation, we first sample the vector ν_l and then sampling the pre-activations conditioned on ν_l and inputs into the layer. Crucially, we can perform all operations without having to sample the high dimensional weights β_l.

4.2.2. Algorithm

We now have a simple prescription for optimizing Equation 5. Recursively sampling the variational posterior of Equation 8 for each layer of the network, allows us to forward

propagate information through the network. Using the reparameterizations (Equation 6), allows us to differentiate through the sampling process. We compute the necessary gradients through reverse mode automatic differentiation tools (Maclaurin et al., 2015). With the gradients in hand, we optimize $\mathcal{L}(\phi)$ with respect to the variational weights ϕ_B, per-unit scales $\phi_{\tau_{kl}}$, per-layer scales ϕ_{υ_l}, and the variational scale for the output layer weights, ϕ_κ using Adam (Kingma and Ba, 2015). For the structured approximation, we found a further heuristic — wherein we restrict the variational distributions such that the parameters Ψ, V, and h lie in the ℓ_2 unit-ball produced better predictive performance. A more rigorous treatment of this heuristic is left as future work. Conditioned on these, the optimal variational posteriors of the auxiliary variables ϑ_l, λ_{kl}, and ρ_κ follow inverse gamma distributions. Fixed point updates that maximize $\mathcal{L}(\phi)$ with respect to $\phi_{\vartheta_l}, \phi_{\lambda_{kl}}, \phi_{\rho_\kappa}$, holding the other variational parameters fixed are available. It can be shown that,

$$q(\lambda_{kl} \mid \phi_{\lambda_{kl}}) = \text{Inv-Gamma}(\lambda_{kl} \mid 1, \mathbb{E}[\frac{1}{\tau_{kl}^2}] + \frac{1}{b_0^2}). \tag{9}$$

The distributions of the other auxiliary variables are analogous. By alternating between gradient and fixed point updates to maximize the ELBO in a coordinate ascent fashion we learn all variational parameters jointly (see Algorithm 1).

4.2.3. Computational Considerations

The primary computational bottleneck for the structured approximation arises in computing the pre-activations in Equation 7. While computing Σ_b in the factorized approximation involves a single inner product, in the structured case it requires the computation of the quadratic form $a^T U_{M_{\beta_l|\nu_l}} a$ and a point wise multiplication with the elements of V_l. Owing to the diagonal plus rank-one structure of $U_{M_{\beta_l|\nu_l}}$, we only need two inner products, followed by a scalar squaring and addition to compute the quadratic form and K_l scalar multiplications for the point-wise multiplication with V_l. Thus the structured approximation is only marginally more expensive. Further, it uses only $(K_l + 2) \times (K_{l-1} + 1)$ weight variance parameters per layer, instead of $K_l \times (K_{l-1} + 1)$ parameters used by the factorized approximation. Not having to compute gradients and update these additional parameters further mitigates the performance difference.

Algorithm 1 Training with Standard and Regularized HS-BNNs

Input Model $p(\mathcal{D}, \theta)$, variational approximation $q(\theta \mid \phi)$, number of iterations T.
Output: Variational parameters ϕ
Initialize variational parameters ϕ.
for `T iterations` **do**
  Update ϕ_c, ϕ_κ, ϕ_γ, $\{\phi_{B_l}\}_l$, $\{\phi_{\upsilon_l}\}_l \leftarrow \text{ADAM}(\mathcal{L}(\phi))$.
  for `all hidden layers` l **do**
    Conditioned on ϕ_{B_l}, ϕ_{υ_l} update ϕ_{ϑ_l}, $\phi_{\lambda_{kl}}$ using fixed point updates (Equation 9).
  end for
  Conditioned on ϕ_κ update ϕ_{ρ_κ} via the corresponding fixed point update.
end for

4.3. Pruning Rule

The group horseshoe and its regularized variant provide strong shrinkage towards zero for small w_{kl}. However, since we infer a posterior distribution the shrunk weights, although tiny, are never actually zero. A user-defined thresholding rule is required to prune away the shrunk weights. In previous work, Louizos et al. (2017) first summarized the inferred posterior distributions using a point estimate and then used the point summary to define a thresholding rule. Here, we propose an alternate thresholding rule that does not rely on a point summary of the posterior. We prune away a unit if $p(\tau_{kl}\upsilon_l < \delta) > p_0$, that is, when the probability of the scale being small is sufficiently large. The scale δ and the probability p_0 are user defined parameters, with $\tau_{kl} \sim q(\tau_{kl} \mid \phi_{\tau_{kl}})$ and $\upsilon_l \sim q(\upsilon_l \mid \phi_{\upsilon_l})$. To see why this rule is sensible, recall that for units which experience strong shrinkage the regularized horseshoe tends to the horseshoe. Under the horseshoe prior, $\tau_{kl}\upsilon_l$ governs the (non-negative) scale of the weight node vector w_{kl}. Therefore, under our thresholding rule, we prune away nodes whose posterior scales, place probability greater than p_0 below a sufficiently small threshold δ. In our experiments, we set $p_0 = 0.9$ and δ to either $1e-3$ or $1e-5$ (for the year data set). Further, under our variational approximations, $\tau_{kl}\upsilon_l$ follows a log-Normal distribution. Evaluating the thresholding rule is as simple as evaluating the log-Normal cumulative distribution.

5. Related Work

Early work on Bayesian neural networks can be traced back to (Buntine and Weigend, 1991; MacKay, 1992; Neal, 1993). Neal (1993) introduced Hamiltonian Monte Carlo (HMC) for exploring the posterior over network weights, while MacKay (1992); Buntine and Weigend (1991) relied on Laplace approximation to characterize the network posterior. HMC remains the gold standard for posterior inference in BNNs. However, it does not scale well to modern architectures or the large data sets required to learn them. Similarly, vanilla application of the Laplace approximation has difficulty scaling to modern architectures with millions of parameters. Recent advances in stochastic MCMC methods (Welling and Teh, 2011; Chen et al., 2014) and stochastic variational methods (Blundell et al., 2015; Rezende et al., 2014), black-box variational inference and alpha-divergence minimization (Hernandez-Lobato et al., 2016; Ranganath et al., 2014), probabilistic backpropagation (Hernández-Lobato and Adams, 2015) as well as scalable Laplace approximations (Ritter et al., 2018) have reinvigorated interest in BNNs by allowing efficient scalable inference. We direct the interested reader to Yao et al. (2019) for a thorough comparison of modern inference algorithms for BNNs.

Work on learning structure in BNNs has received less attention. Blundell et al. (2015) introduce a mixture-of-Gaussians prior on the weights, with one mixture tightly concentrated around zero, thus approximating a spike and slab prior over weights. Others (Kingma et al., 2015; Gal and Ghahramani, 2016) have noticed connections between Dropout and approximate variational inference (Srivastava et al., 2014). In particular, Molchanov et al. (2017) show that the interpretation of Gaussian dropout as performing variational inference in a network with log uniform priors over weights leads to sparsity in weights. The goal of turning off edges is very different than the approach considered here, which performs model selection over the appropriate number of nodes. Yet others (Neklyudov et al., 2017) have proposed pruning units via truncated log-normal priors over unit scales. However, they do not place priors over network weights and are unable to infer posterior uncertainty over weights, which may lead to poorer predictive uncertainties. Related but orthogonal research (Adams et al., 2010; Song et al., 2017) has focused on the problem of structure learning in deep belief networks.

Closely related to this work is the work of Louizos et al. (2017). They also consider group Horseshoe priors for model compression. In this paper, we take a closer look at the model selection properties afforded by such priors. We also provide robust extensions to the model—via the regularized Horseshoe prior—the inference—via structured variational approximations—and provide a thresholding rule to prune units with small scales.

There is also a body of work on learning structure in non-Bayesian neural networks. Early work (LeCun et al., 1990; Hassibi et al., 1993) pruned networks by analyzing second-order derivatives of the objectives. More recently, Wen et al. (2016) describe applications of structured sparsity not only for optimizing filters and layers but also computation time. Closer to our work in spirit, Ochiai et al. (2016); Scardapane et al. (2017); Alvarez and Salzmann (2016); Murray and Chiang (2015), use group sparsity to prune groups of weights—e.g. weights incident to a node. However, these approaches do not model uncertainty in weights and provide uniform shrinkage to all parameters. Our approach provides group shrinkage while retaining weight uncertainties.

6. Experiments

In this section, we carefully evaluate our proposed modeling and inferential contributions. First, on synthetic data we explore the properties of the horseshoe BNN (HS-BNN) and its different parameterizations. We find that employing a non-centered parameterization is necessary for effective model selection. Next, on several real-world data sets we vet both the model selection effects and the predictive performance of the non-centered HS-BNN and find that it is able to provide model selection benefits without significantly sacrificing predictive performance. We next evaluate the impact of our modeling and inference extensions. On the modeling front, we find that the regularized Horseshoe BNN (reg-HS) improves upon the predictive performance of HS-BNN, especially on smaller data sets. Through further controlled comparisons we find that the structured variational approximation improves upon the factorized variants by recovering solutions that exhibit significantly stronger shrinkage towards zero. Finally, we end with experiments that go beyond prediction and demonstrate the value of HS-BNNs for reinforcement learning. In our experiments, unless otherwise mentioned we use a learning rate of 0.005. For regression problems we employ a Gaussian likelihood with an unknown precision γ, $p(y_n|f(\mathcal{W}, x_n), \gamma) = N(y_n|f(\mathcal{W}, x_n), \gamma^{-1})$. We place a vague prior on the precision, $\gamma \sim \text{Gamma}(6, 6)$ and approximate the posterior over γ using another Gamma distribution. For classification problems we use a Categorical distribution parameterized by $\mathcal{S}(f(\mathcal{W}, x_n))$, where $\mathcal{S}$ is the softmax transformation.

6.1. Underfitting variational Bayesian neural networks

We began by exploring the effect of model specification on BNNs learned via fully factorized variational inference. We generated varying amounts of synthetic data from $y \sim \mathcal{N}(sin(x), 0.1)$ and fit fully connected BNN models of varying size. We used ReLU activa-

tions and placed factorized Gaussian priors over the network weights, $w_{ij,l} \sim \mathcal{N}(0,1)$, where $w_{ij,l}$ denotes the weight between nodes i in layer $l-1$ and j in layer l. We used black-box variational inference with locally reparameterized gradients (Kingma et al., 2015). We trained the models to convergence from thirty random restarts and selected the highest evidence lower bound solution.

We found that BNNs trained under this regime were not robust to model specification, quickly increasing the uncertainty of their predictions as more nodes were added to the network while keeping the amount of training data fixed. This is illustrated in Figure 1. Figure 4 visualizes the corresponding weights, where we plot the 2-Wasserstein distance between the inferred variational posterior and the prior, $\mathcal{W}_2(\mathcal{N}(w_{ij,l} \mid \mu_{ij,l}, \sigma^2_{ij,l}), \mathcal{N}(w_{ij,l} \mid 0,1))$. This experiment leads to the following observation. *A sufficiently wide BNN with $\mathcal{N}(0,\sigma^2)$ prior on weights, when conditioning on a sufficiently small data set and trained via fully factorized variational inference produces a posterior approximation that tends towards the prior and consequently a posterior predictive distribution that reverts to the prior predictive distribution.* To gain further insight into this observation, note that the ELBO for a BNN is $\sum_{n=1}^{N} \mathbb{E}_{q(\mathcal{W}|\phi)}[\ln p(y_n \mid f(\mathcal{W}, x_n)] - \mathrm{KL}(q(\mathcal{W} \mid \phi)||p(\mathcal{W})) = \sum_{n=1}^{N} \mathbb{E}_{q(\mathcal{W}|\phi}[\ln p(y_n \mid f(\mathcal{W}, x_n)] - \sum_{j=1}^{P} \mathrm{KL}(q(w_j \mid \phi)||\mathcal{N}(w_j \mid 0,1))$, where the equality follows from the fully factorized variational approximation. When P, the number of weights, is significantly larger than N, the KL term in the ELBO dominates incentivizing the posterior approximation $q(\mathcal{W} \mid \phi)$ to revert to the prior.

6.2. Demonstrations of HS-BNN Behavior on Synthetic Data

In this section, we first demonstrate the value of the Horseshoe prior by returning to the the demonstration in Figure 1. We also demonstrate the importance of non-centered parameterization of the Horseshoe distribution.

6.2.1. Horseshoe priors demonstrate robustness to model specification.

In Figure 5, we reproduce the results of Figure 1 alongside the fit of a 1000 unit HS-BNN. While the fit of the BNN with zero mean, unit variance Gaussian priors quickly deteriorates with increasing capacity conditioned on the limited amount of training data, HS-BNN is robust to model-misspecification, pruning away the additional capacity. Like the BNN, HS-BNN training employed thirty random initializations. Solutions with the highest evidence lower bound are plotted.

To verify that this effect holds across data sets, we also reproduced the noisy polynomial regression problem considered in (Hernández-Lobato and Adams, 2015), $y_n = x_n^3 + \epsilon_n$, $\epsilon_n \sim \mathcal{N}(0,9)$. Figure 6 illustrates the fits recovered by a Bayesian neural network with Gaussian priors against a HS-BNN conditioned on data sampled from $\mathcal{U}(-4,+4)$. We sample a total of 500 points, using a 100 for training and rest for testing. We again find that the BNN underfits the data and significantly overestimates the predictive uncertainty. HS-BNN on the other hand provides improved estimates of predictive uncertainty. We also plot the test predictive log likelihoods averaged over five random trials in Figure 6. Observe that with increasing size, the predictive likelihoods drop significantly for BNN. HS-BNN exhibits robustness with more modest drops in performance. Later in this section we will

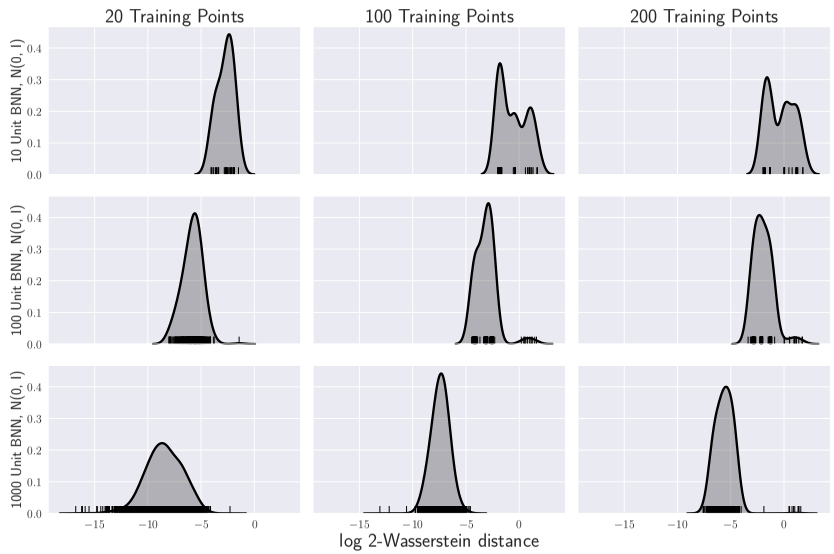


Figure 4: **Analysis of posterior weights of variational BNNs with $\mathcal{N}(0,1)$ priors**. This figure provides the posterior weights corresponding to the predictive distributions presented in Figure 1. Each rug represents the (squared) 2-Wasserstein distance the inferred posterior and the prior, $\mathcal{W}_2(\mathcal{N}(w_{ij,l} \mid \mu_{ijl}, \sigma^2_{ijl}), \mathcal{N}(w_{ij,l} \mid 0,1))^2$ for weight $w_{ij,l}$. A kernel density estimate of the Wasserstein distances are also plotted. Along each column, from top to bottom the width of the network increases while the number of training instances remains constant. Observe that the density plots exhibit a leftward shift from top to bottom along each column and a rightward shift along each row from left to right. These results indicate that for a fixed size network as the number of training instances increase posterior weights deviate more strongly from the prior and consequently exhibit better fits to the data. On the other hand for a fixed amount of training data, increasing the network size encourages the prior to overwhelm the data resulting in posteriors that do not deviate from the prior. *Note that with respect to Figure 1, we have transposed the columns and rows here to more clearly illustrate the effects of network size on posterior inference.*

see that using more sophisticated variational approximations further improves results where predictive performance no longer deteriorates with increasing capacity.

6.2.2. Non-centered parameterizations are crucial for reliable inferences

The results above crucially rely on using a non-centered parameterization. We illustrate the importance of the non-centered parameterization with a simple two dimensional classification problem generated by sampling data uniformly at random from $[-1,+1] \times [-1,+1]$

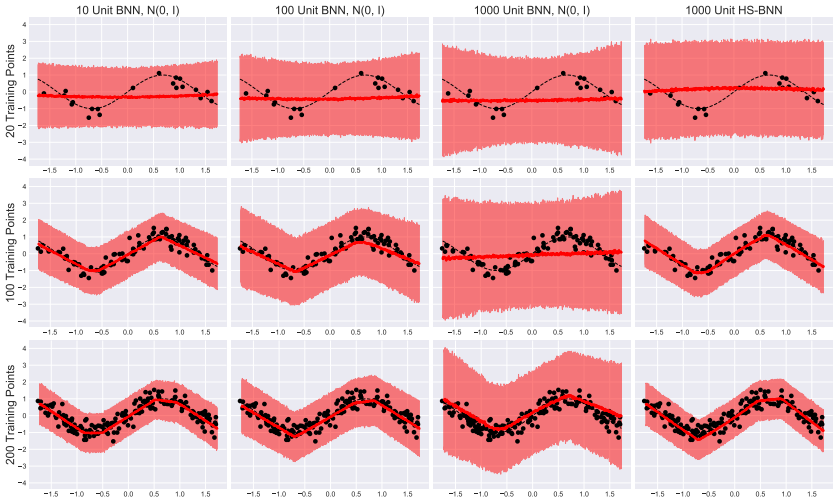


Figure 5: **Horseshoe BNN**. We return to the example from Figure 1. The 1000 node HS-BNN (right-most column) underfits significantly less than its 1000 node standard BNN counterpart.

and using a 2-2-1 network, whose parameters are known *a-priori* to generate the class labels. We train three Bayesian neural networks with a 15-unit hidden layer on these data, with Gaussian priors, with non-centered horseshoe priors, and with centered horseshoe priors. All three models are able to easily fit the data and provide high predictive accuracy. However, the structure learned by the three models are very different. Figure 7 plots the posterior means of weight vectors ($\mathbb{E}[w_{kl}]$) incident onto a unit. Unsurprisingly, the BNN with Gaussian priors does not exhibit shrinkage towards zero. In contrast, models employing the horseshoe prior are able to prune units away by setting all incident weights to tiny values. It is interesting to note that even for this highly stylized example the centered parameterization struggles to recover the true structure of the underlying network. In contrast, the non-centered parameterization prunes away all but two units. The effect remains even with 100 units in the hidden layer (Figure 18) in Appendix C.2.

6.3. Evaluation of HS-BNN Performance on Real Data Sets

In Section 6.2, we demonstrated the basic properties of the HS BNN and the importance of the non-centered parameterization. We now turn to the evaluation of the HS BNN on real data sets. In the experiments below, we use the fully factorized variational approximation, unless otherwise stated; we will return to comparisons between different variational approximations in Section 6.5.

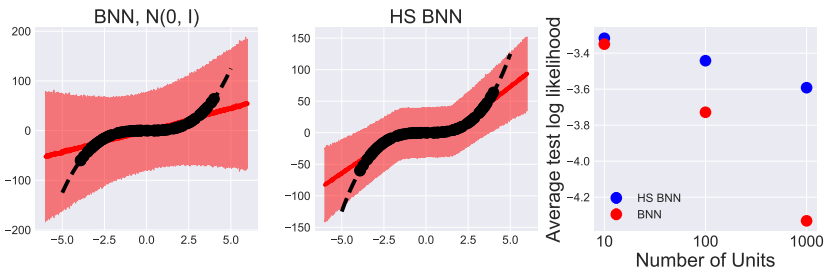


Figure 6: **Noisy polynomial regression** $y_n = x_n^3 + \epsilon_n$, $\epsilon_n \sim \mathcal{N}(0, 9)$. The crosses indicate training data and the dotted line visualizes the noise free cubic function, and in red we visualize the predicted means and $\pm$ 3 standard deviations for single layer networks with 1000 hidden units. Right: Average predictive log likelihood for single layer networks with 10, 100 and 1000 units. The error bars are smaller than the size of the plotted markers. HS-BNN improves upon BNN and the held out log likelihoods deteriorate more slowly with increasing model capacity. In Figure 15 we will see that employing a structured variational approximation leads to even better fits with held out likelihoods no longer deteriorating with extra capacity.

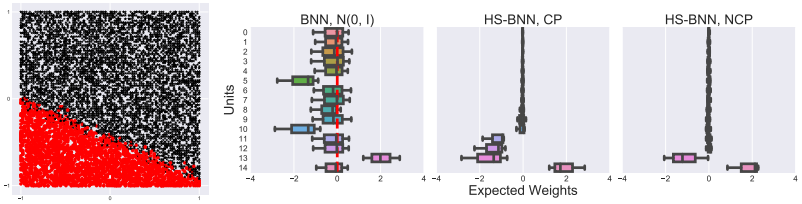


Figure 7: **Non-centered parameterization** is essential for robust inference. From left to right, A synthetic nearly linear classification problem generated from sampling a 2-2-1 network, the two classes are in red and black. Expected weights inferred with a Bayesian neural network with Gaussian priors on weights. Expected weights recovered with a centered horseshoe parameterization. Non-centered horseshoe parameterization. The box-plots display $E_q[w_{kl}]$.

6.3.1. Classification with MNIST: HS-BNN learns sparse models.

We preprocessed the images in the MNIST digits data set by dividing the pixel values with 126. We explored networks with varying widths and depths all employing rectified linear units. For HS-BNN we used Adam with a learning rate of 0.005 and 500 epochs. We found that annealing in the entropy (Sønderby et al., 2016) and cross-entropy terms of the ELBO

as well as scaling the variances of the variational posterior of the weights by the number of incoming nodes improved predictive performance. We did not use a validation set to monitor performance or tune hyper-parameters.

We compare HS-BNN against modern approaches for variational learning of neural networks — variational matrix Gaussian (VMG) (Louizos and Welling, 2016), a BNN with a two-component scale mixture (SM-BNN) prior on weights proposed by Blundell et al. (2015) and a BNN with Gaussian prior (BNN) on weights. VMG uses a structured variational approximation, while the other approaches use fully factorized approximations and differ only in the type of prior used. We trained all models using the parameter settings recommended in the original papers for the competing methods.

Figure 8 summarizes our findings. We showcase results for three architectures with two hidden layers each containing 400, 800 and 1200 rectified linear hidden units. Across architectures, we find our performance to be significantly better than BNN, comparable to SM-BNN, and worse than VMG. More interestingly, we clearly see the strong shrinkage provided by the horseshoe prior. Recall that under the horseshoe prior, $w_{kl} \sim \mathcal{N}(0, \tau_{kl}^2 \upsilon_l^2 \mathbb{I})$. As the scales $\tau_{kl}\upsilon_l$ tend to zero the corresponding units (and all incident weights) are pruned away. SM-BNN also encourages shrinkage to zero, but on weights not nodes. Further, the horseshoe prior with its thicker tails and taller spike at origin encourages stronger shrinkage. To see this we compared the ℓ_2-norms of the inferred expected weight node vectors $\mathbb{E}[w_{kl}]$ found by SM-BNN and HS-BNN. For HS-BNN the inferred scales are small for most units, with a few notable outliers that escape un-shrunk. This causes the corresponding weight vectors to be zero for the majority of units, suggesting that the model is able to effectively "turn off" extra capacity. In contrast, the weight node vectors recovered by SM-BNN (and BNN) are less tightly concentrated at zero. We also plot the density of $\mathbb{E}[w_{kl}]$ with the smallest norm in each of the three architectures. Note that with increasing architecture size (modeling capacity) the density peaks more strongly at zero, suggesting that the model is more confident in turning off the unit and not use the extra modeling capacity.

To further explore the implications of unit versus weight shrinkage, we visualize $\mathbb{E}[w_{kl}]$ learned by SM-BNN and HS-BNN in Figure 8. Weight shrinkage in SM-BNN encourages fundamentally different filters that pick up edges at different orientations. In contrast, HS-BNN's node shrinkage encourages filters that correspond to digits or superpositions of digits and may lead to more interpretable networks. Stronger shrinkage afforded by the horseshoe is again evident when visualizing filters with the lowest norms. HS-BNN filters are nearly all black when scaled with respect to the SM-BNN filters. Pruning away entire units allows for significant test-time speedups relying only on standard dense matrix operations.

Finally, in Figure 9, we provide additional results that illustrate the model selection abilities of HS-BNN. First we visualize the norms of inferred node weight vectors $\mathbb{E}[w_{kl}]$ found by BNN, SM-BNN and HS-BNN for $400-400$, $800-800$ and $1200-1200$ networks (Figure 9). Note that as we increase capacity the model selection abilities of HS-BNN becomes more obvious and as opposed to the other approaches illustrate clear inflection points and it is evident that the model is using only a fraction of its available capacity. Next, we visualize the density of the inferred node weight vectors $\mathbb{E}[w_{kl}]$ under the two models for networks $400-400$, $800-800$ and $1200-1200$. For each network we show the density of the five units with the smallest norms from either layer. Note that in all three cases HS-BNN produces weights that are more tightly concentrated around zero. Moreover for HS-BNN

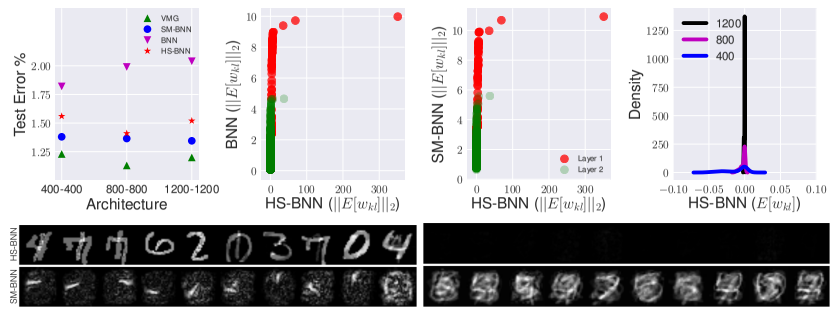


Figure 8: **MNIST Results**. TOP: *From left to right*, Test error rates for different architectures and methods. The center two plots compare the shrinkage properties of the solutions found by HS-BNN, SM-BNN and BNN. For the 1200-1200 network, we compare the expected node weight vectors inferred under the different models. We sort the recovered weight vectors $\mathbb{E}[w_{kl}]$ based on their 2-norm and compare them via scatter plots. Each circle corresponds to one of the 1200 weight node vectors. Compared to competing methods a large number of weight node vectors are zeroed out, while a small number escapes un-shrunk for HS-BNN. The rightmost plot shows the density of the unit with the lowest norm from the the three architectures. BOTTOM: $\mathbb{E}[w_{kl}]$ for the first layer. The left and right columns visualize the ten units with the largest and the smallest norms.

the concentration around zero becomes sharper with increasing modeling capacity (larger architectures), again indicating that we are pruning away additional capacity.

6.3.2. Classification for Gesture Recognition: HS-BNNs again achieve shrinkage while retaining strong predictive performance.

We also experimented with a gesture recognition data set (Song et al., 2011) that consists of 24 unique aircraft handling signals performed by 20 different subjects, each for 20 repetitions. The task consists of recognizing these gestures from kinematic, tracking and video data. However, we only use kinematic and tracking data. A couple of example gestures are visualized in Figure 10. The data set contains 9600 gesture examples. A 12-dimensional vector of body features (angular joint velocities for the right and left elbows and wrists), as well as an 8 dimensional vector of hand features (probability values for hand shapes for the left and right hands) collected by Song et al. (2011) are provided as features for all frames of all videos in the data set. We additionally used the 20 dimensional per-frame tracking features made available by Song et al. (2011). We constructed features to represent each gesture by first extracting frames by sampling uniformly in time and then concatenating the per-frame features of the selected frames to produce 600-dimensional feature vectors.

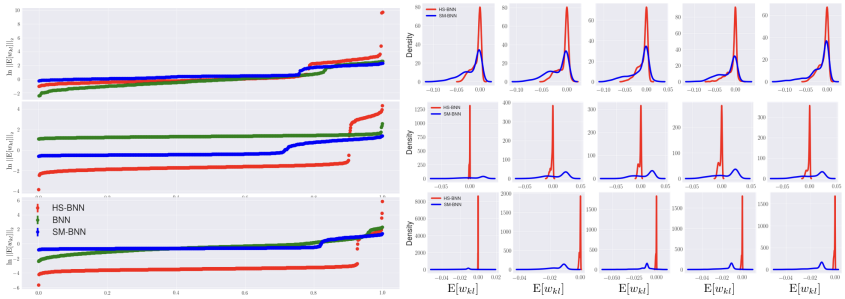


Figure 9: **Model selection in HS-BNN.** *Left*: From top to bottom, we plot log $||\mathbb{E}[w_{kl}]||_2$ for the first layer of a network with 400, 800 and 1200 units. The x-axis has been normalized by the number of units. Further exploration of sparse solutions found by HS-BNN. *Right*: Here we provide density plots for the five smallest (across all layers) node weight vectors w_{kl} found by HS-BNN and SM-BNN for the 400-400 (top row), 800-800 (middle row), 1200-1200 (bottom row) network. Plots are sorted by 2-norm of w_{kl}, from left to right.

This is a much smaller data set than MNIST and recent work (Joshi et al., 2017) has demonstrated that a BNN with Gaussian priors performs well on this task. Figure 10 compares the performance of HS-BNN with competing methods. We train a two layer HS-BNN with each layer containing 400 units. The error rates reported are a result of averaging over five random 75/25 splits of the data. Similar to MNIST, HS-BNN significantly outperforms BNN and is competitive with VMG and SM-BNN and provides strong shrinkage.

6.3.3. Regression on UCI data sets: horseshoe priors continue to learn smaller models with comparable accuracy.

We next apply our HS-BNN to various standard UCI regression data sets. We follow the experimental protocol proposed in (Hernández-Lobato and Adams, 2015) and train a single hidden layer network with 50 rectified linear units for all but the larger "Protein" and "Year" data sets for which we train a 100 unit network. For the smaller data sets we train on a randomly subsampled 90% subset and evaluate on the remainder and repeat this process 20 times. For "Protein" we perform five replications and for "Year" we evaluate on a single split. We compare against VMG with 10 inducing points. Figure 11 summarizes our predictive results and Figure 19 in Appendix C.2 illustrates the shrinkage towards zero exhibited by the recovered solutions. We observe that while HS-BNN is competitive with VMG on the larger data sets, it is outperformed on the smaller ones. In the next section, we describe how the additional regularization of the regularized horseshoe BNN prior provides uniformly better performance.

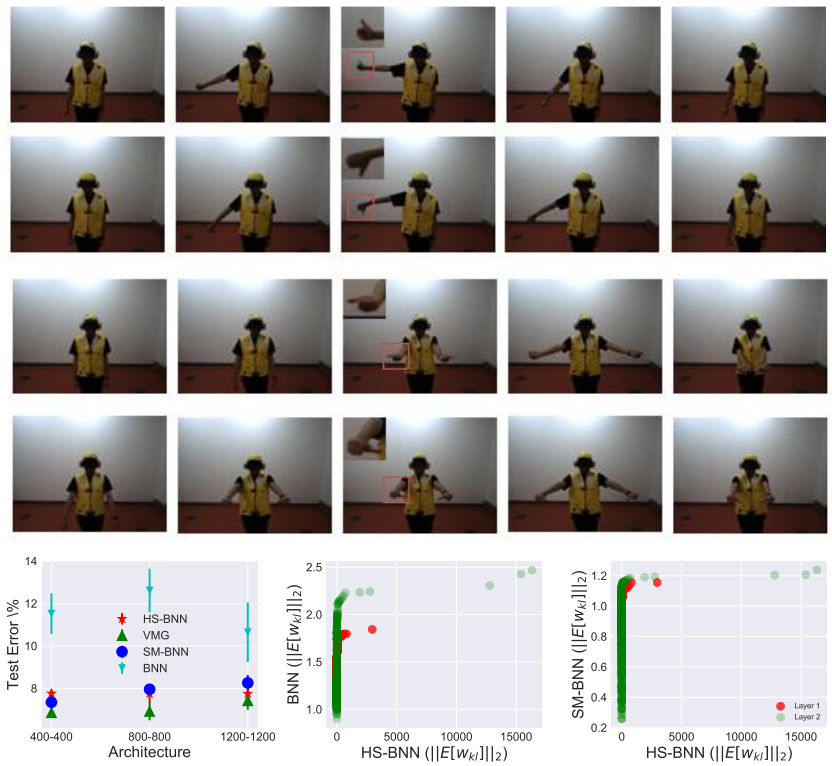


Figure 10: **Gesture recognition**. Top: Example gestures from the NATOPS data set. Bottom: *Left*: Test error rates averaged over 5 runs achieved by competing methods. *Right*: Scatter plots of solutions recovered by HS-BNN and competing methods for the 800-800 architecture on one of the five splits.

6.4. Model Innovation: Regularized horseshoe Priors provide consistent benefits, especially on smaller data sets.

Above, we noted that BNNs learned with standard horseshoe priors (HS) can have higher variance with smaller data sets, resulting in lower predictive performance. The regularized Horsehoe prior (reg-HS) should alleviate this issue. We applied the regularized horseshoe prior BNN to the same collection of UCI data sets, with inverse Gamma hyper-parameters

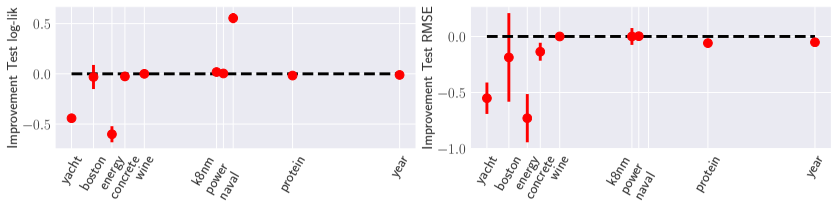


Figure 11: **Regression performance** on a collection of data sets from the UCI repository. Data sets are arranged from left to right in order of increasing number of data points and plotted on the log scale. Plots display relative improvements in test log likelihood and root mean squared error ($\pm$ 1 standard deviation) averaged over 20 random splits of the data. Relative improvement is defined as $(x - y)/max(|x|, |y|)$. VMG performs better on smaller data sets and comparably on larger ones.

$c_a = 2$ and $c_b = 6$ for both the regularized and standard HS-BNN. To focus on the effect of the model, we employed the same factorized variational approximating family for both models. Figure 12 shows that the regularized horseshoe leads to consistent improvements in predictive performance. As expected, the gains are more prominent for the smaller data sets for which the regularization afforded by the regularized horseshoe helps avoid over-fitting.

6.5. Inference Innovations: Value of Various Variational Approximations

All of our previous results focused on using fully-factorized approximating families. However, as we noted in Section 4.1, there are many options for adding structure as well as reducing the number of parameters. These choices can have a large effect: in our previous results with comparisons with standard BNNs, the type of approximating family—e.g. the MVG vs. the fully-factored—had a very large effect on performance. In this section, we explore the properties of different variational approximations. All experiments used a batch size of 128, and $b_g = 10^{-5}$.

Tied approximation speeds up training. We begin by first comparing the tied and fully factorized approximations on the NATOPS gesture recognition data set, with both models employing un-regularized horseshoe priors. Figure 13 contrasts the held-out test performance of the full and tied approximations against wall-clock time.[1] Since we have far fewer variational parameters to learn, the tied approximation converges faster nearly halving the training time with only small deterioration in predictive performance for classification problems; thus one could also imagine using the tied approximation initially and then switching over to one with more parameters.

Structured variational approximations provide stronger shrinkage. In preliminary experiments, we found that of the various approximations described in Section 4.1, the structured approximation outperformed the semi-structured variant. Thus, we focus on a presenting a

1. Experiments were performed on a 2.5 GHz Intel Core i7, with 16GB of RAM.

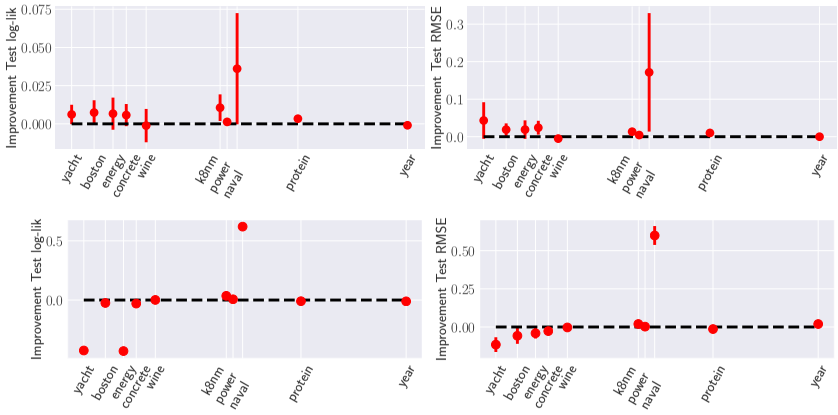


Figure 12: **Regularized horseshoe** results in consistent improvements (TOP) over the vanilla horseshoe prior (values over 0 indicate improvements). The data sets are sorted according to the number of data instances and plotted on the log scale, with 'yacht' being the smallest and 'year' being the largest. Relative improvement is defined as $(x-y)/max(|x|, |y|)$. (BOTTOM) Regularized horseshoe performs similarly to VMG, better on some data sets and worse on others.

detailed comparison between models employing the structured and (fully) factorized variational families. Furthermore, since we have already seen reg-HS outperform vanilla HS, we use regularized horseshoe priors in all subsequent experiments.

Shrinkage on Synthetic Examples. In Figures 14 and 15, we explore the effects of structured and factorized variational approximations on predictive uncertainties. We return to the sinusoid prediction problem from Figure 1. We compare single layer 1000 unit BNNs with regularized horseshoe priors employing factorized and structured variational approximations. The structured approximation best alleviates the under-fitting issues and leads to improved fits. We also revisit the noisy polynomial regression example from Secion 6.2. In Figure 15, we find that consistent with the noisy sinusoid example structured approximation results in tighter uncertainties. Consequently, predictive likelihoods do not degrade as more units are added (right panel).

Shrinkage on UCI benchmarks. We return to the UCI benchmark to carefully vet the different variational approximations. We deviate from prior work, by using networks with significantly more capacity than previously considered for this benchmark. In particular, we use single layer networks with an order of magnitude more hidden units (500) than considered in previous work (50). This additional capacity is more than that needed to explain the UCI benchmark data sets well. With this experimental setup, we are able to

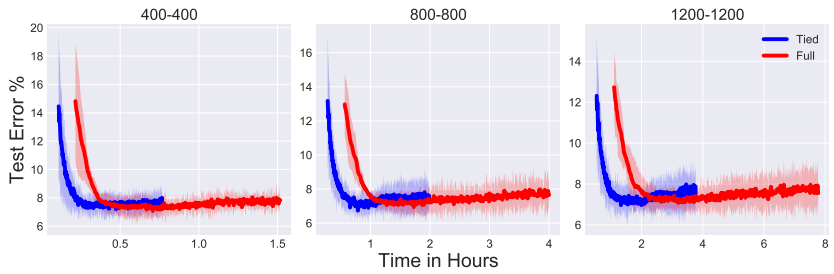


Figure 13: **Tied variational approximation.** We contrast the test accuracy against training time for the tied and full approximations, averaged over five random splits, and run for 500 epochs each on the NATOPS gesture recognition data set.

evaluate how well the proposed methods perform at pruning away extra modeling capacity. For all but the 'year' data set, we report results from 5 trials each trained on a random 90/10 split of the data. For the large year data set, we ran a single trial.

In Figure 16, we see that the factorized and structured variational approximations have similar predictive performance. However, the structured approximation consistently recovers solutions that exhibit much stronger shrinkage towards zero. We have plotted the 50 units with the smallest $||w_{kl}||_2$ weight norms recovered by the factorized and structured approximations, from five random trials. Both approximations provide shrinkage towards zero, but the structured approximation has significantly stronger shrinkage. Further, the degree of shrinkage from the factorized approximation varies significantly between random initializations. In contrast, the structured approximation *consistently* provides strong shrinkage. We compare the shrinkage provided by the two approximations using $||\mathbb{E}[w_{kl}]||_2$ instead of applying the pruning rule from section 4.3 and comparing the resulting compression rates. This is because although the scales $\tau_{kl}\upsilon_l$ inferred by the factorized approximation provide a clear separation between signal and noise, they do not exhibit shrinkage toward zero. On the other hand, $w_{kl} = \tau_{kl}\upsilon_l\beta_{kl}$ does exhibit shrinkage and provides a fair comparison.

Prediction and shrinkage comparison against competing methods. Finally, we compare the reg-HS model with structured variational approximation against the variational matrix Gaussian (VMG) approach of Louizos and Welling (2016), which has previously been shown to outperform other variational approaches to learning BNNs. We used the pruning rule with $\delta = 10^{-3}$ for all but the 'year' data set, for which we set $\delta = 10^{-5}$. Figure 16 demonstrates that structured reg-HS is competitive with VMG in terms of predictive performance. We either perform similarly or better than VMG on the majority of the data sets. More interestingly, structured reg-HS achieves competitive performance while pruning away excess capacity and achieving significant compression. We also experimented with fine-tuning the pruned model by updating the weight means while holding others fixed. Figure 16 reports results from running 20 such fine-tuning epochs. We did not find fine-tuning to

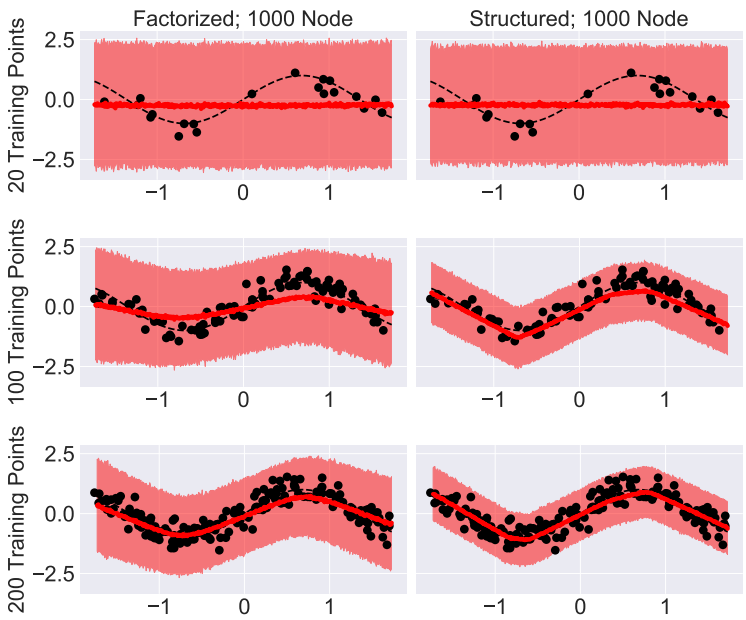


Figure 14: **Regularized horseshoe BNNs** prune away excess capacity and are more resistant to underfitting. Variational approximations aware of model structure improve fits.

significantly improve generalization performance. Additional experiments employing multi-layer architectures are available in the appendix (Figure 23).

6.6. Assessing Calibration: HS-BNN with structured variational approximations are better calibrated

The consistency between the predictive distribution and the empirical frequency of observations is referred to as calibration. When the two match well, the predictive model is deemed well-calibrated. Here, we assess calibration of BNN variants on the synthetic data presented in Figures 1, 5, and 15. Knowing the true data generating process, we sample an additional thousand data points unobserved during training. We use this validation set to assess model calibration and compare a BNN with $\mathcal{N}(0, 1)$ priors, against the regularized HS-BNN employing both the structured and factorized variational approximations. Since calibration is only a meaningful metric provided that the fits are good, we assess performance of the

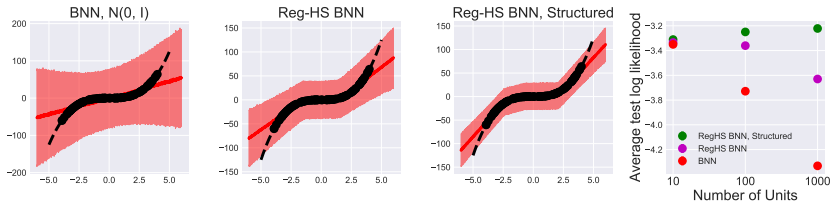


Figure 15: **Structured variational approximations** improve model fits. The crosses indicate training data and the dotted line visualizes the noise free cubic function, and in red we visualize the predicted means and ± 3 standard deviations for single layer networks with 1000 hidden units. Right: Average predictive log likelihood for single layer networks with 10, 100 and 1000 units. Regularized HS-BNN with structured variational approximation provides robust performance in the presence of model misspecification.

models trained on two hundred data points. With these many training instance, all three models achieve a reasonable fit to the data.

Calibrations of continuous responses are typically assessed using the predictor's predictive cumulative distribution functions (CDF). Before presenting our results, we briefly review the definitions of the two types of calibration we consider here and refer the interested reader to (Gneiting et al., 2007) for a more thorough treatment of the material.

Let n index the data $(\{x_n, y_n\}_{n=1}^N)$ in the validation set, F_n be the predictive CDF, i.e., CDF of the predictive distribution $p(y_n \mid x_n) = \int p(y_n \mid x_n, \mathcal{W})p(\mathcal{W} \mid \mathcal{D}_{train})d\mathcal{W}$. In our experiments we used the empirical distribution function of Monte Carlo samples to approximate F_n.[2] Let G_n be the (unobserved) true data generating process. We say that $\{p_n = F_n(y_n)\}_{n=1}^N$ is probabilistically calibrated with respect $\{G_n\}_{n=1}^N$, iff $\frac{1}{N}\sum_{n=1}^N \mathbf{1}(p_n \leq p)$ converges almost surely to p for all $p \in [0, 1]$. The definition can be operationalized by selecting a sufficiently large N, a set of discrete thresholds (p_t) in $[0, 1]$ and computing the fraction of validation instances below the threshold and comparing it to the threshold itself. This is illustrated in Figure 17, where we plot the thresholds p_t (expected confidence level) against $\hat{p}_t = \frac{1}{N}\sum_{n=1}^N \mathbf{1}(p_n \leq p_t)$ (empirical confidence level). Plots closer to the diagonal indicate better calibration. Probabilistic calibration has also been recently used to assess the calibration of deep regression models (Kuleshov et al., 2018). They also proposed a quantitative metric for measuring miscalibration, $\propto \sum_{t=1}^T (p_t - \hat{p}_t)^2$. Although widely used probabilistic calibration is a necessary but not sufficient criteria for calibration (Gneiting et al., 2007; Hamill, 2001). We assess the models according to an additional mode of calibration — marginal calibration, to further lend credence to whether a particular model is indeed well calibrated.

Marginal calibration is said to hold iff the empirical distribution function of the validation set, $\hat{G}(y) = \frac{1}{N}\sum_{n=1}^N \mathbf{1}(y_n < y)$, $y \in \mathbb{R}$ converges almost surely to the average

2. A moment matched Gaussian CDF produced similar results.

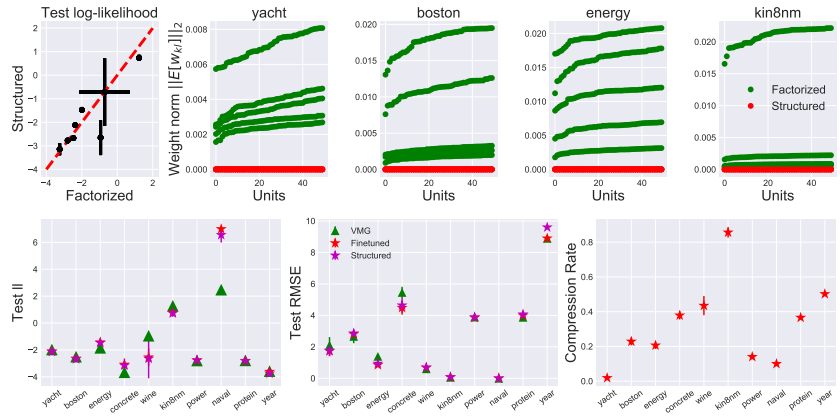


Figure 16: *Top*: **Structured variational approximations** result in similar predictive performance but consistently recover solutions that exhibit stronger shrinkage. The left most figure plots the predictive log likelihoods achieved by the two approximations, each point corresponds to a UCI data set. We also plot the fifty units with the smallest $||E[w_{kl}]||_2$, on a number of data sets. Each point in the plot displays the inferred $||E[w_{kl}]||_2$ for a unit in the network. We plot recovered expected weight norms from all five random trials for both the factorized and structured approximation. The structured approximation (in red) consistently provides stronger shrinkage. The factorized approximation both produces weaker shrinkage and the degree of shrinkage exhibits higher variance with random trials. *Bottom*: The structured approximation is competitive with VMG while using much smaller networks. Fine tuning leads to small improvements. Compression rates are defined as the fraction of un-pruned units.

predictive CDF $\bar{F}(y) = \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^N F_n(y)$ for all $y \in \mathbb{R}$ as $N \to \infty$. The definition can be approximated by selecting a sufficiently large validation set and comparing the empirical distribution function computed from the validation set and $\bar{F}$ both evaluated at a common set of selected thresholds y_t. The empirical distribution function and average predictive CDF are compared in Figure 17. Plots closer to the diagonal again indicate better calibration. We can again quantify the calibration error by computing the error $\propto \sum_{t=1}^T (\hat{G}(y_t) - \bar{F}(y_t))^2$. We define overall calibration error as the arithmetic mean,

$$\text{calibration error} = \frac{1}{2}\left(\frac{1}{T_1}\sum_{t=1}^{T_1}(p_t - \hat{p}_t)^2 + \frac{1}{T_2}\sum_{t=1}^{T_2}(\hat{G}(y_t) - \bar{F}(y_t))^2\right)$$

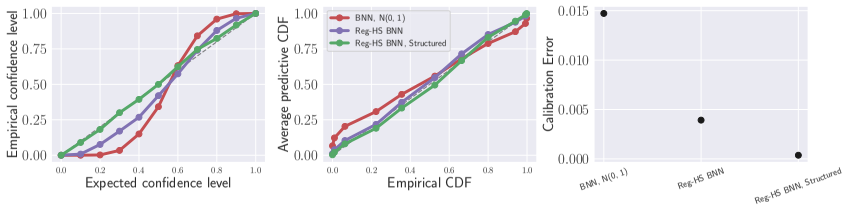


Figure 17: **Calibration assessments**. *Left to Right*: Probabilistic calibration, Marginal calibration, and calibration error. For the first two, plots closer to the diagonal indicate better calibration.

6.7. HS-BNNs Application: Reinforcement Learning

One application area in which having good predictive uncertainty estimates is crucial is in model-based reinforcement learning scenarios (e.g. (Depeweg et al., 2017; Gal et al., 2016b; Killian et al., 2017)): here, it is essential not only to have an estimate of what state an agent may be in after taking a particular action, but also an accurate sense of all the states the agent may end up in. In the following, we apply our regularized HS-BNN with structured approximations to two domains: the 2D map of Killian et al. (2017) and acrobot Sutton and Barto (1998). The goal of the 2D map was to traverse from a start region to a goal region around an obstacle and "wind" or slippery conditions. The state space ($S \in R^2$) of 2D map consists of x-axis and y-axis and at each step, the agent can choose actions from directions N, E, S, W.

The goal of the acrobot was to swing up an under-actuated double pendulum where the goal is to swing it from hanging down to fully up. The state space ($S \in R^4$) consists of two angles (the angle between the pendulum and the ceil and the angle of the joint), the angular velocities of two links and there are three possible actions. The details of the dynamics settings of both domains are described in (Killian et al., 2017).

In each domain, we first collected data by training a DDQN (2 layers of width 256, 512; learning rate $5e-4$) online (updated every episode) and an epsilon-greedy policy that started at 1 and decayed to 0.15 with decay rate 0.995. That is, we started with a randomly initialized DDQN, ran it with a completely random policy (epsilon=1), updated the parameters of the DDQN to estimate the state-action values based on the data. We repeated the process over 500 episodes, gradually decaying epsilon to have the agent converge to the optimal policy. (2 layers of width 256, 512; learning rate $5e-4$). Overall, this procedure ensured that we had a wide variety of samples that were still biased in coverage toward the optimal policy.

While this procedure allowed us to generate a large amount of data around the optimal policy, our goal was learning in resource-constrained settings. Thus, we sampled 1% data randomly from the large cohort. We used these data to train a transition function to predict the next position of the agent given the current position and action. Using a BNN to model the transition function allowed us to capture uncertainty due to limited data, as well as

Table 2: Model-based reinforcement learning. The under-fitting of the standard BNN results in lower task performance, whereas the HS-BNN is more robust to this underfitting.

	2D Map	
	Test RMSE	Avg. Reward
BNN x-500-y	0.187	975.386
BNN x-100-100-y	0.089	966.716
Structured x-500-y	**0.058**	**995.416**
Structured x-100-100-y	0.061	992.893
	Acrobot	
BNN x-500-y	0.924	-156.573
BNN x-100-100-y	0.710	-23.419
Structured x-500-y	**0.558**	-108.443
Structured x-100-100-y	0.656	**-17.530**

stochasticity. For the transition function, we considered two architectures, a single hidden layer network with 500 units, and a two layer network with 100 units per layer as the transition function for each domain. We used learning rate 0.0002, and batch size 32, and trained the network for 2000 episodes. We considered two priors, the standard normal on the weights and our HS-BNN.

Finally, once we had trained an approximate transition function to model the domain, we used that function as a simulator. Now, instead of training a DDQN based on data from the real environment, we trained with data generated from the BNN. Once that policy learning converged, we tested that policy—learned from the approximate BNN transition model—on the original simulator to measure its performance in the true setting.

HS-BNNs improve reinforcement learning performance. As in our prediction results, training a moderately-sized BNN with so few data results in severe underfitting, which in turn, adversely affects the quality of the policy that is learned. That is, if we believe we have a lot of uncertainty about the effect of an action—due to underfitting—we cannot find an appropriate policy. We see in table 2 that the better fitting of the structured reg-HS-BNN results in higher task performance, across domains and model architectures.

7. Discussion and Conclusion

Model selection in Bayesian networks is an important problem. In this work, we demonstrated that a properly parameterized horseshoe prior on the scales of the weights incident to each node is a computationally efficient tool for model selection in Bayesian neural networks. Decomposing the horseshoe prior into inverse gamma distributions led to computationally convenient inference algorithms. However, auxiliary variables introduced by the decomposition can pose further inferential challenges and lead to poorer approximations (Neville et al., 2014). We found that using a non-centered representation ensured a degree of robustness to such issues as well as poor local optima. Designing efficient and ac-

curate inference algorithms as well as metrics for reliably measuring the quality of posterior approximations for BNNs and Horseshoe BNNs remains an exciting direction for followup work. We also showed that the regularized horseshoe prior, combined with a structured variational approximating distribution, provides a computationally efficient tool for model selection in Bayesian neural networks. By retaining crucial posterior dependencies, the structured approximation provided strong shrinkage while being competitive in predictive performance to approaches without shrinkage.

All that said, one might wonder about alternatives for model selection. For example, one could place a simple exponential prior on the weight scale to encourage all incident weights to be zero, but without heavy tails all scales will be forced to be artificially low and prediction will suffer (Molchanov et al., 2017; Wen et al., 2016). In contrast, simply using a heavy-tail prior on the scale parameter, such as a half-Cauchy, will not apply any pressure toward setting small scales to zero. Another alternative is to observe that a node can be pruned if the product $z \cdot w$ is nearly constant for all inputs z—having small weights is sufficient to achieve this property; weights w that are orthogonal to the variation in z is another. Thus, instead of putting a prior over the scale of w, one could put a prior over the scale of the variation in $z \cdot w$. However, in initial experiments, we found that such a formulation has many more local optima and thus harder to optimize. In contrast, inference for the horseshoe prior is fairly robust: following the parameterization considerations and choice of variational family, our code is standard BBVI.

Within Horseshoe priors, there remain several interesting follow-on directions, including, modeling enhancements that use layer, node, or even weight specific weight decay c, or layer specific global shrinkage parameter b_g to provide different levels of shrinkage to different parts of the BNN. One could imagine using the machinery of Horseshoe priors to turn off entire layers or entire skip connections; one could imagine these being governed in some way by the input location x. It would also be interesting to consider how to adapt Horseshoe priors to exhibit interesting limiting behaviors as the number of nodes in a layer grow without bound.

More broadly, model selection via Horseshoe priors may represent a sweet spot, at least for now, between (still the most common) fully factorized priors and recent attempts at specifying function-based priors (Wang et al., 2019; Sun et al., 2019). The latter provide an elegant approach for specifying an implicit prior over BNN weights via an explicit prior over the kinds of functions are likely, but currently these approaches are fairly computationally intensive and involve more challenging optimizations. It would be interesting to see if variants of Horseshoe priors can bring us interesting function classes at relatively low computational and inferential expense.

Acknowledgments

JY acknowledges support from NSF RI-1718306. JY and FDV acknowledge support from the IBM Faculty Award. We would also like to thank the anonymous reviewers for their excellent suggestions, Karen Ullrich and Christos Louizos for their comments on an early version of the manuscript.

Appendix A. Algorithm

A.1. Conditional variational pre-activations

Recall from Section 4.2, that the variational pre-activation distribution is given by $q(b \mid a, \nu_l, \phi_{\beta_l}) = \mathcal{N}(b \mid \mu_b, \Sigma_b) = \mathcal{N}(b \mid M^T_{\beta_l|\nu_l} a, (a^T U_{\beta_l|\nu_l} a)V)$, where $U = \Psi + hh'$, and V is diagonal. The Equation requires $M_{\beta_l|\nu_l}$ and $U_{\beta_l|\nu_l}$. The expressions for these follow directly from the properties of partitioned Gaussians.

For a particular layer l, we drop the explicit dependency on l from the notation. Recall that $B = \begin{bmatrix} \beta \\ \nu^T \end{bmatrix}$, and let $B \in \mathbb{R}^{m \times n}$, $\beta \in \mathbb{R}^{m-1 \times n}$, and $\nu \in \mathbb{R}^{n \times 1}$ $q(B \mid \phi_B) = \mathcal{MN}(B \mid M, U, V)$. From properties of the Matrix normal distribution, we know that a column-wise vectorization of B, $\vec{B} \sim \mathcal{N}(\vec{M}, V \otimes U)$. From this and Gaussian marginalization properties it follows that the jth column $t_j = [\beta_j; \nu_j]$ of B is distributed as $t_j \sim \mathcal{N}(m_j, V_{jj}U)$, where m_j is the appropriate column of M. Conditioning on ν_j then yields, $q(\beta_j \mid \nu_j) = \mathcal{N}(\beta_j \mid \mu_{\beta_j|\nu_j}, \Sigma_{\beta_j|\nu_j})$, where

$$\Sigma_{\beta_j|\nu_j} = V_{jj}(\Psi_\beta + \frac{\Psi_\nu}{\Psi_\nu + h_\nu^2} h_\beta h_\beta^T)$$
$$\mu_{\beta_j|\nu_j} = \mu_{\beta_j} + \frac{h_\nu(\nu_j - \mu_{\nu_j})}{\Psi_\nu + h_\nu^2} h_\beta$$

Rearranging, we can see that, $M_{\beta|\nu}$ is made up of the columns $\mu_{\beta_j|\nu_j}$ and $U_{\beta|\nu} = \Psi_\beta + \frac{\Psi_\nu}{\Psi_\nu + h_\nu^2} h_\beta h_\beta^T$.

A.2. Algorithmic details

The ELBO corresponding to the non-centered regularized HS model is,

$$\begin{aligned}
\mathcal{L}(\phi) = {} & \mathbb{E}_{q(c^2)}[\ln \text{Inv-Gamma}(c^2 \mid c_a, c_b)] + \mathbb{E}_{q(\kappa^2)q(\rho_k)}[\ln \text{Inv-Gamma}(\kappa^2 \mid 1/2, 1/\rho_\kappa)] \\
& + \mathbb{E}_{q(\rho_k)}[\ln \text{Inv-Gamma}(\rho_\kappa \mid 1/2, 1/b_\kappa^2)] + \sum_n \mathbb{E}_{q(\beta, \mathcal{T}, \kappa^2, c^2)}[\ln p(y_n \mid \beta, \mathcal{T}, \kappa^2, c^2, x_n)] \\
& + \sum_{l=1}^{L-1} \sum_{k=1}^{K_L} \mathbb{E}_{q(\lambda_{kl})}[\ln \text{Inv-Gamma}(\lambda_{kl} \mid 1/2, 1/b_0^2)] \\
& + \sum_{l=1}^{L-1} \mathbb{E}_{q(\upsilon_l)q(\vartheta_l)}[\ln \text{Inv-Gamma}(\upsilon_l^2 \mid 1/2, 1/\vartheta_l)] + \mathbb{E}_{q(\vartheta_l)}[\ln \text{Inv-Gamma}(\vartheta_l \mid 1/2, 1/b_g^2)] \\
& + \sum_{l=1}^{L-1} \mathbb{E}_{q(B_l)}[\ln \mathcal{N}(\beta_l \mid 0, \mathbb{I})] + \sum_{l=1}^{L-1} \sum_{k=1}^{K_L} \mathbb{E}_{q(B_l)}[\ln \text{Inv-Gamma}(\tau_{kl}^2 \mid 1/2, 1/\lambda_{kl})] \\
& + \sum_{k=1}^{K_L} \mathbb{E}_{q(\beta_{kL})}[\ln \mathcal{N}(\beta_{kL} \mid 0, \mathbb{I})] + \mathbb{H}[q(\theta \mid \phi)].
\end{aligned}$$

We rely on a Monte-Carlo estimates to evaluate the expectation involving the likelihood $\mathbb{E}_{q(\beta, \mathcal{T}, \kappa^2, c^2)}[\ln p(y_n \mid \beta, \mathcal{T}, \kappa^2, c^2, x_n)]$.

Efficient computation of the Matrix Normal Entropy The entropy of $q(B) = \mathcal{MN}(B \mid M, U, V)$ is given by $\frac{mn}{2}\ln(2\pi e) + \frac{1}{2}\ln|V \otimes U|$. We can exploit the structure of U and V to compute this efficiently. We note that $\ln|V \otimes U| = m\ln|V| + n\ln|U|$. Since V is diagonal $\ln|V| = \sum_j \ln V_{jj}$. Using the matrix determinant lemma we can efficiently compute $|U| = (1 + h'\Psi^{-1}h)|\Psi|$. Owing to the diagonal structure of Ψ, computing it's determinant and inverse is particularly efficient.

Fixed point updates The auxiliary variables ρ_κ, ϑ_l and ϑ_l all follow inverse Gamma distributions. Here we derive for λ_{kl}, the others follow analogously. Consider,

$$\begin{aligned}\ln q(\lambda_{kl}) &\propto \mathbb{E}_{-q_{\lambda_{kl}}}[\ln \text{Inv-Gamma}(\tau_{kl}^2 \mid 1/2, 1/\lambda_{kl})] + \mathbb{E}_{-q_{\lambda_{kl}}}[\ln \text{Inv-Gamma}(\lambda_{kl} \mid 1/2, 1/b_0^2)], \\ &\propto (-1/2 - 1/2 - 1)\ln \lambda_{kl} - (\mathbb{E}[1/\tau_{kl}] + 1/b_0^2)(1/\lambda_{kl}),\end{aligned} \tag{10}$$

from which we see that,

$$\begin{aligned} q(\lambda_{kl}) &= \text{Inv-Gamma}(\lambda_{kl} \mid c, d), \\ c = 1, d &= \mathbb{E}[\frac{1}{\tau_{kl}^2}] + \frac{1}{b_0^2}. \end{aligned} \tag{11}$$

Since, $q(\tau_{kl}^2) = \ln \mathcal{N}(\mu_{\tau_{kl}}, \sigma_{\tau_{kl}}^2)$, it follows that $\mathbb{E}[\frac{1}{\tau_{kl}^2}] = \exp\{-\mu_{\tau_{kl}} + 0.5 * \sigma_{\tau_{kl}}^2\}$. We can thus calculate the necessary fixed point updates for λ_{kl} conditioned on $\mu_{\tau_{kl}}$ and $\sigma_{\tau_{kl}}^2$. Our algorithm uses these fixed point updates given estimates of $\mu_{\tau_{kl}}$ and $\sigma_{\tau_{kl}}^2$ after each Adam step.

Appendix B. Useful Properties of the Log-Normal distribution

These results are well known and included here for completeness. See Johnson et al. (1970), or other appropriate text for more details. If $\ln x \sim \mathcal{N}(\mu, \sigma^2)$ then x follows a log-Normal distribution with mean, $\mathbb{E}[x] = e^{\mu + \frac{1}{2}\sigma^2}$ and variance, $\mathbb{E}[x^2] - \mathbb{E}[x]^2 = (e^{\sigma^2} - 1)e^{2\mu + \sigma^2}$.

B.1. Moments

For any real k, the kth moment of x is,

$$\mathbb{E}[x^k] = e^{k\mu + \frac{1}{2}k^2\sigma^2},$$

this immediately leads to the result $\mathbb{E}[x^{-1}] = e^{-\mu + \frac{1}{2}\sigma^2}$, which we used to compute the parameters of the inverse Gamma distribution in Equation 11.

B.2. Closed under multiplication

If $x_a \sim \text{log-Normal}(\mu_a, \sigma_a^2)$ and $x_b \sim \text{log-Normal}(\mu_b, \sigma_b^2)$, then,

$$y = x_a x_b \sim \text{log-Normal}(\mu_a + \mu_b, \sigma_a^2 + \sigma_b^2),$$

a fact we exploited for computing with the pruning rule of Section 4.3.

Appendix C. Experiments

C.1. Experimental details

For regression problems we use Gaussian likelihoods with an unknown precision γ, $p(y_n \mid f(\mathcal{W}, x_n), \gamma) = \mathcal{N}(y_n \mid f(\mathcal{W}, x_n), \gamma^{-1})$. We place a vague prior on the precision,$\gamma \sim \text{Gamma}(6, 6)$ and approximate the posterior over γ using another variational distribution $q(\gamma \mid \phi_\gamma)$. The corresponding variational parameters are learned via a gradient update during learning.

Classification Experiments For VMG, we used 150 inducing points, a batch size of 128, and a learning rate of 0.001.

Regression Experiments For comparing the reg-HS and HS models we followed the protocol of (Hernandez-Lobato & Adams, 2015) and trained a single hidden layer network with 50 rectified linear units for all but the larger "Protein" and "Year" data sets for which we train a 100 unit network. For the smaller data sets we train on a randomly subsampled 90% subset and evaluate on the remainder and repeat this process 20 times. For "Protein" we perform 5 replications and for "Year" we evaluate on a single split. For, VMG we used 10 pseudo-inputs, a learning rate of 0.001 and a batch size of 128.

The details of the different UCI data sets used are presented in Table 3

data set	N	D
Yacht	308	6
Boston	506	13
Energy	768	8
Concrete	1030	8
Wine	1599	11
Kin8nm	8192	8
Power Plant	9568	4
Naval	11,934	16
Protein	45,730	9
Year	515,345	90

Table 3: UCI Regression results. HS-BNN and VMG are compared.

C.2. Additional Experiments

C.2.1. Non-centered parameterization

In Figure 18 we display that the non-centered parameterization is able to recover the true network even under large model misspecification.

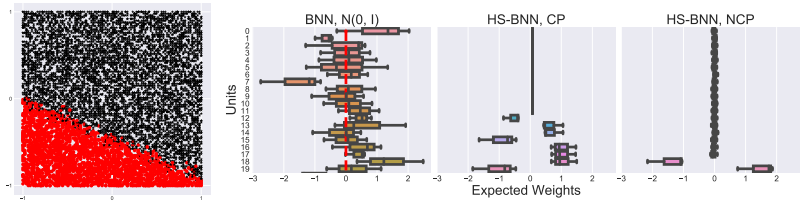


Figure 18: **Non-centered parameterization** is essential for robust inference. From left to right, A synthetic nearly linear classification problem generated from sampling a 2-2-1 network, the two classes are in red and black. Expected weights inferred with a 100 unit Bayesian neural network with Gaussian priors on weights. Expected weights recovered with a 100 unit centered horseshoe parameterization and a 100 unit Non-centered horseshoe parameterization. Only the 20 units with the largest weights are displayed.

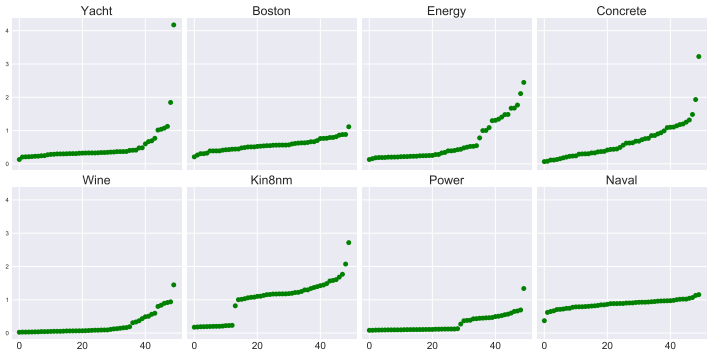


Figure 19: We plot $||w_{kl}||_2$ recovered by the HS-BNN using the fully factorized variational approximation on a number of UCI data sets.

C.2.2. Shrinkage provided by the fully factorized approximation on UCI benchmarks

Figure 19 illustrates the shrinkage afforded by 50 unit HS-BNNs using fully factorized approximations. Note that on some data sets, we do not achieve much compression and all 50 units are used. This is a consequence of the fully factorized approximations providing weaker shrinkage as well as 50 units not being large enough to model the complexity of the data set.

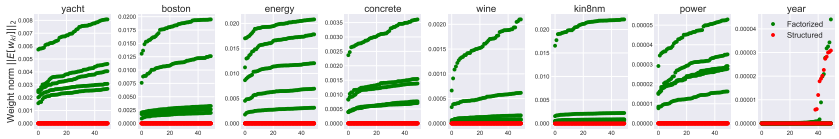


Figure 20: Structured variational approximations consistently recover solutions that exhibit stronger shrinkage. We plot the 50 smallest $||w_{kl}||_2$ recovered by the two approximations on five random trials on a number of UCI data sets.

C.2.3. Scaled Priors and Structured Inference

Here we further explore under-fitting issues in BNNs *not* employing horseshoe priors. As pointed out in the main text, using $\mathcal{N}(0, \sigma^2)$ priors over the network weights encourages the prior predictive variance to increase with the width of the network. It is well known (Neal, 1997) that instead scaling the prior variance with the width of the layer $\mathcal{N}(0, \sigma^2/H)$ does not result in such pathologies and one instead recovers a Gaussian process in the infinite width limit. A natural question then is whether such a scaled prior alleviates the under-fitting issues observed in Figure 1. We find (Figure 21) that this is not the case when employing a fully factorized variational approximation. We find that the posterior again tends to the prior just as it did for BNNs with standard normal priors. Since, here we are using one dimensional inputs the input to hidden layer prior is still a $\mathcal{N}(0, 1)$ distribution while the hidden to output layer employs a $\mathcal{N}(0, 1/1000)$ distribution. Empirically, we find that the higher variance input to hidden layer compensates for the smaller variance output layer prior and cause high predictive uncertainties. A heuristic, sometimes employed, is to divide the input to each layer by the square root of its dimensionality during the variational forward pass. This addresses the high predictive variance issue but not the underfitting problem, resulting in similar poor fits but with lower predictive variances. These observations suggest that in addition to appropriate priors better variational approximations are necessary to avoid pathologies. We further corroborated this hypothesis by instead using layer-wise matrix variate Gaussians (Louizos and Welling, 2016) as our variational approximation. We observe that even with $\mathcal{N}(0, \sigma^2)$ priors improved fits are obtained with approximations that retain more of the correlations among the weights. As illustrated in the main text similar conclusions also hold for HS-BNN — structured inference leads to higher test likelihoods (Figure 16) and better calibrated uncertainties (Figures 14, 17).

C.2.4. Comparisons against Hamiltonian Monte Carlo

Retaining model structure in the variational approximations do typically lead to more accurate but nonetheless biased inferences. Hamiltonian Monte Carlo (Neal, 1993; Duane et al., 1987) on the other hand provides accurate and asymptotically exact inferences and is widely regarded as the gold standard for inference in Bayesian neural networks. They do however have difficulty scaling with the number of data instances. Moreover, the horseshoe is a challenging distribution to sample from, these difficulties are only exacerbated for BNN models with (regularized) horseshoe priors which involve sampling high dimensional poste-

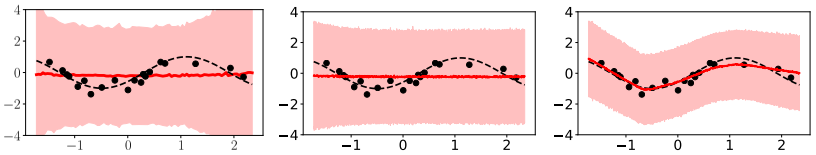


Figure 21: **Model or Inference**. Predictive distributions from a single layer BNN with ReLU activations, containing 1000 units trained on twenty noisy samples from a Sine function. From left to right, we have a BNN with $\mathcal{N}(0, \mathbb{I}/H_{in})$ priors over weights and a fully factorized variational approximation, and BNNs with $\mathcal{N}(0, \mathbb{I})$ priors but employing matrix Gaussian variational distributions (Louizos and Welling, 2016) with one and two pseudo inputs. Similar to Figure 1, underfitting issues can be seen for the scaled priors employing fully factorized variational families. More expressive variational approximations (columns 2 and 3) somewhat alleviate these issues, even for the unscaled $\mathcal{N}(0, \mathbb{I})$ priors.

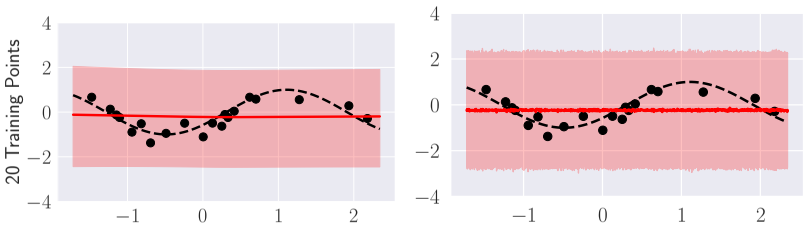


Figure 22: **HMC** (*left*) and **structured VI** (*right*) predictive distributions of a single hidden layer ten unit BNN with a regularized horseshoe prior ($b_g = 10^{-5}$).

riors involving many horseshoe distributions, one for each hidden layer node. Nonetheless, resorting to Riemannian Hamiltonian Monte Carlo (Girolami and Calderhead, 2011) we sampled from the posterior of a small BNN model with a single hidden layer comprising of ten nodes, conditioned on twenty training instances. We used the diagonal *softabs* metric proposed in (Betancourt, 2013). Figure 22 compares the predictive distribution resulting from 1000 samples from HMC against structured VI, with both approaches performing similarly. We emphasize that it is unlikely that structured VI continues to perform similarly to HMC on larger models and larger amounts of data. Developing HMC procedures that reliably sample from larger BNNs employing similar shrinkage priors is an exciting direction of future work.

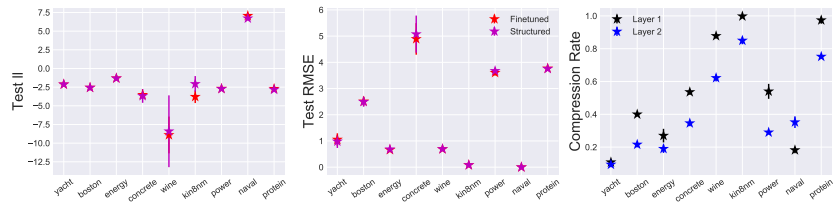


Figure 23: **Structured variational approximations** results for two layer networks each with 100 hidden units. We achieve competitive performance, often improving on the x-500-y networks. For most data sets the second hidden layer is exhibits larger compression. Compression rates are defined as the fraction of un-pruned units.

C.2.5. Additional UCI Regression results

In Figure 23 we present additional regression results for x-100-100-y networks employing regularized horseshoe priors with structured variational approximations. We report results from averaging over five 90/10 random splits of the data set. We use $b_g = 10^{-5}$ and a pruning threshold $\delta = 10^{-3}$. Observe that in addition to competitive predictive performance, we achieve (often significant) compression for both layers. Interestingly, we find that for most data sets our pruning strategy more severely prunes the second layer (closer to the output).

Appendix D. Metrics

For a test point i, with true label y_i^* and posterior predictive samples $\{y_i^1, \ldots, y_i^S\}$ we compute the per data point root mean squared error as,

$$\text{rmse}_i = \sqrt{\frac{1}{S}\sum_{s=1}^{S}(y_i^s - y_i^*)^2},$$

and the average root mean squared error of the test split t with N test points as,

$$\bar{\text{rmse}}_t = \frac{1}{N}\sum_{i=1}^{N}\text{rmse}_i = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{S}\sum_{s=1}^{S}(y_i^s - y_i^*)^2}.$$

The mean and standard deviation of $\bar{\text{rmse}}_t$ across splits $t = 1, \ldots T$ is reported in the text.

We compute test log likelihood for a test point i, $\{y_i^*, x_i^*\}$ given training data $\mathcal{D}$ as,

$$\text{ll}_i = \ln\, p(y_i^* \mid x_i^*, \mathcal{D}) \approx \ln \frac{1}{S}\sum_{s=1}^{S} p(y_i^* \mid x_i^*, \mathcal{W}^s); \quad \mathcal{W}^s \sim q(\mathcal{W} \mid \mathcal{D}).$$

Summaries across data instances and train/test splits are computed and reported analogously to root mean squared error. For split t with N test instances we have,

$$\bar{\mathrm{ll}}_t = \frac{1}{N}\sum_{i=1}^{N} \mathrm{ll}_i = \frac{1}{N}\sum_{i=1}^{N} \ln \frac{1}{S}\sum_{s=1}^{S} p(y_i^* \mid x_i^*, \mathcal{W}^s); \quad \mathcal{W}^s \sim q(\mathcal{W} \mid \mathcal{D}).$$

Appendix E. Sparsity inducing priors.

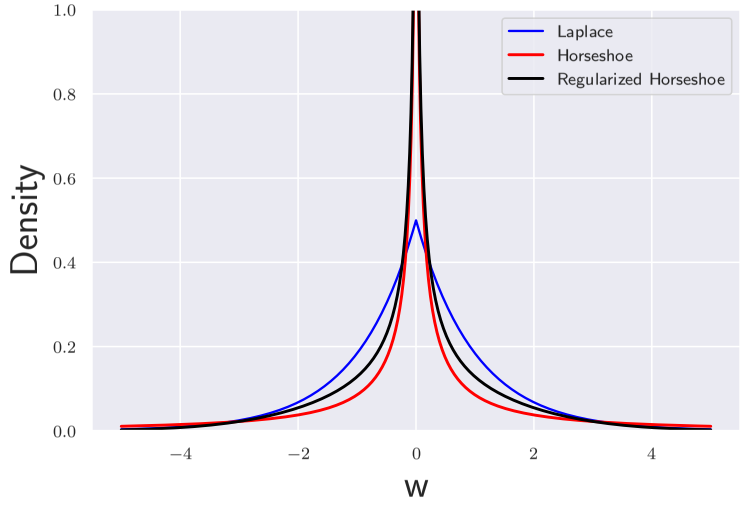


Figure 24: Density for the Laplace distribution ($\frac{1}{2b}e^{-\frac{|x|}{b}}$, with $b = 1$), the horseshoe ($b_0 = b_g = 1$), and the regularized horseshoe distribution ($b_0 = b_g = 1$ and $c = 2$). Both horseshoe variants have significantly larger spikes at zero than the Laplace distribution and the regularized horseshoe has truncated tails.

E.1. Laplace

The Laplace distribution results in the widely used ℓ_1 penalty and leads to sparse MAP solutions. It however exhibits thinner tails and drastically smaller spike at zero than its horseshoe counterparts. See Bhadra et al. (2017) for a more in-depth discussion contrasting horseshoe like distributions against the Laplace.

$$w \mid b \sim \mathrm{Laplace}(w \mid 0, b)$$
$$\mathrm{Laplace}(w \mid 0, b) = \frac{1}{2b}e^{-\frac{|w|}{b}}$$

E.2. Horseshoe

$$w_{kl} \mid \tau_{kl}, \upsilon_l \sim \mathcal{N}(0, (\tau_{kl}^2 \upsilon_l^2)\mathbb{I}),$$
$$\tau_{kl} \sim C^+(0, b_0), \quad \upsilon_l \sim C^+(0, b_g).$$

E.3. Regularized Horseshoe

$$w_{kl} \mid \tau_{kl}, \upsilon_l, c \sim \mathcal{N}(0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}), \tilde{\tau}_{kl}^2 = \frac{c^2 \tau_{kl}^2}{c^2 + \tau_{kl}^2 \upsilon_l^2},$$
$$\tau_{kl} \sim C^+(0, b_0), \quad \upsilon_l \sim C^+(0, b_g).$$

References

Ryan P. Adams, Hanna M Wallach, and Zoubin Ghahramani. Learning the structure of deep sparse graphical models. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Michael Betancourt. A general metric for Riemannian manifold Hamiltonian Monte Carlo. In *International Conference on Geometric Science of Information*, pages 327–334. Springer, 2013.

Michael Betancourt and Mark Girolami. Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79:30, 2015.

Anindya Bhadra, Jyotishka Datta, Nicholas G Polson, and Brandon T Willard. Lasso meets horseshoe: A survey. *arXiv preprint arXiv:1706.10179*, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning (ICML)*, 2015.

Wray L Buntine and Andreas S Weigend. Bayesian back-propagation. *Complex systems*, 5 (6):603–643, 1991.

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling sparsity via the horseshoe. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning (ICML)*, 2014.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. *International Conference on Learning Representations (ICLR)*, 2017.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *Workshop on Bayesian Deep Learning, NIPS*, 2016a.

Yarin Gal, Rowan McAllister, and Carl Edward Rasmussen. Improving PILCO with Bayesian neural network dynamics models. In *Workshop on Data-Efficient Machine Learning, ICML*, 2016b.

Soumya Ghosh and Finale Doshi-Velez. Model selection in Bayesian neural networks via horseshoe priors. *Workshop on Bayesian Deep Learning, NIPS*, 2017a.

Soumya Ghosh and Finale Doshi-Velez. Model selection in Bayesian neural networks via horseshoe priors. *https://arxiv.org/abs/1705.10388*, 2017b.

Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured variational learning of Bayesian neural networks with horseshoe priors. *International Conference on Machine Learning (ICML)*, 2018.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*, volume 104. CRC Press, 1999.

Danijar Hafner, Dustin Tran, Alex Irpan, Timothy Lillicrap, and James Davidson. Reliable uncertainty estimates in deep neural networks using noise contrastive priors. In *Uncertainty in Artificial Intelligence (UAI)*, 2019.

Thomas M Hamill. Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3):550–560, 2001.

Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *Neural Networks, 1993., IEEE Intl. Conf. on*. IEEE, 1993.

Jose Hernandez-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernández-Lobato, and Richard Turner. Black-box alpha divergence minimization. In *International Conference on Machine Learning (ICML)*, 2016.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning (ICML)*, 2015.

John B Ingraham and Debora S Marks. Variational inference for sparse and undirected models. In *International Conference on Machine Learning (ICML)*, 2017.

Pavel Izmailov, Wesley J Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for Bayesian deep learning. In *Uncertainty in Artificial Intelligence (UAI)*, 2019.

Norman Lloyd Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous univariate distributions*, volume 1. Houghton Mifflin Boston, 1970.

Ajjen Joshi, Soumya Ghosh, Margrit Betke, Stan Sclaroff, and Hanspeter Pfister. Personalizing gesture recognition using hierarchical Bayesian neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Taylor W Killian, Samuel Daulton, Finale Doshi-Velez, and George Konidaris. Robust and efficient transfer learning with hidden parameter markov decision processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P Kingma and Max Welling. Stochastic gradient VB and the variational auto-encoder. In *International Conference on Learning Representations (ICLR)*, 2014.

Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning (ICML)*, 2018.

Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NIPS)*, 1990.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *International Conference on Machine Learning (ICML)*, 2016.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

David JC MacKay. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

David JC MacKay. Developments in probabilistic modelling with neural networksensemble learning. In *Neural Networks: Artificial Intelligence and Industrial Applications*, pages 191–198. Springer, 1995.

Dougal Maclaurin, David Duvenaud, and Ryan P Adams. Autograd: Effortless gradients in numpy. In *AutoML Workshop, ICML*, 2015.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Iain Murray and Zoubin Ghahramani. A note on the evidence and Bayesian Occam's Razor. Technical report, Gatsby Unit Technical Report, 2005.

Kenton Murray and David Chiang. Auto-sizing neural networks: With applications to n-gram language models. *arXiv:1508.05051*, 2015.

Radford M Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, 1993.

Radford M Neal. *Bayesian learning for neural networks*. PhD thesis, 1997.

Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry P Vetrov. Structured Bayesian pruning via log-normal multiplicative noise. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Sarah E Neville, John T Ormerod, MP Wand, et al. Mean field variational bayes for continuous sparse signal shrinkage: pitfalls and remedies. *Electronic Journal of Statistics*, 8(1):1113–1151, 2014.

Tsubasa Ochiai, Shigeki Matsuda, Hideyuki Watanabe, and Shigeru Katagiri. Automatic node selection for deep neural networks using group Lasso regularization. *arXiv:1611.05527*, 2016.

Nick Pawlowski, Andrew Brock, Matthew CH Lee, Martin Rajchl, and Ben Glocker. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297*, 2017.

Juho Piironen and Aki Vehtari. On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Nicholas G. Polson and James G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

Melanie F Pradier, Weiwei Pan, Jiayu Yao, Soumya Ghosh, and Finale Doshi-Velez. Latent projection bnns: Avoiding weight-space pathologies by learning latent representations of neural network weights. In *Workshop on Bayesian Deep Learning, NIPS*, 2018.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's Razor. In *Advances in Neural Information Processing Systems (NIPS)*, pages 294–300, 2001.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Yale Song, David Demirdjian, and Randall Davis. Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database. In *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG)*, 2011.

Zhao Song, Yusuke Muraoka, Ryohei Fujimaki, and Lawrence Carin. Scalable model selection for belief networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

J. T. Springenberg, A. Klein, S.Falkner, and F. Hutter. Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.

Shengyang Sun, Changyou Chen, and Lawrence Carin. Learning structured weight uncertainty in Bayesian neural networks. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational Bayesian neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*, volume 1. MIT press Cambridge, 1998.

Michalis Titsias and Miguel Lázaro-gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning (ICML)*, pages 1971–1979, 2014.

Matthew P Wand, John T Ormerod, Simone A Padoan, Rudolf Fuhrwirth, et al. Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6(4), 2011.

Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for Bayesian neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*, 2011.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Christopher KI Williams. Computing with infinite networks. In *Advances in Neural Information Processing Systems (NIPS)*, 1997.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez. Quality of uncertainty quantification for Bayesian neural network inference. In *Workshop on Uncertainty and Robustness in Deep Learning, ICML*, 2019.

Journal of Machine Learning Research 20 (2019) 1-42 Submitted 4/19; Revised 11/19; Published 12/19

Log-concave sampling: Metropolis-Hastings algorithms are fast

Raaz Dwivedi[*,†] RAAZ.RSK@BERKELEY.EDU
Yuansi Chen[*,◇] YUANSI.CHEN@BERKELEY.EDU
Martin J. Wainwright[◇,†,‡] WAINWRIG@BERKELEY.EDU
Bin Yu[◇,†] BINYU@BERKELEY.EDU
Department of Statistics[◇]
Department of Electrical Engineering and Computer Sciences[†]
University of California, Berkeley
Voleon Group[‡], *Berkeley*

Editor: Suvrit Sra

Abstract

We study the problem of sampling from a strongly log-concave density supported on $\mathbb{R}^d$, and prove a non-asymptotic upper bound on the mixing time of the Metropolis-adjusted Langevin algorithm (MALA). The method draws samples by simulating a Markov chain obtained from the discretization of an appropriate Langevin diffusion, combined with an accept-reject step. Relative to known guarantees for the unadjusted Langevin algorithm (ULA), our bounds show that the use of an accept-reject step in MALA leads to an exponentially improved dependence on the error-tolerance. Concretely, in order to obtain samples with TV error at most δ for a density with condition number κ, we show that MALA requires $\mathcal{O}\big(\kappa d \log(1/\delta)\big)$ steps from a warm start, as compared to the $\mathcal{O}\big(\kappa^2 d/\delta^2\big)$ steps established in past work on ULA. We also demonstrate the gains of a modified version of MALA over ULA for weakly log-concave densities. Furthermore, we derive mixing time bounds for the Metropolized random walk (MRW) and obtain $\mathcal{O}(\kappa)$ mixing time slower than MALA. We provide numerical examples that support our theoretical findings, and demonstrate the benefits of Metropolis-Hastings adjustment for Langevin-type sampling algorithms.

Keywords: Log-concave sampling, Langevin algorithms, MCMC algorithms, conductance methods

1. Introduction

Drawing samples from a known distribution is a core computational challenge common to many disciplines, with applications in statistics, probability, operations research, and other areas involving stochastic models. In statistics, these methods are useful for both estimation and inference. Under the frequentist inference framework, samples drawn from a suitable distribution can form confidence intervals for a point estimate, such as those obtained by maximum likelihood. Sampling procedures are also standard in the Bayesian setting, used for exploring posterior distributions, obtaining credible intervals, and solving inverse problems. Estimating the mean, posterior mean in a Bayesian setting, expectations of desired quantities, probabilities of rare events and volumes of particular sets are settings in which Monte Carlo estimates are commonly used.

Recent decades have witnessed great success of Markov Chain Monte Carlo (MCMC) algorithms in generating random samples; for instance, see the handbook by Brooks et al. (2011) and references therein. In a broad sense, these methods are based on two steps. The first step is to construct a Markov chain whose stationary distribution is either equal to the target distribution or close to it in a suitable metric. Given this chain, the second step is to draw samples by simulating the chain for a certain number of steps.

Many algorithms have been proposed and studied for sampling from probability distributions with a density on a continuous state space. Two broad categories of these methods are *zeroth-order methods* and *first-order methods*. On one hand, a zeroth-order method is based on querying the density of the distribution (up to a proportionality constant) at a point in each iteration. By contrast, a first-order method makes use of additional gradient information about the density. A few popular examples of zeroth-order algorithms include Metropolized random walk (MRW) (Mengersen et al., 1996; Roberts and Tweedie, 1996b), Ball Walk (Lovász and Simonovits, 1990; Dyer et al., 1991; Lovász and Simonovits, 1993) and the Hit-and-run algorithm (Bélisle et al., 1993; Kannan et al., 1995; Lovász, 1999; Lovász and Vempala, 2006, 2007). A number of first-order methods are based on the Langevin diffusion. Algorithms related to the Langevin diffusion include the Metropolis adjusted Langevin Algorithm (MALA) (Roberts and Tweedie, 1996a; Roberts and Stramer, 2002; Bou-Rabee and Hairer, 2012), the unadjusted Langevin algorithm (ULA) (Parisi, 1981; Grenander and Miller, 1994; Roberts and Tweedie, 1996a; Dalalyan, 2016), underdamped (kinetic) Langevin MCMC (Cheng et al., 2018; Eberle et al., 2019), Riemannian MALA (Xifara et al., 2014), Proximal-MALA (Pereyra, 2016; Durmus et al., 2018), Metropolis adjusted Langevin truncated algorithm (Roberts and Tweedie, 1996a), Hamiltonian Monte Carlo (Neal, 2011) and Projected ULA (Bubeck et al., 2018). There is now a rich body of work on these methods, and we do not attempt to provide a comprehensive summary in this paper. More details can be found in the survey by Roberts et al. (2004), which covers MCMC algorithms for general distributions, and the survey by Vempala (2005), which focuses on random walks for compactly supported distributions.

In this paper, we study sampling algorithms for sampling from a log-concave distribution Π equipped with a density. The density of log-concave distribution can be written in the form

$$\pi(x) = \frac{e^{-f(x)}}{\int_{\mathbb{R}^d} e^{-f(y)} dy} \quad \text{for all } x \in \mathbb{R}^d, \tag{1}$$

where f is a convex function on $\mathbb{R}^d$. Up to an additive constant, the function $-f$ corresponds to the log-likelihood defined by the density. Standard examples of log-concave distributions include the normal distribution, exponential distribution and Laplace distribution.

Some recent work has provided non-asymptotic bounds on the mixing times of Langevin type algorithms for sampling from a log-concave density. The mixing time corresponds to the number of steps, as A function of both the problem dimension d and the error tolerance δ, to obtain a sample from a distribution that is δ-close to the target distribution in total variation

. *Raaz Dwivedi and Yuansi Chen contributed equally to this work.

©2019 Dwivedi, Chen, Wainwright and Yu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-306.html.

distance or other distribution distances. It is known that both the ULA updates (see, e.g., Dalalyan, 2016; Durmus et al., 2019; Cheng and Bartlett, 2018) as well as underdamped Langevin MCMC (Cheng et al., 2018) have mixing times that scale polynomially in the dimension d, as well the inverse of the error tolerance $1/\delta$.

Both the ULA and underdamped-Langevin MCMC methods are based on evaluations of the gradient ∇f, along with the addition of Gaussian noise. Durmus et al. (2019) shows that for an appropriate decaying step size schedule, the ULA algorithm converges to the correct stationary distribution. However, their results, albeit non-asymptotic, are hard to quantify. In the sequel, we limit our discussion to Langevin algorithms based on constant step sizes, for which there are a number of explicit quantitative bounds on the mixing time. When one uses a fixed step size for these algorithms, an important issue is that the resulting random walks are asymptotically biased: due to the lack of Metropolis-Hastings correction step, the algorithms *will not* converge to the stationary distribution if run for a large number of steps. Furthermore, if the step size is not chosen carefully the chains may become transient (Roberts and Tweedie, 1996a). Thus, the typical theory is based on running such a chain for a pre-specified number of steps, depending on the tolerance, dimension and other problem parameters.

In contrast, the Metropolis-Hastings step that underlies the MALA algorithm ensures that the resulting random walk has the correct stationary distribution. Roberts and Tweedie (1996a) derived sufficient conditions for exponential convergence of the Langevin diffusion and its discretizations, with and without Metropolis-adjustment. However, they considered the distributions with $f(x) = \|x\|_2^\alpha$ and proved geometric convergence of ULA and MALA under some specific conditions. In a more general setting, Bou-Rabee and Hairer (2012) derived non-asymptotic mixing time bounds for MALA. However, all these bounds are non-explicit, and so makes it difficult to extract an explicit dependency in terms of the dimension d and error tolerance δ. A precise characterization of this dependence is needed if one wants to make quantitative comparisons with other algorithms, including ULA and other Langevin-type schemes. Eberle (2014) derived mixing time bounds for MALA albeit in a more restricted setting compared to the one considered in this paper. In particular, Eberle's convergence guarantees are in terms of a modified Wasserstein distance, truncated so as to be upper bounded by a constant, for a subset of strongly concave measures which are four-times continuously differentiable and satisfy certain bounds on the derivatives up to order four. With this context, one of the main contributions of our paper is to provide an explicit upper bound on the mixing time bounds in total variation distance of the MALA algorithm for general log-concave distributions.

Our contributions: This paper contains two main results, both having to do with the upper bounds on mixing times of MCMC methods for sampling. As described above, our first and primary contribution is an explicit analysis of the mixing time of Metropolis adjusted Langevin Algorithm (MALA). A second contribution is to use similar techniques to analyze a zeroth-order method called Metropolized random walk (MRW) and derive an explicit non-asymptotic mixing time bound for it. Unlike the ULA, these methods make use of the Metropolis-Hastings accept-reject step and consequently converge to the target distributions in the limit of infinite steps. Here we provide explicit non-asymptotic mixing time bounds for MALA and MRW, thereby showing that MALA converges significantly

faster than ULA, at least in terms of the best known upper bounds on their respective mixing times.[1] In particular, we show that if the density is strongly log-concave and smooth, the δ-mixing time for MALA scales as $\kappa d \log(1/\delta)$ which is significantly faster than ULA's convergence rate of order $\kappa^2 d/\delta^2$. On the other hand, Moreover, we also show that MRW mixes $\mathcal{O}(\kappa)$ slower when compared to MALA. Furthermore, if the density is weakly log-concave, we show that (a modified version of) MALA converges in $\mathcal{O}\left(d^2/\delta^{1.5}\right)$ time in comparison to the $\mathcal{O}\left(d^3/\delta^4\right)$ mixing time for ULA. As alluded to earlier, such a speed-up for MALA is possible since we can choose a large step size for it which in turn is possible due to its unbiasedness in the limit of infinite steps. In contrast, for ULA the step-size has to be small enough to control the bias of the distribution of the ULA iterates in the limit of infinite steps, leading to a relative slow down when compared to MALA.

The remainder of the paper is organized as follows. In Section 2, we provide background on a suite of MCMC sampling algorithms based on the Langevin diffusion. Section 3 is devoted to the statement of our mixing time bounds for MALA and MRW, along with a discussion of some consequences of these results. Section 4 is devoted to some numerical experiments to illustrate our guarantees. We provide the proofs of our main results in Section 5, with certain more technical arguments deferred to the appendices. We conclude with a discussion in Section 6.

Notation: For two sequences a_ϵ and b_ϵ indexed by a scalar $\epsilon \in I \subseteq \mathbb{R}$, we say that $a_\epsilon = \mathcal{O}(b_\epsilon)$ if there exists a universal constant $c > 0$ such that $a_\epsilon \leq c b_\epsilon$ for all $\epsilon \in I$. The Euclidean norm of a vector $x \in \mathbb{R}^d$ is denoted by $\|x\|_2$. The Euclidean ball with center x and radius r is denoted by $\mathbb{B}(x, r)$. For two distributions $\mathcal{P}_1$ and $\mathcal{P}_2$ defined on the space $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ where $\mathcal{B}(\mathbb{R}^d)$ denotes the Borel-sigma algebra on $\mathbb{R}^d$, we use $\|\mathcal{P}_1 - \mathcal{P}_2\|_{\mathrm{TV}}$ to denote their total variation distance given by

$$\|\mathcal{P}_1 - \mathcal{P}_2\|_{\mathrm{TV}} = \sup_{A \in \mathcal{B}(\mathbb{R}^d)} |\mathcal{P}_1(A) - \mathcal{P}_2(A)| .$$

Furthermore, $\mathrm{KL}(\mathcal{P}_1 \| \mathcal{P}_2)$ denotes their Kullback-Leibler (KL) divergence. We use Π to denote the target distribution with density π.

2. Background and problem set-up

In this section, we briefly describe the general framework for MCMC algorithms and review the rates of convergence of existing random walks for log-concave distributions.

2.1. Markov chains and mixing

Here we consider the task of drawing samples from a *target distribution* Π with density π. A popular class of methods are based on setting up of an irreducible and aperiodic discrete-time Markov chain whose stationary distribution is equal to or close to the target distribution Π in certain metric, e.g., total variation (TV) norm. In order to obtain a δ-accurate sample, one simulates the Markov chain for a certain number of steps k, as determined by a mixing time analysis.

1. Throughout the paper, we make comparisons between sampling algorithms based on known upper bounds on respective mixing times; obtaining matching lower bounds is also of interest.

Going forward, we assume familiarity of the reader with a basic background in Markov chains. See, e.g., Chapters 9-12 in the standard reference textbook (Meyn and Tweedie, 2012) for a rigorous and detailed introduction. For a more rapid introduction to the basics of continuous state-space Markov chains, we refer the reader to the expository paper (Diaconis and Freedman, 1997) or Section 1 and 2 of the papers (Lovász and Simonovits, 1993; Vempala, 2005). We now briefly describe a certain class of Markov chains that are of *Metropolis-Hastings type* (Metropolis et al., 1953; Hastings, 1970). See the books (Robert, 2004; Brooks et al., 2011) and references therein for further background on these chains.

Starting at a given initial density π^0 over $\mathbb{R}^d$, any such Markov chain is simulated in two steps: (1) a proposal step, and (2) an accept-reject step. For the proposal step, we make use of a *proposal function* $p : \mathbb{R}^d \times \mathbb{R}^d \in \mathbb{R}_+$, where $p(x, \cdot)$ is a density function for each $x \in \mathbb{R}^d$. At each iteration, given a current state $x \in \mathbb{R}^d$ of the chain, the algorithm proposes a new vector $z \in \mathbb{R}^d$ by sampling from the proposal density $p(x, \cdot)$. In the second step, the algorithm accepts $z \in \mathbb{R}^d$ as the new state of the Markov chain with probability

$$\alpha(x, z) := \min\left\{1, \frac{\pi(z)p(z, x)}{\pi(x)p(x, z)}\right\}. \tag{2}$$

Otherwise, with probability equal to $1 - \alpha(x, z)$, the chain stays at x. Consequently, the overall transition kernel q for the Markov chain is defined by the function

$$q(x, z) := p(x, z)\alpha(x, z) \qquad \text{for } z \neq x,$$

and a probability mass at x with weight $1 - \int_{\mathcal{X}} q(x, z)dz$. The purpose of the Metropolis-Hastings correction (2) is to ensure that the target density π is stationary for the Markov chain.

Overall, this set-up defines an operator $\mathcal{T}_p$ on the space of probability distributions: given the distribution μ_k of the chain at time k, the distribution at time $k + 1$ is given by $\mathcal{T}_p(\mu_k)$. In fact, with the starting distribution μ_0, the distribution of the chain at kth step is given by $\mathcal{T}_p^k(\mu_0)$. Note that in this notation, the transition distribution at any state x is given by $\mathcal{T}_p(\delta_x)$ where δ_x denotes the dirac-delta distribution at x. Our assumptions and set-up ensure that the chain converges to target distribution in the limit of infinite steps, i.e., $\lim_{k\to\infty} \mathcal{T}_p^k(\mu_0) = \Pi$. However, a more practical notion of convergence is how many steps of the chain suffice to ensure that the distribution of the chain is "close" to the target Π. In order to quantify the closeness, for a given tolerance parameter $\delta \in (0, 1)$ and initial distribution μ_0, we define the δ-*mixing time* as

$$t_{\text{mix}}(\delta; \mu_0) := \min\left\{k \mid \|\mathcal{T}_p^k(\mu_0) - \Pi\|_{\text{TV}} \leq \delta\right\}, \tag{3}$$

corresponding to the minimum number of steps that the chain takes to reach within δ in TV-norm of the target distribution, given that it starts with distribution μ_0.

2.2. Sampling from log-concave distributions

Given the set-up in the previous subsection, we now describe several algorithms for sampling from log-concave distributions. Let $\mathcal{P}_x$ denote the proposal distribution at x corresponding to the proposal density $p(x, \cdot)$. Possible choices of this proposal function include:

- Independence sampler: the proposal distribution does not depend on the current state of the chain, e.g., rejection sampling or when $\mathcal{P}_x = \mathcal{N}(0, \Sigma)$, where Σ is a hyper-parameter.
- Random walk: the proposal function satisfies $p(x, y) = q(y - x)$ for some probability density q, e.g., when $\mathcal{P}_x = \mathcal{N}(x, 2h\mathbb{I}_d)$ where h is a hyper-parameter.
- Langevin algorithm: the proposal distribution is shaped according to the target distribution and is given by $\mathcal{P}_x = \mathcal{N}(x - h\nabla f(x), 2h\mathbb{I}_d)$, where h is chosen suitably.
- Symmetric Metropolis algorithms: the proposal function p satisfies $p(x, y) = p(y, x)$. Some examples are Ball Walk (Frieze et al., 1994), and Hit-and-run (Lovász, 1999).

Naturally, the convergence rate of these algorithms depends on the properties of the target density π, and the degree to which the proposal function p is suited for the task at hand. A key difference between Langevin algorithm and other algorithms is that the former makes use of first-order (gradient) information about the target distribution Π. We now briefly discuss the existing theoretical results about the convergence rate of different MCMC algorithms. Several results on MCMC algorithms have focused on on establishing behavior and convergence of these sampling algorithms in an asymptotic or a non-explicit sense, e.g., geometric and uniform ergodicity, asymptotic variance, and central limit theorems. For more details, we refer the readers to the papers by Talay and Tubaro (1990); Meyn and Tweedie (1994); Roberts and Tweedie (1996b,a); Jarner and Hansen (2000); Roberts and Rosenthal (2001); Roberts and Stramer (2002); Pillai et al. (2012); Roberts and Rosenthal (2014), the survey by Roberts et al. (2004) and the references therein. Such results, albeit helpful for gaining insight, do not provide user-friendly rates of convergence. In other words, from these results, it is not easy to determine the computational complexity of various MCMC algorithms as a function of the problem dimension d and desired accuracy δ. Explicit non-asymptotic convergence bounds, which provide useful information for practice, are the focus of this work. We discuss the results of such type and the Langevin algorithm in more detail in Section 2.2.2. We begin with the Metropolized random walk.

2.2.1. Metropolized random walk

Roberts and Tweedie (1996b) established sufficient conditions on the proposal function p and the target distribution Π for the geometric convergence of several random walk Metropolis-Hastings algorithms. In Section 3, we establish non-asymptotic convergence rate for the Metropolized random walk, which is based on Gaussian proposals. That is when the chain is at state x_k, a proposal is drawn as follows

$$z_{k+1} = x_k + \sqrt{2h}\,\xi_{k+1}, \tag{4}$$

where the noise term $\xi_{k+1} \sim \mathcal{N}(0, \mathbb{I}_d)$ is independent of all past iterates. The chain then makes the transition according to an accept-reject step with respect to Π. Since the proposal distribution is symmetric, this step can be described as

$$x_{k+1} = \begin{cases} z_{k+1} & \text{with probability} \min\left\{1, \dfrac{\pi(z_{k+1})}{\pi(x_k)}\right\} \\ x_k & \text{otherwise.} \end{cases}$$

This sampling algorithm is an instance of a zeroth-order method since it makes use of only the function values of the density π. We refer to this algorithm as MRW in the sequel. It is easy to see that the chain has a positive density of jumping from any state x to y in $\mathbb{R}^d$ and hence is strongly Π-irreducible and aperiodic. Consequently, Theorem 1 by Diaconis and Freedman (1997) implies that the chain has a unique stationary distribution Π and converges to as the number of steps increases to infinity. Note that this algorithm has also been referred to as Random walk Metropolized (RWM) and Random walk Metropolis-Hastings (RWMH) in the literature.

2.2.2. Langevin diffusion and related sampling algorithms

Langevin-type algorithms are based on Langevin diffusion, a stochastic process whose evolution is characterized by the stochastic differential equation (SDE):

$$dX_t = -\nabla f(X_t)dt + \sqrt{2}\, dW_t, \tag{5}$$

where $\{W_t \mid t \geq 0\}$ is the standard Brownian motion on $\mathbb{R}^d$. Under fairly mild conditions on f, it is known that the diffusion (5) has a unique strong solution $\{X_t, t \geq 0\}$ that is a Markov process (Roberts and Tweedie, 1996a; Meyn and Tweedie, 2012). Furthermore, it can be shown that the distribution of X_t converges as $t \to +\infty$ to the invariant distribution Π characterized by the density $\pi(x) \propto \exp(-f(x))$.

Unadjusted Langevin algorithm A natural way to simulate the Langevin diffusion (5) is to consider its forward Euler discretization, given by

$$x_{k+1} = x_k - h\nabla f(x_k) + \sqrt{2h}\xi_{k+1}, \tag{6}$$

where the driving noise $\xi_{k+1} \sim \mathcal{N}(0, \mathbb{I}_d)$ is drawn independently at each time step. The use of iterates defined by equation (6) can be traced back at least to Parisi (1981) for computing correlations; this use was noted by Besag in his commentary on the paper by Grenander and Miller (1994).

However, even when the SDE is well behaved, the iterates defined by this discretization have mixed behavior. For sufficiently large step sizes h, the distribution of the iterates defined by equation (6) converges to a stationary distribution that is no longer equal to Π. In fact, Roberts and Tweedie (1996a) showed that if the step size h is not chosen carefully, then the Markov chain defined by equation (6) can become transient and have no stationary distribution. However, in a series of recent works (Dalalyan, 2016; Durmus et al., 2019; Cheng and Bartlett, 2018), it has been established that with a careful choice of step-size h and iteration count K, running the chain (6) for exactly K steps yields an iterate x_K whose distribution is close to Π. This more recent body of work provides non-asymptotic bounds that explicitly quantify the rate of convergence for this chain. Note that the algorithm (6) does not belong to the class of Metropolis-Hastings algorithms since it does not involve an accept-reject step and does not have the target distribution Π as its stationary distribution. Consequently, in the literature, this algorithm is referred to as the *unadjusted Langevin Algorithm*, or ULA for short.

Metropolis adjusted Langevin algorithm An alternative approach to handling the discretization error is to adopt $\mathcal{N}(x_k - h\nabla f(x_k), 2h\mathbb{I}_d)$ as the proposal distribution, and perform the Metropolis-Hastings accept-reject step. Doing so leads to the *Metropolis-adjusted*

Langevin Algorithm, or MALA for short. We describe the different steps of MALA in Algorithm 1. As mentioned earlier, the Metropolis-Hastings correction ensures that the distribution of the MALA iterates $\{x_k\}$ converges to the correct distribution Π as $k \to \infty$. Indeed, since at each step the chain can reach any state $x \in \mathbb{R}^d$, it is strongly Π-irreducible and thereby ergodic (Meyn and Tweedie, 2012; Diaconis and Freedman, 1997).

Both MALA and ULA are instances of first-order sampling methods since they make use of both the function and the gradient values of f at different points. A natural question is if employing the accept-reject step for the discretization (6) provides any gain in the convergence rate. Our analysis to follow answers this question in the affirmative.

Algorithm 1: Metropolis adjusted Langevin algorithm (MALA)

Input: Step size h and a sample x_0 from a starting distribution μ_0
Output: Sequence $x_1, x_2, \ldots$
for $i = 0, 1, \ldots$ **do**
  $z_{i+1} \sim \mathcal{N}(x_i - h\nabla f(x_i), 2h\mathbb{I}_d)$ % propose a new state
  $\alpha_{i+1} = \min\left\{1, \dfrac{\exp\left(-f(z_{i+1}) - \|x_i - z_{i+1} + h\nabla f(z_{i+1})\|_2^2/4h\right)}{\exp\left(-f(x_i) - \|z_{i+1} - x_i + h\nabla f(x_i)\|_2^2/4h\right)}\right\}$
  $U_{i+1} \sim U[0,1]$
  if $U_{i+1} \leq \alpha_{i+1}$ **then** $x_{i+1} \leftarrow z_{i+1}$ % accept the proposal
  else $x_{i+1} \leftarrow x_i$ % reject the proposal
end

2.3. Problem set-up

We study MALA and MRW and contrast their performance with existing algorithms for the case when the negative log density $f(x) := -\log \pi(x)$ is smooth and strongly convex. A function f is said to be L-smooth if

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \leq \frac{L}{2}\|x - y\|_2^2 \quad \text{for all } x, y \in \mathbb{R}^d. \tag{7a}$$

In the other direction, a convex function f is said to be m-strongly convex if [2]

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \geq \frac{m}{2}\|x - y\|_2^2 \qquad \text{for all } x, y \in \mathbb{R}^d. \tag{7b}$$

The rates derived in this paper apply to log-concave[3] distributions (1) such that f is continuously differentiable on $\mathbb{R}^d$, and is both L-smooth and m-strongly convex. For such a function f, its condition number κ is defined as $\kappa := L/m$. We also refer to κ as the condition number of the target distribution Π. We summarize the mixing time bounds of several sampling algorithms in Tables 1 and 2, as a function of the dimension d, the error-tolerance δ, and the condition number κ. In Table 1, we state the results when the chain

2. See Appendix A for a statement of some well-known properties of smooth and strongly convex functions.
3. While our techniques can yield sharper guarantees under more idealized assumptions, e.g., like when f is Lipschitz (bounded gradients) and the target distribution satisfies an isoperimetry inequality, here we focus on deriving explicit guarantees with log-concave distributions.

Random walk	Strongly log-concave	Weakly log-concave
ULA (Cheng and Bartlett, 2018)	$\mathcal{O}\left(\frac{d\kappa^2 \log((\log \beta)/\delta)}{\delta^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{dL^2}{\delta^6}\right)$
ULA (Dalalyan, 2016)	$\mathcal{O}\left(\frac{d\kappa^2 \log^2(\beta/\delta)}{\delta^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{d^3L^2}{\delta^4}\right)$
MRW (this work)	$\mathcal{O}\left(d\kappa^2 \log\left(\frac{\beta}{\delta}\right)\right)$	$\tilde{\mathcal{O}}\left(\frac{d^3\,L^2}{\delta^2}\right)$
MALA (this work)	$\mathcal{O}\left(\max\left\{d\kappa, d^{0.5}\kappa^{1.5}\right\}\log\left(\frac{\beta}{\delta}\right)\right)$	$\tilde{\mathcal{O}}\left(\frac{d^2\,L^{1.5}}{\delta^{1.5}}\right)$

Table 1. Scalings of upper bounds on δ-mixing time for different random walks in $\mathbb{R}^d$ with target $\pi \propto e^{-f}$. In the second column, we consider smooth and strongly log-concave densities, and report the bounds from a β-warm start for densities such that $m\mathbb{I}_d \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for any $x \in \mathbb{R}^d$ and use $\kappa := L/m$ to denote the condition number of the density. The big-O notation hides universal constants. We remark that the presented bounds for ULA in this column are not stated in the corresponding papers, and are derived by us, using their framework. In the last column, we summarize the scaling for weakly log-concave smooth densities: $0 \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for all $x \in \mathbb{R}^d$. For this case, the $\tilde{\mathcal{O}}$ notation is used to track scaling only with respect to d, δ and L and ignore dependence on the starting distribution and a few other parameters.

Random walk	Distribution $\mu_\star$	$t_{\mathbf{mix}}(\delta;\mu_\star)$
ULA (Cheng and Bartlett, 2018)	$\mathcal{N}(x^\star, m^{-1}\mathbb{I}_d)$	$\mathcal{O}\left(\frac{d\kappa^2 \log(d\kappa/\delta)}{\delta^2}\right)$
ULA (Dalalyan, 2016)	$\mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$	$\mathcal{O}\left(\frac{(d^3 + d\log^2(1/\delta))\kappa^2}{\delta^2}\right)$
MRW (this work)	$\mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$	$\mathcal{O}\left(d^2\kappa^2 \log^{1.5}\left(\frac{\kappa}{\delta}\right)\right)$
MALA (this work)	$\mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$	$\mathcal{O}\left(d^2\kappa \log\left(\frac{\kappa}{\delta}\right)\right)$

Table 2. Scalings of upper bounds on δ-mixing time, from the starting distribution $\mu_\star$ given in column two, for different random walks in $\mathbb{R}^d$ with target $\pi \propto e^{-f}$ such that $m\mathbb{I}_d \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for any $x \in \mathbb{R}^d$ and $\kappa := L/m$. Here $x^\star$ denotes the unique mode of the target density π.

has a warm-start defined below (refer to the definition (8)). Table 2 summarizes mixing time bounds from a particular distribution $\mu_\star$. Furthermore, in Section 3.3 we discuss the case when the f is smooth but not strongly convex and show that a suitable adaptation of MALA has a faster mixing rate compared to ULA for this case.

3. Main results

We now state our main results for mixing time bounds for MALA and MRW. In our results, we use c, c' to denote universal positive constants. Their values can change depending on the context, but do not depend on the problem parameters in all cases. In this section, we begin by discussing the case of strongly log-concave densities. We state results for MALA and MRW from a warm start in Section 3.1, and from certain feasible starting distributions in Section 3.2. Section 3.3 is devoted to the case of weakly log-concave densities.

3.1. Mixing time bounds for warm start

In the analysis of Markov chains, it is convenient to have a rough measure of the distance between the initial distribution μ_0 and the stationary distribution. As in past work on the problem, we adopt the following notion of *warmness*: For a finite scalar $\beta > 0$, the initial distribution μ_0 is said to be β-warm with respect to the stationary distribution Π if

$$\sup_A \left(\frac{\mu_0(A)}{\Pi(A)}\right) \leq \beta, \tag{8}$$

where the supremum is taken over all measurable sets A. In parts of our work, we provide bounds on the quantity

$$t_{\text{mix}}(\delta;\beta) = \sup_{\mu_0 \in \mathcal{P}_\beta(\Pi)} t_{\text{mix}}(\delta;\mu_0)$$

where $\mathcal{P}_\beta(\Pi)$ denotes the set of all distributions that are β-warm with respect to Π. Naturally, as the value of β decreases, the task of generating samples from the target distribution becomes easier.[4] However, access to a good "warm" distribution (small β) may not be feasible for many applications, and thus deriving bounds on mixing time of the Markov chain from non-warm starts is also desirable. Consequently, in the sequel, we also provide practical initialization methods and polynomial-time mixing time guarantees from such starts.

Our mixing time bounds involve the functions r and w given by

$$r(s) = 2 + 2 \cdot \max\left\{\frac{1}{d^{0.25}}\log^{0.25}\left(\frac{1}{s}\right), \frac{1}{d^{0.5}}\log^{0.5}\left(\frac{1}{s}\right)\right\}, \quad \text{and} \tag{9a}$$

$$w(s) = \min\left\{\frac{\sqrt{m}}{r(s)\cdot L\sqrt{dL}}, \frac{1}{Ld}\right\} \qquad \text{for } s \in \left(0, \tfrac{1}{2}\right). \tag{9b}$$

We use $\mathcal{T}_{\text{MALA}(h)}$ to denote the transition operator on probability distributions induced by one step of MALA. With this notation, we have the following mixing time bound for the MALA algorithm for a strongly-log concave measure from a warm start.

Theorem 1 *For any β-warm initial distribution μ_0 and any error tolerance $\delta \in (0,1]$, the Metropolis adjusted Langevin algorithm with step size $h = c\,w(\delta/(2\beta))$ satisfies the bound $\|\mathcal{T}^k_{MALA(h)}(\mu_0) - \Pi\|_{TV} \leq \delta$ for all iteration numbers*

$$k \geq c' \log\left(\frac{2\beta}{\delta}\right) \max\left\{d\kappa,\ d^{0.5}\kappa^{1.5} r\left(\frac{\delta}{2\beta}\right)\right\}, \tag{10}$$

where c, c' denote universal constants.

4. For instance, $\beta = 1$ implies that the chain starts at the stationary distribution and has already mixed.

See Section 5.2 for the proof.

Note that $r(s) \leq 4$ for $s \geq e^{-d}$ and thus we can treat $r(\delta/2\beta)$ as small constant for most interesting values of δ if the warmness parameter β is not too large. Consequently, we can run MALA with a fixed step size h for a large range of error-tolerance δ. Treating the function r as a constant, we obtain that if $\kappa = o(d)$, the mixing time of MALA scales as $\mathcal{O}\left(d\kappa \log(1/\delta)\right)$. Note that the dependence on the tolerance δ is exponentially better than the $\mathcal{O}\left(d\kappa^2 \log^2(1/\delta)/\delta^2\right)$ mixing time of ULA, and has better dependence on κ while still maintaining linear dependence on d. In fact, for any setting of κ, d and δ, MALA always has a better mixing time bound compared to ULA. A limitation of our analysis is that the constant c' is not small. However, we demonstrate in Section 4 that in practice small constants provide performance that matches the scalings suggested by our theoretical bounds.

Let $\mathcal{T}_{\text{MRW}(h)}$ denote the transition operator on the space of probability distributions induced by one step of MRW. We now state the convergence rate for Metropolized random walk for strongly-log concave density.

Theorem 2 *For any β-warm initial distribution μ_0 and any $\delta \in (0, 1]$, the Metropolized random walk with step size $h = \frac{cm}{dL^2 r(\delta/2\beta)}$ satisfies*

$$\|\mathcal{T}^k_{MRW(h)}(\mu_0) - \Pi\|_{TV} \leq \delta \quad \textit{for all} \quad k \geq c'\, d\kappa^2 r\left(\frac{\delta}{2\beta}\right) \log\left(\frac{2\beta}{\delta}\right), \tag{11}$$

where c, c' denote universal constants.

See Section 5.6 for the proof.

Again treating $r(\delta/2\beta)$ as a small constant, we find that the mixing time of MRW scales as $\mathcal{O}\left(d\kappa^2 \log(1/\delta)\right)$ which has an exponential factor in δ better than ULA. Compared to the mixing time bound for MALA, the bound in Theorem 2 has an extra factor of $\mathcal{O}(\kappa)$. While such a factor is conceivable given that MALA's proposal distribution uses first-order information about the target distribution and MRW uses only the function values, it would be interesting to determine if this gap can be improved. See Section 6 for a discussion on possible future work in this direction.

3.2. Mixing time bounds for a feasible start

In many cases, a good warm start may not be available. Consequently, mixing time bounds from a feasible starting distribution can be useful in practice. Letting $x^\star$ denote the unique mode of the target distribution Π, we claim that the distribution $\mu_\star = \mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$ is one such choice. Recalling the condition number $\kappa = L/m$, we claim that the warmness parameter for $\mu_\star$ can be bounded as follows:

$$\sup_A \frac{\mu_\star(A)}{\Pi(A)} \leq \kappa^{d/2} = \beta_\star, \tag{12}$$

where the supremum is taken over all measurable sets A. When the gradient ∇f is available, finding $x^\star$ comes at nominal additional cost: in particular, standard optimization algorithms

such as gradient descent be used to compute a δ-approximation of $x^\star$ in $\mathcal{O}\left(\kappa \log(1/\delta)\right)$ steps (e.g., see the monograph by Bubeck 2015). Also refer to Section 3.2.1 for more details when we have inexact parameters.

Assuming claim (12) for the moment, we now provide mixing time bounds for MALA and MRW with $\mu_\star$ as the starting distribution. For any threshold $\delta \in (0, 1]$, we define the step sizes $h_1 = c' w(\delta/2\beta_\star)$ and $h_2 = \frac{c'm}{dL^2 \cdot r(\delta/2\beta_\star)}$, where the function w was previously defined in equation (9b).

Corollary 3 *With $\mu_\star$ as the starting distribution, we have*

$$\|\mathcal{T}^k_{MRW(h_2)}(\mu_\star) - \Pi\|_{TV} \leq \delta \quad \textit{for all } k \geq c\, d^2\kappa^2 \log^{1.5}\left(\frac{\kappa}{\delta^{1/d}}\right), \quad \textit{and} \tag{13a}$$

$$\|\mathcal{T}^k_{MALA(h_1)}(\mu_\star) - \Pi\|_{TV} \leq \delta \quad \textit{for all } k \geq c\, d^2\kappa \log\left(\frac{\kappa}{\delta^{1/d}}\right) \max\left\{1, \sqrt{\frac{\kappa}{d} \log\left(\frac{\kappa}{\delta^{1/d}}\right)}\right\}. \tag{13b}$$

The proof follows by plugging the bound (12) in Theorems 1 and 2 and is thereby omitted.

We now prove the claim (12). Without loss of generality, we can assume that $f(x^\star) = 0$. Such an assumption is possible because substituting $f(\cdot)$ by $f(\cdot) + \alpha$ for any scalar α leaves the distribution Π unchanged. Since f is m-strongly convex and L-smooth, applying Lemma 8(c) and Lemma 9(c), we obtain that

$$\frac{L}{2}\|x - x^\star\|_2^2 \geq f(x) \geq \frac{m}{2}\|x - x^\star\|_2^2, \quad \forall x \in \mathbb{R}^d.$$

Consequently, we find that $\int_{\mathbb{R}^d} e^{-f(x)} dx \leq (2\pi/m)^{d/2}$. Making note of the lower bound

$$\pi(x) \geq \frac{e^{-\frac{L}{2}\|x - x^\star\|_2^2}}{(2\pi m^{-1})^{d/2}}, \tag{14}$$

and plugging in the expression for the density of $\mu_\star$ yields the claim (12).

We now derive results for the case when we do not have access to exact parameters, e.g., if the mode $x^\star$ is known approximately, and/or we only have an upper bound for the smoothness parameter L—a situation quite prevalent in practice.

3.2.1. Starting distribution with inexact parameters

Note that $x^\star$ is also the unique global minimum of the negative log-density f. For the strongly convex function f, using a first-order method, like gradient descent, we can obtain an ϵ-approximate mode $\tilde{x}$ using $\kappa \log(1/\epsilon)$ evaluations of the gradient ∇f. Suppose we have access to a point $\tilde{x}$ such that $\|\tilde{x} - x^\star\|_2 \leq \epsilon$ and have an upper bound estimate $\tilde{L} \geq L$ for the smoothness.

We now consider the case of starting distribution $\tilde{\mu} = \mathcal{N}(\tilde{x}, (2\tilde{L})^{-1}\mathbb{I}_d)$, as a proxy for the feasible start $\mu_\star = \mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$ discussed above. Note the difference in mean and the covariance between the distributions $\tilde{\mu}$ and $\mu_\star$. Given the handy result in Theorem 1, it suffices to bound the warmness parameter for the distribution $\tilde{\mu}$. Applying the triangle inequality, we find that

$$\|x - \tilde{x}\|_2^2 \geq \frac{1}{2}\|x - x^\star\|_2^2 - \|x^\star - \tilde{x}\|_2^2 \tag{15}$$

and consequently that

$$\begin{aligned}\tilde{\mu}(x) &= (\pi\tilde{L}^{-1})^{-d/2}\exp\left(-\tilde{L}\left\|x-\tilde{x}\right\|_2^2\right)\\ &\le (\pi\tilde{L}^{-1})^{-d/2}\exp\left(-\frac{\tilde{L}\left\|x-x^\star\right\|_2^2}{2}+\tilde{L}\left\|\tilde{x}-x^\star\right\|_2^2\right)\end{aligned}$$

Using the lower bound (14) on the target density, we find that

$$\begin{aligned}\frac{\tilde{\mu}(x)}{\pi(x)} &\le \left(\frac{\tilde{L}}{L}\cdot 2\kappa\right)^{d/2}\exp\left(\tilde{L}\left\|\tilde{x}-x^\star\right\|_2^2-\frac{(\tilde{L}-L)\left\|x-x^\star\right\|_2^2}{2}\right)\\ &\le \exp\left(\frac{d}{2}\log(2\kappa\tilde{L}/L)+\tilde{L}\epsilon^2\right),\end{aligned}$$

where the last inequality follows from the fact that $\tilde{L}\ge L$. In other words, the distribution $\tilde{\mu}$ is $\tilde{\beta}$-warm with respect to the target distribution π, where $\tilde{\beta}=\exp\left(\frac{d}{2}\log(2\kappa\tilde{L}/L)+\tilde{L}\epsilon^2\right)$.

Using Theorem 1, we now derive a mixing time bound for MALA with starting distribution $\tilde{\mu}$. For any threshold $\delta\in(0,1]$, we use the step size $h_3=c'w(\delta/(2\tilde{\beta}))$. Invoking Theorem 1 and plugging in the definition (9a) of w, we find that $\|\mathcal{T}^k_{\mathrm{MALA}(h_3)}(\tilde{\mu})-\Pi\|_{\mathrm{TV}}\le\delta$, for all

$$k\ge cd^2\kappa\left(\log\frac{2\kappa\tilde{L}/L}{\delta^{1/d}}+\frac{\tilde{L}\epsilon^2}{d}\right)\max\left\{1,\sqrt{\frac{\kappa}{d}}\left(\sqrt{\log\frac{2\kappa\tilde{L}/L}{\delta^{1/d}}}+\frac{\sqrt{\tilde{L}}\epsilon}{\sqrt{d}}\right)\right\},\tag{16}$$

which also recovers the bound from corollary 3 for MALA as $\epsilon\to 0$ and $\tilde{L}\to L$. Note that the mixing time increases (additively) by $\mathcal{O}\left(\kappa d\epsilon^2\tilde{L}/L\right)$ when we only have an ϵ-approximate mode, which is an $(\tilde{L}/L\cdot\epsilon/d)$-fraction increase in the mixing time bound with starting distribution $\mu_\star$. A mixing time bound for MRW with starting distribution $\tilde{\mu}$ can be obtained in a similar fashion and is thereby omitted.

3.3. Weakly log-concave densities

In this section, we show that MALA can also be used for approximate sampling from a density which is L-smooth but not necessarily strongly log-concave (also referred to as weakly log-concave, see, e.g., Dalalyan 2016). In simple words, the negative log-density f satisfies the condition (7a) with parameter L and satisfies the condition (7b) with parameter $m=0$, (equivalently we have $L\mathbb{I}_d\succeq\nabla^2 f(x)\succeq 0$; see Appendix A for further details.) Note that we still assume that $\int_{\mathbb{R}^d}e^{-f(x)}dx<\infty$ so that the distribution Π is well-defined.

In order to make use of our previous machinery for such a case, we approximate the given log-concave density Π with a strongly log-concave density $\tilde{\Pi}$ such that $\|\tilde{\Pi}-\Pi\|_{\mathrm{TV}}$ is small. Next, we use MALA to sample from $\tilde{\Pi}$ and consequently obtain an approximate sample from Π. In order to construct $\tilde{\Pi}$, we use a scheme previously suggested by Dalalyan (2016). With λ as a tuning parameter, consider the distribution $\tilde{\Pi}$ given by the density

$$\tilde{\pi}(x)=\frac{1}{\int_{\mathbb{R}^d}e^{-\tilde{f}(y)}dy}e^{-\tilde{f}(x)}\quad\text{where}\quad\tilde{f}(x)=f(x)+\frac{\lambda}{2}\left\|x-x^\star\right\|_2^2.\tag{17}$$

Dalalyan (2016) (Lemma 3) showed that that the total variation distance between Π and $\tilde{\Pi}$ is bounded as follows:

$$\|\tilde{\Pi}-\Pi\|_{\mathrm{TV}}\le\frac{1}{2}\left\|\tilde{f}-f\right\|_{L^2(\pi)}\le\frac{\lambda}{4}\left(\int_{\mathbb{R}^d}\left\|x-x^\star\right\|_2^4\pi(x)dx\right)^{1/2}.$$

Suppose that the original distribution Π has its fourth moment bounded as

$$\int_{\mathbb{R}^d}\left\|x-x^\star\right\|_2^4\pi(x)dx\le d^2\nu^2.\tag{18}$$

We now set $\lambda:=2\delta/(d\nu)$ to obtain $\|\tilde{\Pi}-\Pi\|_{\mathrm{TV}}\le\delta/2$. Since $\tilde{f}$ is $\lambda/2$-strongly convex and $L+\lambda/2$-smooth, the condition number of $\tilde{\Pi}$ is given by $\tilde{\kappa}=1+Ld\nu/\delta$. We substitute $\tilde{\kappa}=Ld\nu/\delta$ to obtain simplified expressions for mixing time bounds in the results that follow. Since now the target distribution is $\tilde{\Pi}$, we suitably modify the step size for MALA as follows:

$$w_{\mathrm{lc}}(s)=\frac{1}{Ld}\min\left\{\frac{\sqrt{s}}{r(s)\sqrt{\nu L}},\ 1\right\}$$

where the function r was previously defined in equation (9a). We refer to this new set-up with a modified target distribution $\tilde{\Pi}$ as the *modified MALA method*. To keep our results simple to state, we assume that we have a warm start with respect to $\tilde{\Pi}$.

Corollary 4 *Assume that* Π *satisfies the condition* (18)*. Then for any given error-tolerance* $\delta\in(0,1)$*, and, any* β*-warm start* μ_0*, the modified MALA method with step size* $h=cw_{lc}(\delta/(2\beta))$ *satisfies* $\|\mathcal{T}^k_{MALA(h)}(\mu_0)-\Pi\|_{TV}\le\delta$ *for all*

$$k\ge c'\log\left(\frac{4\beta}{\delta}\right)\max\left\{\frac{d^2L\nu}{\delta},\ d^2\left(\frac{L\nu}{\delta}\right)^{1.5}r\left(\frac{\delta}{4\beta}\right)\right\},$$

where c,c' *denote universal positive constants.*

The proof follows by combining the triangle inequality, as applied to the TV norm, along with the bound from Theorem 1. Thus, for weakly log-concave densities, modified MALA mixes in $\mathcal{O}\left(d^2/\delta^{1.5}\right)$, which improves upon the $\mathcal{O}\left(d^3/\delta^4\right)$ mixing time bound for a ULA scheme on $\tilde{\Pi}$, as established by Dalalyan (2016). A mixing time bound of $\mathcal{O}\left(d^3/\delta^2\right)$ for MRW can be derived similarly for this case, simply by noting that the new condition number $\tilde{\kappa}=Ld\nu/\delta$ for the modified density and the fact that the mixing time of MRW is $\mathcal{O}\left(d\tilde{\kappa}^2\right)$ in the strongly log-concave setting.

4. Numerical experiments

In this section, we compare MALA with ULA and MRW in various simulation settings. The step-size choice of ULA follows from the paper by Dalalyan (2016) in the case of a warm start. The step-size choice of MALA and MRW used in our experiments in our results are summarized in Table 3.

Summary of experiment set-ups and diagnostic tools: We consider four different experiments: (i) sampling a multivariate Gaussian (Section 4.1), (ii) sampling a Gaussian mixture (Section 4.2), (iii) estimating the MAP with credible intervals in a Bayesian logistic regression set-up (Section 4.3) and (iv) studying the effect of step-size on the accept reject step (Section 4.4). Since TV distance for continuous measures is hard to estimate, we use several proxy measures for convergence diagnostics: (a) errors in quantiles, (b) ℓ_1-distance in histograms (which we refer to as discrete tv-error), (c) error in sample MAP estimate, (d) trace-plot along different coordinates and (e) autocorrelation plot. While the first three measures (a-c) are useful for diagnosing the convergence of random walks over several independent runs, the last two measures (d-e) are useful for diagnosing the rate of convergence of the Markov chain in a single long run.

4.1. Dimension dependence for multivariate Gaussian

The goal of this simulation is to demonstrate the dimension dependence in experiments, for mixing time of ULA, MALA, and MRW when the target is a non-isotropic multivariate Gaussian. Note that Theorems 1 and 2 imply that the dimension dependency for both MALA and MRW is d. We consider sampling from multivariate Gaussian with density π defined by

$$x \mapsto \pi(x) \propto e^{-\frac{1}{2}x^\top \Sigma^{-1} x}, \tag{19}$$

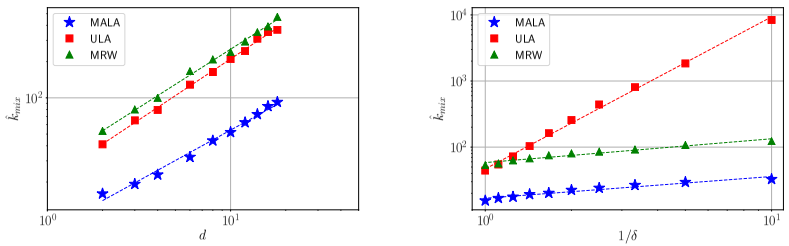
where $\Sigma \in \mathbb{R}^{d\times d}$ the covariance matrix to be specified. For this target distribution, the function f, its derivatives are given by

$$f(x) = \frac{1}{2}x^\top \Sigma^{-1} x, \quad \nabla f(x) = \Sigma^{-1}x, \quad \text{and} \quad \nabla^2 f(x) = \Sigma^{-1}.$$

Consequently, the function f is strongly convex with parameter $m = 1/\lambda_{\max}(\Sigma)$ and smooth with parameter $L = 1/\lambda_{\min}(\Sigma)$. For convergence diagnostics, we use the error in quantiles along different directions. Using the exact quantile information for each direction for Gaussians, we measure the error in the 75% quantile of the sample distribution and the true distribution in the *least favorable direction*, i.e., along the eigenvector of Σ corresponding to the eigenvalue $\lambda_{\max}(\Sigma)$. The *approximate mixing time* $\hat{k}_{\text{mix}}(\delta)$ is defined as the smallest iteration when this error falls below δ. We use $\mu_\star$ as the initial distribution where $\mu_\star = \mathcal{N}\left(0, L^{-1}\mathbb{I}_d\right)$.

4.1.1. Strongly log-concave density

The step-sizes are chosen according to Table 3. For ULA, the error-tolerance δ is chosen to be 0.2. We set Σ as a diagonal matrix with the largest eigenvalue 4.0 and the smallest eigenvalue 1.0 so that the $\kappa = 4$ is fixed across different settings. For a fixed dimension d, we simulate 10 independent runs of the three chains each with $N = 10,000$ samples to determine the approximate mixing time. The final approximate mixing time for each walk is the average of that over these 10 independent runs. Figure 1(a) shows the dependency of the approximate mixing time as a function of dimension d for the three random walks in log-log scale. To examine the dimension dependency, we perform linear regression for approximate mixing time with respect to dimensions in the log-log scale. The computations



(a) (b)

Figure 1. Scaling of the approximate mixing time $\hat{k}_{\text{mix}}$ (refer to the discussion after equation (19) for the definition) on multivariate Gaussian density (19) where the covariance has condition number $\kappa = 4$. (a) Dimension dependency. (b) Error-tolerance dependency.

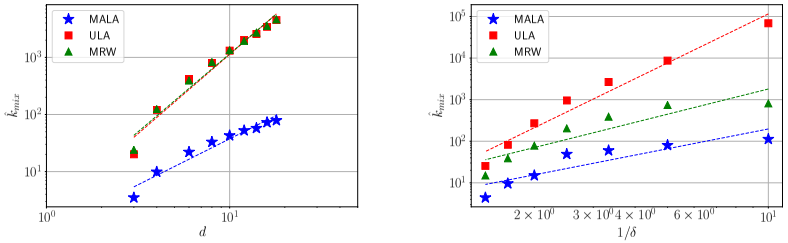
reveal that the dimension dependency of MALA, ULA and MRW are all close to order d (slope 0.84, 1.01 and 0.97). Figure 1(b) shows the dependency of the approximate mixing time on the inverse error $1/\delta$ for the three random walks in log-log scale. For ULA, the step-size is error-dependent, precisely chosen to be 10 times of δ. A linear regression of the approximate mixing time on the inverse error $1/\delta$ yields a slope of 2.23 suggesting the error dependency of order $1/\delta^2$ for ULA. A similar computation for MALA and MRW yields a slope of 0.33 for both the cases which not only suggests a significantly better error dependency for these two chains but also partly verifies their theoretical mixing time bounds of order $\log(1/\delta)$.

Random walk	ULA	MALA	MRW
Step size	$\frac{\delta^2}{d\kappa L}$	$\frac{1}{L}\min\left\{\frac{1}{\sqrt{d\kappa}}, \frac{1}{d}\right\}$	$\frac{1}{d\kappa L}$

Table 3. Step size used in simulations to obtain δ-accuracy for different random walks in $\mathbb{R}^d$ with target $\pi \propto e^{-f}$ such that $m\mathbb{I}_d \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for any $x \in \mathbb{R}^d$ and $\kappa := L/m$.

4.1.2. Weakly log-concave density

We now discuss the convergence of the random walks when the Gaussian is flat along a direction. In particular, we consider the Gaussian distribution such that $\lambda_{\max}(\Sigma) = 1000$ and $\lambda_{\min}(\Sigma) = 1$. Such a setting implies that the strong convexity parameter $m = 0.001 \approx 0$ and hence our target density mimics a weakly log-concave density. For convergence diagnostics, we use the error in quantiles along one direction other than the ones which correspond to $\lambda_{\max}(\Sigma)$ and $\lambda_{\min}(\Sigma)$. Using the exact quantile information for each direction for Gaussians, we measure the error between the 75% quantile of the sample distribution and the true distribution in that direction. The approximate mixing time is defined as the smallest iteration when this error falls below δ. We use $\mu_\star$ as the initial distribution



(a) (b)

Figure 2. Scaling of the approximate mixing time $\hat{k}_{\text{mix}}$ (refer to the discussion after equation (19) for the definition) for a close to weakly log-concave Gaussian density. (a) Dimension dependency. (b) Error-tolerance dependency for fixed dimension .

where $\mu_\star = \mathcal{N}\left(0, L^{-1}\mathbb{I}_d\right)$. The step-sizes are chosen according to Table 3 where m is chosen to be $\delta/(dL)$. For dimension dependence experiments, we fix the error-tolerance δ as 0.2. For a fixed dimension d, we simulate 10 independent runs of the three chains each with $N = 10,000$ samples to determine the approximate mixing time. The final approximate mixing time for each walk is the average of that over these 10 independent runs. Figure 2(a) and 2(b) show the dependency of the approximate mixing time as a function of dimension d and the inverse error $1/\delta$ respectively, for the three random walks on this weakly log-concave density (log-log scale). Linear fits on the log-log scale reveal that the dimension dependence of mixing time for MALA is close to d^2 (slope 1.61), and that for ULA is close to d^3 (slope 2.78) and for MRW it is approximately of order d^3 (slope 2.73). Linear fits of the approximate mixing time on the inverse error $1/\delta$ yield a slope of 3.92 for ULA thereby suggesting an error dependence of order $1/\delta^4$, while for MALA and MRW this dependence is of order $1/\delta^{1.5}$ (slope 1.56) and of order $1/\delta^2$ (slope 2.01), respectively. These scalings partly verify the rates derived in Corollary 4 and demonstrate the gains of MALA over ULA for the weakly log-concave densities.

4.1.3. Warmness in simulations

Strictly speaking, for both the cases considered above, the starting distribution was not warm since we used $\mu_\star$ as the starting distribution and the corresponding warmness parameter $\beta = \mathcal{O}(e^d)$ scales exponentially with dimension d. However, the mixing time observed in the simulations, albeit with a heuristic measure, are d times faster than those stated with $\mu_\star$ as the starting distribution in Corollary 3, and are in fact consistent with the results for the warm-start which are stated in Theorems 1 and 2. We believe that the results stated in Corollary 3, with $\mu_\star$ as the starting distribution can be improved by a factor of d. See Section 6 for a discussion on recent work on this question.

4.2. Behavior for Gaussian mixture distribution

We now consider the task of sampling from a two component Gaussian mixture distribution, as previously considered by Dalalyan (2016) for illustrating the behavior of ULA. Here compare the behavior of MALA to ULA for this case. The target density is given by

$$x \mapsto \pi(x) = \frac{1}{2\left(2\pi\right)^{d/2}}\left(e^{-\|x-a\|_2^2/2} + e^{-\|x+a\|_2^2/2}\right),$$

where $a \in \mathbb{R}^d$ is a fixed vector. This density corresponds to the two-mixture of equal weighted Gaussians $\mathcal{N}(a, \mathbb{I}_d)$ and $\mathcal{N}(-a, \mathbb{I}_d)$. In our notation, the function f and its derivatives are given by: $f(x) = \frac{1}{2}\|x-a\|_2^2 - \log(1 + e^{-2x^\top a})$,

$$\nabla f(x) = x - a + 2a(1 + e^{2x^\top a})^{-1}, \text{ and }, \nabla^2 f(x) = \mathbb{I}_d - 4aa^\top \frac{e^{2x^\top a}}{\left(1 + e^{2x^\top a}\right)^2}.$$

From examination of the Hessian, we see that the function f is smooth with parameter $L = 1$, and whenever $\|a\|_2 < 1$, it is also strongly convex with parameter $m = 1 - \|a\|_2^2$.

For dimension $d = 2$, setting $a = \left(\frac{1}{2}, \frac{1}{2}\right)$ yields the parameters $m = \frac{1}{2}$ and $L = 1$. Figure 3 shows the level sets of the density of this 2D-Gaussian mixture. The initial distribution is chosen as $\mu_\star = \mathcal{N}\left(0, L^{-1}\mathbb{I}_d\right)$ and the step-sizes are chosen according to Table 1, where for ULA, we set three different choices of $\delta = 0.2$ (ULA), $\delta = 0.1$ (small-step ULA) and $\delta = 1.0$ (large-step ULA). Note that choosing a smaller threshold δ implies that the ULA has a smaller step size and consequently the chain takes larger to converge. However, the asymptotic TV error with respect to the target distribution Π for ULA also decreases with a decrease in step size. These different choices of step sizes are made to demonstrate the fundamental trade-off between the rate of convergence and asymptotic error for ULA and its inability to mix faster than MALA for different settings.

Note that one can sample directly from the mixture of Gaussian under consideration by drawing independently a Bernoulli(1/2) random variable y and a standard normal variable $z \sim \mathcal{N}(0, \mathbb{I}_d)$, and then outputting the random variable

$$x = y \cdot (z - a) + (1 - y) \cdot (z + a)$$

This observation makes it easy to diagnose the convergence of our Markov chains with target π. In order to estimate the total variation distance, we discretize the distribution of $N = 250,000$ samples from π over a set of bins, and consider the total variation of this discrete distribution from the empirical distribution of the Markov chain over these bins. We refer to this measure as the discretized TV error. We measure the sum of two discrete TV errors of $250,000$ samples from π with the empirical distribution obtained by simulating the chains ULA, MALA or MRW, projected on two principal directions (u_1 and u_2), over a discrete grid of size $B = 100$. Figure 4 shows the sum of the discretized TV errors along u_1 and u_2, as a function of iterations. The true total variation distance between the distribution of the iterate and the target distribution is upper bounded by the sum of (a) the discretized TV error and (b) the error caused by discretization. In order to obtain a sense of the magnitude of the error of type (b), we simulate 100 runs of the discrete TV error between two independent drawings from the true distribution π. The two black lines

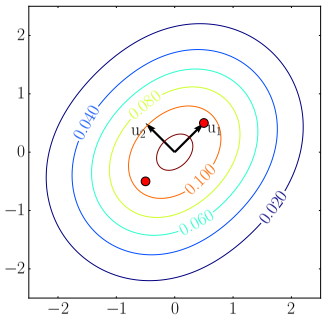
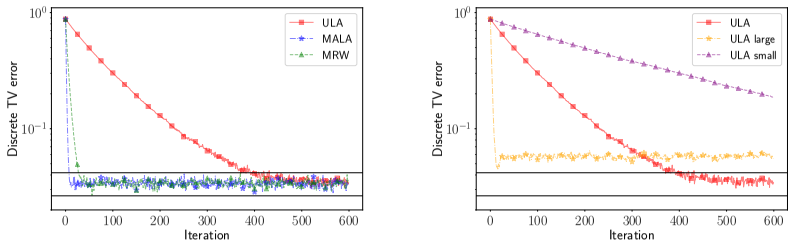


Figure 3. Level set of the density of the 2D Gaussian mixture. The red dots are the location of the means a and $-a$, where a is chosen such that $\|a\|_2^2 = \frac{1}{2}$. The arrows indicate the two principal directions u_1 and u_2 along which the TV error is measured.

in Figure 4 are the maximum and minimum of these 100 values. The sample distribution at convergence is expected to lie between the two black lines.

Figure 4(a) shows that ULA converges significantly slower than MALA to the right distribution. Figure 4(b) illustrates this point further and shows that when compared to the ULA, the small-step ULA ($\delta = 0.1$) converges at a much slower rate and large-step ULA ($\delta = 1.0$) has a larger approximation error (asymptotic bias).

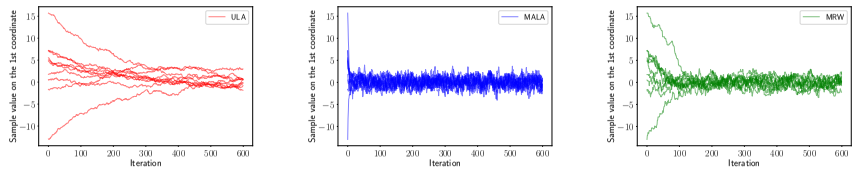


(a) (b)

Figure 4. Discrete TV error on a two component Gaussian mixture. (a) Behavior of three different random walks. (b) Behavior of ULA with different choices of step sizes.

We accompany the study based on exact TV error computation with two classical convergence diagnostic plots for general MCMC algorithms. Figure 5 shows the trace plots of the three sampling algorithms in 10 runs. Comparing the three panels (a)—(c) in Figure 5, we observe that the trace plot of MALA stabilizes much faster than that of ULA and MRW.

Furthermore, to compare the efficiency of the chains in stationarity, Figure 6 shows the autocorrelation function of the three chains. In order to ensure that these autocorrelations are computed for the stationary distribution, we set in practice the burn-in period to be 300 iterations. Again, we observe that MALA is more efficient than ULA and MRW.



(a) (b) (c)

Figure 5. Trace-plot of the first coordinate on a two component Gaussian mixture. (a) Trace-plot of ULA. (b) Trace-plot of MALA. (c) Trace-plot of MRW.

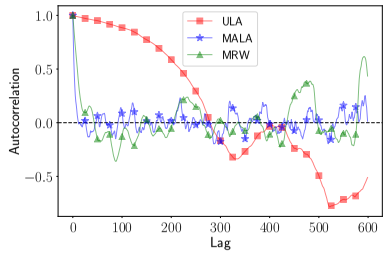


Figure 6. Markov chain autocorrelation function plot. The burn-in time for the plot is set to 300 iterations.

4.3. Bayesian Logistic Regression

We now consider the problem of logistic regression in a frequentist-Bayesian setting, similar to that considered by Dalalyan (2016). Once again, we establish that MALA has superior performance relative to ULA. Given a binary variable $y \in \{0, 1\}$ and a covariate $x \in \mathbb{R}^d$, the logistic model for the conditional distribution of y given x takes the form

$$\mathbb{P}(y = 1|x; \theta) = \frac{e^{\theta^\top x}}{1 + e^{\theta^\top x}}, \tag{20}$$

for some parameter $\theta \in \mathbb{R}^d$.

In a Bayesian framework, we model the parameter θ in the logistic equation as a random variable with a prior distribution π_0. Suppose that we observe a set of independent

samples $\{(x_i, y_i)\}_{i=1}^n$ with $(x_i, y_i) \in \mathbb{R}^d \times \{0, 1\}$, with each y_i conditioned on x_i drawn from a logistic distribution with some unknown parameter θ^*. Using Bayes' rule, we can then compute the posterior distribution of the parameter θ given the data. Drawing samples from this posterior distribution allows us to estimate and draw inferences about the unknown parameter. Under mild conditions, the Bernstein-von-Mises theorem guarantees that the posterior distribution will concentrate around the true parameter θ^*, in which case we expect that the credible intervals formed by sampling from the posterior should contain θ^* with high probability. This fact provides a lens for us to assess the accuracy of our sampling procedure.

Define the vector $Y = (y_1, \ldots, y_n)^\top \in \{0, 1\}^n$ and let X be the $n \times d$ matrix with x_i as i^{th}-row. We choose the prior π_0 to be a Gaussian distribution with zero mean and covariance matrix proportional to the inverse of the sample covariance matrix $\Sigma_X = \frac{1}{n} X^\top X$. Plugging in the formulas for the prior and likelihood, we find that the the posterior density is given by

$$\pi(\theta) = \pi(\theta | X, Y) \propto \exp\left\{ Y^\top X\theta - \sum_{i=1}^n \log\left(1 + e^{\theta^\top x_i}\right) - \alpha \left\| \Sigma_X^{1/2} \theta \right\|_2^2 \right\},$$

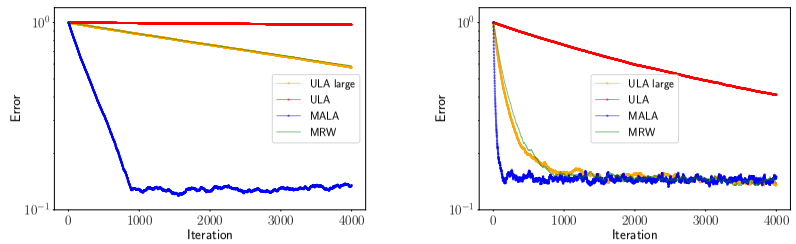
where $\alpha > 0$ is a user-specified parameter. Writing $\pi \propto e^{-f}$, we observe that the function f and its derivatives are given by

$$\begin{aligned} f(\theta) &= -Y^\top X\theta + \sum_{i=1}^n \log\left(1 + e^{\theta^\top x_i}\right) + \alpha \left\| \Sigma_X^{1/2} \theta \right\|_2^2, \\ \nabla f(\theta) &= -X^\top Y + \sum_{i=1}^n \frac{x_i}{1 + e^{-\theta^\top x_i}} + \alpha \Sigma_X \theta, \quad \text{and,} \\ \nabla^2 f(\theta) &= \sum_{i=1}^n \frac{e^{-\theta^\top x_i}}{\left(1 + e^{-\theta^\top x_i}\right)^2} x_i x_i^\top + \alpha \Sigma_X. \end{aligned}$$

With some algebra, we can deduce that the eigenvalues of the Hessian $\nabla^2 f$ are bounded between $L := (0.25n + \alpha)\, \lambda_{\max}(\Sigma_X)$ and $m := \alpha\, \lambda_{\min}(\Sigma_X)$ where $\lambda_{\max}(\Sigma_X)$ and $\lambda_{\min}(\Sigma_X)$ denote the largest and smallest eigenvalues of the matrix Σ_X. We make use of these bounds in our experiments.

As in Dalalyan (2016), we also consider a preconditioned version of the method; more precisely, we first sample from $\pi_g \propto e^{-g}$ where $g(\theta) = f(\Sigma_X^{-1/2}\theta)$, and then transform the obtained random samples $\theta_i \mapsto \Sigma_X^{1/2}\theta_i$ to obtain samples from π. Sampling based on the preconditioned distribution improves the condition number of the problem. After the preconditioning, we have the bounds $L_g \leq 0.25n + \alpha$ and $m_g \geq \alpha$, so that the new condition number is now independent of the eigenvalues of Σ_X.

We randomly draw i.i.d. samples (x_i, y_i) as follows. Each vector $x_i \in \mathbb{R}^d$ is sampled i.i.d. Rademacher components, and then renormalized to Euclidean norm. given x_i, the response y_i is drawn from the logistic model (20) with $\theta = \theta^* = \mathbf{1}_d = (1, \ldots, 1)^\top$. We fix $d = 2, n = 50$ and perform $N = 1000$ experiments. In order to sample from the posterior, we start with the initial distribution as $\mu_0 = \mathcal{N}(0, L^{-1}\mathbb{I}_d)$. As the first error metric, we



(a) (b)

Figure 7. Mean error as a function of iteration number. (a) Without preconditioning. (b) With preconditioning.

measure the ℓ_1-distance between the true parameter θ^* and the sample mean $\hat{\theta}_k$ of the random samples obtained from simulating the Markov chains for k iterations:

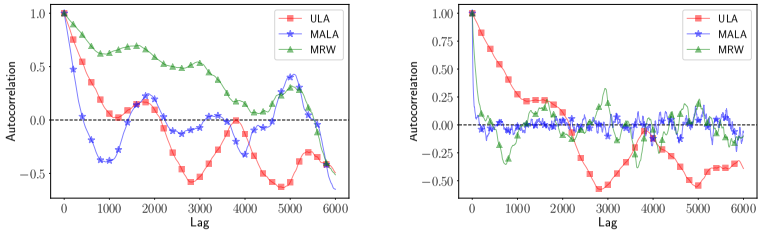
$$e_k = \frac{1}{d} \|\hat{\theta}_k - \theta^*\|_1.$$

Figure 7 provides a log-scale plot of this error versus the iteration number. Since there is always an approximation error caused by the prior distribution, ULA with large step-size ($\delta = 1.0$) can be used. However, our simulation shows that it is still slower than MALA. Also, the condition number κ has a significant effect on the mixing time of ULA and MRW. Their convergence in the preconditioned case is significantly better. Furthermore, the autocorrelation plots in Figure 8 and the plots in Figure 9 of the sample (across experiments) mean and 25% and 75% quantiles, with θ^* subtracted, as a function of iterations suggest a similar story: MALA converges faster than ULA and is less affected by the conditioning of the problem.

4.4. Step size vs accept-reject rate

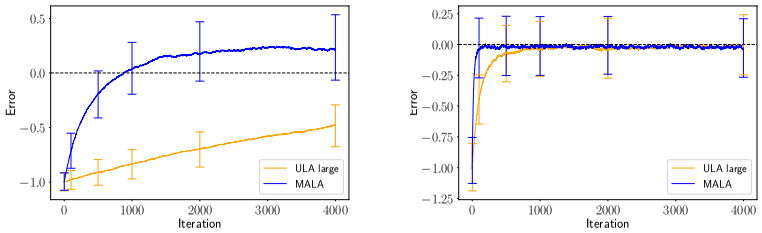
In this section, we provide a few simulations that highlight the effect of step size for MALA and MRW. Note that our bounds from Theorem 1 and 2 suggest a step size choice of order d^{-1} for both MALA and MRW, which in turn led to the mixing time bounds of $\mathcal{O}(d)$. These choices of step sizes arise when we try to provide a worst-case control on the accept-reject step of these algorithms. In particular, these choices ensure that the Markov chains do not get stuck at a given state x, or equivalently, that the proposals at any given state are accepted with constant probability. If instead, one chooses a very large step size, the (worst-case) probability of acceptance may decay exponentially with dimensions. Nonetheless, these worst-case bounds may not hold, which would imply a faster mixing time for these chains if a larger step size were to be used.

To check the validity of larger step sizes, we repeated a few experiments discussed above, albeit with a larger step size. In particular, we simulated the random walks for a wide-range of step sizes $d^{-\gamma}$ for $\gamma \in \{0.2, 0.33, 0.5, 0.67\}$ for MALA, and, $\gamma \in \{0.4, 0.67, 1, 1.33\}$ for



(a) (b)

Figure 8. Autocorrelation function plot of the first coordinate of the estimate as a function lag. The burn-in time for the plot is set to 300 iterations. (a) Without preconditioning. (b) With preconditioning.



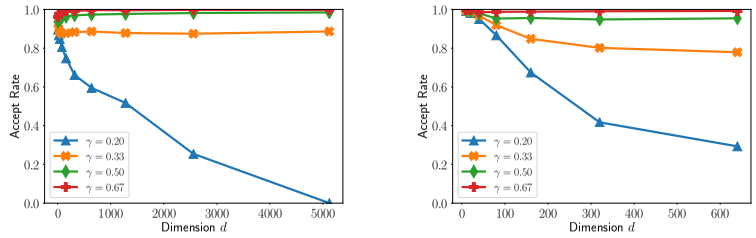
(a) (b)

Figure 9. Mean and 25% and 75% quantiles, with θ^* subtracted, as a function of iteration number. (a) Without preconditioning. (b) With preconditioning.

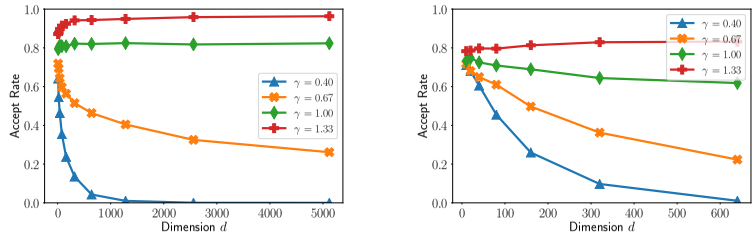
MRW. We ran these chains for two different cases: (a) Sampling from non-isotropic Gaussian density, discussed in Section 4.1, and, (b) Posterior sampling in Bayesian logistic regression, discussed in Section 4.3). In Figure 10, we plot the average acceptance probability for different step sizes $d^{-\gamma}$ as the dimension d increases. These probabilities were computed as the average number of proposals accepted over 100 iterations after a manually tuned burn-in period, and further averaged across 50 independent runs.

We now remark on the observations from Figure 10. We see that for MALA the acceptance probability for the step size choice of $d^{-0.2}$ vanishes as d increases. Indeed, the choice of $d^{-0.5}$ appears to be a safe choice for both cases. In contrast, for MRW, we need a smaller step size. From panels (c) and (d), we see that d^{-1} appears to be the correct choice to ensure that the proposals are accepted with a constant probability when the dimension d is large.

Informally, if a step size choice of $d^{-\gamma}$ was to guarantee a non-vanishing acceptance probability for MALA or MRW, our proof techniques imply a mixing time bound of $\mathcal{O}(d^{\gamma})$.



(a) MALA: Non-isotropic Gaussian (b) MALA: Bayesian logistic regression



(c) MRW: Non-isotropic Gaussian (d) MRW: Bayesian logistic regression

Figure 10. Effect of large step size for accept-reject ratio for MALA and MRW. From panels (a) and (b), we see that for MALA the step size choice of $d^{-0.5}$ has a non-vanishing acceptance probability rate for both cases. On the other hand, panels (c) and (d) show that for MRW d^{-1} is a good choice for the step size.

Combining this argument with the observations above, we suspect that the bounds for MALA from Theorem 1 may not be tight, while for MRW the bounds from Theorem 2 are very likely to be tight. Deriving a faster mixing time for MALA or establishing that the current dimension dependency for MRW is tight, are interesting research directions and we leave the further investigation of these questions for future work.

5. Proofs

We now turn to the proofs of our main results. In Section 5.1, we begin by introducing some background on conductance bounds, before stating three auxiliary lemmas that underlie the proofs of our main theorems. Taking these three lemmas as given, we then provide the proof of Theorem 1 in Section 5.2. Sections 5.3 through 5.5 are devoted to the proofs of our three key lemmas, and we conclude with the proof of Theorem 2 in Section 5.6.

5.1. Conductance bounds and auxiliary results

Our proofs exploit standard conductance-based arguments for controlling mixing times. Consider an ergodic Markov chain defined by a transition operator $\mathcal{T}$, and let Π be its stationary distribution. For each scalar $s \in (0, 1/2)$, we define the s-conductance

$$\Phi_s := \inf_{\Pi(A) \in (s, 1-s)} \frac{\int_A \mathcal{T}_u(A^c)\pi(u)du}{\min\{\Pi(A) - s, \Pi(A^c) - s\}}. \tag{21}$$

In this formula, the notation $\mathcal{T}_u$ is shorthand for the distribution $\mathcal{T}(\delta_u)$ obtained by applying the transition operator to a dirac distribution concentrated on u. In words, the s-conductance measures how much probability mass flows across disjoint sets relative to their stationary mass. By a continuity argument, it can be seen that limiting conductance of the chain is equal to the limiting value of s-conductance—that is, $\Phi = \lim_{s \to 0} \Phi_s$.

For a reversible lazy Markov chain with β-warm start, Lovász 1999 (see also Kannan et al. (1995)) proved that

$$\|\mathcal{T}^k(\mu_0) - \Pi\|_{\text{TV}} \leq \beta s + \beta \left(1 - \frac{\Phi_s^2}{2}\right)^k \leq \beta s + \beta e^{-k\Phi_s^2/2} \quad \text{for any } s \in \left(0, \tfrac{1}{2}\right). \tag{22}$$

In order to make effective use of this lower bound, we need to lower bound the s-conductance Φ_s, and then choose the parameter s so as to optimize the tradeoff between the two terms in the bound. We now state some auxiliary results that are useful.

We start with a result that shows that the probability mass of any strongly log concave distributions is concentrated in a Euclidean ball around the mode. For each $s \in (0, 1)$, we introduce the Euclidean ball

$$\mathcal{R}_s = \mathbb{B}\left(x^\star, r(s)\sqrt{\frac{d}{m}}\right) \tag{23}$$

where the function r was previously defined in equation (9a), and $x^\star := \arg\max_{x \in \mathbb{R}^d} \pi(x)$ denotes the mode.

Lemma 5 *For any $s \in \left(0, \frac{1}{2}\right)$, we have $\Pi(\mathcal{R}_s) \geq 1 - s$.*

See Section 5.3 for the proof of this claim.

In order to establish the conductance bounds inside this ball, we first prove an extension of a result by Lovász (1999). The next result provides a lower bound on the flow of Markov chain with transition distribution $\mathcal{T}_x$ and strongly log concave target distributions Π. Similar results have been used in several prior works to establish fast mixing of several random walks like ball walk, Hit and run (Lovász, 1999; Lovász and Vempala, 2006, 2007), Dikin walk (Narayanan, 2016) and Vaidya and John walks (Chen et al., 2018).

Lemma 6 *Let $\mathcal{K}$ be a convex set such that $\|\mathcal{T}_x - \mathcal{T}_y\|_{TV} \leq 1 - \rho$ whenever $x, y \in \mathcal{K}$ and $\|x - y\|_2 \leq \Delta$. Then for any measurable partition A_1 and A_2 of $\mathbb{R}^d$, we have*

$$\int_{A_1} \mathcal{T}_u(A_2)\pi(u)du \geq \frac{\rho}{4} \min\left\{1, \frac{\log 2 \cdot \Delta \cdot \Pi^2(\mathcal{K}) \cdot \sqrt{m}}{8}\right\} \min\{\Pi(A_1 \cap \mathcal{K}), \Pi(A_2 \cap \mathcal{K})\}. \tag{24}$$

See Section 5.4 for the proof of this lemma.

We next introduce a few pieces of notations to state a MALA specific result. Define a function $\tilde{w} : (0, 1) \times (0, 1) \to \mathbb{R}_+$ as follows:

$$\tilde{w}(s, \epsilon) := \min\left\{\frac{\sqrt{\epsilon}}{8\sqrt{2}r(s)}\frac{\sqrt{m}}{L\sqrt{dL}}, \frac{\epsilon}{64\alpha_\epsilon}\frac{1}{Ld}, \frac{\epsilon^{2/3}}{26(\alpha_\epsilon r^2(s))^{1/3}}\frac{1}{L}\left(\frac{m}{Ld^2}\right)^{1/3}\right\}, \tag{25a}$$

$$\text{where} \quad \alpha_\epsilon := 1 + 2\sqrt{\log(16/\epsilon)} + 2\log(16/\epsilon), \tag{25b}$$

and the function r was defined in equation (9a).

In the next lemma, we show two important properties for MALA: (1) the proposal distributions of MALA at two different points are close if the two points are close, and (2) the accept-reject step of MALA is well behaved inside the ball $\mathcal{R}_s$ provided the step size is chosen carefully. Note that for MALA, the proposal distribution of the chain at x is given by

$$\mathcal{P}_x^{\text{MALA}(h)} = \mathcal{N}(\mu_x, 2h\mathbb{I}_d), \quad \text{where} \quad \mu_x = x - h\nabla f(x). \tag{26}$$

We use $\mathcal{T}_x^{\text{MALA}(h)}$ to denote the transition distribution of MALA.

Lemma 7 *For any step size $h \in \left(0, \frac{2}{L}\right]$, the MALA proposal distribution satisfies the bound*

$$\sup_{\substack{x, y \in \mathbb{R}^d \\ x \neq y}} \frac{\|\mathcal{P}_x^{MALA(h)} - \mathcal{P}_y^{MALA(h)}\|_{TV}}{\|x - y\|_2} \leq \frac{1}{\sqrt{2h}}. \tag{27a}$$

Moreover, given scalars $s \in (0, 1/2)$ and $\epsilon \in (0, 1)$, then the MALA proposal distribution for any step size $h \in \left(0, \tilde{w}(s, \epsilon)\right]$ satisfies the bound

$$\sup_{x \in \mathcal{R}_s} \|\mathcal{P}_x^{MALA(h)} - \mathcal{T}_x^{MALA(h)}\|_{TV} \leq \frac{\epsilon}{8}, \tag{27b}$$

where the truncated ball $\mathcal{R}_s$ was defined in equation (23).

See Section 5.5 for the proof of this claim. With these results in hand, we are now equipped to prove the mixing time bound for MALA.

5.2. Proof of Theorem 1

At a high level, the proof involves three key steps. Our first step is to use Lemma 7 to establish that for an appropriate choice of step size, the MALA update has nice properties inside a high probability region given by Lemma 5. The second step is to apply Lemma 6 so as to obtain a lower bound on the s-conductance Φ_s of the MALA update. Finally, by making an appropriate choice of parameter s, we establish the claimed convergence rate.

In order to simplify notation, we drop the superscripts $\text{MALA}(h)$ from our notation—that is, we use $\mathcal{T}_x$ and $\mathcal{P}_x$, respectively, to denote the transition and proposal distributions at x for MALA, each with step size h. By applying the triangle inequality, we obtain the upper bound

$$\|\mathcal{T}_x - \mathcal{T}_y\|_{\text{TV}} \leq \|\mathcal{P}_x - \mathcal{T}_x\|_{\text{TV}} + \|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}} + \|\mathcal{P}_y - \mathcal{T}_y\|_{\text{TV}}. \tag{28}$$

Now applying claim (27a) from Lemma 7 guarantees that

$$\|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}} \le \epsilon/\sqrt{2} \qquad \text{for all } x, y \in \mathbb{R}^d \text{ such that } \|x-y\|_2 \le \epsilon\sqrt{h}.$$

Furthermore, for any $h \le \tilde{w}(s,\epsilon)$, the bound (27b) from Lemma 7 implies that $\|\mathcal{P}_x - \mathcal{T}_x\|_{\mathrm{TV}} \le \epsilon/8$ for any $x \in \mathcal{R}_s$. Plugging in these bounds in the inequality (28), we find that

$$\|\mathcal{T}_x - \mathcal{T}_y\|_{\mathrm{TV}} \le 1-(1-\epsilon) \quad \forall\, x, y \in \mathcal{R}_s \text{ such that } \|x-y\|_2 \le \epsilon\sqrt{h}.$$

Thus, the transition distribution $\mathcal{T}_x$ satisfies the assumptions of Lemma 6 for

$$\mathcal{K} = \mathcal{R}_s, \quad \rho = (1-\epsilon) \quad \text{and} \quad \Delta = \epsilon\sqrt{h}. \tag{29}$$

We now derive a lower bound on the s-conductance of MALA. Choosing a measurable set A such that $\Pi(A) > s$ and substituting the terms from equation (29) in the inequality (24), we find that

$$\begin{aligned}\int_A \mathcal{T}_u(A^c)\pi(u)du &\ge \frac{(1-\epsilon)}{4}\min\left\{1, \frac{\log 2 \cdot \epsilon\sqrt{h}\cdot \Pi^2(\mathcal{R}_s)\cdot\sqrt{m}}{8}\right\}\cdot \\ &\qquad \cdot \min\left\{\Pi(A\cap\mathcal{R}_s), \Pi(A^c\cap\mathcal{R}_s)\right\} \\ &\overset{(i)}{\ge} \frac{(1-\epsilon)\epsilon\sqrt{h}\cdot\Pi^2(\mathcal{R}_s)\cdot\sqrt{m}}{64}\min\left\{\Pi(A)-s, \Pi(A^c)-s\right\}.\end{aligned}$$

In this argument, inequality (i) follows from the facts that $\log 2 \ge 1/2$ and $\Pi(A), \Pi(A^c) > s$. Moreover, we have applied Lemma 5 to find that $\Pi(\mathcal{R}_s) \ge 1-s$ and hence

$$\Pi(\mathcal{X}\cap\mathcal{R}_s) = \Pi(\mathcal{X}) - \Pi(\mathcal{X}\cap\mathcal{R}_s^c) \ge \Pi(\mathcal{X}) - s \quad \text{for } \mathcal{X} \in \{A, A^c\}.$$

We have also assumed that the second argument of the minimum is less than 1. Applying the definition (21) of Φ_s for MALA, we find that

$$\Phi_s^{\mathrm{MALA}(h)} \ge \frac{(1-\epsilon)\epsilon\cdot\Pi^2(\mathcal{R}_s)\cdot\sqrt{hm}}{64}, \qquad \text{for any } h \le \tilde{w}(s,\epsilon). \tag{30}$$

By making a suitable choice of s, we can now complete the proof. Using Lemma 5, we have that $\Pi(\mathcal{R}_{\delta/2}) \ge 1-\delta/2 \ge 1/2$ for any $\delta \in (0,1)$. Applying the definition (25b) of α_ϵ, we obtain that $\alpha_{1/2} \le 12$. Using this fact and the definitions (9b) and (25a) for the functions $w(\cdot)$ and $\tilde{w}(\cdot,\cdot)$, it is straightforward to verify that $cw(\delta/(2\beta)) \le \tilde{w}(\delta/(2\beta), 1/2)$, for an appropriate choice of universal constant c. Substituting in $s = \delta/(2\beta)$, $\epsilon = 1/2$, and $h = cw(\delta/(2\beta))$, and also making use of the lower bound $\Pi(\mathcal{R}_{\delta/2\beta}) \ge 1/2$ in the bound (30), we find that $\Phi_{\delta/2\beta}^{\mathrm{MALA}(h)} \ge c'\sqrt{mh}$ for some universal constant c'. Using the convergence rate (22), we obtain that

$$\|\mathcal{T}_{\mathrm{MALA}(h)}^k(\mu_0) - \Pi\|_{\mathrm{TV}} \le \beta\frac{\delta}{2\beta} + \beta e^{-kmh/c'} \le \delta \quad \text{for all} \quad k \ge \frac{c'}{mh}\cdot\log\left(\frac{2\beta}{\delta}\right), \tag{31}$$

for a suitably large constant c'. Substituting the expression (9b) for $h = cw(\delta/(2\beta))$, yields the claimed bound on mixing time.

5.3. Proof of Lemma 5

The proof consists of two main steps. First, we establish that the distribution Π is sub-Gaussian, which then guarantees concentration around the mean. Second, we show that the mean and the mode of the distribution Π are not far apart. Combining these two claims yields a high probability region around the mode $x^\star$.

Let x denote the random variable with distribution Π and mean $\bar{x} = \mathbb{E}_{x\sim\Pi}[x]$. We claim that $x - \bar{x}$ is a sub-Gaussian random vector with parameter $1/\sqrt{m}$, meaning that

$$\mathbb{E}_x\left[e^{u^\top(x-\bar{x})}\right] \le e^{\|u\|_2^2/(2m)} \quad \text{for any vector } u \in \mathbb{R}^d.$$

In order to prove this claim, we make use of a result due to Hargé (2004) (Theorem 1.1), which we restate here. Let $y \sim \mathcal{N}(\mu, \Sigma)$ with density e and x be a random variable with density function $q\cdot e$ where q is a log-concave function. Then for any convex function $g : \mathbb{R}^d \mapsto \mathbb{R}$ we have

$$\mathbb{E}_x\left[g(x-\mathbb{E}[x])\right] \le \mathbb{E}_y\left[g(y-\mathbb{E}[y])\right]. \tag{32}$$

From Lemma 8(b) we have that $x \mapsto f(x) - \frac{m}{2}\|x-x^\star\|_2^2$ is a convex function. Thus we can express the density π as the product of a log concave function and the density of a random variable with distribution $\mathcal{N}(x^\star, \mathbb{I}_d/m)$. Letting $y \sim \mathcal{N}(x^\star, \mathbb{I}_d/m)$ and noting that $g(z) := e^{u^\top z}$ is a convex function for each fixed vector u, applying the Hargé bound (32) yields

$$\mathbb{E}_x\left[e^{u^\top(x-\bar{x})}\right] \le \mathbb{E}_y\left[e^{u^\top(y-x^\star)}\right] \overset{(i)}{\le} e^{\|u\|_2^2/2m}.$$

Here inequality (i) follows from the fact that the random vector $y - x^\star$ is sub-Gaussian with parameter $1/\sqrt{m}$.

Using the standard tail bounds for quadratic forms for sub-Gaussian random vectors (e.g., Theorem 1 by Hsu et al. 2012), we find that

$$\mathbb{P}_{x\sim\Pi}\left[\|x-\bar{x}\|_2^2 > \frac{d}{m}\left(1 + 2\sqrt{\frac{t}{d}} + 2\frac{t}{d}\right)\right] \le e^{-t}. \tag{33}$$

Define $\mathcal{B}_1 := \mathbb{B}\left(\bar{x}, \sqrt{\frac{d}{m}}\cdot\tilde{r}(s)\right)$ where $\tilde{r}(s) = 1 + 2\max\left\{\left(\frac{\log(1/s)}{d}\right)^{0.25}, \sqrt{\frac{\log(1/s)}{d}}\right\}$. Observe that $\tilde{r}(s)^2 \ge 1 + 2\sqrt{\frac{\log(1/s)}{d}} + 2\frac{\log(1/s)}{d}$ and consequently the bound (33) implies that $\Pi(\mathcal{B}_1) = \mathbb{P}_{x\sim\Pi}[x \in \mathcal{B}_1] \ge 1-s$. Now applying triangle inequality, we obtain that

$$\mathcal{B}_1 \subseteq \mathbb{B}\left(x^\star, \|\bar{x}-x^\star\|_2 + \sqrt{\frac{d}{m}}\cdot\tilde{r}(s)\right) =: \mathcal{B}_2$$

From Theorem 1 by Durmus et al. (2019), we have that $\mathbb{E}_{x\sim\Pi}\|x-x^\star\|_2^2 \le d/m$. Using Jensen inequality twice, we find that

$$\|\bar{x}-x^\star\|_2 = \|\mathbb{E}_{x\sim\Pi}[x] - x^\star\|_2 \le \mathbb{E}_{x\sim\Pi}\|x-x^\star\|_2 \le \sqrt{\mathbb{E}_{x\sim\Pi}\|x-x^\star\|_2^2} \le \sqrt{\frac{d}{m}}. \tag{34}$$

Noting the relation $r(s) = 1 + \tilde{r}(s)$, we thus obtain that $\|\bar{x}-x^\star\|_2 + \sqrt{\frac{d}{m}}\cdot\tilde{r}(s) \le r(s)\sqrt{\frac{d}{m}}$ and consequently $\mathcal{B}_1 \subseteq \mathcal{B}_2 \subseteq \mathcal{R}_s$. As a result, we obtain $\Pi(\mathcal{R}_s) \ge \Pi(\mathcal{B}_1) \ge 1-s$ as claimed.

5.4. Proof of Lemma 6

The proof of this lemma is based on the ideas employed in prior works to establish conductance bounds, first for Hit-and-run (Lovász, 1999), and since then for several other random walks (Lovász and Vempala, 2007; Narayanan, 2016; Chen et al., 2018). See the survey by Vempala (2005) for further details.

For our setting, a key ingredient is the following isoperimetric inequality for log-concave distributions. Let $\mathbb{R}^d = S_1 \cup S_2 \cup S_3$ be a partition. Let $y \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_d)$ with density e and let Π be a distribution with a density given by $q \cdot e$ where q is a log-concave function. Then Cousins and Vempala (2014) (Theorem 4.4) proved that

$$\Pi(S_3) \geq \frac{\log 2 \cdot d(S_1, S_2)}{\sigma} \Pi(S_1)\Pi(S_2) \tag{35}$$

where $d(S_1, S_2) := \inf \left\{ \|x - y\|_2 \,\middle|\, x \in S_1, y \in S_2 \right\}$.

We invoke this result for the truncated distribution $\Pi_{\mathcal{K}}$ with the density $\pi_{\mathcal{K}}$ defined as

$$\pi_{\mathcal{K}}(x) := \frac{1}{\int_{\mathcal{K}} \pi(y) dy} \pi(x) \mathbb{1}_{\mathcal{K}}(x) = \frac{1}{\int_{\mathcal{K}} e^{-f(y)} dy} e^{-f(x)} \mathbb{1}_{\mathcal{K}}(x), \tag{36}$$

where $\mathbb{1}_{\mathcal{K}}(\cdot)$ denotes the indicator function for the set $\mathcal{K}$, i.e., we have $\mathbb{1}_{\mathcal{K}}(x) = 1$ if $x \in \mathcal{K}$, and 0 otherwise. Let $x^\star = \arg\max \pi(x) = \arg\min f(x)$. Observe that m-strong-convexity of f implies that $x \mapsto f(x) - \frac{m}{2} \|x - x^\star\|_2^2$ is a convex function (Lemma 8(b)). Noting that the function $\mathbb{1}_{\mathcal{K}}(\cdot)$ is log-concave and that log-concavity is closed under multiplication, we conclude that $\pi_{\mathcal{K}}$ can be expressed as a product of log-concave function and density of the Gaussian distribution $\mathcal{N}\left(x^\star, \frac{1}{m}\mathbb{I}_d\right)$. Consequently, we can apply the result (35) with Π replaced by $\Pi_{\mathcal{K}}$ and $\sigma = 1/\sqrt{m}$.

We now prove the claim of the lemma. Define the sets

$$A_1' := \left\{ u \in A_1 \cap \mathcal{K} \;\middle|\; \mathcal{T}_u(A_2) < \frac{\rho}{2} \right\}, \quad A_2' := \left\{ v \in A_2 \cap \mathcal{K} \;\middle|\; \mathcal{T}_v(A_1) < \frac{\rho}{2} \right\}, \tag{37}$$

along with the complement $A_3' := \mathcal{K} \backslash (A_1' \cup A_2')$. See Figure 11 for an illustration. Based on these three sets, we split our proof of the claim (24) into two distinct cases:

- Case 1: $\Pi(A_1') \leq \Pi(A_1 \cap \mathcal{K})/2$ or $\Pi(A_2') \leq \Pi(A_2 \cap \mathcal{K})/2$.
- Case 2: $\Pi(A_i') \geq \Pi(A_i \cap \mathcal{K})/2$ for $i = 1, 2$.

Note that these cases are mutually exclusive, and cover all possibilities.

Case 1 We have $\Pi(A_1 \cap \mathcal{K} \backslash A_1') \geq \Pi(A_1 \cap \mathcal{K})/2$, then

$$\begin{aligned} \int_{A_1} \mathcal{T}_u(A_2)\pi(u)du &\overset{(i)}{\geq} \int_{A_1 \cap \mathcal{K} \backslash A_1'} \mathcal{T}_u(A_2)\pi(u)du \overset{(ii)}{\geq} \frac{\rho}{2}\Pi(A_1 \cap \mathcal{K} \backslash A_1') \\ &\overset{(iii)}{\geq} \frac{\rho}{4}\Pi(A_1 \cap \mathcal{K}), \end{aligned}$$

which implies the claim (24). In the above sequence of inequalities, step (i) is trivially true; step (ii) from the definition (37) of the set A_1', and step (iii) from the assumption for this case.

A similar argument with the roles of A_1 and A_2 switched, establishes the claim when $\Pi(A_2') \leq \Pi(A_2 \cap \mathcal{K})/2$.

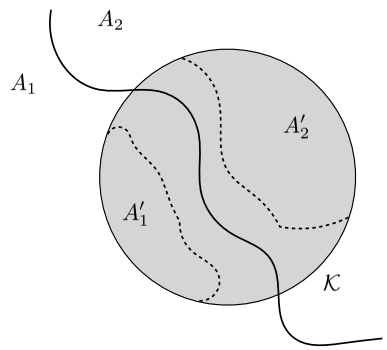


Figure 11. The sets A_1 and A_2 form a partition of $\mathbb{R}^d$, and we use $\mathcal{K}$ to denote a compact convex subset. The sets A_1' and A_2' are defined in equation (37).

Case 2 We have $\Pi(A_i') \geq \Pi(A_i \cap \mathcal{K})/2$ for both $i = 1$ and 2. For any $u \in A_1'$ and $v \in A_2'$, we have that

$$\|\mathcal{T}_u - \mathcal{T}_v\|_{\text{TV}} \geq \mathcal{T}_u(A_1) - \mathcal{T}_v(A_1) \overset{(i)}{=} 1 - \mathcal{T}_u(A_2) - \mathcal{T}_v(A_1) > 1 - \rho,$$

where step (i) follows from the fact that $A_1 = \mathbb{R}^d \backslash A_2$ and thereby $\mathcal{T}_u(A_1) = 1 - \mathcal{T}_u(A_2)$. Since $u, v \in \mathcal{K}$, the assumption of the lemma implies that $\|u - v\|_2 \geq \Delta$ and consequently

$$d(A_1', A_2') \geq \Delta. \tag{38}$$

We claim that

$$\int_{A_1} \mathcal{T}_u(A_2)\pi(u)du = \int_{A_2} \mathcal{T}_v(A_1)\pi(v)dv \tag{39}$$

We provide the proof of this claim at the end. Assuming this claim as given, we now complete the proof. Using equation (39), we have

$$\begin{aligned} \int_{A_1} \mathcal{T}_u(A_2)\pi(u)du &= \frac{1}{2}\left(\int_{A_1} \mathcal{T}_u(A_2)\pi(u)du + \int_{A_2} \mathcal{T}_v(A_1)\pi(v)dv \right) \\ &\geq \frac{1}{4}\left(\int_{A_1 \cap \mathcal{K} \backslash A_1'} \mathcal{T}_u(A_2)\pi(u)du + \int_{A_2 \cap \mathcal{K} \backslash A_2'} \mathcal{T}_v(A_1)\pi(v)dv \right) \\ &\overset{(i)}{\geq} \frac{\rho}{8}\Pi(\mathcal{K} \backslash (A_1' \cup A_2')), \end{aligned} \tag{40}$$

where step (i) follows from the definition (37) of the set $A_3' = \mathcal{K}\backslash(A_1' \cup A_2')$. Further, we have

$$\begin{aligned}\Pi(\mathcal{K}\backslash(A_1' \cup A_2')) &\overset{(i)}{=} \Pi(\mathcal{K}) \cdot \Pi_{\mathcal{K}}(\mathcal{K}\backslash A_1'\backslash A_2') \\ &\overset{(ii)}{\geq} \Pi(\mathcal{K}) \cdot \frac{\log 2 \cdot d(A_1', A_2')}{1/\sqrt{m}} \cdot \Pi_{\mathcal{K}}(A_1') \cdot \Pi_{\mathcal{K}}(A_2') \\ &\overset{(iii)}{\geq} \Pi(\mathcal{K}) \cdot \log 2 \cdot d(A_1', A_2') \cdot \sqrt{m} \cdot \Pi(A_1') \cdot \Pi(A_2') \\ &\overset{(iv)}{\geq} \Pi(\mathcal{K}) \cdot \log 2 \cdot \Delta \cdot \sqrt{m} \cdot \frac{1}{4} \cdot \Pi(A_1 \cap \mathcal{K}) \cdot \Pi(A_2 \cap \mathcal{K}). \end{aligned} \tag{41}$$

where step (i) follows from the definition (36) of the truncated distribution $\Pi_{\mathcal{K}}$, step (ii) follows from applying the isoperimetry (35) for the distribution $\Pi_{\mathcal{K}}$ with $\sigma = 1/\sqrt{m}$, step (iii) from the definition of $\Pi_{\mathcal{K}}$ and step (iv) from inequality (38) and the assumption for this case. Let $\alpha := \Pi(A_1 \cap \mathcal{K})/\Pi(\mathcal{K})$. Note that $\alpha \in [0, 1]$ and $\Pi(A_2 \cap \mathcal{K})/\Pi(\mathcal{K}) = 1 - \alpha$. We have

$$\begin{aligned}\Pi(A_1 \cap \mathcal{K}) \cdot \Pi(A_2 \cap \mathcal{K}) &= \Pi^2(\mathcal{K}) \cdot \alpha(1-\alpha) \\ &\geq \Pi^2(\mathcal{K}) \cdot \frac{1}{2} \min\{\alpha, 1-\alpha\} \\ &= \Pi(\mathcal{K}) \cdot \frac{1}{2} \min\{\Pi(A_1 \cap \mathcal{K}), \Pi(A_2 \cap \mathcal{K})\} \end{aligned} \tag{42}$$

Putting the inequalities (40), (41) and (42) together, establishes the claim (24) of the lemma for this case.

We now prove our earlier claim (39). Note that it suffices to prove that

$$\int_{A_1} \mathcal{T}_u(A_2)\pi(u)du = \int_{A_2} \mathcal{T}_v(A_1)\pi(v)dv.$$

We have

$$\begin{aligned}\int_{A_2} \mathcal{T}_u(A_1)\pi(u)du &\overset{(i)}{=} \int_{\mathbb{R}^d} \mathcal{T}_u(A_1)\pi(u)du - \int_{A_1} \mathcal{T}_u(A_1)\pi(u)du \\ &\overset{(ii)}{=} \Pi(A_1) - \int_{A_1} \mathcal{T}_u(A_1)\pi(u)du \\ &= \int_{A_1} \pi(u)du - \int_{A_1} \mathcal{T}_u(A_1)\pi(u)du \\ &\overset{(iii)}{=} \int_{A_1} \mathcal{T}_u(A_2)\pi(u)du, \end{aligned}$$

where steps (i) and (iii) (respectively) follow from the fact that $A_1 = \mathbb{R}^d\backslash A_2$ and the consequent fact that $1 - \mathcal{T}_u(A_1) = \mathcal{T}_u(A_2)$, and step (ii) follows from the fact that π is the stationary density for the transition distribution $\mathcal{T}_x$ and thereby $\int_{\mathbb{R}^d} \mathcal{T}_u(A_1)\pi(u)du = \Pi(A_1)$.

5.5. Proof of Lemma 7

We prove each claim of the lemma separately. To simplify notation, we drop the superscript from our notations of distributions $\mathcal{T}_x^{\text{MALA}(h)}$ and $\mathcal{P}_x^{\text{MALA}(h)}$.

5.5.1. Proof of claim (27a)

In order to bound the total variation distance $\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}}$, we apply Pinsker's inequality (Cover and Thomas, 1991), which guarantees that $\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}} \leq \sqrt{2\,\text{KL}(\mathcal{P}_x\|\mathcal{P}_y)}$. Given multivariate normal distributions $\mathcal{G}_1 = \mathcal{N}(\mu_1, \Sigma)$ and $\mathcal{G}_2 = \mathcal{N}(\mu_2, \Sigma)$, the Kullback-Leibler divergence between the two is given by

$$\text{KL}(\mathcal{G}_1\|\mathcal{G}_2) = \frac{1}{2}(\mu_1 - \mu_2)^\top \Sigma^{-1}(\mu_1 - \mu_2). \tag{43}$$

Substituting $\mathcal{G}_1 = \mathcal{P}_x$ and $\mathcal{G}_2 = \mathcal{P}_y$ into the above expression and applying Pinsker's inequality, we find that

$$\begin{aligned}\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}} \leq \sqrt{2\,\text{KL}(\mathcal{P}_x\|\mathcal{P}_y)} &= \frac{\|\mu_x - \mu_y\|_2}{\sqrt{2h}} \\ &\overset{(i)}{=} \frac{\|(x - h\nabla f(x)) - (y - h\nabla f(y))\|_2}{\sqrt{2h}}, \end{aligned}$$

where step (i) follows from the definition (26) of the mean μ_x. Consequently, in order to establish the claim (27a), it suffices to show that

$$\|(x - h\nabla f(x)) - (y - h\nabla f(y))\|_2 \leq \|x - y\|_2.$$

Recalling that $\|B\|_{\text{op}}$ denotes the ℓ_2-operator norm of a matrix B (equal to the maximum singular value), we have

$$\begin{aligned}\|(x - h\nabla f(x)) - (y - h\nabla f(y))\|_2 &= \left\| \int_0^1 \left[\mathbb{I} - h\nabla^2 f(x + t(x-y))\right](x-y)dt \right\|_2 \\ &\leq \int_0^1 \left\| \left[\mathbb{I} - h\nabla^2 f(x + t(x-y))\right](x-y) \right\|_2 dt \\ &\overset{(i)}{\leq} \sup_{z \in \mathbb{R}^d} \|\mathbb{I}_d - h\nabla^2 f(z)\|_{\text{op}} \, \|x - y\|_2, \end{aligned}$$

where step (i) follows from the definition of the operator norm. Lemma 8(f) and Lemma 9(f) guarantee that the Hessian is sandwiched as $m\mathbb{I}_d \preceq \nabla^2 f(z) \preceq L\mathbb{I}_d$ for all $z \in \mathbb{R}^d$, where $\mathbb{I}_d$ denotes the d-dimensional identity matrix. From this Hessian sandwich, it follows that

$$\|\mathbb{I}_d - h\nabla^2 f(x)\|_{\text{op}} = \max\{|1 - hL|, |1 - hm|\} < 1.$$

Putting together the pieces yields the claim.

5.5.2. Proof of claim (27b)

Let $\mathcal{P}_1$ be a distribution admitting a density p_1 on $\mathbb{R}^d$, and let $\mathcal{P}_2$ be a distribution which has an atom at x and admitting a density p_2 on $\mathbb{R}^d \backslash \{x\}$. The total variation distance between the distributions $\mathcal{P}_1$ and $\mathcal{P}_2$ is given by

$$\|\mathcal{P}_1 - \mathcal{P}_2\|_{\text{TV}} = \frac{1}{2}\left(\mathcal{P}_2(\{x\}) + \int_{\mathbb{R}^d} |p_1(z) - p_2(z)|\, dz\right). \tag{44}$$

The accept-reject step for MALA implies that

$$\mathcal{T}_x(\{x\}) = 1 - \int_{\mathbb{R}^d} \min\left\{1, \frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)}\right\} p_x(z)dz, \tag{45}$$

where p_x denotes the density corresponding to the proposal distribution $\mathcal{P}_x = \mathcal{N}(x - h\nabla f(x), 2h\mathbb{I}_d)$. From this fact and the formula (44), we find that

$$\begin{aligned}\|\mathcal{P}_x - \mathcal{T}_x\|_{\mathrm{TV}} &= \frac{1}{2}\left(\mathcal{T}_x(\{x\}) + \int_{\mathbb{R}^d} p_x(z)dz - \int_{\mathbb{R}^d} \min\left\{1, \frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)}\right\} p_x(z)dz\right)\\ &= \frac{1}{2}\left(2 - 2\int_{\mathbb{R}^d} \min\left\{1, \frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)}\right\} p_x(z)dz\right)\\ &= 1 - \mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)}\right\}\right].\end{aligned} \tag{46}$$

By applying Markov's inequality, we obtain

$$\mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)}\right\}\right] \geq \alpha\, \mathbb{P}\left[\frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)} \geq \alpha\right] \quad \text{for all } \alpha \in (0,1]. \tag{47}$$

We now derive a high probability lower bound for the ratio $[\pi(z)p_z(x)] \,/\, [\pi(x)p_x(z)]$. Noting that $\pi(x) \propto \exp(-f(x))$ and $p_x(z) \propto \exp\left(-\|x - h\nabla f(x) - z\|_2^2 / (4h)\right)$, we have

$$\begin{aligned}\frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)} &= \frac{\exp\left(-f(z) - \frac{\|x-z+h\nabla f(z)\|_2^2}{4h}\right)}{\exp\left(-f(x) - \frac{\|z-x+h\nabla f(x)\|_2^2}{4h}\right)}\\ &= \exp\left(\frac{4h(f(x)-f(z)) + \|z-x+h\nabla f(x)\|_2^2 - \|x-z+h\nabla f(z)\|_2^2}{4h}\right).\end{aligned} \tag{48}$$

Keeping track of the numerator of this exponent, we find that

$$\begin{aligned}&4h(f(x)-f(z)) + \|z-x+h\nabla f(x)\|_2^2 - \|x-z+h\nabla f(z)\|_2^2\\ &\quad= 4h(f(x)-f(z)) + \|z-x\|_2^2 + \|h\nabla f(x)\|_2^2 + 2h(z-x)^\top\nabla f(x)\\ &\qquad - \|x-z\|_2^2 - \|h\nabla f(z)\|_2^2 - 2h(x-z)^\top\nabla f(z)\\ &\quad= 2h\underbrace{\left(f(x)-f(z)-(x-z)^\top\nabla f(x)\right)}_{M_1} + 2h\underbrace{\left(f(x)-f(z)-(x-z)^\top\nabla f(z)\right)}_{M_2}\\ &\qquad + h^2\underbrace{\left(\|\nabla f(x)\|_2^2 - \|\nabla f(z)\|_2^2\right)}_{M_3}.\end{aligned} \tag{49}$$

Now we provide lower bounds for the terms M_i, $i = 1,2,3$ defined in the above display. Since f is strongly convex and smooth, applying Lemma 8(c) and Lemma 9(c) yields

$$M_1 \geq -\frac{L}{2}\|x-z\|_2^2, \quad \text{and} \quad M_2 \geq \frac{m}{2}\|x-z\|_2^2. \tag{50}$$

In order to lower bound M_3, we observe that

$$\begin{aligned}M_3 = \|\nabla f(x)\|_2^2 - \|\nabla f(z)\|_2^2 &= \langle \nabla f(x) + \nabla f(z), \nabla f(x) - \nabla f(z)\rangle\\ &\overset{(i)}{\geq} -\|\nabla f(x) + \nabla f(z)\|_2 \|\nabla f(x) - \nabla f(z)\|_2\\ &\overset{(ii)}{\geq} -\left(2\|\nabla f(x)\|_2 + L\|x-z\|_2\right) L\|x-z\|_2,\end{aligned} \tag{51}$$

where step (i) follows from the Cauchy-Schwarz's inequality and step (ii) from the triangle inequality and L-smoothness of the function f (cf. Lemma 9(d)).

Combining the bounds (50) and (51) with equations (49) and (48), we have established that

$$\frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)} \geq \exp\left(\underbrace{-\frac{1}{4}(L-m)\|x-z\|_2^2 - \frac{h}{4}\left(2L\|x-z\|_2\|\nabla f(x)\|_2 + L^2\|x-z\|_2^2\right)}_{=:T}\right). \tag{52}$$

Now to provide a high probability lower bound for the term T, we make use of the standard chi-squared tail bounds and the following relation between x and z:

$$z \overset{(d)}{=} x - h\nabla f(x) + \sqrt{2h}\xi,$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$ and $\overset{(d)}{=}$ denotes equality in distribution. We have

$$\|x-z\|_2 = \left\|h\nabla f(x) + \sqrt{2h}\xi\right\|_2 \leq h\|\nabla f(x)\|_2 + \sqrt{2h}\|\xi\|_2,$$

which also implies

$$\|x-z\|_2^2 \leq 2h^2\|\nabla f(x)\|_2^2 + 4h\|\xi\|_2^2.$$

Using these two inequalities, we find that

$$\begin{aligned}T \geq &-\frac{(L-m)h^2}{2}\|\nabla f(x)\|_2^2 - (L-m)h\|\xi\|_2^2 - \frac{Lh^2}{2}\|\nabla f(x)\|_2^2\\ &- \frac{Lh\sqrt{h}}{\sqrt{2}}\|\nabla f(x)\|_2\|\xi\|_2 - \frac{L^2h^3}{2}\|\nabla f(x)\|_2^2 - L^2h^2\|\xi\|_2^2.\end{aligned}$$

Simplifying and using the fact that $Lh \leq 1$, we obtain that

$$T \geq -2\left(Lh^2\|\nabla f(x)\|_2^2 + Lh\|\xi\|_2^2 + Lh\sqrt{h}\|\nabla f(x)\|_2\|\xi\|_2\right).$$

Since $x \in \mathcal{R}_s$, we have

$$\|\nabla f(x)\|_2 = \|\nabla f(x) - \nabla f(x^\star)\|_2 \overset{(i)}{\leq} L\|x - x^\star\|_2 \leq L\sqrt{\frac{d}{m}}r(s) =: \mathcal{D}_s, \tag{53}$$

where inequality (i) follows from the property (d) of Lemma 9. Thus, we have shown that

$$T \geq -2\left(Lh^2\mathcal{D}_s^2 + Lh\left\|\xi\right\|_2^2 + Lh\sqrt{h}\mathcal{D}_s\left\|\xi\right\|_2\right). \tag{54}$$

Standard tail bounds for χ^2-variables guarantee that

$$\mathbb{P}\left[\left\|\xi\right\|_2^2 \leq d\alpha_\epsilon\right] \geq (1-\epsilon/16) \text{ for } \alpha_\epsilon = 1 + 2\sqrt{\log(16/\epsilon)} + 2\log(16/\epsilon).$$

A simple observation reveals that the function $\tilde{w}$ defined in equation (25a) was chosen such that for any $h \leq \tilde{w}(s,\epsilon)$, we have

$$Lh^2\mathcal{D}_s^2 \leq \frac{\epsilon}{128}, \quad Lhd\alpha_\epsilon \leq \frac{\epsilon}{64}, \quad \text{and,} \quad Lh\sqrt{h}\mathcal{D}_s\sqrt{d\alpha_\epsilon} \leq \frac{\epsilon}{128}.$$

Combining this observation with the high probability bound on $\left\|\xi\right\|_2$ and using the inequality (54) we obtain that $T \geq -\epsilon/16$ with probability at least $1-\epsilon/16$. Plugging this bound in the inequality (52), we find that

$$\mathbb{P}\left[\frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)} \geq \exp\left(-\frac{\epsilon}{16}\right)\right] \geq (1-\epsilon/16).$$

Thus, we have derived a desirable high probability lower bound on the accept-reject ratio. Substituting $\alpha = \exp(-\epsilon/16)$ in the inequality (47) and using the fact that $e^{-\epsilon/16} \geq 1-\epsilon/16$ for any scalar $\epsilon > 0$, we find that

$$\mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\pi(z)\cdot p_z(x)}{\pi(x)\cdot p_x(z)}\right\}\right] \geq 1 - \frac{\epsilon}{8}, \quad \text{for any } \epsilon \in (0,1) \text{ and } h \leq \tilde{w}(s,\epsilon).$$

Substituting this bound in the inequality (46) completes the proof.

5.6. Proof of Theorem 2

The proof of this theorem is similar to the proof of Theorem 1. We begin by claiming that

$$\|\mathcal{P}_x^{\mathrm{MRW}(h)} - \mathcal{P}_y^{\mathrm{MRW}(h)}\|_{\mathrm{TV}} = \frac{\epsilon}{\sqrt{2}} \quad \text{for all } x, y \text{ such that } \|x-y\|_2 \leq \epsilon\sqrt{h} \tag{55a}$$

$$\|\mathcal{P}_x^{\mathrm{MRW}(h)} - \mathcal{T}_x^{\mathrm{MRW}(h)}\|_{\mathrm{TV}} = \frac{\epsilon}{8} \quad \text{for all } x \in \mathcal{R}_s, \tag{55b}$$

for any $h \leq c\epsilon^2 m/(\alpha_\epsilon d L^2 r(s))$ for some universal constant c. Plugging $s = \delta/(2\beta)$, $\epsilon = 1/2$ and arguing as in Section 5.2, we find that $\Phi_{\delta/2\beta}^{\mathrm{MRW}(h)} \geq c'\sqrt{mh}$ for some universal constant c'. Using the convergence rate (22), we obtain that

$$\|\mathcal{T}_{\mathrm{MRW}(h)}^k(\mu_0) - \Pi\|_{\mathrm{TV}} \leq \beta\frac{\delta}{2\beta} + \beta e^{-kmh/c'} \leq \delta \quad \text{for all} \quad k \geq \frac{c'}{mh}\cdot\log\left(\frac{2\beta}{\delta}\right), \tag{56}$$

for a suitably large constant c'. Substituting $h \leq cm/\left(dL^2 r(\delta/2\beta)\right)$, yields the claimed bound on mixing time of MRW.

It is now left to establish our earlier claims (55a) and (55b). Note that $\mathcal{P}_x^{\mathrm{MRW}(h)} = \mathcal{N}(x, 2h\mathbb{I}_d)$. For brevity, we drop the superscripts from our notations. Using the expression (43) for the KL-divergence and applying Pinsker's inequality leads to the upper bound

$$\|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}} \leq \sqrt{2\,\mathrm{KL}(\mathcal{P}_x\|\mathcal{P}_y)} = \frac{\|x-y\|_2}{\sqrt{2h}},$$

which implies the claim (55a).

We now prove the bound (55b). Letting p_x to denote the density of the proposal distribution $\mathcal{P}_x$ and using the bounds (46) and (47), it suffices to prove that

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\frac{\pi(z)}{\pi(x)} \geq \exp\left(-\frac{\epsilon}{16}\right)\right] \overset{(i)}{=} \mathbb{P}_{z\sim\mathcal{P}_x}\left[f(x) - f(z) \geq -\frac{\epsilon}{16}\right] \geq (1-\epsilon/16), \tag{57}$$

where step (i) follows from the fact that $\pi(x) \propto e^{-f(x)}$. We have

$$\begin{aligned} f(x) - f(z) &\overset{(i)}{\geq} \nabla f(z)^\top(x-z) = (\nabla f(z) - \nabla f(x))^\top(x-z) + \nabla f(x)^\top(x-z) \\ &\overset{(ii)}{\geq} -L\left\|x-z\right\|_2^2 + \nabla f(x)^\top(x-z) \\ &= -2Lh\left\|\xi\right\|_2^2 + \sqrt{2h}\nabla f(x)^\top\xi \end{aligned} \tag{58}$$

where the step (i) follows from the convexity of the function f, step (ii) the smoothness of the function f (Lemma 9(e)). Note that the random variable $\chi := \nabla f(x)^\top\xi \sim \mathcal{N}(0, \|\nabla f(x)\|_2^2)$ and that $\|\nabla f(x)\|_2 \leq \mathcal{D}_s$ for any $x \in \mathcal{R}_s$. Consequently, we have $\chi \geq -\mathcal{D}_s\cdot 2\sqrt{\log(32/\epsilon)}$ with probability at least $1-\epsilon/32$. On the other hand, using the standard tail bound for a Chi-squared random variable, we obtain that $\mathbb{P}\left[\left\|\xi\right\|_2^2 \geq d\alpha_\epsilon\right] \leq \epsilon/32$ for $\alpha_\epsilon = 1 + 2\sqrt{\log(32/\epsilon)} + 2\log(32/\epsilon)$. Recalling that $\mathcal{D}_s = L\sqrt{\frac{d}{m}}r(s)$ and doing straightforward calculation reveals that for $h \leq \frac{\epsilon^2}{(8192\alpha_\epsilon d\frac{L^2}{m}r(s))}$, we have

$$2Lhd\alpha_\epsilon \leq \frac{\epsilon}{64} \quad \text{and} \quad \sqrt{2h}\mathcal{D}_s 2\sqrt{\log(32/\epsilon)} \leq \frac{3\epsilon}{64}$$

Combining these bounds with the high probability statements above and plugging in the inequality (58), we find that $f(x) - f(z) \geq -\epsilon/16$ with probability at least $1-\epsilon/16$, which yields the claim (57).

6. Discussion

In this paper, we derived non-asymptotic bounds on the mixing time of the Metropolis adjusted Langevin algorithm and Metropolized random walk for log-concave distributions. These algorithms are based on a two-phase scheme: (1) a proposal step followed by (2) an accept-reject step. Our results show that the accept-reject step while leading to significant complications in the analysis is practically very useful: algorithms applying this step mix significantly faster than the ones without it. In particular, we showed that for a strongly log-concave distribution in $\mathbb{R}^d$ with condition number κ, the δ-mixing time for MALA is

of $\mathcal{O}(d\kappa \log(1/\delta))$. This guarantee significantly better than the $\mathcal{O}\left(d\kappa^2/\delta^2\right)$ mixing time for ULA established in the literature. We also proposed a modified version of MALA to sample from non-strongly log-concave distributions and showed that it mixes in $\mathcal{O}\left(d^2/\delta^{1.5}\right)$; thus, this algorithm dependency on the desired accuracy δ when compared to the $\mathcal{O}\left(d^3/\delta^4\right)$ mixing time for ULA for the same task. Furthermore, we established $\mathcal{O}\left(d\kappa^2 \log(1/\delta)\right)$ mixing time bound for the Metropolized random walk for log-concave sampling.

Several fundamental questions arise from our work. All of our results are upper bounds on mixing time, and our simulation results suggest that they are tight for the choice of step size used in the Theorem statements. However, simulations from Section 4.4 suggest that warmness parameter β should not affect the choice of step size too much and hence potentially larger choices of step sizes (and thereby faster mixing) are possible. To this end, in a recent pre-print (Chen et al., 2019), we established faster mixing time bounds for MALA and MRW from a non-warm start where we show that the dependence on warmness can be improved from $\log \beta$ to $\log \log \beta$. Moreover, for a deterministic start, one may consider running ULA for a few steps run to obtain moderate accuracy, and then run MALA initialized with the ULA iterates (thereby providing a warm start to MALA). In practice, we find that this hybrid procedure can generate highly accurate samples in reasonably few number of iterations.

Another open question is to sharply delineate the fundamental gap between the mixing times of first-order sampling methods and that of zeroth-order sampling methods. Noting that MALA is a first-order method while MRW is a zeroth-order method, from our work, we obtain that two class of methods differ in a factor of the condition number κ of the target distribution. It is an interesting question to determine whether this represents a sharp gap between these two classes of sampling methods.

The current state-of-the-art algorithm, namely Hamiltonian Monte Carlo (Neal, 2011) can be seen as a multi-step generalization of MALA. Instead of centering the proposal after one gradient-step, HMC simulates an ODE in an augmented space for a few time steps. Indeed, MALA is equivalent to a particular one-step discretization of the ODE associated with HMC. Nonetheless, the practically used HMC makes use of multi-step discretization and is more involved than MALA. Empirically HMC has proven to have superior mixing times for a broad class of distributions (and not just log-concave distributions). A line of recent work (Bou-Rabee et al., 2018; Mangoubi and Smith, 2017; Mangoubi and Vishnoi, 2018) provides theoretical guarantees for HMC in different settings. Several of these works analyze an idealized version of HMC or the discretized version without the accept-reject step. In a recent pre-print (Chen et al., 2019), we have provided some theoretical guarantees on the convergence of Metropolized HMC, which is the most practical version of HMC.

Acknowledgements

This work was supported by Office of Naval Research grant DOD ONR-N00014 to MJW, and by ARO W911NF1710005, NSF-DMS 1613002 and the Center for Science of Information (CSoI), US NSF Science and Technology Center, under grant agreement CCF-0939370 to BY. In addition, MJW was partially supported by National Science Foundation grant NSF-DMS-1612948, and RD was partially supported by the Berkeley Fellowship.

Appendix A. Some basic properties

In this appendix, we state a few basic properties of strongly-convex and smooth functions that we use in our proofs. See the book (Boyd and Vandenberghe, 2004) for more details.

Lemma 8 (Equivalent characterizations of strong convexity) *For a twice differentiable convex function $f : \mathbb{R}^d \mapsto \mathbb{R}$, the following statements are equivalent:*

(a) The function f is m-strongly-convex.

(b) The function $x \mapsto f(x) - \frac{m}{2}\|x - x^\star\|_2^2$ is convex (for any fixed point $x^\star$).

(c) For any $x, y \in \mathbb{R}^d$, we have

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{m}{2}\|x - y\|_2^2 .$$

(d) For any $x, y \in \mathbb{R}^d$, we have

$$\|\nabla f(x) - \nabla f(y)\|_2 \geq m \|x - y\|_2 .$$

(e) For any $x, y \in \mathbb{R}^d$, we have

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq m \|x - y\|_2^2 .$$

(f) For any $x \in \mathbb{R}^d$, the Hessian is lower bounded as $\nabla^2 f(x) \succeq m\mathbb{I}_d$.

Lemma 9 (Equivalent characterizations of smoothness) *For a twice differentiable convex function $f : \mathbb{R}^d \mapsto \mathbb{R}$, the following statements are equivalent:*

(a) The function f is L-smooth.

(b) The function $x \mapsto \frac{L}{2}\|x - x^\star\|_2^2 - f(x)$ is convex (for any fixed point $x^\star$).

(c) For any $x, y \in \mathbb{R}^d$, we have

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|x - y\|_2^2 .$$

(d) For any $x, y \in \mathbb{R}^d$, we have

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2 .$$

(e) For any $x, y \in \mathbb{R}^d$, we have

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \leq L \|x - y\|_2^2 .$$

(f) For any $x \in \mathbb{R}^d$, the Hessian is upper bounded as $\nabla^2 f(x) \preceq L\mathbb{I}_d$.

References

Claude JP Bélisle, H Edwin Romeijn, and Robert L Smith. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18(2):255–266, 1993.

Nawaf Bou-Rabee and Martin Hairer. Nonasymptotic mixing of the MALA algorithm. *IMA Journal of Numerical Analysis*, 33(1):80–110, 2012.

Nawaf Bou-Rabee, Andreas Eberle, and Raphael Zimmer. Coupling and convergence for Hamiltonian Monte Carlo. *arXiv preprint arXiv:1805.00452*, 2018.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Steve Brooks, Andrew Gelman, Galin L Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

Sébastien Bubeck. Convex optimization: algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

Sébastien Bubeck, Ronen Eldan, and Joseph Lehec. Sampling from a log-concave distribution with projected Langevin Monte Carlo. *Discrete & Computational Geometry*, 59(4): 757–783, 2018.

Yuansi Chen, Raaz Dwivedi, Martin J Wainwright, and Bin Yu. Fast MCMC sampling algorithms on polytopes. *The Journal of Machine Learning Research*, 19(1):2146–2231, 2018.

Yuansi Chen, Raaz Dwivedi, Martin J. Wainwright, and Bin Yu. Fast mixing of Metropolized Hamiltonian Monte Carlo: Benefits of multi-step gradients. *arXiv preprint arXiv:1905.12247*, 2019.

Xiang Cheng and Peter L Bartlett. Convergence of Langevin MCMC in KL-divergence. *PMLR 83*, (83):186–211, 2018.

Xiang Cheng, Niladri S Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Conference On Learning Theory*, pages 300–323, 2018.

Ben Cousins and Santosh Vempala. A cubic algorithm for computing Gaussian volume. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, pages 1215–1228. Society for Industrial and Applied Mathematics, 2014.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

Persi Diaconis and David Freedman. On Markov chains with continuous state space. Technical report, 1997.

Alain Durmus, Eric Moulines, and Marcelo Pereyra. Efficient Bayesian computation by proximal Markov chain Monte Carlo: when Langevin meets Moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018.

Alain Durmus, Eric Moulines, et al. High-dimensional Bayesian inference via the unadjusted Langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019.

Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM (JACM)*, 38(1):1–17, 1991.

Andreas Eberle. Error bounds for Metropolis–Hastings algorithms applied to perturbations of Gaussian measures in high dimensions. *The Annals of Applied Probability*, 24(1):337–377, 2014.

Andreas Eberle, Arnaud Guillin, Raphael Zimmer, et al. Couplings and quantitative contraction rates for langevin dynamics. *The Annals of Probability*, 47(4):1982–2010, 2019.

Alan Frieze, Ravi Kannan, and Nick Polson. Sampling from log-concave distributions. *The Annals of Applied Probability*, pages 812–837, 1994.

Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 549–603, 1994.

Gilles Hargé. A convex/log-concave correlation inequality for Gaussian measure and an application to abstract Wiener spaces. *Probability theory and related fields*, 130(3):415–440, 2004.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Daniel Hsu, Sham Kakade, Tong Zhang, et al. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17, 2012.

Søren Fiig Jarner and Ernst Hansen. Geometric ergodicity of Metropolis algorithms. *Stochastic processes and their applications*, 85(2):341–361, 2000.

Ravi Kannan, László Lovász, and Miklós Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete & Computational Geometry*, 13(1):541–559, 1995.

László Lovász. Hit-and-run mixes fast. *Mathematical Programming*, 86(3):443–461, 1999.

László Lovász and Miklós Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *Proceedings of 31st Annual Symposium on Foundations of Computer Science, 1990*, pages 346–354. IEEE, 1990.

László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms*, 4(4):359–412, 1993.

László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM Journal on Computing*, 35(4):985–1005, 2006.

László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007.

Oren Mangoubi and Aaron Smith. Rapid mixing of Hamiltonian Monte Carlo on strongly log-concave distributions. *arXiv preprint arXiv:1708.07114*, 2017.

Oren Mangoubi and Nisheeth Vishnoi. Dimensionally tight bounds for second-order Hamiltonian Monte Carlo. In *Advances in Neural Information Processing Systems*, pages 6027–6037, 2018.

Kerrie L Mengersen, Richard L Tweedie, et al. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1):101–121, 1996.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

Sean P Meyn and Robert L Tweedie. Computable bounds for geometric convergence rates of Markov chains. *The Annals of Applied Probability*, pages 981–1011, 1994.

Hariharan Narayanan. Randomized interior point methods for sampling and optimization. *The Annals of Applied Probability*, 26(1):597–641, 2016.

Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

G Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3): 378–384, 1981.

Marcelo Pereyra. Proximal Markov chain Monte Carlo algorithms. *Statistics and Computing*, 26(4):745–760, 2016.

Natesh S Pillai, Andrew M Stuart, and Alexandre H Thiéry. Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *The Annals of Applied Probability*, 22(6):2320–2356, 2012.

Christian P Robert. *Monte Carlo methods*. Wiley Online Library, 2004.

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351–367, 2001.

Gareth O Roberts and Jeffrey S Rosenthal. Complexity bounds for MCMC via diffusion limits. *arXiv preprint arXiv:1411.0712*, 2014.

Gareth O Roberts and Osnat Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and computing in applied probability*, 4(4):337–357, 2002.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996a.

Gareth O Roberts and Richard L Tweedie. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1):95–110, 1996b.

Gareth O Roberts, Jeffrey S Rosenthal, et al. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.

Denis Talay and Luciano Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic analysis and applications*, 8(4):483–509, 1990.

Santosh Vempala. Geometric random walks: a survey. *Combinatorial and Computational Geometry*, 52(573-612):2, 2005.

Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics & Probability Letters*, 91:14–19, 2014.

Journal of Machine Learning Research 20 (2019) 1-25 Submitted 6/19; Published 11/19

Why do deep convolutional networks generalize so poorly to small image transformations?

Aharon Azulay AHARON.AZULAY@MAIL.HUJI.AC.IL
ELSC
Hebrew University of Jerusalem

Yair Weiss YWEISS@CS.HUJI.AC.IL
CSE, ELSC
Hebrew University of Jerusalem

Editor: Rob Fergus

Abstract

Convolutional Neural Networks (CNNs) are commonly assumed to be invariant to small image transformations: either because of the convolutional architecture or because they were trained using data augmentation. Recently, several authors have shown that this is not the case: small translations or rescalings of the input image can drastically change the network's prediction. In this paper, we quantify this phenomena and ask why neither the convolutional architecture nor data augmentation are sufficient to achieve the desired invariance. Specifically, we show that the convolutional architecture does not give invariance since architectures ignore the classical sampling theorem, and data augmentation does not give invariance because the CNNs learn to be invariant to transformations only for images that are very similar to typical images from the training set. We discuss two possible solutions to this problem: (1) antialiasing the intermediate representations and (2) increasing data augmentation and show that they provide only a partial solution at best. Taken together, our results indicate that the problem of insuring invariance to small image transformations in neural networks while preserving high accuracy remains unsolved.

Keywords: Machine Learning, Deep Convolutional Neural Networks, Generalization

1. Introduction

Deep convolutional neural networks (CNNs) have revolutionized computer vision. Perhaps the most dramatic success is in the area of object recognition, where performance is now described as "superhuman" (He et al., 2015). A key to the success of any machine learning method is the *inductive bias* of the method, and clearly the choice of architecture in a neural network significantly affects the inductive bias. In particular, the choice of convolution and pooling in CNNs is motivated by the desire to endow the networks with invariance to irrelevant cues such as image translations, scalings, and other small deformations (Fukushima and Miyake, 1982; Zeiler and Fergus, 2014). This motivation was made explicit in the 1980s by Fukushima in describing the "neocognitron" architecture, which served as inspiration for modern CNNs (LeCun et al., 1989). Fukushima pointed out that the fact that all layers in the neocognitron are convolutional means that the response in the final layer "is not af-

©2019 Aharon Azulay, Yair Weiss.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v20/19-519.html.

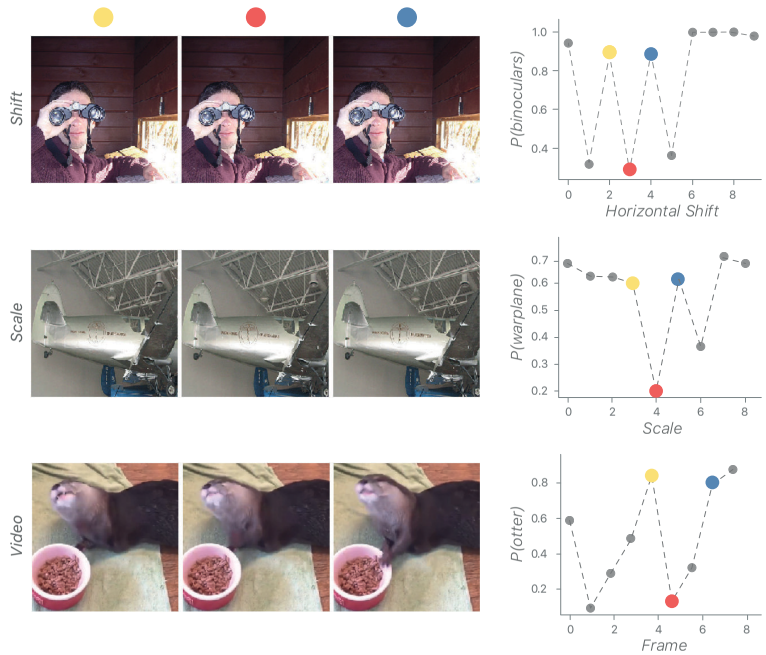


Figure 1: Examples of jagged predictions of modern deep convolutional neural networks. Top: A negligible horizontal shift of the object results in an abrupt decrease in the network's predicted score of the correct class. Middle: A tiny increase in the size of the object produces a dramatic decrease in the network's predicted score of the correct class. Bottom: A very small change in the otters posture results in an abrupt decrease in the network's predicted score of the correct class (see https://youtu.be/MpUdRacvkWk). Colored dots represent images chosen from interesting x-axis locations of the graphs on the right. These dots illustrate sensitivity of modern neural networks to small, insignificant (to a human), and realistic variations in the image.

fected by the shift in position of the stimulus pattern at all. Neither is it affected by a slight change of the shape or the size of the stimulus pattern.". Fukushima also demonstrated experimentally that the neocognitron's output is unchanged when characters are translated, rescaled or slightly deformed, even if the characters were not seen during training.

A second source of inductive bias injected to neural networks is what is known as "data augmentation". When training a CNN for object recognition, the network is presented with a crop of the original image (Simonyan and Zisserman, 2014; Huang et al., 2017): the

location and size of the crop is chosen randomly. Thus any particular input image can be seen by the network at different shifts and rescalings during training. We would therefore expect that the CNN will learn a discriminant that is invariant to resizing the image or to translating the image.

Despite these two sources of inductive bias, modern CNNs are surprisingly brittle when the input is translated, rescaled or otherwise transformed using a small image transformation. Figure 1 shows examples of such failures for the InceptionResnetV2 CNN: a one pixel shift of the image (top), or a one pixel rescaling of the image (middle), or an imperceptible change in the otter's posture, result in a dramatic change in the network's output. This is despite the fact that the network is fully convolutional and was trained using data augmentation for scalings and translations. Why does the inductive bias fail?

In this paper we address this question. We first quantify the effect systematically and show that it occurs in different architectures and when different algorithms for rescaling and translating are used. We show that the chance that a CNN output on a randomly chosen image will change after translating downward *by a single pixel* can be as high as 30%. We then address separately the two sources of inductive bias (1) the convolutional architecture and (2) the use of data augmentation and explain why they are not sufficient to achieve invariance. The convolutional architecture ignores the classical sampling theorem, so that aliasing effects make the output not invariant. Data augmentation causes the network to learn invariance only for images that are very similar to those seen during training, and since the distribution of images in the training set is highly biased, this leads to a lack of invariance on images that do not obey the bias. Finally, we address two possible solutions: antialiasing the internal representations and increasing data augmentation and show that they provide only a partial solution, at best.

2. Quantifying the lack of invariance in modern CNNs

The lack of invariance of modern CNNs to small image deformation was reported in several recent reports (Azulay and Weiss, 2018; Engstrom et al., 2017a; Zhang, 2019). How typical were the examples shown in these reports? In order to systematically quantify the effect we used 6 different CNNs and for each network we showed 1000 images with 4 different protocols.

We tested three networks from the Keras package (VGG16, ResNet50, InceptionResNetV2), and another three from the Pytorch package (VGG16, ResNet50, DenseNet121). The basic experiment we conducted is to choose a random image from the ImageNet test set, choose a one pixel perturbation using one of the four protocols described below, and then measure the network's sensitivity to that perturbation. We measured the sensitivity using two measures. The first, which we call "P(Top-1 change)" is the probability of change in the network's top-1 prediction following a one pixel perturbation. The second, which we call "mean absolute change" measures the average absolute change in the probability calculated by the network following a one pixel perturbation for the top class (i.e. the class that had highest probability in the first of the two frames). We use both measures throughout the paper since they capture different notions of stability: the first measure is invariant to any monotonic transformation of the output of final layer of the network, while the second

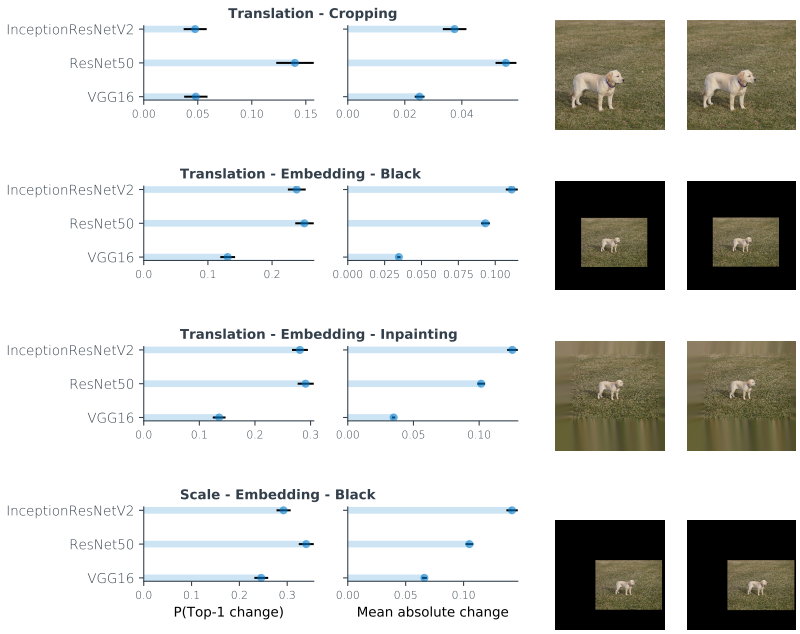


Figure 2: Quantifying the sensitivity of modern CNNs. We tested four different protocols for one pixel perturbations (pairs of images shown on the right, animations of the transformations available at `https://github.com/AzulEye/CNN-Failures/blob/master/GIFS.md`) on 1000 randomly chosen images from the ImageNet validation set. For some of the protocols, the chance that a one pixel perturbation will change the output approaches 30%.

measure helps us rule out the possibility that changes in the top-1 prediction are due to very small differences between the most likely class and the second most likely class.

In order to perturb the input image by a single pixel, we tested four different protocols. All of these protocols are based on the fact that modern CNNs expect their input to be of a particular size (typically 224x224) while the actual image can be of very different dimensions.

In the first protocol, which we call the "crop" protocol we follow the procedure described in (Szegedy et al., 2015) for training CNNs. We choose a random square within the original image and resize the square to be 224x224. The size and location of the square are chosen randomly according to the distribution described in (Szegedy et al., 2015). We then shift that square by one pixel diagonally to create a second image that differs from the first one by translation by a single pixel. In the second protocol, which we call "embedding", we downscale the image so that its minimal dimension is of size 100 while maintaining aspect ratio, and embed it in a random location within the 224x224 image, while filling in the rest of the image with black pixels. We then shift the embedding location by a single pixel, again creating two images that are identical up to a shift by a single pixel. In the third protocol, we repeat the embedding experiment but rather than filling in the rest of the image with black pixels we use a simple inpainting algorithm (each black pixel is replaced by a weighted average of the non black pixels in its neighborhood. Code available at https://github.com/AzulEye/CNN-Failures). The fourth protocol is identical to the second protocol, but rather than shifting the embedding location, we keep the embedding location fixed and change the size of the embedded image by a single pixel (e.g from size 100x100 to size 101x101 pixels).

Each of these protocols has its advantages and disadvantages. The cropping protocol by definition removes visual information that was present in the original image and could be important for discrimination. Furthermore, even though the two frames differ only by a one pixel translation, each of them has slightly different amounts of information in the border pixels. The embedding protocol has the advantage that all the information in the original images is preserved in both frames, but may give atypical appearance of the border pixels (although some images in ImageNet already have black borders). By examining the pairs of images shown in figure 2 we note the following two properties:

- In all four protocols, the difference between the two images is imperceptible to a human.
- In all four protocols, the identity of the objects in the image are recognizable to a human to the same degree as they were in the original image.

Figure 2 shows the two measures of sensitivity for the three Keras networks tested and for the four protocols (table 1 shows that similar numbers are obtained using the networks in the PyTorch package). Despite the fact that the difference between the two frames is imperceptible, the chance of a network changing its prediction can be as high as 30%. In other words, the reports of lack of invariance in previous papers were not isolated examples but can happen with high frequency.

In order to calibrate the significance of a failure probability of 0.3, note that unlike the case of adversarial examples (e.g. (Szegedy et al., 2013; Engstrom et al., 2017a)), the per-

	VGG16	**ResNet50**	**DenseNet121**
P(top-1 change)			
Crop	0.062 (0.05)	0.061 (0.14)	0.068
Black	0.18 (0.11)	0.20 (0.29)	0.19
MAC			
Crop	0.038 (0.027)	0.04 (0.058)	0.045
Black	0.06 (0.03)	0.086 (0.093)	0.079

Table 1: A comparison of the failure probability in images that were trained as part of the PyTorch package and the Keras package. The numbers show P(top1 change) and mean absolute change (MAC) for three networks trained in the PyTorch package for two of the protocols. The numbers in parenthesis are the analogous numbers for the Keras package. Although the numbers vary, the failures are common across packages and architectures.

turbation is fixed and the same for all images. We are not searching for a perturbation that will fool the network, but rather applying a fixed one-pixel perturbation and measuring the probability that it will fool the network. The fact that a fixed, imperceptible perturbation fools modern networks 30% of the time illustrates their brittleness.

A natural criticism of these results is that some of our protocols (e.g. the procedure of resizing the image and then embedding it into a larger image), was something that we introduced during testing and not during the training of the networks (we actually used *pretrained networks* which we downloaded from the Keras and Pytorch websites). In this respect, it is helpful to distinguish between two types of "invariant" recognition systems:

- A fully translation invariant recognition system will give the same output to any pattern and a translated version of that pattern. An example of such a classifier is one which bases its output only on the power spectrum of the input image.
- A partially translation invariant recognition system will give the same output to a pattern and a translated version of that pattern, *provided that pattern appeared in the training set (or was similar to a training pattern).* An example of such a classifier is an SVM that is trained with data augmentation: by construction it is invariant to translations of the training patterns, but not to translations of different patterns.

Clearly, our results indicate that modern CNNs are not fully translation invariant (using the first definition) and can give very different outputs to a pattern and a translation of that pattern by a single pixel.

3. Ignoring the Sampling Theorem

The failure of CNNs to generalize to image translations is particularly puzzling. Intuitively, it would seem that if all layers in a network are convolutional then the representation should simply translate when an image is translated. If the final features for classification are obtained by a global average pooling operation on the representation (as is done for

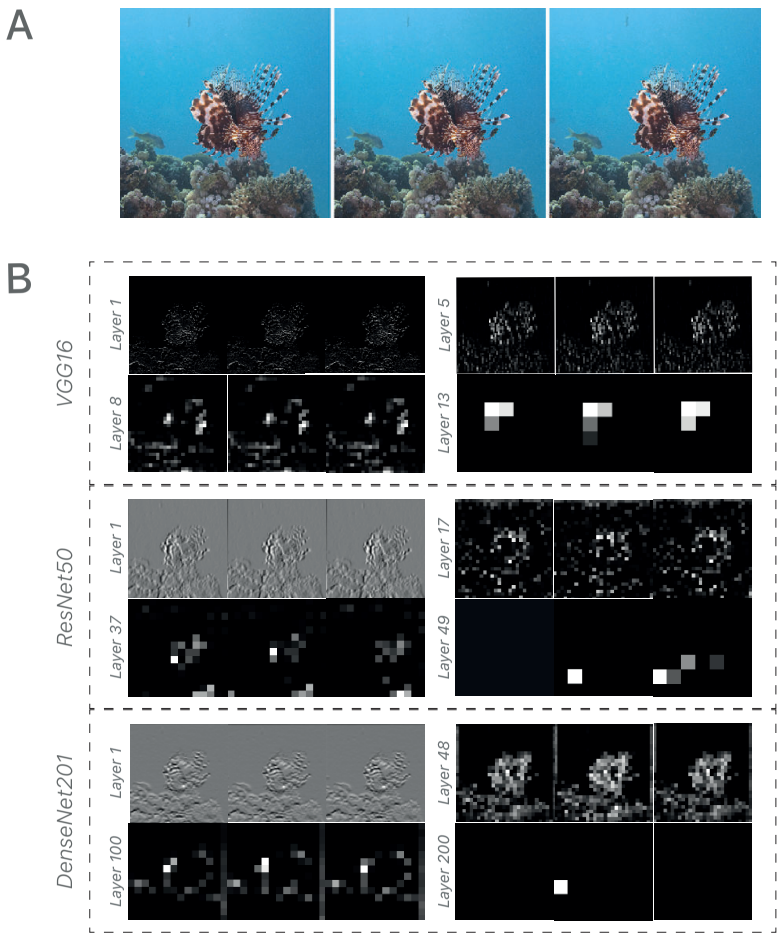


Figure 3: The deeper the layer, the less shiftable are its feature maps. A) A vertical shift of the object in the image plane. B) Examples of feature maps from three different network architectures in response to the translated image. Layer depth assignments reflect the number of trainable convolutional layers preceding the selected layer. The last layer is always the last convolutional layer in each network.

example in ResNet50 and InceptionResNetV2) then these features should be invariant to translation. Where does this intuition fail?

This intuition ignores the subsampling operation which is prevalent in modern CNNs, also known as "stride". This failure of translation invariance in systems with subsampling was explicitly discussed in (Simoncelli et al., 1992) who wrote "We cannot literally expect translation invariance in a system based on convolution and subsampling: translation of the input signal cannot produce simple translations of the transform coefficients, unless the translation is a multiple of each of the subsampling factors in the system". Since deep networks often contain many subsampling operations, the subsampling factor of the deep layers may be very large so that "literal" translation invariance only holds for very special translations. In InceptionResnetV2, for example, the subsampling factor is 60, so we expect exact translation invariance to hold only for $\frac{1}{60^2}$ of possible translations. Indeed in Zhang (2019), it was shown that invariance for modern CNNs does hold when the input pattern is shifted exactly by a multiple of the subsampling factors.

Simoncelli et al. also defined a weaker form of translation invariance, which they called "shiftability" and showed that it can hold for systems with subsampling (this is related to weak translation invariance as defined by (Lenc and Vedaldi, 2015), see also (Esteves et al., 2017; Cohen and Welling, 2014) for related ideas applied to neural networks). Here we extend the basic shiftability result to show that when shiftability holds, then global average pooling will indeed yield invariant representations.

We define $r(x)$ as the response of a feature detector at location x in the image plane. We say that this response is "convolutional" if translating the image by any translation δ yields a translation of the response by the same δ. This definition includes cases when the feature response is obtained by convolving the input image with a fixed filter, but also includes combinations of linear operations and nonlinear operations that do not include any subsampling.

We start by a trivial observation:

Observation: If $r(x)$ is convolutional then global pooling $r = \sum_x r(x)$ is translation invariant.

Proof: This follows directly from the definition of a convolutional response. If $r(x)$ is the feature response to one image and $r_2(x)$ is the feature response to the same image translated, then $\sum_x r(x) = \sum_x r_2(x)$ since the two responses are shifts of each other.

We emphasize that the claim above guarantees that a CNN where the stride is always one, will be fully translation invariant: it will give the same output to any pattern and a translated version of that pattern, regardless of whether such a pattern is similar to one that it saw during training. We now show that this can also be achieved with subsampling, provided the representations are "shiftable".

Definition: A feature detector $r(x)$ with subsampling factor s is called "shiftable" if for any x the detector output at location x can be linearly interpolated from the responses on the sampling grid:

$$r(x) = \sum_i B(x - x_i) r(x_i)$$

where x_i are located on the sampling grid for subsampling factor s and $B(x)$ is the basis function for reconstructing $r(x)$ from the samples.

The classic Shannon-Nyquist theorem tells us that $r(x)$ will be shiftable if and only if the sampling frequency is at least twice the highest frequency in $r(x)$.

Claim: If $r(x)$ is shiftable then global pooling on the sampling grid $r = \sum_i r(x_i)$ is translation invariant.

Proof: This follows from the fact that global pooling on the sampling grid is (up to a constant) the same as global pooling for all x.

$$\sum_x r(x) = \sum_x \sum_i r(x_i) B(x - x_i) \quad (1)$$

$$= \sum_i r(x_i) \sum_x B(x - x_i) \quad (2)$$

$$= K \sum_i r(x_i) \quad (3)$$

where $K = \sum_x B(x - x_i)$ and K does not depend on x_i.

While the claim focuses on a global translation, it can also be extended to piecewise constant transformations.

Corollary: Consider a set of transformations T that are constant on a set of given image subareas. If $r(x)$ is shiftable and for a given image, the support of $r(x)$ and its receptive field is contained in the same subregion for all transformations in T, then global pooling on the sampling grid is invariant to any transformation in T.

Proof: This follows from the fact that applying any transformation in T to an image has the same effect on the feature map $r(x)$ as translating the image.

To illustrate the importance of the sampling theorem in guaranteeing invariance in CNNs, consider a convolutional layer in a deep CNN where each unit acts as a localized "part detector" (this has been reported to be the case for many modern CNNs (Zeiler and Fergus, 2014; Zhou et al., 2014)). Each such part detector has a spatial tuning function and the degree of sharpness of this tuning function will determine whether the feature map can be subsampled while preserving shiftability or not. For example, consider a part detector that fires only when the part is exactly at the center of its receptive field. If there is no subsampling, then as we translate the input image, the feature map will translate as well, and the global sum of the feature map is invariant to translation. But if we subsample by two (or equivalently use a stride of two), then there will only be activity in the feature map when the feature is centered on an even pixel, but not when it is centered on an odd pixel. This means that the global sum of the feature map will *not* be invariant to translation.

In the language of Fourier transforms, the problem with a part detector that fires only when the part is exactly at the center of the receptive field is that the feature map contains many high frequencies and hence it cannot be subsampled while preserving shiftability. On the other hand, if we have a part detector whose spatial tuning function is more broad, it can be shiftable and our claim (above) shows that the global sum of activities in a feature map will be preserved for all translations, even though the individual firing rates of units will still be different when the part is centered at an odd pixel or an even pixel. Our corollary (above), shows the importance of shiftability to other smooth transformations: in this case each "part detector" will translate with a *different* translation but it is still the case that nonshiftable representations will not preserve the global sum of activities as the image is transformed, while shiftable representations will.

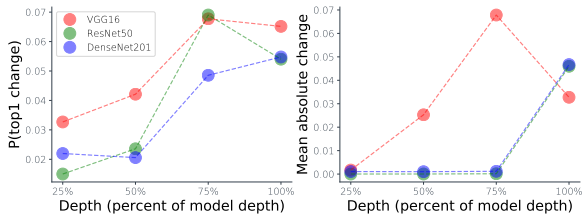


Figure 4: The deeper the representation, the higher the probability for a change in the top-1 prediction (left) and the mean absolute change (right) following a 1-pixel translation (using the cropping protocol). Shown are trained readout layers of three different CNNs at different depths (relative to each CNN).

Figure 3 examines the extent to which the representations learned by modern CNNs are invariant or shiftable. The top row shows an image that is translated vertically, while the bottom three rows show typical representations in different layers for the three CNNs we consider. For VGG16 the representations appears to shift along with the object, including the final layer where the blurred pattern of response is not a simple translation of the original response, but seems to preserve the global sum for this particular image. For the two more modern networks, the responses are sharper but lose their shiftability in the later layers. In particular, the final layers show approximate invariance to one special translation but no response at all to another translation, suggesting that the many layers of subsampling yield a final response that is not shiftable.

To gain a more quantitative measure of how shiftability changes in modern networks as a function of depth, we took the pretrained modern networks and trained a readout layer to classify ImageNet images from intermediate layers. For example, the InceptionResNetV2 network has 134 layers, but we can train a readout layer to classify images based on each of the preceding 133 layers. After training these readout layers, we measure whether a one pixel shift of the input would cause a change in the networks output which allows us to quantify the amount of translation invariance in the different layers.

Results are shown in figure 4: when we train classifiers based on the early layers in the input, the chance of a one pixel shift changing the output is below 5% but as we go deeper and deeper into the network, the subsampling operations and the nonlinearities make the representations not shiftable and the network loses its invariance.

How can we guarantee that representations in CNNs will be shiftable? As explained above, we need to make sure that any feature map that uses stride does not contain frequencies above the Nyquist frequency. If CNNs were purely linear, we could simply blur the input images so that they would not include any frequencies higher than the Nyquist limit determined by the final sampling factor of the network. But since CNNs also include nonlinearities, they can add high frequencies that were not present in the input.

Observation 2: Let $r(x)$ be a representation obtained by first subsampling an image by a factor of s and then applying a series of convolutions and pointwise nonliearities. Let

$r_2(x)$ be a representation obtained without subsampling the image, and then applying the same series of convolutions and pointwise nonlinearities, but where all convolution kernels are dilated by a factor of s. Then subsampling $r_2(x)$ by a factor of s yields $r(x)$.

Proof: This follows directly from the definition of convolution.

Corollary: Consider a layer in a CNN with stride s. For any *subsequent* layer in the CNN, $r(x)$, consider the equivalent $r_2(x)$ as defined in observation 2. If $r_2(x)$ contains frequencies above the Nyquist frequency of s, then $r(x)$ is not shiftable.

Proof: This follows from observation 2, which states that we can calculate $r(x)$ by first computing $r_2(x)$ and then subsampling by a factor of s.

Observation 2 and its corollary highlight an important difference between CNNs and linear systems. In a linear system, we can always avoid aliasing by *blurring before subsampling*. The blur removes frequencies above the Nyquist limit, so that the subsampling does not cause aliasing. Furthermore, since all subsequent operations are linear, they cannot introduce frequencies that were zeroed out by the blur, hence all subsequent representations are shiftable. But in nonlinear systems, this no longer holds. Even if we are careful to remove all frequencies above the Nyquist limit before sampling, there is no guarantee that the subsequent representations will remain shiftable.

Translated to the language of neural networks the principle of blurring before subsampling means that stride (i.e. subsampling) should always be combined with pooling (i.e. blurring) in the preceding layer. Indeed if we have an arbitrarily deep CNN where all the layers use stride=1 followed by one layer that has a stride greater than one, then by choosing the pooling window appropriately we can guarantee that the final layer will still be shiftable. If we do not use appropriate pooling then there is no guarantee that this layer will be shiftable. Even if we use appropriate pooling that ensures that a given layer is shiftable, the subsequent nonlinearities in a CNN may not preserve the shiftability, as the nonlinearities may again introduce high frequencies. The recent paper of (Zhang, 2019) reports on experiments that add blur prior to subsampling in modern CNNs (specifically they add the blur after the ReLU operation). We will examine the extent that such antialiasing helps invariance in section 5.1.

4. Why don't modern CNNs learn to be invariant from data augmentation?

While the preceding discussion suggests that the CNN architecture will not yield translation invariance "for free", there is still the possibility that the CNN will learn a translation invariant prediction from the training examples, i.e. by learning representations that satisfy the conditions for shiftability. This can be achieved by the network learning filters whose Fourier transform is predominantly concentrated in low frequencies. We would expect the network to learn such filters when they are trained using data augmentation. As mentioned previously, the input to the networks were obtained using a random crop of a training image, so that the network was equally likely to see a particular image and a one pixel translation of that image. Why doesn't this cause the network to learn a discriminant function that is invariant to one pixel translations?

One possible answer to this question is that the network can drive the training error to zero by learning a discriminant function that is only invariant to translations of the

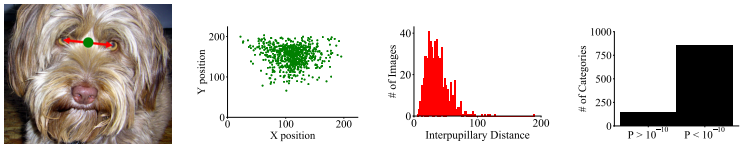


Figure 5: Photographer's biases in the ImageNet dataset. Left: Example of the hand-labeling procedure of the Tibetan terrier category. Middle: Location and Histogram of distances between the dogs eyes. Notice the bias in both the objects position and scale. Right: Quantitative results for all ImageNet categories (Chi-squared test).

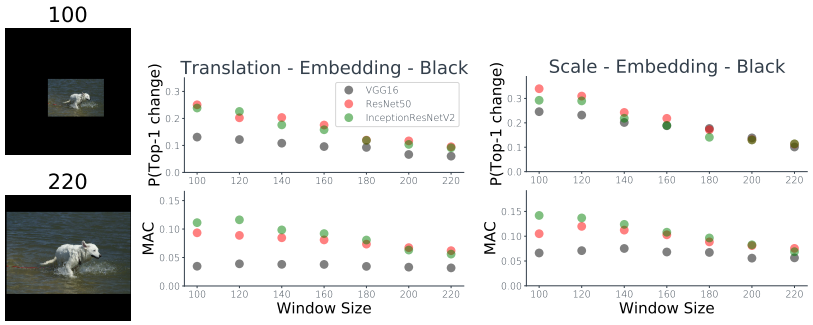


Figure 6: The effect of the embedding window size on the observed failures. When the embedding size is different from the size that the network expects, the lack of invariance increases. MAC stands for the mean absolute change measure.

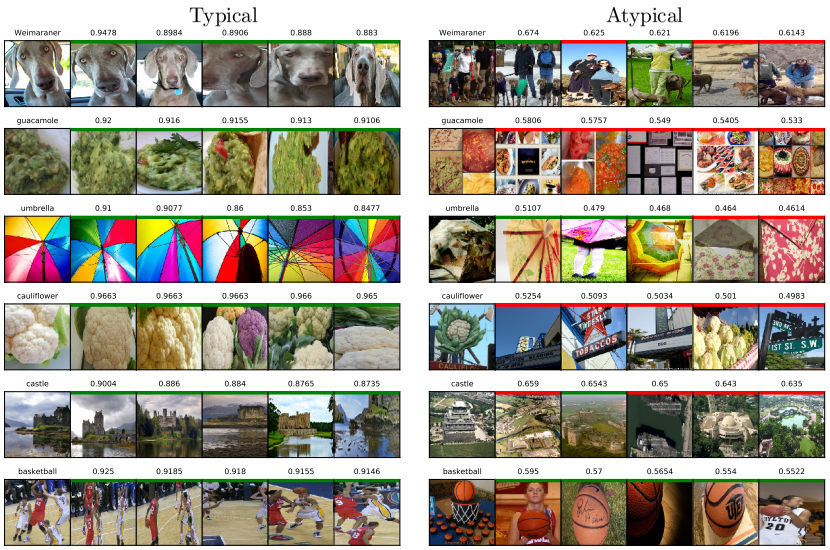


Figure 7: Our definition of typical images (i.e. ones that satisfy photographer's bias). Examples of randomly chosen, center cropped validation images, for which we calculate the normalized correlation of representation in the global-average-pooling layer of ResNet50, to the same representation of the entire ImageNet training set. Left: Images with a high mean correlation for their top 10 nearest neighbors from the training set (top 5 NN are shown on the right of each validation image, together with their correlation to the validation image). Right: The same as the left side, but with images with a low mean correlation for their top 10 nearest neighbors from the training set.

training images. There is nothing in the training loss to encourage it to learn a discriminant function that is invariant to the translation of *any input image*. Of course we expect the network to generalize what it learned to images that it did not see during training, but this generalization may be limited to images that are similar to those seen during training.

The notion of "images similar to those seen during training" is particularly relevant if the training set includes significant bias. Consistent with previous results on "dataset bias" (Simon et al., 2007; Raguram and Lazebnik, 2008; Berg and Berg, 2009; Torralba and Efros, 2011; Weyand and Leibe, 2011; Mezuman and Weiss, 2012) we find that the ImageNet dataset is extremely biased in terms of the available sizes and locations of objects. As one illustration, figure 5 shows the distribution of the locations and distances between the eyes of a "Tibetan terrier". Note that almost all images of that category show a centered closeup of the dog's face. Thus a network can drive the training error to zero, by being invariant

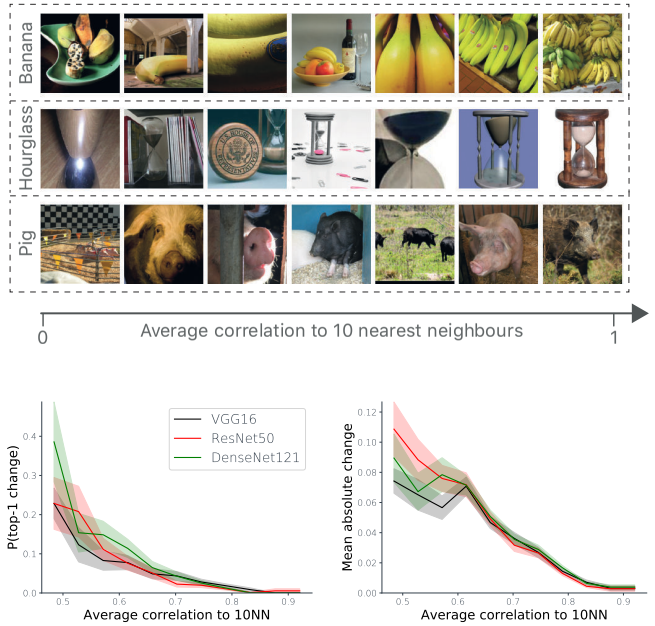


Figure 8: CNN sensitivity to one pixel perturbations depends on how typical the images are. We plot our two measures of sensitivity as a function of typicality for 1000 randomly chosen, center cropped images. Both measures of sensitivity depend strongly on the typicality of the images: when images obey the bias of photographers, sensitivity is low. Error bars represent the standard error of the mean.

to translations within that narrow range of possible sizes. If this is the case, then we would expect the learned invariance to fail once we present the network with examples of terriers of different sizes. Indeed, the difference in results in figure 2 is consistent with this view. When we use the crop protocol, then the object sizes are similar to what was observed during training, but when the embedding protocol is used, the sizes may be very different. Is the higher sensitivity in the embedding protocol due to the fact that this causes images to be less similar to the training images, or is there something intrinsic to the embedding protocol that makes networks more sensitive to perturbations?

Figure 6 summarizes experiments that suggest that the important effect is the similarity to the training images. We used the same protocol as was used in figure 2: we randomly chose images from the ImageNet validation set, resized them, and embedded them in a larger image. We systematically varied the size of the embedded images: when the embedded size is close to the input the network expects (224 for VGG16 and ResNet50,299 for InceptionResNetV2) then the resizing operation yields images that are similar to those that the network saw during training. However, when the embedded size is very different from the size that the network expects, then the resizing operation yields images that are less similar to the ones that the network saw during training. If most of the translation invariance we see in modern CNNs is due to the network learning partial invariance during its training, then we should expect the network to be more invariant for large embedded images and less invariant for small embedded images.

Figure 6 shows this intuition to be correct. For embedded sizes close to 220, a one pixel shift or scaling of an image causes a change in the network prediction approximately 10% of the time, while for embedded sizes close to 100, this probability increases to approximately 30% for the modern networks.

For a more direct test of the hypothesis that the networks only learn invariance to images that are similar to what it saw during training, we quantified the sensitivity as a function of the typicality of an input image. Typicality is defined via the "perceptual similarity" of an image to the 10 nearest neighbors in the training set, where"perceptual similarity" between the two images is the normalized correlation between the features in the penultimate layer of ResNet50. Various authors have shown that this definition of perceptual similarity is invariant to the particular network that is used to define the features (Zhang et al., 2018).

Figure 7 shows examples of typical and atypical images for various categories. When an image is taken in a way that satisfies the bias of photographers (e.g. when a dog is captured with a closeup that shows the face in the center of the frame) then the nearest neighbors have a high degree of perceptual similarity. But when when the same dog is photographed at a distance, together with several other objects, the image does not satisfy the bias of photographers and the nearest neighbors have a much lower degree of similarity.

Figure 8 shows that this measure very clearly predicts the sensitivity of the network to one pixel perturbations. When a test image has high perceptual similarity to the training images, then both measures of sensitivity approach zero. But when an image has low perceptual similarity, the average sensitivity is quite high (P(top-1 change) approaches 0.3), even though all images here use the crop protocol that was also used during data augmentation. Note also the similarity between the probabilities of failure of the three different networks: since they were trained using the same biased dataset, all three networks learn to be invariant only to images that satisfy the photographer's bias.

The similarity of an image to its most similar training examples is expected to also change the confidence of the CNN in its prediction of the image: images that have many similar neighbors should yield more confident predictions from the CNN. In the appendix we show that using one CNN's confidence also predicts the amount of invariance other CNNs will exhibit, although the effect seems weaker than when we directly use the similarity to training examples (figure 12).

5. Possible Solutions

To summarize, our results show that modern CNNs are not shift invariant and a one pixel shift of an input image can drastically change the network's output. This is despite the convolutional architecture and the use of data augmentation during training. We now discuss three possible solutions and examine the extent to which they solve the problem.

5.1. Antialiasing

As we discussed in section 3, an important principle of signal processing is that one should always blur before subsampling, and modern CNN architectures do not obey this principle. In the recent work of (Zhang, 2019), an elegant proposal was made regarding how to embody this principle in modern CNNs. Specifically, they proposed performing the pooling operation in two steps: a max-pool layer with stride 1, followed by a convolutional layer with stride=2. Thus the two layers in combination perform bluring before subsampling. They showed that by changing modern CNNs to incorporate these two layers instead of the standard max-pool layers and then retraining, they can significantly improve the invariance of networks trained on CIFAR10. However, for networks trained on Imagenet, the improvement was much smaller.

Figure 9 shows that when the antialiased networks are tested using the protocols that we investigated, then the effect of antialiasing is relatively small. In particular, when the embedding protocol is used and the size of the embedded image is different from what the network saw during training, a one-pixel shift can still cause the network to change its prediction in about 15% of the time. Similarly, the bottom of figure 9 shows the same dependence on typicality in the antialiased networks as we saw in the original networks. When images are similar to those seen during training, then the antialiased networks are invariant, but for atypical images there is still strong sensitivity to one pixel shifts (up to 20% chance that a one-pixel shift will change the top-1 prediction). Clearly, the antialiased networks do not satisfy our definition of a fully translation-invariant system, which should be invariant to the translation of any pattern.

We believe the reason for the lack of significant improvement is what we discussed in section 3: even though blurring before subsampling is sufficient for avoiding aliasing in linear systems, the presence of nonlinearities may introduce aliasing even in the presence of blur before subsampling. Table 2 shows an example: when we use the bin5 filter and a subsampling ratio of 2, a linear system is still shiftable following the subsampling. But this is *not* the case when we use a ReLU nonlinearity: despite using the same filter and the same subampling ratio, the system is no longer shiftable. In the appendix, we show a similar example when the bin5 filter is replaced with the ideal low pass filter. This means that the question of whether the system will be shiftable or not depends on the learnt

	signal1	signal2	global sum1	global sum2
Original:	(0,0,0,1,0,0,0)	(0,0,1,0,0,0,0)	1	1
Blurred:	(0,1,4,6,4,1,0)	(1,4,6,4,1,0,0)	16	16
Sampled:	(0,4,4,0)	(1,6,1,0)	8	8
ReLU (bias=5):	(0,0,0,0)	(0,6,0,0)	0	6

Table 2: When nonlinearities are present, blurring before subsampling does *not* guarantee shiftability.

weights and biases, and we cannot guarantee shiftability simply by virtue of the blurring before subsampling.

5.2. Increased Data Augmentation

Engstrom et al. (Engstrom et al., 2017b) suggested another possible solution for the lack of translation invariance: increased data augmentation. They first showed that networks that were trained using the standard data augmentation protocol were not invariant when tested with images that were resized and translated with a black background (the second protocol in our section 2). They then showed that by training the networks with additional data augmentation that also included the second protocol, they could increase the invariance.

Figure 10 shows experiments we have conducted using the CIFAR10 dataset that aims to study what CNNs learn from data augmentation. Do they learn to be invariant to the translation of any input image, or do they simply learn to be invariant for images that are visually similar to those seen during training? The results support the second alternative: data augmentation mainly causes an increase in translation invariance for images that were obtained using the same protocol that was used during training. We do not see any evidence of the network generalizing from augmented data to the general notion of translation invariance.

5.3. Reducing subsampling

The simplest way to reduce aliasing artifacts is to avoid subsampling. Unfortunately this comes with a large computational price but for small images this is possible. Figure 11 shows that when we train a subsampling-free CNN on CIFAR10 images it achieves high accuracy and is now truly translation invariant. This does not seem feasible for larger size images (although see (Chen et al., 2014, 2017; Yu and Koltun, 2015; Yu et al., 2017) for ways to make this more efficient). In these experiments, we tested the network with images that were rescaled and embedded into a black image, and the network never saw images of this size during training. Nevertheless, the convolutional architecture without subsampling guarantees translation invariance for any input pattern.

6. Discussion

CNN architectures were designed based on an intuition that the convolutional structure and pooling operations will give invariance to translations and small image deformations "for

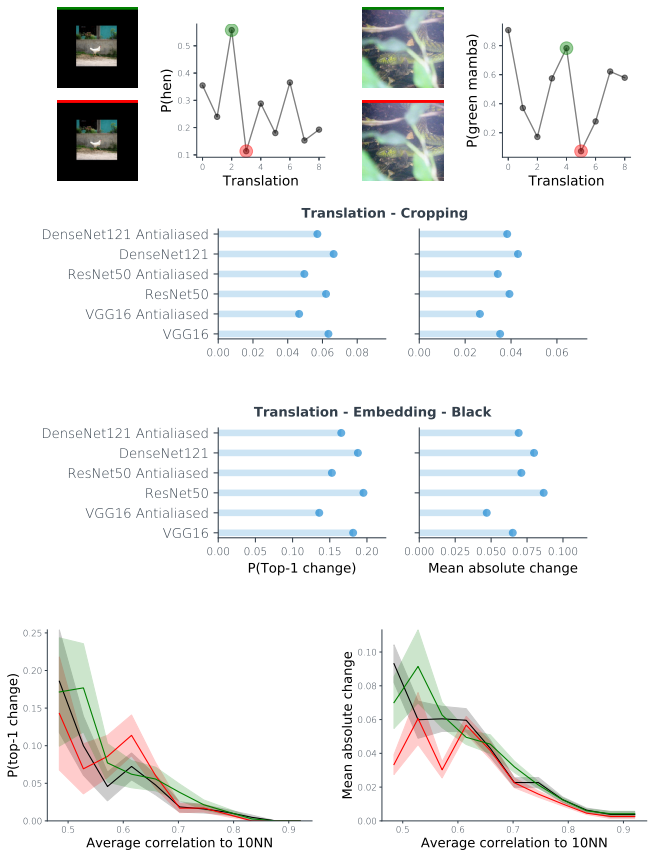


Figure 9: Antialiasing is only a partial solution. Top: Examples of jagged predictions following a 1-pixel translation in the "Black" and "Crop" protocols (see figure 13 for more examples). Middle: a comparison of the senstitivity measures for two of the protocols from figure 2 with antialiased versions of three modern CNNs ("bin-5" antialiasing taken from (Zhang, 2019)) and the non-antialiased ones. The bottom shows sensitivity as a function of image typicality when using the crop protocol. When images are not typical, antialiased networks still show high degree of sensitivity to single pixel translations.

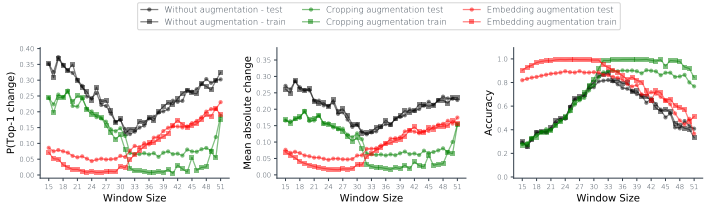


Figure 10: Possible solution - data augmentation. Plotted are two ResNet50 CNNs trained on the CIFAR-10 dataset: Gray - training without data augmentation. green - training with data that is randomly downsized and translated (black background). Red - training with data that is randomly upscaled and cropped. Left - the probability of change to the top-1 prediction following a 1-pixel translation. Middle: the mean absolute change in prediction following a 1-pixel translation. Right - Accuracy of the same networks. Notice that data augmentation is effective mainly in the range of transformations used during training.

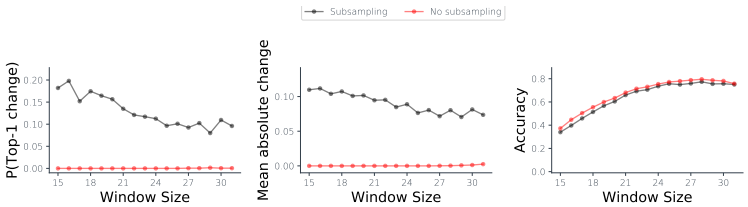


Figure 11: A CNN without subsampling is perfectly translation invariant. Plotted are two Alexnet style CNNs on the CIFAR-10 dataset: with (gray) and without (red) subsampling. The two CNNs achieve similar accuracies of about 0.8.

free". In this paper we have shown that this intuition breaks down once subsampling, or "stride" is used and we have presented empirical evidence that modern CNNs do not display the desired invariances since the architecture ignores the classic sampling theorem. This still leaves open the possibility of a CNN learning invariance from the data augmentation procedure but we have shown that this is not the case. Rather, data augmentation teaches the network to be invariant to translations but only for images that are visually similar to typical images seen during training, i.e. images that obey the photographer's bias.

In addition to pointing out these failures, the sampling theorem also suggests a way to impose translation invariance by ensuring that all representations are sufficiently blurred to overcome the subsampling. Alternatively, one could use specially designed features in which invariance is hard coded or neural network architectures that explicitly enforce invariance (Sifre and Mallat, 2013; Gens and Domingos, 2014; Cheng et al., 2016a,b; Dieleman et al., 2016, 2015; Xu et al., 2014; Worrall et al., 2017; Cohen and Welling, 2016). However, trying to guarantee invariance to transformation for *any* input pattern, may end up hurting performance in datasets that contain significant photographer's bias.

Acknowledgments

We thank Tal Arkushin for the helpful comments. Support by the ISF and the Gatsby Foundation is gratefully acknowledged.

References

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

Tamara L Berg and Alexander C Berg. Finding iconic images. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2009.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016a.

Gong Cheng, Peicheng Zhou, and Junwei Han. Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2884–2893. IEEE, 2016b.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999, 2016.

Taco S Cohen and Max Welling. Transformation properties of learned visual representations. *arXiv preprint arXiv:1412.7659*, 2014.

Sander Dieleman, Kyle W Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly notices of the royal astronomical society*, 450(2):1441–1459, 2015.

Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *arXiv preprint arXiv:1602.02660*, 2016.

Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *CoRR*, abs/1712.02779, 2017a. URL `http://arxiv.org/abs/1712.02779`.

Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017b.

Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. *arXiv preprint arXiv:1709.01889*, 2017.

Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

Robert Gens and Pedro M Domingos. Deep symmetry networks. In *Advances in neural information processing systems*, pages 2537–2545, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 991–999. IEEE, 2015.

Elad Mezuman and Yair Weiss. Learning about canonical views from internet image collections. In *Advances in Neural Information Processing Systems*, pages 719–727, 2012.

Rahul Raguram and Svetlana Lazebnik. Computing iconic summaries of general visual concepts. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1233–1240. IEEE, 2013.

Ian Simon, Noah Snavely, and Steven M Seitz. Scene summarization for online image collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

Eero P Simoncelli, William T Freeman, Edward H Adelson, and David J Heeger. Shiftable multiscale transforms. *IEEE transactions on Information Theory*, 38(2):587–607, 1992.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

Tobias Weyand and Bastian Leibe. Discovering favorite views of popular places with iconoid shift. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1132–1139. IEEE, 2011.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. Scale-invariant convolutional neural networks. *arXiv preprint arXiv:1411.6369*, 2014.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Richard Zhang. Making convolutional networks shift-invariant again. *arXiv preprint arXiv:1904.11486*, 2019.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014. URL `http://arxiv.org/abs/1412.6856`.

Appendix A.

A.1. Pipeline for producing the bottom row of figure 1

We download this video: https://youtu.be/akseo5DuXgU using an online downloader. We load the video frames and crop them to a 1x1 aspect ratio while making sure that the object is visible for the entire duration of the video. After the crop, we resize the frames to 299 by 299 as used by the standard Keras applications framework (https://keras.io/applications/). We preprocess the frames using the standard Keras preprocessing function. Finally, we use the predictions of the InceptionResNetV2 model to demonstrate the jagged behavior shown in figure 1.

A.2. The network architectures used in figure 2

Network	Top-1	Top-5	Parameters	Depth
VGG16	0.715	0.901	138,357,544	16
ResNet50	0.759	0.929	25,636,712	50
InceptionResNetV2	0.804	0.953	55,873,736	134

Table 3: The networks used (taken from (`https://keras.io/applications/`))

A.3. ReLU causes aliasing even with ideal low pass filter

Table 2 shows an example of how nonlinearity can destroy shiftability even if blurring is used prior to subsampling. This example can be extended to the case when the blur filter is the ideal low pass filter. In this case, the two blurred signals will simply be infinite signals $sinc(k/2), sinc((k-1)/2)$, whose central part is $(0, 0.6336, 1, 0.6336, 0)$. After subsampling, the first signal will be a delta function, whose central part is $(0, 0, 1, 0, 0, 0)$ and the shifted signal will be infinite with a central part of $(-0.2122, 0.6336, 0.6336, -0.2122)$. The global sum of both these signals is exactly 1, showing again that shiftability preserves the global sum. But if we apply a ReLU with any bias between 0.6336 and 1, then the first signal will remain the delta function, while the second signal will be zero everywhere.

A.4. CNN confidence can predict sensitivity

We have shown that the similarity of an image to its 10 most similar training examples can predict the sensitivity to a one pixel perturbation by different CNNs. We also expect the CNN's confidence to vary as a function of this similarity: when an image has many very similar training images, we expect the CNN's confidence in its prediction to be large. This suggests using the CNN's confidence for a given image as a predictor of how sensitive different networks will be to one pixel perturbations of that image. Figure 12 shows that this can be used as a predictor, but the effect seems to be weaker than using the similarity of an image to its most similar training examples.

A.5. More examples of failure of antialiased networks

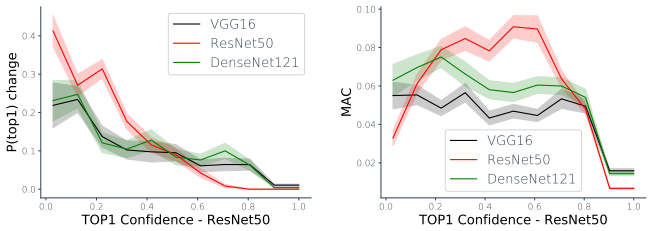


Figure 12: CNN sensitivity to one pixel perturbations depends on the confidence the network assigns to the top1 prediction. We plot our two measures of sensitivity as a function of top1 confidence for 1000 randomly chosen, center cropped images. Error bars represent the standard error of the mean.

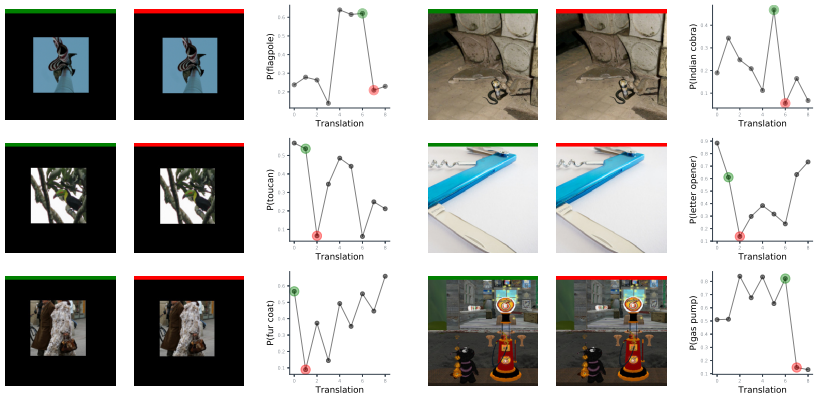


Figure 13: Additional examples of failures of antialiased networks in the "Black" and "Crop" protocols.